



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΚΠΑΙΔΕΥΤΙΚΗ ΕΦΑΡΜΟΓΗ ΒΕΛΤΙΩΣΗΣ ΤΗΣ ΟΡΘΟΓΡΑΦΙΚΗΣ ΙΚΑΝΟΤΗΤΑΣ ΓΙΑ ΜΑΘΗΤΕΣ

ΣΙΑΣΙΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ, 2117168

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Ζυγούρης Νικόλαος (Ψυχολόγος)

Λαμία 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΚΠΑΙΔΕΥΤΙΚΗ ΕΦΑΡΜΟΓΗ ΒΕΛΤΙΩΣΗΣ ΤΗΣ
ΟΡΘΟΓΡΑΦΙΚΗΣ ΙΚΑΝΟΤΗΤΑΣ ΓΙΑ ΜΑΘΗΤΕΣ

ΣΙΑΣΙΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ, 2117168

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

Ζυγούρης Νικόλαος (Ψυχολόγος)

Λαμία 2022



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

EDUCATIONAL APPLICATION FOR
STUDENTS TO IMPROVE THE SPELLING
ABILITY

SIASIOS KONSTANTINOS 2117168

FINAL THESIS

ADVISOR

Zygouris Nikolaos (Psychologist)

Lamia 2022

Εκπαιδευτική Εφαρμογή Βελτίωσης Της Ορθογραφικής Ικανότητας Για
Μαθητές

Η πτυχιακή εργασία εγκρίθηκε από:

Ζυγούρης Νικόλαος, Επίκουρος Καθηγητής

Οικονόμου Παναγιώτης, Επίκουρος Καθηγητής

Δημητρίου Γεώργιος, Επίκουρος Καθηγητής

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 11/3/2023

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Περίληψη

Το θέμα που μελετάται στην παρούσα εργασία είναι η ανάπτυξη και χρήση εργαλείων που στοχεύουν στην ενίσχυση και υποστήριξη του εκπαιδευτικού τομέα. Πιο συγκεκριμένα, μελετάται η διαδικασία ανάπτυξης εφαρμογής με κεντρική ιδέα την βελτίωση των ορθογραφικών ικανοτήτων μαθητών του δημοτικού αλλά και του τομέα της δευτεροβάθμιας εκπαίδευσης. Βάση δίνεται επίσης και στη διαχείριση και αποθήκευση ευαίσθητων δεδομένων που αφορούν τους μαθητές, δηλαδή τον τελικό χρήστη.

Βασικές προϋποθέσεις κατά την ανάπτυξη της εφαρμογής ήταν η ασφάλεια των δεδομένων αφήνοντας περιθώρια για χρήση κάποιον από αυτά για την εξαγωγή πορίσματος για την εξέλιξη των μαθητών, η ανεξαρτησία της από μελλοντικές ενημερώσεις τεχνολογιών, καθώς και η ευκολία στη χρήση τόσο από τους μαθητές και τους γονείς τους, όσο και από το διαχειριστή της, ο οποίος θα εξάγει και το τελικό πόρισμα.

Λέξεις Κλειδιά: Μαθησιακές δυσκολίες, Δυσορθογραφία, Δυσγραφία, Εφαρμογές ιστού, Εκπαιδευτικές εφαρμογές

Abstract

The subject being studied in the present paper is the development and use of various tools aimed at strengthening and supporting the educational system. More specifically, what is being studied is the development process of an application with the central idea of improving the spelling abilities of primary and secondary school students. Attention is also paid to the storage and management of sensitive data concerning the students, which are also the end users.

Key requirements during the development process of the application were the security of the data, leaving room for using a part of them to draw conclusions about the progress of the students, its independence from future updates in the technologies in use and possible feature additions, as well as ease of use by both students and their parents, and its administrator, who shall draw the final conclusion about the students' progress.

Keywords: Learning Disabilities, Spelling Disability, Dysgraphia, Web Applications, Educational Applications

Περιεχόμενα

Περίληψη	8
Abstract	14
Περιεχόμενα	15
Λίστα Εικόνων.....	20
ΚΕΦΑΛΑΙΟ 1 - Εισαγωγή	22
ΚΕΦΑΛΑΙΟ 2 - Μαθησιακές Δυσκολίες	24
Ορισμός 2.1	24
Είδη Μαθησιακών Δυσκολιών 2.2	26
Εισαγωγή 2.2.α	26
Δυσλεξία 2.2.β.....	26
Δυσαριθμησία 2.2.γ	27
Δυσγραφία 2.2.δ.....	27
Δυσορθογραφία 2.2.ε	27
Αντιμετώπιση Μαθησιακών Δυσκολιών 2.3.....	28
Εισαγωγή 2.3.α	28
Μέθοδοι αντιμετώπιση δυσκολιών στην ορθογραφία 2.3.β.....	29
Μέθοδος Gillingham.....	29
Πολύ-αισθητηριακή προσέγγιση Fernald	29
Αυτό-διόρθωση.....	29
ΚΕΦΑΛΑΙΟ 3 - Η Τεχνολογία στην Εκπαίδευση	30
Ο Ρόλος της Τεχνολογίας στην Εκπαίδευση 3.1	30
Αντίστοιχες Εφαρμογές 3.2.....	31
Dictionary.com: English Words	31
Mod Math.....	33
Πειραματική εφαρμογή για τη βελτίωση της ανάγνωσης.....	34

Πειραματική εφαρμογή για τη νευρολογική προσέγγισης της δυσαριθμησίας	35
εΜαΔύς	36
ΚΕΦΑΛΑΙΟ 4 - Ανάλυση Αναγκών Εφαρμογής	38
Εισαγωγή 4.1	38
Τεχνολογίες Ιστού 4.2	38
Back End 4.2.α.....	38
Στοιβα ανάπτυξης LAMP	38
Front End 4.2.β.....	40
HTML5	40
CSS3	41
JavaScript	41
Σχεδιασμός Βρόχου Παιχνιδιού 4.3	42
Σενάρια Χρήσης Εφαρμογής 4.4	45
Βάση Δεδομένων 4.5	46
Άλλα Χρήσιμα Εργαλεία 4.6.....	48
ΚΕΦΑΛΑΙΟ 5 - Υλοποίηση Εφαρμογής Ιστού.....	50
Σχεδιασμός 5.1	50
Προφίλ Χρηστών 5.2.....	54
Χρήσιμα Εργαλεία 5.3	57
ΚΕΦΑΛΑΙΟ 6 - Υλοποίηση Βάσης Δεδομένων	60
Κρυπτογράφηση Δεδομένων 6.1.....	60
Παράρτημα	65
Κώδικας	65
Φάκελος components	65
Footer.php	65
Navbar.php	65
Φάκελος css	68

account.css	68
Add_form.css	70
Admin_panel.css	74
End_game.css	76
grades.css	80
index.css	83
login.css.....	87
quiz.css.....	94
rules.css	97
styles.css.....	100
user_info.css	119
Φάκελος includes.....	123
Prototype.php	123
completeGame.php	123
db.info.php	124
deleteAccount.php	125
deleteQuestion.php	126
deleteQuiz.php.....	128
deleteRule.php	131
exportXLSX.php	132
fetchGrades.php.....	139
fetchQuestions.php	144
fetchQuizzes.php	147
fetchRules.php.....	148
fetchUserData.php	150
flushGrades.php.....	151
functions.php	152

loginHandler.php	173
logoutHandler.php	174
newQuestionHandler.php	174
newQuizHandler.php	180
newRuleHandler.php	182
registerHandler.php	185
setGrades.php	187
updateUserInfo.php	193
updateUserPassword.php	194
Φάκελος js	197
dragDropGame.js.....	197
fetchAdminData.js	199
formHandler.js.....	200
global.js	205
sweetAlertFunctions.js.....	207
textFormat.js.....	210
Φάκελος routes	211
account.php	211
add_question.php	221
add_quiz.php	235
add_rule.php	240
admin_panel.php	253
end_game.php.....	266
quiz.php.....	268
rules.php	283
user_info.php	288
Φάκελος story	294

Story_lines.json	294
Λοιπά αρχεία.....	295
authentication.php.....	295
header.php	303
index.php	305
Αναφορές – Πηγές	318

Λίστα Εικόνων

Εικόνα 1 Διαδικτυακή εφαρμογή Dictionary.com.	32
Εικόνα 2 Παράδειγμα χρήσης Mod Math για την επίλυση κάθετης διαίρεσης.	33
Εικόνα 3 Παράδειγμα χρήσης Mod Math για την επίλυση αλγεβρικού προβλήματος....	34
Εικόνα 4 Παιχνίδι με συρόμενες συλλαβές, στο οποίο ο παίκτης καλείται να τις ταιριάξει με τη σωστή λέξη.	35
Εικόνα 5 Παιχνίδι στο οποίο ο παίκτης καλείται να τοποθετήσει τις λέξεις σε σειρά ώστε να συμπληρώσει την ιστορία.	35
Εικόνα 6 Οπτική διάκριση μοτίβου για την επίλυση του προβλήματος.....	36
Εικόνα 7 Πρόβλημα κατά το οποίο ο παίκτης καλείται να επιλέξει την πράξη που θα εκτελέσει για να το αντιμετωπίσει.	36
Εικόνα 8 Διάγραμμα που περιγράφει το βρόχο παιχνιδιού της εφαρμογής.	43
Εικόνα 9 Διάλογος παρουσίασης παιχνιδιού.	43
Εικόνα 10 Παράδειγμα παιχνιδιού με συρόμενη απάντηση.	44
Εικόνα 11 Σελίδα μελέτης κανόνων.....	44
Εικόνα 12 Τελική οθόνη μετά την επιτυχία του παίκτη.	45
Εικόνα 13 Σχεσιολογικό διάγραμμα μεταξύ των οντοτήτων «Ερώτηση» και «Απάντηση».....	47
Εικόνα 14 Σχεσιολογικό διάγραμμα μεταξύ των οντοτήτων «Αξιολόγηση», «Κανόνες», «Ερωτηματολόγιο», «Ερώτηση» και «Χρήστης».....	48
Εικόνα 15 Διάγραμμα που περιγράφει τον τρόπο διαχωρισμού των διαχειριστών από τους απλούς χρήστες.	48
Εικόνα 16 Απεικόνιση του σχεδιασμού όλων των ειδών κουμπιών της εφαρμογής.	50
Εικόνα 17 Απεικόνιση της μπάρας περιήγησης.....	51
Εικόνα 18 Απεικόνιση του πίνακα ρυθμίσεων.....	52
Εικόνα 19 Απεικόνιση των διαφόρων κουμπιών που υπάρχουν στο σύστημα διαλόγου της εφαρμογής.....	53
Εικόνα 20 Απεικόνιση των διαφόρων κουμπιών που υπάρχουν στο σύστημα διαλόγου της εφαρμογής.....	53
Εικόνα 21 Απεικόνιση των διαφόρων κουμπιών που υπάρχουν στο σύστημα διαλόγου της εφαρμογής.....	53
Εικόνα 22 Σύστημα εγγραφής και σύνδεσης χρηστών στην εφαρμογή.....	54
Εικόνα 23 Πίνακας διαχειριστή.	55

Εικόνα 24 Δεδομένα χρήστη.....	56
Εικόνα 25 Λειτουργία εξαγωγής δεδομένων χρηστών σε αρχείο CSV.....	58
Εικόνα 26 Απεικόνιση αποτελεσμάτων του μορφοποιητή κειμένου.....	59
Εικόνα 27 Οπτική περιγραφή του αλγορίθμου AES στα στάδια της κρυπτογράφησης και αποκρυπτογράφησης.	64

ΚΕΦΑΛΑΙΟ 1 - Εισαγωγή

Παρ' όλο που η τεχνολογία αναπτύσσεται συνεχώς και οι τεχνολογίες παγκόσμιου ιστού έχουν αναπτυχθεί ραγδαία τον εικοστό πρώτο αιώνα, λίγες είναι οι εφαρμογές στην ελληνική αγορά που κάνουν χρήση αυτών για την εξέλιξη του εκπαιδευτικού τομέα.

Έχει, βέβαια, σημειωθεί πρόοδος στον εκπαιδευτικό τομέα, αλλά η πλειοψηφία των εξελίξεων σχετίζεται με την αγγλική γλώσσα. Ιδιαίτερα, με την ανάπτυξη της τηλεργασίας και της τηλεεκπαίδευσης, το κενό έχει μεγαλώσει και έχει εκδηλωθεί μια αδυναμία του εκπαιδευτικού συστήματος να ανταποκριθεί στο έπακρο στις ανάγκες των μαθητών.

Με αφορμή αυτό το κενό και στα πλαίσια της παρούσας πτυχιακής εργασίας, τέθηκε υπό ανάπτυξη μια εφαρμογή υποβοηθητικού χαρακτήρα που έχει στόχο να βοηθήσει μαθητές της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης να αναπτύξουν τις δεξιότητές τους στην ορθογραφία της ελληνικής γλώσσας. Με το πέρας της ανάπτυξης της εφαρμογής, επόμενος στόχος είναι η δοκιμή της σε ένα περιορισμένο δείγμα του πληθυσμού και η εξαγωγή στατιστικών δεδομένων που θα αφορούν την αποτελεσματικότητα της.

Σε ένα εγχείρημα τέτοιου μεγέθους όμως, είναι δεδομένο ότι θα υπάρξουν δυσκολίες και περιορισμοί κατά την διαδικασία της ανάπτυξης. Το παρόν δεν αποτελεί εξαίρεση στον κανόνα, αφού σε αρκετά στάδια της διαδικασίας εμφανίστηκαν εμπόδια, άλλα μεγαλύτερου και άλλα μικρότερου μεγέθους. Μία κύρια δυνατότητα που θα περιείχε η εφαρμογή από την απαρχή της ιδέας είναι αυτή της αφήγησης. Δηλαδή η δυνατότητα για το μαθητή να μπορεί να έχει μία ακουστική περιγραφή της ερώτησης και των κανόνων που εμπεριέχονται στην εφαρμογή. Αν και υπάρχουν αυθύπαρκτα εργαλεία ενσωματωμένα στους σύγχρονους φυλλομετρητές, είναι εξαρτώμενα από τις γλώσσες που ο τελικός χρήστης έχει εγκατεστημένες στο σύστημά του. Για λόγους οικουμενικότητας, λοιπόν, και για την επίλυση του προβλήματος αυτού έγινε χρήση βιβλιοθηκών τρίτων που δίνουν τη δυνατότητα χρήσης γλωσσών της Google ως εργαλεία. Ένα ακόμη σημείο που προξένησε δυσκολίες ήταν η διεπαφή χρήστη. Στόχοι κατά το σχεδιασμό της εφαρμογής ήταν η ευκολία χρήσης, η οπτική ιεραρχία μεταξύ των συνθετικών, αλλά και η αποφυγή της απόσπαση της προσοχής των χρηστών από τα σημαντικά σημεία της. Τελική κατάληξη του σχεδιασμού, είναι ένα μονόχρωμο, ουδέτερο θέμα σε αποχρώσεις του γκρι, με χρωματικές λεπτομέρειες σε σημαντικά

στοιχεία όπως κουμπιά, υπερσυνδέσμους και απαντήσεις στις ερωτήσεις των αξιολογήσεων. Τέλος, λόγω της πολυπλοκότητας και της μοναδικότητας της εφαρμογής, πολλά συστήματα, που αφορούν κυρίως το κομμάτι της αξιολόγησης των μαθητών, αναπτύχθηκαν από τη βάση τους, αυξάνοντας εκθετικά το φόρτο εργασίας.

ΚΕΦΑΛΑΙΟ 2 - Μαθησιακές Δυσκολίες

Ορισμός 2.1

Με τον όρο «Μαθησιακές δυσκολίες» αναφερόμαστε στο σύνολο διαταραχών που μειώνουν τις μαθησιακές και επικοινωνιακές ικανότητες ενός ατόμου. Είναι βέβαια δύσκολο να δοθεί ένας καταληκτικός ορισμός και έχει υπάρξει σύγχυση στο παρελθόν γύρω από αυτόν τον όρο. Μέχρι τα τέλη της δεκαετίας του 1950, διάφοροι ήταν οι όροι που χρησιμοποιούνταν από τους επιστήμονες της εποχής για να περιγράψουν τις δυσκολίες αυτές, που αργότερα θα ενταχθούν στο φάσμα των μαθησιακών δυσκολιών. (Berk, R.A., 1983). Στις αρχές της δεκαετίας του 1960, και πιο συγκεκριμένα το 1962, ο Samuel Kirk προτείνει για πρώτη φορά τον όρο «Μαθησιακές Δυσκολίες» και, μέσω της Εθνικής Συμβουλευτικής Επιτροπής των ΗΠΑ, λίγα χρόνια αργότερα καταφέρνει να εισάγει στο νομοσχέδιο την εκπαίδευση όλων των παιδιών με ειδικές ανάγκες.

Σύμφωνα με το Kirk, μαθησιακές δυσκολίες είναι οι διαταραχές που παρουσιάζονται στην κατανόηση και τη χρήση του προφορικού ή/και του γραπτού λόγου. Προσθέτει ότι τα συμπτώματα αυτά δεν μπορούν να αποδοθούν σε συναισθηματικές διαταραχές, σε αισθητηριακές βλάβες ή σε ελλειπείς συνθήκες εκπαίδευσης, αλλά οφείλονται σε άλλους παράγοντες, κυρίως ψυχολογικούς. Έτσι, οι μαθησιακές ικανότητες των μαθητών με μαθησιακές δυσκολίες περιορίζονται σημαντικά, με αποτέλεσμα να τους είναι απαραίτητο ένα κατάλληλα διαμορφωμένο πρόγραμμα που θα τους καλύπτει τις εκπαιδευτικές τους ανάγκες. (Παπαγεωργίου, Δ., 2019).

«Έχω χρησιμοποιήσει το όρο ‘μαθησιακές δυσκολίες’ για να περιγράψω ένα σύνολο παιδιών που έχουν διαταραχές στην ανάπτυξη της γλώσσας, της ομιλίας, της ανάγνωσης και δεξιοτήτων, συνυφασμένων με την ικανότητα της επικοινωνίας, που είναι απαραίτητες για την κοινωνική αλληλεπίδραση.» αναφέρει ο Kirk και συνεχίζει, «Στο σύνολο αυτό δεν περιλαμβάνω παιδιά που έχουν αισθητηριακές αναπηρίες όπως τύφλωση ή κώφωση, αφού έχουμε μεθόδους διαχείρισης και εκπαίδευσης των τυφλών και των κωφών. Αποκλείω επίσης από το σύνολο τα παιδιά που έχουν γενικευμένη νοητική υστέρηση.» (Kirk 1963 στο NASET 2022)

Κάποιοι από τους επικρατέστερους ορισμούς είναι οι εξής:

1. Ειδική Μαθησιακή Δυσκολία σημαίνει μία ανωμαλία σε μια ή περισσότερες βασικές ψυχολογικές εξελίξεις που εμπεριέχεται στην κατανόηση ή την χρήση της γλώσσας, προφορικής ή γραπτής, στην οποία

η ανωμαλία πιθανόν να εκδηλωθεί στο άτομο με ατελή ικανότητα να ακούσει, να μιλήσει, να γράψει, να συλλαβίσει ή να κάνει μαθηματικούς υπολογισμούς. Ο όρος περιλαμβάνει τέτοιες καταστάσεις όπως αντιληπτικές ανικανότητες, εγκεφαλικό τραύμα, ελάχιστη εγκεφαλική δυσλειτουργία, δυσλεξία και αναπτυξιακή αφασία. Ο όρος δεν περιλαμβάνει παιδιά με μαθησιακές δυσκολίες που είναι κυρίως αποτέλεσμα οπτικών, ακουστικών ή κινητικών ανωμαλιών ή νοητικής καθυστέρησης ή συναισθηματικής διαταραχής ή περιβαλλοντικού πολιτιστικού ή οικονομικού μειονεκτήματος. (USOE, 1977 : 65083 στο Mercer, C.D. & Jordan, L. & Allsopp, D.H. & Mercer, A.R., 1996.)

2. Μαθησιακές Δυσκολίες είναι ένας γενικός όρος που αναφέρεται σε μια ετερογενή ομάδα διαταραχών που εκδηλώνονται με συγκεκριμένες δυσκολίες στην απόκτηση και χρήση ακουστικών, προφορικών, αναγνωστικών ικανοτήτων και ικανοτήτων γραφής, σκέψης και μαθηματικών. Αυτές οι ανωμαλίες είναι ενδογενείς στο άτομο και θεωρείται ότι οφείλεται σε δυσλειτουργία του Κεντρικού Νευρικού Συστήματος. Ακόμα κι όταν μια μαθησιακή δυσκολία μπορεί να συμβεί μαζί με άλλες αναπηρικές καταστάσεις (π.χ. αντιληπτική βλάβη, νοητική καθυστέρηση, κοινωνική και συναισθηματική διαταραχή) ή περιβαλλοντικές επιρροές (π.χ. πολιτιστικές διαφορές, ανεπαρκή-ακατάλληλη καθοδήγηση, ψυχογενείς παράγοντες), δεν είναι το άμεσο αποτέλεσμα εκείνων των καταστάσεων ή επιρροών. (Hammill, Leigh, McNutt & Larsen, 1981 : 336 στο Swanson, H.L., 1991)

3. Η Ειδική Μαθησιακή Δυσκολία ορίζεται ως μία νευρολογική διαταραχή η οποία εκδηλώνεται με «εξαιρετικές δυσκολίες στην απόκτηση και χρήση δεξιοτήτων συνδεδεμένων με την ακοή, την ομιλία, την ανάγνωση, τη γραφή, τη λογική ή τα μαθηματικά ... εγγενών στο άτομο, θεωρείται ότι οφείλονται στη δυσλειτουργία του κεντρικού νευρικού συστήματος και μπορεί να εμφανιστεί στη διάρκεια της ζωής του ατόμου». (NJCD, 1998, p.1 στο Woodcock, 2013)

Παρ' όλο που οι όροι «μαθησιακές δυσκολίες» και «μαθησιακές διαταραχές» θεωρούνται από πολλούς αλληλένδετες, οι δύο έννοιες διαφέρουν κατά πολύ. Οι

μαθησιακές διαταραχές αναφέρονται σε προβλήματα μάθησης σε κάποιον από τους ακαδημαϊκούς τομείς αλλά δεν αποτελούν αυτές καθ' αυτές αίτιο για μία επίσημη διάγνωση. Από την άλλη πλευρά, οι μαθησιακές δυσκολίες αποτελούν επίσημη διάγνωση από κάποιον αρμόδιο φορέα. Για τη διάγνωση αυτή, το εν λόγω άτομο θα πρέπει να πληροί κάποια προκαθορισμένα, από τον υπεύθυνο για τη διάγνωση, κριτήρια.

Είδη Μαθησιακών Δυσκολιών 2.2

Εισαγωγή 2.2.α

Υπάρχουν διάφοροι τύποι μαθησιακών διαταραχών και χωρίζονται ανάλογα με τον τομέα στον οποίο εμφανίζονται. Ο τομέας αυτός μπορεί να είναι ο λόγος, γραπτός ή προφορικός, η ανάγνωση, τα μαθηματικά αλλά και η λογική. Πολλοί από τους μαθητές που αντιμετωπίζουν μαθησιακές δυσκολίες, παρουσιάζουν ελλειμματική πορεία όσον αφορά την ανάπτυξη των μαθησιακών τους ικανοτήτων, ενώ το νοητικό τους επίπεδο συνήθως δεν είναι χαμηλό, γεγονός που προβληματίζει την επιστημονική κοινότητα.

Στην ελληνική πραγματικότητα, γνωστότερη και πιο συνυφασμένη με τις μαθησιακές δυσκολίες είναι η δυσλεξία, η οποία σχετίζεται με τον προφορικό αλλά και το γραπτό λόγο.

Δυσλεξία 2.2.β

Η αναπτυξιακή δυσλεξία ορίζεται ως μια απροσδόκητη ειδική και επίμονη αποτυχία απόκτησης αποτελεσματικών δεξιοτήτων ανάγνωσης παρά τη συμβατική διδασκαλία, την επαρκή νοημοσύνη και τις κοινωνικοπολιτιστικές ευκαιρίες. Η αναπτυξιακή δυσλεξία επηρεάζει το 5 έως 15% του μαθητικού πληθυσμού. Διαγιγνώσκεται όταν οι δεξιότητες γραμματισμού ενός παιδιού είναι σημαντικά χαμηλότερες από εκείνες που αναμένονται από την ηλικιακή του ομάδα. Μία από τις μεγαλύτερες δυσκολίες που αντιμετωπίζουν οι δυσλεξικοί είναι να μεταφράζουν αποτελεσματικά τα γραπτά γράμματα (γραφήματα) στους αντίστοιχους ήχους τους (φωνήματα), μια αδυναμία που περιγράφεται ως φωνολογικό έλλειμμα. (Zygouris, Avramidis, Karapetsas & Stamoulis, 2017). Υπολογίζεται ότι σχετίζεται με τη συντριπτική πλειοψηφία των περιπτώσεων ατόμων με μαθησιακές δυσκολίες, καθιστώντας την μία από τις πιο συνήθεις και μελετημένες μαθησιακές δυσκολίες. Με την πάροδο των χρόνων και την συνεχή μελέτη από τους επιστήμονες, μπορεί πλέον να αντιμετωπιστεί από τη σκοπιά της ιατρικής με διάφορες μεθόδους για την αντιμετώπισή της. (Shaywitz 1998 : 307).

Δυσαριθμησία 2.2.γ

Η Δυσαριθμησία είναι μια νευροαναπτυξιακή διαταραχή που επηρεάζει την ικανότητα του παιδιού να μάθει αριθμητική, παρά τη φυσιολογική νοημοσύνη, τη σωστή σχολική φοίτηση, το κατάλληλο περιβάλλον, την κοινωνικοοικονομική κατάσταση, τη συναισθηματική σταθερότητα και τα απαραίτητα κίνητρα. Τα παιδιά με δυσαριθμησία χαρακτηρίζονται από δυσκολίες στη μάθηση και στη μνήμη αριθμητικών γεγονότων, καθώς και από διαφορές στις διαδικασίες υπολογισμού των στελεχών και κακές στρατηγικές επίλυσης προβλημάτων. Επιπλέον, οι μαθητές με δυσαριθμησία μπορεί να παρουσιάζουν δυσκολίες στην κατανόηση των απλών εννοιών αριθμών, να μην έχουν διαισθητική κατανόηση των αριθμών και να έχουν προβλήματα στην εκμάθηση αριθμητικών γεγονότων και διαδικασιών. (Zygouris, Vlachos, Dadaliaris, Oikonomou, Stamoulis, Vavougiou, Nerantzaki & Striftou, 2017)

Δυσγραφία 2.2.δ

Η δυσγραφία είναι η διαταραχή της γραπτής έκφρασης. Συχνά σχετίζεται με την κινητική δημιουργία της γραφής, τη διαμόρφωση των γραμμάτων και την ταχύτητα της γραφής, καθώς και με την οργάνωσή της στο χώρο. Ένα άτομο διαγνωσμένο με δυσγραφία μπορεί να εμφανίσει αδυναμία συντονισμού των μυών του χεριού ώστε να μπορεί να παράγει γραπτό λόγο, αδυναμία οργάνωσης της θεματολογίας ή και αδυναμία στις σημασιολογικές του ικανότητες. Μπορεί επίσης να συνδεθεί με άλλες μαθησιακές δυσκολίες όπως τη διαταραχή ελλειμματικής προσοχής (ADHD), τη διαταραχή ομιλίας ή άλλες αναπτυξιακές διαταραχές. Το εν λόγω άτομο, συνηθίζει να αδυναμεί να διατηρήσει το σχήμα και το μέγεθος των γραμμάτων του όταν δεν έχει οπτική επαφή με το γραπτό του, ενώ μπορεί να αντιμετωπίζει και άλλες δυσκολίες στην καθημερινότητά του σε λεπτές κινητικές δεξιότητες. (Giordano & Maiorana, 2015; Gkeka & Drigas, 2022; Wikipedia, 2021)

Δυσορθογραφία 2.2.ε

Η δυσορθογραφία αποτελεί την κύρια μαθησιακή δυσκολία που το παρόν εγχείρημα καλείται να αντιμετωπίσει. Στην περίπτωση της δυσορθογραφίας, οι πάσχοντες παρουσιάζουν αδυναμία της ανάκλησης της σωστής ορθογραφίας των λέξεων που επιθυμούν να αποτυπώσουν. Ένα άλλο σύμπτωμα που μπορεί να υποδηλώνει την παρουσία της πάθησης αυτής είναι η αδυναμία ανάκλησης της καταληκτικής ορθογραφίας των λέξεων. Εκτός από τους ορθογραφικούς, συμπεριλαμβάνονται και οι γραμματικοί κανόνες σε εκείνους στους οποίους τα άτομα με δυσορθογραφία

δυσκολεύονται να κατανοήσουν και εν συνεχεία να ανακαλέσουν όταν χρειαστεί. Έτσι, βρίσκονται αντιμέτωποι με δυσκολίες στον διαχωρισμό των λέξεων σε λογικές κατηγορίες. Η παραγωγή λόγου ατόμων με δυσορθογραφία συχνά δεν μπορεί να γίνει κατανοητή από έναν τρίτο αναγνώστη, λόγω των βασικών λαθών που συνδέονται με αυτή τη μαθησιακή δυσκολία. Μεταξύ άλλων, η παράληψη ή προσθήκη γραμμάτων, συλλαβών ή και ολόκληρων λέξεων, η χρήση ομόηχων λέξεων στη θέση των σωστών αντίστοιχών τους, η αντιστροφή των γραμμάτων και η κακή δομή του γραπτού λόγου γενικότερα στη μορφή πολύ μικρών ή πολύ μεγάλων κενών μεταξύ των λέξεων, αλλά και μη συμμόρφωση με τη στοίχιση που έχει τεθεί από το μέσο στο οποίο παράγεται ο λόγος (φύλλο τετραδίου). (Kuure, L., Kuure, O., Sandbäck, A.M., Yliherva, A., 1992).

Αντιμετώπιση Μαθησιακών Δυσκολιών 2.3

Εισαγωγή 2.3.α

Λάθη στα πλαίσια μαθησιακών δυσκολιών, εκτός από την ικανότητά τους να μάθουν, μπορεί να επηρεάσουν και την ψυχολογία των μαθητών. Κοινωνικές δυσκολίες που μπορεί να παρουσιάζονται στους μαθητές μεταξύ άλλων είναι η δυσκολία να εκφράσουν τις απόψεις τους για διάφορα θέματα, η πεποίθηση ότι δεν είναι ευφυείς λόγω της δυσκολίας τους στον προφορικό και γραπτό λόγο και κατά συνέπεια, η δυσκολία τους να κοινωνικοποιηθούν. Για τους λόγους αυτούς οι μαθησιακές δυσκολίες είναι αναγκαίο να αντιμετωπίζονται από όσο το δυνατό πιο νωρίς, πριν γιγαντωθούν οι συνέπειες που μπορεί να προκαλέσουν στην καθημερινότητα του ατόμου.

Διάφορες μέθοδοι έχουν αναπτυχθεί για την αντιμετώπιση των μαθησιακών δυσκολιών, κάθε μία εκ των οποίων απαιτεί διαφορετικό χρονικό διάστημα για να λάβει δράση. Οι Βλάχου, Λίλα και Μυρσιάδη (2019) διακρίνουν στο κείμενό τους επτά τομείς στους οποίους έχουν αναπτυχθεί μέθοδοι θεραπείας και έχουν ως εξής:

1. Φωνολογική Ενημερότητα
2. Ανάγνωση
3. Γραφή
4. Ορθογραφία
5. Οπτική Διάκριση
6. Κατανόηση του Κειμένου
7. Αποκατάσταση Μνήμης

Μέθοδοι αντιμετώπιση δυσκολιών στην ορθογραφία 2.3.β

Όσον αφορά τον τομέα της ορθογραφίας, ιδιαίτερα δύσκολη είναι η αντιμετώπιση των δυσκολιών, αφού κάθε μαθητής μπορεί να παρουσιάσει δυσκολίες σε διαφορετικά φάσματα και σε διαφορετικά πλήθη αυτών, ανάλογα με το εκπαιδευτικό και κοινωνικό περιβάλλον του. Δεν μπορεί επομένως να ξεχωρίσει μία καταληκτική μέθοδος που να λειτουργεί οικουμενικά. Μερικές από τις γνωστότερες μεθόδους αναλύονται παρακάτω.

Μέθοδος Gillingham

Η μέθοδος Gillingham είναι μία επαναληπτική μέθοδος που δίνει βάση στην εξάσκηση του μαθητή με λέξεις στις οποίες διακρίνεται αντιστοιχία φωνημάτων. Η δυσκολία της μεθόδου και των προκλήσεων που καλείται ο μαθητής να αντιμετωπίσει αυξάνονται σταδιακά, έως ότου ο μαθητής να κατακτήσει την ικανότητα να παράγει προτάσεις με λέξεις που διδάχθηκαν κατά τη διάρκεια της θεραπείας.

Πολύ-αισθητηριακή προσέγγιση Fernald

Κατά την προσέγγιση αυτή, ο μαθητής καλείται να κάνει χρήση πληθώρας των αισθήσεων του εμπλέκει οπτική, κιναισθητική, ακουστική και απτική αίσθηση. Αρχικά, ο καθηγητής παρουσιάζει τη λέξη στον μαθητή, γραπτά και προφορικά. Έπειτα, ο μαθητής αναλύει και σχηματίζει τη λέξη εικονικά αλλά και στο χαρτί, δημιουργώντας έτσι μία αίσθηση μνήμης. Στο τελευταίο βήμα της τεχνικής, ο μαθητής καλείται να αποτυπώσει τη λέξη στο χαρτί από τη μνήμη του και, σε περίπτωση που η λέξη έχει γραφτεί σωστά, να την τοποθετήσει σε μία στοίβα όπου αποθηκεύει όλες τις λέξεις που κατάφερε να ορθογραφήσει.

Αυτό-διόρθωση

Όπως υποδηλώνει και το όνομά της, στη μέθοδο αυτή ο μαθητής αναλαμβάνει να διορθώσει ο ίδιος τα λάθη του, αρχικά γράμμα-γράμμα και στη συνέχεια, όταν πλέον αναπτύξει επαρκείς στρατηγικές αυτό-διόρθωσης, λέξεων και προτάσεων. (Φώτης Παπαναστασίου, 2017 στο Βλάχου, Λίλα & Μυρσιάδη, 2019)

ΚΕΦΑΛΑΙΟ 3 - Η Τεχνολογία στην Εκπαίδευση

Ο Ρόλος της Τεχνολογίας στην Εκπαίδευση 3.1

Ραγδαία ανάπτυξη έχει επέλθει την τεχνολογία τις τελευταίες δεκαετίες και πολλές πτυχές της έχουν συζητηθεί και αναλυθεί σε βάθος. Ο ρόλος της στον εκπαιδευτικό τομέα είναι πολύ σημαντικός και η αξία της έχει αναγνωριστεί σε εθνικό επίπεδο, με εφαρμογές όπως το Scratch, της γλώσσας προγραμματισμού που διδάσκεται ήδη από τις τάξεις του Γυμνασίου, αλλά και το διερμηνευτή της ΓΛΩΣΣΑΣ, μίας ψευδογλώσσας που δημιουργήθηκε για την ένταξη των μαθητών στον προγραμματισμό και τις έννοιές του. Σε μία έρευνα τους, οι Γιαβρίμης, Παπάνης, Νεοφώτιστος και Βαλκάνος (2010) βρήκαν ότι η πλειοψηφία των συμμετεχόντων εκπαιδευτικών έχει θετική στάση σχετικά με την ένταξη της τεχνολογίας στην εκπαίδευσή της και τη συμβολή της σε αυτή. Αναφέρουν βέβαια ότι η τεχνολογία θα πρέπει να χρησιμοποιείται ως βοηθητικό μέσο και όχι για την αντικατάσταση των εκπαιδευτικών και του ήδη υπάρχοντος συστήματος. Και φυσικά, η τεχνολογία από μόνη της δεν μπορεί να έχει καλό ή κακό χαρακτήρα και αντίκτυπο στην εκπαίδευση. Όπως αναφέρει και ο Hoyles (1992), οι αλλαγές που η τεχνολογία επιτάσσει στην κοινωνία και την εκπαίδευση εξαρτώνται από τον τρόπο χρήσης της.

Παρόλο που υπάρχουν και χρησιμοποιούνται εφαρμογές τέτοιου είδους στο ελληνικό εκπαιδευτικό σύστημα, είναι εμφανές ότι πρόκειται για εφαρμογές που στοχεύουν στη διευκόλυνση της διδασκαλίας εννοιών του εκπαιδευτικού προγράμματος και όχι στην αντιμετώπιση υπαρχόντων προβλημάτων, όπως οι μαθησιακές δυσκολίες. Σκοπός του παρόντος εγχειρήματος είναι η εισαγωγή ενός εργαλείου με τις παραπάνω προσδοκίες στον εκπαιδευτικό τομέα. Το ιδανικό σενάριο, αλλά και ο απώτερος σκοπός, είναι η τοποθέτηση των βάσεων για τον σταδιακό κορεσμό της αγοράς από τέτοιου είδους εργαλεία, που εξυπηρετούν, όχι μόνο τη διδακτική διαφόρων εκπαιδευτικών εννοιών, αλλά και τη βελτίωση της ποιότητας ζωής των μαθητών.

Στα πλαίσια της παρούσας εφαρμογής, η δυσορθογραφία βρίσκεται στο επίκεντρο και καλείται να μελετηθεί και κατά το δυνατό να αντιμετωπιστεί από τη σκοπιά της τεχνολογίας. Όσον αφορά τις δυσκολίες των μαθητών με δυσορθογραφία, το σύστημα αξιολόγησης είναι εμπλουτισμένο με ερωτήσεις καθαρής ορθογραφίας, βασισμένες στους κανόνες ορθογραφίας της νέας ελληνικής γλώσσας, ερωτήσεις που κάνουν χρήση ομόηχων λέξεων που συχνά συγχέονται από τους μαθητές, ερωτήσεις με καταλήξεις λέξεων αλλά και άλλα είδη ερωτήσεων που στοχεύουν στην αντιμετώπιση των επιμέρους

συμπτωμάτων της δυσκολίας αυτής. Έχει επίσης εφαρμοστεί ένα σύστημα χρωματισμού των απαντήσεων ώστε να μπορέσει ο εκάστοτε μαθητής να συνδυάσει τις σωστές απαντήσεις στη μνήμη του με χρώματα και να βοηθηθεί τοιουτοτρόπως η ικανότητά του να ανακαλεί τη σωστή ορθογραφία των λέξεων.

Αντίστοιχες Εφαρμογές 3.2

Η ιδέα της χρήσης εφαρμογών ιστού ή για έξυπνες συσκευές στον εκπαιδευτικό τομέα δεν είναι καινούρια και επομένως η παρούσα εφαρμογή δεν προέκυψε με παρθενογένεση, αλλά από την ανάγκη για τέτοιες εφαρμογές στην ελληνική εκπαίδευση.

Μερικές από αυτές τις εφαρμογές είναι οι εξής:

Dictionary.com: English Words

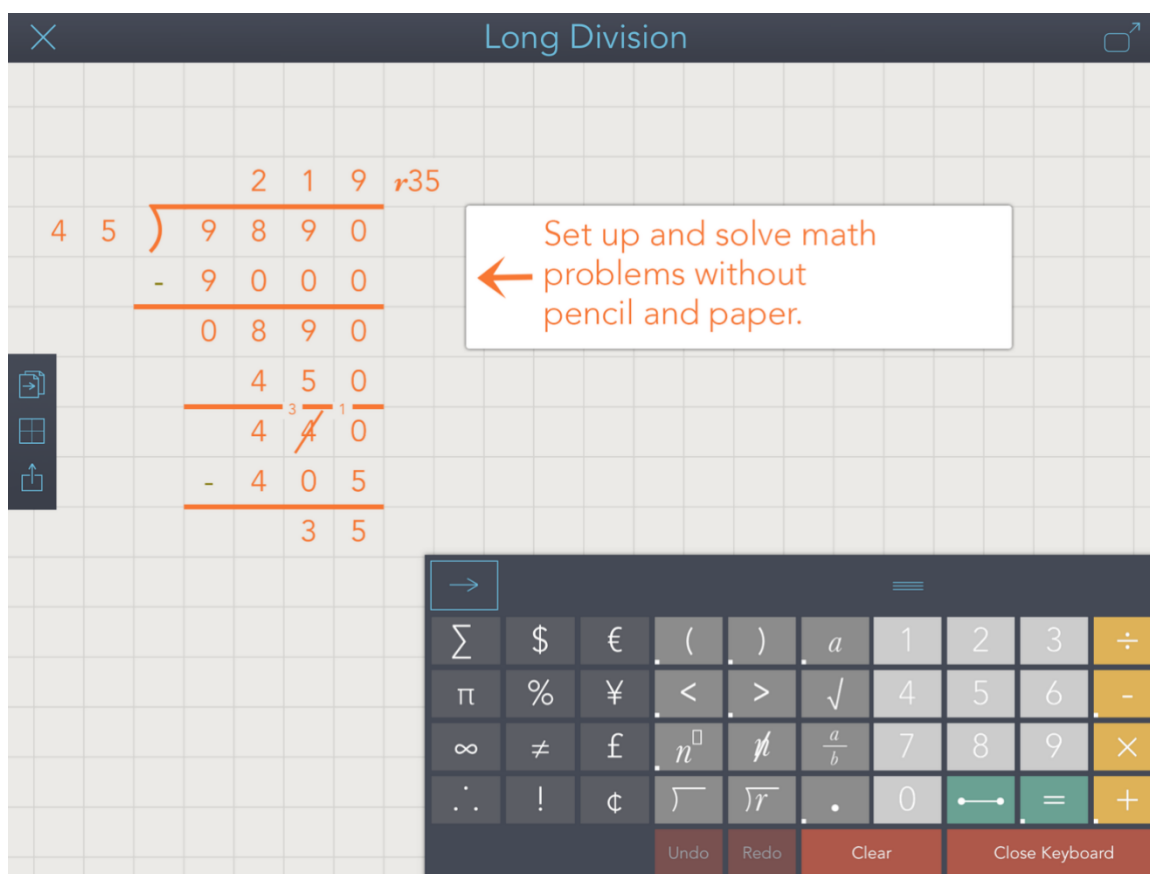
Το Dictionary.com είναι μία από τις δημοφιλέστερες εφαρμογές ψηφιακών λεξικών στον κόσμο που παρέχει εκατομμύρια αγγλικούς ορισμούς, ορθογραφία, προφορές λέξεων, παραδείγματα χρήσης των λέξεων σε προτάσεις και ετυμολογίες. Μέσα από την ιστοσελίδα του, το Dictionary.com προσφέρει επίσης λειτουργίες εκμάθησης αγγλικών λέξεων με ημερήσια παιχνίδια όπως τις λέξεις της ημέρας, τα συνώνυμα της ημέρας και άλλα παιχνίδια με λέξεις.

Το Dictionary.com διατίθεται επίσης και σε κινητές συσκευές Android και iOS, γεγονός που το καθιστά ένα πανίσχυρο, φορητό εργαλείο που βοηθά στην εκμάθηση της αγγλικής γλώσσας και στη βελτίωση του λεξιλογίου επ' αυτής. (Dictionary.com, 2022)

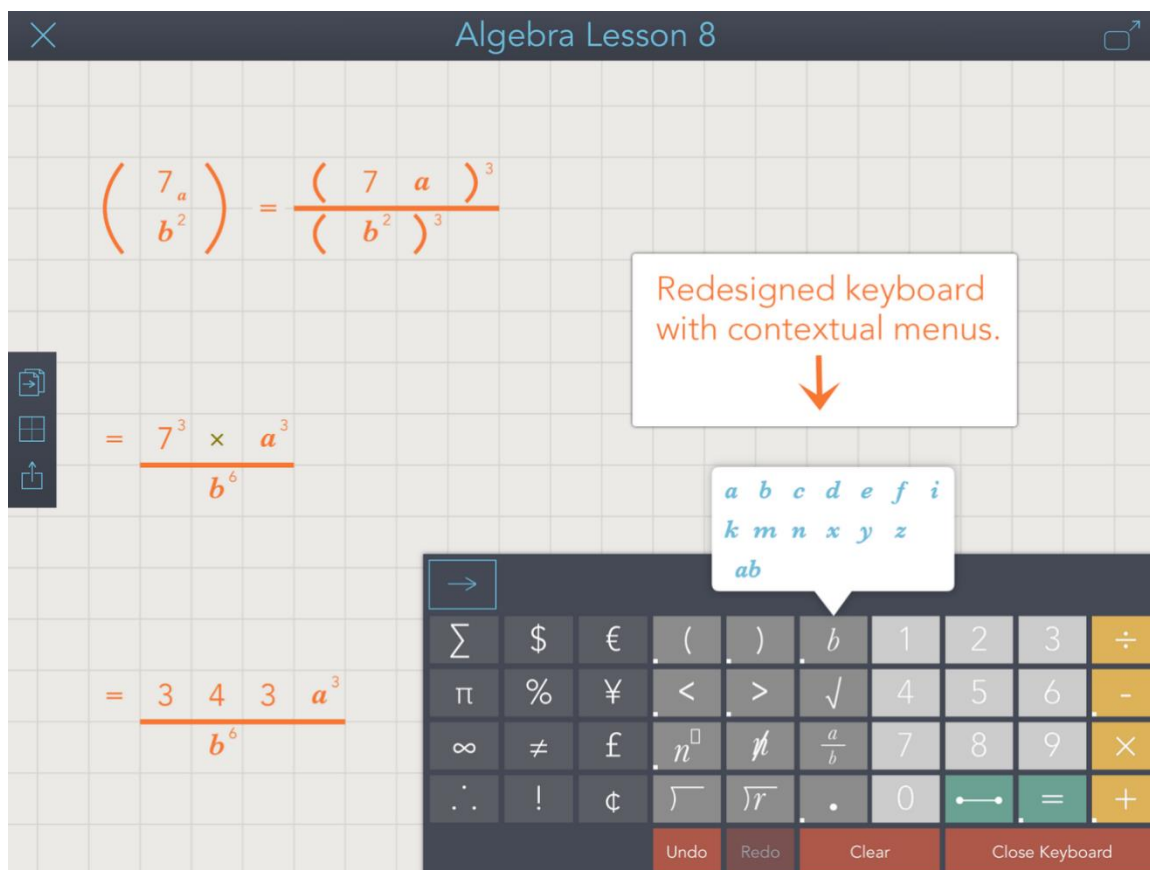
Εικόνα 1 Διαδικτυακή εφαρμογή Dictionary.com.

Mod Math

Σύμφωνα με τους δημιουργούς του Mod Math, για κάποιους, το πιο δύσκολο κομμάτι στα μαθηματικά είναι να γράψουν το πρόβλημα. Έτσι, ανέπτυξαν την πρώτη δωρεάν εφαρμογή μαθηματικών για συσκευές iOS με σκοπό να βοηθήσουν μαθητές με δυσλεξία και δυσγραφία να ασχοληθούν με τα μαθηματικά. Η εφαρμογή διασφαλίζει ότι ο μαθητής δεν θα χρειαστεί παραδοσιακούς τρόπους γραφής για να λύσει μαθηματικά προβλήματα, απαλείφοντας έτσι από την εξίσωση την δυσκολία κατά τη διαδικασία της γραφής. Επίσης, εκτός από τα βασικά μαθηματικά προβλήματα, το Mod Math μπορεί να χειρίζεται σύνθετες αλγεβρικές εξισώσεις όπως τον πολλαπλασιασμό, τη διαίρεση αλλά και τις δευτεροβάθμιες εξισώσεις.



Εικόνα 2 Παράδειγμα χρήσης Mod Math για την επίλυση κάθετης διαίρεσης.



Εικόνα 3 Παράδειγμα χρήσης Mod Math για την επίλυση αλγεβρικού προβλήματος.

Πειραματική εφαρμογή για τη βελτίωση της ανάγνωσης

Στα πλαίσια μίας έρευνας για τη βελτίωση της ικανότητας της ανάγνωσης, οι Στριφτού, Ζυγούρης, Σταμούλης και Βαβουγιός (2020) ανέπτυξαν μία εφαρμογή πιλοτικού χαρακτήρα, ονόματι «Poke the Reading Ability» (PtRA), με σκοπό να βοηθήσουν τους επικείμενους αναγνώστες να βελτιώσουν την ταχύτητα, την ακρίβεια και την κατανόηση του λόγου. Η εφαρμογή διαθέτει γραφικό περιβάλλον και διάφορα

παιχνίδια μέσω των οποίων ο χρήστης εξασκεί τις αναγνωστικές του ικανότητες, ενώ έχει αναπτυχθεί με γνώμονα τον ιστό με τεχνολογίες όπως η HTML, η CSS και η JavaScript.



Εικόνα 4 Παιχνίδι με συρόμενες συλλαβές, στο οποίο ο παίκτης καλείται να τις ταιριάζει με τη σωστή λέξη.

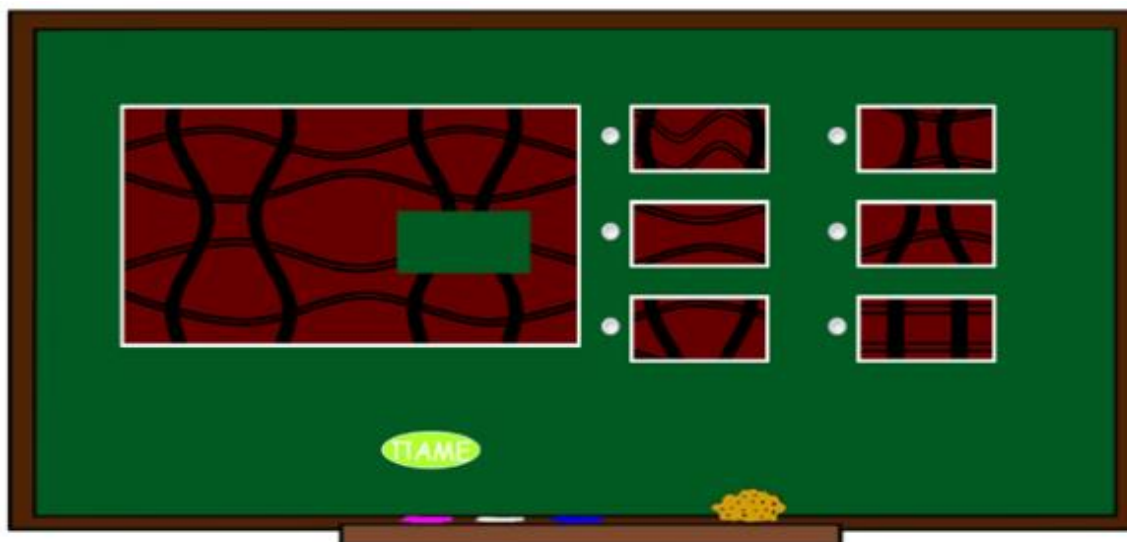


Εικόνα 5 Παιχνίδι στο οποίο ο παίκτης καλείται να τοποθετήσει τις λέξεις σε σειρά ώστε να συμπληρώσει την ιστορία.

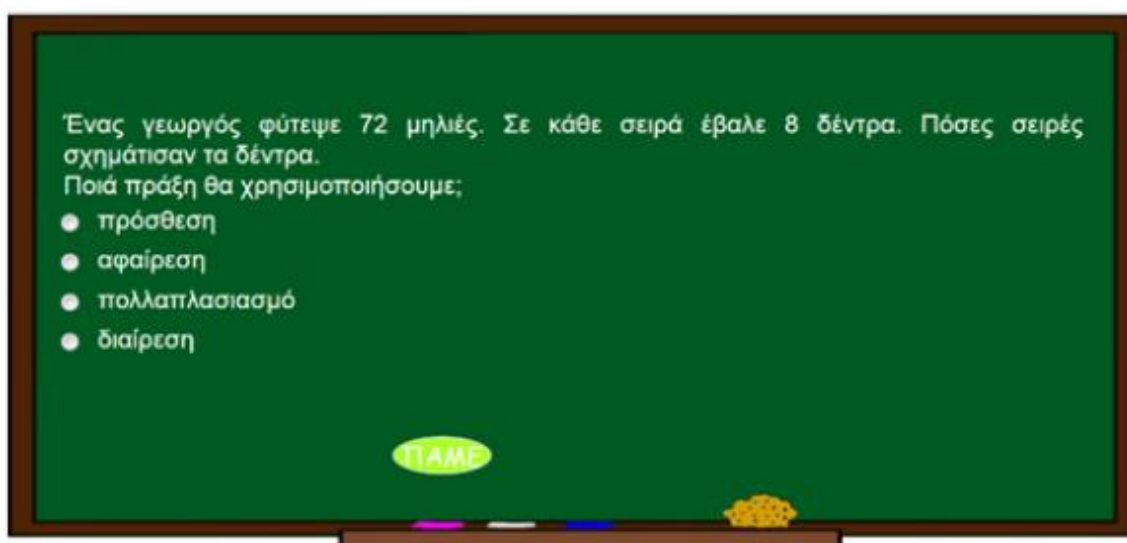
Πειραματική εφαρμογή για τη νευρολογική προσέγγισης της δυσαριθμησίας

Υποστηρικτικό χαρακτήρα σε μία έρευνα υπό την αιγίδα του Πανεπιστημίου Θεσσαλίας (Ζυγούρης, Βλάχος, Δαδαλιάρης, Οικονόμου, Σταμούλης, Βαβουγιός, Νεραντζάκη & Στριφτού 2017) κατέχει η εφαρμογή αυτή, κατά την χρήση της οποίας ο παίκτης καλείται να συμμετέχει σε διάφορα παιχνίδια που σκοπό έχουν να βοηθήσουν

στην κατανόηση, και τελικά στη βελτίωση των ικανοτήτων του στα μαθηματικά. Όπως παρουσιάζεται και παρακάτω (Εικόνες 6 & 7), τα παιχνίδια αντιμετωπίζουν τη δυσαριθμησία τόσο από τη σκοπιά της μνήμης, όσο και από τη σκοπιά της κατανόησης του δοθέντος προβλήματος και της ικανότητάς του να αναγνωρίζει μοτίβα σε αυτά.



Εικόνα 6 Οπτική διάκριση μοτίβου για την επίλυση του προβλήματος.



Εικόνα 7 Πρόβλημα κατά το οποίο ο παίκτης καλείται να επιλέξει την πράξη που θα εκτελέσει για να το αντιμετωπίσει.

εΜαΔύς

Μία από τις προσπάθειες ανάπτυξης λογισμικού για τον τομέα των μαθησιακών δυσκολιών αποτελεί και το εΜαΔύς, το οποίο παρότι δεν διατίθεται πλέον, είναι άξιο αναφοράς. Το εΜαΔύς είναι ένα λογισμικό που αναπτύχθηκε από το Ινστιτούτο

Επεξεργασίας του Λόγου (ΙΕΛ) με σκοπό τον αυτόματο εντοπισμό ύπαρξης μαθησιακών δυσκολιών στον εξεταζόμενο μαθητή. Όπως αναφέρουν στα κείμενά τους για το 9ο Συνέδριο Πανελλήνιου Συλλόγου Λογοπεδικών οι Πρωτόπαπας, Σκαλούμπακας και Νικολόπουλος (2003), το εΜαΔύς δίνει βάση στην αξιολόγηση των γνωστικών και αισθητηριακών δεκτών του μαθητή, δημιουργώντας ένα προφίλ για την κατάταξη της ειδικής μαθησιακής δυσκολίας που έχει ή όχι ο εξεταζόμενος χρήστης. Η αξιολόγηση αυτή γίνεται με τη χρήση οκτώ παιχνιδιών-ασκήσεων, που εξετάζουν μεταξύ άλλων την αναγνωστική ταχύτητα και κατανόηση, την ορθογραφία και τη φωνολογική επεξεργασία και μνήμη. Φροντίδα του ΙΕΛ ήταν επίσης η δυνατότητα επέκτασης της εφαρμογής για χρήση της και σε μαθητές διαφορετικών ηλικιών, χωρίς την ανάγκη για περεταίρω ανάπτυξη ή αλλαγή του λογισμικού κατά το δυνατό.

ΚΕΦΑΛΑΙΟ 4 - Ανάλυση Αναγκών Εφαρμογής

Εισαγωγή 4.1

Πριν από την ανάπτυξη της εφαρμογής, για την αποφυγή χρονοβόρων διορθώσεων κατά το επόμενο στάδιο, πρέπει να γίνει προετοιμασία των εργαλείων και ανάλυση των αναγκών της εφαρμογής. Στο κεφάλαιο αυτό, αναλύεται η διαδικασία της προετοιμασίας και της ανάλυσης αυτής.

Τεχνολογίες Ιστού 4.2

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η στοίβα τεχνολογιών γνωστή ως LAMP (Linux, Apache, MySQL, PHP) για το μέρος του διακομιστή και HTML5, CSS3 και JavaScript για το οπτικό και λειτουργικό μέρος. Η επιλογή έγινε με βάση τόσο τις τεχνολογίες που διδάσκονται στο προπτυχιακό πρόγραμμα σπουδών του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Θεσσαλίας, όσο και με τις τεχνολογίες που βρίσκονται χρόνια στο χώρο και θεωρούνται από πολλούς πιο σταθερές για την ανάπτυξη ιστοσελίδων.

Το οπτικό και λειτουργικό κομμάτι της εφαρμογής αναπτύχθηκε σε απλή HTML5, CSS3 και JavaScript, χωρίς δηλαδή τη χρήση τρίτων βιβλιοθηκών και πακέτων επέκτασης λειτουργιών, ώστε να είναι ανεξάρτητη από μελλοντικές αναβαθμίσεις και αλλαγές που θα μπορούσαν να υποστούν ανάλογες βιβλιοθήκες. Χρήση τέτοιων βιβλιοθηκών έγινε για λειτουργίες χαμηλής σημασίας όπως η εκτύπωση λαθών κατά τη λειτουργία της εφαρμογής (SweetAlert2), η ανάγνωση των κειμένων (ResponsiveVoice) και τα εικονίδια που κοσμούν διάφορα εξαρτήματα που χρησιμοποιούνται στη σελίδα (IonIcon).

Back End 4.2.α

Στοιβα ανάπτυξης LAMP

Ως LAMP ορίζεται το πακέτο ή στοίβα λογισμικών που χρησιμοποιείται για την ανάπτυξη εφαρμογών ιστού. Είναι μία από τις γνωστότερες και πιο συνήθεις στοίβες για το σκοπό αυτό, ενώ τα αρχικά του όρου υποδηλώνουν τις τεχνολογίες που βρίσκονται σε αυτή. Ονομαστικά είναι οι:

- Linux
- Apache HTTP Server

- MySQL
- PHP

Linux

Τα Linux είναι ένα λειτουργικό σύστημα που είναι βασισμένο στον πυρήνα του Unix και αναπτύχθηκε από τον Linus Torvalds στις αρχές της δεκαετίας του 1990. Τα Linux είναι ανοιχτού κώδικα και διανέμονται συνήθως ως εκδόσεις (distributions ή distros), κάτι που τα καθιστά άκρως ευέλικτα όσον αφορά την εμφάνισή τους και την εμπειρία χρήσης αλλά ταυτόχρονα και πολύ ασφαλή σαν λειτουργικό σύστημα, αφού οποιοσδήποτε μπορεί να ελέγξει την ορθότητα του κώδικα που βρίσκεται πίσω από την εμπειρία χρήσης και το γραφικό περιβάλλον που πολλές εκδόσεις προσφέρουν. Ανάμεσα στις πιο δημοφιλείς εκδόσεις Linux βρίσκονται τα Ubuntu, Debian, Arch Linux, Kali Linux, Red Hat και Linux Mint. (Wikipedia contributors, 2001).

Apache HTTP Server

Σύμφωνα με την ομάδα τεκμηρίωσης του The Apache Software Foundation (Documentation Group, χ.χ.), το δημιουργό και εκδότη του, ο Apache HTTP Server Project είναι μια προσπάθεια ανάπτυξης και διατήρησης ενός διακομιστή HTTP ανοιχτού κώδικα για σύγχρονα λειτουργικά συστήματα, συμπεριλαμβανομένων των UNIX και Windows. Ο στόχος αυτού του έργου είναι να παρέχει έναν ασφαλή, αποτελεσματικό και επεκτάσιμο διακομιστή που παρέχει υπηρεσίες HTTP σε συγχρονισμό με τα τρέχοντα πρότυπα HTTP. Η πρώτη έκδοση του διακομιστή κυκλοφόρησε το 1995 και είναι ο πιο δημοφιλής διακομιστής ιστού στο Διαδίκτυο από τον Απρίλιο του 1996.

MySQL

Η MySQL είναι μια μεγάλη και συχνά πολύπλοκη συλλογή λογισμικού πελάτη και διακομιστή για αποθήκευση και ανάκτηση δεδομένων. (Suehring, 2002). Ανήκει στη γενικότερη κατηγορία των Συστημάτων Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS) και βασίζεται στη γλώσσα SQL, τον κύριο τρόπο επικοινωνίας του διακομιστή με τη βάση δεδομένων. (Lawton, 2005).

Αρχικά, η MySQL έγινε ευρέως δημοφιλής λόγω της ταχύτητας και της απλότητάς της. Υπήρχε όμως και κριτική, επειδή δεν είχε χαρακτηριστικά όπως συναλλαγές και υποστήριξη ξένου κλειδιού, βασικό χαρακτηριστικό των σχεσιακών βάσεων. Πλέον παρέχει τα χαρακτηριστικά αυτά αλλά και πολλά περισσότερα, αφού συνεχίζει και

αναπτύσσεται από τη μητρική της εταιρία αλλά και την κοινότητά της, όντας λογισμικό ανοιχτού κώδικα. Αυτές οι δυνατότητες μεταφέρουν τη MySQL στη σφαίρα των εταιρικών εφαρμογών. Ως αποτέλεσμα, έχει καταφέρει να εδραιωθεί στον επαγγελματικό χώρο αποτελώντας ένα ισχυρό εργαλείο στη σουίτα λογισμικού πολλών εταιριών. (DuBois, 2008).

PHP

Η PHP είναι μια γλώσσα προγραμματισμού σεναρίων ενεργειών (scripting language) που έχει σχεδιαστεί ειδικά για εφαρμογές ιστού και εκτελείται στην πλευρά του διακομιστή. Από την απαρχή της στα μέσα της δεκαετίας του 1990, έχει εδραιωθεί στο χώρο της ανάπτυξης ιστοσελίδων, αφού επιτρέπει στον προγραμματιστή να δημιουργεί δυναμικούς ιστοτόπους. Ο κώδικας που γράφεται σε PHP ερμηνεύεται από το διακομιστή και δημιουργείται η έξοδος σε HTML την οποία μπορεί ο φυλλομετρητής να παρουσιάσει στο χρήστη. Ο τελικός χρήστης δεν έχει πρόσβαση στον κώδικα της PHP, σε αντίθεση με τη JavaScript ή την HTML που οκ χρήστης μπορεί ελεύθερα να δει και να επεξεργαστεί. Μέχρι τις αρχές της δεύτερης χιλιετίας, η PHP είχε αποκτήσει βασικό ρόλο σε πάνω από εννέα εκατομμύρια ιστοτόπους παγκοσμίως. Όπως και οι τεχνολογίες που προαναφέρθηκαν, έτσι και η PHP είναι ανοιχτού κώδικα, κάτι που τη βοηθά να αναπτυχθεί ραγδαία αλλά και να καταστήσει δημοφιλή την απασχόληση με την ανάπτυξη ιστοσελίδων, δυναμικών με περισσότερο περιεχόμενο από ποτέ. Το ακρωνύμιο αρχικά σήμαινε Personal Home Page, αλλά αργότερα άλλαξε και έφτασε στη σημερινή του σημασία, PHP Hypertext Preprocessor. (Lerdorf & Tatroe, 2002; Welling & Thomson, 2003).

Front End 4.2.β

HTML5

Η HTML, ή Γλώσσα Σήμανσης Υπερκειμένου στα ελληνικά (Hyper-Text Markup Language), είναι η γλώσσα που χρησιμοποιείται για τη δημιουργία της βασικής δομής κάθε ιστοσελίδας στον Παγκόσμιο Ιστό. Αποτελείται από «ετικέτες» που περικλείουν το περιεχόμενο της σελίδας και καθορίζουν το σκοπό και τη λειτουργία του εκάστοτε υπομέρους της σελίδας. Μερικές από αυτές τις ετικέτες έχουν καθαρά δομικό χαρακτήρα, χωρίς να εξυπηρετούν άλλο σκοπό εκτός από την οργάνωση του περιεχομένου. Άλλες προσφέρουν ειδικές λειτουργίες στον προγραμματιστή, όπως η ετικέτα εικόνας () για την προσθήκη εικόνων, η ετικέτα υπερσυνδέσμου (<a>) για την προσθήκη υπερσυνδέσμων που οδηγούν σε άλλο αρχείο του ιστοτόπου ή του διαδικτύου, ή την

ετικέτα εισόδου (<input>) για την λήψη ανάδρασης από τον τελικό χρήστη. Η HTML έκανε την εμφάνισή της το 1989 φέροντας την υπογραφή του Tim Berners-Lee, στον οποίο και αποδίδεται η δημιουργία του παγκόσμιου ιστού. Η δημιουργία του είχε σκοπό να διευκολύνει τους ερευνητές ανά τον κόσμο στο διαμοιρασμό δεδομένων ηλεκτρονικά. Στον Berners-Lee αποδίδονται επίσης και η σύλληψη της ιδέας του υπερσυνδέσμου αλλά και η μορφοποίηση των διευθύνσεων του παγκόσμιου ιστού (www.domain.top-level-domain). Έκτοτε, η HTML αποτελεί τον δομικό λίθο για κάθε σελίδα του διαδικτύου και συνεχίζει να εξελίσσεται και να εμπλουτίζεται με νέα χαρακτηριστικά και λειτουργίες. (Hoy, 2011)

CSS3

Η CSS (Cascading Style Sheets) είναι ένας απλός μηχανισμός για την προσθήκη στυλ, όπως γραμματοσειρές, χρώματα και διάστημα, σε έγγραφα ιστού και αναπτύχθηκε από το World Wide Web Consortium (W3C) το 1996. (Cascading Style Sheets, χ.χ.). Σύμφωνα με τον McFarland (2012), η CSS δίνει στον προγραμματιστή δημιουργικό έλεγχο στη διάταξη και τη σχεδίαση των ιστοσελίδων. Μέχρι την ανάπτυξη της CSS, οι σχεδιαστές-προγραμματιστές ήταν περιορισμένοι στον τρόπο δομής που προσέφερε η HTML. Αποτέλεσμα αυτού ήταν η ανάπτυξη ιστοσελίδων με μηδαμινή δημιουργικότητα, συμβατική δομή και όχι ιδιαίτερα ελκυστικά χρώματα. Με την είσοδο της CSS στη σκηνή των τεχνολογιών ιστού, οι προγραμματιστές πλέον μπορούν να οργανώνουν το περιεχόμενο της εκάστοτε σελίδας κατά παραγγελία. Διάφορα εφέ κατά την αλληλεπίδραση του χρήστη με τα στοιχεία της σελίδας μπορούν να εφαρμοστούν, ώστε να παρέχουν στο χρήστη μια πλούσια εμπειρία χρήσης. Ένα άλλο χαρακτηριστικό της CSS είναι το γεγονός ότι είναι απλή στη χρήση της και διευκολύνει τον προγραμματιστή τόσο στην εκμάθησή της, όσο και στη χρήση της στην καθημερινότητά του.

JavaScript

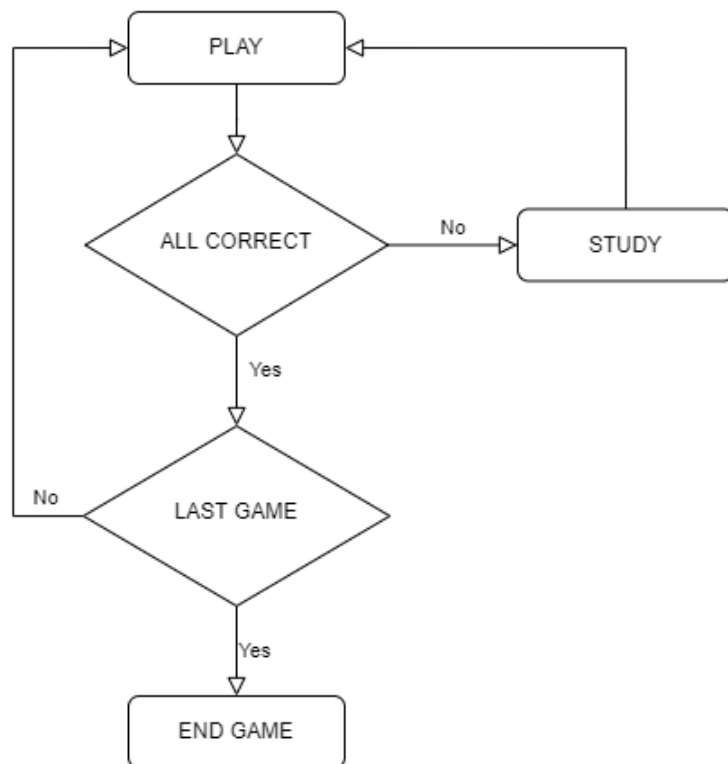
Η JavaScript είναι μία γλώσσα προγραμματισμού που αρχικά δημιουργήθηκε για να «δώσει ζωή τις ιστοσελίδες». Τα προγράμματα σε αυτή τη γλώσσα ονομάζονται σενάρια (scripts) και μπορούν να γραφτούν είτε απευθείας μέσα σε κώδικα HTML μιας ιστοσελίδας ή σε ξεχωριστά αρχεία και να εκτελούνται αυτόματα κατά τη φόρτωση της σελίδας. Τα σενάρια παρέχονται ως απλό κείμενο, ενώ δεν χρειάζονται κάποια ειδική προετοιμασία ή μεταγλώττιση για να εκτελεστούν.

Σήμερα, η JavaScript μπορεί να εκτελεστεί όχι μόνο στο φυλλομετρητή, αλλά και στον διακομιστή και γενικότερα σε οποιαδήποτε συσκευή διαθέτει μηχανή JavaScript, ένα ειδικό πρόγραμμα που είναι υπεύθυνο για την εκτέλεση σεναρίων της JavaScript. Στους φυλλομετρητές περιέχεται μια μηχανή για τον σκοπό αυτό και ονομάζεται «εικονική μηχανή JavaScript». Η σύγχρονη JavaScript είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, δεν παρέχει δηλαδή πρόσβαση στη μνήμη ή στον επεξεργαστή, αφού αρχικά δημιουργήθηκε για φυλλομετρητές που δεν απαιτούν τέτοιες λειτουργίες. Η JavaScript είναι διαθέσιμη για την πληθώρα πλαϊσίων (frameworks) που την διέπουν και επεκτείνουν τις δυνατότητές της. Συνεπώς, πάντα έχει να προσφέρει κάτι νέο στους προγραμματιστές και δικαίως έχει εγερθεί στην κορυφή των τεχνολογιών ιστού.

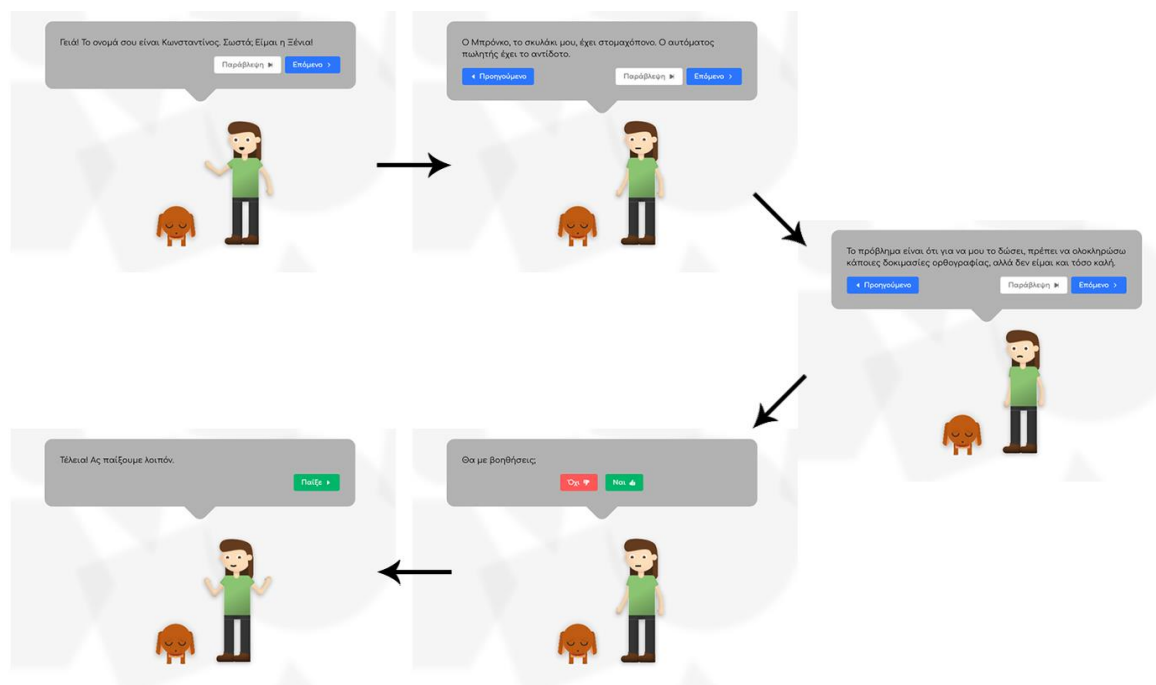
Σχεδιασμός Βρόχου Παιχνιδιού 4.3

Με το εγχείρημα αυτό να απευθύνεται σε παιδιά και έχοντας σαν στόχο του να ελκύει τους μαθητές στην επανειλημμένη χρήση του με απώτερο σκοπό την αυτοβελτιωσή τους, απαραίτητη ήταν η δημιουργία ενός ευχάριστου περιβάλλοντος στο οποίο θα ένιωθαν ευχάριστα και άνετα.

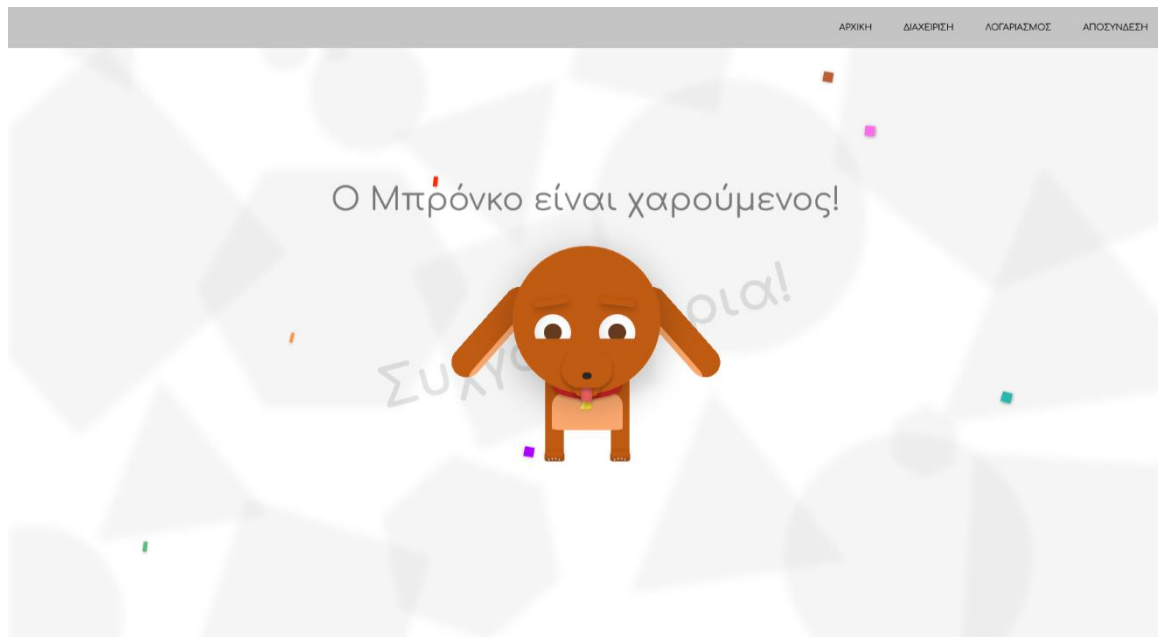
Για το ρόλο αυτό επιλέχθηκε ένα παιχνίδι, το οποίο θα παρουσιάζει στον χρήστη-παίκτη μία πρόκληση την οποία θα πρέπει να αντιμετωπίσει. Η πρόκληση αυτή δεν πρέπει να είναι υψηλής σημαντικότητας, για την αποφυγή δημιουργίας άγχους και μίας δυσάρεστης γενικότερα ατμόσφαιρας. Οι επιπτώσεις πιθανής αποτυχίας θα πρέπει να δημιουργούν στο χρήστη κίνητρο για να ξαναπροσπαθήσει. Στην προκειμένη περίπτωση, ο βασικός χαρακτήρας, η Ξένια, ζητά τη βοήθεια του παίκτη στην προσπάθειά της να κάνει το κατοικίδιό της να νιώσει καλύτερα. Ο παίκτης καλείται να ολοκληρώσει κάποιες ερωτήσεις ορθογραφίας και με το πέρας της πρόκλησης καταφέρνει τελικά να βοηθήσει την Ξένια και το κατοικίδιό της. Η αφήγηση του σεναρίου της πρόκλησης γίνεται από τη βασική χαρακτήρα μέσω διαλόγου. Η πρόκληση αυτή καθ' αυτή είναι μία σειρά από ερωτήσεις ορθογραφικού χαρακτήρα, στις οποίες ο χρήστης καλείται να «σύρει» αυτή που θεωρεί σωστή απάντηση στο πεδίο που ενδείκνυται. Σε περίπτωση λαθών, ο χρήστης ανακατευθύνεται σε σελίδα στην οποία παρουσιάζεται με κανόνες αντίστοιχους της ενότητας στην οποία έγινε το λάθος. Στη σελίδα αυτή, ο παίκτης μπορεί να μελετήσει τους κανόνες και να υποβληθεί εκ νέου σε αξιολόγηση μετά το πέρας της μελέτης. Οι προσπάθειες των χρηστών, τόσο της μελέτης όσο και της αξιολόγησης, χρονομετρούνται για την καλύτερη κατανόηση της αποτελεσματικότητας της εφαρμογής και την εξαγωγή συμπερασμάτων για την πρόοδο τους.



Εικόνα 8 Διάγραμμα που περιγράφει το βρόχο παιχνιδιού της εφαρμογής.



Εικόνα 9 Διάλογος παρουσίασης παιχνιδιού.



Εικόνα 12 Τελική οθόνη μετά την επιτυχία του παίκτη.

Σενάρια Χρήσης Εφαρμογής 4.4

Το πιο σύνηθες μοντέλο σεναρίων χρήσης μιας εφαρμογής είναι αυτό των τριών ειδών χρηστών. Σε αυτό το μοντέλο αναλύονται τα σενάρια του χρήστη-επισκέπτη, του εγγεγραμμένου χρήστη και του χρήστη-διαχειριστή. Ο χρήστης-επισκέπτης είναι ο χρήστης ο οποίος μόλις επισκέφθηκε τη σελίδα που αναλύεται και δεν έχει κάποια ιδιαίτερα προνόμια σε σχέση με οποιονδήποτε άλλο χρήστη. Ένα παράδειγμα χρήστη-επισκέπτη θα ήταν ο χρήστης που επισκέπτεται μια ιστοσελίδα για να διαβάσει κάποιο άρθρο. Από την άλλη, ο εγγεγραμμένος χρήστης έχει περισσότερα προνόμια που αλλάζουν με βάση την εφαρμογή, θυσιάζοντας κάποια πράγματα όπως τα προσωπικά του στοιχεία που χρησιμοποιούνται κατά τη διαδικασία της εγγραφής και της αυθεντικοποίησης. Στο προηγούμενο παράδειγμα, ένας εγγεγραμμένος χρήστης θα μπορούσε να έχει το δικαίωμα να προσθέσει το εν λόγω άρθρο σε μια λίστα με όλα τα αγαπημένα του άρθρα ή να αναρτήσει σχόλια που αφορούν το άρθρο. Τέλος, ο χρήστης διαχειριστής απολαμβάνει τα περισσότερα προνόμια από κάθε άλλο χρήστη της σελίδας. Συνήθως έχει την πλήρη εξουσία σε μία εφαρμογή με επιπλέον δικαιώματα όπως η διαγραφή σχολίων, η ανάρτηση, επεξεργασία και διαγραφή άρθρων και σε μερικές περιπτώσεις και η διαγραφή χρηστών και άλλων δεδομένων από τη βάση.

Στη εφαρμογή του θέματος μελετώνται μόνο τα σενάρια χρήσης του εγγεγραμμένου χρήστη και του χρήστη διαχειριστή, αφού για την εξαγωγή αποτελεσμάτων για κάθε

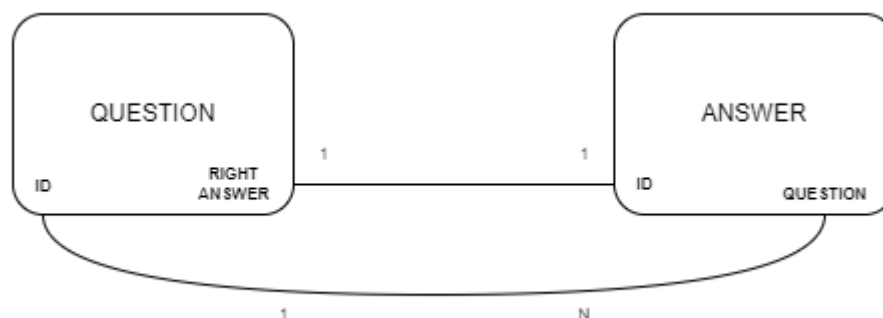
χρήστη απαραίτητη είναι η ταυτοποίησή του, γεγονός που καθιστά το σενάριο του χρήστη-επισκέπτη ανούσιο προς μελέτη. Έτσι, τα δικαιώματα που απολαμβάνει ο εγγεγραμμένος χρήστης είναι προφανή. Το παιχνίδι και η αξιολόγηση αποτελούν δύο από αυτά και είναι αδιάσπαστα από την εφαρμογή στοιχεία. Επιπλέον, στον εγγεγραμμένο χρήστη προσφέρεται η δυνατότητα να επεξεργαστεί τα προσωπικά του στοιχεία, καθώς και να διαγράψει το λογαριασμό του και μαζί του και όλα τα δεδομένα που τον αφορούν ανά πάσα στιγμή.

Περνώντας στο λογαριασμό χρηστών διαχειριστών, ένας διαχειριστής απολαμβάνει τα προνόμια που απολαμβάνουν και οι εγγεγραμμένοι χρήστες με την προσθήκη κάποιων επιπλέον. Ένα από τα πιο σημαντικά προνόμια των διαχειριστών της εφαρμογής είναι η εξαγωγή δεδομένων. Ο διαχειριστής θα μπορεί σε τρίτο χρόνο να εξάγει από τη βάση δεδομένων στοιχεία και στατιστικά για τους χρήστες και τις αξιολογήσεις τους, που θα του επιτρέπουν να διαμορφώσει μία εμπεριστατωμένη άποψη για την πρόοδο των χρηστών κατά την χρήση της εφαρμογής.

Βάση Δεδομένων 4.5

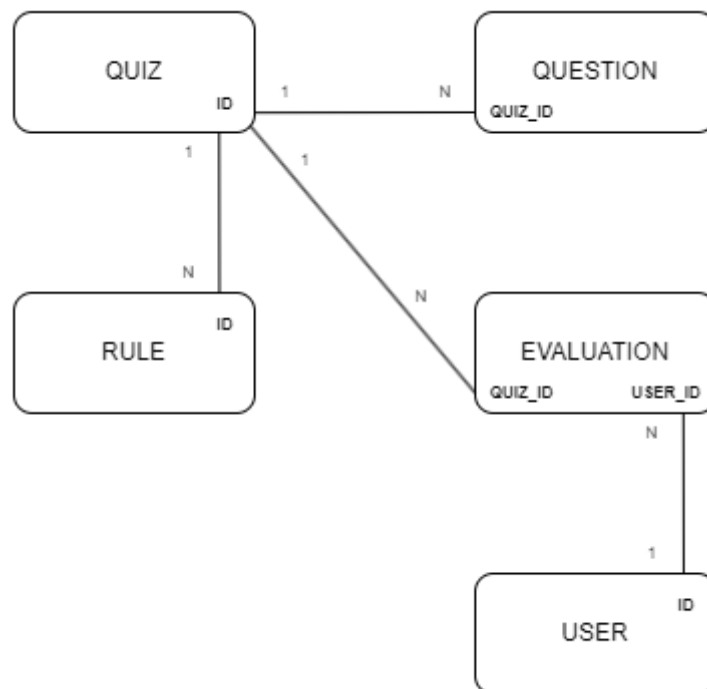
Για τη σωστή λειτουργία της εφαρμογής, απαραίτητη είναι η αποθήκευση μερικών δεδομένων χρηστών, όπως το ονοματεπώνυμό τους, το μορφωτικό τους επίπεδο, ο χρόνος τον οποίο έκαναν να απαντήσουν τις ερωτήσεις, ο χρόνος μελέτης πριν από κάθε ερωτηματολόγιο καθώς και οι βαθμολογίες τους ανά αξιολόγηση. Επιπλέον, θα πρέπει να είναι αποθηκευμένες οι ερωτήσεις κάθε αξιολόγησης μαζί με τις απαντήσεις τους, σωστές ή λανθασμένες. Τέλος, θα πρέπει να μοντελοποιηθούν οι σχέσεις μεταξύ αυτών των οντοτήτων.

Οι βασικές οντότητες είναι οι χρήστες (users), οι κανόνες (rules), οι ερωτήσεις (questions), οι απαντήσεις (answers), τα ερωτηματολόγια (quizzes) και οι βαθμοί (grades). Μεταξύ τους δημιουργούνται διάφορες σχέσεις εξάρτησης που πρέπει κι αυτές να αντιμετωπιστούν. Πρώτη και πιο εμφανής σχέση είναι μεταξύ ερωτήσεων και απαντήσεων. Μία ερώτηση μπορεί να έχει διάφορες απαντήσεις, μία εκ των οποίων είναι η σωστή, ενώ μία απάντηση ανήκει σε μία ερώτηση.



Εικόνα 13 Σχεσιασολογικό διάγραμμα μεταξύ των οντοτήτων «Ερώτηση» και «Απάντηση».

Έπειτα, κάθε ερωτηματολόγιο αποτελείται από μία ή περισσότερες ερωτήσεις και κάθε ερώτηση υπάγεται σε ένα μόνο ερωτηματολόγιο. Στη συνέχεια, κάθε κανόνας σχετίζεται μοναδικά με ένα ερωτηματολόγιο. Το αντίθετο δεν συμβαίνει, αφού ένα ερωτηματολόγιο μπορεί να περιέχει ερωτήσεις που αναφέρονται σε διαφορετικούς κανόνες. Τελευταία και από τις πιο σημαντικές συσχετίσεις μεταξύ αυτών των οντοτήτων είναι αυτή μεταξύ του χρήστη και των βαθμολογιών. Η σχέση αυτή, βέβαια, δεν είναι τόσο εμφανής όσο οι προηγούμενες αφού κρύβει μία όχι και τόσο εμφανή συσχέτιση. Ένας χρήστης μπορεί να έχει πολλές βαθμολογίες, ενώ μία βαθμολογία ανήκει σε ένα χρήστη. Υλοποιώντας καθ' αυτόν τον τρόπο τη συσχέτιση, δημιουργείται νέα σχέση μεταξύ βαθμολογίας και ερωτηματολογίου που υποδηλώνει την ταυτότητα του ερωτηματολογίου για το οποίο βαθμολογείται ο χρήστης. Δημιουργείται, λοιπόν, σχέση μεταξύ τριών οντοτήτων – χρήστης, βαθμολογία και ερωτηματολόγιο. Για την καλύτερη διαχείριση του όγκου πληροφορίας, εισάγεται μία νέα οντότητα στο σχήμα. Η νέα οντότητα ονομάζεται αξιολόγηση (evaluation) και σκοπός της είναι να αποθηκεύει το αναγνωριστικό του χρήστη, το αναγνωριστικό του ερωτηματολογίου και την πρώτη και τελευταία βαθμολογία του χρήστη για το συγκεκριμένο ερωτηματολόγιο.



Εικόνα 14 Σχεσιολογικό διάγραμμα μεταξύ των οντοτήτων «Αξιολόγηση», «Κανόνες», «Ερωτηματολόγιο», «Ερώτηση» και «Χρήστης».

Στην περίπτωση των δικαιωμάτων χρηστών, για το διαχωρισμό των απλών εγγεγραμμένων χρηστών από τους διαχειριστές, δυνατή είναι η χρήση ενός πεδίου που θα σημαίνει την ύπαρξη ή όχι δικαιωμάτων διαχειριστή για κάθε χρήστη. Παρ' ότι είναι ένας λειτουργικός τρόπος, ο όγκος δεδομένων που καλείται να αποθηκευτεί για κάθε χρήστη αυξάνεται σημαντικά. Μία πιο αποδοτική μέθοδος είναι η σύσταση μίας ακόμη οντότητας ονόματι διαχειριστές (administrators) η οποία θα αποθηκεύει τα αναγνωριστικά μόνο των χρηστών που έχουν δικαιώματα διαχειριστή, μειώνοντας κατά πολύ τον όγκο δεδομένων προς αποθήκευση.



Εικόνα 15 Διάγραμμα που περιγράφει τον τρόπο διαχωρισμού των διαχειριστών από τους απλούς χρήστες.

Άλλα Χρήσιμα Εργαλεία 4.6

Κύριος στόχος του εγχειρήματος αυτού είναι η βοήθεια μαθητών που φοιτούν σε διάφορα επίπεδα του ελληνικού εκπαιδευτικού συστήματος. Για τη διευκόλυνση της

κατανόησης των κανόνων και των ερωτήσεων της εφαρμογής και για την πρόληψη τυχόντων μαθησιακών δυσκολιών που μπορεί κάποιοι από τους μαθητές να αντιμετωπίζουν, απαραίτητο είναι να συμπεριληφθεί δυνατότητα υπαγόρευσης των κειμένων της εφαρμογής. Λόγω της έλλειψης εργοστασιακών εργαλείων που να λειτουργούν οικουμενικά ντετερμινιστικά, τη λύση δίνει μία βιβλιοθήκη αναπτυγμένη από τρίτους, που δεν είναι δηλαδή προσβάσιμη άμεσα από κάθε φυλλομετρητή ως βιβλιοθήκη της JavaScript. Η βιβλιοθήκη αυτή, υπό την ονομασία ResponsiveVoice, προσφέρει υπαγόρευση κειμένου με την ίδια φωνή σε κάθε φυλλομετρητή ανεξαρτήτου λειτουργικού συστήματος, σε αντίθεση με τα ήδη υπάρχοντα εργαλεία που είναι έως τώρα διαθέσιμα στη JavaScript και τα πρότυπα της HTML. Η δυνατότητα αυτή χρησιμοποιείται στους κανόνες και στις ερωτήσεις που καλείται ο μαθητής να απαντήσει, καθώς και στους διαλόγους του με την κύρια χαρακτήρα της εφαρμογής που εισάγει τον χρήστη – παίκτη στον βρόχο παιχνιδιού.

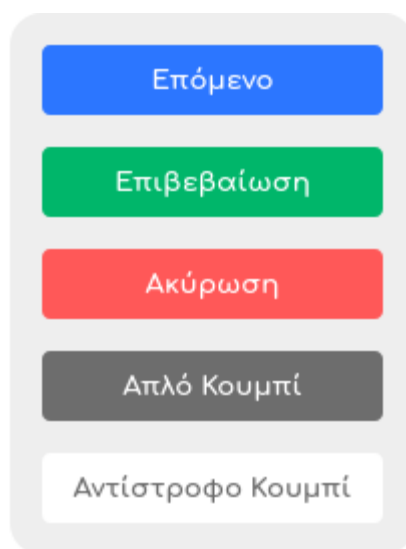
Επιπλέον, χρήση βιβλιοθηκών γίνεται για εικαστικούς σκοπούς και την παρουσίαση μηνυμάτων με πιο ευχάριστο τρόπο στον τελικό χρήστη. Πρώτη βιβλιοθήκη που υπηρετεί αυτό το σκοπό είναι η IonIcons, μια ανοιχτού κώδικα συλλογή από διάφορα εικονίδια που κοσμούν τη σελίδα σε διάφορα σημεία για τον καλύτερο διαχωρισμό μεταξύ των πεδίων κειμένου και των διαφόρων κουμπιών της εφαρμογής, κάτι που εξυπηρετεί και στη διευκόλυνση των χρηστών με προβλήματα σχετικά με τα χρώματα και το διαχωρισμό τους. Όσον αφορά τα μηνύματα προς το χρήστη, και καθαρά για εικαστικούς λόγους και για τη δημιουργία ενός πιο ευχάριστου περιβάλλοντος, γίνεται χρήση της βιβλιοθήκης SweetAlert2. Μια βιβλιοθήκη που προσφέρει ένα ολοκληρωμένο σύστημα ειδοποιήσεων με όμορφο σχεδιασμό και χρώματα «ευχάριστα στο μάτι», ομαλή απεικόνιση κινήσεων και πλήρη έλεγχο της μορφής των ειδοποιήσεων για την εξατομίκευσή τους ανάλογα με το συνολικό προφίλ της εκάστοτε εφαρμογής.

ΚΕΦΑΛΑΙΟ 5 - Υλοποίηση Εφαρμογής Ιστού

Σχεδιασμός 5.1

Πρωτεύον κοινό στο οποίο απευθύνεται η εφαρμογή είναι κυρίως άτομα μικρά σε ηλικία και άνθρωποι του εκπαιδευτικού χώρου. Για το λόγο αυτό, ο σχεδιασμός της σελίδας θα πρέπει να αποτελείται από μία λιτή παλέτα χρωμάτων, με τη δυνατότητα εξατομίκευσης της στην πορεία. Ο βασικός σχεδιασμός υλοποιήθηκε στο εργαλείο σχεδιασμού Figma, το οποίο διατίθεται δωρεάν προς χρήση από τους δημιουργούς του.

Όσον αφορά την χρωματική παλέτα, χρησιμοποιούνται γνώριμα στους χρήστες χρώματα για τις διάφορες λειτουργίες που παρέχονται από την εφαρμογή. Το κόκκινο, γνωστό ως χρώμα που επιδεικνύει επικίνδυνες ενέργειες ή απαγόρευση, χρησιμοποιείται στα κουμπιά που είναι υπεύθυνα για τη διαγραφή του λογαριασμού του χρήστη, τη διαγραφή των δεδομένων από τη βάση και άλλες, αντίστοιχου βαθμού επικινδυνότητας, ενέργειες. Το πράσινο, από την άλλη, δηλώνει ενέργειες με θετικό χαρακτήρα ή με αντίκτυπο που δεν επηρεάζει αρνητικά το χρήστη. Έτσι, χρησιμοποιείται σε κουμπιά όπως αυτό της αποθήκευσης δεδομένων από τη βάση στη σελίδα του διαχειριστή. Τα δύο αυτά χρώματα χρησιμοποιούνται επίσης και για την απάντηση, πράσινο για τη θετική και κόκκινο για την αρνητική, σε διαλόγους του χρήστη με την κύρια χαρακτήρα. Σαν ουδέτερου χαρακτήρα χρώμα χρησιμοποιείται το μπλε. Σε διάφορα κουμπιά που η ενέργεια δεν έχει θετική ή αρνητική «χροιά», όπως η επιβεβαίωση κάποιας απάντησης ή η πρόοδος σε επόμενο κανόνα κατά τη διαδικασία της μελέτης, ταιριάζει απόλυτα το μπλε ως χρώμα με ουδέτερο χαρακτήρα.



Εικόνα 16 Απεικόνιση του σχεδιασμού όλων των ειδών κουμπιών της εφαρμογής.

Στο βασικό κορμό της εφαρμογής χρησιμοποιείται κατά κόρον το γκρι σε διάφορες αποχρώσεις για να δηλώσει την ιεραρχία μεταξύ διαφόρων εξαρτημάτων της εφαρμογής. Ένας άλλος λόγος που χρησιμοποιείται το γκρι στη θέση οποιουδήποτε άλλου χρώματος είναι το γεγονός ότι δεν αποσπά την προσοχή του χρήστη από το κυρίως περιεχόμενο, βοηθώντας έτσι στην ενίσχυση της συγκέντρωσης, με απώτερο σκοπό τα βέλτιστα αποτελέσματα.

Ένας άλλος σημαντικός τομέας που χρήζει προσοχή είναι η περιήγηση στις διάφορες λειτουργίες και σελίδες της εφαρμογής. Ακολουθώντας την «πεπατημένη», μία ενότητα που περιέχει όλους τους υπερσυνδέσμους της εφαρμογής κοσμεί την κορυφή της σε μόνιμη βάση. Η χρήση της ενότητας αυτής, γνωστή ως μπάρα περιήγησης (navigation bar), καθιστά την περιήγηση στα υποτμήματα της εφαρμογής πιο εύκολη όντας γνώριμο εργαλείο στους χρήστες του ιστού. Οι υπερσύνδεσμοι ακολουθούν λιτό σχεδιασμό, με μεγάλα κενά μεταξύ τους και υπογράμμιση όταν ο δείκτης βρίσκεται από πάνω τους.



Εικόνα 17 Απεικόνιση της μπάρας περιήγησης.

Αν και συνήθως ο μέσος χρήστης δεν ξοδεύει πολλή από το χρόνο του στις ρυθμίσεις σε μία σελίδα, η διατήρηση του ευχάριστου περιβάλλοντος προωθεί το σχεδιασμό ενός εύχρηστου πίνακα διαχείρισης των προσωπικών στοιχείων και του λογαριασμού του χρήστη. Έτσι, ο πίνακας έχει σχεδιαστεί λιτά, όπως και το υπόλοιπο της εφαρμογής, και έχει χωριστεί σε τρεις τομείς επικινδυνότητας κατά αύξουσα ταξινόμηση. Ο πρώτος από τους τρεις τομείς περιέχει προσωπικά στοιχεία του χρήστη όπως το όνομα και το επώνυμο του. Ο δεύτερος τομέας επιτρέπει στο χρήστη την αλλαγή του κωδικού πρόσβασής του, με την προϋπόθεση ότι γνωρίζει τον τρέχοντα κωδικό του. Τελευταίος και πιο επικίνδυνος τομέας του πίνακα είναι αυτός που επιτρέπει στον χρήστη να διαγράψει το λογαριασμό του με το πάτημα ενός κουμπιού, αφού βέβαια απαντήσει θετικά στο διάλογο επιβεβαίωσης της πράξης που πρόκειται να πραγματοποιηθεί.

Ρυθμίσεις

Προσωπικά Στοιχεία

kSiasios

Κωνσταντίνος

Σιάσιος

Αποθήκευση

Αλλαγή Κωδικού Πρόσβασης

Κωδικός Πρόσβασης

Νέος Κωδικός Πρόσβασης

Επανάληψη Νέου Κωδικού Πρόσβασης

Αποθήκευση

Διαγραφή Λογαριασμού

Διαγραφή

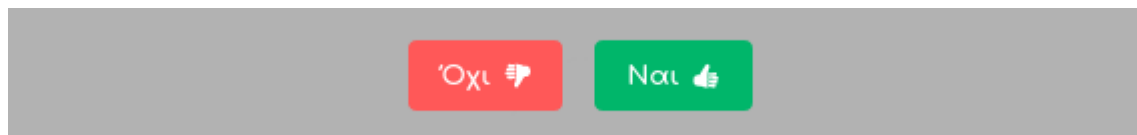
Εικόνα 18 Απεικόνιση του πίνακα ρυθμίσεων.

Περνώντας στα πιο σημαντικά στοιχεία της εφαρμογής, αυτά με τα οποία ο χρήστης θα έχει την περισσότερη επαφή στο χρόνο του στη σελίδα, συναντάται το σύστημα διαλόγου του χρήστη με την κύρια χαρακτήρα του βρόχου παιχνιδιού. Εμπνευσμένο από τα «σύννεφα» των κόμικς, έχει σχεδιαστεί ένας τομέας στον οποίο αναγράφονται οι διάλογοι της πρωταγωνίστριας. Οι διάλογοι εμφανίζονται σταδιακά, ένα γράμμα τη φορά, ώστε να προσδοθεί περισσότερο ενδιαφέρον στη διαδικασία και να κινήσει το ενδιαφέρον του χρήστη να προχωρήσει το διάλογο. Ακριβώς κάτω από τον τομέα των διαλόγων υπάρχει ένας ακόμη τομέας που ενθυλακώνει τα κουμπιά αλληλεπίδρασης του χρήστη με το σύστημα διαλόγου. Τα τρία βασικά κουμπιά είναι αυτά της προόδου του διαλόγου, της επιστροφής σε προηγούμενο διάλογο και της παράλειψης του διαλόγου. Το τελευταίο εκ των τριών έχει προστεθεί για λόγους ευκολίας, ώστε να μπορεί ο χρήστης που συνδέεται επανειλημμένα στην εφαρμογή να μεταβαίνει πιο γρήγορα στην αξιολόγηση. Δύο επιπλέον κουμπιά που εμφανίζονται στον τομέα αυτό είναι αυτά της θετικής και αρνητικής απάντησης του χρήστη στο διάλογό του με την πρωταγωνίστρια, επεκτείνοντας έτσι τις δυνατότητες του συστήματος για διαλόγους μεγαλύτερης πολυπλοκότητας. Τέλος, στον τομέα φωλιάζεται το κουμπί που μεταφέρει το χρήστη

στην αξιολόγηση και εμφανίζεται μόνο στο πέρας του διαλόγου, είτε αυτό είναι σειριακό ή με τη χρήση του κουμπιού παράλειψης.



Εικόνα 19 Απεικόνιση των διαφόρων κουμπιών που υπάρχουν στο σύστημα διαλόγου της εφαρμογής.



Εικόνα 20 Απεικόνιση των διαφόρων κουμπιών που υπάρχουν στο σύστημα διαλόγου της εφαρμογής.



Εικόνα 21 Απεικόνιση των διαφόρων κουμπιών που υπάρχουν στο σύστημα διαλόγου της εφαρμογής.

Η σελίδα της αξιολόγησης, μία από τις δύο σελίδες που απολαμβάνουν τον περισσότερο χρόνο από το χρήστη, έχει σχεδιαστεί στο ίδιο μοτίβο με το υπόλοιπο της εφαρμογής. Λιτός σχεδιασμός με τρεις τομείς, ο καθένας με το δικό του ρόλο. Ο πρώτος φιλοξενεί την ερώτηση αυτή καθ' αυτή και το πεδίο στο οποίο ο χρήστης θα μπορεί να σύρει την απάντησή του, όντας σύστημα ερωτοαπαντήσεων με συρόμενες απαντήσεις (drag and drop). Έπεται ο τομέας που στεγάζει τις συρόμενες απαντήσεις μία προς μία τοποθετημένες σε κάθετη διάταξη. Η κάθε μία από αυτές είναι χρωματισμένη με διαφορετικό χρώμα ανάλογα με το περιεχόμενό της ώστε να βοηθά τον χρήστη στο διαχωρισμό τους και ιδανικά στη βελτίωσή του στην ορθογραφία. Η τριάδα των τομέων ολοκληρώνεται με το κουμπί της προόδου σε επόμενη ερώτηση, το οποίο έχει και τις περισσότερες λειτουργίες στη σελίδα των αξιολογήσεων. Με το πάτημά του, αν ο χρήστης δεν έχει δώσει απάντηση, ενεργοποιείται το σύστημα ειδοποιήσεων με μήνυμα προς το χρήστη ότι δεν έχει απαντήσει στη δοθείσα ερώτηση. Σε διαφορετική περίπτωση, ελέγχεται η ορθότητα της απάντησης που δόθηκε από το χρήστη, γίνεται καταμέτρηση του χρόνου που έκανε για να απαντήσει και ανανεώνεται η σελίδα με νέα ερώτηση και απαντήσεις, εφόσον η τρέχουσα ερώτηση δεν είναι η τελευταία της λίστας.

Ασχολίαστη δεν πρέπει να μείνει η σελίδα των κανόνων, η οποία είναι μία από τις σελίδες στις οποίες ο χρήστης θα ξοδεύει τον περισσότερο από το χρόνο του. Το κυρίως μέρος της αποτελείται από το εξάρτημα που στεγάζει τον κανόνα και τα επιμέρους στοιχεία του. Πρώτο στοιχείο που κοσμεί τον εν λόγω εξάρτημα είναι ο τίτλος του κανόνα. Ακολουθεί η περιγραφή του κανόνα και ενίοτε τυχόντα παραδείγματα που μπορεί να έχουν παρασχεθεί για το εκάστοτε κανόνα. Μετά το πέρας των στοιχείων του κανόνα βρίσκεται ένας τομέας που περιέχει κουμπιά που, όπως και στους τομείς προηγουμένως αναλυθέντων σελίδων, παραλείπουν τους κανόνες, μεταφέρουν το χρήστη σε επόμενο κανόνα ή τερματίζουν τη διαδικασία της μελέτης και μεταφέρουν τον χρήστη σε νέα αξιολόγηση.

Προφίλ Χρηστών 5.2

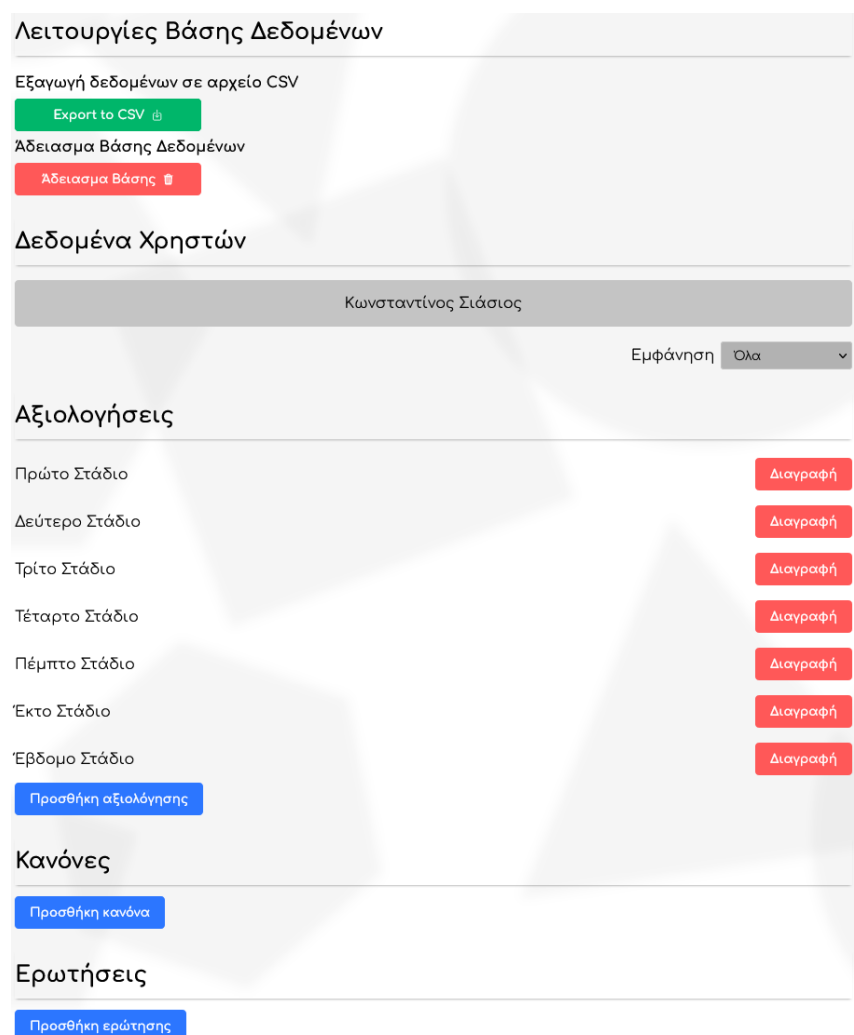
Όπως αναφέρθηκε και προηγουμένως, στα πλαίσια της εφαρμογής υπάρχουν δύο είδη χρηστών. Οι απλοί εγγεγραμμένοι χρήστες και οι χρήστες με δικαιώματα διαχειριστή. Οι διαφορές των δύο αναλύθηκαν σε επίπεδο εφαρμογής, αλλά διαχωρισμός πρέπει να γίνει και στο σχεδιαστικό κομμάτι. Για το διαχωρισμό των χρηστών έχει χρησιμοποιηθεί ένα σύστημα εγγραφών και σύνδεσης τους με τη βάση δεδομένων.

The image displays two screenshots of the 'Ορθοπαίγνια' application interface. The left screenshot shows the registration ('Εγγραφή') screen, which is divided into two columns: 'Στοιχεία χρήστη' (User details) and 'Στοιχεία μαθητή' (Student details). The 'User details' column includes fields for 'Όνομα Χρήστη' (Username), 'Ηλ. Ταχυδρομείο' (Email), 'Κωδικός Πρόσβασης' (Access code), and 'Επανάληψη Κωδικού Πρόσβασης' (Repeat access code). The 'Student details' column includes fields for 'Όνομα Μαθητή' (Student name), 'Επώνυμο Μαθητή' (Student surname), and a dropdown menu for 'Επιλέξτε τάξη' (Select grade). A 'Εγγραφή' (Register) button is at the bottom. The right screenshot shows the login ('Σύνδεση') screen, which includes fields for 'Όνομα Χρήστη' (Username) and 'Κωδικός Πρόσβασης' (Access code), and a 'Σύνδεση' (Login) button.

Εικόνα 22 Σύστημα εγγραφής και σύνδεσης χρηστών στην εφαρμογή.

Ο χρήστης-διαχειριστής, ως χρήστης με περισσότερα δικαιώματα από τον απλό χρήστη, θα πρέπει να έχει ένα χώρο μέσα στη σελίδα στον οποίο θα έχει εκείνος και μόνον εκείνος πρόσβαση. Ο πιο κοινός τρόπος υλοποίησης των επιπρόσθετων δικαιωμάτων των διαχειριστών είναι ο πίνακας διαχείρισης, ο οποίος στην περίπτωση της εφαρμογής γίνεται προσβάσιμος μόνο από τους διαχειριστές από την μπάρα περιήγησης.

Στην καινούργια αυτή σελίδα ο διαχειριστής έχει πρόσβαση στα στοιχεία των χρηστών της σελίδας και τα στατιστικά τους με το πάτημα ενός κουμπιού. Επιπλέον, έχει τη δυνατότητα να προσθέτει και να διαγράφει προς και από τη βάση δεδομένων αξιολογήσεις, κανόνες και ερωτήσεις. Επιπροσθέτως, του δίνεται η δυνατότητα να εξαγάγει τα στατιστικά στοιχεία των χρηστών σε αρχείο Microsoft Excel για τη διευκόλυνσή του στην ανάλυση των στοιχείων. Στις επικίνδυνες των δυνατοτήτων ενός διαχειριστή προστίθεται και αυτή της ολικής και μη αναστρέψιμης διαγραφής όλων των στοιχείων από τη βάση δεδομένων. Οι παραπάνω ενότητες χωρίζονται οπτικά σε πέντε κατηγορίες και ακολουθούν τη γενικότερη νοοτροπία της εφαρμογής σχετικά με το σχεδιασμό.



Εικόνα 23 Πίνακας διαχειριστή.

Η πρώτη των κατηγοριών, τα δεδομένα χρηστών, περιέχει μία λίστα από όλους τους χρήστες της εφαρμογής ταξινομημένους από τον παλαιότερο αυτών προς τον νεότερο. Το

όνομα κάθε χρήστη είναι ένα κουμπί, το οποίο, μετά την πίεση του, μεταφέρει τον διαχειριστή σε μία νέα σελίδα που περιέχει τα στοιχεία του συγκεκριμένου χρήστη.

The screenshot displays a web interface for user management. At the top, a header bar labeled 'Δεδομένα Χρηστών' (User Data) contains the name 'Κωνσταντίνος Σιάσιος' (Konstantinos Siasios). Below this, a large arrow points to a detailed view of the user. The main content area is titled 'Κωνσταντίνος Σιάσιος - ksiasios@uth.gr' and 'Δ' Δημοτικού' (4th Primary). Below the title, there is a section for 'Αξιολόγηση #33' (Evaluation #33). This section contains two columns of data, each representing a different evaluation point.

Βαθμός 0	Βαθμός 1
Ημερομηνία: <u>Wed Oct 5 15:23:46 CEST 2022</u>	Ημερομηνία: <u>Mon Nov 7 10:58:25 UTC 2022</u>
Ποσοστό Επιτυχίας: <u>80%</u>	Ποσοστό Επιτυχίας: <u>100%</u>
Χρόνος Ολοκλήρωσης: <u>26s</u>	Χρόνος Ολοκλήρωσης: <u>35.1s</u>
Χρόνος Μελέτης: <u>5.9s</u>	Χρόνος Μελέτης: <u>40.8s</u>

Εικόνα 24 Δεδομένα χρήστη.

Οι επόμενες τρεις κατηγορίες περιέχουν τις αξιολογήσεις, τους κανόνες και τις ερωτήσεις που βρίσκονται αποθηκευμένες στη βάση δεδομένων, και τιτλοφορούνται ομώνυμα με το περιεχόμενό τους. Δίπλα από κάθε στοιχείο των λιστών υπάρχει ένα κουμπί, η ενέργεια του οποίου διαγράφει το εκάστοτε στοιχείο. Το κουμπί χρωματίζεται κόκκινο για την ένδειξη κινδύνου. Στο τέλος κάθε κατηγορίας από τις τρεις προαναφερθείσες, τοποθετημένο βρίσκεται ένα κουμπί μπλε χρώματος που επιτρέπει στο διαχειριστή να προσθέσει στη βάση κάποια νέα αξιολόγηση, κανόνα ή ερώτηση αντίστοιχα, αφού τον μεταφέρει στην αντίστοιχη σελίδα, προσβάσιμη από τους διαχειριστές και μόνο. Δοχείο των λειτουργιών της βάσης δεδομένων είναι η τελευταία κατηγορία, καθώς περιέχει δύο κουμπιά που σχετίζονται με τη βάση δεδομένων. Το δεύτερο, και άκρως επικίνδυνο κουμπί, είναι υπεύθυνο για τη διαγραφή των δεδομένων από τη βάση δεδομένων. Για το λόγο αυτό έχει χρωματιστεί κατάλληλα, ενώ έχει προστεθεί και ενδεικτικό εικονίδιο που περιγράφει την ενέργεια που πρόκειται να πραγματοποιηθεί. Το πιο χρήσιμο στα πλαίσια της εφαρμογής κουμπί εκ των δύο, το πρώτο, δίνει στο διαχειριστή τη δυνατότητα να εξάγει σε αρχείο CSV τα στοιχεία των χρηστών προς ανάλυση. Στα παραπάνω περιέχονται το ονοματεπώνυμο, η διεύθυνση ηλεκτρονικού ταχυδρομείου, το όνομα χρήστη, το μορφωτικό επίπεδο του χρήστη, στοιχεία για την πρώτη του αξιολόγηση,

στοιχεία για την τελευταία του αξιολόγηση και στοιχεία για τον χρόνο τον οποίο μελέτησε ο χρήστης μεταξύ των αξιολογήσεων.

Χρήσιμα Εργαλεία 5.3

Σε μία εφαρμογή τόσο συγκεκριμένου περιεχομένου, απαραίτητη είναι η ανάπτυξη διαφόρων εργαλείων στα μέτρα και τις απαιτήσεις της για τη διευκόλυνση τόσο των απλών χρηστών, όσο και των χρηστών-διαχειριστών. Τα δύο εργαλεία-πρωταγωνιστές της παρούσας εφαρμογής είναι η εξαγωγή των δεδομένων σε αρχείο CSV και ο μορφοποιητής κειμένου που χρησιμοποιείται στις λειτουργίες διαχειριστή.

Για την εξαγωγή των δεδομένων, το εργαλείο αρχικά λαμβάνει από τη βάση δεδομένων τα στοιχεία όλων των χρηστών ταξινομημένα από τον παλιότερο προς το νεότερο των χρηστών. Στον όγκο δεδομένων που λαμβάνεται, υπάρχουν δεδομένα που είτε είναι άχρηστα για τον χρήστη που αναλαμβάνει την ανάλυση, είτε δεν θα πρέπει να είναι προσβάσιμα από τρίτους, όπως για παράδειγμα ο κωδικός πρόσβασης των χρηστών στην εφαρμογή. Πρέπει λοιπόν να φιλτραριστούν και στη συνέχεια, εφόσον είναι κρυπτογραφημένα, να αποκρυπτογραφηθούν και να πλασθούν σε μία μορφή κατάλληλη για εξαγωγή. Το αρχείο ονοματοδοτείται κατάλληλα για να είναι ευκολότερη η ευρετηρίαση, κάνοντας χρήση του ονόματος της εφαρμογής ακολουθούμενο από την τρέχουσα ημερομηνία. Το λειτουργικό κομμάτι αναλαμβάνουν οι υποδομές της γλώσσας που χρησιμοποιείται στο πίσω μέρος (backend) της εφαρμογής, δηλαδή της PHP.

Εξαγωγή δεδομένων σε αρχείο CSV

Export to CSV 



	A	B	C	D	E	F	G	H	I	J
1	ID	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	E-MAIL	ΟΝΟΜΑ ΧΡΗΣΤΗ	ΕΚΠΑΙΔΕΥΣΗ				
2		13	Κωνσταντίνος	ksiasios@uth.gr	ksiasios	Δι' Δημοτικού				
3										
4										
5	ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ	ΧΡΗΣΤΗΣ	ΠΡΩΤΗ ΠΡΟΣΠΑΘΕΙΑ				ΤΕΛΕΥΤΑΙΑ ΠΡΟΣΠΑΘΕΙΑ			
6			ΗΜΕΡΟΜΗΝΙΑ	ΠΟΣΟΣΤΟ ΕΠΙΤΥΧΙΑΣ	ΧΡΟΝΟΣ ΟΛΟΚΛΗΡ.	ΧΡΟΝΟΣ ΜΕΛΕΤΗΣ	ΗΜΕΡΟΜΗΝΙΑ	ΠΟΣΟΣΤΟ ΕΠΙΤΥΧΙΑΣ	ΧΡΟΝΟΣ ΟΛΟΚΛΗΡΩΣΗΣ	ΧΡΟΝΟΣ ΜΕΛΕΤΗΣ
7										
8	Πρώτο Στάδιο	ksiasios	Wed Oct 5 15:03:57 CEST 2022	100%	43.8	0	Mon Nov 7 10:54:06 UTC 2022	100%	36.1	0
9										
10	Δεύτερο Στάδιο	ksiasios	Fri Oct 7 10:30:53 CEST 2022	100%	18.5	7.7	Mon Nov 7 10:55:18 UTC 2022	100%	56.6	0
11										
12	Τρίτο Στάδιο	ksiasios	Mon Oct 24 16:03:33 UTC 2022	0%	0	0	Mon Nov 7 10:55:19 UTC 2022	0%	0	0
13										
14	Τέταρτο Στάδιο	ksiasios	Mon Oct 24 16:04:03 UTC 2022	100%	20.6	0	Mon Nov 7 10:55:37 UTC 2022	100%	14.2	0
15										
16	Πέμπτο Στάδιο	ksiasios	Mon Oct 24 16:04:05 UTC 2022	0%	0	0	Mon Nov 7 10:55:37 UTC 2022	0%	0	0
17										
18	Έκτο Στάδιο	ksiasios	Mon Oct 24 16:04:27 UTC 2022	100%	15.7	0	Mon Nov 7 10:56:01 UTC 2022	100%	13.6	0
19										
20	Εβδομο Στάδιο	ksiasios	Wed Oct 5 15:23:46 CEST 2022	80%	26	5.9	Mon Nov 7 10:58:25 UTC 2022	100%	35.1	40.8
21										
22										
23										
24										
25										
26										
27										

Εικόνα 25 Λειτουργία εξαγωγής δεδομένων χρηστών σε αρχείο CSV.

Ακολουθώντας στο μοτίβο της διευκόλυνσης του χρήστη και της δημιουργίας ενός ευχάριστου περιβάλλοντος για τον τελικό χρήστη, απαραίτητη κρίνεται η προσθήκη χρωμάτων σε διάφορα σημεία που σχετίζονται με τη μελέτη αλλά και την αξιολόγηση του χρήστη. Προκύπτει, λοιπόν, η ανάγκη για ένα σύστημα ικανό να δημιουργεί το επιθυμητό αποτέλεσμα χωρίς να είναι πολύπλοκο στη χρήση από τον διαχειριστή. Από την ανάγκη αυτή δημιουργήθηκε ο μορφοποιητής κειμένου της εφαρμογής, ένα σύστημα φιλικό προς τον συντάκτη του περιεχομένου της εφαρμογής. Στα σημεία της εφαρμογής που είναι διαθέσιμος ο μορφοποιητής, με τη χρήση απλών συμβόλων, ο συντάκτης μπορεί να μετατρέψει το κείμενό του. Το κείμενο αλλάζει μορφή στη έξοδό του ανάλογα με τους χαρακτήρες που το περικλείουν. Επιπλέον, διάφορα σύμβολα μπορούν να γίνουν διαθέσιμα με την προσθήκη λίγων γραμμών στον επεξεργαστή κειμένου. Το τελικό προϊόν διαθέτει δέκα διαφορετικούς χρωματισμούς για το κείμενο, βέλος με δεξιά κατεύθυνση και πεδίο στο οποίο ο χρήστης θα μπορεί να σύρει την απάντησή του κατά τη διαδικασία της αξιολόγησης. Έτσι, ο διαχειριστής έχει πλήρη έλεγχο πάνω στο περιεχόμενο των κανόνων, των ερωτήσεων, αλλά και των απαντήσεων που εμπεριέχονται στην εφαρμογή.

---	→	
!!κείμενο/!!	→	κείμενο
κείμενο/	→	κείμενο
\$κείμενο/\$	→	κείμενο
#κείμενο/#	→	κείμενο
%κείμενο/%	→	κείμενο
^κείμενο/^	→	κείμενο
&κείμενο/&	→	κείμενο
@κείμενο/@	→	κείμενο
?κείμενο/?	→	κείμενο
[κείμενο/]	→	κείμενο
=>	→	→

Εικόνα 26 Απεικόνιση αποτελεσμάτων του μορφοποιητή κειμένου.

ΚΕΦΑΛΑΙΟ 6 - Υλοποίηση Βάσης Δεδομένων

Κρυπτογράφηση Δεδομένων 6.1

Όσο το πλήθος των χρηστών μιας εφαρμογής διογκώνεται, ο όγκος δεδομένων που καλείται να αποθηκευτεί μεγαλώνει. Μία βάση δεδομένων με πληροφορίες που αφορούν μεγάλο εύρος χρηστών γίνεται συχνά στόχος για επιτήδειους χρήστες του διαδικτύου. Σημαντικό είναι λοιπόν, ακόμη και να επιτευχθεί η υποκλοπή δεδομένων από την εν λόγω βάση δεδομένων, τα δεδομένα να μην έχουν κάποια αξία για τον παραλήπτη των κλοπιμαίων. Το κενό αυτό έρχεται να καλύψει η κρυπτογραφία.

Σε ένα ουτοπικό σύστημα διαχείρισης βάσεων δεδομένων, όλα τα αποθηκευμένα δεδομένα είναι κρυπτογραφημένα. Ένα τέτοιο εγχείρημα όμως έχει μεγάλη πολυπλοκότητα χώρου, αφού ένα κρυπτογραφημένο μήνυμα ξεπερνά συνήθως κατά πολύ το αρχικό μήνυμα σε όγκο χαρακτήρων. Ένα ακόμη μειονέκτημα αυτής της στρατηγικής είναι ο αυξημένος χρόνος επεξεργασίας των δεδομένων κατά την εγγραφή τους στη βάση δεδομένων, αλλά και κατά την ανάγνωσή τους από αυτή σε δεύτερο χρόνο. Η κρυπτογράφηση και αποκρυπτογράφηση δεδομένων είναι διαδικασίες που απαιτούν αρκετό χρόνο, ανάλογα με τον αλγόριθμο που χρησιμοποιείται κάθε φορά. Γίνεται εύκολα κατανοητό επομένως, ότι θα πρέπει να γίνει διαχωρισμός μεταξύ των σημαντικών δεδομένων, των οποίων δηλαδή η κρυπτογράφηση είναι απαραίτητη πάση θυσεία, και των λιγότερο σημαντικών δεδομένων που σε περίπτωση υποκλοπής δεν θέτουν την ακεραιότητα των χρηστών σε κίνδυνο. Στην παρούσα εφαρμογή, όντας εγχείρημα σχετικό κυρίως με στοιχεία και δεδομένα ανηλίκων, είναι σαφές ότι το διακύβευμα είναι σημαντικό.

Τα πιο ευαίσθητα δεδομένα είναι αυτά που συνδέονται άμεσα με τους χρήστες, όσα δηλαδή αποθηκεύονται στον πίνακα «χρήστες». Σε αυτά περιλαμβάνονται το μοναδικό αναγνωριστικό κάθε χρήστη, το ονοματεπώνυμο, η διεύθυνση ηλεκτρονικού ταχυδρομείου, το όνομα χρήστη, το συνθηματικό χρήστη και το μορφωτικό επίπεδο του χρήστη. Το αναγνωριστικό δεν μπορεί να κρυπτογραφηθεί αφού δημιουργείται αυτόματα κατά την εισαγωγή ενός νέου χρήστη στη βάση δεδομένων. Το γεγονός αυτό όμως δεν δημιουργεί πρόβλημα όσον αφορά την ασφάλεια των χρηστών, αφού το αναγνωριστικό δεν εκφράζει κάποια πολύτιμη πληροφορία σχετικά με τον χρήστη, εκτός από τις σχέσεις του με τις υπόλοιπες οντότητες της βάσης δεδομένων, οι οποίες καθίστανται άχρηστες εφόσον παραμένουν απρόσωπες. Για τους ίδιους λόγους, μπορεί να αποφευχθεί και η

κρυπτογράφηση του μορφωτικού επιπέδου, μειώνοντας έτσι περισσότερο το χρόνο που αφιερώνεται στην διαδικασία αυτή. Όλα τα υπόλοιπα στοιχεία του πίνακα θα πρέπει να κρυπτογραφηθούν, αφού ταυτοποιούν άμεσα ή έμμεσα το χρήστη με σχετική ευκολία. Εφόσον ο χρήστης παραμένει ανώνυμος, οι υπόλοιπες πληροφορίες που σχετίζονται με αυτόν είναι άχρηστες και δεν χρειάζεται να κρυπτογραφηθούν.

Για την κρυπτογράφηση των δεδομένων αυτών δεν αρκεί μία απλή διαδικασία ανάμειξης των χαρακτήρων και ολίσθησης του αλφαβήτου. Πρέπει να χρησιμοποιηθεί κάποιος αλγόριθμος που να είναι πρακτικά απόρθητος. Η λύση βρίσκεται στην επιλεγμένη γλώσσα προγραμματισμού, την PHP, για την οποία και έχουν αναπτυχθεί βιβλιοθήκες και μέθοδοι που αναλαμβάνουν την πραγμάτωση της διαδικασίας. Χρησιμοποιώντας τις μεθόδους αναπτυγμένες από τον οργανισμό OpenSSL και δίνοντας σαν παράμετρο τον επιλεγμένο αλγόριθμο, επιτυγχάνεται η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης.

Περνώντας στην επιλογή αλγορίθμου, γρήγορα γίνεται κατανοητό ότι ο αλγόριθμος θα πρέπει να είναι αναγνωρισμένος και αποδεκτός από το ευρύ κοινό και τους ειδήμονες του κλάδου. Ο πιο γνωστός και ευρύτερα διαδεδομένος αλγόριθμος κρυπτογραφίας είναι ο AES (Advanced Encryption Standard), γνωστός και ως Rijndael, ο οποίος καθιερώθηκε από το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας των ΗΠΑ (NIST) το 2001. Ο AES αναπτύχθηκε για τους σκοπούς των ομοσπονδιακών υπηρεσιών των ΗΠΑ. Στα μέσα της δεκαετίας του 1970, οι ομοσπονδιακές υπηρεσίες χρησιμοποιούσαν για την κρυπτογράφηση των δεδομένων τους ένα άλλο πρότυπο, τον αλγόριθμο DES (Data Encryption Standard), το οποίο μέχρι τότε θεωρούνταν ικανότατο και αδιάσπαστο. Όσο το υλισμικό συνέχισε να αναπτύσσεται, έγινε όλο και πιο εύκολο για τους υπολογιστές της εποχής να διασπάσουν τις ασφάλειες του DES. Έτσι, στη δύση χιλιετίας άρχισε να εκφράζεται η ανάγκη για ένα νέο πρότυπο ασφαλείας των δεδομένων και μέσω ενός κρατικού διαγωνισμού που εξέθετε τους αλγορίθμους των διαγωνιζομένων στο κοινό, αναδείχθηκε ο αλγόριθμος Rijndael. Παρόμοιος στη φύση του με τον DES, ο αλγόριθμος των Vincent Rijmen και Joan Daemen, δεν άργησε να υιοθετηθεί από την Υπηρεσία Εθνικής Ασφάλειας των ΗΠΑ (NSA). Το γεγονός ότι ο αλγόριθμος κρίθηκε ικανός να προστατεύσει δεδομένα εθνικής ασφαλείας, του έδωσε την ώθηση που χρειαζόταν για να εδραιωθεί. Οι χρήσεις του είναι διάφορες και εκτείνονται από υλοποιήσεις σε διάφορες γλώσσες προγραμματισμού μέχρι διαχειριστές συνθηματικών και εικονικά ιδιωτικά δίκτυα (VPN).

Ο AES είναι ένας αλγόριθμος κρυπτογράφησης συμμετρικού κλειδιού, κάνει δηλαδή χρήση ενός μοναδικού κλειδιού για την κρυπτογράφηση και αποκρυπτογράφηση του όγκου πληροφορίας. Στην περίπτωση αυτή, ο αποστολέας και ο παραλήπτης πρέπει να γνωρίζουν το κλειδί. Σημαντικό είναι να μεταφερθεί το κλειδί με ασφαλή τρόπο από το ένα μέλος της επικοινωνίας στο άλλο. Αν και πιο γρήγοροι, οι αλγόριθμοι συμμετρικού κλειδιού δεν έχουν τρόπο να ολοκληρώσουν την προαναφερθείσα διαδικασία σε εφαρμογές διαδικτύου. Για το λόγο αυτό χρησιμοποιούνται αλγόριθμοι ασύμμετρου κλειδιού σε εφαρμογές τέτοιου είδους, που ο αποστολέας δεν έχει ασφαλή τρόπο να μεταφέρει το κλειδί στον παραλήπτη. Στην παρούσα εφαρμογή, αυτό δεν αποτελεί πρόβλημα, καθώς η κρυπτογράφηση και αποκρυπτογράφηση γίνονται στο ίδιο σημείο, στο διακομιστή της εφαρμογής, και επομένως δεν χρειάζεται η μεταφορά του κλειδιού.

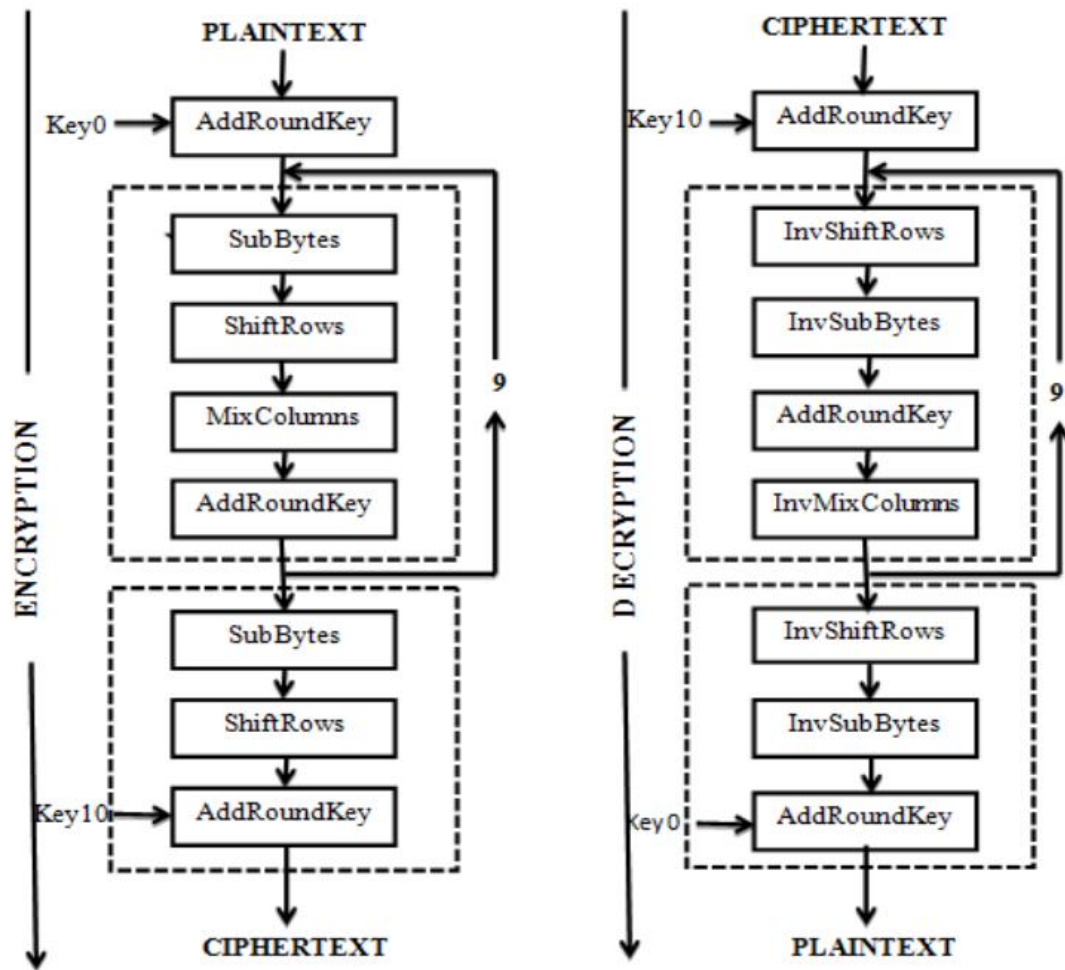
Ως αλγόριθμος δέσμης (block cipher), ο AES χωρίζει την πληροφορία προς κρυπτογράφηση σε τμήματα γνωστά ως δέσμες. Υπάρχουν τρεις εκδοχές του αλγορίθμου και διαφοροποιούνται από το μέγεθος της δέσμης που χρησιμοποιούν, 128, 192 και 256 bits αντίστοιχα, από τον λιγότερο ασφαλή προς τον ασφαλέστερο εκ των τριών. Στα πλαίσια της παρούσας εφαρμογής γίνεται χρήση της εκδοχής των 256 bits. Η βασική αρχή του αλγορίθμου είναι η αντικατάσταση κάθε δέσμης πληροφορίας από μία διαφορετική, ανάλογα με το κλειδί που έχει παρασχεθεί. Η διαφορά του AES με τους υπόλοιπους αλγορίθμους, και αυτή που τον καθιστά πιο ασφαλή, είναι το γεγονός ότι χρησιμοποιεί μία τεχνική επέκτασης του αρχικού κλειδιού, δημιουργώντας έτσι σε κάθε επανάληψη ένα νέο κλειδί βασισμένο στο κλειδί της προηγούμενης επανάληψης.

Στο πρώτο στάδιο της κρυπτογράφησης, τα bits του αρχικού κλειδιού συνδυάζονται με τα bits του μηνύματος ένα προς ένα με τη λογική πράξη XOR, όπου το αποτέλεσμα της είναι 1 αν ένα και μόνο εκ των δύο bit είναι 1, και οργανώνονται σε δυσδιάστατο πίνακα μεγέθους τέσσερα επί τέσσερα. Έπειτα, κάθε byte του αποτελέσματος του προηγούμενου βήματος αντικαθίσταται από κάποιο άλλο byte, προκαθορισμένο από το «κουτί αντικατάστασης» (S-box, substitution-box) του αλγορίθμου Rijndael. Στη συνέχεια, εφαρμόζεται αριστερή ολίσθηση σε κάθε γραμμή του πίνακα κατά μία θέση περισσότερη από την προηγούμενη της. Ακολουθεί ανάμειξη των στηλών του δυσδιάστατου πίνακα, αφού κάθε στήλη του πίνακα πολλαπλασιάζεται με ένα προκαθορισμένο πίνακα. Στο τελευταίο στάδιο κάθε γύρου, το νέο κλειδί συνδυάζεται με τη δέσμη κάνοντας και πάλι χρήση της λογικής πράξης XOR. Η διαδικασία επαναλαμβάνεται για το πλήθος των γύρων που καθορίζει κάθε εκδοχή του αλγορίθμου,

10, 12 και 14 γύρους για τις εκδοχές των 128, 192 και 256 bits αντίστοιχα. Για την αποκρυπτογράφηση του μηνύματος, η διαδικασία επαναλαμβάνεται αντίστροφα, παραδίδοντας έτσι το αρχικό μήνυμα στην έξοδο της.

Ο αλγόριθμος μπορεί να διαχωριστεί σε τέσσερα διακριτά στάδια, το καθένα από τα οποία υπηρετεί κάποιο σημαντικό σκοπό. Το αρχικό στάδιο αλλά και το τελευταίο κάθε γύρου, ονομάζεται AddRoundKey και σκοπός του είναι να παρέχει στον αλγόριθμο ένα νέο κλειδί σε κάθε γύρο. Έτσι το αποτέλεσμα κάθε γύρου γίνεται όλο και πιο πολύπλοκο με την πάροδο του αλγορίθμου. Στο επόμενο στάδιο, το SubBytes, το περιεχόμενο της δέσμης τροποποιείται μη γραμμικά με αποτέλεσμα να μην υπάρχει εμφανής σχέση με την αρχική πληροφορία. Τέλος, στα δύο στάδια της ανάμειξης των στοιχείων του πίνακα, τα ShiftRows και MixColumns, γίνεται οριζόντια και κατακόρυφη αντίστοιχα ανακατανομή των στοιχείων του πίνακα, περιπλέκοντας έτσι την διαδικασία αλλά και το αποτέλεσμα του αλγορίθμου.

Σε εφαρμογές ιστού με διακομιστές όπως η παρούσα, υπάρχουν δύο ειδών χώροι δεδομένων. Αυτοί που είναι προσβάσιμοι, άμεσα ή μέσω διαφόρων λειτουργιών της σελίδας, από τους χρήστες και αυτοί που είναι προσβάσιμοι από το διακομιστή και μόνο. Για την προστασία του μυστικού κλειδιού απαραίτητη είναι η αποθήκευσή του στο χώρο δεδομένων που έχει πρόσβαση μόνο ο διακομιστής. Σε ξεχωριστό αρχείο, στη δομή αρχείων του διακομιστή όπου οι τελικοί χρήστες δεν έχουν πρόσβαση, θα αποθηκευτεί το κλειδί, αποφεύγοντας έτσι τον κίνδυνο υπεξαίρεσής και χρήσης του για κακόβουλους σκοπούς.



Εικόνα 27 Οπτική περιγραφή του αλγορίθμου AES στα στάδια της κρυπτογράφησης και αποκρυπτογράφησης.

Παράρτημα

Κώδικας

Φάκελος components

Footer.php

```
<div class="footer">
```

```
</div>
```

```
</body>
```

```
</html>
```

Navbar.php

```
<?php
```

```
if (!file("variables.env")) {
```

```
    $baseUrl = file("../variables.env")[0];
```

```
} else {
```

```
    $baseUrl = file("variables.env")[0];
```

```
}
```

```
?>
```

```
<script src="<?php echo $baseUrl ?>/js/global.js"></script>
```

```
<div class="backdrop"></div>
```

```
<nav>
```

```
<ul class="closed">
```

```
<li class="nav-link"><a href="<?php echo $baseUrl ?>/">ΑΡΧΙΚΗ</a></li>
```

```
<?php
```

```
if (isset($_SESSION["isAdmin"])) {
```

```
        echo '<li class="nav-link"><a href="' . $baseUrl .  
'/routes/admin_panel.php">ΔΙΑΧΕΙΡΙΣΗ</a></li>';
```

```
    } ?>
```

```
    <li class="nav-link"><a href="<?php echo $baseUrl  
?>/routes/account.php">ΛΟΓΑΡΙΑΣΜΟΣ</a></li>
```

```
<?php
```

```
if (isset($_SESSION["logged"])) {
```

```
    echo '<li class="nav-link"><a href="#"  
onclick="logoutHandler()">ΑΠΟΣΥΝΔΕΣΗ</a></li>';
```

```
    }
```

```
?>
```

```
</ul>
```

```
<a href="#" class="hamburger closed">
```

```
<ion-icon name="menu"></ion-icon>
```

```
</a>
```

```
</nav>
```

```
<script>
```

```
let links = document.querySelectorAll("nav ul li");
```

```
switch (window.location.pathname) {
```

```
    case `${baseUrl}/`:
```

```
        links[0].classList.add("active");
```

```
        break;
```

```
    case `${baseUrl}/index.php`:
```

```

        links[0].classList.add("active");

        break;

    case `${baseUrl}/routes/category.php`:

        links[1].classList.add("active");

        break;

    case `${baseUrl}/routes/account.php`:

        links[3].classList.add("active");

        break;

    default:

        break;

}

const hamburgerBtn = document.querySelector(".hamburger");
const navUl = document.querySelector("nav").querySelector("ul");

const closeIcon = '<ion-icon name="close"></ion-icon>';
const openIcon = '<ion-icon name="menu"></ion-icon>';

hamburgerBtn.addEventListener("click", (e) => {

    hamburger();

});

function hamburger() {

    if (hamburgerBtn.classList.contains("closed")) {

```

```

    hamburgerBtn.classList.add("open");

    hamburgerBtn.classList.remove("closed");

    hamburgerBtn.innerHTML = closeIcon;

    navUl.classList.add("open");

    navUl.classList.remove("closed");

} else {

    hamburgerBtn.classList.remove("open");

    hamburgerBtn.classList.add("closed");

    hamburgerBtn.innerHTML = openIcon;

    navUl.classList.add("closed");

    navUl.classList.remove("open");

}

}

</script>

```

Φάκελος css

account.css

```

input {

    height: 2rem;

    width: 100%;

    border: none;

    background: var(--secondary-clr);

    padding-inline: 0.5em;

    border-radius: 0.25em;

}

```

```
input::placeholder {  
  color: var(--secondary-clr-dark);  
}
```

```
.settings {  
  background: red;  
  width: 80%;  
  max-width: 50em;  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
  gap: 2em;  
  padding: 3em 1em;  
  border-radius: 1em;  
  background-color: var(--primary-clr);  
}
```

```
.settings div {  
  width: 50%;  
}
```

```
.settings div form {  
  display: flex;
```

```
justify-content: center;

align-items: center;

gap: 0.5em;

}
```

```
.settings div h4 {

padding-block: 1em;

opacity: 70%;

}
```

```
.delete-section {

display: inline-flex;

align-items: center;

justify-content: space-between;

}
```

Add_form.css

```
.form-container {

width: 80%;

max-width: 50em;

background: var(--primary-clr);

padding: 2.5em;

padding-block: 2.5rem;

border-radius: 1em;


display: flex;
```

```
flex-direction: column;

gap: 1rem;

}
```

```
input[type="text"] {

border: none;

width: 100%;

border-radius: 0.25em;

padding: 0.5em;

}
```

```
.form-section {

width: 100%;

display: flex;

flex-direction: column;

}
```

```
.form-section * {

margin-block: 0.5em;

}
```

```
.form-section textarea {

border: none;

border-radius: 0.25em;
```

```
    resize: none;  
  }
```

```
.answer-controls {  
  display: inline-flex;  
  width: 100%;  
  gap: 1em;  
}
```

```
form {  
  gap: 1em;  
  padding-block-start: 1em;  
}
```

```
.form-buttons {  
  width: 100%;  
  display: inline-flex;  
  align-items: flex-end;  
  justify-content: flex-end;  
  gap: 1em;  
}
```

```
.format-helper {  
  background: var(--primary-clr);
```



```
margin: 2em;

padding: 1em;

border-radius: 1em;
}
```

```
.format-helper tr {

margin-block: 0.5em;
}
```

```
.format-helper tr td:first {

text-align: end;
}
```

```
.format-helper tr td:last-child {

width: 80px;
}
```

```
.section-header {

width: 100%;

display: inline-flex;

align-items: center;

justify-content: space-between;
}
```

```
select {  
  
    margin-inline: 0;  
  
}
```

```
.form-wrapper {  
  
    display: flex;  
  
    flex-direction: column;  
  
    justify-content: center;  
  
    align-items: center;  
  
    min-width: 17em;  
  
    width: 80%;  
  
    max-width: 50em;  
  
}
```

Admin_panel.css

```
.admin-panel-container {  
  
    width: 80%;  
  
    display: flex;  
  
    flex-direction: column;  
  
    padding-block-end: 2em;  
  
}
```

```
.filter {  
  
    display: flex;  
  
    justify-content: flex-end;  
  
    align-items: center;
```

```
}
```

```
.quiz,
```

```
.rule,
```

```
.question {
```

```
    display: inline-flex;
```

```
    justify-content: space-between;
```

```
    align-items: center;
```

```
    width: 100%;
```

```
    margin-block: 0.5em;
```

```
}
```

```
.db-functions {
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    gap: var(--gap, 0.5em);
```

```
}
```

```
.db-functions div {
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    gap: var(--gap, 0.5em);
```

```
}
```

```
.db-functions div button {
```

```
width: 100%;  
  
max-width: 12.5rem;  
  
}
```

```
.user-link {  
  
width: 100%;  
  
padding: 1em;  
  
margin-block: 1em;  
  
background: var(--primary-clr);  
  
border-radius: 0.25em;  
  
transition: var(--transition);  
  
}
```

```
.user-link:hover,  
.user-link:focus {  
  
background: var(--secondary-clr-dark);  
  
color: var(--hover-color);  
  
}
```

End_game.css

```
.confetti-container {  
  
width: 100vw;  
  
height: 100vh;  
  
position: absolute;  
  
top: 0;  
  
left: 0;
```

```

overflow: hidden;

pointer-events: none !important;
}

.confetti {

width: 15px;

height: 15px;

background-color: #ffd730;

position: absolute;

left: 50%;

animation: confetti 5s ease-in-out -2s infinite;

transform-origin: left top;

pointer-events: none !important;

box-shadow: 1px 1px 5px -1px rgba(0, 0, 0, 0.3);
}

.confetti:nth-child(2n) {

background-color: #69b1f5;

left: 10%;

animation-delay: 0;
}

.confetti:nth-child(2n + 1) {

background-color: #5cbf7d;

```

```
    left: 20%;

    animation-delay: -5s;
}

.confetti:nth-child(2n + 2) {

    background-color: #ff9643;

    left: 30%;

    animation-delay: -3s;
}

.confetti:nth-child(2n + 3) {

    background-color: #ff2a00;

    left: 40%;

    animation-delay: -2.5s;
}

.confetti:nth-child(2n + 4) {

    background-color: #ae00ff;

    left: 50%;

    animation-delay: -4s;
}

.confetti:nth-child(2n + 5) {

    background-color: #0c00b0;

    left: 60%;

    animation-delay: -6s;
}

.confetti:nth-child(2n + 6) {
```

```

background-color: #bf5e34;

left: 70%;

animation-delay: -1.5s;
}

.confetti:nth-child(2n + 7) {

background-color: #ff6ceb;

left: 80%;

animation-delay: -2s;
}

.confetti:nth-child(2n + 8) {

background-color: #27b8b0;

left: 90%;

animation-delay: -3.5s;
}

@keyframes confetti {
0% {

transform: rotateZ(15deg) rotateY(0deg) translate(0, 0);

}

25% {

transform: rotateZ(5deg) rotateY(360deg) translate(-5vw, 20vh);

}

50% {

transform: rotateZ(15deg) rotateY(720deg) translate(5vw, 60vh);

```

```

}

75% {

    transform: rotateZ(5deg) rotateY(1080deg) translate(-10vw, 80vh);

}

100% {

    transform: rotateZ(15deg) rotateY(1440deg) translate(10vw, 110vh);

}

}

```

```

h2 {

    position: absolute;

    box-shadow: none;

    transform: rotate(-15deg);

    font-size: 5em;

    opacity: 10%;

}

```

```

p {

    font-size: 3em;

    opacity: 50%;

}

```

```

    grades.css

.category-grades,

.rule-grades,

.question-grades {

```



```
width: 100%;  
  
display: flex;  
  
flex-direction: column;  
  
align-items: center;  
  
justify-content: center;  
  
margin-block: 1em;  
  
}
```

```
.category-grade,  
.rule-grade,  
.question-grade {  
  
width: 80%;  
  
max-width: 50em;  
  
display: inline-flex;  
  
justify-content: space-between;  
  
align-items: center;  
  
box-shadow: 0 2px 2px -2px var(--underline);  
  
padding-block: 1em;  
  
}
```

```
.category-progress-bar,  
.rule-progress-bar,  
.question-progress-bar {  
  
width: 4em;
```

```
aspect-ratio: 1;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

position: relative;

background: green;

z-index: 5;

}
```

```
.category-progress-bar::before,
.rule-progress-bar::before,
.question-progress-bar::before {

  position: absolute;

  content: "";

  width: 130%;

  height: 130%;

  border-radius: 50%;

  background: conic-gradient(

    var(--conic-gradient-color, red) var(--conic-gradient-percentage, 0%),

    var(--underline) 0

  );

  z-index: 4;

}
```

```
.category-progress-bar::after,  
.rule-progress-bar::after,  
.question-progress-bar::after {  
    position: absolute;  
    content: "";  
    content: attr(data-grade);  
    width: 100%;  
    height: 100%;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    border-radius: 50%;  
    background: var(--bg);  
    z-index: 6;  
}
```

index.css

```
.dialog {  
    width: 80%;  
    max-width: 40em;  
    background: var(--secondary-clr);  
    border-radius: 1em;  
    display: flex;  
    flex-direction: column;  
    padding: 2em;
```

```
line-height: 1.5;

position: relative;

position: absolute;
}
```

```
.dialog::before {

position: absolute;

content: "";

width: 5em;

aspect-ratio: 1;

background: inherit;

align-self: center;

transform: rotate(45deg);

bottom: -1em;

border-radius: 1em;
}
```

```
#story-lines {

margin-block-end: 1em;
}
```

```
.dialog-buttons {

width: 100%;

display: flex;

flex-direction: row-reverse;
```

```
    justify-content: space-between;
}
```

```
.answer-buttons {
    width: 100%;
    display: none;
    flex-direction: row-reverse;
    justify-content: center;
    gap: 1em;
}
```

```
.next-btns {
    display: inline-flex;
    flex-direction: row !important;
    gap: var(--gap, 0.5em);
}
```

```
.story-images {
    display: inline-flex;
    align-items: flex-end;
    max-height: 20em;
    margin-block-start: 14em;
    width: 100%;
}
```

```
.xenia,  
  
.bronco {  
  
    height: 100%;  
  
    max-height: 50em;  
  
    display: flex;  
  
    align-items: center;  
  
    margin-block: 3em;  
  
    width: 100%;  
  
}
```

```
.xenia {  
  
    justify-content: flex-start;  
  
}
```

```
.bronco {  
  
    justify-content: flex-end;  
  
}
```

```
.xenia img,  
  
.bronco img {  
  
    height: 100%;  
  
    max-height: 17em;  
  
}
```

```
.bronco img {  
    max-height: 6em;  
}
```

```
.bronco {  
    height: 50%;  
}
```

```
.play-buttons {  
    display: inline-flex;  
    width: 100%;  
    align-items: center;  
    justify-content: flex-end;  
    gap: var(--gap, 0.5em);  
}
```

login.css

```
.login-form {  
    background: var(--primary-clr);  
    width: 100%;  
    margin: 1rem;  
    padding: 1rem;  
    display: flex;  
    flex-direction: column;  
    align-items: center;
```

```
border-radius: 1rem;

height: 100%;


min-width: 50%;

width: max-content;

}
```

```
.form-header {

font-size: x-large;

margin-block: 1.5rem;

}
```

```
.form-selector {

display: flex;

flex-direction: row;

margin-block-end: 1.5rem;

}
```

```
.form-selector a {

padding: 0.5rem;

background: var(--secondary-clr);

}
```

```
.form-selector a:first-child,
```



```
.input-container label {  
  border-radius: 0.25em 0 0 0.25rem;  
}
```

```
.form-selector a:last-child,  
.input-container input {  
  border-radius: 0 0.25em 0.25em 0;  
}
```

```
.form-selector a:hover,  
.form-selector a:focus {  
  color: var(--hover-color);  
  background: var(--secondary-clr-dark);  
  max-width: 100%;  
}
```

```
.form-inputs {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  width: 100%;  
  margin-block-end: 2rem;  
}
```

```
.input-container {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  margin-block: 0.5rem;  
  height: 2rem;  
  width: 100%;  
  max-width: 20rem;  
}
```

```
.input-container input,  
.input-container label {  
  height: 100%;  
  width: 100%;  
  border: none;  
}
```

```
.input-container input {  
  background: var(--secondary-clr);  
  padding-inline: 0.5em;  
}
```

```
.input-container input::placeholder {
```

```
    color: var(--secondary-clr-dark);  
}
```

```
.input-container label {  
    background: var(--secondary-clr-dark);  
    width: 2rem;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
}
```

```
.input-container label ion-icon {  
    color: var(--bg);  
}
```

```
.form-extras a {  
    text-decoration: underline;  
}
```

```
.form-extras a:hover,  
.form-extras a:focus {  
    color: var(--hover-color);  
}
```

```
a.active {  
  
  background: var(--secondary-clr-dark);  
  
  color: var(--hover-color);  
  
}
```

```
.input-sections {  
  
  display: flex;  
  
  flex-direction: row;  
  
  gap: var(--gap, 1em);  
  
  width: max-content;  
  
}
```

```
select {  
  
  border: 0px solid var(--primary-clr);  
  
  border-radius: 0 0.25em 0.25em 0;  
  
  cursor: pointer;  
  
  padding: 0.5em;  
  
  margin-inline-start: 0;  
  
  background: var(--secondary-clr);  
  
  margin: 0;  
  
  height: 100%;  
  
}
```

```
label {  
  
  aspect-ratio: 1;
```

```
}
```

```
@media only screen and (max-width: 36em) {
```

```
  .input-sections {
```

```
    flex-direction: column;
```

```
  }
```

```
}
```

```
p.header {
```

```
  margin-block: 1em;
```

```
  text-align: center;
```

```
}
```

```
label.custom-label {
```

```
  width: max-content;
```

```
  height: 100%;
```

```
  background: var(--secondary-clr);
```

```
  border-radius: 0;
```

```
  padding: 0.5em;
```

```
  white-space: nowrap;
```

```
  font-size: 0.8em;
```

```
}
```

```
.page-content {
```

```

    height: auto;
}

    quiz.css
.quiz-container {
    background: var(--primary-clr);
    border-radius: 1em;
    width: 80%;
    max-width: 50em;
    padding: 1em;
}

.quiz-question {
    padding: 0.5em 1em;
}

.quiz-question,
.quiz-answers {
    width: 100%;
}

.quiz-question p {
    display: inline-flex;
    align-items: center;
    gap: 1em;
}

```

```
.quiz-answers {  
  display: flex;  
  flex-direction: column;  
  align-items: flex-start;  
  justify-content: center;  
  gap: var(--gap, 0.5em);  
}
```

```
.answer-container {  
  width: 100%;  
  background: var(--secondary-clr);  
  display: flex;  
  border-radius: 0.25em;  
}
```

```
.answer-container:focus-within,  
.answer-container:hover {  
  background: var(--secondary-clr-dark);  
  color: var(--hover-color);  
}
```

```
.answer-container a {  
  width: 100%;
```

```
display: flex;

justify-content: flex-start;

padding: 0.5em 1em;

}
```

```
.answer-container a:hover,

.answer-container a:focus {

    color: var(--hover-color);

}
```

```
.quiz-answers div[draggable="true"] {

    cursor: grab;

    width: fit-content;

    padding: 1em;

    display: flex;

}
```

```
.page-content {

    justify-content: center;

    padding-block-start: 0em;

}
```

```
.quiz-indicator {

    width: 1em;
```



```
height: 3em;

position: absolute;

left: 0;

top: 0;

background: red;

color: white;

font-size: 4rem;

font-weight: 900;

text-align: end;

border-radius: 0 0 4em 0;

display: flex;

align-items: flex-end;

justify-content: flex-start;


padding: 0 1em 0.1em 0;
}
```

```
.empty span {

background: none;

}
```

rules.css

```
.rule-container {

width: 80%;

max-width: 100rem;

background: var(--primary-clr);
```

```
display: flex;

flex-direction: column;

align-items: center;

border-radius: 1rem;

padding: 1em;

margin-block-end: 1em;

padding-block-start: 3em;
}
```

```
.rule-body {

width: 100%;

display: flex;

flex-direction: column;

align-items: center;
}
```

```
.rule {

width: 80%;

display: flex;

align-items: center;

justify-content: center;

background: var(--secondary-clr);

padding: 1rem;

margin: 1rem;
}
```

```
border-radius: 1rem;

font-size: 1.2rem;

}
```

```
.rule-header {

font-size: 3rem;

margin-block-end: 1rem;

}
```

```
.rule-example-header {

text-decoration: underline;

font-weight: bold;

text-align: center;

}
```

```
.rule-example .example {

text-align: center;

}
```

```
.rule-section p {

margin-block-end: 1rem;

}
```

```
.rule-section p:last-child {

margin-block-end: 0;
```

```
}
```

```
.buttons-container {  
  width: 80%;  
  display: flex;  
  gap: 1em;  
  align-items: center;  
  justify-content: space-between;  
}
```

```
.next-buttons {  
  display: flex;  
  gap: 1em;  
  align-items: center;  
  justify-content: flex-end;  
}
```

styles.css

```
@import  
url("https://fonts.googleapis.com/css2?family=Comfortaa:wght@300;400;500;600;700&  
display=swap");
```

```
* {  
  margin: 0;  
  box-sizing: border-box;
```

```
z-index: 0;

font-family: Comfortaa, sans-serif;
}
```

```
html {

  --bg: white;

  --primary-clr: #c4c4c4;

  --secondary-clr: #b2b2b2;

  --secondary-clr-dark: #6d6d6d;

  --underline: rgba(0, 0, 0, 0.5);

  --hover-bg: rgba(0, 0, 0, 0.25);

  --hover-color: white;

  --special-text-color: white;

  --btn-clr: white;

  --text-clr: black;

  --transition: 0.5s ease-in-out;
}
```

```
body {

  background: var(--bg);

  min-height: 100vh;

  height: 100%;
}
```

```
a {  
    color: inherit;  
    text-decoration: none;  
    display: flex;  
    align-content: center;  
    justify-content: center;  
}
```

h1,

h2,

h3,

h4,

h5,

h6,

```
p {
```

```
    color: var(--text-clr);
```

```
}
```

```
nav ul {
```

```
    display: flex;
```

```
    flex-direction: row;
```

```
    align-items: center;
```

```
    justify-content: end;
```

```
    width: 100%;
```

```
height: 100%;  
}
```

```
.nav-link {  
  
  list-style: none;  
  
  transition: var(--transition);  
  
  font-size: 1rem;  
  
  display: flex;  
  
  align-content: center;  
  
  justify-content: center;  
  
  
  font-size: small;  
  
  
  color: var(--text-clr) !important;  
}
```

```
.hamburger {  
  
  position: absolute;  
  
  left: 0;  
  
  top: 0;  
  
  padding: 1em;  
  
  margin: 0.1em;  
  
  font-size: larger;  
  
  border-radius: 1em;
```

```
transition: var(--transition);  
  
display: none;  
  
}
```

```
.hamburger:hover,  
  
.hamburger:focus {  
  
background: var(--hover-bg);  
  
color: var(--hover-color);  
  
}
```

```
@media only screen and (max-width: 49em) {  
  
nav ul {  
  
padding-inline: 0;  
  
display: flex;  
  
flex-direction: column;  
  
align-items: center;  
  
justify-content: flex-end;  
  
width: 100%;  
  
height: 100%;  
  
}
```

```
nav ul.closed {  
  
display: none;  
  
}
```



```
nav ul.open {  
  display: flex;  
}
```

```
.nav-link {  
  list-style: none;  
  transition: var(--transition);  
  font-size: 1rem;  
  display: flex;  
  align-content: center;  
  justify-content: flex-end;  
  font-size: small;  
  width: 100%;  
}
```

```
.hamburger {  
  display: flex;  
}  
}
```

```
.nav-link:hover,  
.nav-link:focus-within,  
.nav-link.active:hover,
```

```
.nav-link.active:focus-within {  
  
  background: var(--hover-bg);  
  
  color: var(--hover-color) !important;  
  
  cursor: pointer;  
  
  text-decoration: underline;  
  
}
```

```
.nav-link.active {  
  
  background: var(--secondary-clr);  
  
}
```

```
.nav-link a {  
  
  display: block;  
  
  padding: 1.5rem;  
  
  width: 100%;  
  
  text-align: end;  
  
}
```

```
nav {  
  
  background: var(--primary-clr);  
  
  min-height: 3.875em;  
  
  position: absolute;  
  
  top: 0;  
  
  left: 0;
```

```
width: 100%;  
z-index: 999;  
}
```

```
.page-content {  
width: 100%;  
height: 100vh;  
top: 0;  
z-index: -1;  
display: flex;  
align-items: center;  
justify-content: flex-start;  
padding-block-start: 8em;  
flex-direction: column;  
}
```

```
button,  
a.button {  
position: relative;  
padding: 0.75em 1rem;  
border: none;  
cursor: pointer;  
background-color: #00000;  
z-index: 1;
```

```
display: flex;

align-items: center;

justify-content: center;

background: var(--secondary-clr-dark);

color: var(--hover-color);

color: var(--btn-clr);

border-radius: 0.25rem;

gap: 0.5em;

transition: var(--transition);


font-weight: 700;
}
```

```
button .fi,

a.button .fi {

    color: inherit;

}
```

```
button:hover,

button:focus,

a.button:hover,

a.button:focus {

    background: var(--bg);

    color: var(--secondary-clr-dark);
```

```
    box-shadow: 0 1em 1em -1em var(--underline);  
}
```

```
form {  
    width: 100%;  
    height: 100%;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    flex-direction: column;  
}
```

```
/* BUTTON SPECIFIC CLASSES */
```

```
.blue {  
    background: rgb(44, 118, 255) !important;  
}
```

```
.blue:hover,  
.blue:focus {  
    background: rgb(122, 168, 255) !important;  
    color: var(--hover-color) !important;  
}
```

```
.red {
```

```
background: rgb(255, 88, 88) !important;
}

.red:hover,
.red:focus {
background: rgb(255, 162, 162) !important;
color: var(--hover-color) !important;
}

.green {
background: rgb(0, 182, 106) !important;
}

.green:hover,
.green:focus {
background: rgb(122, 216, 177) !important;
color: var(--hover-color) !important;
}

.inverse {
background: white !important;
color: var(--secondary-clr-dark) !important;
}
```

```
.inverse:hover,  
.inverse:focus {  
    background: var(--secondary-clr-dark) !important;  
    color: var(--hover-color) !important;  
}
```

```
.yellow {  
    background: rgb(255, 213, 0) !important;  
    text-shadow: 1px 1px black;  
}
```

```
.yellow:hover,  
.yellow:focus {  
    background: rgb(255, 231, 109) !important;  
    color: var(--hover-color) !important;  
}
```

```
.purple {  
    background: rgb(132, 0, 255) !important;  
}
```

```
.purple:hover,  
.purple:focus {  
    background: rgb(193, 60, 255) !important;
```

```
    color: var(--hover-color) !important;
}
```

```
.green-inverse {
    background: rgb(0, 182, 106) !important;
    color: white;
    font-weight: bold;
    box-shadow: inset 0 0 0 0.12em rgb(0, 182, 106);
}
```

```
.green-inverse:focus,
.green-inverse:hover {
    background: var(--bg) !important;
    color: rgb(0, 182, 106);
    font-weight: bold;
    box-shadow: inset 0 0 0 0.12em rgb(0, 182, 106);
}
```

```
i {
    display: flex;
    align-items: center;
    justify-content: center;
}
```



```
.fade-in {  
  
  animation-name: fade-in;  
  
  animation-duration: 0.5s;  
  
  animation-timing-function: ease-in;  
  
}
```

```
@keyframes fade-in {  
  
  0% {  
  
    opacity: 0%;  
  
  }
```

```
  100% {  
  
    opacity: 100%;  
  
  }  
}
```

```
select {  
  
  border: 1px solid var(--primary-clr);  
  
  border-radius: 0.25em;  
  
  cursor: pointer;  
  
  padding: 0.5em 1em;  
  
  margin-inline-start: 0.5em;  
  
  background: var(--secondary-clr);  
  
}
```

```
h2 {  
  
  width: 100%;  
  
  margin-block: 1em 0.5em;  
  
  padding-block: 0.5em;  
  
  box-shadow: 0 2px 2px -2px var(--underline);  
  
}
```

```
.special-text-0,  
.special-text-1,  
.special-text-2,  
.special-text-3,  
.special-text-4,  
.special-text-5,  
.special-text-6,  
.special-text-7,  
.special-text-8,  
.special-text-9,  
.special-text-10 {  
  
  position: relative;  
  
  padding: 0.3em;  
  
  border-radius: 0.25em;  
  
  line-height: 1.8;  
  
  color: var(--special-text-color);
```

text-shadow: 1px 1px black;

min-width: 75px;

min-height: 40px;

text-shadow: none;

font-weight: 700;

text-shadow: 1px 1px rgba(255, 255, 255, 0.3), -1px -1px rgba(0, 0, 0, 0.3);

background: rgba(255, 255, 255, 0.5);

}

.special-text-0 {

--special-text-color: white;

text-shadow: none;

}

.special-text-1 {

--special-text-color: #69b1f5;

}

.special-text-2 {

--special-text-color: #5cbf7d;

}

.special-text-3 {

```
--special-text-color: #ffd730;  
}
```

```
.special-text-4 {  
  --special-text-color: #ff9643;  
}
```

```
.special-text-5 {  
  --special-text-color: #ff2a00;  
}
```

```
.special-text-6 {  
  --special-text-color: #ae00ff;  
}
```

```
.special-text-7 {  
  --special-text-color: #0c00b0;  
}
```

```
.special-text-8 {  
  --special-text-color: #bf5e34;  
}
```

```
.special-text-9 {
```

```
--special-text-color: #ff6ceb;  
}
```

```
.special-text-10 {  
  --special-text-color: #27b8b0;  
}
```

```
tr {  
  display: flex;  
  align-items: center;  
  justify-content: flex-end;  
}
```

```
td {  
  display: inline-flex;  
  gap: 1em;  
  margin-inline: 0.5em 1em;  
}
```

```
input:disabled {  
  cursor: not-allowed;  
}
```

```
.footer {
```

```
width: 100%;  
height: 2em;  
position: absolute;  
bottom: 0;  
pointer-events: none;  
}
```

```
.swal2-title {  
  box-shadow: 0 0;  
}
```

```
.empty {  
  background: #eee;  
  border-radius: 0.25em;  
  width: 5em;  
  height: 2.5em;
```

```
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
ion-icon {  
  aspect-ratio: 1;
```

```
}
```

```
.backdrop {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    background: url(../images/Shapes.png);  
    background-position: center;  
    z-index: -1;  
    filter: blur(5px);  
    opacity: 35%;  
    background-repeat: no-repeat;  
    background-size: cover;  
}
```

user_info.css

```
.user-info {  
    font-size: 2em;  
    font-weight: 900;  
}
```

```
.user-info p {  
    text-align: center;  
}
```

```
.user-name-email {  
  
  display: inline-flex;  
  
  gap: 0.5em;  
  
}
```

```
.evaluations,  
  
.evaluations-list,  
  
.evaluation {  
  
  width: 100%;  
  
  display: grid;  
  
  place-items: center;  
  
}
```

```
.evaluations-list {  
  
  display: flex;  
  
  flex-direction: column;  
  
  gap: 2em;  
  
  margin-block: 2em;  
  
  padding: 0;  
  
}
```

```
.evaluation {  
  
  width: 100%;
```



```
gap: 1em;
```

```
padding: 1em;
```

```
}
```

```
.evaluation:nth-child(2n) {
```

```
background: rgba(0, 0, 0, 0.1);
```

```
}
```

```
.evaluation-grades {
```

```
display: inline-flex;
```

```
gap: 2em;
```

```
}
```

```
.evaluation-item {
```

```
border-radius: 0.5em;
```

```
padding: 0.5em;
```

```
}
```

```
.evaluation:nth-child(2n) .evaluation-item:nth-child(2n + 1) {
```

```
background: rgba(0, 0, 0, 0.15);
```

```
}
```

```
.evaluation:nth-child(2n + 1) .evaluation-item:nth-child(2n) {
```

```
background: rgba(0, 0, 0, 0.15);
```

```
}
```

```
.grade-info {  
  display: flex;  
  flex-direction: column;  
}
```

```
.grade-info p {  
  display: inline-flex;  
  align-items: center;  
  width: 100%;  
  padding-block: 0.5em;  
}
```

```
.value {  
  width: 100%;  
  text-align: end;  
  font-style: italic;  
  text-decoration: underline;  
}
```

```
.evaluation-item {  
  list-style: none;  
}
```

Φάκελος includes

Prototype.php

completeGame.php

```
<?php
```

```
session_start();
```

```
if (!isset($_SESSION['logged'])) {  
    header("location: " . $baseUrl);  
    exit();  
}
```

```
require_once "db.info.php";
```

```
if (!isset($_POST["quizID"])) {  
    echo '{"error": "noIDProvided"}';  
}
```

```
$quizToCheck = $_POST["quizID"];
```

```
$sql = "SELECT * FROM quizzes ORDER BY quizID DESC;";
```

```
$stmt = mysqli_stmt_init($conn);
```

```
if (!mysqli_stmt_prepare($stmt, $sql)) {  
    echo ('{"error": "stmtFailed"}');
```

```

        exit();
    }

mysqli_stmt_execute($stmt);

$resultData = mysqli_stmt_get_result($stmt);
if ($row = mysqli_fetch_assoc($resultData)) {
    if ($row["quizID"] == $quizToCheck) {
        echo '{"error": "none", "answer": "true"}';
        exit();
    }
}
echo '{"error": "none", "answer": "false"}';
exit();

```

db.info.php

```

<?php

$serverName = "localhost";

$dbUsername = "root";

$dbPassword = "";

$dbName = "orthopaignia ";

$conn = mysqli_connect($serverName, $dbUsername, $dbPassword, $dbName);

if (!$conn) {

```

```

        die("Connection Failed!" . mysqli_connect_error());
    }

    deleteAccount.php
<?php

session_start();

require_once "db.info.php";
require_once "functions.php";

if (isset($_SESSION['logged']) && isset($_SESSION['username']) &
isset($_POST["submit"])) {

    // CHECK IF USER EXISTS

    if ($row = uidExists($conn, $_SESSION["username"], $_SESSION["username"])) {

        deleteEvaluationsForUser($conn, $row['userID']);

        // DELETE FROM ADMINISTRATORS

        $sql = "DELETE FROM administrators WHERE userID = ?;";

        $stmt = mysqli_stmt_init($conn);

        if (!mysqli_stmt_prepare($stmt, $sql)) {

            echo ("error=administratorsDeletionFailed");

            exit();

        }

        mysqli_stmt_bind_param($stmt, "i", $row['userID']);

        mysqli_stmt_execute($stmt);

```

```

// FINALLY DELETE FROM USERS

$sql = "DELETE FROM users WHERE userID = ?;";

$stmt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmt, $sql)) {

    echo ("error=userDeletionFailed");

    exit();

}

mysqli_stmt_bind_param($stmt, "i", $row['userID']);

mysqli_stmt_execute($stmt);


session_unset();

session_destroy();


echo ("error=none");

exit();

}

} else {

    echo ("error=notLoggedOrNoUsername");

}


deleteQuestion.php
<?php


session_start();


if (!isset($_SESSION['logged']) && !isset($_SESSION['isAdmin'])) {

```

```

    echo "error=unauthorized";

    exit();
}

require_once "db.info.php";

if (!isset($_POST['submit']) || !isset($_POST['questionID'])) {

    echo "error=notEnoughVariables";

    exit();
}

$questionID = $_POST['questionID'];

$sqlDeleteAnswers = "DELETE FROM answers WHERE questionID = ?;";

$stmtDeleteAnswers = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtDeleteAnswers, $sqlDeleteAnswers)) {

    echo ("error=stmtFailed");

    exit();
}

mysqli_stmt_bind_param($stmtDeleteAnswers, "i", $questionID);

mysqli_stmt_execute($stmtDeleteAnswers);

mysqli_stmt_close($stmtDeleteAnswers);

```

```
$sqlDeleteQuestion = "DELETE FROM questions WHERE questionID = ?;";
```

```
$stmtDeleteQuestion = mysqli_stmt_init($conn);
```

```
if (!mysqli_stmt_prepare($stmtDeleteQuestion, $sqlDeleteQuestion)) {
```

```
    echo ("error=stmtFailed");
```

```
    exit();
```

```
}
```

```
mysqli_stmt_bind_param($stmtDeleteQuestion, "i", $questionID);
```

```
mysqli_stmt_execute($stmtDeleteQuestion);
```

```
mysqli_stmt_close($stmtDeleteQuestion);
```

```
echo "error=none";
```

```
    deleteQuiz.php
```

```
<?php
```

```
session_start();
```

```
if (!isset($_SESSION['logged']) && !isset($_SESSION['isAdmin'])) {
```

```
    echo "error=unauthorized";
```

```
    exit();
```

```
}
```

```
require_once "db.info.php";
```

```
require_once "functions.php";
```



```

if (!isset($_POST['submit']) || !isset($_POST['quizID'])) {

    echo "error=notEnoughVariables";

    exit();

}

$quizID = $_POST['quizID'];

// DELETE CONNECTED RULES

$sqlGetRules = "SELECT * FROM rules WHERE quizID = ?";

$stmtGetRules = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtGetRules, $sqlGetRules)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmtGetRules, "i", $quizID);

mysqli_stmt_execute($stmtGetRules);

$resultData = mysqli_stmt_get_result($stmtGetRules);

while ($row = mysqli_fetch_assoc($resultData)) {

    // FOREACH RULE, DELETE RULE

    deleteRule($conn, $row["ruleID"]);

```

```

}

deleteEvaluationsForQuiz($conn, $quizID);

$sqlGetQuestions = "SELECT * FROM questions WHERE quizID = ?;";

$stmtGetQuestions = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtGetQuestions, $sqlGetQuestions)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmtGetQuestions, "i", $quizID);

mysqli_stmt_execute($stmtGetQuestions);

$resultData = mysqli_stmt_get_result($stmtGetQuestions);

while ($row = mysqli_fetch_assoc($resultData)) {

    // FOREACH EVALUATION, DELETE CONNECTED GARDES

    deleteQuestion($conn, $row["questionID"]);

}

mysqli_stmt_close($stmtGetQuestions);

$sqlDeleteQuiz = "DELETE FROM quizzes WHERE quizID = ?;";

$stmtDeleteQuiz = mysqli_stmt_init($conn);

```

```

if (!mysqli_stmt_prepare($stmtDeleteQuiz, $sqlDeleteQuiz)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmtDeleteQuiz, "i", $quizID);

mysqli_stmt_execute($stmtDeleteQuiz);

mysqli_stmt_close($stmtDeleteQuiz);


echo "error=none";

    deleteRule.php
<?php

session_start();

if (!isset($_SESSION['logged']) && !isset($_SESSION['isAdmin'])) {

    echo "error=unauthorized";

    exit();

}

require_once "db.info.php";

require_once "functions.php";

if (!isset($_POST['submit']) || !isset($_POST['ruleID'])) {

    echo "error=notEnoughVariables";

```

```

        exit();
    }

    $ruleID = $_POST['ruleID'];

    deleteRulesFunction($conn, $ruleID);

    echo "error=none";

    exportXLSX.php
<?php

session_start();

if (!isset($_SESSION['logged']) && !isset($_SESSION['isAdmin'])) {
    header("location: " . $baseUrl);
    exit();
}

require_once "db.info.php";
require_once "functions.php";

function filterData(&$str)
{
    $str = preg_replace("/\t/", "\\t", $str);
    $str = preg_replace("/\r?\n/", "\\n", $str);
    if (strpos($str, "")) {

```

```

        $str = "" . str_replace("'", "''", $str) . "'";
    }
}

$filename = "orthopaignia_data_" . date('d-m-Y') . ".csv";

$fields = array("ID", "ONOMA", "ΕΠΩΝΥΜΟ", "E-MAIL", "ONOMA ΧΡΗΣΤΗ",
"EΚΠΑΙΔΕΥΣΗ");

// CSV Format

$excelData = implode(", ", array_values($fields)) . "\n";

// FETCH USERDATA FROM DATABASE

$sql = "SELECT * FROM users ORDER BY userID ASC;";

$stmt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmt, $sql)) {
    echo ("error=stmtFailed");

    exit();
}

mysqli_stmt_execute($stmt);

$resultData = mysqli_stmt_get_result($stmt);

$users = array();

```

```

while ($row = mysqli_fetch_assoc($resultData)) {

    // FETCH DATA FOR EACH INDIVIDUAL

    $lineData = array(

        $row["userID"],

        decrypt($row["userFirstName"]),

        decrypt($row["userLastName"]),

        decrypt($row["userEmail"]),

        decrypt($row["userUsername"]),

        convertEducationToReadable($row["userEducation"])

    );

    array_walk($lineData, 'filterData');

    $excelData .= implode(", ", array_values($lineData)) . "\n";

    array_push($users, $lineData);

}

```

```

mysqli_stmt_close($stmt);

```

```

// FOREACH QUIZ, FETCH USER EVALUATIONS

```

```

// FOR IT TO HAVE THE BELOW STRUCTURE

```

```

// |      | USER | GRADE_1 | GRADE_1 |

```

```

// | STAGE_1 |

```

```

// |      | USER_1 | GRADE_1 | GRADE_2 |

```

```

// |      | USER_2 | GRADE_1 | GRADE_2 |

```

```

// | STAGE_2 |

```

```

// |      | USER_1 | GRADE_1 | GRADE_2 |
// |      | USER_2 | GRADE_1 | GRADE_2 |

$fields2 = array("ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ", "ΧΡΗΣΤΗΣ", "ΠΡΩΤΗ ΠΡΟΣΠΑΘΕΙΑ", "",
"", "", "ΤΕΛΕΥΤΑΙΑ ΠΡΟΣΠΑΘΕΙΑ", "", "", "");

$fields3 = array("", "", "ΗΜΕΡΟΜΗΝΙΑ", "ΠΟΣΟΣΤΟ ΕΠΙΤΥΧΙΑΣ", "ΧΡΟΝΟΣ
ΟΛΟΚΛΗΡΩΣΗΣ", "ΧΡΟΝΟΣ ΜΕΛΕΤΗΣ", "ΗΜΕΡΟΜΗΝΙΑ", "ΠΟΣΟΣΤΟ
ΕΠΙΤΥΧΙΑΣ", "ΧΡΟΝΟΣ ΟΛΟΚΛΗΡΩΣΗΣ", "ΧΡΟΝΟΣ ΜΕΛΕΤΗΣ");

$excelData .= "\n\n";

$excelData .= implode(", ", array_values($fields2)) . "\n";

$excelData .= implode(", ", array_values($fields3)) . "\n";


// FETCH ALL DATA

$sqlQuizzes = "SELECT * FROM quizzes;";

$stmtQuizzes = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtQuizzes, $sqlQuizzes)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_execute($stmtQuizzes);


$quizzesResultData = mysqli_stmt_get_result($stmtQuizzes);

$quizzes = array();

```

```
while ($rowQuizzes = mysqli_fetch_assoc($quizzesResultData)) {  
    array_push($quizzes, $rowQuizzes);  
}
```

```
mysqli_stmt_close($stmtQuizzes);
```

```
$sqlEvals = "SELECT * FROM evaluations;";
```

```
$stmtEvals = mysqli_stmt_init($conn);
```

```
if (!mysqli_stmt_prepare($stmtEvals, $sqlEvals)) {
```

```
    echo ("error=stmtFailed");
```

```
    exit();
```

```
}
```

```
mysqli_stmt_execute($stmtEvals);
```

```
$evalsResultData = mysqli_stmt_get_result($stmtEvals);
```

```
$evals = array();
```

```
while ($rowEvals = mysqli_fetch_assoc($evalsResultData)) {
```

```
    array_push($evals, $rowEvals);
```

```
}
```

```
mysqli_stmt_close($stmtEvals);
```



```

$sqlGrades = "SELECT * FROM grades;";

$stmtGrades = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtGrades, $sqlGrades)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_execute($stmtGrades);


$gradesResultData = mysqli_stmt_get_result($stmtGrades);


$grades = array();


while ($rowGrades = mysqli_fetch_assoc($gradesResultData)) {

    array_push($grades, $rowGrades);

}


mysqli_stmt_close($stmtGrades);


foreach ($quizzes as $quiz) {

    $emptyRow = array("");

    $excelData .= implode(", ", array_values($emptyRow)) . "\n";

    $quizTitle = array($quiz["quizTitle"]);

```

```
$excelData .= implode(" ", array_values($quizTitle)) . "\n";
```

```
foreach ($evals as $eval) {
```

```
    if ($quiz["quizID"] == $eval["quizID"]) {
```

```
        $connectedUser = "";
```

```
        foreach ($users as $user) {
```

```
            if ($user[0] == $eval["userID"]) {
```

```
                $connectedUser = $user[4];
```

```
            }
```

```
        }
```

```
$firstAttempt = array();
```

```
$latestAttempt = array();
```

```
foreach ($grades as $grade) {
```

```
    if ($grade["gradeID"] == $eval["firstAttempt"]) {
```

```
        $firstAttempt = $grade;
```

```
    }
```

```
    if ($grade["gradeID"] == $eval["latestAttempt"]) {
```

```
        $latestAttempt = $grade;
```

```
    }
```

```
}
```

```

        $userData = array("", $connectedUser, $firstAttempt["gradeDate"],
($firstAttempt["successRatio"] * 100) . "%", $firstAttempt["answerTime"],
$firstAttempt["studyTime"], $latestAttempt["gradeDate"],
($latestAttempt["successRatio"] * 100) . "%", $latestAttempt["answerTime"],
$latestAttempt["studyTime"]);

```

```

        $excelData .= implode(", ", array_values($userData)) . "\n";

```

```

    }

```

```

}

```

```

}

```

```

// HEADERS FOR DOWNLOADING

```

```

header("Content-Type: text/csv");

```

```

header("Content-Disposition: attachment; filename=\"{$filename}\"");

```

```

echo $excelData;

```

```

exit;

```

```

    fetchGrades.php

```

```

<?php

```

```

session_start();

```

```

if (!isset($_SESSION['logged'])) {

```

```

    echo '{"error": "unauthorized"}';

```

```

    exit();

```

```

}

```

```

require_once "db.info.php";

require_once "functions.php";


if (!isset($_POST['submit']) || !isset($_POST['user'])) {

    echo '{"error": "notEnoughVariables"}';

    exit();

}


$userID = $_POST["user"];


if (!isset($_POST["evaluationID"])) {

    $returnData = array();

    $sqlEvals = "SELECT * FROM evaluations WHERE userID = ?;";

    $stmtEvals = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtEvals, $sqlEvals)) {

        echo ('{"error": "stmtFailed"}');

        exit();

    }

    mysqli_stmt_bind_param($stmtEvals, "i", $userID);

    mysqli_stmt_execute($stmtEvals);

    $evaluations = mysqli_stmt_get_result($stmtEvals);

```

```

$index = 0;

$evalData = array();

while ($row = mysqli_fetch_assoc($evaluations)) {

    // FOREACH EVALUATION, FETCH FIRST AND LATEST ATTEMPT AND
    RETURN INFO

    array_push($evalData, $row);


    // FETCH FIRST ATTEMPT

    $sqlFirstAttempt = "SELECT * FROM grades WHERE gradeID = ?;";

    $stmtFirstAttempt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtFirstAttempt, $sqlFirstAttempt)) {

        echo ('{"error": "stmtFailed"}');

        exit();

    }

    mysqli_stmt_bind_param($stmtFirstAttempt, "i", $row["firstAttempt"]);

    mysqli_stmt_execute($stmtFirstAttempt);


    $firstAttempt = mysqli_stmt_get_result($stmtFirstAttempt);

    $index = 0;


    while ($row2 = mysqli_fetch_assoc($firstAttempt)) {

        // FOREACH GRADE, RETURN ITS INFO

        array_push($evalData, "{\"firstAttemptData\": $row2}");

    }

```

```

// FETCH LATEST ATTEMPT

$sqlLatestAttempt = "SELECT * FROM grades WHERE gradeID = ?;";

$stmtLatestAttempt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtLatestAttempt, $sqlLatestAttempt)) {

    echo ('{"error": "stmtFailed"}');

    exit();

}

mysqli_stmt_bind_param($stmtLatestAttempt, "i", $row["latestAttempt"]);

mysqli_stmt_execute($stmtLatestAttempt);


$latestAttempt = mysqli_stmt_get_result($stmtLatestAttempt);

$index = 0;


while ($row3 = mysqli_fetch_assoc($latestAttempt)) {

    // FOREACH GRADE, RETURN ITS INFO

    array_push($evalData, "{\"latestAttemptData\": $row3}");

}

}

array_push($returnData, $evalData);

array_push($returnData, '{"error": "none"}');

$jsonData = json_encode($returnData);

```

```

    echo $jsonData;

} else {

    $evaluationID = $_POST["evaluationID"];

    $returnData = array();

    $sql = "SELECT * FROM grades WHERE evaluationID = ?;";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo ('{"error": "stmtFailed"}');

        exit();

    }

    mysqli_stmt_bind_param($stmt, "i", $evaluationID);

    mysqli_stmt_execute($stmt);

    $resultData = mysqli_stmt_get_result($stmt);

    $index = 0;

    while ($row = mysqli_fetch_assoc($resultData)) {

        // FOREACH GRADE, RETURN ITS INFO

        array_push($returnData, $row);

    }

    array_push($returnData, '{"error": "none"}');

    $jsonData = json_encode($returnData);

```

```

        echo $jsonData;
    }

    fetchQuestions.php
<?php

session_start();

if (!isset($_SESSION['logged']) && !isset($_SESSION['isAdmin'])) {

    header("location: " . $baseUrl);

    exit();

}

require_once "db.info.php";

$returnData = array();

if (isset($_POST["quizID"])) {

    $quizID = $_POST["quizID"];

    // FETCH QUESTIONS OF A RULE

    $sql = "SELECT * FROM questions WHERE quizID = ?";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo '{"error": "stmtFailed"}';

        exit();
    }

```



```
}
```

```
mysqli_stmt_bind_param($stmt, "i", $quizID);
```

```
mysqli_stmt_execute($stmt);
```

```
$resultData = mysqli_stmt_get_result($stmt);
```

```
$index = 0;
```

```
$answerData = array();
```

```
while ($row = mysqli_fetch_assoc($resultData)) {
```

```
    // FETCH ANSWERS FOREACH QUESTION
```

```
    array_push($answerData, $row);
```

```
    $answerIndex = 0;
```

```
    $sqlFetchWrongAnswers = "SELECT * FROM answers WHERE questionID = ?;";
```

```
    $stmtFetchWrongAnswers = mysqli_stmt_init($conn);
```

```
    if (!mysqli_stmt_prepare($stmtFetchWrongAnswers, $sqlFetchWrongAnswers)) {
```

```
        echo '{"error": "stmtFailed"}';
```

```
        exit();
```

```
    }
```

```
    mysqli_stmt_bind_param($stmtFetchWrongAnswers, "i", $row["questionID"]);
```

```
    mysqli_stmt_execute($stmtFetchWrongAnswers);
```

```

$resultWrongAnswers = mysqli_stmt_get_result($stmtFetchWrongAnswers);

while ($rowWrongAnswers = mysqli_fetch_assoc($resultWrongAnswers)) {

    array_push($answerData, $rowWrongAnswers);

    $answerIndex++;

}

array_push($returnData, $answerData);

    $index++;

}

} else {

    $sql = "SELECT * FROM questions;";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo ("error=stmtFailed");

        exit();

    }

    mysqli_stmt_execute($stmt);

    $resultData = mysqli_stmt_get_result($stmt);

    while ($row = mysqli_fetch_assoc($resultData)) {

```

```

        array_push($returnData, $row);
    }
}

$jsonData = json_encode($returnData);

echo $jsonData;

exit();

fetchQuizzes.php
<?php

session_start();

if (!isset($_SESSION['logged']) && !isset($_SESSION['isAdmin'])) {
    header("location: " . $baseUrl);
    exit();
}

require_once "db.info.php";

$sql = "SELECT * FROM quizzes;";

$stmt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmt, $sql)) {
    echo ("error=stmtFailed");
    exit();
}

```

```

}

mysqli_stmt_execute($stmt);

$resultData = mysqli_stmt_get_result($stmt);

$returnData = array();

while ($row = mysqli_fetch_assoc($resultData)) {

    array_push($returnData, $row);

}

$jsonData = json_encode($returnData);

echo $jsonData;

exit();

fetchRules.php
<?php

session_start();

if (!isset($_SESSION['logged']) && !isset($_SESSION['isAdmin'])) {

    header("location: " . $baseUrl);

    exit();

}

require_once "db.info.php";

```

```

if (isset($_POST["quizIndex"])) {

    $sql = "SELECT * FROM rules WHERE quizID = ?;";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo ("error=administratorsDeletionFailed");

        exit();

    }

    mysqli_stmt_bind_param($stmt, "i", $_POST["quizIndex"]);

} else {

    $sql = "SELECT * FROM rules;";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo ("error=stmtFailed");

        exit();

    }

}

mysqli_stmt_execute($stmt);

$resultData = mysqli_stmt_get_result($stmt);

$returnData = array();

while ($row = mysqli_fetch_assoc($resultData)) {

    array_push($returnData, $row);

}

```

```
$jsonData = json_encode($returnData);
```

```
echo $jsonData;
```

```
exit();
```

```
fetchUserData.php
```

```
<?php
```

```
session_start();
```

```
if (!isset($_SESSION['logged']) || !isset($_SESSION['isAdmin'])) {
```

```
    echo "error=unauthorized";
```

```
    exit();
```

```
}
```

```
require_once "db.info.php";
```

```
require_once "functions.php";
```

```
if (isset($_POST["multiple"])) {
```

```
    if ($_POST["multiple"] == "true") {
```

```
        // FETCH DATA FOR ALL THE USERS
```

```
        $sql = "SELECT * FROM users";
```

```
        $stmt = mysqli_stmt_init($conn);
```

```
        if (!mysqli_stmt_prepare($stmt, $sql)) {
```

```
            echo ("error=noUsersFound");
```

```

        exit();
    }

    mysqli_stmt_execute($stmt);

    $resultData = mysqli_stmt_get_result($stmt);

    $responseData = array();

    while ($row = mysqli_fetch_assoc($resultData)) {
        array_push($responseData, json_decode(fetchUserData($conn, $row["userID"])));
    }

    echo json_encode($responseData);

    exit();
}

}

echo fetchUserData($conn, $_POST["user"]);

exit();

flushGrades.php
<?php

session_start();

```

```

if (!isset($_SESSION['logged']) || !isset($_SESSION['isAdmin'])) {

    echo "error=unauthorized";

    exit();

}

require_once "db.info.php";

if (!isset($_POST['submit'])) {

    echo "error=notEnoughVariables";

    exit();

}

$sql = "DELETE FROM grades;";

$stmt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmt, $sql)) {

    echo ('{"error": "stmtFailed"}');

    exit();

}

mysqli_stmt_execute($stmt);

mysqli_stmt_close($stmt);

    functions.php
<?php

function emptyInputRegister($username, $email, $password, $studentName,
$studentLastName, $studentGrade)

{

```



```
    if (empty($username) || empty($email) || empty($password) || empty($studentName) ||  
empty($studentLastName) || empty($studentGrade))
```

```
        $result = true;
```

```
    else
```

```
        $result = false;
```

```
    return $result;
```

```
}
```

```
function invalidUID($username)
```

```
{
```

```
    if (!preg_match("/^[a-zA-Z0-9_]*$/", $username))
```

```
        $result = true;
```

```
    else
```

```
        $result = false;
```

```
    return $result;
```

```
}
```

```
function invalidEmail($email)
```

```
{
```

```
    if (!filter_var($email, FILTER_VALIDATE_EMAIL))
```

```
        $result = true;
```

```
    else
```

```
        $result = false;
```

```
    return $result;
```

```
}
```

```

function uidExists($conn, $uid, $uemail)
{
    $sql = "SELECT * FROM users;";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {
        header("location: ../?error=stmtFailed");

        exit();
    }

    mysqli_stmt_execute($stmt);

    $resultData = mysqli_stmt_get_result($stmt);

    while ($row = mysqli_fetch_assoc($resultData)) {
        if ($uid == decrypt($row["userUsername"]) || $uemail ==
decrypt(($row["userEmail"]))) {
            return $row;
        }
    }

    return false;
}

```

```

function createUser($conn, $email, $username, $password, $studentName,
$studentLastname, $studentGrade)

```

```

{

    $sql = "INSERT INTO users (userEmail, userUsername, userPassword, userFirstName,
userLastName, userEducation) VALUES (?, ?, ?, ?, ?, ?)";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo ("error=stmtFailed");

        exit();

    }


    // HASH PASSWORD

    $hash = password_hash($password, PASSWORD_DEFAULT);

    // ENCRYPT DATA

    $encryptedEmail = encrypt($email);

    $encryptedUsername = encrypt($username);

    $encryptedStudentName = encrypt($studentName);

    $encryptedStudentLastName = encrypt($studentLastname);


    mysqli_stmt_bind_param($stmt, "ssssss", $encryptedEmail, $encryptedUsername,
$hash, $encryptedStudentName, $encryptedStudentLastName, $studentGrade);

    mysqli_stmt_execute($stmt);

    mysqli_stmt_close($stmt);


    if ($row = uidExists($conn, $username, $username)) {

        // USER EXISTS => CHECK PASSWORD

        logUserIn($conn, $password, $row["userPassword"]);
    }
}

```

```

    }

    echo ("error=none");

    exit();
}

function logUserIn($conn, $pwd, $pwdHash)
{
    if (password_verify($pwd, $pwdHash)) {
        session_start();

        $_SESSION['logged'] = time();

        // GET USER ID

        $sql = "SELECT * FROM users;";

        $stmt = mysqli_stmt_init($conn);

        if (!mysqli_stmt_prepare($stmt, $sql)) {
            echo ("error=adminFetchFailed");

            exit();
        }

        mysqli_stmt_execute($stmt);

        $userID = -1;

        $resultData = mysqli_stmt_get_result($stmt);

        while ($row = mysqli_fetch_assoc($resultData)) {

```

```

        if (decrypt($row["userUsername"]) == $_POST['username'] ||
decrypt($row["userEmail"]) == $_POST['username']) {

```

```

        $userID = $row["userID"];

        $_SESSION['username'] = decrypt($row['userUsername']);

        $_SESSION['firstname'] = decrypt($row['userFirstName']);

        $_SESSION['lastname'] = decrypt($row['userLastName']);

    }

}

```

```

if ($userID == -1) {

    echo "error=userNotFound";

    exit();

}

```

```

// CHECK IF USER IS ADMIN

```

```

$sql = "SELECT * FROM administrators WHERE administrators.userID = ?";

```

```

$stmt = mysqli_stmt_init($conn);

```

```

if (!mysqli_stmt_prepare($stmt, $sql)) {

```

```

    echo ("error=adminFetchFailed");

    exit();

}

```

```

mysqli_stmt_bind_param($stmt, "i", $userID);

```

```

mysqli_stmt_execute($stmt);

```

```

$resultData = mysqli_stmt_get_result($stmt);

if ($row = mysqli_fetch_assoc($resultData)) {

    $_SESSION['isAdmin'] = true;

}

} else {

    echo ("error=wrongPassword");

    exit();

}

}

function deleteRule($conn, $ruleID)
{

    // DELETE RULE

    $sqlDeleteRule = "DELETE FROM rules WHERE ruleID = ?;";

    $stmtDeleteRule = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtDeleteRule, $sqlDeleteRule)) {

        echo ("error=stmtFailed");

        exit();

    }

    mysqli_stmt_bind_param($stmtDeleteRule, "i", $ruleID);

    mysqli_stmt_execute($stmtDeleteRule);

    mysqli_stmt_close($stmtDeleteRule);

```

```

}

function deleteQuestion($conn, $questionID)
{
    $sqlGetAnswers = "SELECT * FROM answers WHERE questionID = ?;";
    $stmtGetAnswers = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtGetAnswers, $sqlGetAnswers)) {
        echo ("error=stmtFailed");
        exit();
    }

    mysqli_stmt_bind_param($stmtGetAnswers, "i", $questionID);
    mysqli_stmt_execute($stmtGetAnswers);

    $resultData = mysqli_stmt_get_result($stmtGetAnswers);
    while ($row = mysqli_fetch_assoc($resultData)) {
        deleteAnswer($conn, $row["answerID"]);
    }

    // DELETE ALL QUESTIONS THAT USE THIS RULE
    $sqlDeleteQuestion = "DELETE FROM questions WHERE questionID = ?;";
    $stmtDeleteQuestion = mysqli_stmt_init($conn);

```

```

if (!mysqli_stmt_prepare($stmtDeleteQuestion, $sqlDeleteQuestion)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmtDeleteQuestion, "i", $questionID);

mysqli_stmt_execute($stmtDeleteQuestion);

mysqli_stmt_close($stmtDeleteQuestion);

}

function deleteAnswer($conn, $answerID)
{

    $sqlDeleteAnswers = "DELETE FROM answers WHERE answerID = ?;";

    $stmtDeleteAnswers = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtDeleteAnswers, $sqlDeleteAnswers)) {

        echo ("error=stmtFailed");

        exit();

    }

    mysqli_stmt_bind_param($stmtDeleteAnswers, "i", $answerID);

    mysqli_stmt_execute($stmtDeleteAnswers);

    mysqli_stmt_close($stmtDeleteAnswers);

}

```



```

function deleteEvaluation($conn, $evalID)
{
    // DELETE CONNECTED GRADES

    $sqlGetGrades = "SELECT * FROM grades WHERE evaluationID = ?;";
    $stmtGetGrades = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtGetGrades, $sqlGetGrades)) {
        echo ("error=stmtFailed");
        exit();
    }

    mysqli_stmt_bind_param($stmtGetGrades, "i", $evalID);
    mysqli_stmt_execute($stmtGetGrades);

    $resultData = mysqli_stmt_get_result($stmtGetGrades);
    while ($row = mysqli_fetch_assoc($resultData)) {
        deleteGrade($conn, $row["gradeID"]);
    }

    mysqli_stmt_close($stmtGetGrades);

    // DELETE EVALUATION

    $sqlDeleteEval = "DELETE FROM evaluations WHERE evaluationID = ?;";

```

```

$stmtDeleteEval = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtDeleteEval, $sqlDeleteEval)) {
    echo ("error=deleteEvaluationsFailed");
    exit();
}

mysqli_stmt_bind_param($stmtDeleteEval, "i", $evalID);
mysqli_stmt_execute($stmtDeleteEval);
mysqli_stmt_close($stmtDeleteEval);
}

function deleteGrade($conn, $gradeID)
{
    $sqlDeleteGrades = "DELETE FROM grades WHERE gradeID = ?";
    $stmtDeleteGrades = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmtDeleteGrades, $sqlDeleteGrades)) {
        echo ("error=deleteGradesFailed");
        exit();
    }

    mysqli_stmt_bind_param($stmtDeleteGrades, "i", $gradeID);
    mysqli_stmt_execute($stmtDeleteGrades);
    mysqli_stmt_close($stmtDeleteGrades);
}

function deleteEvaluationsForUser($conn, $userID)

```

```

{
    $sqlUserEval = "SELECT * FROM evaluations WHERE userID = ?;";
    $stmtUserEval = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmtUserEval, $sqlUserEval)) {
        echo ("error=evaluationsDeletionFailed");
        exit();
    }

    mysqli_stmt_bind_param($stmtUserEval, "i", $userID);
    mysqli_stmt_execute($stmtUserEval);

    $evaluations = mysqli_stmt_get_result($stmtUserEval);
    while ($rowEval = mysqli_fetch_assoc($evaluations)) {
        deleteEvaluation($conn, $rowEval["evaluationID"]);
    }
}

```

```

function deleteEvaluationsForQuiz($conn, $quizID)
{
    $sqlUserEval = "SELECT * FROM evaluations WHERE quizID = ?;";
    $stmtUserEval = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmtUserEval, $sqlUserEval)) {
        echo ("error=evaluationsDeletionFailed");
        exit();
    }
}

```

```

mysqli_stmt_bind_param($stmtUserEval, "i", $quizID);

mysqli_stmt_execute($stmtUserEval);


$evaluations = mysqli_stmt_get_result($stmtUserEval);

while ($rowEval = mysqli_fetch_assoc($evaluations)) {

    deleteEvaluation($conn, $rowEval["evaluationID"]);

}

}

$alphabet = " !\"#$%&()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~";


function randomChars($num)

{

    global $alphabet;

    $i = 0;

    $arr = "";

    for ($i = 0; $i < $num; $i++) {

        $index = rand(0, strlen($alphabet) - 1);

        $arr[$i] = $alphabet[$index];

    }

    return $arr;

}

$key = "OrthopaigniaSecureKeyForEncryptionDecryption";

```

```

function encrypt($message)
{
    global $key;

    $encryptionKey = base64_decode($key);

    $iv = openssl_random_pseudo_bytes(openssl_cipher_iv_length('aes-256-cbc'));

    $finalMessage = openssl_encrypt($message, 'aes-256-cbc', $encryptionKey, 0, $iv);

    return base64_encode($finalMessage . "::" . $iv);
}

```

```

function decrypt($message)
{
    global $key;

    $encryptionKey = base64_decode($key);

    list($encryptedData, $iv) = array_pad(explode("::", base64_decode($message), 2), 2,
    null);

    return openssl_decrypt($encryptedData, 'aes-256-cbc', $encryptionKey, 0, $iv);
}

```

```

function getUserID($conn, $credential)
{
    $sql = "SELECT * FROM users;";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo ("error=adminFetchFailed");
    }
}

```

```

        exit();
    }

    mysqli_stmt_execute($stmt);

    $resultData = mysqli_stmt_get_result($stmt);

    while ($row = mysqli_fetch_assoc($resultData)) {

        if ($credential == decrypt($row["userUsername"]) || $credential ==
decrypt($row["userEmail"])) {

            return $row["userID"];

        }

    }

    return false;

    mysqli_stmt_close($stmt);
}

function getQuestionByText($conn, $text)
{
    $sqlQuestionID = "SELECT * FROM questions WHERE questionText = ?;";

    $stmtQuestionID = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtQuestionID, $sqlQuestionID)) {

        echo '{"error": "stmtFailed"}';

        exit();

    }

    mysqli_stmt_bind_param($stmtQuestionID, "s", $text);

```

```

mysqli_stmt_execute($stmtQuestionID);

$resultData = mysqli_stmt_get_result($stmtQuestionID);
if ($row = mysqli_fetch_assoc($resultData)) {
    return $row;
} else {
    return '{"error":"questionIDNotFound"}';
    exit();
}
mysqli_stmt_close($stmtQuestionID);
}

```

```

function convertEducationToReadable($level)
{
    switch ($level) {
        case '3':
            return "Γ' Δημοτικού";
            break;
        case '4':
            return "Δ' Δημοτικού";
            break;
        case '5':
            return "Ε' Δημοτικού";
            break;
    }
}

```

```

case '6':

    return "ΣΤ' Δημοτικού";

    break;

case 'other':

    return "Δευτεροβάθμια";

    break;


default:

    return "error";

    break;

}

}

function checkForeignKey($conn, $check)

{

    if ($check) {

        # code...

        $sql = "SET FOREIGN_KEY_CHECKS = 1;";

    } else {

        $sql = "SET FOREIGN_KEY_CHECKS = 0;";

    }


    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

```



```

        echo ("error=stmtFailed");

        exit();
    }

    mysqli_stmt_execute($stmt);

    mysqli_stmt_close($stmt);
}

function fetchUserData($conn, $userID)
{
    // GET USER DATA

    $sql = "SELECT * FROM users WHERE userID = ?;";

    $stmt = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmt, $sql)) {

        echo ("error=noUsersFound");

        exit();
    }

    mysqli_stmt_bind_param($stmt, "i", $userID);

    mysqli_stmt_execute($stmt);

    $resultData = mysqli_stmt_get_result($stmt);

    $responseData = array();

```

```

if ($row = mysqli_fetch_assoc($resultData)) {

    $row["userFirstName"] = decrypt($row["userFirstName"]);

    $row["userLastName"] = decrypt($row["userLastName"]);

    $row["userUsername"] = decrypt($row["userUsername"]);

    $row["userEmail"] = decrypt($row["userEmail"]);


    $noPass = array();

    foreach ($row as $key => $value) {

        if ($key == "userPassword") {

            continue;

        }

        $noPass[$key] = $value;

    }


    $responseData["user"] = $noPass;


    $sqlEvals = "SELECT * FROM evaluations WHERE userID = ?;";

    $stmtEvals = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtEvals, $sqlEvals)) {

        echo ("error=noUsersFound");

        exit();

    }

    mysqli_stmt_bind_param($stmtEvals, "i", $userID);

    mysqli_stmt_execute($stmtEvals);

```

```

$resultDataEvals = mysqli_stmt_get_result($stmtEvals);

$evaluations = array();

// FETCH CONNECTED EVALUATIONS

while ($rowEvals = mysqli_fetch_assoc($resultDataEvals)) {

    $evaluation = array();

    $evaluation["evaluation"] = $rowEvals;

    $grades = array();

    //   FOR EACH EVALUATION FETCH GARDES

    $sqlGrades = "SELECT * FROM grades WHERE gradeID = ? OR gradeID = ?;";

    $stmtGrades = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtGrades, $sqlGrades)) {

        echo ("error=noUsersFound");

        exit();

    }

    mysqli_stmt_bind_param($stmtGrades, "ii", $rowEvals['firstAttempt'],
$rowEvals['latestAttempt']);

    mysqli_stmt_execute($stmtGrades);

    $resultDataGrades = mysqli_stmt_get_result($stmtGrades);

    while ($rowGrades = mysqli_fetch_assoc($resultDataGrades)) {

        array_push($grades, $rowGrades);

    }
}

```

```

        $evaluation["grades"] = $grades;

        //    PUSH EVALUATIONS TO A NEW ARRAY CALLED EVALUATIONS

        array_push($evaluations, $evaluation);
    }

    $responseData["user"]["evaluations"] = $evaluations;

    return json_encode($responseData);
} else {
    return "error=noUsersFound";
}
}

function debugEcho($reset = false)
{
    if ($reset) {
        $_SESSION["debugLine"] = 1;
    }

    echo "CHECKPOINT " . $_SESSION["debugLine"] . "\n";

    $_SESSION["debugLine"]++;
}

function getEvaluations($conn, $userID, $quizID)
{

```

```
$sqlGetEvaluations = "SELECT * FROM evaluations WHERE userID = ? AND  
quizID = ?";
```

```
$stmtGetEvaluations = mysqli_stmt_init($conn);
```

```
if (!mysqli_stmt_prepare($stmtGetEvaluations, $sqlGetEvaluations)) {
```

```
    echo '{"error": "stmtFailed"}';
```

```
    exit();
```

```
}
```

```
mysqli_stmt_bind_param($stmtGetEvaluations, "ii", $userID, $quizID);
```

```
mysqli_stmt_execute($stmtGetEvaluations);
```

```
$resultData = mysqli_stmt_get_result($stmtGetEvaluations);
```

```
if ($rowEval = mysqli_fetch_assoc($resultData)) {
```

```
    return $rowEval;
```

```
}
```

```
return false;
```

```
}
```

```
loginHandler.php
```

```
<?php
```

```
require_once "db.info.php";
```

```
require_once "functions.php";
```

```
if (isset($_POST['submit'])) {
```

```

// CHECK IF USER EXISTS

if ($row = uidExists($conn, $_POST["username"], $_POST["username"])) {

    // USER EXISTS => CHECK PASSWORD

    logUserIn($conn, $_POST["password"], $row["userPassword"]);

    echo "error=none";

} else {

    echo "error=userDoesNotExist";

}

} else {

    header("location: ../");

}

    logoutHandler.php
<?php

session_start();

session_unset();

session_destroy();

exit();

    newQuestionHandler.php
<?php

if (isset($_POST['submit'])) {

    require_once "db.info.php";

    require_once "functions.php";

```

```

$questionType = $_POST["question-type"];

$questionText = $_POST["question"];

$rightAnswer = $_POST["right-answer"];

$quizID = $_POST["quizID"];


$questionID;

$answerID;


if (!($questionText != "" && $rightAnswer != "" && $quizID != "")) {

    echo "error=emptyInputs";

    exit();

}


// -----

// CREATE QUESTION

// GET QUESTION ID

$sqlCreateQuestion = "INSERT INTO questions(questionText, quizID, questionType)
VALUES (?, ?, ?);";

$stmtCreateQuestion = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtCreateQuestion, $sqlCreateQuestion)) {

    echo ("error=stmtFailed");

    exit();

}

```

```

    mysqli_stmt_bind_param($stmtCreateQuestion, "sis", $questionText, $quizID,
$questionType);

    mysqli_stmt_execute($stmtCreateQuestion);

    mysqli_stmt_close($stmtCreateQuestion);


$sqlGetQuestionID = "SELECT * FROM questions WHERE questionText = ?;";

$stmtGetQuestionID = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtGetQuestionID, $sqlGetQuestionID)) {

    echo ("error=stmtFailed");

    exit();

}


mysqli_stmt_bind_param($stmtGetQuestionID, "s", $questionText);

mysqli_stmt_execute($stmtGetQuestionID);


$resultData = mysqli_stmt_get_result($stmtGetQuestionID);

if ($row = mysqli_fetch_assoc($resultData)) {

    $questionID = $row["questionID"];

} else {

    echo "error=questionNotCreatedOrNoID";

    mysqli_stmt_close($stmtGetQuestionID);

    exit();

}

```



```

// -----

// CREATE RIGHT ANSWER

// GET THE ID OF THE RIGHT ANSWER

$sqlCreateRightAnswer = "INSERT INTO answers(answerText, questionID)
VALUES (?, ?)";

$stmtCreateRightAnswer = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtCreateRightAnswer, $sqlCreateRightAnswer)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmtCreateRightAnswer, "si", $rightAnswer, $questionID);

mysqli_stmt_execute($stmtCreateRightAnswer);

mysqli_stmt_close($stmtCreateRightAnswer);


$sqlGetRightAnswerID = "SELECT answerID FROM answers WHERE answerText =
?";

$stmtGetRightAnswerID = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtGetRightAnswerID, $sqlGetRightAnswerID)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmtGetRightAnswerID, "s", $rightAnswer);

mysqli_stmt_execute($stmtGetRightAnswerID);

```

```

$resultAnswerID = mysqli_stmt_get_result($stmtGetRightAnswerID);

if ($row2 = mysqli_fetch_assoc($resultAnswerID)) {

    $answerID = $row2["answerID"];

} else {

    echo "error=rightAnswerNotCreatedOrNoID";

    mysqli_stmt_close($stmtGetRightAnswerID);

    exit();

}

// -----

// CREATE WRONG ANSWERS

$sqlCreateWrongAnswers = "INSERT INTO answers(answerText, questionID)
VALUES";

$index = 0;

$bindValues = [];

$bindTypes = "";

foreach ($_POST as $postVarKey => $postVarVal) {

    if (str_contains($postVarKey, "wrong-answer")) {

        if ($index != 0) {

            $sqlCreateWrongAnswers .= ",";

        }

        $sqlCreateWrongAnswers .= "(?, ?)";

        $index++;

        array_push($bindValues, $postVarVal, $questionID);
    }
}

```

```

        $bindTypes .= "si";
    }
}

$sqlCreateWrongAnswers .= ";;

if ($index == 0) {

    // THERE ARE NO WRONG ANSWERS

    echo "error=noWrongAnswers";

    exit();
}

$stmtCreateWrongAnswers = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtCreateWrongAnswers, $sqlCreateWrongAnswers)) {

    echo ("error=stmtFailed");

    exit();
}

mysqli_stmt_bind_param($stmtCreateWrongAnswers, $bindTypes, ...$bindValues);

mysqli_stmt_execute($stmtCreateWrongAnswers);

mysqli_stmt_close($stmtCreateWrongAnswers);

// -----

// UPDATE QUESTION TO PROVIDE THE RIGHT ANSWER ID

checkForeignKey($conn, false);

```

```
$sqlUpdateQuestion = "UPDATE questions SET rightAnswer = ? WHERE questionID  
= ?;";
```

```
$stmtUpdateQuestion = mysqli_stmt_init($conn);
```

```
if (!mysqli_stmt_prepare($stmtUpdateQuestion, $sqlUpdateQuestion)) {
```

```
    echo ("error=stmtFailed");
```

```
    exit();
```

```
}
```

```
mysqli_stmt_bind_param($stmtUpdateQuestion, "ii", $answerID, $questionID);
```

```
mysqli_stmt_execute($stmtUpdateQuestion);
```

```
mysqli_stmt_close($stmtUpdateQuestion);
```

```
checkForeignKey($conn, true);
```

```
echo "error=none";
```

```
} else {
```

```
    echo ("error=accessDenied");
```

```
    exit();
```

```
}
```

```
    newQuizHandler.php
```

```
<?php
```

```
if (isset($_POST['submit'])) {
```

```
    $quizText = $_POST["quiz-text"];
```

```

require_once "db.info.php";

require_once "functions.php";


// CHECK IF QUIZ ALREADY EXISTS


$sql = "SELECT * FROM quizzes WHERE quizTitle = ?;";

$stmt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmt, $sql)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmt, "s", $quizText);

mysqli_stmt_execute($stmt);


$resultData = mysqli_stmt_get_result($stmt);

if ($row = mysqli_fetch_assoc($resultData)) {

    echo "error=quizAlreadyExists";

    mysqli_stmt_close($stmt);

    exit();

} else {

    mysqli_stmt_close($stmt);


    $sql = "INSERT INTO quizzes(quizTitle) VALUES (?);";

    $stmt = mysqli_stmt_init($conn);

```

```

        if (!mysqli_stmt_prepare($stmt, $sql)) {

            echo ("error=stmtFailed");

            exit();

        }

        mysqli_stmt_bind_param($stmt, "s", $quizText);

        mysqli_stmt_execute($stmt);

        mysqli_stmt_close($stmt);

    }

    echo "error=none";

} else {

    echo ("error=accessDenied");

    exit();

}

newRuleHandler.php
<?php

```

```

if (isset($_POST['submit'])) {

    $quizID = intval($_POST["rule-quiz"], $base = 10);

    if ($quizID == null) {

        echo "error=quizNotProvided";

        exit();

    }

}

```

```

require_once "db.info.php";

require_once "functions.php";


// CHECK IF RULE ALREADY EXISTS

$sql = "SELECT * FROM rules WHERE ruleName = ?;";

$stmt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmt, $sql)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmt, "s", $_POST["rule-name"]);

mysqli_stmt_execute($stmt);


$resultData = mysqli_stmt_get_result($stmt);

if ($row = mysqli_fetch_assoc($resultData)) {

    echo "error=ruleAlreadyExists";

    mysqli_stmt_close($stmt);

    exit();

} else {


// CHECK IF RULE'S QUIZ EXISTS

mysqli_stmt_close($stmt);

$sqlCheckQuiz = "SELECT * FROM quizzes WHERE quizID = ?;";

$stmtCheckQuiz = mysqli_stmt_init($conn);

```

```

if (!mysqli_stmt_prepare($stmtCheckQuiz, $sqlCheckQuiz)) {

    echo ("error=stmtFailed");

    exit();

}

mysqli_stmt_bind_param($stmtCheckQuiz, "i", $quizID);

mysqli_stmt_execute($stmtCheckQuiz);

$resultDataCheckQuiz = mysqli_stmt_get_result($stmtCheckQuiz);

if ($row2 = mysqli_fetch_assoc($resultDataCheckQuiz)) {

    mysqli_stmt_close($stmtCheckQuiz);

    $sqlInsert = "INSERT INTO rules(ruleName, ruleText, quizID) VALUES (?, ?,
?);";

    $stmtInsert = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtInsert, $sqlInsert)) {

        echo ("error=stmtFailed");

        exit();

    }

    mysqli_stmt_bind_param($stmtInsert, "ssi", $_POST["rule-name"],
$_POST["rule-text"], $quizID);

    mysqli_stmt_execute($stmtInsert);

    echo "error=none";

    mysqli_stmt_close($stmtInsert);

```



```

        exit();

    } else {

        echo "error=quizNotFound";

        exit();

    }

}

} else {

    echo ("error=accessDenied");

    exit();

}

```

registerHandler.php

<?php

```

if (isset($_POST['submit'])) {

    // USER DATA

    $username = $_POST["username"];

    $email = $_POST["email"];

    $password = $_POST["password"];

    // STUDENT DATA

    $studentName = $_POST["student-name"];

    $studentLastName = $_POST["student-lastname"];

    $studentGrade = $_POST["student-grade"];

    require_once "db.info.php";

    require_once "functions.php";

```

```

if (emptyInputRegister($username, $email, $password, $studentName,
$studentLastName, $studentGrade) !== false) {

    // THERE ARE EMPTY INPUTS

    echo ("error=emptyInput");

    exit();

}

if (invalidUID($username) !== false) {

    // THERE ARE EMPTY INPUTS

    echo ("error=invalidUsername");

    exit();

}

if (invalidEmail($email) !== false) {

    // THERE ARE EMPTY INPUTS

    echo ("error=invalidEmail");

    exit();

}

if (uidExists($conn, $username, $email) !== false) {

    // THERE ARE EMPTY INPUTS

    echo ("error=userExists");

    exit();

}

createUser($conn, $email, $username, $password, $studentName, $studentLastName,
$studentGrade);

```

```

        echo "error=none";
    } else {
        echo ("error=accessDenied");
        exit();
    }

    setGrades.php
<?php

session_start();

if (!isset($_SESSION["logged"])) {
    echo '{"error": "notLoggedIn"}';
    exit();
}

if (!isset($_POST["submit"])) {
    echo '{"error": "unauthorized"}';
    exit();
}

require_once "db.info.php";
require_once "functions.php";

$relevantAttemptsNumber = 5;

```

```

$results = json_decode($_POST["results"]);

$quizID = $_POST["quizID"];

if (isset($_POST["studyTime"])) {

    $studyTime = floatval($_POST["studyTime"]) / 10;

} else {

    $studyTime = 0;

}

$userID = getUserID($conn, $_SESSION["username"]);

if (!$userID) {

    echo '{"error": "userNotFound"}';

    exit();

}

$rightAnswersCount = 0;

for ($i = 0; $i < sizeof($results); $i++) {

    // FOREACH QUESTION ANSWERED, GET ID, UPDATE ITS GRADE,
    RELEVANT_GRADE, ATTEMPTS_PER_QUESTION, RELEVANT_ATTEMPTS

    if ($results[$i][1] == 1) {

        $rightAnswersCount++;

    }

}

```

```

}

$evalExists = false;

$evaluationID = "";

if ($eval = getEvaluations($conn, $userID, $quizID)) {

    // We have an evaluation

    $evalExists = true;

    $evaluationID = $eval["evaluationID"];

} else {

    $evalExists = false;

    // Create evaluation

    $sqlCreateEval = "INSERT INTO evaluations(userID, quizID) VALUES (?, ?)";

    $stmtCreateEval = mysqli_stmt_init($conn);

    if (!mysqli_stmt_prepare($stmtCreateEval, $sqlCreateEval)) {

        echo '{"error": "stmtFailed"}';

        exit();

    }

    mysqli_stmt_bind_param($stmtCreateEval, "ii", $userID, $quizID);

    mysqli_stmt_execute($stmtCreateEval);

    mysqli_stmt_close($stmtCreateEval);

    // Get the id of the evaluation

    if ($createdEval = getEvaluations($conn, $userID, $quizID)) {

```

```

        $evaluationID = $createdEval["evaluationID"];
    } else {
        echo '{"error": "failedToCreateEval"}';
        exit();
    }
}

// Create grade

$successRatio = $rightAnswersCount / sizeof($results);

$currentDate = date("D M j G:i:s T Y");

$totalTime = floatval($_POST["totalTime"]) / 10;

$sqlCreateGrade = "INSERT INTO grades(answerTime, successRatio, evaluationID,
gradeDate, studyTime) VALUES (?, ?, ?, ?, ?)";

$stmtCreateGrade = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmtCreateGrade, $sqlCreateGrade)) {
    echo '{"error": "stmtFailed"}';
    exit();
}

mysqli_stmt_bind_param($stmtCreateGrade, "ddisd", $totalTime, $successRatio,
$evaluationID, $currentDate, $studyTime);

mysqli_stmt_execute($stmtCreateGrade);

mysqli_stmt_close($stmtCreateGrade);

```

```
$gradeID = "";
```

```
$sqlGetGrade = "SELECT * FROM grades WHERE evaluationID = ? AND gradeDate =  
?";
```

```
$stmtGetGrade = mysqli_stmt_init($conn);
```

```
if (!mysqli_stmt_prepare($stmtGetGrade, $sqlGetGrade)) {
```

```
    echo '{"error": "stmtFailed"}';
```

```
    exit();
```

```
}
```

```
mysqli_stmt_bind_param($stmtGetGrade, "is", $evaluationID, $currentDate);
```

```
mysqli_stmt_execute($stmtGetGrade);
```

```
$resultGetGrade = mysqli_stmt_get_result($stmtGetGrade);
```

```
if ($rowGrade = mysqli_fetch_assoc($resultGetGrade)) {
```

```
    $gradeID = $rowGrade["gradeID"];
```

```
} else {
```

```
    echo '{"error": "failedToCreateGrade"}';
```

```
    exit();
```

```
}
```

```
mysqli_stmt_close($stmtGetGrade);
```

```
if (!$evalExists) {
```

```

// Update firstAttempt too

$sqlUpdateFirstAttempt = "UPDATE evaluations SET firstAttempt = ? WHERE
evaluationID = ?;";

$stmtUpdateFirstAttempt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtUpdateFirstAttempt, $sqlUpdateFirstAttempt)) {

    echo '{"error": "stmtFailed"}';

    exit();

}

mysqli_stmt_bind_param($stmtUpdateFirstAttempt, "ii", $gradeID, $evaluationID);

mysqli_stmt_execute($stmtUpdateFirstAttempt);

mysqli_stmt_close($stmtUpdateFirstAttempt);

}

// Update latestAttempt

$sqlUpdateLatestAttempt = "UPDATE evaluations SET latestAttempt = ? WHERE
evaluationID = ?;";

$stmtUpdateLatestAttempt = mysqli_stmt_init($conn);

if (!mysqli_stmt_prepare($stmtUpdateLatestAttempt, $sqlUpdateLatestAttempt)) {

    echo '{"error": "stmtFailed"}';

    exit();

}

mysqli_stmt_bind_param($stmtUpdateLatestAttempt, "ii", $gradeID, $evaluationID);

```



```
mysqli_stmt_execute($stmtUpdateLatestAttempt);
```

```
mysqli_stmt_close($stmtUpdateLatestAttempt);
```

```
echo '{"error":"none"}';
```

```
    updateUserInfo.php
```

```
<?php
```

```
session_start();
```

```
if (!isset($_SESSION['logged'])) {
```

```
    header("location: " . $baseUrl);
```

```
    exit();
```

```
}
```

```
if (isset($_POST['submit'])) {
```

```
    require_once "db.info.php";
```

```
    $firstName = $_POST["firstname"];
```

```
    $lastName = $_POST["lastname"];
```

```
    $sql = "UPDATE users SET userFirstName = ?, userLastName = ? WHERE  
userUsername = ?";
```

```
    $stmt = mysqli_stmt_init($conn);
```

```
    if (!mysqli_stmt_prepare($stmt, $sql)) {
```

```

        echo ("error=stmtFailed");

        exit();

    }

    mysqli_stmt_bind_param($stmt, "sss", $firstName, $lastName,
$_SESSION["username"]);

    mysqli_stmt_execute($stmt);

    mysqli_stmt_close($stmt);

    $_SESSION["firstname"] = $firstName;

    $_SESSION["lastname"] = $lastName;

    echo ("error=none");
} else {

    echo ("error=accessDenied");

    exit();

}

    updateUserPassword.php
<?php

session_start();

if (!isset($_SESSION['logged'])) {

    header("location: " . $baseUrl);

```

```

        exit();
    }

    if (isset($_POST['submit'])) {

        require_once "db.info.php";

        require_once "functions.php";

        $oldPassword = $_POST['old-password'];

        $newPassword = $_POST['new-password'];

        $userID = getUserID($conn, $_SESSION["username"]);

        // CHECK IF USERS HAS TYPED THE CORRECT PREVIOUS PASSWORD

        $sql = "SELECT * FROM users WHERE userID = ?;";

        $stmt = mysqli_stmt_init($conn);

        if (!mysqli_stmt_prepare($stmt, $sql)) {

            echo ("error=stmtFailed");

            exit();

        }

        mysqli_stmt_bind_param($stmt, "i", $userID);

        mysqli_stmt_execute($stmt);

        $resultData = mysqli_stmt_get_result($stmt);
    }

```

```

if ($row = mysqli_fetch_assoc($resultData)) {

    if (password_verify($oldPassword, $row["userPassword"])) {

        $hash = password_hash($newPassword, PASSWORD_DEFAULT);

        $updateSQL = "UPDATE users SET userPassword = ? WHERE userID = ?";

        $stmtUpdate = mysqli_stmt_init($conn);

        if (!mysqli_stmt_prepare($stmtUpdate, $updateSQL)) {

            echo ("error=stmtFailed");

            exit();

        }

        mysqli_stmt_bind_param($stmtUpdate, "si", $hash, $row["userID"]);

        mysqli_stmt_execute($stmtUpdate);

        mysqli_stmt_close($stmtUpdate);

        echo ("error=none");

    } else {

        echo ("error=wrongPassword");

        exit();

    }

} else {

    echo ("error=userNotFound");

```

```

        exit();
    }

    mysqli_stmt_close($stmt);
} else {

    echo ("error=accessDenied");

    exit();
}

```

Φάκελος js

dragDropGame.js

```

let empty;

let answers;

let answersContainer;

let currentAnswer;

let previousAnswer;


function initializeElements() {

    setTimeout(() => {

        answersContainer = document.querySelector(".quiz-answers");

        answers = document.querySelectorAll("div[draggable=true]");

        empty = document.querySelector(".empty");


        previousAnswer = answers[0];

        currentAnswer = answers[0];
    });
}

```

```

for (const answer of answers) {

  answer.addEventListener("dragstart", dragStart);

  answer.addEventListener("dragend", dragEnd);

}

empty.addEventListener("dragover", dragOver);

empty.addEventListener("dragenter", dragEnter);

empty.addEventListener("dragleave", dragLeave);

empty.addEventListener("drop", dragDrop);

}, 1000);

}

// Drag Functions

function dragStart(event) {

  this.classList.add("hold");

  setTimeout(() => {

    this.classList.add("invisible");

  }, 0);

  currentAnswer = event.target;

  document.body.style.cursor = "grabbing";

}

function dragEnd(event) {

  this.classList.remove("hold", "invisible");

  document.body.style.cursor = "auto";

```

```

}

function dragOver(event) {
    event.preventDefault();
}

function dragEnter(event) {
    event.preventDefault();
    empty.classList.add("hovered");
}

function dragLeave(event) {
    empty.classList.remove("hovered");
    empty.innerHTML = "";
    answersContainer.appendChild(currentAnswer);
}

function dragDrop(event) {
    empty.classList.remove("hovered");
    if (empty !== previousAnswer && previousAnswer !== null) {
        answersContainer.appendChild(previousAnswer);
    }
    empty.appendChild(currentAnswer);
    previousAnswer = currentAnswer;
}

fetchAdminData.js
baseUrl = window.location.pathname.split("/")[1].replace(/(?:\r\n|\r|\n)/g, "");

```

```
fetch(`${baseUrl}/includes/fetchAdminData.php`)
```

```
.then((res) => {
```

```
    return res.text();
```

```
})
```

```
.then((text) => {})
```

```
.catch((err) => {});
```

formHandler.js

```
baseUrl = window.location.pathname.split("/")[1].replace(/(?:\r\n|\r|\n)/g, "");
```

```
function submitLogin() {
```

```
    const formData = new FormData(
```

```
        document.querySelector(".form-inputs.login").getElementsByName("form")[0]
```

```
    );
```

```
    const searchParams = new URLSearchParams();
```

```
    for (const pair of formData) {
```

```
        if (pair[0] == "" || pair[0] == null || pair[1] == "" || pair[0] == null) {
```

```
            sweetAlertWarning({
```

```
                title: "Προσοχή!",
```

```
                text: "Κάποια πεδία είναι κενά!",
```

```
                confirmText: "Εντάξει",
```

```
            });
```

```
            return;
```

```
        }
```



```

searchParams.append(pair[0], pair[1]);
}

searchParams.append("submit", "submit");

fetch(`${baseUrl}/includes/loginHandler.php`, {
  method: "POST",
  body: searchParams,
})
.then(function (response) {
  return response.text();
})
.then(function (text) {
  let error = text.split("=")[1];
  switch (error) {
    case "none":
      window.location = `${baseUrl}`;
      break;
    case "userDoesNotExist":
      sweetAlertError({
        text: "Δεν υπάρχει χρήστης με αυτό το όνομα / email.",
      });
      break;
    case "wrongPassword":

```

```

    sweetAlertError({
      text: "Ο κωδικός που δώσατε είναι λάθος!",
    });

    break;

    default:

      break;

  }

})

.catch(function (error) {

  console.log(error);

});

}

```

```

function submitRegister() {

  const formData = new FormData(

    document

    .querySelector(".form-inputs.register")

    .getElementsByTagName("form")[0]

  );

  const searchParams = new URLSearchParams();

  for (const pair of formData) {

    if (pair[0] == "" || pair[0] == null || pair[1] == "" || pair[0] == null) {

```

```

sweetAlertWarning({
    title: "Προσοχή!",
    text: "Κάποια πεδία είναι κενά!",
    confirmText: "Εντάξει",
});

return;
}

searchParams.append(pair[0], pair[1]);
}

if (searchParams.get("password") !== searchParams.get("repeat-password")) {
    sweetAlertWarning({
        title: "Προσοχή!",
        text: "Οι κωδικοί δεν είναι ίδιοι. Ελέγξτε ότι επαναλάβατε τον κωδικό σας σωστά!",
        confirmText: "Εντάξει",
    });

    return;
}

searchParams.append("submit", "submit");

fetch(`${baseUrl}/includes/registerHandler.php`, {
    method: "POST",
    body: searchParams,

```

```

}))

.then(function (response) {

    return response.text();

}))

.then(function (text) {

    let error = text.split("=")[1];

    switch (error) {

        case "none":

            window.location = `/${baseUrl}`;

            break;

        case "invalidEmail":

            sweetAlertError({ text: "Αυτό το email δεν είναι αποδεκτό!" });

            break;

        case "emptyInput":

            sweetAlertError({ text: "Κάποιο πεδίο είναι κενό!" });

            break;

        case "invalidUsername":

            sweetAlertError({ text: "Αυτό το όνομα χρήστη δεν είναι αποδεκτό!" });

            break;

        case "userExists":

            sweetAlertWarning({

                text: "Αυτό το όνομα χρήστη ή το email δεν είναι διαθέσιμο!",

            });

```

```

        break;

default:

    sweetAlertError({ text: `Υπήρξε ένα ασυνήθιστο λάθος: ${error}` });

        break;

    }

    console.log(`Response Text: ${text}`);

})

.catch(function (error) {

    console.log(error);

});

}

    global.js

baseUrl = window.location.pathname.split("/")[1].replace(/(?:\r\n|\r|\n)/g, "");

function logoutHandler() {

    console.log(sessionStorage);

    console.log("CLEARING SESSION");


    localStorage.clear();

    window.sessionStorage.clear();


    fetch(`${baseUrl}/includes/logoutHandler.php`, {

        method: "GET",

    })

```

```

.then(function () {
    window.location = `${baseUrl}`;
})

.catch(function (error) {
    console.log(error);
});
}

function replaceSpecialCharacters(str) {
    return str.replaceAll('"', "").replaceAll("\n", "");
}

function convertEducationToReadable(level) {
    switch (level) {
        case "3":
            return "Γ' Δημοτικού";
            break;
        case "4":
            return "Δ' Δημοτικού";
            break;
        case "5":
            return "Ε' Δημοτικού";
            break;
        case "6":

```

```

        return "ΣΤ' Δημοτικού";

        break;

    case "other":

        return "Δευτεροβάθμια";

        break;

    default:

        return "error";

        break;

    }

}

```

```

    sweetAlertFunctions.js
function sweetAlertWarning({

    title = "Σίγουρα;",

    text = "Η διαδικασία είναι μη αναστρέψιμη!",

    confirmText = "Ναι, σίγουρα",

    cancelText = "Ακύρωση",

    redirect = "",

} = {}) {

    Swal.fire({

        title: title,

        text: text,

        icon: "warning",

        confirmButtonColor: "#3085d6",

        cancelButtonColor: "#d33",

        confirmButtonText: confirmText,

```

```

}).then(() => {

  if (redirect !== "") {

    window.location = redirect;

  }

});

}

```

```

function sweetAlertError({

  title = "Κάτι πήγε στραβά!",

  text = "Συνέβη ένα άγνωστο σφάλμα!",

  confirmText = "Εντάξει",

  cancelText = "Ακύρωση",

  redirect = "",

} = {}) {

  Swal.fire({

    title: title,

    text: text,

    icon: "error",

    confirmButtonColor: "#3085d6",

    cancelButtonColor: "#d33",

    confirmButtonText: confirmText,

  }).then(() => {

    if (redirect !== "") {

      window.location = redirect;

```



```

    }

    });

}

function sweetAlertSuccess({

    title = "Επιτυχία",

    text = "Η ενέργεια ολοκληρώθηκε με επιτυχία!",

    confirmText = "Εντάξει",

    cancelText = "Ακύρωση",

    redirect = "",

} = {}) {

    Swal.fire({

        title: title,

        text: text,

        icon: "success",

        confirmButtonColor: "#3085d6",

        cancelButtonColor: "#d33",

        confirmButtonText: confirmText,

    }).then(() => {

        if (redirect != "") {

            window.location = redirect;

        }

    });

}

```

```

    textFormat.js
function filterText(text) {

    let filteredText;

    // SPECIAL TEXT 1

    filteredText = text;

    filteredText = filteredText.replaceAll(

        "==>",

        "<ion-icon name='arrow-forward'></ion-icon>"

    );

    filteredText = filteredText.replaceAll("/!!", "</span>");
    filteredText = filteredText.replaceAll("/**", "</span>");
    filteredText = filteredText.replaceAll("/$", "</span>");
    filteredText = filteredText.replaceAll("/#", "</span>");
    filteredText = filteredText.replaceAll("/%", "</span>");
    filteredText = filteredText.replaceAll("/^", "</span>");
    filteredText = filteredText.replaceAll("/&", "</span>");
    filteredText = filteredText.replaceAll("/@", "</span>");
    filteredText = filteredText.replaceAll("/?", "</span>");
    filteredText = filteredText.replaceAll("/]", "</span>");


    filteredText = filteredText.replaceAll("___", "<div class='empty'></div>");


    filteredText = filteredText.replaceAll("/!!", "<span class='special-text-1'>");
    filteredText = filteredText.replaceAll("/**", "<span class='special-text-2'>");
    filteredText = filteredText.replaceAll("/$", "<span class='special-text-3'>");

```

```

filteredText = filteredText.replaceAll("#", "<span class='special-text-4'>");
filteredText = filteredText.replaceAll("%", "<span class='special-text-5'>");
filteredText = filteredText.replaceAll("^", "<span class='special-text-6'>");
filteredText = filteredText.replaceAll("&", "<span class='special-text-7'>");
filteredText = filteredText.replaceAll("@", "<span class='special-text-8'>");
filteredText = filteredText.replaceAll("?", "<span class='special-text-9'>");
filteredText = filteredText.replaceAll("[", "<span class='special-text-10'>");

// REPLACE LINE BREAKS

filteredText = filteredText.replaceAll("\n", "<br />");
filteredText = filteredText.replaceAll("\r\n", "<br />");

return filteredText;
}

```

Φάκελος routes

account.php

```

<?php

session_start();

$title = "Λογαριασμός";

$stylesheets =

    '<link rel="stylesheet" href="../css/styles.css">'

    '<link rel="stylesheet" href="../css/account.css">'

    ';

include '../header.php';

if (!isset($_SESSION["logged"])) {

```

```

    echo "<script>window.location = '' . str_replace('\n', '', $baseURL) . ''</script>";

    exit();

}

?>

<body>

    <?php include '../components/navbar.php'; ?>

    <div class="page-content">

        <div class="settings">

            <h3>Ρυθμίσεις</h3>

            <div class="settings-section">

                <h4>Προσωπικά Στοιχεία</h4>

                <form id="change-info-form">

                    <input type="text" disabled name="username" id="settings-username"
placeholder="Όνομα Χρήστη" value="<?php if (isset($_SESSION['username'])) echo
$_SESSION['username'] ?>">

                    <input type="text" name="firstname" id="settings-firstname"
placeholder="Όνομα" value="<?php if (isset($_SESSION['username'])) echo
$_SESSION['firstname'] ?>">

                    <input type="text" name="lastname" id="settings-lastname"
placeholder="Επώνυμο" value="<?php if (isset($_SESSION['username'])) echo
$_SESSION['lastname'] ?>">

                    <button class="button green" type="submit" form="change-info-
form">Αποθήκευση</button>

                </form>

            </div>

```

```

<div class="change-password">

    <h4>Αλλαγή Κωδικού Πρόσβασης</h4>

    <form id="change-password-form">

        <input type="password" name="old-password" id="old-password"
placeholder="Κωδικός Πρόσβασης">

        <input type="password" name="new-password" id="new-password"
placeholder="Νέος Κωδικός Πρόσβασης">

        <input type="password" name="rep-new-password" id="rep-new-password"
placeholder="Επανάληψη Νέου Κωδικού Πρόσβασης">

        <button class="button green" type="submit" form="change-password-
form">Αποθήκευση</button>

    </form>

</div>

<div class="delete-section">

    <h4>Διαγραφή Λογαριασμού</h4>

    <button class="red" onclick="deleteAccount()">Διαγραφή</button>

</div>

</div>

</div>

<script>

    let userInfoForm = document.querySelector(".settings-
section").getElementsByTagName("form")[0];

    let userPasswordForm = document.querySelector(".change-
password").getElementsByTagName("form")[0];

    userInfoForm.addEventListener(

        "submit",

```

```

function(event) {

    event.preventDefault();

    updateUserInfo();

},

true

);

```

```

userPasswordForm.addEventListener("submit",

function(event) {

    event.preventDefault();

    updateUserPassword();

},

true);

```

```

function deleteAccount() {

    Swal.fire({

        title: 'Σίγουρα;',

        text: 'Η διαδικασία είναι μη αναστρέψιμη!',

        icon: "warning",

        showCancelButton: true,

        confirmButtonColor: '#3085d6',

        cancelButtonColor: '#d33',

        confirmButtonText: 'Ναι, σίγουρα',

        cancelButtonText: "Ακύρωση"
    });
}

```

```

    })

    .then(() => {

        const searchParams = new URLSearchParams();

        searchParams.append("submit", "submit");

        fetch(`/${baseUrl}/includes/deleteAccount.php`, {

            method: "POST",

            body: searchParams,

        })

        .then(function(response) {

            return response.text();

        })

        .then(function(text) {

            let error = text.split("=")[1];

            switch (error) {

                case "none":

                    location.reload();

                    break;

                case "notLoggedOrNoUsername":

                    sweetAlertError({

                        text: "Δεν έχετε πρόσβαση σε αυτή τη σελίδα!",

                        redirect: `/${baseUrl}`

                    });

                    break;

```

```

        case "evaluationsDeletionFailed":

            sweetAlertError();

            break;

        case "administratorsDeletionFailed":

            location.reload();

            break;

        case "userDeletionFailed":

            location.reload();

            break;

        default:

            console.log(`Not Updated: ${error}`);

            break;

    }

})

.catch((error) => {

    console.log(`${error}`);

});

});

}

```

```

function updateUserInfo() {

    const userInfoData = new FormData(userInfoForm);

    const searchParams = new URLSearchParams();

```



```

for (const pair of userInfoData) {

    searchParams.append(pair[0], pair[1]);

}

searchParams.append("submit", "submit");

fetch(`${baseUrl}/includes/updateUserInfo.php`, {

    method: "POST",

    body: searchParams,

})

.then(function(response) {

    return response.text();

})

.then(function(text) {

    let error = text.split("=")[1];

    switch (error) {

        case "none":

            sweetAlertSuccess({

                text: "Τα στοιχεία σας ανανεώθηκαν επιτυχώς!"

            })

            break;

        default:

            console.log("Not Updated");

```

```

        sweetAlertError()

        break;
    }

})

.catch((error) => {

    console.log(`${error}`);

});
}

```

```

function updateUserPassword() {

    console.log("updating user password");

    const userPasswordData = new FormData(userPasswordForm);

    const searchParams = new URLSearchParams();

    for (const pair of userPasswordData) {

        if (pair[0] == "" || pair[0] == null || pair[1] == "" || pair[0] == null) {

            sweetAlertError({

                text: "Κάποια πεδία είναι κενά!"

            });

            return;

        }

        searchParams.append(pair[0], pair[1]);

    }
}

```

```
searchParams.append("submit", "submit");
```

```
if (searchParams.get("new-password") !== searchParams.get("rep-new-  
password")) {  
  
    sweetAlertError({  
  
        text: "Οι κωδικοί δεν είναι ίδιοι! Βεβαιωθείται ότι επαναλαμβάνετε τον νέο  
σας κωδικό σωστά."  
  
    });  
  
    return;  
  
}
```

```
fetch(`${baseUrl}/includes/updateUserPassword.php`, {  
  
    method: "POST",  
  
    body: searchParams,  
  
})  
  
.then(function(response) {  
  
    return response.text();  
  
})  
  
.then(function(text) {  
  
    let error = text.split("=")[1];  
  
    switch (error) {  
  
        case "none":  
  
            sweetAlertSuccess({  
  
                title: "",  
  
                text: "Ο κωδικός σας ανανεώθηκε!"  
  
            })  
  
        }  
  
    }  
  
})
```

```

    })

    break;

case "userNotFound":

    sweetAlertError({

        text: "Δεν βρέθηκε ο χρήστης!"

    });

    break;

case "wrongPassword":

    sweetAlertError({

        text: "Ο παλιός κωδικός που δώσατε είναι λάθος!"

    });

    break;

default:

    console.log("Not Updated");

    sweetAlertError({

        text: "Ο κωδικός σας δεν ανανεώθηκε! (Άγνωστο Σφάλμα)"

    });

    break;

}

})

.catch((error) => {

    console.log(`${error}`);

});

}

```

```

</script>

<?php include '../components/footer.php'; ?>

    add_question.php
<?php

session_start();

if (!(isset($_SESSION['logged']) && isset($_SESSION['isAdmin']))) {

    // IF USERS IS NOT LOGGED IN OR

    // IS LOGGED IN AND IS NOT ADMIN,

    // EXIT

    echo "<script>window.location = '' . str_replace('\n', '', $baseURL) . ''</script>";

    exit();

}

$title = "Προσθήκη Ερώτησης";

$stylesheets =

    '<link rel="stylesheet" href="../css/styles.css">

    <link rel="stylesheet" href="../css/add_form.css">

    ';

include '../header.php';

?>

<body>

    <?php include '../components/navbar.php'; ?>

    <div class="page-content">

```

```

<div class="form-wrapper">

  <div class="form-container">

    <label for="choose-question-type">Τύπος Ερώτησης</label>

    <select name="choose-question-type" id="choose-question-type"
onchange="changeQuestionType()">

      <option value="multiple-choice">Multiple Choice</option>

      <option value="drag-drop" selected>Drag n' Drop</option>

    </select>

    <form action="" id="form-multiple-choice">

      <div class="form-section">

        <label for="question-text">Ερώτηση</label>

        <input type="text" name="question-text" id="question-text">

      </div>

      <div class="form-section">

        <label for="question-quiz">Αντίστοιχη Αξιολόγηση</label>

        <select name="question-quiz" id="question-quiz">

          <option value="" selected disabled>Επιλέξτε Αξιολόγηση</option>

        </select>

      </div>

      <div class="form-section">

        <label for="right-answer">Σωστή Απάντηση</label>

        <input type="text" name="right-answer-text" id="right-answer">

      </div>

      <div class="form-section">

```

```

        <label for="wrong-answer-1">Λάθος Απάντηση</label>

        <input type="text" name="wrong-answer-1-text" class="wrong-answer"
key="1">

    </div>

    <div class="form-buttons">

        <a class="button" href="#" onclick="addAnswerInput()">Προσθήκη
Απάντησης</a>

        <a class="button green" href="#"
onclick="submitForm()">Ολοκλήρωση</a>

    </div>

</form>

<form action="" id="form-drag-drop" style="display: none;">

    <div class="form-section">

        <label for="question-text">Ερώτηση</label>

        <input type="text" name="question-text" id="question-text">

    </div>

    <div class="form-section">

        <label for="question-answer-text">Δεδομένο Σκέλος Απάντησης</label>

        <input type="text" name="question-answer-text" id="question-answer-
text">

    </div>

    <div class="form-section">

        <label for="question-quiz">Αντίστοιχη Αξιολόγηση</label>

        <select name="question-quiz" id="question-quiz">

            <option value="" selected disabled>Επιλέξτε Αξιολόγηση</option>

```

```

        </select>

    </div>

    <div class="form-section">

        <label for="right-answer">Σωστή Απάντηση</label>

        <input type="text" name="right-answer-text" id="right-answer">

    </div>

    <div class="form-section">

        <label for="wrong-answer-1">Λάθος Απάντηση</label>

        <input type="text" name="wrong-answer-1-text" class="wrong-answer"
key="1">

    </div>

    <div class="form-buttons">

        <a class="button" href="#" onclick="addAnswerInput()">Προσθήκη
Απάντησης</a>

        <a class="button green" href="#"
onclick="submitForm()">Ολοκλήρωση</a>

    </div>

</form>

</div>

<div class="format-helper">

    <table>

        <tr>

            <td style="letter-spacing: 0.3em">___ <ion-icon name='arrow-
forward'></ion-icon>

        </td>

```



```

        <td class="empty"></td>

</tr>

<tr>

    <td>!!κείμενο/!! <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-1">κείμενο</td>

</tr>

<tr>

    <td>**κείμενο/** <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-2">κείμενο</td>

</tr>

<tr>

    <td>$κείμενο/$ <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-3">κείμενο</td>

</tr>

<tr>

    <td>#κείμενο/# <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-4">κείμενο</td>

</tr>

<tr>

    <td>%κείμενο/% <ion-icon name='arrow-forward'></ion-icon>

```

```

</td>

<td class="special-text-5">κείμενο</td>

</tr>

<tr>

<td>^κείμενο/^ <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-6">κείμενο</td>

</tr>

<tr>

<td>&κείμενο/& <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-7">κείμενο</td>

</tr>

<tr>

<td>@κείμενο/@ <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-8">κείμενο</td>

</tr>

<tr>

<td>?κείμενο/? <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-9">κείμενο</td>

</tr>

<tr>

```

```

        <td>[κείμενο/] <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-10 test">κείμενο</td>

</tr>

</table>

</div>

</div>

<script src="<?php echo $baseURL ?>/js/textFormat.js"></script>

<script>

let formButtons;

const multipleChoiceForm = document.querySelector("#form-multiple-choice");

const dragDropForm = document.querySelector("#form-drag-drop");


let selectQuiz;


changeQuestionType();


function addAnswerInput(questionType) {

    let form;

    if (questionType == "multiple-choice") {

        form = multipleChoiceForm;

        formButtons = multipleChoiceForm.querySelector(".form-buttons");

    } else {

```

```

    form = dragDropForm;

    formButtons = dragDropForm.querySelector(".form-buttons");
}

let inputElement = document.createElement("input");

let labelElement = document.createElement("label");


let wrongAnswersCount = form.querySelectorAll(".wrong-answer").length;


inputElement.setAttribute("type", "text");

inputElement.classList.add("wrong-answer");

inputElement.setAttribute("key", wrongAnswersCount + 1);

inputElement.setAttribute("name", `wrong-answer-${wrongAnswersCount + 1}-
text`);


labelElement.setAttribute("for", `wrong-answer-${wrongAnswersCount + 1}-text`);

labelElement.innerText = `Λάθος Απάντηση`;


formButtons.remove();


let formSection = document.createElement("div");

formSection.classList.add("form-section");

formSection.setAttribute("id", `wrong-answer-${wrongAnswersCount + 1}-
section`);

formSection.appendChild(labelElement);

```

```

let answerControls = document.createElement("div");

answerControls.classList.add("answer-controls");


let deleteAnswerButton = document.createElement("a");

deleteAnswerButton.classList.add("button", "red");

deleteAnswerButton.innerText = "Διαγραφή";

deleteAnswerButton.addEventListener("click", () => {

    deleteAnswerInput(wrongAnswersCount + 1);

});


answerControls.appendChild(inputElement);

answerControls.appendChild(deleteAnswerButton);


formSection.appendChild(answerControls);


form.appendChild(formSection);

form.appendChild(formButtons);

}


function deleteAnswerInput(idToDelete) {

    let elementToDelete = document.querySelector(`#wrong-answer-${idToDelete}-section`);

    elementToDelete.remove();

}

```

```

function submitForm() {

    const questionType = document.querySelector("#choose-question-type").value;

    let questionText;

    let rightAnswer;

    let wrongAnswers;

    let quizID;

    const searchParams = new URLSearchParams();

    if (questionType === "multiple-choice") {

        questionText = multipleChoiceForm.querySelector(".form-section #question-
text").value;

        rightAnswer = multipleChoiceForm.querySelector(".form-section #right-
answer").value;

        wrongAnswers = multipleChoiceForm.querySelectorAll(".form-section .wrong-
answer");

        quizID = multipleChoiceForm.querySelector(".form-section #question-
quiz").value;

    } else if (questionType === "drag-drop") {

        questionText = `${dragDropForm.querySelector(".form-section #question-
text").value}`;

        rightAnswer =

            `<div draggable='true'>${filterText(dragDropForm.querySelector(".form-
section #right-answer").value)}</div>`;

        wrongAnswers = dragDropForm.querySelectorAll(".form-section .wrong-
answer");
    }
}

```

```

    quizID = dragDropForm.querySelector(".form-section #question-quiz").value;

    let givenAnswerSection = dragDropForm.querySelector(".form-section #question-
answer-text").value;

    questionText = `${questionText} \n ${filterText(givenAnswerSection)}`;
}

if (questionText === "" ||
    rightAnswer === "" ||
    quizID === ""
) {
    sweetAlertError({
        text: "Κάποια πεδία είναι κενά!",
    });
    return;
}

searchParams.append("question-type", questionType);
searchParams.append("question", questionText);
searchParams.append("right-answer", rightAnswer);
for (const [i, ans] of wrongAnswers.entries()) {
    if (ans.value === "") {
        sweetAlertError({
            text: "Κάποια πεδία είναι κενά!",
        });
        return;
    }
}

```

```

if (questionType === "multiple-choice") {

    searchParams.append(`wrong-answer-${i}`, ans.value);

} else {

    searchParams.append(`wrong-answer-${i}`, `

232


```



```

    });

    break;

case "rightAnswerNotCreatedOrNoID":

    sweetAlertError({

        text: "Υπήρξε πρόβλημα στη δημιουργία της σωστής απάντησης!",

    });

    break;

case "emptyInputs":

    sweetAlertError({

        text: "Κάποια πεδία είναι κενά!",

    });

    break;

case "questionNotCreatedOrNoID":

    sweetAlertError({

        text: "Υπήρξε πρόβλημα στη δημιουργία της ερώτησης!",

    });

    break;

case "accessDenied":

    sweetAlertError({

        text: "Δεν έχετε πρόσβαση σε αυτή τη λειτουργία!",

    });

    break;

default:

    break;

```

```

    }

  })

  .catch(function(error) {

    console.log(error);

  });

}

```

```

function changeQuestionType() {

  const dropdownValue = document.querySelector("#choose-question-type").value;

  console.log(`Question type: ${dropdownValue}`);

  switch (dropdownValue) {

    case "multiple-choice":

      multipleChoiceForm.style.display = "flex";

      dragDropForm.style.display = "none";

      selectQuiz = multipleChoiceForm.querySelector("select#question-quiz");

      break;

    default:

      multipleChoiceForm.style.display = "none";

      dragDropForm.style.display = "flex";

      selectQuiz = dragDropForm.querySelector("select#question-quiz");

      break;

  }

}

```

```

fetch(`/${baseUrl}/includes/fetchQuizzes.php`).then((res) => {

    return res.json();

}).then((jsonArray) => {

    jsonArray.forEach(element => {

        const option = document.createElement("option");

        option.classList.add("quiz");

        option.innerText = element.quizTitle

        option.setAttribute("value", element.quizID)


        selectQuiz.appendChild(option);

    });

}).catch((error) => {

    console.error(`${error}`);

});

</script>

<script src="<?php echo $baseUrl ?>/js/testingFile.js"></script>

<?php include '../components/footer.php'; ?>

    add_quiz.php

<?php

session_start();

if (!(isset($_SESSION['logged']) && isset($_SESSION["isAdmin"]))) {

    // IF USERS IS NOT LOGGED IN OR

    // IS LOGGED IN AND IS NOT ADMIN,

    // EXIT

```

```

    echo "<script>window.location = '' . str_replace("\n", '', $baseURL) . ''</script>";

    exit();
}

```

```

$title = "Προσθήκη Αξιολόγησης";

```

```

$stylesheets =

```

```

    '<link rel="stylesheet" href="../css/styles.css">

```

```

    <link rel="stylesheet" href="../css/add_form.css">

```

```

    ';

```

```

include '../header.php';

```

```

?>

```

```

<body>

```

```

    <?php include '../components/navbar.php'; ?>

```

```

    <div class="page-content">

```

```

        <div class="form-container">

```

```

            <form>

```

```

                <div class="form-section">

```

```

                    <label for="quiz-text">Όνομα Αξιολόγησης</label>

```

```

                    <input type="text" name="quiz-text" id="quiz-text">

```

```

                </div>

```

```

                <div class="form-buttons">

```

```

                    <a class="button green" href="#"

```

```

onclick="submitForm()">Ολοκλήρωση</a>

```

```

                </div>

```

```

        </form>

    </div>

</div>

<script>

let quizForm = document.querySelector(".form-container");

quizForm.addEventListener(

    "submit",

    function(event) {

        event.preventDefault();

        submitForm();

    },

    true

);

function submitForm() {

    const formData = new FormData(

        document

        .querySelector(".form-container")

        .getElementsByTagName("form")[0]

    );

    const searchParams = new URLSearchParams();

```

```

for (const pair of formData) {
  if (pair[0] == "" || pair[0] == null || pair[1] == "" || pair[0] == null) {
    sweetAlertError({
      text: "Κάποια πεδία είναι κενά!",
    });
    return;
  }
  searchParams.append(pair[0], pair[1]);
}

```

```

searchParams.append("submit", "submit");

```

```

fetch(`${baseUrl}/includes/newQuizHandler.php`, {
  method: "POST",
  body: searchParams,
})
.then(function(response) {
  return response.text();
})
.then(function(text) {
  let error = text.split("=")[1];

  switch (error) {
    case "none":

```

```

        location.reload();

        break;

    case "stmtFailed":

        sweetAlertError({

            text: "Κάτι πήγε στραβά! Προσπαθήστε πάλι αργότερα.",

        });

        break;

    case "quizAlreadyExists":

        sweetAlertError({

            text: "Υπάρχει ήδη αξιολόγηση με αυτό το όνομα.",

        });

        break;

    default:

        sweetAlertError({

            text: `Υπήρξε ένα ασυνήθιστο λάθος: ${error}`,

        });

        break;

    }

})

.catch(function(error) {

    console.log(error);

});

}

</script>

```

```

<?php include '../components/footer.php'; ?>

    add_rule.php
<?php
session_start();

if (!(isset($_SESSION['logged']) && isset($_SESSION['isAdmin']))) {

    // IF USERS IS NOT LOGGED IN OR

    // IS LOGGED IN AND IS NOT ADMIN,

    // EXIT

    echo "<script>window.location = '' . str_replace(\"\\n\", \"\", $baseUrl) . \"\"</script>";

    exit();

}

$title = "Προσθήκη Κανόνα";

$stylesheets =

    '<link rel="stylesheet" href="../css/styles.css">'

    '<link rel="stylesheet" href="../css/add_form.css">'

    ';

include '../header.php';

?>

<body>

    <?php include '../components/navbar.php'; ?>

    <div class="page-content">

        <div class="form-wrapper">

```



```

<div class="form-container">

  <form>

    <div class="form-section">

      <label for="rule-name">Όνομα Κανόνα</label>

      <input type="text" name="rule-name" id="rule-name">

    </div>

    <!-- DROPDOWN TO SELECT CORRESPONDING QUIZ -->

    <div class="form-section">

      <label for="rule-quiz">Αντίστοιχη Αξιολόγηση</label>

      <select name="rule-quiz" id="rule-quiz">

        <option value="" selected disabled>Επιλέξτε Αξιολόγηση</option>

      </select>

    </div>

    <!-- DYNAMIC SECTIONS THAT HAVE A TEXT SECTION
EXPLAINING THE RULE AND AN EXAMPLE SECTION
FOR POTENTIAL EXAMPLES -->

    <div class="form-section rule-form-section" id="form-section-1">

      <div class="sectionHeader">

        <h3>Τμήμα Κανόνα</h3>

      </div>

      <label for="rule-section-1-text">Κείμενο Τμήματος</label>

      <textarea name="rule-section-1-text" class="rule-section-text"
key="1"></textarea>

      <label for="rule-section-1-example">Παράδειγμα Τμήματος</label>

```

```

        <textarea name="rule-section-1-example" class="rule-section-example"
key="1"></textarea>

    </div>

    <div class="form-buttons">

        <a class="button" href="#" onclick="addSectionInput()">Προσθήκη
Τμήματος</a>

        <a class="button green" href="#"
onclick="submitForm()">Ολοκλήρωση</a>

    </div>

</form>

</div>

<div class="format-helper">

    <table>

        <tr>

            <td>!!κείμενο!! <ion-icon name='arrow-forward'></ion-icon>

            </td>

            <td class="special-text-1">κείμενο</td>

        </tr>

        <tr>

            <td>**κείμενο/** <ion-icon name='arrow-forward'></ion-icon>

            </td>

            <td class="special-text-2">κείμενο</td>

        </tr>

        <tr>

            <td>$κείμενο/$ <ion-icon name='arrow-forward'></ion-icon>

```

```

</td>

<td class="special-text-3">κείμενο</td>

</tr>

<tr>

<td>#κείμενο/# <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-4">κείμενο</td>

</tr>

<tr>

<td>%κείμενο/% <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-5">κείμενο</td>

</tr>

<tr>

<td>^κείμενο/^ <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-6">κείμενο</td>

</tr>

<tr>

<td>&κείμενο/& <ion-icon name='arrow-forward'></ion-icon>

</td>

<td class="special-text-7">κείμενο</td>

</tr>

<tr>

```

```

        <td>@κείμενο/@ <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-8">κείμενο</td>

</tr>

<tr>

    <td>?κείμενο/? <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-9">κείμενο</td>

</tr>

<tr>

    <td>[κείμενο/] <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-10 test">κείμενο</td>

</tr>

<tr>

    <td>==> <ion-icon name='arrow-forward'></ion-icon>

    </td>

    <td class="special-text-0 test">

        <ion-icon name='arrow-forward'></ion-icon>

    </td>

</tr>

</table>

</div>

</div>

```

```

<div class="export-HTML-btn">

    <button class="yellow" onclick="exportHTML()">Export HTML</button>

</div>

<div class="exported-HTML" style="width:80%;max-width:50em;text-align:center;line-height:2">


</div>

</div>

<script src="<?php echo $baseURL ?>/js/textFormat.js"></script>

<script>

let form = document.querySelector("form");

let formButtons = document.querySelector(".form-buttons");

let selectQuiz = document.querySelector("#rule-quiz");

let exportedHTML;

function addSectionInput() {

    let formSectionCount = document.querySelectorAll(".rule-form-section").length;

    let formSection = document.createElement("div");

    formSection.classList.add("form-section");

    formSection.classList.add("rule-form-section");

    formSection.setAttribute("id", `form-section-${formSectionCount + 1}`);

    let sectionHeader = document.createElement("div");

```

```
sectionHeader.classList.add("section-header");
```

```
let header = document.createElement("h3");
```

```
header.innerText = "Τμήμα Κανόνα";
```

```
let delBtn = document.createElement("a");
```

```
delBtn.classList.add("button", "red");
```

```
delBtn.innerText = "Διαγραφή";
```

```
delBtn.addEventListener("click", () => {
```

```
    deleteFormSection(formSectionCount + 1);
```

```
});
```

```
sectionHeader.appendChild(header);
```

```
sectionHeader.appendChild(delBtn);
```

```
let inputElement1 = document.createElement("textarea");
```

```
let labelElement1 = document.createElement("label");
```

```
let inputElement2 = document.createElement("textarea");
```

```
let labelElement2 = document.createElement("label");
```

```
labelElement1.setAttribute("for", `rule-section-${formSectionCount + 1}-text`);
```

```
labelElement1.innerText = "Κείμενο Τμήματος";
```

```
inputElement1.setAttribute("name", `rule-section-${formSectionCount + 1}-text`);
```

```
inputElement1.setAttribute("key", `${formSectionCount + 1}`);
```

```
inputElement1.classList.add("rule-section-text");
```

```

labelElement2.setAttribute("for", `rule-section-${formSectionCount + 1}-example`);

labelElement2.innerText = "Παράδειγμα Τμήματος";

inputElement2.setAttribute("name", `rule-section-${formSectionCount + 1}-example`);

inputElement2.setAttribute("key", `${formSectionCount + 1}`);

inputElement2.classList.add("rule-section-example");


formSection.appendChild(sectionHeader);

formSection.appendChild(labelElement1);

formSection.appendChild(inputElement1);

formSection.appendChild(labelElement2);

formSection.appendChild(inputElement2);


formButtons.remove();


form.appendChild(formSection);

form.appendChild(formButtons);
}


function deleteFormSection(idToDelete) {

    let elementToDelete = document.querySelector(`#form-section-${idToDelete}`);

    elementToDelete.remove();

}


function submitForm() {

```

```

const returnedData = getFormData();

if (returnedData[0] == "" || returnedData[1] == "" || returnedData[2] == "") {

    sweetAlertWarning({

        title: "Προσοχή!",

        text: "Κάποια πεδία είναι κενά!",

        confirmText: "Εντάξει",

    });

    return;
}

const searchParams = new URLSearchParams();

searchParams.append("rule-name", returnedData[0]);
searchParams.append("rule-quiz", returnedData[1]);
searchParams.append("rule-text", returnedData[2]);

searchParams.append("submit", "submit");

exportedHTML = searchParams;

fetch(`${baseUrl}/includes/newRuleHandler.php`, {

    method: "POST",

```



```

    body: searchParams,
  }).then(function(response) {
    return response.text();
  })
  .then(function(text) {
    let error = text.split("=")[1];
    switch (error) {
      case "none":
        location.reload();
        break;
      case "accessDenied":
        sweetAlertError({
          text: "Δεν έχετε πρόσβαση σε αυτή τη σελίδα!",
          confirmText: "Εντάξει",
          cancelText: "Ακύρωση",
          redirect: `/${baseUrl}`
        })
        break;
      case "quizNotProvided":
        sweetAlertWarning({
          title: "Προσοχή!",
          text: "Κάποια πεδία είναι κενά!",
          confirmText: "Εντάξει",
        });
    }
  });

```

```

        break;

    case "quizNotFound":

        sweetAlertError({

            text: "Δεν βρέθηκε η αξιολόγηση!",

            confirmText: "Εντάξει",

            cancelText: "Ακύρωση"

        })

        break;

    case "stmtFailed":

        break;

    case "ruleAlreadyExists":

        sweetAlertWarning({

            title: "Προσοχή!",

            text: "Αυτός ο κανόνας υπάρχει ήδη!",

            confirmText: "Εντάξει",

        });

        break;

    default:

        break;

    }

})

.catch(function(error) {

    console.log(error);

});

```

```
}
```

```
function getFormData() {

    const ruleName = document.querySelector("#rule-name").value;

    const ruleQuiz = selectQuiz.value;

    const ruleSections = document.querySelectorAll(".rule-form-section");

    const ruleSectionText = [];

    const ruleSectionExample = [];

    let textStr = "";

    for (const rule of ruleSections) {

        const text = rule.querySelector(".rule-section-text");

        const example = rule.querySelector(".rule-section-example");

        textStr += '<div class="rule"><div class="rule-section">';

        if (text) {

            ruleSectionText.push(filterText(text.value));

            textStr += `<p class="rule-text">${filterText(text.value)}</p>`;

        }

        if (example && example.value !== "") {

            ruleSectionExample.push(filterText(example.value));

            textStr +=

                `<div class="rule-example"><p class="rule-example-
header">Παράδειγμα</p><p class="example">${filterText(example.value)}</p></div>`;

        }

    }

}
```

```

        textStr += '</div></div>';
    }

    return [ruleName, ruleQuiz, textStr];
}

fetch(`${baseUrl}/includes/fetchQuizzes.php`).then((res) => {
    return res.json();
}).then((jsonArray) => {
    jsonArray.forEach(element => {
        const option = document.createElement("option");
        option.setAttribute("value", element.quizID);
        option.innerText = element.quizTitle;

        selectQuiz.appendChild(option);

    });
}).catch((error) => {
    console.error(`${error}`);
});

function exportHTML() {
    const container = document.querySelector(".exported-HTML");
    const arr = getFormData();

```

```

        container.innerHTML = `${String(arr).replace("==>", "<ion-icon name='arrow-
forward'></ion-icon>").replace(/&/g, "&amp;")

        .replace(/</g, "&lt;")

        .replace(/>/g, "&gt;")

        .replace(/"/g, "\\&quot;")

        .replace(/'/g, "&#039;")} `;

    }

</script>

<?php include '../components/footer.php'; ?>

    admin_panel.php
<?php

session_start();

$title = "Διαχείριση";

$stylesheets =

    '<link rel="stylesheet" href="../css/styles.css">

    <link rel="stylesheet" href="../css/admin_panel.css">';

include '../header.php';

if (!isset($_SESSION["isAdmin"])) {

    echo "<script>window.location = '' . str_replace('\\n', '', $baseUrl) . ''</script>";

    exit();

}

?>

```

```

<body>

<script src="<?php echo $baseUrl ?>/js/fetchAdminData.js"></script>

<?php include '../components/navbar.php'; ?>

<div class="page-content">

    <div class="admin-panel-container">

        <div class="user-data-container">

            <h2>Δεδομένα Χρηστών</h2>

            <div id="users-data"></div>

        </div>

        <div class="filter">

            <label for="filter-assets">Εμφάνιση</label>

            <select onchange="filterAssets()" name="filter-assets" id="filter-assets">

                <option selected value="all">Όλα</option>

                <option value="quiz">Αξιολογήσεις</option>

                <option value="rule">Κανόνες</option>

                <option value="question">Ερωτήσεις</option>

            </select>

        </div>

        <div class="quizzes">

            <h2>Αξιολογήσεις</h2>

            <div class="quizzes-container"></div>

            <button class="blue" onclick="window.location = 'add_quiz.php';">Προσθήκη
αξιολόγησης</button>

        </div>

        <div class="rules">

```

<h2>Κανόνες</h2>

<div class="rules-container"></div>

<button class="blue" onclick="window.location = 'add_rule.php';">Προσθήκη
κανόνα</button>

</div>

<div class="questions">

<h2>Ερωτήσεις</h2>

<div class="questions-container"></div>

<button class="blue" onclick="window.location =
'add_question.php';">Προσθήκη ερώτησης</button>

</div>

<div class="db-functions">

<h2>Λειτουργίες Βάσης Δεδομένων</h2>

<div class="db-functions-buttons">

<div class="excel-exporter">

<h4>Εξαγωγή δεδομένων σε αρχείο Microsoft Excel</h4>

<button class="green-inverse" onclick="exportXLSX()">

Export to Excel

<ion-icon name="download-outline"></ion-icon>

</button>

</div>

<div class="db-flusher">

<h4>Άδειασμα Βάσης Δεδομένων</h4>

<button class="red" onclick="flushDatabase()">

Άδειασμα Βάσης

```
<ion-icon name="trash"></ion-icon>
```

```
</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script>
```

```
const quizzesContainer =  
(document.querySelector(".quizzes").querySelector(".quizzes-container"));  
  
const quizzes = (document.querySelector(".quizzes"));  
  
const rulesContainer = (document.querySelector(".rules").querySelector(".rules-  
container"));  
  
const rules = (document.querySelector(".rules"));  
  
const quesContainer =  
(document.querySelector(".questions").querySelector(".questions-container"));  
  
const ques = (document.querySelector(".questions"));  
  
const dropdown = document.querySelector("select");  
  
  
const usersContainer = document.querySelector("#users-data");  
  
  
function filterAssets() {  
  switch (dropdown.value) {  
    case "quiz":
```



```

        quizzes.style.display = "block";

        rules.style.display = "none";

        ques.style.display = "none";

        break;

    case "rule":

        quizzes.style.display = "none";

        rules.style.display = "block";

        ques.style.display = "none";

        break;

    case "question":

        quizzes.style.display = "none";

        rules.style.display = "none";

        ques.style.display = "block";

        break;

    default:

        quizzes.style.display = "block";

        rules.style.display = "block";

        ques.style.display = "block";

        break;

    }

}

```

```

function deleteRule(index) {

    const searchParams = new URLSearchParams();

```

```

searchParams.append("submit", "submit");

searchParams.append("ruleID", index);


fetch(`${baseUrl}/includes/deleteRule.php`, {
  method: "POST",
  body: searchParams,
}).then((res) => {
  return res.text();
}).then((text) => {
  const error = text.split("=")[1];
  switch (error) {
    case "none":
      location.reload();
      break;
    default:
      break;
  }
}).catch((error) => {
  console.log(error);
});
}

function deleteQuiz(index) {

```

```

console.log(`DELETING QUIZ ${index}`);

const searchParams = new URLSearchParams();

searchParams.append("submit", "submit");

searchParams.append("quizID", index);

fetch(`${baseUrl}/includes/deleteQuiz.php`, {
  method: "POST",
  body: searchParams,
}).then((res) => {
  return res.text();
}).then((text) => {
  const error = text.split("=")[1];

  switch (error) {
    case "none":
      location.reload();
      break;
    default:
      break;
  }
}).catch((error) => {
  console.log(error);
});
}

```

```

function deleteQuestion(index) {

    const searchParams = new URLSearchParams();

    searchParams.append("submit", "submit");

    searchParams.append("questionID", index);

    fetch(`/${baseURL}/includes/deleteQuestion.php`, {

        method: "POST",

        body: searchParams,

    }).then((res) => {

        return res.text();

    }).then((text) => {

        const error = text.split("=")[1];

        switch (error) {

            case "none":

                location.reload();

                break;

            default:

                break;

        }

    }).catch((error) => {

        console.log(error);

    });
}

```

```
}
```

```
function flushDatabase() {  
  
    const searchParams = new URLSearchParams();  
  
    searchParams.append("submit", "submit");  
  
    fetch(`${baseUrl}/includes/flushGrades.php`, {  
  
        method: "POST",  
  
        body: searchParams  
  
    }).then((res) => {  
  
        location.reload();  
  
    }).catch((error) => {  
  
        console.error(`${error}`);  
  
    });  
}
```

```
function exportXLSX() {  
  
    window.location = `${baseUrl}/includes/exportXLSX.php`;  
  
}
```

```
fetch(`${baseUrl}/includes/fetchQuizzes.php`).then((res) => {  
  
    return res.json();  
  
}).then((jsonArray) => {  
  
    jsonArray.forEach(element => {  
  
        const div = document.createElement("div");
```

```

div.classList.add("quiz");

const paragraph = document.createElement("p");
paragraph.innerText = element.quizTitle;
paragraph.classList.add("quiz-name");
div.appendChild(paragraph);

const delButton = document.createElement("button");
delButton.classList.add("red");
delButton.innerText = "Διαγραφή";
delButton.addEventListener("click", () => {
    deleteQuiz(element.quizID)
});
div.appendChild(delButton);

quizzesContainer.appendChild(div);

});

}).catch((error) => {
    console.error(`${error}`);
});

fetch(`${baseUrl}/includes/fetchRules.php`).then((res) => {
    return res.json();

```

```

}).then((jsonArray) => {

  jsonArray.forEach(element => {

    const div = document.createElement("div");

    div.classList.add("rule");

    const paragraph = document.createElement("p");

    paragraph.innerText = element.ruleName;

    paragraph.classList.add("rule-name");

    div.appendChild(paragraph);

    const delButton = document.createElement("button");

    delButton.classList.add("red");

    delButton.innerText = "Διαγραφή";

    delButton.addEventListener("click", () => {

      deleteRule(element.ruleID)

    });

    div.appendChild(delButton);

    rulesContainer.appendChild(div);

  });

}).catch((error) => {

  console.error(`${error}`);

```

```
});
```

```
fetch(`/${baseUrl}/includes/fetchQuestions.php`).then((res) => {  
    return res.json();  
}).then((jsonArray) => {
```

```
    jsonArray.forEach(element => {  
        const div = document.createElement("div");  
        div.classList.add("question");  
  
        const paragraph = document.createElement("p");  
        paragraph.classList.add("question-text");  
        paragraph.innerText = element.questionText.split("<")[0];  
  
        const button = document.createElement("button");  
        button.classList.add("red");  
        button.addEventListener("click", () => {  
            deleteQuestion(element.questionID);  
        })  
        button.innerText = "Διαγραφή";  
  
        div.appendChild(paragraph);  
        div.appendChild(button);
```



```

        quesContainer.appendChild(div);

    });

}).catch((error) => {

    console.error(`${error}`);

});

const searchParams = new URLSearchParams();

searchParams.append("multiple", "true");


fetch(`${baseUrl}/includes/fetchUserData.php`, {

    method: "POST",

    body: searchParams,

}).then((res) => {

    return res.text();

}).then((text) => {

    const jsonArray = JSON.parse(text);

    console.log(jsonArray);

    jsonArray.forEach(element => {

        // CREATE LINK FOR THE USER

        const userLink = document.createElement("a");

        userLink.setAttribute("href",

            `${baseUrl}/routes/user_info.php?user=${element.user.userID}`);

        userLink.innerText = `${element.user.userFirstName}

${element.user.userLastName}`;

        userLink.classList.add("user-link");

```

```

        usersContainer.appendChild(userLink);

    });

}).catch((error) => {

    console.error(`${error}`);

});

function exportSQLData() {

    window.location = `${baseUrl}/includes/exportSQLData.php`;

}

</script>

<?php include '../components/footer.php'; ?>

    end_game.php
<?php
session_start();

$title = "Συγχαρητήρια";

$stylesheets =

    '<link rel="stylesheet" href="../css/styles.css">

    <link rel="stylesheet" href="../css/quiz.css">

    <link rel="stylesheet" href="../css/end_game.css">

    ';

include '../header.php';

if (!isset($_SESSION["logged"])) {

    echo "<script>window.location = '' . str_replace(\"\\n\", \"\", $baseUrl) . \"\"</script>";

    exit();

```

}

?>

<body>

<?php include '../components/navbar.php'; ?>

<div class="page-content"

style="display: flex; flex-direction: column; align-items: center; justify-content: center;">

<h2 style="text-align: center;">Συγχαρητήρια!</h2>

<p>Ο Μπρόνκο είναι χαρούμενος!</p>

<div class="bronco fade-in">

</div>

<div class="confetti-container">


```

        </div>

    </div>

    <?php include '../components/footer.php'; ?>

    quiz.php
    <?php
    session_start();

    $title = "Quiz";

    $stylesheets =

        '<link rel="stylesheet" href="../css/styles.css">'

        '<link rel="stylesheet" href="../css/quiz.css">'

        ';

    include '../header.php';

    if (!isset($_SESSION["logged"])) {

        echo "<script>window.location = '' . str_replace(\"\\n\", \"\", $baseUrl) . \"\"</script>";

        exit();

    }

    require_once "../includes/functions.php";

    ?>

    <body>

        <?php include '../components/navbar.php'; ?>

        <div class="page-content">

```

```

    <div class="quiz-container">

    </div>

</div>

<script>

let forQuiz = localStorage.getItem("quizProgress");

if (forQuiz !== null) {

    console.log(`forQuiz: ${forQuiz}`)

} else {

    console.log(`forQuiz is null: ${forQuiz}`)

}

const questionResults = [];

let questions;

let currentQuestionIndex = 0;

let score = 0;

let counter = 0;

let totalCounter = 0;


let studyTime = 0;

const urlParams = new URLSearchParams(window.location.search);

if (urlParams.has('studyTime')) {

    studyTime = parseFloat(urlParams.get('studyTime'));

```

```

}

setInterval(() => {

    counter++;

}, 100);

const quizContainer = document.querySelector(".quiz-container");

fetchQuestions(forQuiz);

function fetchQuestions(forQuiz) {

    const searchParams = new URLSearchParams();

    searchParams.append("submit", "submit");

    if (forQuiz !== "" && forQuiz !== null) {

        searchParams.append("quizID", forQuiz);

        fetchNeededData(searchParams);

    } else {

        console.log("No forQuiz... Fetching quiz")

        fetchQuizzes().then((res) => {

            const jsonArray = res;

            console.log(jsonArray);

            forQuiz = jsonArray[0].quizID;

            console.log("For quiz is: " + forQuiz);

```

```

        localStorage.setItem("quizProgress", forQuiz);

        searchParams.append("quizID", forQuiz);

        fetchNeededData(searchParams);

    })
}

function fetchNeededData(searchParams) {

    fetch(`/${baseURL}/includes/fetchQuizQuestions.php`, {

        method: "POST",

        body: searchParams

    }).then((res) => {

        return res.text();

    }).then((text) => {

        questions = replaceSpecialCharacters(text);

        populateQuiz(currentQuestionIndex);

    }).catch((err) => {

        console.error(err);

    });

}

}

async function populateQuiz(questionIndex) {

```

```

// Call drag-drop function to initialize element references

initializeElements();

const questionsJSON = JSON.parse(questions);

delete questionsJSON["error"];

if (!Object.keys(questionsJSON)[questionIndex]) {

    await setQuestionGrades(questionResults);

    // If score is equal to the amount of questions fetched,
    // The player have answered all the questions correctly
    // Move on to the next quiz

    console.log(
        `Now check if the player has answered correctly. To do so score (${score})
        must be equal to ${Object.keys(questionsJSON).length}`
    )

    if (score == Object.keys(questionsJSON).length) {

        const searchParams = new URLSearchParams();

        searchParams.append("submit", "submit");

        searchParams.append("quizID", localStorage.getItem("quizProgress"));

        fetch(`${baseUrl}/includes/completeGame.php`, {

```



```

    method: "POST",

    body: searchParams
  }).then((res) => {

    return res.text();

  }).then((text) => {

    const jsonResponse = JSON.parse(text);

    if (jsonResponse.error == "none") {

      if (jsonResponse.answer === "true") {

        console.log("Was last quiz");

        // REDIRECT TO END GAME

        window.location = `${baseUrl}/routes/end_game.php`;

        return;

      } else {

        console.log("Was NOT last quiz");

        // INCREASE QUIZ PROGRESS

        fetchQuizzes().then((res) => {

          const jsonArray = res;

          console.log(jsonArray);

          for (let index = 0; index < jsonArray.length; index++) {

            const element = jsonArray[index];

```

```

        console.log(
            `Is ${element.quizID} equal to
            ${localStorage.getItem("quizProgress")}?`
        );

        if (element.quizID == localStorage.getItem("quizProgress")) {
            console.log(`Yes, at index ${index}`);
        } else {
            console.log("No");
        }

        if (element.quizID == localStorage.getItem("quizProgress")) {
            localStorage.setItem("quizProgress", jsonArray[index + 1]
                .quizID);

            // RELOAD PAGE

            location.reload(true);

            return;
        }
    }

    })

}

} else {
    console.log("Unexpected error!");
}

})

.catch(err => {
    console.error(err);

```

```

    })

    return;

}

// Else, redirect to the corresponding rules

window.location =

`${baseUrl}/routes/rules.php?index=${localStorage.getItem("quizProgress")}`;

return;

}


quizContainer.innerHTML = "";

let wantedKey = Object.keys(questionsJSON)[questionIndex];


const element = questionsJSON[wantedKey];


const quizQuestion = document.createElement("div");

quizQuestion.classList.add("quiz-question");

const quizQuestionText = document.createElement("p");

quizQuestionText.innerHTML = element.text;

quizQuestion.appendChild(quizQuestionText);


quizQuestionText.addEventListener("click", (event) => {

    responsiveVoice.speak(`${event.target.innerText}`, "Greek Female");

});

quizQuestionText.style.cursor = "help";

```

```
quizContainer.appendChild(quizQuestion);
```

```
const quizAnswers = document.createElement("div");
```

```
quizAnswers.classList.add("quiz-answers");
```

```
const answersArray = [];
```

```
let index = 0;
```

```
for (const elementKey in element) {
```

```
  if (Object.hasOwnProperty.call(element, elementKey)) {
```

```
    const childElement = element[elementKey];
```

```
    if (elementKey !== "text" && elementKey !== "type") {
```

```
      const answer = [];
```

```
      answer.push(elementKey, childElement);
```

```
      answersArray.push(answer);
```

```
    }
```

```
  }
```

```
}
```

```
const shuffledAnswers = answersArray.sort((a, b) => 0.5 - Math.random());
```

```
shuffledAnswers.forEach((elem) => {
```

```
  const answerContainer = document.createElement("div")
```

```
  answerContainer.classList.add("answer-container");
```

```

let answer;

if (element["type"] == "drag-drop") {
    answer = new DOMParser().parseFromString(elem[1],
"text/html").body.firstElementChild;
    quizAnswers.appendChild(answer);
    index++;
}

if (element["type"] == "multiple-choice") {
    answer = document.createElement("a");
    answer.setAttribute("href", "#");
    answer.setAttribute("id", index);
    answer.addEventListener("click", (e) => {
        answerQuestion(wantedKey, elem[1]);
    })
    answer.innerHTML = elem[1];
    answerContainer.appendChild(answer);
    quizAnswers.appendChild(answerContainer);

    index++;
}

});

quizContainer.appendChild(quizAnswers);

if (element["type"] == "drag-drop") {

```

```

const confirmButton = document.createElement("button");

confirmButton.classList.add("green");

confirmButton.innerText = "Επόμενο";

confirmButton.style.marginBlockEnd = "1em";

confirmButton.style.marginInlineStart = "1em";

confirmButton.addEventListener("click", () => {

    if (document.querySelector(".empty").innerHTML == "") {

        sweetAlertWarning({

            title: "Προσοχή!",

            text: "Δεν έχει απαντήσει στην ερώτηση!",

            confirmText: "Εντάξει",

        });

        return;

    }

    answerQuestion(wantedKey, currentAnswer.innerText);

});

quizContainer.appendChild(confirmButton);

}

}

function answerQuestion(questionIndex, answerText) {

```

```

const questionsJSON = JSON.parse(questions);

const result = [];

let rightAnswer;

rightAnswer = questionsJSON[questionIndex]["answer-0"];

if (questionsJSON[questionIndex].type === "drag-drop") {

    const answerTemp = document.createElement("div");

    answerTemp.innerHTML = questionsJSON[questionIndex]["answer-0"];

    rightAnswer = answerTemp.innerText;

}

result.push(questionsJSON[questionIndex]["text"]);

if (rightAnswer === answerText) {

    score++;

    result.push(1);

} else {

    result.push(0);

}

questionResults.push(result);

```

```

currentQuestionIndex++;

populateQuiz(currentQuestionIndex);

totalCounter += counter;

counter = 0;
}

function setQuestionGrades(results) {
  return new Promise((resolve, reject) => {
    const searchParams = new URLSearchParams();
    searchParams.append("submit", "submit");
    searchParams.append("studyTime", studyTime);
    searchParams.append("results", JSON.stringify(questionResults));
    searchParams.append("totalTime", totalCounter);
    searchParams.append("quizID", localStorage.getItem("quizProgress"));

    resolve(fetch(`${baseUrl}/includes/setGrades.php`, {
      method: "POST",
      body: searchParams
    })).then((res) => {
      return res.text();
    }).then((text) => {

```



```

const error = JSON.parse(text).error;

switch (error) {

  case "none":

    console.log("Hooray! Grades set successfully!");

    break;

  default:

    console.log("No-ray!");

    break;

}

}).catch((error) => {

  console.log(error);

}))

});

}

function completeQuiz() {

  fetchQuizzes().then((res) => {

    const jsonResponse = res;

    console.log(jsonArray);

    console.log(`Gonna check if any of the above IDs match
    ${localStorage.getItem("quizProgress")}`)

```

```

    jsonResponse.forEach((element, index) => {

        if (parseInt(element.quizID) ===
parseInt(localStorage.getItem("quizProgress"))) {

            console.log(`Found Quiz in the list at index: ${index}`);

            if (jsonResponse[jsonResponse.length - 1] == element) {

                console.log("Last Item");

            }

            if (jsonResponse.length > 1) {

                localStorage.setItem("quizProgress", jsonResponse[index + 1].quizID);

                location.reload();

            }

            return;

        }

    });

    console.log("Not matched!!!");

})

}

function fetchQuizzes() {

    console.log("INSIDE fetchQuizzes()")

```

```

return new Promise((resolve, reject) => {

    resolve(fetch(`${baseUrl}/includes/fetchQuizzes.php`)

        .then((res) => {

            return res.text();

        }).then((text) => {

            const error = JSON.parse(text).error;

            return JSON.parse(text);

        }).catch((error) => {

            console.log(error);

        })))

});

}

</script>

<script src="<?php echo $baseUrl ?>/js/dragDropGame.js"></script>

<?php include '../components/footer.php'; ?>

    rules.php
<?php
session_start();

$title = "Καρόνες";

$stylesheets =

    '<link rel="stylesheet" href="../css/styles.css">

    <link rel="stylesheet" href="../css/rules.css">';

include '../header.php';

```

```

if (!isset($_SESSION["logged"])) {

    echo "<script>window.location = '' . str_replace("\n", '', $baseURL) . ''</script>";

    exit();

}

?>

```

```

<body>

    <?php include '../components/navbar.php'; ?>

    <div class="page-content">

        <div class="rule-container">

            <div class="rule-header">

                <p></p>

            </div>

            <div class="rule-body"></div>

            <div class="buttons-container">

                <div class="prev-button">

                    <button class="blue" onclick="updateRule(-1)">

                        <ion-icon name="caret-back"></ion-icon>

                        Προηγούμενο

                    </button>

                </div>

                <div class="next-buttons">

                    <button class="blue" onclick="updateRule(1)">

                        Επόμενο

```

```

        <ion-icon name="chevron-forward"></ion-icon>

    </button>

</div>

<button class="blue" id="play-button"

    onclick="redirectTo(`/${baseUrl}/routes/quiz.php${totalCounter !== 0 ?
'?studyTime=' + totalCounter : ''}`)">

```

Παίξε

```

        <ion-icon name="caret-forward"></ion-icon>

    </button>

</div>

</div>

<script>

const ruleBody = document.querySelector(".rule-body");

const buttons = document.querySelector(".buttons-container");

let rulesObj;

let ruleIndex = 0;


let totalCounter = 0;


setInterval(() => {

    totalCounter++;

}, 100);


const searchParams = new URLSearchParams();

```

```
searchParams.append("submit", "submit");

searchParams.append("quizIndex", "<?php echo $_GET['index'] ?>");
```

```
fetch(`${baseUrl}/includes/fetchRules.php`, {
  method: "POST",
  body: searchParams,
}).then((res) => {
  return res.json();
}).then((jsonArray) => {
  rulesObj = jsonArray;
  updateRule(ruleIndex);
}).catch((error) => {
  console.log(error);
});
```

```
function updateRule(index) {
  ruleIndex += index;

  const element = rulesObj[ruleIndex];

  const prevBtn = buttons.querySelector(".prev-button");
  const nextBtns = buttons.querySelector(".next-buttons");
  const playBtn = buttons.querySelector("#play-button");

  if (ruleIndex <= 0) {
    ruleIndex = 0;
    prevBtn.style.display = "none";
  }
}
```

```

        buttons.style.justifyContent = "flex-end";
    } else {

        prevBtn.style.display = "block";

        buttons.style.justifyContent = "space-between";
    }

    if (ruleIndex >= Object.keys(rulesObj).length - 1) {

        ruleIndex = Object.keys(rulesObj).length - 1;

        nextBtns.style.display = "none";

        playBtn.style.display = "flex";
    } else {

        playBtn.style.display = "none";

        nextBtns.style.display = "flex";
    }

    buttons.remove();

    ruleBody.innerHTML = element.ruleText;

    let ruleHeader = document.querySelector(".rule-header").querySelector("p");

    ruleHeader.innerText = element.ruleName;

    ruleBody.parentElement.appendChild(buttons);
}

function redirectTo(url) {

    window.location = url;

}

```

```

</script>

<?php include '../components/footer.php'; ?>

    user_info.php
<?php

session_start();

$title = "Δεδομένα Χρήστη";

$stylesheets =

    '<link rel="stylesheet" href="../css/styles.css">'

    '<link rel="stylesheet" href="../css/user_info.css">';

include '../header.php';

if (!isset($_SESSION["isAdmin"])) {

    echo "<script>window.location = '' . str_replace(\"\\n\", \"\", $baseUrl) . \"\"</script>";

    exit();

}

if (!isset($_GET['user']) || $_GET['user'] == '') {

    echo "<script>window.location = '' . str_replace(\"\\n\", \"\", $baseUrl) . \"\"</script>";

    exit();

}

?>

<body>

```



```

<?php include '../components/navbar.php'; ?>

<div class="page-content">

    <div class="user-info">

        <div class="user-name-email">

            <p class="user-name"></p> - <p class="user-email"></p>

        </div>

        <p class="user-education"></p>

    </div>

    <div class="evaluations">

        <ul class="evaluations-list">

        </ul>

    </div>

</div>

<script>

const searchParams = new URLSearchParams();

const url = new URL(window.location);

const params = new URLSearchParams(url.search);

if (!params.get("user")) {

    console.log("Param not found!");

    window.location = `/${baseURL}`;

}

searchParams.append("submit", "submit");

```

```

searchParams.append("user", params.get("user"));

fetch(`/${baseUrl}/includes/fetchUserData.php`, {
  method: "POST",
  body: searchParams
}).then((res) => {
  return res.text();
}).then((text) => {
  if (text.includes("error")) {
    const error = text.split("=")[1];
    switch (error) {
      case "none":
        window.location = `/${baseUrl}`;
        break;
      case "noUsersFound":
        sweetAlertError({
          text: "Δεν βρέθηκε ο συγκεκριμένος χρήστης.",
          redirect: `/${baseUrl}/routes/admin_panel.php`
        })
        break;
      default:
        break;
    }
  } else {

```

```

const jsonResponse = JSON.parse(text);

console.log(jsonResponse);

document.querySelector("p.user-name").innerText =
    `${jsonResponse.user.userFirstName} ${jsonResponse.user.userLastName}`;

document.querySelector("p.user-email").innerText =
    `${jsonResponse.user.userEmail}`;

document.querySelector("p.user-education").innerText =
    `${convertEducationToReadable(jsonResponse.user.userEducation)}`;

const evalList = document.querySelector("ul.evaluations-list");

jsonResponse.user.evaluations.forEach(evaluation => {

    const evalLI = document.createElement("li");

    evalLI.classList.add("evaluation");

    const evalHeader = document.createElement("h3")

    evalHeader.classList.add("evaluation-title");

    evalHeader.innerText = `Αξιολόγηση #${evaluation.evaluation.evaluationID}`;

    evalLI.appendChild(evalHeader);

    const evalGradesUL = document.createElement("ul");

    evalGradesUL.classList.add("evaluation-grades");

```

```

evaluation.grades.forEach((grade, index) => {

    const gradeElem = document.createElement("li");

    gradeElem.classList.add("evaluation-item");

    const gradeInfo = document.createElement("div");

    gradeInfo.classList.add("grade-info");

    gradeElem.appendChild(gradeInfo);

    const gradeHeader = document.createElement("h4");

    gradeHeader.innerText = `Βαθμός ${index}`;

    gradeInfo.appendChild(gradeHeader);


    const gradeDate = document.createElement("p");

    const successRatio = document.createElement("p");

    const completionTime = document.createElement("p");

    const studyTime = document.createElement("p");

    gradeDate.classList.add("grade-date");

    successRatio.classList.add("success-ratio");

    completionTime.classList.add("completion-time");

    studyTime.classList.add("study-time");

    gradeDate.innerText = "Ημερομηνία: ";

    successRatio.innerText = "Ποσοστό Επιτυχίας: ";

    completionTime.innerText = "Χρόνος Ολοκλήρωσης: ";

    studyTime.innerText = "Χρόνος Μελέτης: ";

```

```

const valSpan0 = document.createElement("span");
valSpan0.classList.add("value");
valSpan0.innerText = `${grade.gradeDate}`;
const valSpan1 = document.createElement("span");
valSpan1.classList.add("value");
valSpan1.innerText = `${grade.successRatio * 100}%`;
const valSpan2 = document.createElement("span");
valSpan2.classList.add("value");
valSpan2.innerText = `${grade.answerTime}s`;
const valSpan3 = document.createElement("span");
valSpan3.classList.add("value");
valSpan3.innerText = `${grade.studyTime}s`;
gradeDate.appendChild(valSpan0);
successRatio.appendChild(valSpan1);
completionTime.appendChild(valSpan2);
studyTime.appendChild(valSpan3);

gradeInfo.appendChild(gradeDate);
gradeInfo.appendChild(successRatio);
gradeInfo.appendChild(completionTime);
gradeInfo.appendChild(studyTime);

evalGradesUL.appendChild(gradeElem);
})

```

```

evalLI.appendChild(evalGradesUL);

evalList.appendChild(evalLI);
});

}

}).catch((error) => {

    console.error(`${error}`);

});

</script>

<?php include '../components/footer.php'; ?>

Φάκελος story
Story_lines.json
{

    "0": "Γειά! Το ονομά σου είναι <name>. Σωστά; Είμαι η Ξένια!",

    "1": "Ο Μπρόνκο, το σκυλάκι μου, έχει στομαχόπονο. Ο αυτόματος πωλητής έχει το αντίδοτο.",

    "2": "Το πρόβλημα είναι ότι για να μου το δώσει, πρέπει να ολοκληρώσω κάποιες δοκιμασίες ορθογραφίας, αλλά δεν είμαι και τόσο καλή.",

    "3": "Θα με βοηθήσεις;",

    "4": {

        "0": "Τέλεια! Ας παίξουμε λοιπόν.",

        "1": "Κρίμα. Θα συνεχίσω να προσπαθώ. Αν θελήσεις να βοηθήσεις, θα είμαι εδώ."

    }

}

```

```
}
```

Λοιπά αρχεία

authentication.php

```
<?php
```

```
session_start();
```

```
if (isset($_SESSION['logged'])) {
```

```
    header("location: $baseUrl/");
```

```
    exit();
```

```
}
```

```
$title = "Σύνδεση";
```

```
$stylesheets =
```

```
    '<link rel="stylesheet" href="css/styles.css">
```

```
    <link rel="stylesheet" href="css/login.css">';
```

```
include './header.php';
```

```
?>
```

```
<body>
```

```
    <div class="backdrop"></div>
```

```
    <div class="page-content">
```

```
        <div class="login-form">
```

```
            <div class="form-header">
```

```
                <p>Ορθοπαίγνια</p>
```

```
            </div>
```

```
            <div class="form-selector">
```

```

<a href="#" onclick="toggleForm('login')">Σύνδεση</a>

<a href="#" onclick="toggleForm('register')">Εγγραφή</a>

</div>

<div class="form-inputs login">

  <form method="POST">

    <div class="input-container">

      <label for="username">

        <ion-icon name="person"></ion-icon>

      </label>

      <input type="text" name="username" id="username-input login"
placeholder="Όνομα Χρήστη" />

    </div>

    <div class="input-container">

      <label for="password">

        <ion-icon name="key"></ion-icon>

      </label>

      <input type="password" name="password" id="password-input login"
placeholder="Κωδικός Πρόσβασης" />

    </div>

    <div class="form-submit">

      <button value="submit" type="submit" name="submit">

        Σύνδεση

      </button>

    </div>

  </form>

```



```

</div>

<div class="form-inputs register">

  <form method="POST">

    <div class="input-sections">

      <div class="authentication-data">

        <div class="section-header">

          <p class="header">Στοιχεία χρήστη</p>

        </div>

        <div class="input-container">

          <label for="username">

            <ion-icon name="person"></ion-icon>

          </label>

          <input type="text" name="username" id="username-input register"

            placeholder="Όνομα Χρήστη" />

        </div>

        <div class="input-container">

          <label for="email">

            <ion-icon name="at"></ion-icon>

          </label>

          <input type="email" name="email" id="email-input"

placeholder="Ηλ. Ταχυδρομείο" />

        </div>

        <div class="input-container">

          <label for="password">

            <ion-icon name="key"></ion-icon>

```

```

        </label>

        <input type="password" name="password" id="password-input
register"

        placeholder="Κωδικός Πρόσβασης" />

    </div>

    <div class="input-container">

        <label for="password">

            <ion-icon name="key"></ion-icon>

        </label>

        <input type="password" name="repeat-password" id="repeat-
password-input"

        placeholder="Επανάληψη Κωδικού Πρόσβασης" />

    </div>

</div>

<div class="personal-data">

    <div class="section-header">

        <p class="header">Στοιχεία μαθητή</p>

    </div>

    <div class="input-container">

        <label for="student-name">

            <ion-icon name="person"></ion-icon>

        </label>

        <input type="text" name="student-name" id="student-name-input
register"

        placeholder="Όνομα Μαθητή" />

```

```

</div>

<div class="input-container">

  <label for="student-lastname">

    <ion-icon name="person"></ion-icon>

  </label>

  <input type="text" name="student-lastname" id="student-lastname-
input register"

    placeholder="Επώνυμο Μαθητή" />

</div>

<div class="input-container">

  <label for="student-grade">

    <ion-icon name="school"></ion-icon>

  </label>

  <select name="student-grade" id="student-grade">

    <option value="none" disabled selected>Επιλέξτε τάξη</option>

    <option value="3">Γ' τάξη</option>

    <option value="4">Δ' τάξη</option>

    <option value="5">Ε' τάξη</option>

    <option value="6">ΣΤ' τάξη</option>

    <option value="other">Δευτεροβάθμια εκπαίδευση</option>

  </select>

</div>

</div>

<div class="form-submit">

```

```

        <button value="submit" type="submit" name="submit">
            Εγγραφή
        </button>
    </div>
</form>

</div>

<div class="form-extras">
</div>
</div>
</div>
<script>
let registerForm = document.querySelector(".form-inputs.register");
let loginForm = document.querySelector(".form-inputs.login");

registerForm.addEventListener(
    "submit",
    (event) => {
        event.preventDefault();
        submitRegister();
    },
    true
);

```

```
loginForm.addEventListener(  
  "submit",  
  (event) => {  
    event.preventDefault();  
    submitLogin();  
  },  
  true  
);
```

```
let formExtras = document.querySelector(".form-extras");
```

```
let loginLink = document.querySelectorAll(".form-selector a")[0];
```

```
let registerLink = document.querySelectorAll(".form-selector a")[1];
```

```
registerForm.style.display = "none";
```

```
loginForm.style.display = "flex";
```

```
loginLink.classList.add("active");
```

```
registerLink.classList.remove("active");
```

```
function toggleForm(mode) {
```

```
  if (mode == "login") {
```

```
    registerForm.style.display = "none";
```

```
loginForm.style.display = "flex";

formExtras.style.display = "block";

document.title = `Σύνδεση - ${document.title.split(" - ")[1]}`;

loginLink.classList.add("active");
registerLink.classList.remove("active");
} else {
    registerForm.style.display = "flex";
    loginForm.style.display = "none";

    formExtras.style.display = "none";

    document.title = `Εγγραφή - ${document.title.split(" - ")[1]}`;

    loginLink.classList.remove("active");
    registerLink.classList.add("active");
}
}

</script>

<script src="js/formHandler.js"></script>

</body>
```

```

</html>

    header.php
<?php
if (!file("variables.env")) {

    $baseUrl = file("../variables.env")[0];

} else {

    $baseUrl = file("variables.env")[0];

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <?php

    echo $stylesheets;

    ?>

    <!-- Include IonIcon Icons -->

    <script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>

    <script nomodule
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>

```

```

<!-- Include Flaticon Icons -->

<link rel="stylesheet" href="https://cdn-uicons.flaticon.com/uicons-regular-
rounded/css/uicons-regular-rounded.css" />

<link rel="stylesheet" href="https://cdn-uicons.flaticon.com/uicons-bold-
rounded/css/uicons-bold-rounded.css" />

<link rel="stylesheet" href="https://cdn-uicons.flaticon.com/uicons-bold-
straight/css/uicons-bold-straight.css" />

<link rel="stylesheet" href="https://cdn-uicons.flaticon.com/uicons-solid-
rounded/css/uicons-solid-rounded.css" />

<link rel="stylesheet" href="https://cdn-uicons.flaticon.com/uicons-regular-
straight/css/uicons-regular-straight.css" />

<link rel="stylesheet" href="https://cdn-uicons.flaticon.com/uicons-solid-
straight/css/uicons-solid-straight.css" />

<link rel="stylesheet" href="https://cdn-uicons.flaticon.com/uicons-brands/css/uicons-
brands.css" />


<!-- Include the SweetAlert2 Library -->

<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<script src="<?php echo $baseUrl ?>/js/sweetalertFunctions.js"></script>

<!-- Include ResponsiveVoice -->

<script
src="https://code.responsivevoice.org/responsivevoice.js?key=9EndP8Cp"></script>

<link rel="icon" type="image/x-icon" href="<?php echo $baseUrl
?>/images/favicon.ico">

<title>

<?php echo $title ?>

```



```

- Ορθοπαίγνια</title>

<script>

    let baseURL = window.location.pathname.split("/")[1].replace(/(?:\r\n|\r|\n)/g, "");

</script>

</head>

    index.php
<?php
session_start();

if (!isset($_SESSION['logged'])) {

    header("location: authentication.php");

}

$title = "Αρχική";

$stylesheets =

    '<link rel="stylesheet" href="css/styles.css">'

    '<link rel="stylesheet" href="css/index.css">';

include './header.php';

?>

<body>

    <?php include 'components/navbar.php'; ?>

    <div class="page-content">

        <div class="dialog fade-in">

            <p id="story-lines">

```

</p>

<div class="dialog-buttons">

<div class="next-btns">

<button class="inverse" id="skip-dialog" onclick="skipDialog()">

Παράβλεψη

<ion-icon name="play-skip-forward"></ion-icon>

</button>

<button class="blue" id="next-dialog" onclick="changeStoryLine(1)">

Επόμενο

<ion-icon name="chevron-forward"></ion-icon>

</button>

</div>

<button class="blue" id="prev-dialog" onclick="changeStoryLine(-1)">

<ion-icon name="caret-back"></ion-icon>

Προηγούμενο

</button>

</div>

<div class="answer-buttons">

<button class="green" id="yes-dialog" onclick="answerDialog('y')">Ναι

<ion-icon name="thumbs-up"></ion-icon>

</button>

<button class="red" id="no-dialog" onclick="answerDialog('n')">Όχι

<ion-icon name="thumbs-down"></ion-icon>

</button>

```

</div>

<div class="play-buttons">

    <button class="green" id="prev-dialog" onclick="window.location =
`/${baseUrl}/routes/quiz.php`">Παίξε

        <ion-icon name="caret-forward"></ion-icon>

    </button>

</div>

</div>

<div class="story-images">

    <div class="bronco fade-in">

    </div>

    <div class="xenia fade-in">

    </div>

</div>

</div>

<script>

let storyLines;

let storyLineIndex = 0;


let prevDialogBtn = document.querySelector("#prev-dialog");

prevDialogBtn.style.display = "none";

```

```
let nextDialogBtn = document.querySelector("#next-dialog");
```

```
let dialogText = document.querySelector("#story-lines");
```

```
let dialogBtns = document.querySelector(".dialog-buttons");
```

```
let answerBtns = document.querySelector(".answer-buttons");
```

```
let playButton = document.querySelector(".play-buttons");
```

```
playButton.remove();
```

```
let canPressButton = true;
```

```
let stopFunc = false;
```

```
const xeniaImg = document.querySelector(".xenia").querySelector("img");
```

```
const broncoImg = document.querySelector(".bronco").querySelector("img");
```

```
fetch(`${baseUrl}/story/story_lines.json`)
```

```
  .then((response) => {
```

```
    return response.json();
```

```
  })
```

```
  .then((json) => {
```

```
    storyLines = json;
```

```
    changeStoryLine(0);
```

```
  });
```

```

function recursiveObjectPrint(obj) {

    let storyLinesElement = document.querySelector(".story-lines");

    for (var key in obj) {

        var value = obj[key];

        if (!!value && value.constructor === Object) {

            recursiveObjectPrint(value);

        } else {

            typeWriter(value, storyLinesElement);

        }

    }

}

```

```

function typeWriter(str, element) {

    canPressButton = false;

    let i = 0;

    const interval = setInterval(typeWriterIteration, 25);

```

```

function typeWriterIteration() {

    if (i >= str.length) {

        clearInterval(interval);

        canPressButton = true;

    } else {

        i++;

        if (element != null) {

```

```

        element.innerHTML = str.slice(0, i);
    }
}
}
}

```

```

function changeStoryLine(index, answer) {
    if (canPressButton) {
        if (
            storyLineIndex + index < 0 ||
            storyLineIndex + index > Object.keys(storyLines).length - 1
        ) {
            console.error("StoryLineIndex is out of bounds!");
            return;
        } else {
            storyLineIndex += index;
        }
        if (storyLineIndex > 0) {
            prevDialogBtn.style.display = "flex";
        } else {
            prevDialogBtn.style.display = "none";
        }

        if (storyLineIndex < Object.keys(storyLines).length - 1) {

```

```

    nextDialogBtn.style.display = "flex";

    nextDialogBtn.parentElement.style.flexDirection = "row-reverse";

} else {

    nextDialogBtn.style.display = "none";

    nextDialogBtn.parentElement.style.flexDirection = "row";

}

if (typeof storyLines[storyLineIndex + 1] === "object") {

    // WE HAVE DIALOG DEPENDING ON THE ANSWER (Y OR N)

    // MAKE Y OR N BUTTONS VISIBLE, HIDE OTHER BUTTONS

    answerBtns.style.display = "flex";

    dialogBtns.style.display = "none";

    typeWriter(

        storyLines[storyLineIndex].replaceAll(

            "<name>",

            "<?php echo $_SESSION['firstname'] ?>"

        ),

        dialogText

    );

    responsiveVoice.speak(`${storyLines[storyLineIndex].replaceAll(

        "<name>",

        "<?php echo $_SESSION['firstname'] ?>"

    )}` , "Greek Female");

} else {

    answerBtns.style.display = "none";

```

```

dialogBtns.style.display = "flex";

if (answer === true) {

    if (!storyLines[storyLineIndex + 1]) {

        // ENABLE PLAY BUTTON ON THE LAST DIALOG

        answerBtns.parentElement.appendChild(playButton);

        // DISABLE ALL OTHER BUTTONS

        answerBtns.remove();

        dialogBtns.remove();

    } else {

        playButton.remove();

    }

    typeWriter(

        storyLines[storyLineIndex][0].replaceAll(

            "<name>",

            "<?php echo $_SESSION['firstname'] ?>"

        ),

        dialogText

    );

    responsiveVoice.speak(`${storyLines[storyLineIndex][0].replaceAll(

        "<name>",

        "<?php echo $_SESSION['firstname'] ?>"

    )}` , "Greek Female");

} else if (answer === false) {

    nextDialogBtn.parentElement.style.display = "none";

```



```

dialogBtns.style.flexDirection = "row";

typeWriter(

    storyLines[storyLineIndex][1].replaceAll(

        "<name>",

        "<?php echo $_SESSION['firstname'] ?>"

    ),

    dialogText

);

responsiveVoice.speak(`${storyLines[storyLineIndex][1].replaceAll(

    "<name>",

    "<?php echo $_SESSION['firstname'] ?>"

)}}`, "Greek Female");

} else {

    typeWriter(

        storyLines[storyLineIndex].replaceAll(

            "<name>",

            "<?php echo $_SESSION['firstname'] ?>"

        ),

        dialogText

    );

    responsiveVoice.speak(`${storyLines[storyLineIndex].replaceAll(

        "<name>",

        "<?php echo $_SESSION['firstname'] ?>"

```

```

        )}` , "Greek Female");

    }

}

}

if (storyLines[storyLineIndex + 1]) {

    playButton.remove();

}

switch (storyLineIndex) {

    case 0:

        xeniaImg.src = `/${baseUrl}/svg/Xenia Greet.svg`;

        break;

    case 1:

        xeniaImg.src = `/${baseUrl}/svg/Xenia Troubled.svg`;

        break;

    case 2:

        xeniaImg.src = `/${baseUrl}/svg/Xenia Sad.svg`;

        break;

    case 3:

        xeniaImg.src = `/${baseUrl}/svg/Xenia Troubled.svg`;

        break;

    case 4:

```

```

    if (answer) {

        xeniaImg.src = `/${baseUrl}/svg/Xenia Happy.svg`;

    } else {

        xeniaImg.src = `/${baseUrl}/svg/Xenia Sad.svg`;

    }

    break;

default:

    xeniaImg.src = `/${baseUrl}/svg/Xenia Greet.svg`;

    break;

}

}

```

```

function answerDialog(ans) {

    let answer;

    switch (ans) {

        case 'y':

            console.log("Answered YES");

            answer = true;

            break;

        case 'n':

            console.log("Answered NO");

            answer = false;

            break;

        default:

```

```

        console.error("Invalid answer!");

        break;
    }

    changeStoryLine(1, answer);
}

function skipDialog() {
    if (canPressButton) {
        answerBtns.parentElement.appendChild(playButton);

        // DISABLE ALL OTHER BUTTONS

        answerBtns.remove();

        dialogBtns.remove();

        xeniaImg.src = `${baseUrl}/svg/Xenia Happy.svg`;

        typeWriter(
            storyLines[Object.keys(storyLines).length - 1][0].replaceAll(
                "<name>",
                "<?php echo $_SESSION['firstname'] ?>"
            ),
            dialogText
        );

        responsiveVoice.speak(`${storyLines[Object.keys(storyLines).length -
1][0].replaceAll(
                "<name>",

```

```
        "<?php echo $_SESSION['firstname'] ?>"
    )} ` , "Greek Female");

    }

}

</script>

<script src="js/formHandler.js"></script>

<?php include 'components/footer.php'; ?>
```

Αναφορές – Πηγές

1. Παπαγεωργίου, Δ. (2019). Μαθησιακές δυσκολίες σε Δημοτικό και Γυμνάσιο: Δυσγραφία-Δυσορθογραφία.
2. Berk, R. A. (1983). Toward a definition of learning disabilities: Progress or regress?. *Education and Treatment of Children*, 6(3), 285-310.
3. National Association of Special Education Teachers: Introduction to Learning Disabilities. (2011, February 3). <https://www.naset.org/?id=2522> (τελευταία πρόσβαση 20. 10. 2022)
4. Mercer, C. D., Jordan, L., Allsopp, D. H., & Mercer, A. R. (1996). Learning disabilities definitions and criteria used by state education departments. *Learning Disability Quarterly*, 19(4), 217-232.
5. Swanson, H. L. (1991). Operational definitions and learning disabilities: An overview. *Learning Disability Quarterly*, 14(4), 242-254.
6. Woodcock, S. (2013). Trainee teachers' attitudes towards students with specific learning disabilities. *Australian Journal of Teacher Education (Online)*, 38(8), 16-29.
7. Shaywitz, S. E. (1998). Dyslexia. *New England Journal of Medicine*, 338(5), 307-312.
8. Wikipedia contributors. (2021, May 29). Δυσγραφία. Βικιπαίδεια. <https://el.wikipedia.org/wiki/%CE%94%CF%85%CF%83%CE%B3%CF%81%CE%B1%CF%86%CE%AF%CE%B1> (τελευταία επίσκεψη 15 Οκτ, 2022)
9. Kuure, L., Kuure, O., Sandbäck, A. M., & Yliherva, A. (1992). Dysorthographia and bilingualism: Case studies of Swedish as a second language. *Scandinavian Journal of Logopedics and Phoniatrics*, 17(3-4), 191-201.
10. Βλάχου, Σ., Λίλα, Α., & Μυρσιάδη, Μ. (2019). Πεποιθήσεις και στάσεις καθηγητών τριτοβάθμιας εκπαίδευσης για φοιτητές με μαθησιακές δυσκολίες.
11. Zygouris, N.C., Vlachos, F., Dadaliaris, A.N., Oikonomou, P., Stamoulis, G.I., Vavougiotis, D., Nerantzaki, E. and Striftou, A., 2017. A neuropsychological approach of developmental dyscalculia and a screening test via a web application. *International Journal of Engineering Pedagogy (iJEP)*, 7(4), pp.51-65.
12. Zygouris, N. C., Avramidis, E., Karapetsas, A. V., & Stamoulis, G. I. (2018). Differences in dyslexic students before and after a remediation program: A

- clinical neuropsychological and event related potential study. *Applied Neuropsychology: Child*, 7(3), 235-244.
13. Giordano, D., & Maiorana, F. (2015, May). A Mobile Web Game Approach for Improving Dysgraphia. In *CSEDU* (1) (pp. 328-333).
 14. Gkeka, E., & Drigas, A. (2022). Ict's and Dysgraphia. *Technium Social Sciences Journal*, 31, 228-240.
 15. Understanding AES 256 Encryption. (2022, August 22). N-able. <https://www.n-able.com/blog/aes-256-encryption-algorithm> (τελευταία επίσκεψη 3 Οκτ, 2022)
 16. Low Power Implementation of Sbox for AES on FPGA - Scientific Figure on ResearchGate. Διαθέσιμο στο: https://www.researchgate.net/figure/AES-Process-4-This-square-outline-is-standard-for-AES-details-It-contains-various_fig1_325545251 (τελευταία επίσκεψη 29 Οκτ, 2022)
 17. Wikipedia contributors. (2001, November 20). Linux. Wikipedia. <https://en.wikipedia.org/wiki/Linux> (τελευταία επίσκεψη 31 Οκτ, 2022)
 18. Lawton, G. (2005). LAMP lights enterprise development efforts. *Computer*, 38(9), 18-20.
 19. Documentation Group. (χ.χ.). Welcome! - The Apache HTTP Server Project. Ανακτήθηκε 31 Οκτωβρίου, 2022, από <https://httpd.apache.org/>
 20. Suehring, S. (2002). *Mysql Bible w/Cd* (Pap/Cdr). John Wiley & Sons.
 21. DuBois, P. (2008). *MySQL*. Pearson Education.
 22. Lerdorf, R., & Tatroe, K. (2002). *Programming PHP* (1st ed.). O'Reilly Media.
 23. Welling, L., & Thomson, L. (2003). *PHP and MySQL Web Development*. Sams.
 24. McFarland, D. S. (2012). *CSS3: The Missing Manual*. Van Duuren Media.
 25. *Cascading Style Sheets*. (χ.χ.). Ανακτήθηκε 1 Νοεμβρίου, 2022, από <https://www.w3.org/Style/CSS/Overview.en.html>
 26. Hoy, M. B. (2011). HTML5: a new standard for the Web. *Medical reference services quarterly*, 30(1), 50-55.
 27. Kantor, I. (χ.χ.). An Introduction to JavaScript. Ανακτήθηκε 29 Οκτωβρίου, 2022, από <https://javascript.info/intro>
 28. Γιαβρίμης, Π., Παπάνης, Ε., Νεοφώτιστος, Β., & Βαλκάνος, Ε. (2010). Απόψεις εκπαιδευτικών για την εφαρμογή των ΤΠΕ στην εκπαίδευση. Πρακτικά Εργασιών 7ου Πανελλήνιου Συνεδρίου μεΔ ιεθνή Συμμετοχή «Οι ΤΠΕ στην Εκπαίδευση», τόμος II, 633-640.

29. Hoyles, C. (1992). Illuminations and reflections: Teachers, methodologies and mathematics. In Proceedings of the 16 th Conference of the International Group for the Psychology of Mathematics Education (PME). Durham, NH., USA: University of New Hampshire (Vol. 3, pp. 263-286).
30. Dictionary.com: English Words - Apps on Google Play. (χ.χ.). Ανακτήθηκε 15 Νοεμβρίου, 2022, από <https://play.google.com/store/apps/details?id=com.dictionary>
31. Support, D. (2021c, January 19). About. Dictionary.com. <https://www.dictionary.com/e/about/>
32. About. (χ.χ.). ModMath. <http://www.modmath.com/about>
33. Striftou, A., Zygouris, N. C., Stamoulis, G. I., & Vavougiou, D. (2020, September). Work in Progress: Web-Delivered Reading Improvement Battery of Tasks. In International Conference on Interactive Collaborative Learning (pp. 643-654).
34. Zygouris, N. C., Vlachos, F., Dadaliaris, A. N., Oikonomou, P., Stamoulis, G. I., Vavougiou, D., Nerantzaki, E., & Striftou, A. (2017). A Neuropsychological Approach of Developmental Dyscalculia and a Screening Test Via a Web Application. *International Journal of Engineering Pedagogy (IJEPE)*, 7(4), 51. <https://doi.org/10.3991/ijep.v7i4.7434>
35. Σκαλούμπακας, Χ., Πρωτόπαπας, Α., & Νικολόπουλος, Δ. (2003). Παρουσίαση μιας κλίμακας μαθησιακής αξιολόγησης για την εξέταση των μαθησιακών δυσκολιών και στοιχεία από την χορήγησή της σε μαθητές πρώτης γυμνασίου. Στο Μ. Γλύκας & Γ. Καλομοίρης (επιμ.) Διαταραχές Επικοινωνίας και Λόγου: Πρόληψη, Έρευνα, Παρέμβαση και Νέες Τεχνολογίες στην Υγεία, 55-70.
36. Πρωτόπαπας, Α., Σκαλούμπακας, Χ., & Νικολόπουλος, Δ. (2003). Αυτόματη ανίχνευση μαθησιακών δυσκολιών με το λογισμικό εΜαΔύς: εγκυρότητα και προοπτικές. Υπό δημοσίευση στα πρακτικά του 9ου Συνεδρίου του Πανελληνίου Συλλόγου Λογοπεδικών, Αθήνα, 17-18.