



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΕΞΕΤΑΣΕΩΝ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

ΑΘΑΝΑΣΙΟΥ ΕΛΕΝΗ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιάλλας Γρηγόριος
Καθηγητής

Λαμία 2025



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΤΗΝ ΔΙΑΧΕΙΡΙΣΗ ΕΞΕΤΑΣΕΩΝ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

ΑΘΑΝΑΣΙΟΥ ΕΛΕΝΗ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιάλλας Γρηγόριος
Καθηγητής

Λαμία 2025



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

DEVELOPMENT OF A WEB APPLICATION FOR CREATION AND MANAGEMENT OF MULTIPLE CHOICE EXAMS

ATHANASIOU ELENI

FINAL THESIS

ADVISOR

Tziallas Grigorios
Professor

Lamia 2025

«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 επεξεργάσθηκαν, 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: 27/06/2025

Η Δηλούσα

(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.»

Η παρούσα πτυχιακή εργασία πραγματεύεται τον σχεδιασμό και την υλοποίηση μιας διαδικτυακής εφαρμογής με σκοπό τη δημιουργία, διαχείριση και αξιολόγηση ερωτήσεων πολλαπλής επιλογής. Το έργο αναπτύχθηκε με στόχο να καλύψει τις εκπαιδευτικές ανάγκες οργανισμών και μεμονωμένων χρηστών, προσφέροντας ένα ευέλικτο και επεκτάσιμο εργαλείο διαχείρισης quiz. Αφετηρία της μελέτης αποτέλεσε η ανάλυση υπάρχουσών λύσεων όπως τα Microsoft Forms, Google Forms και Kahoot!, όπου αναδείχθηκαν τόσο τα πλεονεκτήματά τους όσο και οι περιορισμοί τους, ειδικά σε ό,τι αφορά την παραμετροποίηση, την ασφάλεια και τη δυνατότητα ενσωμάτωσης σε προσαρμοσμένα περιβάλλοντα.

Η υλοποίηση της εφαρμογής πραγματοποιήθηκε με χρήση της τεχνολογικής στοίβας MERN (MongoDB, Express.js, React.js και Node.js), επιτρέποντας την κατασκευή μιας μοντέρνας Single Page Application. Το frontend της εφαρμογής αναπτύχθηκε με React για την εξασφάλιση δυναμικής εμπειρίας χρήστη, ενώ το backend διαχειρίζεται την επιχειρησιακή λογική και τη σύνδεση με τη βάση δεδομένων MongoDB. Οι βασικές λειτουργίες περιλαμβάνουν δημιουργία και επεξεργασία quiz, οργάνωση ερωτήσεων, διαχείριση ομάδων χρηστών, ανάθεση quiz, υποβολή απαντήσεων και αυτόματη αξιολόγηση.

Ιδιαίτερη βαρύτητα δόθηκε στη διαχείριση ρόλων (διαχειριστής/χρήστης), στην ασφάλεια πρόσβασης και στην υποστήριξη της εμπειρίας σε φορητές συσκευές. Τέλος, καταγράφονται τα συμπεράσματα της υλοποίησης, οι τεχνικές προκλήσεις που προέκυψαν, καθώς και οι δυνατότητες μελλοντικής επέκτασης, όπως η υποστήριξη οργανισμών, η συμμετοχή μέσω κωδικών και η εισαγωγή νέων τύπων ερωτήσεων. Μέσα από την εργασία αυτή αναδεικνύεται η αξία της δημιουργίας εξατομικευμένων εφαρμογών που ανταποκρίνονται σε πραγματικές ανάγκες με τρόπο ευέλικτο, ασφαλή και οικονομικά βιώσιμο.

ABSTRACT

This thesis presents the design and development of a web-based application for the creation, management, and evaluation of multiple-choice questions. The project aims to address educational needs for both organizations and individual users by offering a flexible and extensible quiz management tool. The study begins with an analysis of existing platforms such as Microsoft Forms, Google Forms, and Kahoot!, highlighting their advantages as well as their limitations, particularly regarding customization, data security, and integration into specialized environments.

The application was developed using the MERN technology stack (MongoDB, Express.js, React.js, and Node.js), enabling the implementation of a modern Single Page Application. The frontend, built with React, ensures a dynamic user experience, while the backend manages business logic and interaction with the MongoDB database. Core features include quiz creation and editing, question organization, user group management, quiz assignment, answer submission, and automatic evaluation.

Significant emphasis was placed on role-based access (admin/user), secure session management, and mobile-friendly interface design. The thesis also discusses the outcomes of the implementation, the technical challenges encountered, and the potential for future enhancements such as organizational management support, code-based quiz participation, and the introduction of additional question types. This work highlights the value of building tailored applications that respond effectively to real-world needs in a secure, user-friendly, and cost-efficient manner.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	0
ABSTRACT	1
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	2
ΕΙΣΑΓΩΓΗ	5
ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ	6
ΚΕΦΑΛΑΙΟ 1: ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΩΤΗΣΕΩΝ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ	8
ΥΠΑΡΧΟΥΣΕΣ ΛΥΣΕΙΣ.....	8
ΥΠΑΡΧΟΥΣΕΣ ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΦΑΡΜΟΓΕΣ	8
MICROSOFT FORMS	8
GOOGLE FORMS.....	8
ΚΑΗΟΟΤ!	9
ΣΥΓΚΡΙΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ	10
MICROSOFT FORMS	10
GOOGLE FORMS.....	10
ΚΑΗΟΟΤ!	10
ΕΦΑΡΜΟΓΗ ΣΤΟ ΠΛΑΙΣΙΟ ΔΙΕΝΕΡΓΕΙΑΣ ΕΞΕΤΑΣΕΩΝ	11
ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΩΝ ΕΤΟΙΜΩΝ ΛΥΣΕΩΝ	11
ΠΛΕΟΝΕΚΤΗΜΑΤΑ	11
ΜΕΙΟΝΕΚΤΗΜΑΤΑ.....	12
ΠΡΟΣΑΡΜΟΣΜΕΝΕΣ ΛΥΣΕΙΣ	13
ΚΕΦΑΛΑΙΟ 2: ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ ΓΙΑ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ.....	14
ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ	14
ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΚΑΙ ΕΞΕΛΙΞΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ.....	14
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ	14
ΠΡΟΚΛΗΣΕΙΣ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ	15
ΤΕΧΝΟΛΟΓΙΕΣ FRONTEND	15
HTML, CSS ΚΑΙ JAVASCRIPT	15
ΒΙΒΛΙΟΘΗΚΕΣ ΚΑΙ FRONTEND FRAMEWORKS.....	15
RESPONSIVE DESIGN ΚΑΙ MOBILE-FIRST ΠΡΟΣΕΓΓΙΣΗ	16
ΔΙΑΧΕΙΡΙΣΗ ΚΑΤΑΣΤΑΣΗΣ ΔΕΔΟΜΕΝΩΝ (STATE MANAGEMENT).....	16
ΤΕΧΝΟΛΟΓΙΕΣ BACKEND	16
ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ BACKEND	16
ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΓΙΑ BACKEND.....	17
BACKEND FRAMEWORKS ΚΑΙ ΠΛΑΙΣΙΑ ΑΝΑΠΤΥΞΗΣ	17
RESTFUL APIs ΚΑΙ JSON	17
ΕΝΔΙΑΜΕΣΟ ΛΟΓΙΣΜΙΚΟ (MIDDLEWARE)	18

SESSION MANAGEMENT ΚΑΙ AUTHENTICATION	18
ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ BACKEND	18
ΚΛΙΜΑΚΑ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ	18
ΑΣΦΑΛΕΙΑ BACKEND.....	19
ΔΙΑΧΕΙΡΙΣΗ ΣΦΑΛΜΑΤΩΝ ΚΑΙ LOGGING	19
DEVOPS ΚΑΙ CI/CD ΓΙΑ BACKEND	19
ΠΡΟΟΠΤΙΚΕΣ ΚΑΙ ΤΑΣΕΙΣ ΣΤΗΝ ΕΞΕΛΙΞΗ ΤΩΝ BACKEND ΤΕΧΝΟΛΟΓΙΩΝ	19
ΤΕΧΝΟΛΟΓΙΕΣ ΑΠΟΘΗΚΕΥΣΗΣ ΔΕΔΟΜΕΝΩΝ	20
ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ	20
ΣΧΕΣΙΑΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ (SQL).....	20
ΜΗ ΣΧΕΣΙΑΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ (NoSQL)	20
ΔΙΑΣΗΜΕΣ ΣΥΓΧΡΟΝΕΣ ΤΕΧΝΟΛΟΓΙΚΕΣ ΣΤΟΙΒΕΣ	21
MERN (MONGODB, EXPRESS, REACT, NODE.JS).....	21
MEAN (MONGODB, EXPRESS, ANGULAR, NODE.JS)	21
.NET STACK (C#, ASP.NET, MICROSOFT SQL SERVER)	21
DJANGO STACK (PYTHON, DJANGO, POSTGRESQL).....	22
RUBY ON RAILS STACK (RUBY, RAILS, POSTGRESQL / MYSQL).....	22

ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ΕΛΕΝΙ FORMS..... 23

ΠΕΡΙΓΡΑΦΗ.....	23
ΕΠΙΛΟΓΗ ΤΕΧΝΟΛΟΓΙΩΝ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	23
ΤΕΧΝΟΛΟΓΙΚΗ ΣΤΟΙΒΑ MERN	23
ΟΦΕΛΗ ΤΩΝ ΕΠΙΛΕΓΜΕΝΩΝ ΤΕΧΝΟΛΟΓΙΩΝ	23
MONGODB	24
EXPRESS.JS	24
REACT.JS.....	24
NODE.JS	25
ΣΤΑΔΙΑ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	25
ΑΝΑΛΥΣΗ.....	25
ΣΧΕΔΙΑΣΗ	25
ΥΛΟΠΟΙΗΣΗ.....	25
ΔΟΚΙΜΕΣ.....	26
ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	26
ΑΡΧΙΤΕΚΤΟΝΙΚΗ	26
FRONTEND	27
BACKEND	29
ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	32

ΚΕΦΑΛΑΙΟ 4: ΕΠΙΔΕΙΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ..... 34

ΟΔΗΓΟΣ ΧΡΗΣΗΣ.....	34
ΔΙΑΧΕΙΡΙΣΤΕΣ.....	34
ΑΠΛΟΙ ΧΡΗΣΤΕΣ	34
ΕΡΩΤΗΜΑΤΟΛΟΓΙΑ.....	34
ΜΕΤΡΑ ΠΡΟΛΗΨΗΣ ΑΝΤΙΓΡΑΦΗΣ	35
ΚΕΝΤΡΙΚΟ ΠΑΝΕΛ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ	35
ΚΕΝΤΡΙΚΟ ΠΑΝΕΛ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΗΣΤΩΝ	35

ΠΑΝΕΛ ΔΙΑΧΕΙΡΙΣΗΣ ΟΜΑΔΩΝ.....	36
ΣΕΛΙΔΑ ΠΡΟΦΙΛ	36
ΠΕΡΙΗΓΗΣΗ ΣΤΗ ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ	36
ΓΕΝΙΚΑ ΓΡΑΦΙΚΑ ΣΤΟΙΧΕΙΑ.....	36
ΌΦΕΙΣ ΚΕΝΤΡΙΚΟΥ ΠΑΝΕΛ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ	37
ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ.....	37
ΔΗΜΙΟΥΡΓΙΑ ΛΟΓΑΡΙΑΣΜΟΥ	38
ΣΥΝΔΕΣΗ ΧΡΗΣΤΗ	38
ΣΥΜΜΕΤΟΧΗ ΣΕ ΟΜΑΔΑ	39
ΑΠΑΝΤΗΣΗ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ	40
ΠΡΟΒΟΛΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΩΣ ΧΡΗΣΤΗΣ	41
ΠΡΟΒΟΛΗ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΛΟΓΑΡΙΑΣΜΟΥ	42
ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΟΜΑΔΩΝ	43
ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ	44
ΠΕΡΙΗΓΗΣΗ ΣΤΟ ΠΑΝΕΛ ΔΙΑΧΕΙΡΙΣΗΣ ΟΜΑΔΩΝ.....	46
ΠΕΡΙΗΓΗΣΗ ΣΤΟ ΠΑΝΕΛ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΗΣΤΩΝ	47
ΠΡΟΒΟΛΗ ΚΑΙ ΕΞΑΓΩΓΗ ΑΠΑΝΤΗΣΕΩΝ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ ΩΣ ΔΙΑΧΕΙΡΙΣΤΗΣ	47
ΣΧΟΛΙΑΣΜΟΣ ΚΑΙ ΧΕΙΡΟΚΙΝΗΤΗ ΒΑΘΜΟΛΟΓΗΣΗ ΕΡΩΤΗΣΕΩΝ	49
ΠΡΟΒΟΛΗ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ	49
 <u>ΚΕΦΑΛΑΙΟ 5: ΕΠΙΛΟΓΟΣ</u>	<u>50</u>
 ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΠΟ ΤΗΝ ΥΛΟΠΟΙΗΣΗ	50
ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	50
 <u>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</u>	<u>52</u>

Η ραγδαία πρόοδος των τεχνολογιών πληροφορικής και επικοινωνιών τα τελευταία χρόνια έχει επηρεάσει καθοριστικά πολλούς τομείς της ανθρώπινης δραστηριότητας, με έναν από τους σημαντικότερους να είναι η εκπαίδευση. Στο πλαίσιο αυτό, η αξιοποίηση διαδικτυακών εφαρμογών για την υποστήριξη της εκπαιδευτικής διαδικασίας αποτελεί πλέον κοινή πρακτική σε όλα τα επίπεδα εκπαίδευσης. Ειδικότερα, η ανάγκη για συστήματα που επιτρέπουν τη δημιουργία, διαχείριση και αξιολόγηση ερωτήσεων και quiz πολλαπλής επιλογής είναι πιο έντονη από ποτέ, καθώς τα εργαλεία αυτά διευκολύνουν την αλληλεπίδραση, την αξιολόγηση γνώσεων, και την ανατροφοδότηση των μαθητών, τόσο σε φυσικά όσο και σε εξ αποστάσεως μαθησιακά περιβάλλοντα.

Η παρούσα πτυχιακή εργασία πραγματεύεται τον σχεδιασμό και την ανάπτυξη μιας διαδικτυακής εφαρμογής για τη δημιουργία και διαχείριση ερωτήσεων πολλαπλής επιλογής, η οποία φέρει τον τίτλο «Eleni Forms». Η εφαρμογή αποσκοπεί στο να καλύψει την ανάγκη των εκπαιδευτικών για ένα εύχρηστο, προσβάσιμο και επεκτάσιμο σύστημα αξιολόγησης, το οποίο να παρέχει δυνατότητες οργάνωσης ερωτήσεων σε quiz, εκχώρησης των quiz σε ομάδες χρηστών, και υποβολής απαντήσεων με δυνατότητα αυτόματης αξιολόγησης.

Αν και στην αγορά υπάρχουν δημοφιλή εργαλεία όπως τα Google Forms, Microsoft Forms και Kahoot!, τα οποία καλύπτουν μέρος αυτής της ανάγκης, παρουσιάζουν περιορισμούς στην παραμετροποίηση, την εξατομίκευση, την επέκταση και τη διαχείριση των δεδομένων. Ειδικά όταν πρόκειται για εκπαιδευτικά ιδρύματα ή μικρούς οργανισμούς με περιορισμένους πόρους, η εξάρτηση από λύσεις τρίτων μπορεί να καταστεί μη βιώσιμη, είτε λόγω κόστους είτε λόγω έλλειψης ευελιξίας. Στο πλαίσιο αυτό, η ανάπτυξη μιας ιδιόκτητης εφαρμογής, προσαρμοσμένης στις ανάγκες του φορέα που τη χρησιμοποιεί, αποτελεί μια αποδοτική και ελκυστική λύση.

Η εργασία αυτή εστιάζει όχι μόνο στην υλοποίηση της εφαρμογής, αλλά και στην κατανόηση των τεχνολογικών και λειτουργικών απαιτήσεων για τη δημιουργία ενός ολοκληρωμένου συστήματος διαχείρισης ερωτήσεων. Συγκεκριμένα, χρησιμοποιείται η τεχνολογική στοίβα MERN (MongoDB, Express.js, React.js, Node.js), η οποία επιτρέπει την κατασκευή μοντέρνων Single Page Applications (SPA) με διαχωρισμό frontend και backend, ασύγχρονη επικοινωνία μέσω REST APIs, και ευελιξία στην αποθήκευση και ανάκτηση δεδομένων.

Στο πλαίσιο της πτυχιακής αναλύεται η υπάρχουσα κατάσταση στον χώρο των εκπαιδευτικών εφαρμογών quiz, παρουσιάζονται οι βασικές τεχνολογίες ανάπτυξης διαδικτυακών εφαρμογών και στη συνέχεια τεκμηριώνεται η αρχιτεκτονική, η σχεδίαση και η υλοποίηση του λογισμικού «Eleni Forms». Τέλος, γίνεται μια ανασκόπηση των δυσκολιών και των μαθημάτων που αποκομίστηκαν από την ανάπτυξη της εφαρμογής, καθώς και προτάσεις για πιθανές επεκτάσεις και αξιοποίηση της πλατφόρμας σε ευρύτερα περιβάλλοντα. Η εργασία στοχεύει να αποτελέσει ένα παράδειγμα πρακτικής εφαρμογής σύγχρονων τεχνολογιών στο πλαίσιο μιας ρεαλιστικής εκπαιδευτικής ανάγκης.

Η ραγδαία πρόοδος των τεχνολογιών πληροφορικής και επικοινωνιών έχει επηρεάσει καθοριστικά τη φύση και τις μεθόδους της εκπαιδευτικής αξιολόγησης, οδηγώντας στη σταδιακή μετάβαση από τα παραδοσιακά έντυπα εργαλεία σε διαδικτυακές πλατφόρμες. Στο πλαίσιο αυτό, η δημιουργία και διαχείριση ερωτήσεων πολλαπλής επιλογής έχει αναδειχθεί ως σημαντικό αντικείμενο μελέτης, με στόχο την υποστήριξη της εκπαιδευτικής διαδικασίας και τη διευκόλυνση της αλληλεπίδρασης μεταξύ διδασκόντων και εκπαιδευομένων.

Η σχετική βιβλιογραφία επικεντρώνεται αφενός στην ανάλυση των υπάρχόντων εργαλείων που προσφέρονται για την ανάπτυξη ερωτηματολογίων, και αφετέρου στη μελέτη των τεχνολογικών προσεγγίσεων που υιοθετούνται για την υλοποίηση τέτοιων λύσεων. Ιδιαίτερο ενδιαφέρον παρουσιάζουν πλατφόρμες που προσφέρουν δυνατότητες αυτόματης αξιολόγησης, αποθήκευσης απαντήσεων και εξαγωγής στατιστικών στοιχείων, οι οποίες βρίσκουν εφαρμογή τόσο στην πρωτοβάθμια και δευτεροβάθμια εκπαίδευση, όσο και σε ακαδημαϊκά και επαγγελματικά περιβάλλοντα.

Ενδεικτικά, εργαλεία όπως το Microsoft Forms, το Google Forms και το Kahoot! (Sofiani & Prasetya, 2024) (Özdemir, 2024) έχουν μελετηθεί εκτενώς για την ευκολία χρήσης τους, την προσβασιμότητα και τις βασικές δυνατότητες που παρέχουν για την εκπαιδευτική αξιολόγηση. Παράλληλα, έχουν αναδειχθεί και τα περιοριστικά τους στοιχεία, όπως η μειωμένη δυνατότητα παραμετροποίησης, η εξάρτηση από εξωτερικούς παρόχους και οι περιορισμοί ως προς την ασφάλεια και την προστασία των δεδομένων των χρηστών. Η σχετική διεθνής βιβλιογραφία υπογραμμίζει ότι, αν και τα έτοιμα συστήματα καλύπτουν σε μεγάλο βαθμό βασικές ανάγκες, συχνά δεν επαρκούν σε περιβάλλοντα με αυξημένες απαιτήσεις ευελιξίας ή σε περιπτώσεις που απαιτείται υψηλός βαθμός προσαρμογής (Bensfiya, Boubih, Janati-Idrissi, & Madrane, 2023) (Ramadhani, Sukyadi, & Purnawarman, 2023).

Αναπόσπαστο μέρος της επισκόπησης αποτελεί και η αναφορά στις τεχνολογικές εξελίξεις που υποστηρίζουν την ανάπτυξη εκπαιδευτικών εφαρμογών. Ιδιαίτερη έμφαση δίνεται στις σύγχρονες αρχιτεκτονικές ανάπτυξης λογισμικού, οι οποίες βασίζονται σε διαχωρισμό μεταξύ frontend και backend, στην αξιοποίηση RESTful υπηρεσιών και στην εφαρμογή μοντέλων όπως οι Single Page Applications (SPA) (Kornienko, Mishina, & Melnikov, 2021). Επιπλέον, η βιβλιογραφία αναδεικνύει τη σημασία της υιοθέτησης τεχνολογικών στοιχείων που προσφέρουν ευελιξία, επεκτασιμότητα και υποστήριξη σε δυναμικά εκπαιδευτικά περιβάλλοντα.

Ένα ακόμα διακριτό ρεύμα στη βιβλιογραφία σχετίζεται με την ανάγκη εύρεσης μιας λύσης η οποία θα μπορεί να ανταπεξέλθει στις απαιτήσεις μιας διαδικασίας αξιολόγησης η οποία να διασφαλίζει την ορθή πραγματοποίησή τους. Μια πρόταση είναι μέσω δημιουργίας προσαρμοσμένων εφαρμογών, οι οποίες να ανταποκρίνονται στις ιδιαίτερες απαιτήσεις εκπαιδευτικών ιδρυμάτων, ομάδων χρηστών ή οργανισμών. Η ανάπτυξη ιδιόκτητων λύσεων θεωρείται, σε ορισμένες περιπτώσεις, περισσότερο βιώσιμη και αποτελεσματική, καθώς επιτρέπει τον έλεγχο επί των λειτουργιών, της ασφάλειας, του σχεδιασμού και της διαχείρισης των δεδομένων.

Συνοψίζοντας, η υπάρχουσα βιβλιογραφία αναδεικνύει αφενός τη σημασία της ψηφιακής αξιολόγησης μέσω ερωτηματολογίων και αφετέρου τις εναλλακτικές

προσεγγίσεις που προσφέρονται για την υλοποίησή της. Οι εξελίξεις στο πεδίο υποδεικνύουν μια αυξανόμενη ανάγκη για λύσεις οι οποίες να είναι όχι μόνο λειτουργικές, αλλά και ευέλικτες, ασφαλείς και παραμετροποιήσιμες, προκειμένου να μπορούν να προσαρμοστούν στις πραγματικές ανάγκες των χρηστών. Η παρούσα εργασία επιχειρεί να ενταχθεί σε αυτό το πλαίσιο, αξιοποιώντας τα διδάγματα της υφιστάμενης βιβλιογραφίας για την ανάπτυξη μιας εφαρμογής που ανταποκρίνεται στις απαιτήσεις της σύγχρονης εκπαιδευτικής αξιολόγησης.

ΚΕΦΑΛΑΙΟ 1: ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΩΤΗΣΕΩΝ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

Υπάρχουσες Λύσεις

Υπάρχουσες διαδικτυακές εφαρμογές

Η διαχείριση και η αξιολόγηση γνώσεων μέσω ερωτήσεων πολλαπλής επιλογής αποτελεί εδώ και δεκαετίες μία από τις πλέον διαδεδομένες πρακτικές στον χώρο της εκπαίδευσης, αλλά και σε διαδικασίες πιστοποίησης, προσλήψεων ή συλλογής πληροφοριών. Με τη ραγδαία εξάπλωση των ψηφιακών τεχνολογιών, οι παραδοσιακές έντυπες μορφές αξιολόγησης έχουν αντικατασταθεί από διαδικτυακές εφαρμογές που προσφέρουν εύχρηστα περιβάλλοντα για τη δημιουργία, την απάντηση και την ανάλυση ερωτηματολογίων.

Υπάρχει σήμερα πληθώρα εργαλείων που επιτρέπουν τη δημιουργία ερωτήσεων πολλαπλής επιλογής με διαφορετικά επίπεδα λειτουργικότητας, εμφάνισης και αλληλεπίδρασης. Στην ενότητα αυτή παρουσιάζονται ορισμένες από τις πιο γνωστές και διαδεδομένες πλατφόρμες: **Microsoft Forms**, **Google Forms** και **Kahoot!**, οι οποίες αποτελούν ισχυρά παραδείγματα ψηφιακών λύσεων για τη δημιουργία διαδραστικών ερωτηματολογίων.

Microsoft Forms

Το **Microsoft Forms** είναι μία υπηρεσία που παρέχεται από τη Microsoft στο πλαίσιο της συνδρομής Microsoft 365 και επιτρέπει τη δημιουργία quiz, ερωτηματολογίων και ερευνών αξιολόγησης. Χαρακτηρίζεται από μια απλή και καθαρή διεπαφή, με υποστήριξη για διαφορετικούς τύπους ερωτήσεων όπως πολλαπλής επιλογής, ανοιχτού τύπου, κατατάξεων και αξιολογήσεων.

Οι χρήστες μπορούν να δημιουργήσουν φόρμες μέσα από ένα φιλικό γραφικό περιβάλλον, να προσαρμόσουν το σχεδιασμό τους με διαθέσιμα θέματα και να μοιραστούν το τελικό ερωτηματολόγιο μέσω συνδέσμου ή email. Μία από τις σημαντικές δυνατότητες του **Microsoft Forms** είναι η αυτόματη συλλογή των απαντήσεων σε πραγματικό χρόνο, με παράλληλη δημιουργία γραφημάτων και πινάκων στατιστικών στοιχείων.

Τα αποτελέσματα μπορούν να εξαχθούν άμεσα σε αρχείο Excel για περαιτέρω επεξεργασία. Επιπλέον, όταν χρησιμοποιείται σε συνδυασμό με το Microsoft Teams, το Forms ενσωματώνεται πλήρως στη ροή εργασίας εκπαιδευτικών ιδρυμάτων, επιτρέποντας στους διδάσκοντες να αναθέτουν quiz στους μαθητές και να παρακολουθούν την επίδοσή τους συγκεντρωτικά. Η υποστήριξη για αυτόματη βαθμολόγηση είναι επίσης ένα σημαντικό πλεονέκτημα για την εξοικονόμηση χρόνου.

Παρ' όλα αυτά, η λειτουργικότητα του Microsoft Forms παραμένει βασική και περιορισμένη για σενάρια σύνθετης εκπαιδευτικής αξιολόγησης. Δεν υποστηρίζει, για παράδειγμα, ερωτήσεις με παραμετρικό περιεχόμενο, δυναμικά ερωτηματολόγια ή σύνθετη λογική διαδρομής.

Google Forms

Το **Google Forms** αποτελεί ένα από τα δημοφιλέστερα εργαλεία δημιουργίας ερωτηματολογίων παγκοσμίως, προσφερόμενο δωρεάν μέσω της πλατφόρμας **Google Drive**. Είναι ευρέως χρησιμοποιούμενο τόσο στον τομέα της εκπαίδευσης όσο και σε εταιρικά ή ερευνητικά περιβάλλοντα, χάρη στην απλότητά του και την πλήρη ενσωμάτωσή του με τις υπόλοιπες υπηρεσίες της **Google**, όπως τα **Sheets** και το **Classroom**.

Το **Google Forms** προσφέρει υποστήριξη για πολλούς τύπους ερωτήσεων, όπως επιλογή μίας ή πολλαπλών απαντήσεων, σύντομο ή μακροσκελές κείμενο, επιλογή από λίστα, κατατάξεις και γραμμικές κλίμακες. Η δημιουργία των ερωτηματολογίων γίνεται μέσω ενός οπτικού περιβάλλοντος όπου οι χρήστες μπορούν εύκολα να προσθέσουν ερωτήσεις, να καθορίσουν τις σωστές απαντήσεις και να διαμορφώσουν τις φόρμες τους.

Ένα από τα ισχυρότερα χαρακτηριστικά του **Google Forms** είναι η δυνατότητα συλλογής και απεικόνισης των απαντήσεων σε πραγματικό χρόνο, καθώς και η άμεση εξαγωγή των δεδομένων σε **Google Sheets**. Οι δημιουργοί έχουν επίσης τη δυνατότητα να ενεργοποιούν τη συλλογή email, να περιορίζουν την πρόσβαση μόνο σε χρήστες ενός οργανισμού, ή να ρυθμίζουν την επανάληψη απαντήσεων.

Στο εκπαιδευτικό πλαίσιο, το **Google Forms** προσφέρει επιλογές αυτόματης βαθμολόγησης και παροχής ανατροφοδότησης στους μαθητές. Ωστόσο, όπως και το **Microsoft Forms**, δεν προσφέρει εξελιγμένα εργαλεία αξιολόγησης ή εξατομίκευσης της εμπειρίας με βάση τις απαντήσεις του χρήστη.

[Kahoot!](#)

Το **Kahoot!** (Özdemir, 2024) αποτελεί μια διαφορετικής φύσης πλατφόρμα που έχει ως στόχο τη δημιουργία διαδραστικών **quiz** σε μορφή παιχνιδιού (**Gamification**) (Anuradhani, Yatigammana, & Wijayarathna, 2024). Σε αντίθεση με τις δύο προηγούμενες εφαρμογές, το **Kahoot** είναι προσανατολισμένο κυρίως στην άμεση συμμετοχή του κοινού, την ταχύτητα απάντησης και την ενίσχυση του ανταγωνισμού και της διασκέδασης κατά τη διάρκεια της διαδικασίας.

Η βασική χρήση του **Kahoot** στηρίζεται σε ένα "ζωντανό" παιχνίδι ερωτήσεων (**live quiz**), όπου ένας εκπαιδευτικός ή παρουσιαστής προβάλλει τις ερωτήσεις στην οθόνη και οι συμμετέχοντες απαντούν μέσα από τις δικές τους συσκευές (υπολογιστές, κινητά ή **tablets**), εντός συγκεκριμένου χρόνου. Οι απαντήσεις καταγράφονται αυτόματα και η εφαρμογή εμφανίζει σε πραγματικό χρόνο ποιος απάντησε σωστά και ταχύτερα.

Η δυναμική του παιχνιδιού ενισχύεται από τον πίνακα κατάταξης και τη δυνατότητα άμεσης ανατροφοδότησης, γεγονός που αυξάνει τη συμμετοχή, ειδικά σε μαθητές μικρότερων ηλικιών. Η πλατφόρμα προσφέρει επίσης λειτουργίες όπως επαναληπτικά **quiz**, ομαδική συμμετοχή, **quiz** μέσω βιντεοδιάσκεψης και ερωτήσεις τύπου **"True or False"**.

Ωστόσο, το **Kahoot** εστιάζει περισσότερο στην εμπειρία του παιχνιδιού και όχι στην αποθήκευση και στατιστική ανάλυση των απαντήσεων. Παρότι υποστηρίζει περιορισμένα στατιστικά στοιχεία, δεν προσφέρει εύκολη εξαγωγή των δεδομένων ή δυνατότητα δημιουργίας πιο πολύπλοκων σεναρίων αξιολόγησης. Επιπλέον, η δημιουργία και η πρόσβαση σε ορισμένες δυνατότητες απαιτεί συνδρομή.

Σύγκριση χαρακτηριστικών

Οι πλατφόρμες δημιουργίας και διαχείρισης ερωτήσεων πολλαπλής επιλογής, όπως Microsoft Forms, Google Forms και Kahoot!, προσφέρουν ισχυρά εργαλεία για εκπαιδευτικούς, οργανισμούς και ερευνητές, αλλά διαφέρουν σημαντικά όσον αφορά τα χαρακτηριστικά και τη λειτουργικότητά τους.

Microsoft Forms

Ένα από τα κύρια πλεονεκτήματα του Microsoft Forms είναι η ενσωμάτωσή του στο ευρύτερο οικοσύστημα της Microsoft. Αυτό το καθιστά ιδιαίτερα ισχυρό για οργανισμούς που χρησιμοποιούν το Microsoft 365, επιτρέποντας την απρόσκοπτη συνεργασία μέσω εργαλείων όπως το Microsoft Teams, το Outlook και το Excel. Προσφέρει επίσης αυτόματη συλλογή απαντήσεων, δημιουργία γραφημάτων και δυνατότητα εξαγωγής δεδομένων σε Excel για περαιτέρω ανάλυση. Επιπλέον, υποστηρίζει την αυτόματη βαθμολόγηση των απαντήσεων, εξοικονομώντας χρόνο για τους διδάσκοντες. Ωστόσο, παρουσιάζει περιορισμούς σε πιο σύνθετες μορφές ερωτήσεων και δεν προσφέρει δυνατότητες για δυναμικά ερωτηματολόγια με προσαρμογή ερωτήσεων με βάση τις απαντήσεις του χρήστη.

Google Forms

Το Google Forms είναι ευρέως γνωστό για την απλότητα χρήσης του και την πλήρη ενσωμάτωση με τις υπόλοιπες υπηρεσίες της Google, όπως το Google Drive, τα Google Sheets και το Google Classroom. Αυτή η ενσωμάτωση διευκολύνει τη συλλογή και την ανάλυση δεδομένων, καθώς οι απαντήσεις αποθηκεύονται απευθείας σε φύλλα εργασίας, επιτρέποντας την εύκολη δημιουργία πινάκων και γραφημάτων. Ένα άλλο σημαντικό πλεονέκτημα είναι η δυνατότητα αυτόματης βαθμολόγησης και παροχής άμεσης ανατροφοδότησης στους μαθητές. Ωστόσο, η πλατφόρμα αυτή παρουσιάζει περιορισμούς όσον αφορά την εξατομίκευση των φορμών και την υποστήριξη για πιο σύνθετες λογικές ροές. Επίσης, δεν διαθέτει εξελιγμένα εργαλεία διαδραστικότητας, όπως πίνακες κατάταξης ή live games, που είναι κοινά σε πλατφόρμες όπως το Kahoot!

Kahoot!

Αντίθετα, το Kahoot! επικεντρώνεται στη διαδραστικότητα και τη διασκέδαση, προσφέροντας μια παιχνιδοκεντρική προσέγγιση στη μάθηση. Η πλατφόρμα αυτή είναι ιδανική για ζωντανά quiz, με χαρακτηριστικά όπως άμεση ανατροφοδότηση, πίνακες κατάταξης και έντονα γραφικά που ενθαρρύνουν τη συμμετοχή. Αυτά τα χαρακτηριστικά το καθιστούν εξαιρετικά δημοφιλές σε σχολικές αίθουσες και σεμινάρια, όπου η ενεργή συμμετοχή των μαθητών είναι κρίσιμη. Ωστόσο, το Kahoot! περιορίζεται στην ανάλυση των αποτελεσμάτων, με λιγότερο εξελιγμένα εργαλεία εξαγωγής και διαχείρισης δεδομένων σε σχέση με τα Forms της Microsoft και της Google. Επίσης, η χρήση ορισμένων πιο προηγμένων χαρακτηριστικών απαιτεί συνδρομή, κάτι που μπορεί να αυξήσει το κόστος χρήσης σε οργανισμούς με μεγάλες ομάδες χρηστών.

Συνολικά, κάθε μία από αυτές τις πλατφόρμες έχει τα δικά της πλεονεκτήματα και μειονεκτήματα, ανάλογα με τις ανάγκες των χρηστών. Ενώ τα quiz των

Microsoft και Google προσφέρουν απλότητα και ευκολία χρήσης, το Kahoot! ξεχωρίζει για την εμπειρία που προσφέρει στους συμμετέχοντες, δημιουργώντας μια πιο διαδραστική και διασκεδαστική εμπειρία μάθησης.

Εφαρμογή στο πλαίσιο διενέργειας εξετάσεων

Οι πλατφόρμες Microsoft Forms, Google Forms και Kahoot! έχουν καθιερωθεί ως δημοφιλή εργαλεία για τη δημιουργία και διαχείριση εξετάσεων, καλύπτοντας ένα ευρύ φάσμα εκπαιδευτικών και επαγγελματικών αναγκών. Καθεμία από αυτές τις εφαρμογές προσφέρει μοναδικά χαρακτηριστικά που τις καθιστούν κατάλληλες για διαφορετικά σενάρια χρήσης.

Το Microsoft Forms είναι ιδιαίτερα διαδεδομένο σε οργανισμούς που χρησιμοποιούν το Microsoft 365, επιτρέποντας την απρόσκοπτη δημιουργία τεστ πολλαπλής επιλογής, ερωτήσεων ανοιχτού τύπου και κατατάξεων. Οι απαντήσεις συλλέγονται αυτόματα και μπορούν να εξαχθούν σε μορφή Excel για περαιτέρω ανάλυση. Επιπλέον, η πλατφόρμα υποστηρίζει αυτόματη βαθμολόγηση, εξοικονομώντας πολύτιμο χρόνο για τους εκπαιδευτικούς. Ένα από τα μεγάλα πλεονεκτήματά του είναι η ενσωμάτωσή του με το Microsoft Teams, επιτρέποντας την οργάνωση εξετάσεων σε διαδικτυακά μαθήματα με δυνατότητα άμεσης παρακολούθησης των αποτελεσμάτων.

Το Google Forms διακρίνεται για την απλότητα χρήσης του και την πλήρη ενσωμάτωσή του με τα Google Sheets, καθιστώντας το ιδανικό για τη συλλογή και ανάλυση δεδομένων σε πραγματικό χρόνο. Προσφέρει ευελιξία μέσω της λογικής διαδρομής, επιτρέποντας τη δημιουργία προσαρμοσμένων εξετάσεων όπου οι επόμενες ερωτήσεις προσαρμόζονται με βάση τις απαντήσεις του μαθητή. Αυτό είναι ιδιαίτερα χρήσιμο σε διαγνωστικές αξιολογήσεις, όπου η δυσκολία των ερωτήσεων πρέπει να προσαρμόζεται δυναμικά ανάλογα με τις γνώσεις του εξεταζόμενου.

Από την άλλη πλευρά, το Kahoot! ξεχωρίζει για την έμφαση που δίνει στη διαδραστικότητα και τη συμμετοχή. Η πλατφόρμα αυτή είναι σχεδιασμένη για ζωντανά quiz, όπου οι μαθητές απαντούν σε πραγματικό χρόνο από τις συσκευές τους και οι βαθμολογίες εμφανίζονται άμεσα σε πίνακες κατάταξης. Αυτό το μοντέλο είναι ιδιαίτερα αποτελεσματικό για επαναληπτικές εξετάσεις και διαγωνισμούς γνώσεων, δημιουργώντας ένα ανταγωνιστικό περιβάλλον μάθησης που ενθαρρύνει την ενεργή συμμετοχή σε μαθήματα όλων των επιστημονικών πεδίων (Rašić, Aleksic-Maslac, & Vranesic, 2018) (Alf & Rabail, 2020).

Συνολικά, οι πλατφόρμες αυτές προσφέρουν διαφορετικά εργαλεία για την αξιολόγηση γνώσεων, επιτρέποντας στους εκπαιδευτικούς να επιλέγουν την κατάλληλη λύση ανάλογα με τις ανάγκες της τάξης τους και τους στόχους της εξέτασης. Παρά τα πλεονεκτήματά τους, ωστόσο, καμία από αυτές δεν καλύπτει πλήρως τις απαιτήσεις πιο σύνθετων εξετάσεων, γεγονός που αναδεικνύει την ανάγκη για πιο εξειδικευμένες λύσεις.

Πλεονεκτήματα και μειονεκτήματα των έτοιμων λύσεων

Πλεονεκτήματα

Οι έτοιμες λύσεις για τη δημιουργία και διαχείριση ερωτήσεων πολλαπλής επιλογής, όπως τα Microsoft Forms, Google Forms και Kahoot!, προσφέρουν πολλά πλεονεκτήματα που τις καθιστούν ιδιαίτερα δημοφιλείς σε εκπαιδευτικά

και επαγγελματικά περιβάλλοντα. Ένα από τα σημαντικότερα πλεονεκτήματα είναι η ευκολία χρήσης. Οι περισσότερες από αυτές τις πλατφόρμες διαθέτουν φιλικά προς τον χρήστη γραφικά περιβάλλοντα, τα οποία δεν απαιτούν προγραμματιστικές γνώσεις για τη δημιουργία φορμών και ερωτηματολογίων. Αυτό επιτρέπει στους εκπαιδευτικούς και τους διοργανωτές να δημιουργούν και να διαχειρίζονται εξετάσεις χωρίς την ανάγκη εξειδικευμένων τεχνικών δεξιοτήτων (Sofiani & Prasetya, 2024).

Επιπλέον, τα έτοιμα εργαλεία προσφέρουν αυτοματοποιημένη συλλογή και ανάλυση δεδομένων, εξοικονομώντας πολύτιμο χρόνο. Για παράδειγμα, οι απαντήσεις σε Google Forms αποθηκεύονται αυτόματα σε Google Sheets, επιτρέποντας την άμεση δημιουργία γραφημάτων και στατιστικών αναφορών. Αντίστοιχα, το Microsoft Forms παρέχει τη δυνατότητα εξαγωγής των αποτελεσμάτων σε Excel, ενώ το Kahoot! καταγράφει τις απαντήσεις σε πραγματικό χρόνο με αυτόματους πίνακες κατάταξης.

Ένα ακόμα σημαντικό πλεονέκτημα είναι η ενσωμάτωση με άλλες υπηρεσίες. Το Microsoft Forms συνεργάζεται στενά με το Microsoft Teams και άλλες εφαρμογές του Microsoft 365, καθιστώντας το ιδανικό για οργανισμούς που χρησιμοποιούν ήδη αυτές τις υπηρεσίες. Παρομοίως, τα Google Forms είναι πλήρως ενσωματωμένα με το οικοσύστημα της Google, προσφέροντας άμεση πρόσβαση σε εργαλεία όπως το Google Drive και το Google Classroom.

Τέλος, οι πλατφόρμες αυτές προσφέρουν δυνατότητες εξ αποστάσεως αξιολόγησης, υποστηρίζοντας υβριδικά μοντέλα μάθησης. Με την πανδημία COVID-19 να επιταχύνει την ανάγκη για διαδικτυακές εκπαιδευτικές λύσεις, τα εργαλεία αυτά απέδειξαν την αξία τους, επιτρέποντας την οργάνωση και την εκτέλεση εξετάσεων χωρίς γεωγραφικούς περιορισμούς (Souto Romero, Arias-Oliva, O., & Pérez-Portabella, 2024).

Μειονεκτήματα

Παρά τα σημαντικά τους πλεονεκτήματα, οι έτοιμες λύσεις για τη διαχείριση ερωτήσεων πολλαπλής επιλογής δεν είναι χωρίς περιορισμούς. Ένα από τα κύρια μειονεκτήματά τους είναι η περιορισμένη ευελιξία. Αν και προσφέρουν εύχρηστα περιβάλλοντα για τη δημιουργία απλών ερωτηματολογίων, συχνά υστερούν όταν πρόκειται για πιο σύνθετες λογικές ροές ή προσαρμοσμένες εμπειρίες χρήστη. Για παράδειγμα, ενώ τα Google Forms επιτρέπουν τη δημιουργία "λογικής διαδρομής" με προσαρμοσμένες ερωτήσεις ανάλογα με τις απαντήσεις του χρήστη, η λειτουργία αυτή παραμένει περιορισμένη σε σύγκριση με εξειδικευμένες λύσεις.

Επιπλέον, οι περισσότερες από αυτές τις πλατφόρμες έχουν περιορισμένες επιλογές προσαρμογής εμφάνισης και στυλ, καθιστώντας δύσκολη την προσαρμογή τους σε εταιρικές ή ακαδημαϊκές απαιτήσεις για branding. Για παράδειγμα, το Kahoot! προσφέρει πολύ περιορισμένες επιλογές προσαρμογής στη δωρεάν έκδοσή του, ενώ οι πιο προηγμένες λειτουργίες του απαιτούν συνδρομή.

Ακόμη, πολλά από αυτά τα εργαλεία δεν προσφέρουν προηγμένες δυνατότητες ανάλυσης δεδομένων. Αν και οι απαντήσεις μπορούν να εξαχθούν σε μορφή υπολογιστικών φύλλων, η επεξεργασία των δεδομένων για βαθύτερη ανάλυση απαιτεί πρόσθετα εργαλεία ή χειροκίνητη εργασία. Αυτό είναι ιδιαίτερα εμφανές στο Kahoot!, το οποίο επικεντρώνεται περισσότερο στην αμεσότητα της συμμετοχής παρά στη λεπτομερή ανάλυση αποτελεσμάτων.

Τέλος, οι περισσότερες από αυτές τις πλατφόρμες δεν προσφέρουν ισχυρή υποστήριξη ασφάλειας και απορρήτου, ιδιαίτερα όταν χρησιμοποιούνται για

εξετάσεις υψηλής σημασίας ή για την αποθήκευση προσωπικών δεδομένων (Chondrogiannos, 2021). Η διαχείριση των δικαιωμάτων πρόσβασης και η προστασία των δεδομένων μπορεί να είναι δύσκολη, ειδικά σε οργανισμούς που απαιτούν αυστηρά μέτρα ασφαλείας.

Προσαρμοσμένες Λύσεις

Η ανάγκη για προσαρμοσμένες λύσεις διαχείρισης ερωτήσεων γίνεται όλο και πιο έντονη, καθώς οι έτοιμες πλατφόρμες συχνά παρουσιάζουν περιορισμούς στη λειτουργικότητα και την παραμετροποίηση, ενώ μπορεί να επιφέρουν σημαντικό κόστος για τους οργανισμούς που τις χρησιμοποιούν. Παρά το γεγονός ότι εργαλεία όπως τα **Microsoft Forms**, **Google Forms** και **Kahoot!** προσφέρουν ευκολία χρήσης και γρήγορη ενσωμάτωση, συχνά απαιτούν συνδρομές για την πρόσβαση σε προηγμένες λειτουργίες, όπως η ανάλυση δεδομένων, η διαχείριση χρηστών και η ενσωμάτωση με άλλες επιχειρησιακές πλατφόρμες. Αυτό καθιστά δύσκολη την υιοθέτησή τους από μικρότερους οργανισμούς, εκπαιδευτικά ιδρύματα και μη κερδοσκοπικούς φορείς με περιορισμένους πόρους.

Η ανάπτυξη εξατομικευμένων εφαρμογών εντός των ίδιων των οργανισμών προσφέρει τη δυνατότητα πλήρους ελέγχου και ευελιξίας, επιτρέποντας την προσαρμογή της λειτουργικότητας στις συγκεκριμένες ανάγκες των χρηστών. Επιπλέον, η χρήση ανοιχτού κώδικα και η εκμετάλλευση των σύγχρονων τεχνολογιών μπορεί να μειώσει σημαντικά το κόστος ανάπτυξης και συντήρησης. Αυτή η προσέγγιση όχι μόνο ενισχύει την προσβασιμότητα σε νέες τεχνολογίες, αλλά συμβάλλει και στη βιώσιμη ανάπτυξη, δημιουργώντας λύσεις που είναι πλήρως προσαρμοσμένες στις επιχειρησιακές απαιτήσεις και μπορούν να εξελισσονται μαζί με τις ανάγκες των χρηστών.

ΚΕΦΑΛΑΙΟ 2: ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ ΓΙΑ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

Τεχνολογίες Ανάπτυξης Διαδικτυακών Εφαρμογών

Ιστορική Αναδρομή και Εξέλιξη του Διαδικτύου

Το διαδίκτυο ξεκίνησε ως ένα ερευνητικό έργο στις Ηνωμένες Πολιτείες τη δεκαετία του 1960, με στόχο τη δημιουργία ενός ανθεκτικού, αποκεντρωμένου δικτύου επικοινωνίας. Το πρώτο σημαντικό βήμα έγινε με την ανάπτυξη του **ARPANET (Advanced Research Projects Agency Network)** το 1969 (Packard, 2020), το οποίο συνέδεσε αρχικά τέσσερα πανεπιστήμια στις ΗΠΑ. Το **ARPANET** χρησιμοποίησε την τεχνολογία μεταγωγής πακέτων (**packet switching**), η οποία επέτρεπε την ταυτόχρονη μεταφορά δεδομένων μεταξύ πολλών υπολογιστών.

Κατά τη δεκαετία του 1980, το διαδίκτυο επεκτάθηκε με την υιοθέτηση του πρωτοκόλλου **TCP/IP (Transmission Control Protocol/Internet Protocol)**, το οποίο έγινε το βασικό πρότυπο για τη μεταφορά δεδομένων. Σημαντική πρόοδος σημειώθηκε το 1990 με την εφεύρεση του **World Wide Web** από τον **Tim Berners-Lee**, που εισήγαγε την έννοια των υπερσυνδέσμων (**hyperlinks**) και των ιστοσελίδων, καθιστώντας το διαδίκτυο προσιτό στο ευρύ κοινό.

Σήμερα, το διαδίκτυο αποτελεί τη βάση για επικοινωνία, ψυχαγωγία, ηλεκτρονικό εμπόριο και ανταλλαγή πληροφοριών, επηρεάζοντας σχεδόν κάθε πτυχή της σύγχρονης ζωής.

Αρχιτεκτονική Διαδικτυακών Εφαρμογών

Η αρχιτεκτονική των διαδικτυακών εφαρμογών βασίζεται στο μοντέλο **Client-Server** (Haroon- Sulyman, 2014), όπου ο **Client** (πελάτης) είναι το λογισμικό που τρέχει στο πρόγραμμα περιήγησης του χρήστη και ο **Server** (διακομιστής) είναι ο κεντρικός υπολογιστής που επεξεργάζεται τα αιτήματα και αποστέλλει δεδομένα στον πελάτη. Αυτή η αρχιτεκτονική επιτρέπει την αλληλεπίδραση μεταξύ χρηστών και εφαρμογών μέσω του διαδικτύου, προσφέροντας μια δομημένη προσέγγιση για την ανταλλαγή δεδομένων.

Μία από τις πιο διαδεδομένες προσεγγίσεις για την επικοινωνία μεταξύ client και server είναι τα **REST APIs (Representational State Transfer Application Programming Interfaces)**. Τα **REST APIs** χρησιμοποιούν απλά **HTTP** αιτήματα (**GET, POST, PUT, DELETE**) για την αποστολή και λήψη δεδομένων σε μορφή **JSON** ή **XML**. Αυτή η αρχιτεκτονική είναι ελαφριά, ευέλικτη και κλιμακούμενη, καθιστώντας την ιδανική για την ανάπτυξη μοντέρνων εφαρμογών.

Παράλληλα, η έννοια των **Single Page Applications (SPA)** έχει κερδίσει δημοτικότητα (Kornienko, Mishina, & Melnikov, 2021), καθώς επιτρέπει τη φόρτωση του **frontend** μιας εφαρμογής σε μία μόνο σελίδα. Αυτό βελτιώνει την ταχύτητα και την εμπειρία του χρήστη, καθώς τα δεδομένα ανανεώνονται δυναμικά χωρίς την ανάγκη επαναφόρτωσης ολόκληρης της σελίδας. Δημοφιλή **frameworks** όπως το **React**, **Angular** και **Vue.js** βασίζονται σε αυτήν την προσέγγιση, προσφέροντας ταχύτητα και διαδραστικότητα σε σύγχρονες

διαδικτυακές εφαρμογές. Τύποι Διαδικτυακών Εφαρμογών (**Static**, Dynamic, Single Page Applications, Progressive Web Applications)

Προκλήσεις στην Ανάπτυξη Διαδικτυακών Εφαρμογών

Η ανάπτυξη διαδικτυακών εφαρμογών αντιμετωπίζει πολλές προκλήσεις, οι οποίες ποικίλλουν ανάλογα με την πολυπλοκότητα και το εύρος της εφαρμογής. Μία από τις κύριες προκλήσεις είναι η **διαχείριση της απόδοσης**, καθώς οι εφαρμογές πρέπει να είναι γρήγορες και αποδοτικές, ακόμη και υπό υψηλό φόρτο χρηστών. Αυτό απαιτεί βελτιστοποίηση του **frontend** και του **backend**, καθώς και σωστή χρήση των δικτύων διανομής περιεχομένου (**CDNs**) για τη μείωση των χρόνων φόρτωσης.

Η **ασφάλεια** αποτελεί επίσης σημαντικό παράγοντα, καθώς οι εφαρμογές είναι εκτεθειμένες σε απειλές όπως **SQL Injection**, **Cross-Site Scripting (XSS)**, **Cross-Site Request Forgery (CSRF)** και επιθέσεις **DDoS** (Rauf, Rivaldo, & Sekolah Tinggi Sandi Negara). Επιπλέον, η διαχείριση των προσωπικών δεδομένων πρέπει να συμμορφώνεται με κανονισμούς όπως το **GDPR**.

Επιπρόσθετα, οι σύγχρονες εφαρμογές πρέπει να είναι **responsive** και να λειτουργούν απρόσκοπτα σε διάφορες συσκευές και προγράμματα περιήγησης. Τέλος, η σύνδεση μεταξύ frontend και backend και η διαχείριση δεδομένων είναι συχνά περίπλοκη, ειδικά σε εφαρμογές μεγάλης κλίμακας, όπου απαιτείται σταθερότητα και επεκτασιμότητα.

Τεχνολογίες Frontend

HTML, CSS και JavaScript

Η **HTML (HyperText Markup Language)**, **CSS (Cascading Style Sheets)** και **JavaScript** είναι οι τρεις βασικές τεχνολογίες που αποτελούν τον πυρήνα του frontend ανάπτυξης για ιστοσελίδες και διαδικτυακές εφαρμογές. Η **HTML** είναι μια γλώσσα σήμανσης που καθορίζει τη δομή και το περιεχόμενο μιας ιστοσελίδας, χρησιμοποιώντας ετικέτες (tags) για να δημιουργήσει στοιχεία όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους και φόρμες. Η **CSS** χρησιμοποιείται για τη μορφοποίηση και την αισθητική διάταξη των στοιχείων HTML, προσθέτοντας στυλ όπως χρώματα, γραμματοσειρές, περιθώρια και animations. Τέλος, η **JavaScript** είναι μια γλώσσα προγραμματισμού που προσθέτει διαδραστικότητα και δυναμική λειτουργικότητα στις ιστοσελίδες. Χρησιμοποιείται για την εκτέλεση υπολογισμών, την επικοινωνία με το backend μέσω AJAX ή Fetch API, και τη διαχείριση συμβάντων (events) όπως κλικ και πληκτρολόγηση. Ο συνδυασμός αυτών των τεχνολογιών επιτρέπει την ανάπτυξη σύγχρονων και δυναμικών ιστοσελίδων που προσφέρουν πλούσια εμπειρία χρήστη.

Βιβλιοθήκες και Frontend Frameworks

Οι βιβλιοθήκες και τα frontend frameworks είναι εργαλεία που διευκολύνουν την ανάπτυξη δυναμικών και διαδραστικών εφαρμογών, παρέχοντας έτοιμες λειτουργίες και δομές που επιταχύνουν την ανάπτυξη. Τα πιο δημοφιλή frameworks περιλαμβάνουν το **React**, το **Angular** και το **Vue.js**, τα οποία χρησιμοποιούνται ευρέως για τη δημιουργία **Single Page Applications (SPA)** (Jonathan & Supriyadi, 2023). Το **React**, που αναπτύχθηκε από το **Facebook**, είναι μια βιβλιοθήκη για τη δημιουργία επαναχρησιμοποιήσιμων UI components

και τη διαχείριση της κατάστασης της εφαρμογής. Το **Angular**, που αναπτύχθηκε από την Google, είναι ένα πλήρες framework με ισχυρά εργαλεία για τη δημιουργία μεγάλων και σύνθετων εφαρμογών. Το **Vue.js** είναι γνωστό για την απλότητά του και τη γρήγορη καμπύλη εκμάθησης, καθιστώντας το ιδανικό για μικρότερες εφαρμογές ή έργα με περιορισμένους πόρους. Αυτά τα εργαλεία προσφέρουν σημαντικά πλεονεκτήματα, όπως ταχύτερη ανάπτυξη, καλύτερη οργάνωση κώδικα και αυξημένη απόδοση εφαρμογών.

Responsive Design και Mobile-First Προσέγγιση

Το **Responsive Design** είναι μια τεχνική σχεδίασης ιστοσελίδων που διασφαλίζει ότι το περιεχόμενο προσαρμόζεται αυτόματα σε διαφορετικά μεγέθη οθονών και συσκευές (Noorkaran, Aditi, & Mehta, 2023). Αυτή η προσέγγιση είναι απαραίτητη στη σύγχρονη ανάπτυξη ιστοσελίδων, καθώς οι χρήστες πλέον επισκέπτονται ιστοσελίδες από μια ποικιλία συσκευών, όπως smartphones, tablets, desktops και smartTVs. Το **Mobile-First Design** είναι μια φιλοσοφία ανάπτυξης που ξεκινά με τον σχεδιασμό για μικρές οθόνες και στη συνέχεια επεκτείνεται για μεγαλύτερες οθόνες. Αυτό εξασφαλίζει καλύτερη απόδοση και εμπειρία χρήστη σε κινητές συσκευές. Για την υλοποίηση responsive design, χρησιμοποιούνται εργαλεία όπως το **Bootstrap**, το **Tailwind CSS** και το **Material UI**, τα οποία παρέχουν προκαθορισμένες κλάσεις CSS και έτοιμα UI components. Επιπλέον, η χρήση media queries, flexbox και CSS grid είναι κρίσιμη για τη δημιουργία layouts που προσαρμόζονται δυναμικά ανάλογα με το μέγεθος της οθόνης και τις προδιαγραφές της συσκευής.

Διαχείριση Κατάστασης Δεδομένων (State Management)

Η διαχείριση κατάστασης (state management) είναι κρίσιμη σε μοντέρνες διαδικτυακές εφαρμογές, καθώς αυτές συχνά απαιτούν την παρακολούθηση και την ενημέρωση πολλών στοιχείων και δεδομένων ταυτόχρονα. Το state αναφέρεται στην τρέχουσα κατάσταση της εφαρμογής, όπως τα δεδομένα χρήστη, η κατάσταση σύνδεσης και οι επιλογές πλοήγησης. Σε μικρές εφαρμογές, η διαχείριση του state μπορεί να γίνει απευθείας με **React hooks** όπως **useState** και **useReducer**. Ωστόσο, για μεγαλύτερες εφαρμογές με περίπλοκη λογική και πολλές αλληλεπιδράσεις, χρησιμοποιούνται εξειδικευμένα εργαλεία όπως το **Redux**, το **Zustand**, το **MobX** και το **Context API**. Αυτά τα εργαλεία προσφέρουν κεντρική αποθήκευση και διαχείριση του state, επιτρέποντας την εύκολη κοινή χρήση δεδομένων μεταξύ των components και βελτιώνοντας την απόδοση της εφαρμογής. Η σωστή διαχείριση του state είναι κρίσιμη για την αποφυγή προβλημάτων όπως "state inconsistency" και την εξασφάλιση ομαλής εμπειρίας χρήστη.

Τεχνολογίες Backend

Εισαγωγή στις Τεχνολογίες Backend

Οι τεχνολογίες backend αποτελούν τον πυρήνα των διαδικτυακών εφαρμογών, καθώς είναι υπεύθυνες για την επεξεργασία των δεδομένων, την επιχειρηματική λογική και τη σύνδεση με τις βάσεις δεδομένων. Το backend αναλαμβάνει την επεξεργασία αιτημάτων από τον πελάτη (client), την αποθήκευση και ανάκτηση πληροφοριών και την επικοινωνία με άλλες υπηρεσίες. Σε αντίθεση με το frontend, το οποίο είναι υπεύθυνο για την εμφάνιση και την αλληλεπίδραση με τον χρήστη, το backend λειτουργεί στο παρασκήνιο και χειρίζεται την ασφάλεια,

την κλιμάκωση και την αξιοπιστία της εφαρμογής. Χρησιμοποιεί γλώσσες προγραμματισμού όπως **JavaScript (Node.js)**, **Python (Django)**, **PHP (Laravel)** και **Java (Spring Boot)**, σε συνδυασμό με βάσεις δεδομένων για την αποθήκευση δεδομένων.

Γλώσσες Προγραμματισμού για Backend

Η επιλογή γλώσσας προγραμματισμού για το backend αποτελεί θεμελιώδη απόφαση στον σχεδιασμό μιας διαδικτυακής εφαρμογής, καθώς επηρεάζει τη λειτουργικότητα, την απόδοση και τη συντηρησιμότητα μιας διαδικτυακής εφαρμογής. Η JavaScript, μέσω του Node.js, υποστηρίζει ασύγχρονο και event-driven προγραμματισμό, γεγονός που την καθιστά κατάλληλη για εφαρμογές πραγματικού χρόνου όπως συνομιλίες και υπηρεσίες μετάδοσης ροής(streaming). Η Python ξεχωρίζει για την απλότητα και τη γρήγορη ανάπτυξη, καθιστώντας την κατάλληλη για APIs, ανάλυση δεδομένων και εκπαιδευτικά περιβάλλοντα.

Η Java παραμένει ισχυρή επιλογή σε εταιρικά περιβάλλοντα, προσφέροντας ασφάλεια, πολυνηματικότητα και σταθερότητα σε έργα μεγάλης κλίμακας. Η PHP χρησιμοποιείται ευρέως σε ιστοσελίδες και CMS, λόγω της απλότητας και της ευρείας υποστήριξης, ενώ οι νεότερες εκδόσεις της ενισχύουν τις επιδόσεις και την ασφάλεια. Η C#, μέσω του .NET οικοσυστήματος, ενδείκνυται για επιχειρησιακές εφαρμογές, με ισχυρή τυποποίηση και σύγχρονα εργαλεία ανάπτυξης. Η Ruby, τέλος, προσφέρει έναν ευέλικτο και φιλικό προς τον προγραμματιστή τρόπο ανάπτυξης, ιδανική για μικρές ομάδες και startups.

Η καταλληλότητα κάθε γλώσσας εξαρτάται από τις απαιτήσεις του έργου, την εμπειρία της ομάδας και τη διαθεσιμότητα βιβλιοθηκών και εργαλείων. Κάθε γλώσσα έχει τα δικά της πλεονεκτήματα και μειονεκτήματα, όπως η ταχύτητα ανάπτυξης, η απόδοση, η ασφάλεια και η επεκτασιμότητα.

Backend Frameworks και Πλαίσια Ανάπτυξης

Τα frameworks backend είναι εργαλεία που προσφέρουν έτοιμες δομές και βιβλιοθήκες για την ανάπτυξη διαδικτυακών εφαρμογών, διευκολύνοντας την ταχύτερη και πιο οργανωμένη ανάπτυξη. Δημοφιλή frameworks περιλαμβάνουν το **Express.js** για **Node.js**, το **Django** και το **Flask** για Python, το **Ruby on Rails** για Ruby, το **Laravel** για PHP και το **Spring Boot** για Java. Αυτά τα εργαλεία προσφέρουν λειτουργίες όπως routing, data validation, authentication και session management, μειώνοντας την ανάγκη για χειροκίνητο κώδικα και αυξάνοντας την παραγωγικότητα των προγραμματιστών. Η επιλογή του κατάλληλου framework εξαρτάται από τη γλώσσα προγραμματισμού, την κλίμακα του έργου και τις απαιτήσεις της εφαρμογής.

RESTful APIs και JSON

Τα **RESTful APIs (Representational State Transfer Application Programming Interfaces)** είναι ένας ευρέως χρησιμοποιούμενος τρόπος επικοινωνίας μεταξύ του frontend και του backend σε διαδικτυακές εφαρμογές. Τα REST APIs βασίζονται σε αρχές όπως το stateless design, η χρήση HTTP methods (GET, POST, PUT, DELETE) και η ανταλλαγή δεδομένων σε μορφή **JSON** (JavaScript Object Notation). Αυτή η προσέγγιση επιτρέπει την εύκολη διασύνδεση διαφορετικών συστημάτων, προσφέροντας ταχύτητα και επεκτασιμότητα. Τα REST APIs είναι ανεξάρτητα από την πλατφόρμα και

μπορούν να χρησιμοποιηθούν με οποιοδήποτε frontend framework, όπως το **React**, το **Angular** ή το **Vue.js**. Αν και είναι απλά στην ανάπτυξη, απαιτούν προσεκτικό σχεδιασμό για την αποφυγή προβλημάτων ασφαλείας και απόδοσης.

Ενδιάμεσο Λογισμικό (Middleware)

Το **middleware** είναι ένα κρίσιμο κομμάτι της αρχιτεκτονικής backend, το οποίο λειτουργεί ως ενδιάμεσο στρώμα μεταξύ του server και των αιτημάτων του πελάτη. Χρησιμοποιείται για την επεξεργασία των αιτημάτων πριν φτάσουν στις κύριες λειτουργίες της εφαρμογής, προσφέροντας λειτουργίες όπως authentication, logging, data validation και error handling. Για παράδειγμα, σε ένα Express.js backend, τα middleware modules μπορούν να χρησιμοποιηθούν για τον έλεγχο των δικαιωμάτων πρόσβασης ή την ανίχνευση κακόβουλων ενεργειών. Τα middleware είναι συνήθως επαναχρησιμοποιήσιμα και επιτρέπουν τη modular ανάπτυξη, καθιστώντας την εφαρμογή πιο οργανωμένη και ασφαλή.

Session Management και Authentication

Η διαχείριση των συνδέσεων (session management) και η ταυτοποίηση χρηστών (authentication) είναι κρίσιμα στοιχεία για την ασφάλεια των διαδικτυακών εφαρμογών. Οι sessions επιτρέπουν την αναγνώριση ενός χρήστη κατά τη διάρκεια μιας επίσκεψης στην εφαρμογή, χρησιμοποιώντας cookies, tokens ή άλλες τεχνικές. Οι μέθοδοι authentication περιλαμβάνουν τη χρήση usernames και passwords, **JWT (JSON Web Tokens)** για ασύγχρονη ταυτοποίηση, και πρωτόκολλα όπως **OAuth** και **OpenID Connect** για εξωτερική πιστοποίηση. Αυτές οι τεχνικές βοηθούν στην προστασία των δεδομένων και στην αποτροπή μη εξουσιοδοτημένης πρόσβασης, βελτιώνοντας την ασφάλεια και την εμπειρία χρήστη.

Αρχιτεκτονικές Backend

Οι αρχιτεκτονικές backend καθορίζουν τον τρόπο με τον οποίο μια εφαρμογή διαχειρίζεται τα αιτήματα, την αποθήκευση δεδομένων και την επικοινωνία με εξωτερικές υπηρεσίες. Παραδοσιακά, οι εφαρμογές χρησιμοποιούσαν **monolithic** αρχιτεκτονικές, όπου ολόκληρη η εφαρμογή ήταν ενιαία και αδιαίρετη. Σήμερα, οι **microservices** είναι δημοφιλείς, καθώς διαχωρίζουν την εφαρμογή σε μικρότερα, ανεξάρτητα services, επιτρέποντας καλύτερη κλιμάκωση και συντήρηση. Επίσης, αρχιτεκτονικές όπως **Serverless** και **Event-Driven** χρησιμοποιούνται για εφαρμογές με υψηλές απαιτήσεις απόδοσης και δυναμικής κλιμάκωσης. Η επιλογή της αρχιτεκτονικής εξαρτάται από τις ανάγκες του έργου, τον διαθέσιμο προϋπολογισμό και την τεχνολογική στοίβα.

Κλίμακα και Βελτιστοποίηση

Η κλιμάκωση και η βελτιστοποίηση είναι κρίσιμα στοιχεία για τη διατήρηση της απόδοσης και της αξιοπιστίας των backend συστημάτων. Η κλιμάκωση μπορεί να είναι **οριζόντια** (προσθήκη περισσότερων servers) ή **κάθετη** (αναβάθμιση του υπάρχοντος hardware). Για την αποτελεσματική κλιμάκωση, χρησιμοποιούνται τεχνικές όπως εξισορρόπηση φόρτου (**load balancing**), προσωρινής αποθήκευσης (**caching**) και διαχωρισμού της βάσης δεδομένων (**database sharding**). Τα εργαλεία όπως **Redis** και **Memcached** βοηθούν στην αποθήκευση προσωρινών δεδομένων για ταχύτερη ανάκτηση. Επιπλέον, η βελτιστοποίηση περιλαμβάνει τη μείωση της καθυστέρησης (latency), τη βελτίωση του χρόνου απόκρισης των APIs

και την αποφυγή σημείων συμφόρησης (**bottlenecks**) μέσω τεχνικών όπως **query optimization** και τη χρήση δεξαμενών συνδέσεων (**connection pooling**). Η χρήση εργαλείων παρακολούθησης απόδοσης, όπως **New Relic** και **Prometheus**, επιτρέπει την έγκαιρη αναγνώριση και αντιμετώπιση προβλημάτων, εξασφαλίζοντας την απρόσκοπτη λειτουργία της εφαρμογής.

Ασφάλεια Backend

Η ασφάλεια του backend είναι κρίσιμη για την προστασία των δεδομένων και των χρηστών μιας εφαρμογής. Οι προγραμματιστές πρέπει να λαμβάνουν μέτρα για την προστασία από κοινές απειλές, όπως **SQL Injection**, **Cross-Site Scripting (XSS)**, **Cross-Site Request Forgery (CSRF)** και επιθέσεις **DDoS**. Οι τεχνικές προστασίας περιλαμβάνουν την κρυπτογράφηση δεδομένων, τη χρήση ισχυρών πρωτοκόλλων ταυτοποίησης (**JWT**, **OAuth**), τη διασφάλιση ασφαλών συνδέσεων μέσω **HTTPS** και την εφαρμογή πολιτικών πρόσβασης (**Role-Based Access Control**). Επιπλέον, η χρήση firewall και εργαλείων ασφάλειας, όπως **OWASP ZAP** και **Burp Suite**, βοηθά στην αναγνώριση και την αποτροπή πιθανών επιθέσεων. Η συνεχής παρακολούθηση και ενημέρωση του backend είναι απαραίτητη για την αντιμετώπιση νέων τρωτών σημείων και την ενίσχυση της ασφάλειας.

Διαχείριση Σφαλμάτων και Logging

Η διαχείριση σφαλμάτων (**error handling**) και το **logging** είναι ζωτικής σημασίας για τη σταθερότητα και την ευκολία συντήρησης των backend εφαρμογών. Τα σφάλματα μπορεί να είναι **client-side**, **server-side** ή **network-related**, και απαιτούν κατάλληλο χειρισμό για να αποφευχθεί η διακοπή της λειτουργίας της εφαρμογής. Τα εργαλεία **logging**, όπως **Winston** για Node.js, **Log4j** για Java και **Sentry** για πολυγλωσσική υποστήριξη, επιτρέπουν την καταγραφή σημαντικών συμβάντων και την ανίχνευση προβλημάτων σε πραγματικό χρόνο. Καλή πρακτική είναι η διάκριση των σφαλμάτων σε κατηγορίες (π.χ. 404 Not Found, 500 Internal Server Error) και η αποστολή ειδοποιήσεων για κρίσιμα προβλήματα. Επιπλέον, η χρήση **log rotation** και **log aggregation** βοηθά στην αποφυγή υπερφόρτωσης των server και στη βελτίωση της απόδοσης.

DevOps και CI/CD για Backend

Το **DevOps** είναι μια μεθοδολογία που συνδυάζει την ανάπτυξη λογισμικού (**Development**) με τη διαχείριση υποδομών (**Operations**), προωθώντας τη συνεχή βελτίωση και την ταχεία παράδοση κώδικα. Τα εργαλεία **CI/CD (Continuous Integration/Continuous Deployment)**, όπως **Jenkins**, **GitHub Actions** και **GitLab CI/CD**, επιτρέπουν την αυτόματη εκτέλεση δοκιμών, την κατασκευή εφαρμογών και την άμεση παράδοση νέων χαρακτηριστικών στους χρήστες. Οι διαδικασίες αυτές περιλαμβάνουν τη συνεχή ενσωμάτωση κώδικα, την αυτόματη δημιουργία **containers** με εργαλεία όπως **Docker** και την κλιμάκωση σε πλατφόρμες cloud, όπως **AWS**, **Azure** και **Google Cloud Platform**. Το **CI/CD** μειώνει τον χρόνο ανάπτυξης, βελτιώνει την ποιότητα του λογισμικού και επιτρέπει την ταχύτερη απόκριση στις ανάγκες των χρηστών.

Προοπτικές και Τάσεις στην Εξέλιξη των Backend Τεχνολογιών

Οι τεχνολογίες backend αποτελούν τη ραχοκοκαλιά των σύγχρονων διαδικτυακών εφαρμογών, παρέχοντας την απαραίτητη υποδομή για την επεξεργασία δεδομένων, την ασφάλεια και την κλιμάκωση. Καθώς οι απαιτήσεις για υψηλή απόδοση και αξιοπιστία αυξάνονται, οι προγραμματιστές πρέπει να επιλέγουν τεχνολογίες που επιτρέπουν την ταχεία ανάπτυξη και τη σταθερή λειτουργία. Η μετάβαση σε αρχιτεκτονικές *microservices*, η χρήση *serverless* μοντέλων και η υιοθέτηση *API-first* προσεγγίσεων είναι μερικές από τις τάσεις που αναμένεται να κυριαρχήσουν στο μέλλον. Επιπλέον, η συνεχής βελτίωση της ασφάλειας και η ενσωμάτωση τεχνητής νοημοσύνης για τη διαχείριση των δεδομένων θα παίξουν κρίσιμο ρόλο στην εξέλιξη του backend.

Τεχνολογίες Αποθήκευσης Δεδομένων

Αρχιτεκτονικές Βάσεων Δεδομένων

Οι βάσεις δεδομένων χρησιμοποιούνται για την αποθήκευση, την οργάνωση και την ανάκτηση πληροφοριών σε διαδικτυακές εφαρμογές. Οι αρχιτεκτονικές βάσεων δεδομένων ποικίλλουν ανάλογα με τη δομή των δεδομένων και τις απαιτήσεις της εφαρμογής. Οι πιο κοινές αρχιτεκτονικές περιλαμβάνουν **Σχεσιακές Βάσεις Δεδομένων (SQL)**, **Μη Σχεσιακές Βάσεις Δεδομένων (NoSQL)**, **Column-Family Stores**, **Document Stores**, **Key-Value Stores** και **Graph Databases**. Οι σχεσιακές βάσεις χρησιμοποιούν πίνακες με καλά καθορισμένα σχήματα (*schemas*) και υποστηρίζουν πολύπλοκες σχέσεις μεταξύ των δεδομένων, ενώ οι μη σχεσιακές βάσεις είναι πιο ευέλικτες και προσαρμόσιμες για ημιδομημένα ή μη δομημένα δεδομένα. Επιπλέον, οι βάσεις δεδομένων μπορεί να είναι **distributed** ή **centralized**, ανάλογα με το αν τα δεδομένα είναι αποθηκευμένα σε ένα μόνο server ή καταναμημένα σε πολλαπλά nodes. Η επιλογή της κατάλληλης αρχιτεκτονικής εξαρτάται από τις ανάγκες της εφαρμογής, την κλίμακα των δεδομένων και τις απαιτήσεις απόδοσης.

Σχεσιακές Βάσεις Δεδομένων (SQL)

Οι **Σχεσιακές Βάσεις Δεδομένων (SQL)** (Munzert, Christian, Meißner, & Nyhuis, 2015) είναι οι πιο παραδοσιακές και διαδεδομένες μορφές βάσεων δεδομένων, οι οποίες χρησιμοποιούν δομημένα σχήματα (*schemas*) και οργανώνουν τα δεδομένα σε πίνακες με σειρές και στήλες. Κάθε πίνακας αντιπροσωπεύει μια οντότητα, όπως χρήστες, προϊόντα ή παραγγελίες, και οι σχέσεις μεταξύ των πινάκων καθορίζονται μέσω **Primary Keys** και **Foreign Keys**. Οι σχεσιακές βάσεις δεδομένων χρησιμοποιούν τη γλώσσα **SQL (Structured Query Language)** για την εκτέλεση ερωτημάτων, την εισαγωγή, την ενημέρωση και τη διαγραφή δεδομένων. Παραδείγματα σχεσιακών βάσεων δεδομένων περιλαμβάνουν τα **MySQL**, **PostgreSQL**, **Microsoft SQL Server** και **Oracle Database**. Αυτά τα συστήματα προσφέρουν υψηλή αξιοπιστία, συνέπεια και ακεραιότητα δεδομένων, γεγονός που τα καθιστά ιδανικά για εφαρμογές που απαιτούν σύνθετες αναφορές και συναλλαγές, όπως τραπεζικά συστήματα, ERP και εφαρμογές διαχείρισης πελατών (CRM).

Μη Σχεσιακές Βάσεις Δεδομένων (NoSQL)

Οι **Μη Σχεσιακές Βάσεις Δεδομένων (NoSQL)** (Borad, Tejash, & Mehta, 2017) είναι σχεδιασμένες για να προσφέρουν μεγαλύτερη ευελιξία και κλιμάκωση

σε σύγκριση με τις παραδοσιακές σχεσιακές βάσεις. Αντί να χρησιμοποιούν πίνακες με προκαθορισμένα σχήματα, οι NoSQL βάσεις αποθηκεύουν δεδομένα σε μορφές όπως έγγραφα, key-value ζεύγη, γραφήματα και στήλες. Αυτή η προσέγγιση επιτρέπει την αποθήκευση ημιδομημένων ή μη δομημένων δεδομένων, όπως JSON αρχεία, χωρίς την ανάγκη για προκαθορισμένα σχήματα. Δημοφιλή παραδείγματα NoSQL βάσεων δεδομένων περιλαμβάνουν το **MongoDB** (document-based), το **Redis** και το **Cassandra** (key-value), καθώς και το **Neo4j** (graph-based). Οι NoSQL βάσεις είναι ιδανικές για εφαρμογές με υψηλές απαιτήσεις κλιμάκωσης, real-time analytics και μεγάλα σύνολα δεδομένων, όπως πλατφόρμες κοινωνικών μέσων, συστήματα IoT και εφαρμογές big data. Ωστόσο, συχνά θυσιάζουν την ισχυρή συνέπεια για να επιτύχουν υψηλή διαθεσιμότητα και ταχύτητα ανάγνωσης/εγγραφής.

Διάσημες Σύγχρονες Τεχνολογικές Στοιβές

MERN (MongoDB, Express, React, Node.js)

Η στοίβα **MERN** (Iguenfer, Benhayoun, & Bouchaib, 2024) αποτελεί μία από τις πιο δημοφιλείς τεχνολογικές στοιβές για την ανάπτυξη σύγχρονων διαδικτυακών εφαρμογών. Περιλαμβάνει το **MongoDB** για την αποθήκευση των δεδομένων, το **Express.js** για τη διαχείριση του backend, το **React** για το frontend και το **Node.js** για την εκτέλεση του server-side κώδικα σε **JavaScript**. Το κύριο πλεονέκτημα της **MERN** στοίβας είναι ότι όλες οι τεχνολογίες που την απαρτίζουν χρησιμοποιούν **JavaScript**, επιτρέποντας στους προγραμματιστές να αναπτύσσουν τόσο τον client όσο και τον server κώδικα με μία μόνο γλώσσα. Αυτό απλοποιεί την ανάπτυξη, τη συντήρηση και την κλιμάκωση των εφαρμογών. Επιπλέον, το React προσφέρει ταχύτητα και απόδοση μέσω της χρήσης ενός εικονικού **DOM**, ενώ το **MongoDB** παρέχει ευελιξία στην αποθήκευση ημιδομημένων δεδομένων. Η **MERN** στοίβα είναι ιδανική για εφαρμογές **Single Page (SPA)** και εφαρμογές με υψηλές απαιτήσεις διαδραστικότητας.

MEAN (MongoDB, Express, Angular, Node.js)

Η στοίβα **MEAN** (Iguenfer, Benhayoun, & Bouchaib, 2024) είναι παρόμοια με τη **MERN** στοίβα, αλλά χρησιμοποιεί το **Angular** αντί για το **React** στο frontend. Το **Angular** είναι ένα ισχυρό framework που αναπτύχθηκε από την **Google** και προσφέρει δομημένη αρχιτεκτονική και ισχυρά εργαλεία για την ανάπτυξη εφαρμογών μεγάλης κλίμακας. Η στοίβα **MEAN** περιλαμβάνει επίσης το **MongoDB** για την αποθήκευση δεδομένων, το **Express.js** για τη διαχείριση του server-side λογισμικού και το **Node.js** για την εκτέλεση **JavaScript** στον server. Η **MEAN** στοίβα είναι γνωστή για την απόδοσή της και την ενσωμάτωσή της με το **Angular**, το οποίο προσφέρει ισχυρές δυνατότητες διαχείρισης της κατάστασης της εφαρμογής, **data binding** και **dependency injection**. Είναι ιδανική για εφαρμογές που απαιτούν σύνθετες διαδρομές χρηστών και εφαρμογές με υψηλό βαθμό αλληλεπίδρασης.

.NET Stack (C#, ASP.NET, Microsoft SQL Server)

Η **.NET Stack** είναι μία στοίβα τεχνολογιών της **Microsoft**, η οποία περιλαμβάνει την γλώσσα προγραμματισμού **C#**, το framework **ASP.NET** για την ανάπτυξη εφαρμογών και τον **Microsoft SQL Server** για την αποθήκευση δεδομένων. Η στοίβα αυτή είναι ιδιαίτερα δημοφιλής σε εταιρικά περιβάλλοντα

και οργανισμούς που χρησιμοποιούν το οικοσύστημα της **Microsoft**. Το **ASP.NET** προσφέρει ένα ισχυρό και ασφαλές περιβάλλον για την ανάπτυξη διαδικτυακών εφαρμογών με υψηλές απαιτήσεις ασφάλειας και επιδόσεων. Χρησιμοποιεί **server-side rendering** για την ταχύτερη παράδοση περιεχομένου και υποστηρίζει μοντέλα ανάπτυξης όπως **MVC (Model-View-Controller)** και **Web API**. Ο **Microsoft SQL Server** παρέχει σταθερότητα και ισχυρές δυνατότητες ανάλυσης δεδομένων, καθιστώντας αυτή τη στοίβα κατάλληλη για επιχειρηματικές εφαρμογές και συστήματα διαχείρισης μεγάλου όγκου δεδομένων.

Django Stack (Python, Django, PostgreSQL)

Η **Django Stack** είναι μία από τις πιο δημοφιλείς επιλογές για την ανάπτυξη διαδικτυακών εφαρμογών με **Python**. Το **Django** είναι ένα υψηλού επιπέδου web **framework** που προσφέρει ένα ολοκληρωμένο σύνολο εργαλείων και βιβλιοθηκών για την ανάπτυξη ασφαλών και επεκτάσιμων εφαρμογών. Χρησιμοποιεί την αρχιτεκτονική **MVC (Model-View-Controller)** και παρέχει έτοιμα εργαλεία για authentication, authorization, **ORM (Object-Relational Mapping)** και form handling. Η βάση δεδομένων **PostgreSQL** συνδυάζεται συχνά με το **Django** λόγω της ισχυρής υποστήριξης για σύνθετα queries και data integrity. Η **Django Stack** είναι ιδανική για εφαρμογές που απαιτούν υψηλό επίπεδο ασφάλειας και ταχύτητα ανάπτυξης, όπως πλατφόρμες κοινωνικών μέσων, ηλεκτρονικά καταστήματα και συστήματα διαχείρισης περιεχομένου.

Ruby on Rails Stack (Ruby, Rails, PostgreSQL / MySQL)

Η στοίβα **Ruby on Rails** χρησιμοποιεί τη γλώσσα προγραμματισμού **Ruby** και το framework **Rails** για την ανάπτυξη εφαρμογών διαδικτύου. Το Rails είναι γνωστό για την προσέγγιση "**Convention over Configuration**" και την αρχή **DRY (Don't Repeat Yourself)**, που επιταχύνει την ανάπτυξη και μειώνει την πολυπλοκότητα του κώδικα. Οι βάσεις δεδομένων που χρησιμοποιούνται συχνά με το Rails είναι οι **PostgreSQL** και **MySQL**, οι οποίες παρέχουν σταθερότητα και επεκτασιμότητα για εφαρμογές μεγάλης κλίμακας. Η στοίβα αυτή είναι γνωστή για την ευκολία χρήσης της, την καθαρότητα του κώδικα και την ισχυρή κοινότητα υποστήριξης, καθιστώντας την ιδανική για startups και ταχέως αναπτυσσόμενες εφαρμογές. Διάσημες πλατφόρμες όπως το **GitHub**, το **Airbnb** και το **Shopify** έχουν αναπτυχθεί χρησιμοποιώντας αυτή τη στοίβα.

ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ELENi FORMS

Περιγραφή

Η εφαρμογή **Eleni Forms** είναι μια διαδικτυακή πλατφόρμα για τη δημιουργία, διαχείριση και απάντηση σε ερωτηματολόγια πολλαπλής επιλογής. Σχεδιάστηκε για να καλύψει τις ανάγκες εκπαιδευτικών, εταιρειών και οργανισμών που θέλουν να αξιολογήσουν τις γνώσεις και τις δεξιότητες των χρηστών τους. Η εφαρμογή υποστηρίζει δύο κύριους ρόλους χρηστών: **Admins** και **Regular Users**. Οι Admins έχουν τη δυνατότητα να δημιουργούν και να διαχειρίζονται quiz, ερωτήσεις και ομάδες χρηστών (Groups), ενώ οι Regular Users μπορούν να συμμετέχουν σε quiz και να υποβάλλουν απαντήσεις. Κάθε κουίζ μπορεί να είναι δημόσιο ή ιδιωτικό και να σχετίζεται με μια συγκεκριμένη ομάδα χρηστών. Οι απαντήσεις καταγράφονται σε πραγματικό χρόνο και οι βαθμολογίες υπολογίζονται αυτόματα, παρέχοντας άμεσα στατιστικά αποτελέσματα. Το Eleni Forms είναι κατασκευασμένο ως **Single Page Application (SPA)** με **React** για το frontend, **Express.js** για το backend και **MongoDB** για την αποθήκευση δεδομένων, προσφέροντας μια ταχύτερη και πιο διαδραστική εμπειρία χρήστη.

Επιλογή τεχνολογιών για την ανάπτυξη της εφαρμογής

Τεχνολογική στοίβα MERN

Η τεχνολογική στοίβα **MERN (MongoDB, Express.js, React.js, Node.js)** επιλέχθηκε για την ανάπτυξη της εφαρμογής **Eleni Forms** λόγω της συνοχής της και της ευελιξίας που προσφέρει για τη δημιουργία πλήρως λειτουργικών, μοντέρνων διαδικτυακών εφαρμογών. Η MERN στοίβα βασίζεται αποκλειστικά σε JavaScript, επιτρέποντας στους προγραμματιστές να χρησιμοποιούν την ίδια γλώσσα προγραμματισμού τόσο στο frontend όσο και στο backend, μειώνοντας έτσι την πολυπλοκότητα της ανάπτυξης και της συντήρησης. Η στοίβα αυτή περιλαμβάνει το **MongoDB** για την αποθήκευση δεδομένων, το **Express.js** για τη διαχείριση των server-side λογικών και των REST APIs, το **React.js** για τη δημιουργία του δυναμικού frontend και το **Node.js** για την εκτέλεση server-side JavaScript κώδικα. Αυτή η ομοιομορφία στη χρήση γλώσσας διευκολύνει την ταχύτερη ανάπτυξη, την καλύτερη ενσωμάτωση των συστημάτων και την επεκτασιμότητα της εφαρμογής. Επιπλέον, το MERN είναι ιδανικό για εφαρμογές **Single Page Application (SPA)**, όπως το Eleni Forms, καθώς υποστηρίζει την ανταλλαγή δεδομένων σε πραγματικό χρόνο και τη δημιουργία διαδραστικών διεπαφών.

Οφέλη των Επιλεγμένων Τεχνολογιών

Οι τεχνολογίες που επιλέχθηκαν για την ανάπτυξη του **Eleni Forms** προσφέρουν πολλά πλεονεκτήματα σε όρους ταχύτητας, ευελιξίας και επεκτασιμότητας. Η χρήση του **MongoDB** για την αποθήκευση δεδομένων προσφέρει μεγάλη ευελιξία στην αποθήκευση ημιδομημένων δεδομένων, επιτρέποντας την εύκολη προσθήκη νέων πεδίων χωρίς την ανάγκη αναδιάρθρωσης του σχήματος. Το **Express.js**

απλοποιεί τη δημιουργία REST APIs και τη διαχείριση αιτημάτων από το frontend, προσφέροντας υψηλή απόδοση και ευκολία στην ενσωμάτωση middleware για authentication και data validation. Το **React.js** είναι γνωστό για την ταχύτητα του και την επαναχρησιμοποίηση components, βελτιώνοντας την εμπειρία χρήστη μέσω δυναμικών UI και γρήγορων αλληλεπιδράσεων. Τέλος, το **Node.js** επιτρέπει την εκτέλεση JavaScript στον server, μειώνοντας την πολυπλοκότητα του έργου και επιταχύνοντας την ανάπτυξη μέσω ενός ενιαίου οικοσυστήματος εργαλείων και βιβλιοθηκών.

MongoDB

Η **MongoDB** επιλέχθηκε ως η βάση δεδομένων για την εφαρμογή **Eleni Forms** λόγω της ευελιξίας και της επεκτασιμότητάς της. Ως NoSQL βάση δεδομένων, το MongoDB χρησιμοποιεί ένα μοντέλο αποθήκευσης εγγράφων (document-based), όπου τα δεδομένα αποθηκεύονται σε μορφή JSON, καθιστώντας το ιδανικό για εφαρμογές που απαιτούν γρήγορη και ευέλικτη αποθήκευση ημιδομημένων δεδομένων. Αυτή η προσέγγιση επιτρέπει την εύκολη προσθήκη νέων χαρακτηριστικών και πεδίων χωρίς την ανάγκη αλλαγής του σχήματος της βάσης. Επιπλέον, το MongoDB προσφέρει υψηλή απόδοση σε εγγραφές και αναζητήσεις, καθώς και ενσωματωμένες δυνατότητες κλιμάκωσης μέσω sharding και replication. Για την εφαρμογή Eleni Forms, το MongoDB χρησιμοποιείται για την αποθήκευση πληροφοριών σχετικά με χρήστες, ομάδες, κουίζ, ερωτήσεις και απαντήσεις, επιτρέποντας την ταχεία ανάκτηση και επεξεργασία μεγάλων όγκων δεδομένων.

Express.js

Το **Express.js** είναι ένα ελαφρύ και ευέλικτο framework για την ανάπτυξη server-side λογικής σε εφαρμογές Node.js. Επιλέχθηκε για το backend της εφαρμογής **Eleni Forms** λόγω της απλότητας, της απόδοσης και της εκτεταμένης κοινότητας υποστήριξής του. Το Express προσφέρει μια απλή αλλά ισχυρή προσέγγιση για τη διαχείριση HTTP αιτημάτων, τη δημιουργία REST APIs και την ενσωμάτωση middleware για authentication, authorization και data validation. Η χρήση του Express επιτρέπει στους προγραμματιστές να δημιουργούν καθαρές και modular αρχιτεκτονικές, διευκολύνοντας τη συντήρηση και την επέκταση της εφαρμογής. Επιπλέον, το Express παρέχει εργαλεία για τον χειρισμό ασφαμάτων και τη διαχείριση ασύγχρονων λειτουργιών, καθιστώντας το ιδανικό για εφαρμογές με υψηλές απαιτήσεις απόδοσης και ασφάλειας, όπως το Eleni Forms, όπου η προστασία των προσωπικών δεδομένων των χρηστών είναι κρίσιμη.

React.js

Το **React.js** χρησιμοποιήθηκε για την ανάπτυξη του frontend της εφαρμογής **Eleni Forms** λόγω της ταχύτητας και της ευελιξίας που προσφέρει για τη δημιουργία δυναμικών και διαδραστικών διεπαφών χρήστη. Το React επιτρέπει τη δημιουργία επαναχρησιμοποιήσιμων components, βελτιώνοντας την αναγνωσιμότητα του κώδικα και μειώνοντας την πολυπλοκότητα της ανάπτυξης. Ένα από τα κύρια πλεονεκτήματά του είναι η χρήση του **Virtual DOM**, το οποίο βελτιώνει σημαντικά την απόδοση της εφαρμογής, καθώς ενημερώνει μόνο τα απαραίτητα τμήματα του DOM όταν αλλάζουν τα δεδομένα. Το React είναι επίσης ιδανικό για την ανάπτυξη Single Page Applications (SPA), καθώς υποστηρίζει

γρήγορες μεταβάσεις μεταξύ σελίδων χωρίς την ανάγκη πλήρους επαναφόρτωσης. Για την εφαρμογή Eleni Forms, το React χρησιμοποιείται σε συνδυασμό με hooks όπως **useState** και **useEffect** για την καλύτερη διαχείριση της κατάστασης (state) και την αλληλεπίδραση με το backend μέσω Axios.

Node.js

Το **Node.js** επιλέχθηκε ως η πλατφόρμα για την ανάπτυξη του backend της εφαρμογής **Eleni Forms** λόγω της απόδοσης και της ευελιξίας του. Το Node.js επιτρέπει την εκτέλεση JavaScript στον server, προσφέροντας μια ενιαία γλώσσα προγραμματισμού τόσο για το frontend όσο και για το backend, κάτι που απλοποιεί την ανάπτυξη και μειώνει την πολυπλοκότητα της εφαρμογής. Επιπλέον, το Node.js βασίζεται σε ένα ασύγχρονο, event-driven μοντέλο εισόδου/εξόδου, το οποίο το καθιστά εξαιρετικά γρήγορο και αποτελεσματικό για εφαρμογές με πολλές ταυτόχρονες συνδέσεις. Αυτό είναι ιδιαίτερα σημαντικό για την εφαρμογή Eleni Forms, όπου η ταχύτητα απόκρισης και η δυνατότητα χειρισμού πολλών χρηστών ταυτόχρονα είναι κρίσιμες. Το Node.js υποστηρίζει επίσης μια εκτεταμένη βιβλιοθήκη πακέτων μέσω του **npm**, επιτρέποντας την εύκολη προσθήκη λειτουργιών όπως authentication, routing και data validation.

Στάδια Ανάπτυξης της Εφαρμογής

Ανάλυση

Η ανάπτυξη της εφαρμογής **Eleni Forms** ξεκίνησε με την ανάλυση των απαιτήσεων και των λειτουργικών αναγκών των χρηστών. Σε αυτό το στάδιο, προσδιορίστηκαν οι κύριες λειτουργίες της εφαρμογής, όπως η δημιουργία και διαχείριση quiz, η υποβολή απαντήσεων και η οργάνωση χρηστών σε ομάδες. Η ανάλυση περιλάμβανε επίσης τη μελέτη των ρόλων χρηστών (Admins και Regular Users) και των διαφορετικών επιπέδων πρόσβασης που απαιτούνται για τη διασφάλιση της ασφάλειας των δεδομένων. Εξετάστηκαν επίσης οι απαιτήσεις για την ταχύτητα, την επεκτασιμότητα και την εμπειρία χρήστη (UX), καθώς και οι επιλογές για την τεχνολογική στοίβα που θα χρησιμοποιηθεί. Η διαδικασία αυτή επέτρεψε τη σαφή καθορισμό των στόχων του έργου και τη δημιουργία μιας βάσης για τα επόμενα στάδια της ανάπτυξης.

Σχεδίαση

Η σχεδίαση της εφαρμογής **Eleni Forms** περιλάμβανε τη δημιουργία των δομών δεδομένων, των διαδρομών (routes) και των διεπαφών χρήστη (UI). Σε αυτό το στάδιο σχεδιάστηκαν τα σχήματα των βάσεων δεδομένων για χρήστες, κουίζ, ομάδες και απαντήσεις, λαμβάνοντας υπόψη τη δομή των εγγράφων σε MongoDB. Επιπλέον, σχεδιάστηκαν τα REST APIs για την επικοινωνία μεταξύ frontend και backend, καθώς και τα κύρια components της React εφαρμογής, όπως τα forms για τη δημιουργία quiz και οι πίνακες για την προβολή στατιστικών. Η σχεδίαση περιλάμβανε επίσης την επιλογή του χρωματικού σχήματος και των στυλ, καθώς και τη δημιουργία wireframes για τις κύριες σελίδες της εφαρμογής. Στόχος ήταν η δημιουργία μιας φιλικής προς τον χρήστη διεπαφής που να είναι ταυτόχρονα λειτουργική και αισθητικά ευχάριστη.

Υλοποίηση

Η υλοποίηση της εφαρμογής **Eleni Forms** περιλάμβανε την ανάπτυξη του frontend με **React.js**, του backend με **Express.js** και της βάσης δεδομένων με **MongoDB**. Το frontend δημιουργήθηκε ως **Single Page Application (SPA)**, με χρήση React components για την επαναχρησιμοποίηση κώδικα και την ταχύτερη ανάπτυξη. Το backend δημιουργήθηκε με Node.js και Express.js για τη διαχείριση των αιτημάτων από τον πελάτη, ενώ η βάση δεδομένων σχεδιάστηκε με MongoDB για την αποθήκευση ημιδομημένων δεδομένων. Οι λειτουργίες αυθεντικοποίησης, διαχείρισης ομάδων και δημιουργίας quiz υλοποιήθηκαν με τη χρήση middleware για την προστασία των endpoints και την επικύρωση των δεδομένων. Η εφαρμογή σχεδιάστηκε για να είναι επεκτάσιμη και εύκολη στη συντήρηση, με καθαρή αρχιτεκτονική και modular προσέγγιση.

Δοκιμές

Οι δοκιμές της εφαρμογής **Eleni Forms** περιλάμβαναν τη διεξοδική αξιολόγηση της λειτουργικότητας, της ασφάλειας και της απόδοσης του συστήματος. Χρησιμοποιήθηκαν τεχνικές manual testing για την επαλήθευση της σωστής λειτουργίας των βασικών χαρακτηριστικών, όπως η δημιουργία και διαχείριση quiz, η εγγραφή χρηστών και η αποστολή απαντήσεων. Επιπλέον, πραγματοποιήθηκαν δοκιμές ασφαλείας για να διασφαλιστεί η προστασία των δεδομένων και η αποτροπή κακόβουλων επιθέσεων, όπως SQL Injection και XSS. Η σταθερότητα της εφαρμογής ελέγχθηκε με τη χρήση εργαλείων παρακολούθησης απόδοσης και καταγραφής σφαλμάτων, ενώ τα τελικά αποτελέσματα αξιολογήθηκαν με βάση την ταχύτητα απόκρισης και τη δυνατότητα ταυτόχρονης διαχείρισης μεγάλου αριθμού χρηστών.

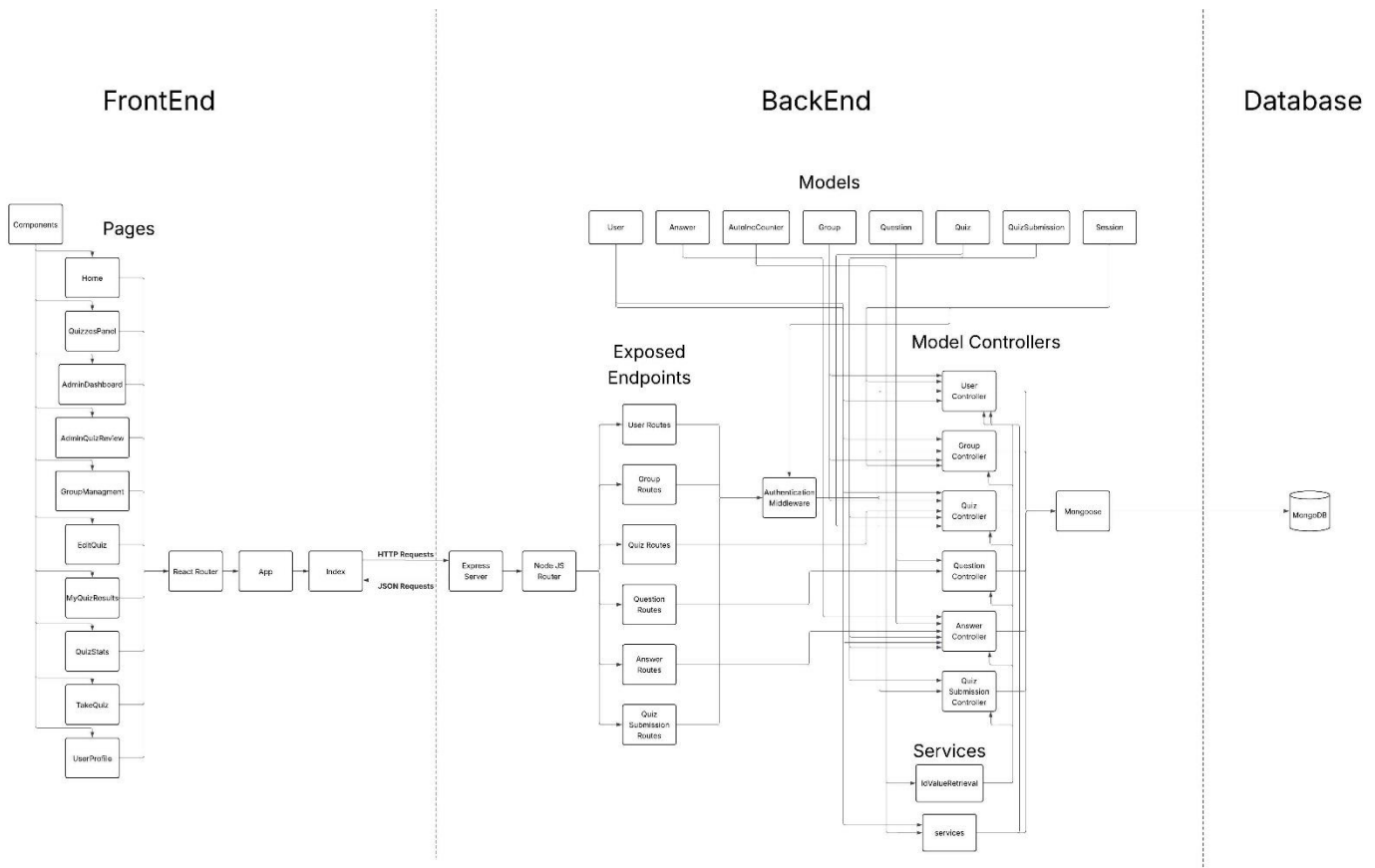
Υλοποίηση της εφαρμογής

Αρχιτεκτονική

Η εφαρμογή **Eleni Forms** σχεδιάστηκε με βάση το μοντέλο **MVC (Model-View-Controller)** για το backend και μια αρχιτεκτονική **Single Page Application (SPA)** για το frontend, επιτυγχάνοντας υψηλή επεκτασιμότητα, ευκολία στη συντήρηση και ταχύτητα απόκρισης. Το **MVC** είναι ένα δημοφιλές αρχιτεκτονικό πρότυπο που διαχωρίζει τη λογική της εφαρμογής σε τρία κύρια μέρη: **Model**, **View** και **Controller**. Στο **backend**, τα **Models** είναι υπεύθυνα για τον ορισμό των σχημάτων της βάσης δεδομένων και την επικοινωνία με τη MongoDB μέσω του **Mongoose ODM**. Οι **Controllers** περιέχουν τη λογική της εφαρμογής και είναι υπεύθυνοι για την επεξεργασία των αιτημάτων από τους πελάτες και την αποστολή απαντήσεων. Τέλος, τα **Views** δεν είναι μέρος του backend, καθώς η εφαρμογή είναι σχεδιασμένη ως SPA, με το React να αναλαμβάνει την απόδοση των διεπαφών χρήστη.

Το **frontend** είναι αναπτυγμένο σε **React.js**, το οποίο προσφέρει ταχύτητα και διαδραστικότητα μέσω του **Virtual DOM** και της modular αρχιτεκτονικής με components. Η διαχείριση της κατάστασης (state) γίνεται με hooks, όπως **useState** και **useEffect**, ενώ το **React Router** χρησιμοποιείται για την πλοήγηση μεταξύ των σελίδων χωρίς την ανάγκη πλήρους επαναφόρτωσης. Αυτή η προσέγγιση επιτρέπει την ταχύτερη ανάπτυξη και τη δημιουργία καθαρών, επαναχρησιμοποιήσιμων στοιχείων, βελτιώνοντας την εμπειρία χρήστη και την απόδοση της εφαρμογής.

Η γενική ροή της εφαρμογής ακολουθεί την πληροφορία από τις διάφορες όψεις στο frontend, μέσω του react router και μέσω **HTTP Requests** προς τον **server** όπου εκεί δρομολογείται το κάθε αίτημα ανάλογα με την εκτεθειμένη επαφή που ζητάει (**endpoint**) και ανατίθεται στον ανάλογο ελεγκτή για διεκπεραίωση και αποστολή της απάντησης προς την όψη. Παρακάτω ακολουθεί το συλλογικό διάγραμμα διασύνδεσης των μερών της εφαρμογής



Frontend

Το frontend της εφαρμογής **Eleni Forms** υλοποιήθηκε με **React.js** για τη δημιουργία δυναμικών και διαδραστικών διεπαφών χρήστη. Χρησιμοποιήθηκαν επαναχρησιμοποιήσιμα components για την οργάνωση της διεπαφής, όπως buttons, forms και tables, ενώ η διαχείριση της κατάστασης (state) επιτεύχθηκε με τη χρήση hooks, όπως **useState**, **useEffect** και **useContext**. Για την επικοινωνία με το backend, χρησιμοποιήθηκε το **Axios** για την αποστολή αιτημάτων HTTP και την επεξεργασία των απαντήσεων από τον server. Επιπλέον, χρησιμοποιήθηκε το **React Router** για την πλοήγηση μεταξύ των σελίδων χωρίς την ανάγκη πλήρους επαναφόρτωσης, βελτιώνοντας την εμπειρία χρήστη. Το styling έγινε με τη χρήση **Bootstrap** και προσαρμοσμένων CSS styles για τη δημιουργία responsive σελίδων που προσαρμόζονται σε διαφορετικά μεγέθη οθόνης.

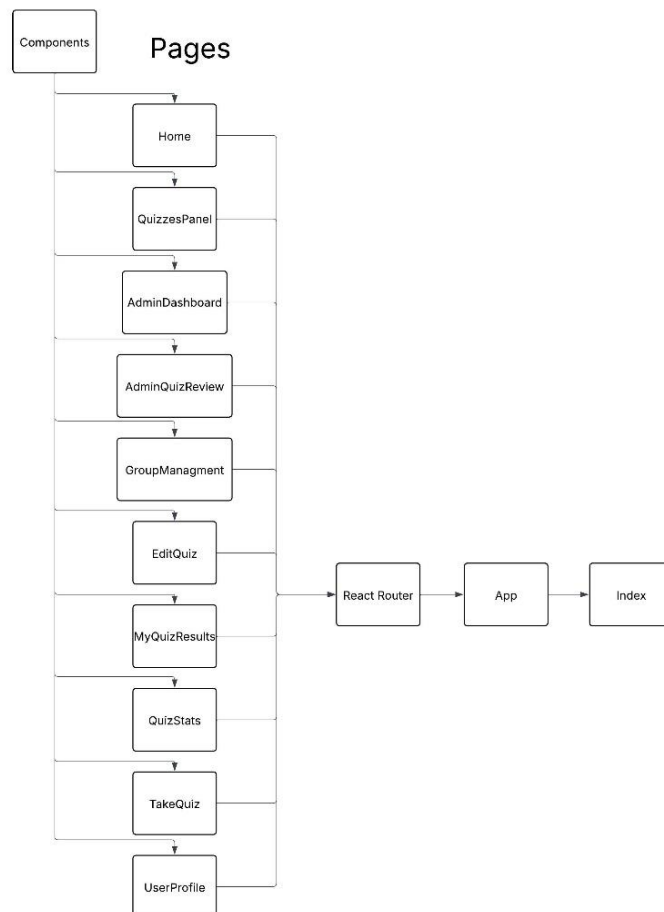
Η δομή του frontend έχει σαν κεντρικό σημείο αναφοράς τις διαφορετικές όψεις της εφαρμογής. Μέσω της βοήθειας του δρομολογητή της React, ο χρήστης μπορεί να περιηγηθεί μέσα στην εφαρμογή δίχως καθυστερήσεις για φόρτωση της κάθε όψης. Η συγκρότηση των κατάλληλων στοιχείων σε κάθε όψη γίνεται

δυναμικά ανάλογα με τον τύπο χρήστη έτσι ώστε να αποφευχθεί η δημιουργία πολλαπλών αρχείων για την ίδια όψη ανάλογα με τον χρήστη. Μαζί συγκροτούνται και τα κοινά στοιχεία όπως η γραμμή πλοήγησης.

Για την ασφάλεια της εφαρμογής, οι όψεις χωρίζονται σε προστατευμένες και μη. Προστατευμένες χαρακτηρίζονται οι όψεις της εφαρμογής οι οποίες μπορούν να προσπελαθούν μετά την επιτυχή είσοδο ενός χρήστη και άρα απαιτούν την ταυτοποίηση και αυθεντικοποίηση του χρήστη για να γίνουν ορατές. Ο χαρακτηρισμός αυτός σε επίπεδο υλοποίησης αποτρέπει την δυνατότητα κατάχρησης του πλήκτρου φόρτωσης επόμενης ή προηγούμενης σελίδας από τον φυλλομετρητή για μη εξουσιοδοτημένη πρόσβαση στην εφαρμογή. Σαν πρόσθετο μέτρο ασφαλείας, κατά την αποσύνδεση ενός χρήστη, ο δρομολογητής της React αρχικοποιείται καθιστώντας έτσι αδύνατη την φόρτωση προηγούμενων όψεων της εφαρμογής.

Η γενική δομή και οργάνωση του frontend της εφαρμογής **Eleni Forms** μπορεί να παρατηρηθεί και από το ακόλουθο σχεδιάγραμμα. Επιπρόσθετα η ροή της πληροφορίας μιας ενέργειας θα παρουσιαστεί μέσω της διαδικασίας δημιουργίας ενός νέου χρήστη.

Frontend



```

function RegisterModal({ show, onHide }) {
  const [form, setForm] = useState({
    name: "",
    surname: "",
    email: "",
    password: "",
    confirmPassword: "",
    phone: "",
  });

  const [alert, setAlert] = useState(null);
  const [showPassword, setShowPassword] = useState(false);
  const [showConfirm, setShowConfirm] = useState(false);

  const handleChange = (field, value) => {
    setForm({ ...form, [field]: value });
  };

  const handleSubmit = async () => {
    const { name, surname, email, password, confirmPassword, phone } = form;

    if (
      !name ||
      !surname ||
      !email ||
      !password ||
      !confirmPassword ||
      !phone
    ) {
      setAlert({ type: "danger", message: "Όλα τα πεδία είναι υποχρεωτικά." });
      return;
    }

    if (password !== confirmPassword) {
      setAlert({ type: "danger", message: "Οι κωδικοί δεν ταιριάζουν." });
      return;
    }

    try {
      const res = await api.post("/users/register", {
        name,
        surname,
        email,
        password,
        phone,
      });

      setAlert({
        type: "success",
        message: "Επιτυχής εγγραφή! Μπορείτε να συνδεθείτε.",
      });

      setTimeout(() => {
        setAlert(null);
        onHide();
      }, 1500);
    } catch (err) {
      const msg =
        err?.response?.data?.message || "Αποτυχία εγγραφής. Δοκιμάστε ξανά.";
      setAlert({ type: "danger", message: msg });
    }
  };
}

```

Συνάρτηση αποστολής δεδομένων νέου χρήστη

Backend

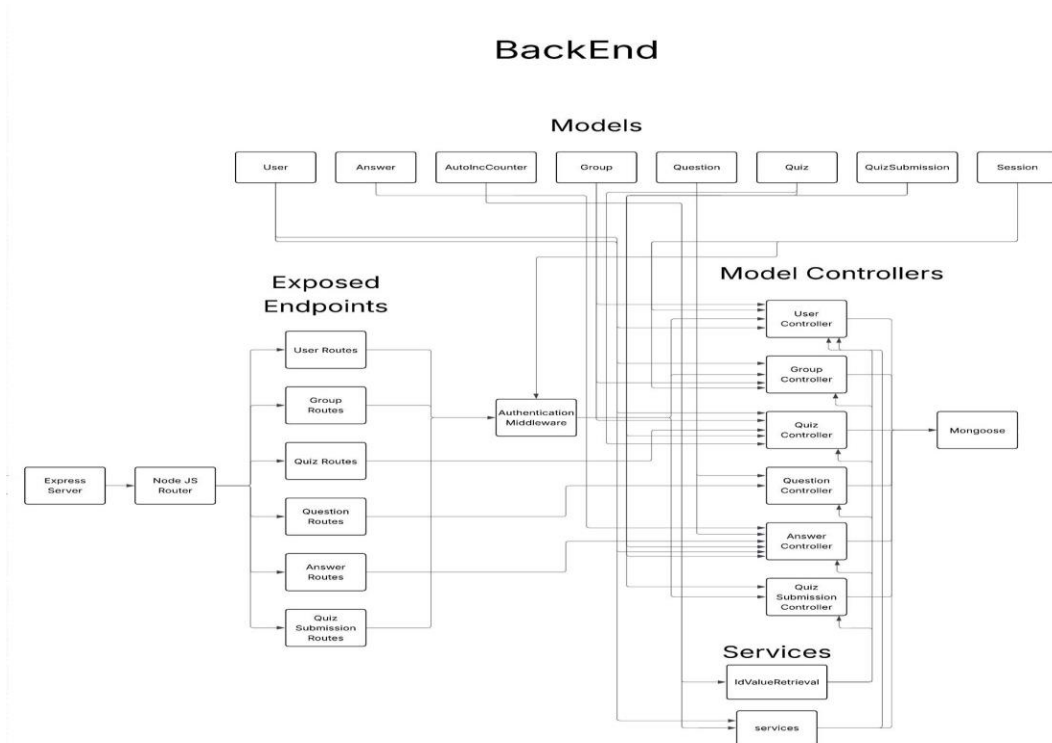
Το backend της εφαρμογής **Eleni Forms** αναπτύχθηκε με **Node.js** και **Express.js**, επιτρέποντας την ταχεία και αποτελεσματική ανάπτυξη REST APIs για την επικοινωνία με το frontend. Το Express.js χρησιμοποιήθηκε για τη διαχείριση των routes και τη δημιουργία middleware για την αυθεντικοποίηση, την επικύρωση δεδομένων και τον χειρισμό σφαλμάτων. Τα session tokens χρησιμοποιήθηκαν για την προστασία των endpoints και την ταυτοποίηση των χρηστών. Οι controllers οργανώθηκαν σε ξεχωριστά αρχεία για τη διαχείριση χρηστών, κουίζ, ερωτήσεων και ομάδων, διευκολύνοντας την επέκταση της εφαρμογής. Η διασύνδεση με τη βάση δεδομένων πραγματοποιήθηκε μέσω **Mongoose ODM**, το οποίο παρέχει ένα εύχρηστο API για την αλληλεπίδραση με **MongoDB**.

Κατά την λήψη ενός αιτήματος HTTP από το frontend της εφαρμογής, ο εξυπηρετητής αφού προσπελάσει το endpoint στο οποίο αναφέρεται το συγκεκριμένο αίτημα, το αποστέλλει μέσω του δρομολογητή της express προς τον κατάλληλο controller για επεξεργασία. Ορισμένα endpoints λόγω της ευαισθησίας των λειτουργιών τους τοποθετούν ένα ενδιάμεσο βήμα ταυτοποίησης και αυθεντικοποίησης στο οποίο πιστοποιείται ότι ο χρήστης που στέλνει το αίτημα έχει πραγματοποιήσει κάποια ενεργή σύνδεση σε ένα χρονικό διάστημα έως μιας ώρας πριν το αίτημα. Σε ορισμένες περιπτώσεις ταυτοποιείται ότι ο χρήστης έχει

και τα δικαιώματα που απαιτούνται για την διεκπεραίωση του αιτήματος όπως για παράδειγμα την αύξηση δικαιωμάτων ενός χρήστη σε διαχειριστή που απαιτεί την επεξεργασία μέσω ενός άλλου διαχειριστικού λογαριασμού. Αφού περαστεί αυτό το ενδιαμέσο βήμα ασφαλείας, όπου απαιτείται, το αίτημα προωθείται στον κατάλληλο controller.

Οι controllers είναι υπεύθυνοι για την επεξεργασία και πραγματοποίηση αιτημάτων του frontend. Κάθε controller αποτελεί μια συλλογή από συναρτήσεις οι οποίες έχουν την ευθύνη διεκπεραίωσης μιας συγκεκριμένης λειτουργίας μέσα στο σύνολο των δυνατών λειτουργιών και ακολουθούν μια παρόμοια αφηρημένη ροή. Στο πρώτο στάδιο μετά την ασύγχρονη παραλαβή ενός αιτήματος HTTP, η κάθε συνάρτηση ελέγχει αν τα δεδομένα που απαιτούνται για την υλοποίηση του αιτήματος είναι παρόντα στο σώμα της αίτησης όπως για παράδειγμα ο κωδικός και το email για την σύνδεση ενός χρήστη. Αν εντοπιστεί κάποια έλλειψη σε κάποιο από τα δεδομένα που απαιτούνται, αποστέλλεται κατάλληλο μήνυμα σφάλματος στον χρήστη. Αν όλα τα δεδομένα είναι παρόντα, τότε το δεύτερο στάδιο είναι η πιθανή συλλογή περεταίρω πληροφοριών που μπορεί να απαιτούνται για την υλοποίηση του αιτήματος καθώς η εφαρμογή είναι σχεδιασμένη να χρησιμοποιεί το μικρότερο δυνατό πλήθος πληροφοριών στις επικοινωνίες της μεταξύ frontend και backend για λόγους αποσυμφόρησης του δικτύου. Τέλος αφού έχουν συλλεχθεί όλα τα δεδομένα, πραγματοποιούνται οι απαιτούμενες ενέργειες στην βάση δεδομένων μέσω της mongoose και αποστέλλεται επιτυχής απάντηση στο frontend ώστε να ενημερώσει την όψη όπως χρειάζεται.

Η οργάνωση της δομής του backend καθώς και της ροής των δεδομένων από τα εισερχόμενα αιτήματα του frontend, μπορούν να παρατηρηθούν και στο ακόλουθο διάγραμμα στο οποίο αναλύεται η αρχιτεκτονική του. Παράλληλα ακολουθεί η συνέχεια του παραδείγματος της προηγούμενης ενότητας μετά το βήμα της αποστολής των δεδομένων του νέου χρήστη από το frontend μέσω της φόρμας εγγραφής.



```

const authMiddleware = require("../Middleware/authMiddleware.js");

const usersRouter = express.Router();

// --- Authentication & User Management ---
usersRouter.post("/register", registerUser);
usersRouter.post("/login", loginUser);
usersRouter.delete("/logout", authMiddleware, logoutUser);
usersRouter.post("/change-password", authMiddleware, changePassword);

// --- Password Reset Flow (No Auth Middleware Needed) ---
usersRouter.post("/forgot-password", forgotPassword);
usersRouter.post("/reset-password", resetPassword);

// --- User Profile ---
usersRouter.get("/me", authMiddleware, getUserProfileMe);
usersRouter.put("/update/:id", authMiddleware, updateUser);
usersRouter.delete("/delete/:id", authMiddleware, deleteUser);

// --- Admin Specific ---
usersRouter.get("/all", authMiddleware, getAllUsers);
usersRouter.post("/add-admin", authMiddleware, addAdmin);

```

Endpoints απτημάτων χρηστών

```

const registerUser = async (req, res) => {
  try {
    const { name, surname, email, password, phone, isActive } = req.body;

    if (!name || !surname || !email || !password || !phone) {
      return res.status(400).json({ message: "Missing required fields" });
    }

    const existingUser = await userService.findUserByEmail(email);
    if (existingUser) {
      return res.status(400).json({ message: "User already exists" });
    }

    const newUser = await userService.createUser({
      name,
      surname,
      email,
      password,
      phone,
      isActive,
    });

    res.status(201).json({ message: "User created", user: newUser });
  } catch (error) {
    console.error("User creation error:", error);
    res.status(500).json({ message: "Server error", error: error.message });
  }
};

```

Συνάρτηση δημιουργίας χρήστη

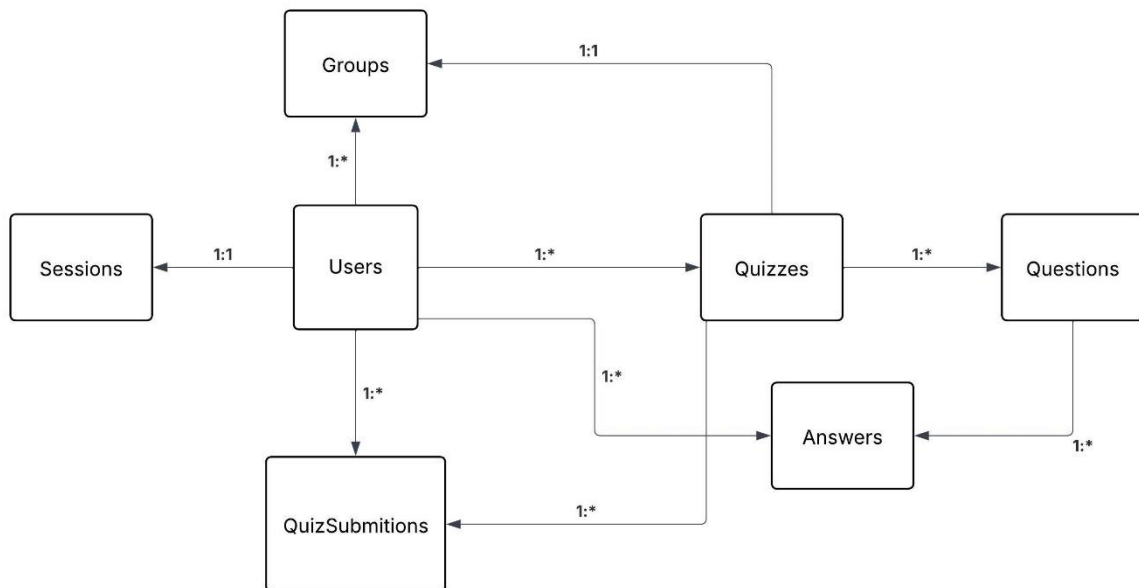
Η βάση δεδομένων της εφαρμογής **Eleni Forms** υλοποιήθηκε με **MongoDB**, μια **document-based NoSQL** βάση δεδομένων που επιτρέπει την αποθήκευση ημιδομημένων δεδομένων σε μορφή **JSON**. Τα δεδομένα οργανώθηκαν σε **collections**, όπως **Users**, **Groups**, **Quizzes**, **Questions** και **Answers**, με τη χρήση αναφορών (references) για τη σύνδεση μεταξύ των εγγράφων. Αυτό επέτρεψε την αποθήκευση σύνθετων σχέσεων μεταξύ χρηστών, ομάδων και κουίζ χωρίς την ανάγκη πολύπλοκων SQL join queries. Η βάση δεδομένων σχεδιάστηκε για να είναι επεκτάσιμη και εύκολη στη διαχείριση, με γρήγορες αναζητήσεις και αποθήκευση δεδομένων μέσω του **Mongoose**, το οποίο προσφέρει ένα ισχυρό API για τη δημιουργία και τη διαχείριση σχημάτων. Τα δύο κεντρικά μοντέλα πάνω στα οποία δημιουργούνται οι υπόλοιπες δομές είναι τα μοντέλα του χρήστη και της ομάδας.

Για την σύνδεση ενός χρήστη, δημιουργείται ένα νέο έγγραφο στην συλλογή **Sessions** στο οποίο περιέχονται το όνομα του χρήστη, ένα μοναδικό αναγνωριστικό μεγέθους 32 χαρακτήρων με το οποίο γίνεται η ταυτοποίηση του χρήστη κατά την σύνδεση του μέσω ενός φυλλομετρητή. Επιπλέον περιέχονται χρονοσημάνσεις δημιουργίας αλλά και ένα πεδίο που βοηθάει την εφαρμογή να αναγνωρίσει αν η συνεδρία ενός χρήστη έχει λήξει.

Για την δημιουργία ενός νέου ερωτηματολογίου, δημιουργείται ένα έγγραφο στην συλλογή **Quizzes** στο οποίο περιλαμβάνονται γενικές πληροφορίες για το ερωτηματολόγιο όπως το όνομα και η περιγραφή του ερωτηματολογίου, η λίστα με τις αναφορές για τα έγγραφα των ερωτήσεων τις οποίες περιλαμβάνει το ερωτηματολόγιο, τον δημιουργό του, και τις γενικές του ρυθμίσεις. Στην συλλογή των ερωτημάτων μετά δημιουργούνται έγγραφα τα οποία περιέχουν τις λεπτομέρειες των ερωτημάτων του ερωτηματολογίου όπως σε ποιο ερωτηματολόγιο ανήκουν, το κείμενο και τις επιλογές αν υπάρχουν και τις βαθμολογίες της ερώτησης.

Κατά την υποβολή απαντήσεων ενός χρήστη σε ένα ερωτηματολόγιο καταγράφεται σε ένα έγγραφο στην συλλογή **QuizSubmission** με τα στοιχεία για την ημερομηνία υποβολής, την βαθμολογία και το αναγνωριστικό του χρήστη που πραγματοποιεί την υποβολή. Μαζί δημιουργούνται στην συλλογή **Answers** και έγγραφα τα οποία αποθηκεύουν τις απαντήσεις του χρήστη στις ερωτήσεις του ερωτηματολογίου όπως και πιθανά σχόλια του διαχειριστή.

Όλες οι αλληλεπιδράσεις μεταξύ των διαφορετικών οντοτήτων της βάσης μπορούν να παρατηρηθούν στο παρακάτω διάγραμμα. Τέλος παρουσιάζεται και το μοντέλο που χρησιμοποιείται για την δημιουργία των εγγράφων των χρηστών στην βάση δεδομένων.



```

const UserSchema = new mongoose.Schema({
  uid: { type: Number, unique: true },
  name: { type: String, required: true },
  surname: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  phone: { type: String, required: true },
  isActive: { type: Boolean, default: true },
  isAdmin: { type: Boolean, default: false },
  groups: [{ type: mongoose.Schema.Types.ObjectId, ref: "Group" }]
}, { timestamps: true });

UserSchema.pre("save", async function (next) {
  if (!this.isNew) return next(); // Only increment for new documents
  this.uid = await getNextSequenceValue("User");

  // Hash the password if it has been modified
  if (this.isModified("password")) {
    try {
      const salt = await bcrypt.genSalt(10); // Generate salt
      this.password = await bcrypt.hash(this.password, salt); // Hash the password
    } catch (error) {
      return next(error); // Pass the error to the next middleware
    }
  }

  next();
});

```

Μοντέλο εγγράφου χρήστη

ΚΕΦΑΛΑΙΟ 4: ΕΠΙΔΕΙΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Οδηγός Χρήσης

Το Eleni Forms είναι σχεδιασμένο για την υποστήριξη συλλογής απαντήσεων σε ερωτηματολόγια ερωτήσεων διαφόρων τύπων. Παρέχει στους χρήστες της δυνατότητες συλλογής, βαθμολόγησης και σχολιασμού των αποτελεσμάτων αλλά και εξαγωγής των αποτελεσμάτων για διατήρηση αρχείων. Μέσα από τις λειτουργίες της εφαρμογής γίνεται δυνατή η κεντροποιημένη διαχείριση τόσο των αποτελεσμάτων όσο και των δημιουργημένων ερωτηματολογίων.

Διαχειριστές

Ο χρήστες χωρίζονται σε δύο κατηγορίες, τους διαχειριστές και τους απλούς χρήστες. Οι διαχειριστές είναι αυτοί στους οποίους δίνονται οι δυνατότητες διαχείρισης τόσο των ερωτηματολογίων όσο και των εγγεγραμμένων χρηστών. Από το πάνελ διαχείρισης δίνεται η δυνατότητα τόσο δημιουργίας και επεξεργασίας χρηστών όσο και διαγραφής τους. Στην επεξεργασία, ένας διαχειριστής μπορεί να επιλέξει να αναβαθμίσει κάποιον απλό χρήστη σε διαχειριστή και αντίστροφα. Στην προβολή των απαντήσεων του ερωτηματολογίου, ο διαχειριστής μπορεί να δει αναλυτικά τις απαντήσεις κάθε χρήστη για κάθε ερώτηση, να βαθμολογήσει χειροκίνητα τις απαντήσεις ελεύθερου κειμένου και να αφήσει σχόλια σε οποιαδήποτε απάντηση χρήστη.

Απλοί Χρήστες

Η δεύτερη κατηγορία χρηστών είναι οι απλοί χρήστες. Οι λογαριασμοί αυτοί κατέχουν πιο περιορισμένη πρόσβαση καθώς μπορούν να δουν μόνο το κεντρικό πάνελ ερωτηματολογίων και το προφίλ τους. Στο κεντρικό πάνελ ερωτηματολογίων τους εμφανίζονται όλα τα ορατά ερωτηματολόγια από τις ομάδες στις οποίες συμμετέχουν. Σε όσα ερωτηματολόγια δεν έχει καταγραφεί υποβολή απαντήσεων, εμφανίζεται η επιλογή έναρξης συμπλήρωσης του ερωτηματολογίου ενώ σε όσα ερωτηματολόγια έχει καταγραφεί προηγούμενη υποβολή απάντησης, οι χρήστες μπορούν να δουν τις απαντήσεις καθώς και την επίδοσή τους αλλά και πιθανά σχόλια που μπορεί να έχει αφήσει ο δημιουργός σε κάποιες από τις απαντήσεις τους.

Ερωτηματολόγια

Για την δημιουργία και επεξεργασία ενός ερωτηματολογίου, δίνονται στον διαχειριστή διάφορες επιλογές για να φέρει το τελικό αποτέλεσμα στα μέτρα του. Οι δυνατές επιλογές χωρίζονται στις ρυθμίσεις για όλο το ερωτηματολόγιο και στις ρυθμίσεις ανά ερώτηση. Στις γενικές ρυθμίσεις του ερωτηματολογίου, ο δημιουργός μπορεί να επιλέξει τον τίτλο και την περιγραφή του ερωτηματολογίου, αν επιθυμεί να θέσει συγκεκριμένη χρονική διάρκεια για την απάντηση του ερωτηματολογίου και της διάρκειας αυτής και αν θέλει τυχαία σειρά ερωτήσεων και επιλογών.

Σε κάθε ερώτηση που προστίθεται στο ερωτηματολόγιο, ο δημιουργός έχει την δυνατότητα να ορίσει ορισμένες παραμέτρους για την ερώτηση όπως την βαθμολογία της και το αν επιθυμεί οι λανθασμένες απαντήσεις να δέχονται βαθμολογική ποινή. Επίσης δίνεται και η επιλογή για τον τύπο της ερώτησης

ανάμεσα σε τέσσερις διαθέσιμες επιλογές: Ερωτήσεις πολλαπλών επιλογών με μοναδικής σωστής επιλογής, Ερωτήσεις πολλαπλών επιλογών με πολλαπλές σωστές επιλογές, ερωτήσεις ελεύθερου κειμένου και ερωτήσεις σωστής σειράς. Με αυτό τον τρόπο παρέχονται όλα τα απαραίτητα εργαλεία ώστε να μπορούν να δημιουργηθούν ερωτηματολόγια προσαρμοσμένα στις διδακτικές απαιτήσεις του κάθε μαθήματος κάνοντας έτσι πιο εύκολη την διεξαγωγή ασύγχρονων και σύγχρονων εξετάσεων και δοκιμασιών.

Μέτρα Πρόληψης Αντιγραφής

Για την πρόληψη και αντιμετώπιση πιθανών περιστατικών αντιγραφής κατά την διάρκεια μιας εξέτασης, έχουν εφαρμοστεί ορισμένα μέτρα αποτροπής. Κατά την απάντηση σε ένα ερωτηματολόγιο και αμέσως μόλις ο χρήστης πατήσει την επιλογή Έναρξη, η σελίδα αλλάζει σε εμφάνιση πλήρους οθόνης ώστε να αποτρέψει την περιήγηση εκτός του ερωτηματολογίου κατά την διάρκεια της απάντησης. Αυτό έχει ως σκοπό την αποτροπή του χρήστη από την δημιουργία νέων καρτελών στον φυλλομετρητή του ή την εναλλαγή σε άλλους φυλλομετρητές ή εφαρμογές που μπορεί να του δώσουν κάποιο άδικο πλεονέκτημα. Τέλος κατά την διάρκεια απάντησης ενός ερωτηματολογίου, είναι απενεργοποιημένη η εμφάνιση των επιλογών μέσω της χρήσης του δεξιού κουμπιού του ποντικιού καθώς και η χρήση της μετάβασης σε προηγούμενη σελίδα έως την ολοκλήρωση και υποβολή των απαντήσεων του ερωτηματολογίου.

Κεντρικό Πάνελ Ερωτηματολογίων

Η αρχική όψη μετά από την επιτυχημένη σύνδεση ενός χρήστη κάθε τύπου είναι το κεντρικό πάνελ των ερωτηματολογίων. Εκεί παρουσιάζονται όλα τα διαθέσιμα ερωτηματολόγια είτε που έχει δημιουργήσει ο ίδιος χρήστης αν είναι διαχειριστής είτε τα ερωτηματολόγια τα οποία ανήκουν στις ομάδες που έχει συμμετάσχει ο χρήστης αν είναι απλός χρήστης. Οι δημιουργοί από αυτή την όψη έχουν την δυνατότητα να επεξεργαστούν την δομή κάποιου ερωτηματολογίου είτε να το διαγράψουν, να δουν τα στατιστικά των απαντήσεων του ερωτηματολογίου, τις απαντήσεις που έχουν συλλεχθεί και τις οποίες μπορούν να εξαγάγουν σε αρχεία csv ή pdf μορφής και τέλος μπορούν να ορίσουν την ορατότητα του ερωτηματολογίου στην ομάδα μεταξύ «Δημόσιο» ή «Ιδιωτικό» για να μπορούν να ελέγξουν την δυνατότητα συλλογής απαντήσεων.

Κεντρικό Πάνελ Διαχείρισης Χρηστών

Το κεντρικό πάνελ διαχείρισης χρηστών είναι διαθέσιμο μόνο σε διαχειριστικούς λογαριασμούς και δίνει την δυνατότητα σε έναν διαχειριστή να εκτελέσει ενέργειες πάνω σε χρήστες. Το πάνελ περιλαμβάνει μια λίστα με τα στοιχεία όλων των εγγεγραμμένων χρηστών και επιτρέπει ενέργειες όπως την αναζήτηση και επεξεργασία των στοιχείων ή διαγραφή κάποιου χρήστη καθώς και την δημιουργία νέων χρηστών. Για την αύξηση των δικαιωμάτων ενός απλού χρήστη σε διαχειριστή, είναι δυνατή η σχετική επιλογή από την επεξεργασία πληροφοριών του χρήστη στην λίστα. Για λόγους ασφαλείας, αυτή η επιλογή είναι διαθέσιμη μόνο από την επεξεργασία των στοιχείων και άρα απαιτείται ένας άλλος διαχειριστής για να πραγματοποιηθεί η αλλαγή καθώς όλοι οι χρήστες κατά την δημιουργία τους είναι απλοί χρήστες από προεπιλογή.

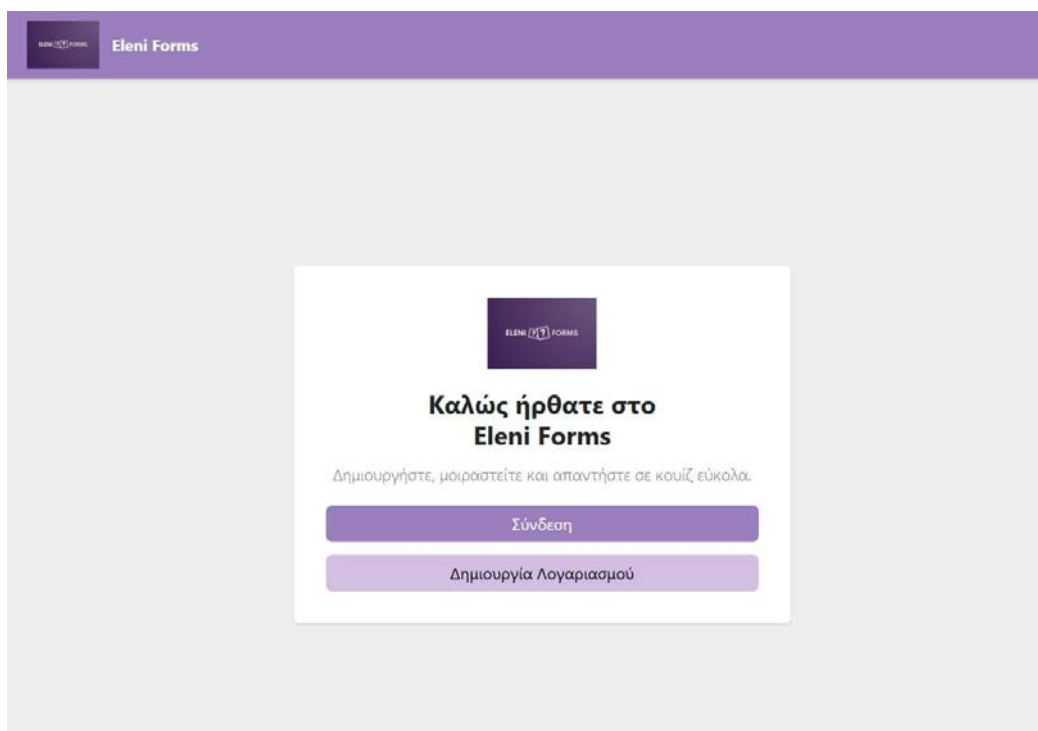
Η πρόσβαση στα ερωτηματολόγια στο Eleni Forms γίνεται μέσω ομάδων. Μέσω των ομάδων οι δημιουργοί μπορούν να οργανώσουν τα ερωτηματολόγια τους με βάση όποιο κριτήριο επιθυμούν όπως μαθήματα, θεματικές ενότητες, τάξεις κ.α. έτσι ώστε να μπορούν να κατευθύνουν τους χρήστες με μεγαλύτερη ευκολία και ακρίβεια. Στο πάνελ διαχείρισης ομάδων, οι δημιουργοί μπορούν να δημιουργήσουν νέες ομάδες, να επεξεργαστούν τα ονόματα και να διαγράψουν ομάδες. Μέσω αυτών των ομάδων, οι χρήστες μπορούν να βλέπουν τα ερωτηματολόγια που ανήκουν σε κάποια ομάδα απλά και μόνο επιλέγοντας να συμμετέχουν σε αυτήν.

Σελίδα Προφίλ

Όλοι οι χρήστες ανεξαιρέτως τύπου έχουν πρόσβαση στην σελίδα «Το προφίλ μου». Εκεί εμφανίζονται πληροφορίες για τον συνδεδεμένο λογαριασμό, τις ομάδες στις οποίες συμμετέχει και δίνεται η δυνατότητα αλλαγής των πληροφοριών του λογαριασμού, αλλαγής του κωδικού πρόσβασης, απεγγραφής από ομάδες και ολικής διαγραφής του λογαριασμού.

Περιήγηση στη Διεπαφή Χρήστη

Γενικά Γραφικά Στοιχεία



 Eleni Forms

testadmin@gmail.com ▾

Τα Κουίζ μου (Δημιουργός)

+ Δημιουργία Κουίζ+ Συμμετοχή σε Ομάδες

Προγραμματισμός
Ομάδα: Πληροφορική • Δημόσια (Ορατό στην Ομάδα)
Εισαγωγή στον Προγραμματισμό

✎ Επεξεργασία

✕ Διαγραφή

📊 Στατιστικά

📝 Απαντήσεις

🔴 Δημόσιο

 Eleni Forms

testuser@gmail.com ▾

Διαθέσιμα Κουίζ

+ Συμμετοχή σε Ομάδες

Προγραμματισμός
Ομάδα: Πληροφορική
Εισαγωγή στον Προγραμματισμό

📝 Προβολή Απαντήσεων

Απαντήθηκε

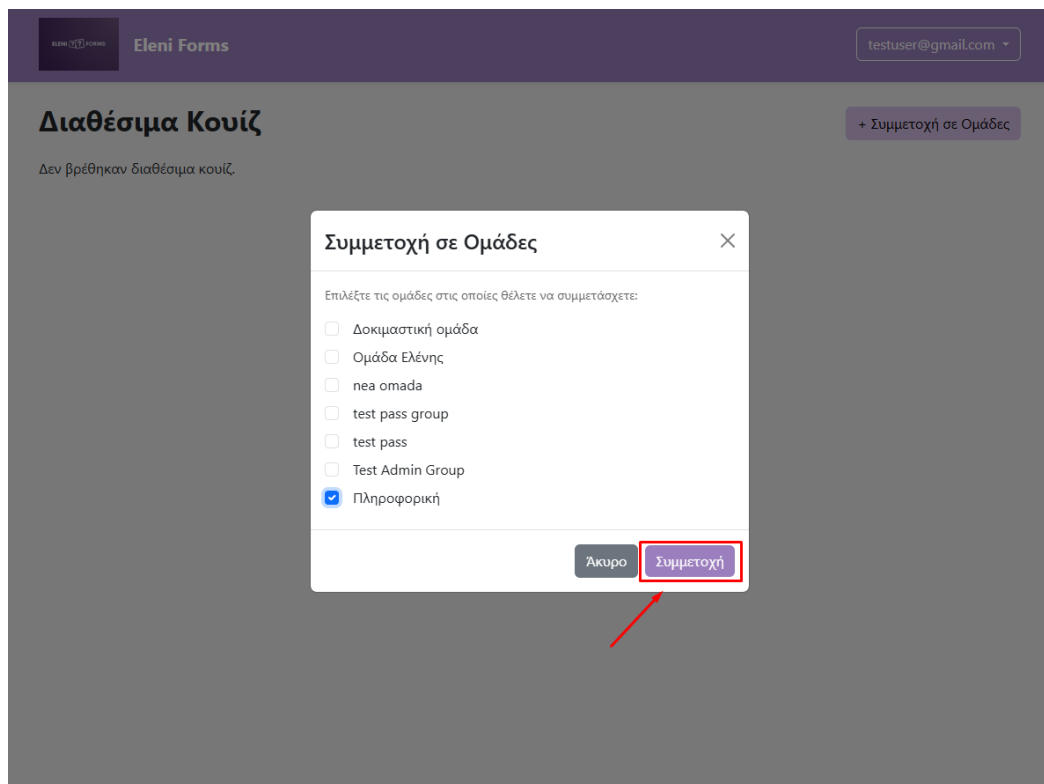
The screenshot shows a web application interface with a dark purple header containing the text 'Eleni Forms'. A modal window titled 'Δημιουργία Λογαριασμού' (Create Account) is centered on the screen. The form contains the following fields: 'Όνομα' (Name) with the value 'test', 'Επώνυμο' (Surname) with the value 'user', 'Email' with the value 'testuser@gmail.com', 'Κινητό' (Mobile) with the value '6999999999', 'Κωδικός' (Password) with masked characters and a 'Show' button, and 'Επιβεβαίωση Κωδικού' (Confirm Password) with masked characters and a 'Show' button. At the bottom of the form are two buttons: 'Άκυρο' (Cancel) and 'Εγγραφή' (Register).

The screenshot shows the same web application interface. A modal window titled 'Σύνδεση' (Login) is centered on the screen. The form contains the following fields: 'Email' with the value 'testuser@gmail.com' and 'Κωδικός' (Password) with masked characters and a 'Show' button. Below the password field is a link that reads 'Ξεχάσατε τον κωδικό σας' (Forgot your password?). At the bottom of the form are two buttons: 'Άκυρο' (Cancel) and 'Σύνδεση' (Login). Below the modal window, two buttons are visible on the background: 'Σύνδεση' (Login) and 'Δημιουργία Λογαριασμού' (Create Account).

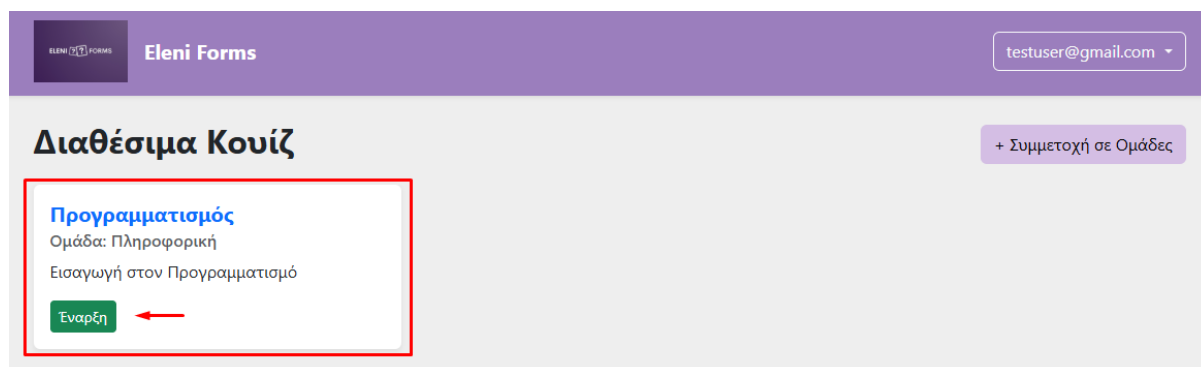
The screenshot shows a login modal titled "Σύνδεση" (Login) with a close button (X) in the top right corner. Inside the modal, there is a red error message box that says "User not found". Below this, there are two input fields: "Email" with the value "useruser@gmail.com" and "Κωδικός" (Password) with masked characters "*****" and a toggle icon. A link "Ξεχάσατε τον κωδικό σας" (Forgot your password?) is located below the password field. At the bottom of the modal, there are two buttons: "Άκυρο" (Cancel) and "Σύνδεση" (Login). Below the modal, there is a button labeled "Δημιουργία Λογαριασμού" (Create Account).

Συμμετοχή σε Ομάδα

The screenshot shows the main interface of Eleni Forms. The header includes the "Eleni Forms" logo and a user profile dropdown showing "testuser@gmail.com". The main content area has a heading "Διαθέσιμα Κουίζ" (Available Quizzes) and a message "Δεν βρέθηκαν διαθέσιμα κουίζ." (No quizzes found). A red arrow points to a button labeled "+ Συμμετοχή σε Ομάδες" (Participate in Groups) in the top right corner. The footer contains the text "ni-forms.netlify.app".



Απάντηση Ερωτηματολογίου



Προγραμματισμός

Group: Πληροφορική

Χρόνος που απομένει: 28:00

1. Η αρίθμηση ενός πίνακα ξεκινάει από το:

(10 points)

- ☐ -1
- ☒ 0
- ☐ 1
- ☐ 2

2. Βάλε τις εντολές στη σωστή σειρά ώστε να γίνει αντιμετάθεση δύο τιμών a , b .

(6 points)

Σύρε τις επιλογές στη σωστή σειρά.

```
≡ temp=a
≡ a=b
≡ b=temp
```

3. Γράψε έναν αλγόριθμο ο οποίος θα υπολογίζει το άθροισμα όλων των στοιχείων ενός πίνακα $A[10]$.

(10 points)

```
sum=0;
for(i=0; i<10; i++){
    sum=sum+A[i];
}
```

4. Ποια από τα παρακάτω δεν αποτελούν έγκυρα ονόματα μεταβλητών;

(10 points, -2 per wrong)

- ☐ name3
- ☒ +name
- ☐ _name
- ☒ 3name

Υποβολή απαντήσεων

Προβολή Αποτελεσμάτων ως Χρήστης

Διαθέσιμα Κουίζ

- + Συμμετοχή σε Ομάδες

Προγραμματισμός

Απαντήθηκε

Ομάδα: Πληροφορική

Εισαγωγή στον Προγραμματισμό

Προβολή Απαντήσεων

ELENI FORMS

Eleni Forms

testuser@gmail.com

Αποτελέσματα: Προγραμματισμός

Τελική βαθμολογία: 36

Οι απαντήσεις σου:

1. Η αρίθμηση ενός πίνακα ξεκινάει από το:

10 / 10 Πόντοι

1

-1

0

2

2. Βάλε τις εντολές στη σωστή σειρά ώστε να γίνει αντιμετάθεση δύο τιμών a, b.

6 / 6 Πόντοι

Η απάντησή σου:

1. temp=a
2. a=b
3. b=temp

3. Γράψε έναν αλγόριθμο ο οποίος θα υπολογίζει το άθροισμα όλων των στοιχείων ενός πίνακα A[10].

10 / 10 Πόντοι

Η απάντησή σου:

sum=0;
for(i=0; i<10; i++){
 sum=sum+A[i];
}

Σχόλια Βαθμολογητή:

Μπράβο!

4. Ποια από τα παρακάτω δεν αποτελούν έγκυρα ονόματα μεταβλητών;

10 / 10 Πόντοι

_name

3name

name3

+name

Πίσω στα Κουίζ

Προβολή και Επεξεργασία Στοιχείων Λογαριασμού

Eleni Forms testadmin@gmail.com

Διαχείριση Ομάδων

+ Δημιουργία Ομάδας

Όνομα Ομάδας	Ενέργειες
Δοκιμαστική ομάδα	
Ομάδα Ελένης	
Ανάλυση 1	
Ανάλυση 2	
Φυσική	
Test Admin Group	

Δημιουργία Ομάδας

Όνομα Ομάδας

Πληροφορική

Ακύρωση Δημιουργία

Δημιουργία και Επεξεργασία Ερωτηματολογίου

Eleni Forms testadmin@gmail.com

Τα Κουίζ μου (Δημιουργός)

Δεν βρέθηκαν διαθέσιμα κουίζ.

+ Δημιουργία Κουίζ + Συμμετοχή σε Ομάδες

Eleni Forms testadmin@gmail.com

Τα Κουίζ μου (Δημιουργός)

Δεν βρέθηκαν διαθέσιμα κουίζ.

+ Δημιουργία Κουίζ + Συμμετοχή σε Ομάδες

Δημιουργία Κουίζ

Τίτλος

Προγραμματισμός

Περιγραφή

Εισαγωγή στον Προγραμματισμό

Διάρκεια (λεπτά)

10

Ομάδα

Πληροφορική

☒ Δημόσιο Κουίζ

Άκυρο Δημιουργία

Eleni Forms

testadmin@gmail.com

Τα Κουίζ μου (Δημιουργός)

+ Δημιουργία Κουίζ

+ Συμμετοχή σε Ομάδες

Προγραμματισμός

Ομάδα: Πληροφορική • Δημόσιο (Ορατό στην Ομάδα)

Εισαγωγή στον Προγραμματισμό

Επεξεργασία

Διαγραφή

Στατιστικά

Απαντήσεις

Δημόσιο

Eleni Forms

testadmin@gmail.com

Επεξεργασία ερωτηματολογίου

Γενικές πληροφορίες και ρυθμίσεις

Τίτλος

Προγραμματισμός

Περιγραφή

Εισαγωγή στον Προγραμματισμό

Χρονική διάκριση

Ενεργοποιημένη

Διάρκεια (Λεπτά)

10

Τυχαία σειρά

Τυχαία σειρά ερωτήσεων

Τυχαία σειρά επιλογών

Ερωτήσεις

+ Νέα Ερώτηση

← Πίσω στα ερωτηματολόγια

Save Changes

45

Institutional Repository - Library & Information Centre - University of Thessaly
11/09/2026 22:30:32 EEST - 216.73.216.250

Eleni Forms

testadmin@gmail.com

Πίνακας Διαχείρισης

Διαχείριση Χρηστών
 Προβολή, επεξεργασία και δημιουργία λογαριασμών.

Γρήγορη Πρόσβαση
 Γρήγορη πρόσβαση σε ερωτηματολόγια και ομάδες χρηστών.

Νέος Χρήστης

Ερωτηματολόγια
 Διαχείριση Ομάδων

Λίστα Χρηστών (Σύνολο: 17)

Αναζήτηση...

Όνομα	Επώνυμο	Email	Τηλέφωνο	Ρόλος	Ενέργειες
Vassilis	Antonakakis	antonakakisbill@gmail.com	6918843187	Διαχειριστής	
Ελένη	Αθ	email@gmail.com	6999999999	Διαχειριστής	
Χαράλαμπος	Καλύμνιος	kalymnioscharalamos@gmail.com	6996317821	Χρήστης	
test	User	testuser@gmail.com	691234567	Χρήστης	
test	Admin	testadmin@gmail.com	6999999999	Διαχειριστής	
Panos	Dourdounis	panosdourdounas@gmail.com	6993989030	Χρήστης	
test	test	test@test.com	6999999999	Χρήστης	
Δημήτρης	Ριζος	dimitrisrizos511@gmail.com	6981058382	Χρήστης	
νεος	χρηστης	neos@gmail.com	1234567891	Χρήστης	

Eleni Forms

testadmin@gmail.com

Προγραμματισμός — User Answers

Export CSV

Export PDF

1. Η αρίθμηση ενός πίνακα ξεκινάει από το: (10 points)

User	1	-1	0	2	Score	Comment
test User					10	

2. Βάλε τις εντολές στη σωστή σειρά ώστε να γίνει αντιμετάθεση δύο τιμών a, b. (6 points)

User	temp=a	a=b	b=temp	Score	Comment
test User	temp=a	a=b	b=temp	6	

3. Γράψε έναν αλγόριθμο ο οποίος θα υπολογίζει το άθροισμα όλων των στοιχείων ενός πίνακα A[10]. (10 points)

User	User's Text Answer	Score	Comment
test User	<pre>sum=0; for(i=0; i<10; i++){ sum=sum+A[i]; }</pre>	10 Save	

4. Ποια από τα παρακάτω δεν αποτελούν έγκυρα ονόματα μεταβλητών; (10 points)

User	_name	3name	name3	+name	Score	Comment
test User					10	

Back

Programmatismos ? User Answers (Detailed)

User	Q#	Question	User Answer	Correct Answer	Score	Perfect?	Comment
test User	1	H arithmihsh enos pinaka xekinaei apo to:	0	0	10	Yes	
test User	2	Bale tis entoles sth swsth seira wste na ginei antimetathesh dyo timwn a, b.	1.temp=a, 2.a=b, 3.b=temp	1.temp=a, 2.a=b, 3.b=temp	6	Yes	
test User	3	Grapshe enan algorithmo o opoios tha ypologizei to athroisma olwn twv stolicheiwn enos pinaka A[10].	sum=0; for(i=0; i<10; i++){ sum=sum+A[i]; }	(Manual)	10	Yes	Mprabol!
test User	4	Poia apo ta parakataw den apoteloun egkyra onomata metablhtwn;	3name, +name	3name, +name	10	Yes	

User	Question No.	Question Text	User Answer(s)	Correct Answer(s)	Score	Perfectly Correct	Admin Comment
test User	1	H arithmishsh enos pinaka xekinaei apo to:	0	0	10	Yes	
test User	2	Bale tis entolēs sth swstē seirā wstē na ginei antimetathesh dyō timōn a, b.	1. temp=a 2. a=b 3. b=temp	1. temp=a 2. a=b 3. b=temp	6	Yes	
test User	3	Grāψε enan alghorithmo o opoios tha ypologizei to athroisma olwn twv stolicheiwn enos pinaka A[10].	sum=0;for(i=0; i<10; i++){ sum=sum+A[i]; }	(Manual Grade)	10	Yes	Mprabol!
test User	4	Poia apo ta parakataw den apoteloun egkyra onomata metablhtwn;	3name +name	3name +name	10	Yes	

Σχολιασμός και Χειροκίνητη Βαθμολόγηση Ερωτήσεων

Eleni Forms testadmin@gmail.com

2. Βάλε τις εντολές

User: test User

3. Γράψε έναν αλγόριθμο ο οποίος θα υπολογίζει το άθροισμα όλων των στοιχείων ενός πίνακα A[10]. (10 points)

User: test User

4. Ποια από τα παρακάτω είναι σωστά; (10 points)

User: test User

Back

Add/Edit Comment

Question: Γράψε έναν αλγόριθμο ο οποίος θα υπολογίζει το άθροισμα όλων των στοιχείων ενός πίνακα A[10].

User's Answer:

```
sum=0;
for(i=0; i<10; i++){
    sum=sum+A[i];
}
```

Your Comment:

Μπράβο!

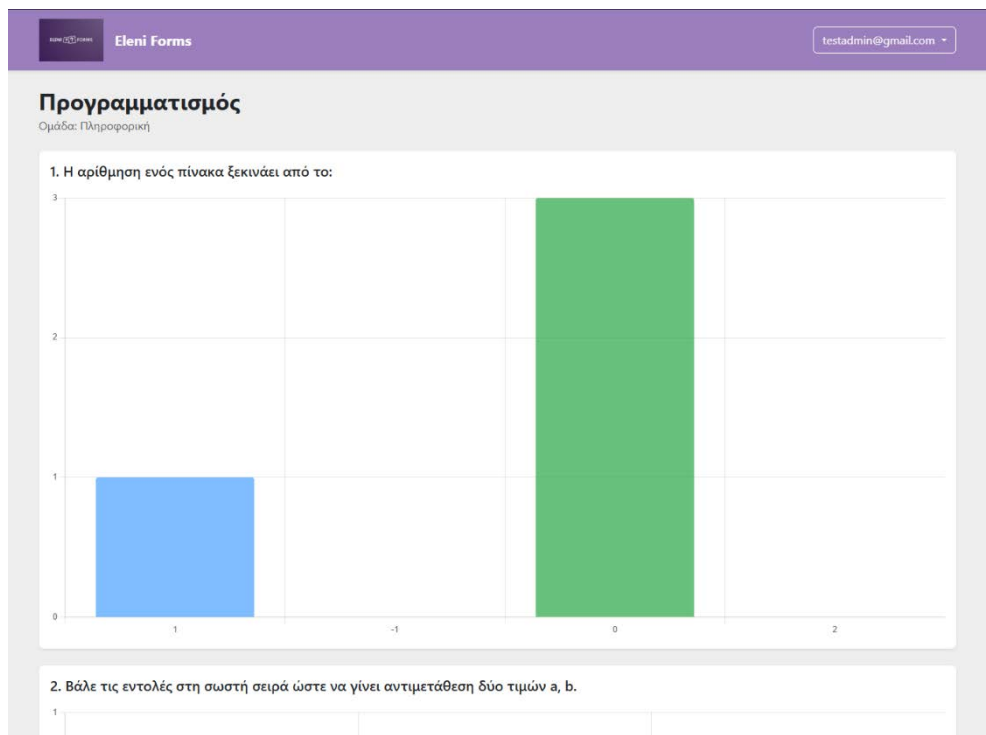
Cancel Save Comment

User	_name	3name	name3	+name	Score	Comment
test User		✓		✓	10	

3. Γράψε έναν αλγόριθμο ο οποίος θα υπολογίζει το άθροισμα όλων των στοιχείων ενός πίνακα A[10]. (10 points)

User	User's Text Answer	Score	Comment
test User	<pre>sum=0; for(i=0; i<10; i++){ sum=sum+A[i]; }</pre>	10 Save	

Προβολή Στατιστικών Στοιχείων Ερωτηματολογίου



Συμπεράσματα από την Υλοποίηση

Η υλοποίηση της εφαρμογής **Eleni Forms** ανέδειξε τόσο τις τεχνικές προκλήσεις της ανάπτυξης μιας σύγχρονης διαδικτυακής εφαρμογής όσο και τις ευκαιρίες που παρέχει η χρήση της σε εκπαιδευτικά περιβάλλοντα. Κατά την ανάπτυξη, κατέστη σαφές ότι η επιλογή της αρχιτεκτονικής **MERN** ήταν ιδιαίτερα αποτελεσματική, παρέχοντας την ευελιξία και την επεκτασιμότητα που απαιτούνται για τέτοιου τύπου εφαρμογές. Η χρήση της **React** για το **frontend** εξασφάλισε μια δυναμική εμπειρία χρήστη, ενώ ο συνδυασμός **Express** και **MongoDB** για το **backend** επέτρεψε την ταχύτατη υλοποίηση **RESTful** υπηρεσιών και την ασφαλή αποθήκευση δεδομένων.

Η διαδικασία αυτή, όμως, δεν ήταν απαλλαγμένη από δυσκολίες. Ειδικά η διαχείριση της κατάστασης του χρήστη, ο χειρισμός συνεδριών, καθώς και η προστασία από ανεπιθύμητη πρόσβαση απαιτούσαν ιδιαίτερη προσοχή. Επίσης, η ανάγκη για **responsive** σχεδίαση ώστε να διασφαλιστεί η λειτουργικότητα σε κινητές συσκευές αποτέλεσε βασικό άξονα ανάπτυξης. Παρά τις δυσκολίες αυτές, το τελικό αποτέλεσμα είναι μια λειτουργική εφαρμογή που υποστηρίζει τον ρόλο του διαχειριστή και του χρήστη με ξεκάθαρη ροή και δομή. Η εργασία κατέδειξε τη σημασία της καλής αρχιτεκτονικής, του σαφούς σχεδιασμού και της συνεχούς δοκιμής για την ανάπτυξη αξιόπιστων συστημάτων. Η επιτυχής ολοκλήρωση του έργου επιβεβαιώνει την αξία της εξατομικευμένης ανάπτυξης εφαρμογών έναντι των γενικών, έτοιμων λύσεων.

Πιθανές Επεκτάσεις της Εφαρμογής

Η εφαρμογή **Eleni Forms** στην παρούσα της μορφή παρέχει ένα ολοκληρωμένο σύνολο βασικών λειτουργιών, όπως η δημιουργία **quiz**, η οργάνωση χρηστών σε ομάδες, ο ρόλος του διαχειριστή, και η δυνατότητα υποβολής και αποθήκευσης απαντήσεων. Ωστόσο, υπάρχουν πολυάριθμες δυνατότητες επέκτασης που μπορούν να ενισχύσουν τη χρηστικότητα, την ασφάλεια, την επεκτασιμότητα και την ευρύτερη αποδοχή της πλατφόρμας.

Μια βασική προσθήκη θα ήταν η υλοποίηση **οργανισμών**. Οι οργανισμοί θα μπορούσαν να λειτουργούν ως διαχειριστικές οντότητες στις οποίες ανήκουν χρήστες, ομάδες και ερωτηματολόγια. Με τον τρόπο αυτό, κάθε οργανισμός θα διαχειρίζεται ανεξάρτητα τα δικά του δεδομένα, καθιστώντας την εφαρμογή κατάλληλη για **πολυθεσμική** χρήση (π.χ. από διαφορετικά σχολεία ή τμήματα ενός πανεπιστημίου), ενώ παράλληλα διασφαλίζεται η απομόνωση των πληροφοριών.

Επιπλέον, μπορεί να προστεθεί η δυνατότητα **προστασίας των ομάδων**, ώστε η ένταξη σε αυτές να απαιτεί είτε έγκριση από τον διαχειριστή είτε την παροχή ειδικού κωδικού πρόσβασης. Αυτό προσφέρει μεγαλύτερο έλεγχο στους διαχειριστές ως προς το ποιοι χρήστες μπορούν να συμμετέχουν στις ομάδες τους και εξασφαλίζει την εγκυρότητα της συμμετοχής. Συναφής με αυτή την ιδέα είναι η δυνατότητα **συμμετοχής σε quiz με χρήση μοναδικού κωδικού**, αντί της τυπικής σύνδεσης με λογαριασμό. Έτσι, χρήστες που δεν διαθέτουν λογαριασμό

στην εφαρμογή θα μπορούσαν να συμμετέχουν με έναν απλό κωδικό (π.χ. σε ημερίδες ή προσωρινές εξετάσεις), χωρίς να απαιτείται πλήρης εγγραφή.

Μια άλλη σημαντική επέκταση αφορά την **υποστήριξη πολυμέσων** στις ερωτήσεις (εικόνες, ήχος, βίντεο), που θα επιτρέψει την αξιολόγηση δεξιοτήτων πέραν της θεωρητικής γνώσης. Επιπλέον, η προσθήκη ερωτήσεων τύπου **αντιστοίχισης**, ή **συμπλήρωσης κενού** θα εμπλουτίσει σημαντικά τη μαθησιακή εμπειρία.

Μια ακόμα στρατηγική επέκταση θα μπορούσε να είναι η **υλοποίηση mobile εφαρμογής** με χρήση εργαλείων όπως το React Native, ώστε να είναι δυνατή η συμμετοχή και η διαχείριση quiz από φορητές συσκευές με εγγενή λειτουργικότητα και offline υποστήριξη.

Τέλος, ιδιαίτερη αξία θα είχε η **ενσωμάτωση real-time λειτουργιών**, όπως η ταυτόχρονη επίλυση quiz με live ανατροφοδότηση, βαθμολογία και πίνακες κατάταξης. Με την προσθήκη αυτών των χαρακτηριστικών, η «Eleni Forms» μπορεί να μετεξελιχθεί από μια βασική εφαρμογή quiz σε ένα πλήρες εκπαιδευτικό σύστημα διαχείρισης και αξιολόγησης, ικανό να καλύψει ποικίλα σενάρια σε σχολικά, πανεπιστημιακά και επαγγελματικά περιβάλλοντα.

- [1] Alf, I. W., & Rabail, T. (2020, 05). The effect of using Kahoot! for learning – A literature review. *Computers & Education*.
- [2] Anuradhani, N., Yatigammana, K., & Wijayarathna, G. (2024, 6). Defining gamification: a systematic literature review for developing a process-oriented definition. *Journal of Multidisciplinary & Translational Research (JMTR)*, σσ. 65-84.
- [3] Bensfiya, A., Boubih, S., Janati-Idrissi, R., & Madrane, M. (2023, 12). EXPLORING DIGITAL ALTERNATIVES FOR LEARNING ASSESSMENT: A COMPARATIVE STUDY OF GOOGLE FORMS AND QCM CAM. *INTERNATIONAL JOURNAL OF SCIENCE DIDACTICS AND EDUCATIONAL ENGINEERING (IJSDEE)*, σσ. 44-55.
- [4] Borad, M., Tejash, C., & Mehta, H. (2017, 11). A Review on NoSQL Databases . *International Journal of Science and Research (IJSR)*, σσ. 26-28.
- [5] Chondrogiannos, T. (2021, 11 28). *govwatch*. Ανάκτηση από The Cisco case: The government violates GDPR and personal data protection legislation: <https://govwatch.gr/en/finds/the-cisco-case-the-government-violates-gdpr-and-personal-data-protection-legislation/>
- [6] Haroon- Sulyman, S. (2014, 1). Client-Server Model. *IOSR Journal of Computer Engineering (IOSR-JCE)*, σσ. 67-71.
- [7] Iguenfer, F. z., Benhayoun, M., & Bouchaib, D. F. (2024, 5). An Overview of MEAN, MERN, and LAMP Stacks: A Comparative Analysis of Architecture, Cost, Learning Curve, and Community Support.
- [8] Jonathan, R., & Suprihadi. (2023, 12 30). Development of Front-End Web Applications Utilizing Single Page Application Framework and React.js Library. *International Journal Software Engineering and Computer Science (IJSECS)*, σσ. 529-536.
- [9] Kornienko, D. V., Mishina, S. V., & Melnikov, M. O. (2021). The Single Page Application architecture when developing secure Web services. *Journal of Physics: Conference Series*.
- [10] Munzert, S., Christian, R., Meißner, P., & Nyhuis, D. (2015). SQL and relational databases. *Automated Data Collection with R A Primer on Web and Data Technologies*, σσ. 164 - 195.
- [11] Noorkaran, B., Aditi, P., & Mehta, D. A. (2023, 4). Responsive Web Design and Its Impact on User Experience . *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*.
- [12] Özdemir, O. (2024, 10 13). Kahoot! Game-based digital learning platform:A comprehensive meta-analysis. *Journal of Computer Assisted Learning*.
- [13] Packard, N. (2020). The ARPANET Into the Internet: A Tale of Two Networks. Στο *Studies in Media and Communication*. Redfame Publishing.

- [14] Ramadhani, F., Sukyadi, D., & Purnawarman, P. (2023, 9 26). ENGLISH TEACHERS' ATTITUDES TOWARDS GOOGLE FORM AS AN ONLINE ASSESSMENT TOOL. *TELL-US Journal*.
- [15] Rašić, M., Aleksic-Maslac, K., & Vranesic, P. (2018). Influence of gamification on student motivation in the educational process in courses of different fields. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija.
- [16] Rauf, R. M., Rivaldo, T. E., & Sekolah Tinggi Sandi Negara, A. A. (x.x.). A review of attacks, objects, and mitigations on web services. *IOP Conference Series: Materials Science and Engineering*. Purpose-led publishing.
- [17] Sofiani, F., & Prasetya, R. E. (2024, 12 29). ASSESSING THE EFFECTIVENESS OF GOOGLE FORMS AND MICROSOFT FORMS IN ENGLISH LANGUAGE EVALUATION. *International Journal of English and Applied Linguistics*.
- [18] Souto Romero, M., Arias-Oliva, M., O., L. R., & & Pérez-Portabella, A. (2024, 3 19). Online assessment in higher education during the COVID-19. *Journal of Management and Business Education*, oo. 207-219.