



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

ΕSHOP ΜΕ ΠΡΟΪΟΝΤΑ ΤΕΧΝΟΛΟΓΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΠΕΪΛΕΡΙ ΦΡΑΝΤΖΕΣΚΟ (ΑΜ: 3120260)

Επιβλέπων: ΚΟΚΚΟΡΑΣ ΦΩΤΙΟΣ, *Επικ. Καθηγητής*

ΛΑΡΙΣΑ, ΙΟΥΝΙΟΣ 2024



UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

ESHOP WITH TECHNOLOGY PRODUCTS

BACHELOR THESIS

BEJLERI FRANCESKO (RN: 3120260)

Supervisor: *KOKKORAS FOTIOS, Asist. Professor*

LARISSA, JUNE 2024

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

ΜΠΕΪΛΕΡΙ ΦΡΑΝΤΖΕΣΚΟ

25/6/2024

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα	Ονοματεπώνυμο Επιβλέποντα Βαθμίδα/ιδιότητα επιβλέποντα, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	Ονοματεπώνυμο Μέλους 1 Βαθμίδα/ιδιότητα μέλους 1, Τμήμα/Ιδρυμα μέλους 1
Μέλος	Ονοματεπώνυμο Μέλους 2 Βαθμίδα/ιδιότητα μέλους 2, Τμήμα/Ιδρυμα μέλους 2

Ημερομηνία έγκρισης: Δεκ-2023

Περίληψη

Η πτυχιακή αφορά στην δημιουργία ενός eshop με προϊόντα τεχνολογίας. Αρχικά γίνεται αναφορά στα σύγχρονα on-line καταστήματα και ειδικότερα στο Skrutz, καθώς και στις σχετικές τεχνολογίες που θα χρησιμοποιηθούν. Ειδικότερα γίνεται αναφορά στις html, css, JavaScript και php, όπου η πρώτη βοηθάει στην δόμηση μιας ιστοσελίδας, η δεύτερη καθορίζει την εμφάνιση της ιστοσελίδας (διαρρύθμιση και μορφοποίηση), η τρίτη στην διαδραστικότητα των ιστοσελίδων, και η τελευταία υλοποιεί την λογική κατασκευής των ιστοσελίδων διασυνδέοντας και τη βάση δεδομένων (MySQL) στην οποία αποθηκεύονται τα δεδομένα για προϊόντα, χρήστες και συναλλαγές.

Στη συνέχεια γίνεται ανάλυση (σε επίπεδο κώδικα) βασικών μηχανισμών που υλοποιήθηκαν (εγγραφή χρήση, είσοδος/έξοδος χρήστη και διαχειριστή, καλάθι αγορών, λίστα επιθυμιών, αγορά προϊόντος, διαχείριση προϊόντων και κατηγοριών, κτλ.

Τέλος, γίνεται εκτενής αναφορά στις δυσκολίες που υπήρξαν στην υλοποίηση της εφαρμογής.

Abstract

The diploma concerns the creation of an eshop with technology products. Initially, reference is made to modern online stores and particularly to Skrutz, as well as to the relevant technologies that will be used, specifically, to html, css, javascript and php, where the first helps to describe the content of the web pages, the second determines the appearance of the website (layout and formatting), the third improves the interactivity of the pages, and the last one implements the logic of the web pages in cooperation with a database in which product, user, and transaction data is stored.

Then, an analysis is made (at the code level) of the main mechanisms implemented (user registration, user and administrator login/logout, shopping cart, wish list, product purchase, product and category management, etc.).

Finally, extensive reference is made to the difficulties encountered in the implementation of the application.

Ευχαριστίες

Με την ολοκλήρωση αυτής της πτυχιακής εργασίας, θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες σε όλους όσους συνέβαλαν στην επιτυχή έκβαση αυτού του έργου. Πρώτα και κύρια, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Κόκκορα Φώτιο, για την αμέριστη καθοδήγηση, την πολύτιμη υποστήριξη και την ανεξάντλητη υπομονή του καθ' όλη τη διάρκεια της προσπάθειας για την δημιουργία του κώδικα. Οι συμβουλές και οι παρατηρήσεις του ήταν ανεκτίμητες και συνέβαλαν καθοριστικά στην ποιότητα της εργασίας αυτής. Επίσης, θα ήθελα να ευχαριστήσω τα μέλη της επιτροπής αξιολόγησης για τον χρόνο και τον κόπο που αφιέρωσαν στην ανάγνωση και αξιολόγηση της εργασίας μου, καθώς και για τα εποικοδομητικά τους σχόλια και τις προτάσεις. Ιδιαίτερες ευχαριστίες οφείλω στους γονείς μου, για την αμέριστη ηθική και οικονομική τους υποστήριξη κατά τη διάρκεια των σπουδών μου. Χωρίς τη συνεχή τους παρουσία και ενθάρρυνση, δεν θα μπορούσα να φτάσω ως εδώ. Ευχαριστώ επίσης τους φίλους και συμφοιτητές μου για την ενθάρρυνση και την παρέα τους στις δύσκολες στιγμές της μελέτης και της έρευνας. Η αλληλοϋποστήριξη και οι συζητήσεις μας ήταν πηγή έμπνευσης και κουράγιου. Τέλος, ευχαριστώ όλους εκείνους που με οποιονδήποτε τρόπο συνέβαλαν στη διαμόρφωση και ολοκλήρωση της παρούσας πτυχιακής εργασίας. Η συμβολή τους, έστω και έμμεσα, υπήρξε σημαντική και αναγνωρίζεται με ευγνωμοσύνη.

Ονοματεπώνυμο: Μπεϊλέρι Φραντζέσκο

Ημερομηνία: 11/06/2024

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	VII
ABSTRACT	IX
ΕΥΧΑΡΙΣΤΙΕΣ	XI
ΠΕΡΙΕΧΟΜΕΝΑ	XIII
1 ΕΙΣΑΓΩΓΗ	1
2 ΕΠΙΤΥΧΗΜΕΝΑ ESHOP ΚΑΙ ΚΩΔΙΚΕΣ	3
2.1 Η ΕΠΙΤΥΧΗΜΕΝΗ ΔΙΑΔΡΟΜΗ ΤΟΥ ΣΚΡΟΥΤΖ.....	3
2.2 Η ΔΙΑΔΡΟΜΗ ΕΠΙΤΥΧΙΑΣ ΤΟΥ E-SHOP.GR.....	4
2.3 Η ΕΠΙΤΥΧΙΑ ΤΟΥ E-GATE.GR.....	5
2.4 ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΚΑΤΑΣΤΗΜΑ ΤΟΥ PUBLIC	6
2.5 HTML– CSS – JAVASCRIPT – PHP	7
2.5.1 HTML.....	7
2.5.2 CSS	8
2.5.3 JavaScript.....	8
2.5.4 PHP	9
3 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ	11
3.1 ΕΓΓΡΑΦΗ - ΕΙΣΟΔΟΣ - ΈΞΟΔΟΣ	11
3.2 ΓΡΗΓΟΡΗ ΠΡΟΒΟΛΗ.....	14
3.3 ΚΑΛΑΘΙ ΑΓΟΡΩΝ ΚΑΙ ΛΙΣΤΑ ΕΠΙΘΥΜΙΩΝ.....	15
3.3.1 Καλάθι αγορών και λίστα επιθυμιών	15
3.3.2 Λίστα επιθυμιών.....	16
3.3.3 Καλάθι αγορών	17
3.3.4 Από το καλάθι αγορών στην αγορά	19
3.4 ΣΕΛΙΔΑ ΑΝΑΖΗΤΗΣΗΣ ΚΑΙ ΑΓΟΡΑΣ ΠΡΟΪΟΝΤΩΝ.....	20
3.4.1 Σελίδα αναζήτησης	20
3.4.2 Αγορά προϊόντων	21
3.5 ΕΠΙΚΟΙΝΩΝΙΑ - ΜΗΝΥΜΑΤΑ	22
3.5.1 Επικοινωνία των χρηστών με τους διαχειριστές.....	22
3.5.2 Μηνύματα που παίρνουν οι διαχειριστές από τους χρήστες	23
3.6 ΑΡΧΙΚΗ ΣΕΛΙΔΑ - JAVASCRIPT ΚΩΔΙΚΑΣ.....	24
3.6.1 Αρχική σελίδα.....	24
3.6.2 Javascript κώδικας	26
3.7 ΠΑΡΑΓΓΕΛΙΕΣ	27
3.7.1 Παραγγελίες χρηστών.....	27
3.7.2 Παραγγελίες που παρακολουθούν οι διαχειριστές.....	28
3.8 ΚΑΤΑΧΩΡΗΣΗ ΠΡΟΪΟΝΤΩΝ ΣΤΟ ESHOP	29
3.9 ΕΝΗΜΕΡΩΣΗ ΠΡΟΦΙΛ ΧΡΗΣΤΗ & ΔΙΑΧΕΙΡΙΣΤΗ.....	31
3.9.1 Ενημέρωση προφίλ χρήστη	31
3.9.2 Ενημέρωση του προφίλ του διαχειριστή.....	33
3.10 ΛΟΓΑΡΙΑΣΜΟΙ ΧΡΗΣΤΩΝ - ΔΙΑΧΕΙΡΙΣΤΩΝ	34

3.10.1	Λογαριασμοί Χρηστών	34
3.10.2	Λογαριασμοί Διαχειριστών.....	35
3.11	ΕΝΗΜΕΡΩΣΗ ΠΡΟΪΟΝΤΩΝ - ΚΕΦΑΛΙΔΑ ΧΡΗΣΤΗ	35
3.11.1	Ενημέρωση των Προϊόντων	35
3.11.2	Κεφαλίδα Χρήστη.....	37
3.12	ΥΠΟΣΕΛΙΔΟ, ΠΛΗΡΟΦΟΡΙΕΣ ΚΑΙ ΑΞΙΟΛΟΓΗΣΕΙΣ.....	39
3.12.1	Υποσέλιδο σελίδας χρήστη.....	39
3.12.2	Πληροφορίες και αξιολογήσεις του eshop.....	40
4	Η ΔΥΣΚΟΛΙΕΣ ΤΟΥ ΚΩΔΙΚΑ	43
4.1	ΚΑΛΑΘΙ ΑΓΟΡΩΝ ΚΑΙ ΛΙΣΤΑ ΕΠΙΘΥΜΙΩΝ	43
4.2	ΤΟΠΟΘΕΤΗΣΗ ΠΡΟΪΟΝΤΩΝ ΑΠΟ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ	44
4.3	ΕΝΗΜΕΡΩΣΗ ΤΩΝ ΠΡΟΪΟΝΤΩΝ ΑΠΟ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ	44
4.4	ΔΥΣΚΟΛΙΕΣ ΣΤΗΝ ΠΛΗΡΩΜΗ ΤΩΝ ΠΡΟΪΟΝΤΩΝ.....	44
4.5	ΔΥΣΚΟΛΙΕΣ ΣΤΗΝ ΓΡΗΓΟΡΗ ΠΡΟΒΟΛΗ & ΑΝΑΖΗΤΗΣΗ	45
4.6	ΔΥΣΚΟΛΙΕΣ ΣΤΗΝ ΕΝΗΜΕΡΩΣΗ ΠΡΟΦΙΛ	45
4.7	ΔΥΣΚΟΛΙΕΣ ΣΕ ΑΡΧΙΚΗ ΣΕΛΙΔΑ & ΚΑΤΗΓΟΡΙΕΣ ΧΡΗΣΤΩΝ	46
4.8	ΔΥΣΚΟΛΙΕΣ ΣΤΗΝ ΕΓΓΡΑΦΗ ΚΑΙ LOGIN	46
5	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	49

1 Εισαγωγή

Σε αυτήν την πτυχιακή αναφερόμαστε στη δημιουργία ενός eshop με ηλεκτρονικά προϊόντα. Αρχικά, μιλάμε για το πώς μεγάλες ελληνικές εταιρίες τις Ελλάδας στον παγκόσμιο ιστότοπο κατάφεραν να θριαμβεύσουν με τα δικά τους eshop και να αλλάξουν την νοοτροπία των Ελλήνων προς την αγορά των προϊόντων από καταστήματα του δρόμου σε ιντερνετικά καταστήματα όπως το skroutz, public και άλλα, τα οποία επίσης με βοήθησαν παίρνοντας ιδέες από αυτά να δημιουργήσω το δικό μου eshop. Στη συνέχεια, αναφερόμαστε στους κώδικες που χρησιμοποίησα για να δημιουργήσω το δικό μου eshop όπως η html η οποία είναι μια γλώσσα προγραμματισμού όπου βάζει τα θεμέλια για την δημιουργία μιας ιστοσελίδας, η css η οποία βοηθάει στην εμφάνιση τις ιστοσελίδας γενικότερα ώστε να υπάρχει μια ομοιομορφία, μετά έχουμε την javascript η οποία βοηθάει στο δυναμικό τρόπο εμφάνισης της ιστοσελίδας αλλά και του eshop, τέλος έχουμε την php η οποία η δουλειά τις βρίσκεται σχεδόν όλη πίσω από μια ιστοσελίδα και η οποία σε συνεργασία με την βάση δεδομένων λειτουργεί με δυναμικό ώστε για παράδειγμα να αποθηκεύει κωδικούς, ονόματα, προϊόντα και άλλα πράγματα. Επιπρόσθετα, θα αναλύσουμε των κώδικα σε όλα τα σημεία του όπως για το πως δημιουργήσαμε είσοδο-έξοδο-εγγραφή για τον χρήστη ή τον διαχειριστεί, την υλοποίηση του καλαθιού αγορών, τις λίστες επιθυμιών, τις κατηγορίες των προϊόντων, τις εισαγωγής των προϊόντων από των διαχειριστεί ώστε να τα βλέπει ο χρήστης, τις παραγγελίας των προϊόντων και άλλα. Τέλος, θα αναφερθούμε και για τις δυσκολίες που προκάλεσαν σε εμένα μέχρι να δημιουργηθεί το eshop στην βέλτιστη κατάσταση, όσον αφορά για το γενικό κομμάτι του eshop οποιοσδήποτε μπορούσε να τοποθετήσει κακόβουλο κώδικα οπότε για να αποφευχθεί αυτό προσθεσα κάποια ασφάλεια για των κώδικα, άλλα προβλήματα ήταν η εισαγωγή των προϊόντων στο eshop και άλλα.

2 Επιτυχημένα Eshop και Κώδικες

Στην παρούσα εργασία, θα εξετάσουμε την εξέλιξη του Skrutz, τους κύριους παράγοντες που συνέβαλαν στην επιτυχία του, καθώς και τις στρατηγικές που υιοθετήθηκαν για το μέλλον. Επίσης, θα μιλήσουμε και για το e-shop.gr που έχει καταφέρει να καθιερωθεί ως ένα από τα πιο αξιόπιστα και δημοφιλή ηλεκτρονικά καταστήματα στη χώρα και αποτελεί έναν από τους πρωτοπόρους στον τομέα του ηλεκτρονικού εμπορίου στην Ελλάδα.

2.1 Η επιτυχημένη Διαδρομή του Σκρουτζ

Το Skrutz ιδρύθηκε το 2005 από τους Γιάννης Κουράκης και Γιώργος Μαρούλης, με στόχο να προσφέρει στους καταναλωτές έναν εύκολο και γρήγορο τρόπο για σύγκριση τιμών προϊόντων online. Από τότε, το Skrutz έχει διανύσει μια εντυπωσιακή πορεία ανάπτυξης, επεκτείνοντας την παρουσία του σε νέες αγορές και αναβαθμίζοντας συνεχώς την υπηρεσία του. Οι κύριοι παράγοντες που συνέβαλαν στην επιτυχία του Skrutz περιλαμβάνουν την αξιοποίηση της τεχνολογίας για τη βελτίωση της εμπειρίας του χρήστη, τη σταθερή ανάπτυξη του αριθμού των συνεργαζόμενων καταστημάτων και προμηθευτών, καθώς και τη διαρκή προσαρμογή στις ανάγκες της αγοράς. Μια από τις βασικές δυνάμεις του Skrutz είναι η συνεχής καινοτομία στον τομέα της τεχνολογίας. Με εφαρμογές όπως η ανάπτυξη ευφών αλγορίθμων σύστασης προϊόντων και η επέκταση στον τομέα του mobile commerce, το Skrutz προσφέρει στους χρήστες του πρωτοποριακές λύσεις. Επίσης, Το Skrutz έχει ένα θετικό αντίκτυπο στην ελληνική οικονομία, συμβάλλοντας στη δημιουργία θέσεων εργασίας και στην προώθηση της ψηφιακής μετάβασης των ελληνικών επιχειρήσεων. Παρά την επιτυχία του, το Skrutz αντιμετωπίζει και προκλήσεις, όπως η αυξημένη ανταγωνιστικότητα και οι αλλαγές στις καταναλωτικές συνήθειες. Μελλοντικές στρατηγικές περιλαμβάνουν την επέκταση σε νέους τομείς και την ενίσχυση της παρουσίας του στο διεθνές περιβάλλον. Το Skrutz αποτελεί ένα παράδειγμα επιχειρηματικής επιτυχίας στον τομέα του ψηφιακού εμπορίου. Μέσα από σταθερή καινοτομία και στρατηγική προσαρμογής, το Skrutz έχει καταφέρει να καθιερωθεί

ως ένα από τα κορυφαία brands στην ελληνική αγορά και να διαμορφώσει το μέλλον του ψηφιακού εμπορίου. Για το eshop μου με βοήθησε στο να κτίσω κυρίως το backend κομμάτι όπως για παράδειγμα να δημιουργήσω το καλάθι αγορών μου, την λίστα επιθυμιών και την γρήγορη προβολή των προϊόντων. Γενικότερα, με βοήθησε και για όλα τα άλλα κομμάτια του eshop μου όσον αφορά το backend όμως πήρα και λίγο ιδέες και για το frontend όπως την γρήγορη προβολή κομμάτι των προϊόντων. Επίσης, με βοήθησε και στην admin σελίδα ώστε να μπορέσω να δημιουργήσω έναν κώδικα για να μπορώ να ανεβάσω τα προϊόντα αλλά και να μπορώ να ελέγχω τις παραγγελίες των πελατών και να τα αποδεχτώ ή όχι.



Εικόνα 1: skroutz

2.2 Η Διαδρομή Επιτυχίας του e-shop.gr

Η ιστορία του e-shop.gr ξεκινά το 1998, όταν οι ιδρυτές του, Σάββας Τριανταφύλλου και Αλέξανδρος Καρατζιάς, είδαν την ευκαιρία στην αναπτυσσόμενη αγορά του ηλεκτρονικού εμπορίου και αποφάσισαν να ξεκινήσουν τη δική τους επιχείρηση. Από ένα μικρό ηλεκτρονικό κατάστημα, το e-shop.gr εξελίχθηκε σε έναν από τους μεγαλύτερους παίκτες στον χώρο του ηλεκτρονικού εμπορίου στην Ελλάδα και όχι μόνο. Οι κύριοι παράγοντες που συνέβαλαν στην επιτυχία του e-shop.gr περιλαμβάνουν την ποιότητα των προϊόντων που προσφέρει, την άριστη εξυπηρέτηση πελατών, την αξιοπιστία και την ταχύτητα της αποστολής, καθώς και την εξειδίκευσή του σε διάφορους τομείς, όπως η τεχνολογία και η ηλεκτρονική. Ένα από τα κύρια χαρακτηριστικά του e-shop.gr είναι η σταθερή εξέλιξη καινοτόμων τεχνολογιών για τη βελτίωση της εμπειρίας αγοράς των πελατών του. Με σύγχρονες διαδικτυακές πλατφόρμες, εφαρμογές κινητής τεχνολογίας και εξειδικευμένους αλγόριθμους συστάσεων προϊόντων, το e-shop.gr προσφέρει μια ολοκληρωμένη και εύκολη εμπειρία αγοράς στους πελάτες του. Το e-shop.gr έχει έναν

θετικό αντίκτυπο στην ελληνική οικονομία, συμβάλλοντας στη δημιουργία θέσεων εργασίας και στην αύξηση των εσόδων του κράτους μέσω της φορολογίας και των εισφορών. Παρά την επιτυχία του, το e-shop.gr αντιμετωπίζει προκλήσεις και προοπτικές για το μέλλον. Στρατηγικές όπως η επέκταση σε νέους τομείς, η ενίσχυση της παρουσίας του στο διεθνές επίπεδο και η συνεχής καινοτομία αναμένεται να διαμορφώσουν το μέλλον του e-shop.gr στον ψηφιακό κόσμο. Το e-shop.gr αποτελεί ένα φωτεινό παράδειγμα επιχειρηματικής επιτυχίας στον τομέα του ηλεκτρονικού εμπορίου. Με σταθερή προσήλωση στην ποιότητα, την καινοτομία και την εξυπηρέτηση των πελατών του, το e-shop.gr συνεχίζει να διαμορφώνει το μέλλον του ψηφιακού εμπορίου στην Ελλάδα και να προσφέρει μια εξαιρετική εμπειρία αγοράς σε όλους τους πελάτες του. Όσον αφορά τον κώδικά μου με βοήθησε και στο backend και στο frontend αλλά κυρίως έδωσα έμφαση στο frontend για να δημιουργήσω το καλάθι αγορών, την λίστα επιθυμιών, τις κατηγορίες των προϊόντων και την γρήγορη προβολή. Επίσης, με βοήθησε στην αρχική της ιστοσελίδας μου αλλά και στην admin σελίδα όπως για τα εισερχόμενα μηνύματα που ερχόντουσαν από τους πελάτες μέχρι και για τις παραγγελίες που στέλνουν οι πελάτες.

2.3 Η επιτυχία του e-gate.gr

Το e-gate.gr έχει καταφέρει να ξεχωρίσει ως μια από τις κορυφαίες πλατφόρμες ηλεκτρονικού εμπορίου στην Ελλάδα, εστιάζοντας αποκλειστικά στην πώληση τεχνολογικών προϊόντων. Η επιτυχία του οφείλεται στην εξειδίκευση στον κλάδο, προσφέροντας ένα ευρύ φάσμα προϊόντων υψηλής ποιότητας από κορυφαίες μάρκες, όπως υπολογιστές, smartphones, tablets και περιφερειακά. Η έμφαση στην ποιότητα και η συνεχής ανανέωση του καταλόγου με τις πιο πρόσφατες τεχνολογικές εξελίξεις διασφαλίζουν την ικανοποίηση των πελατών. Επιπλέον, οι ανταγωνιστικές τιμές και οι συχνές εκπτώσεις προσφέρουν ελκυστική αξία στους καταναλωτές. Η εξαιρετική εξυπηρέτηση πελατών μέσω τηλεφώνου, email και live chat, καθώς και η φιλική πολιτική επιστροφών, ενισχύουν την εμπιστοσύνη των πελατών. Η επένδυση σε νέες τεχνολογίες και καινοτόμες λύσεις βελτιώνει την εμπειρία χρήστη, ενώ η αξιοπιστία και η μακροχρόνια παρουσία του e-gate.gr στον χώρο του ηλεκτρονικού εμπορίου δημιουργούν μια σταθερή σχέση εμπιστοσύνης με τους καταναλωτές. Τέλος, η δυναμική στρατηγική marketing, που περιλαμβάνει διαφημίσεις σε ψηφιακές πλατφόρμες και συνεργασίες με influencers, προσελκύει νέους πελάτες και διατηρεί το ενδιαφέρον των υπαρχόντων. Όλοι αυτοί οι παράγοντες συνδυάζο-

νται για να δημιουργήσουν μια μοναδική και ικανοποιητική εμπειρία αγορών, καθιστώντας το e-gate.gr μια από τις κορυφαίες επιλογές για τεχνολογικά προϊόντα στην Ελλάδα. Οι επιλογές που έκανα από αυτό το eshop με βοήθησαν και για το backend και για το frontend όσον αφορά κυρίως τη λειτουργία του καλάθι αγορών μαζί με την προσθήκη των στοιχείων του πελάτη.

2.4 Το Ηλεκτρονικό κατάστημα του Public

Το eshop του Public αποτελεί ένα από τα μεγαλύτερα και πιο δημοφιλή ηλεκτρονικά καταστήματα στην Ελλάδα, προσφέροντας μια ευρεία γκάμα προϊόντων από ηλεκτρονικά είδη και βιβλία μέχρι παιχνίδια και είδη σπιτιού. Ξεκίνησε τη λειτουργία του το 2005 με το πρώτο φυσικό κατάστημα στη Σταδίου στην Αθήνα, και η είσοδός του στον ψηφιακό κόσμο έγινε με τη δημιουργία του eshop, το οποίο γρήγορα έγινε γνωστό για την πλούσια ποικιλία προϊόντων και τις ανταγωνιστικές τιμές του. Η πλατφόρμα του eshop του Public είναι σχεδιασμένη με γνώμονα τη φιλικότητα προς τον χρήστη και την ευκολία πλοήγησης, χρησιμοποιώντας προηγμένες τεχνολογίες για την ασφάλεια των συναλλαγών, όπως SSL κρυπτογράφηση και αυθεντικοποίηση δύο παραγόντων, διασφαλίζοντας την προστασία των προσωπικών δεδομένων των χρηστών. Η γκάμα των προϊόντων καλύπτει κάθε ανάγκη των καταναλωτών, περιλαμβάνοντας ηλεκτρονικά είδη, βιβλία, παιχνίδια και ψυχαγωγία, καθώς και είδη σπιτιού και lifestyle. Επιπλέον, προσφέρει πολλές υπηρεσίες όπως δωρεάν παράδοση σε παραγγελίες άνω ενός συγκεκριμένου ποσού, δυνατότητα παραλαβής από κατάστημα (Click & Collect), και ευέλικτες επιλογές πληρωμής όπως δόσεις με πιστωτική κάρτα και PayPal. Το eshop του Public χρησιμοποιεί μια πολυδιάστατη στρατηγική μάρκετινγκ για την προσέλκυση και διατήρηση των πελατών του, που περιλαμβάνει online και offline διαφημίσεις, τακτικές εκπτώσεις και προσφορές, καθώς και προγράμματα επιβράβευσης για τους πιστούς πελάτες. Η ανάλυση αγοράς δείχνει ότι έχει καταφέρει να προσελκύσει ένα ευρύ φάσμα καταναλωτών, τόσο από τις νεότερες όσο και από τις παλαιότερες ηλικιακές ομάδες, χάρη στην ποικιλία προϊόντων και τις ανταγωνιστικές τιμές. Το eshop του Public έχει καταφέρει να εδραιωθεί ως ένας από τους κορυφαίους προορισμούς για ηλεκτρονικές αγορές στην Ελλάδα, προσφέροντας μια ολοκληρωμένη εμπειρία αγορών που συνδυάζει ποικιλία, ποιότητα και καινοτομία. Για τη συνεχή βελτίωση και ανάπτυξη, το Public θα μπορούσε να εξετάσει την περαιτέρω εξατομίκευση, τη διεθνοποίηση και την ενσωμάτωση βιώσιμων πρακτικών, για να διατηρήσει την ανταγωνιστικότητά του και να συνεχίσει να ικανοποιεί τις ανάγκες

των καταναλωτών σε μια δυναμική και συνεχώς εξελισσόμενη αγορά. Τέλος, όσον αφορά τον κώδικα άντλησα κάποιες ιδέες για το κόμματα δημιουργίας του καλαθιού αλλά και στην κατηγοριοποίηση των προϊόντων γενικότερα.



Εικόνα 2: κατάστημα Public

2.5 HTML– CSS – JAVASCRIPT – PHP

2.5.1 HTML

Η HTML (HyperText Markup Language) αναδεικνύεται ως απαραίτητο εργαλείο στη διαμόρφωση και παρουσίαση πληροφοριών στο διαδίκτυο. Αποτελεί τη βάση επάνω στην οποία κτίζονται οι ιστοσελίδες, παρέχοντας μια δομημένη προσέγγιση για τον οργανισμό του περιεχομένου. Χρησιμοποιώντας την HTML, μπορούμε να δημιουργήσουμε ιστοσελίδες που περιέχουν πληροφορίες όπως κείμενο, εικόνες, συνδέσμους και πίνακες. Η HTML αποτελείται από μια ποικιλία ετικετών, οι οποίες περιγράφουν τη δομή και το περιεχόμενο της ιστοσελίδας, όπως οι ετικέτες <head>, <title>, <body>, <p>, <h1> έως <h6>, <a>, και πολλές άλλες.



Εικόνα 3: Html

Η διαμόρφωση του περιεχομένου μπορεί να γίνει μέσω της επιλογής και της χρήσης αυτών των ετικετών, παρέχοντας διάφορα επίπεδα δομής και στυλιστικής μορφοποίησης. Αν και η HTML παρέχει ένα βασικό επίπεδο μορφοποίησης, η εμφάνιση και το στυλ μπορούν να βελτιωθούν περαιτέρω με τη χρήση εσωτερικών ή εξωτερικών φύλλων στυλισμού (CSS). Συνολικά, η HTML αποτελεί ένα ισχυρό εργαλείο για τη δημιουργία ιστοσελίδων και τη μετάδοση πληροφοριών στο διαδίκτυο, προσφέροντας ένα σταθερό και δομημένο πλαίσιο για την οργάνωση και την παρουσίαση περιεχομένου.

2.5.2 CSS

Η CSS (Cascading Style Sheets) αναδεικνύεται ως κρίσιμο εργαλείο στον χώρο του web design και της ανάπτυξης ιστοσελίδων. Αν και η HTML παρέχει τη δομή και το περιεχόμενο μιας ιστοσελίδας, η CSS επιτρέπει την ανάθεση στυλ και εμφάνισης σε αυτό το περιεχόμενο, ενισχύοντας την εμπειρία του χρήστη και την αισθητική του ιστότοπου. Μέσω της CSS, μπορούμε να προσδιορίσουμε τα χρώματα, τις γραμματοσειρές, τις διαστάσεις και τις διάφορες γραφικές λεπτομέρειες του περιεχομένου, προσφέροντας ένα προσαρμοσμένο και εντυπωσιακό στυλ στην ιστοσελίδα. Η CSS επιτρέπει τη δημιουργία ενός συνεκτικού και συμβατικού στυλ σε ολόκληρη την ιστοσελίδα, βοηθώντας στη διατήρηση της συνοχής και της επαγγελματικής εικόνας του brand. Επιπλέον, η CSS επιτρέπει τη δημιουργία διαφόρων εφέ και animation για να κάνει την ιστοσελίδα πιο δυναμική και εντυπωσιακή. Αυτό μπορεί να περιλαμβάνει μεταβάσεις χρωμάτων, κίνηση σελίδας, εφέ hover και πολλά άλλα, που μπορούν να προσφέρουν μια μοναδική εμπειρία στους χρήστες. Συνολικά, η CSS αποτελεί κρίσιμο στοιχείο στον σχεδιασμό και την ανάπτυξη ιστοσελίδων, επιτρέποντας τη δημιουργία ευέλικτων, ευανάγνωστων και εντυπωσιακών ιστοσελίδων που ξεχωρίζουν στο διαδίκτυο.

2.5.3 JavaScript

Η JavaScript αναδεικνύεται ως κρίσιμο εργαλείο στον κόσμο της web ανάπτυξης, προσφέροντας δυνατότητες διαδραστικότητας και δυναμικότητας στις ιστοσελίδες. Αποτελεί μια προγραμματιστική γλώσσα που χρησιμοποιείται για την επεξεργασία και την αλληλεπίδραση με τα διάφορα στοιχεία της ιστοσελίδας, ενισχύοντας τη δυνατότητα του χρήστη να αλληλεπίδραση με την πλατφόρμα. Μέσω της JavaScript, μπορούμε να προσθέσουμε διάφορες λειτουργίες στην ιστοσελίδα, όπως δυναμικά περιεχόμενα, αλλαγή στην

εμφάνιση και τη συμπεριφορά της σελίδας ανάλογα με τις ενέργειες του χρήστη, επικοινωνία με διακομιστές για ανάκτηση και αποστολή δεδομένων χωρίς ανανέωση της σελίδας, και πολλά άλλα. Με την JavaScript, μπορούμε να δημιουργήσουμε δυναμικές εφαρμογές και ιστοσελίδες που προσφέρουν πλοήγηση με αποτελέσματα που εντυπωσιάζουν. Η γλώσσα αυτή επιτρέπει επίσης τη δημιουργία παιχνιδιών, εφαρμογών για κινητά, και πολλά άλλα επεκτάσιμα και προσαρμόσιμα έργα. Συνολικά, η JavaScript αποτελεί κρίσιμο εργαλείο στον χώρο της web ανάπτυξης, επιτρέποντας τη δημιουργία δυναμικών, διαδραστικών και εντυπωσιακών ιστοσελίδων και εφαρμογών που ανταποκρίνονται στις απαιτήσεις του σύγχρονου διαδικτύου.

2.5.4 PHP

Η PHP (Hypertext Preprocessor) είναι μια δυναμική γλώσσα προγραμματισμού που ευρέως χρησιμοποιείται για την ανάπτυξη διαδικτυακών εφαρμογών και ιστοσελίδων. Αποτελεί ένα από τα πιο δημοφιλή εργαλεία στον χώρο της web ανάπτυξης, καθώς παρέχει ευελιξία, δυνατότητες επεξεργασίας δεδομένων και διασύνδεσης με βάσεις δεδομένων. Η PHP επιτρέπει τη δημιουργία δυναμικών ιστοσελίδων, όπου οι χρήστες μπορούν να αλληλεπιδρούν με το περιεχόμενο και να λαμβάνουν προσαρμοσμένες απαντήσεις. Μέσω της PHP, μπορούμε να δημιουργήσουμε δυναμικές φόρμες επικοινωνίας, εφαρμογές συνομιλίας, συστήματα διαχείρισης περιεχομένου (CMS), ηλεκτρονικά καταστήματα και πολλά άλλα. Με την PHP, μπορούμε επίσης να διαχειριστούμε δεδομένα από βάσεις δεδομένων, όπως MySQL, PostgreSQL και άλλες, προσφέροντας έτσι δυνατότητες επεξεργασίας και αποθήκευσης δεδομένων με ασφάλεια και αποδοτικότητα. Συνολικά, η PHP αποτελεί ένα ισχυρό εργαλείο για την ανάπτυξη διαδικτυακών εφαρμογών και ιστοσελίδων, προσφέροντας ευελιξία, δυνατότητες προσαρμογής και απόδοσης που την καθιστούν αναπόσπαστο κομμάτι της web ανάπτυξης.



Εικόνα 4: php

3 Ανάλυση Κώδικα

Σε αυτήν την ενότητα θα αναλύσουμε γενικά όλα τα κομμάτια του κώδικα και κυρίως αυτά που χρειαζόντουσαν περισσότερο χρόνο για να δημιουργηθούν όπως για παράδειγμα το πως θα μεταφέρω ένα προϊόν στο καλάθι ή πως θα μεταφέρω ένα προϊόν στο wishlist.

3.1 Εγγραφή - Είσοδος - Έξοδος

Σε αυτήν την υποένότητα θα μιλήσουμε για το πως δημιουργήσαμε την εγγραφή του χρήστη, την είσοδο του λογαριασμού του στο eshop όπως και για την έξοδο του λογαριασμού του από αυτό.

Εγγραφή χρήστη

Αρχικά, για να δημιουργήσω το register έλεγξα αν ορίζεται ο user_id. Στην Συνέχεια, έλεγξα αν γίνεται υποβολή της φόρμας με if(isset(\$_POST['submit'])) και μέσα σε αυτό έκανα φιλτράρισμα κάποιων δεδομένων όπως του ονόματος, του email, του κωδικού και του επαναλαμβανόμενου κωδικού. Μετά, έκανα έλεγχο του email στην βάση δεδομένων, όπως επίσης έλεγχο αν υπάρχει ήδη στην βάση δεδομένων το email και αν οι κωδικοί πρόσβασης ταιριάζουν και στο τέλος κάνει εισαγωγή του νέου χρήστη στη βάση δεδομένων. Στο HTML κομμάτι δημιουργούμε την φόρμα εγγραφής του νέου χρήστη τοποθετώντας το όνομα, το email, τον κωδικό και το επαναλαμβανόμενο κωδικό.

```
if(isset($_SESSION['user_id'])){
    $user_id = $_SESSION['user_id'];
}else{
    $user_id = '';
};
if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $name = filter_var($name, FILTER_SANITIZE_STRING);
    $email = $_POST['email'];
    $email = filter_var($email, FILTER_SANITIZE_STRING);
    $pass = sha1($_POST['pass']);
    $pass = filter_var($pass, FILTER_SANITIZE_STRING);
    $cpass = sha1($_POST['cpass']);
    $cpass = filter_var($cpass, FILTER_SANITIZE_STRING);
```

```

        $select_user = $conn->prepare("SELECT * FROM `users` WHERE email = ?");
        $select_user->execute([$email,]);
        $row = $select_user->fetch(PDO::FETCH_ASSOC);

        if($select_user->rowCount() > 0){
            $message[] = 'email already exists!';
        }else{
            if($pass != $cpass){
                $message[] = 'confirm password not matched!';
            }else{
                $insert_user = $conn->prepare("INSERT INTO `users`(name, email, password)
VALUES(?,?,?)");
                $insert_user->execute([$name, $email, $cpass]);
                $message[] = 'registered successfully, login now please!';
            }
        }
    }
<section class="form-container">

    <form action="" method="post">
        <h3>register now</h3>
        <input type="text" name="name" required placeholder="enter your username"
maxlength="20" class="box">

        <input type="email" name="email" required placeholder="enter your email"
maxlength="50" class="box" oninput="this.value = this.value.replace(/\s/g, '')">

        <input type="password" name="pass" required placeholder="enter your password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
        <input type="password" name="cpass" required placeholder="confirm your pass-
word" maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
        <input type="submit" value="register now" class="btn" name="submit">

        <p>already have an account?</p>
        <a href="user_login.php" class="option-btn">login now</a>
    </form>
</section>

```

Εικόνα 5: κώδικας εγγραφής χρήστη

Είσοδος Χρήστη

Αρχικά όσον αφορά την είσοδο του χρήστη στο eshop έχει παρόμοιο κώδικα με την εγγραφή του χρήστη με καποιές μικροδιαφορές όπως στον έλεγχο υποβολής της φόρμας φιλτράρει μόνο τα δεδομένα του email και του κρυπτογραφημένου κωδικού, κάνει έλεγχο του χρήστη στη βάση δεδομένων και τέλος ελέγχει αν υπάρχει χρήστης στις παρεχόμενες πληροφορίες. Στο HTML κομμάτι η φόρμα που είναι Post περιέχει την εισαγωγή του email και του κωδικού πρόσβασης.

```

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

    $email = $_POST['email'];
    $email = filter_var($email, FILTER_SANITIZE_STRING);

```

```

$pass = sha1($_POST['pass']);
$pass = filter_var($pass, FILTER_SANITIZE_STRING);

$select_user = $conn->prepare("SELECT * FROM `users` WHERE email = ? AND password = ?");
$select_user->execute([$email, $pass]);
$row = $select_user->fetch(PDO::FETCH_ASSOC);

if($select_user->rowCount() > 0){
    $_SESSION['user_id'] = $row['id'];      header('location:home.php');    }else{
    $message[] = 'incorrect username or password!';
}
}

<section class="form-container">

    <form action="" method="post">
        <h3>login now</h3>

        <input type="email" name="email" required placeholder="enter your email"
maxlength="50" class="box" oninput="this.value = this.value.replace(/\s/g, '')">

        <input type="password" name="pass" required placeholder="enter your password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">

        <input type="submit" value="login now" class="btn" name="submit">

        <p>don't have an account?</p>
        <a href="user_register.php" class="option-btn">register now</a>
    </form>
</section>

```

Εικόνα 6: κώδικας εισόδου χρήστη

Έξοδος χρήστη

Αρχικά, για την αποσύνδεση του χρήστη από το eshop χρησιμοποιούμε την έναρξη μιας συνεδρίας μετά διαγράφουμε την συνεδρία και τέλος λήγουμε την συνεδρία και διαγράφονται όλα τα δεδομένα.

```

<?php
include 'connect.php';

session_start();
session_unset();
session_destroy();

header('location:../home.php');

?>

```

Εικόνα 7: κώδικας εξόδου χρήστη

Εγγραφή - Είσοδος - Έξοδος Διαχειριστή

Σε αυτήν την περίπτωση ο κώδικας για την εγγραφή διαχειριστή είναι σχεδόν πανομοιότυπος με του χρήστη με την διαφορά ότι κάνουμε έλεγχο για ύπαρξη ονόματος του χρήστη στη βάση δεδομένων του διαχειριστή. Για την είσοδο του διαχειριστή ισχύουν σχεδόν τα ίδια με την διαφορά ότι κάνουμε ανάκτηση διαχείρισης βάση ονόματος και κωδικού. Τέλος, η έξοδος του διαχειριστή έχει τον ίδιο κώδικα με την έξοδο του χρήστη

```
//Εγγραφή Διαχειριστή
$select_admin = $conn->prepare("SELECT * FROM `admins` WHERE name = ?");
$select_admin->execute([$name]);
if($select_admin->rowCount() > 0){
    $message[] = 'username already exist!';
}else{

    if($pass != $cpass){
        $message[] = 'confirm password not matched!';
    }else{

        $insert_admin = $conn->prepare("INSERT INTO `admins` (name, password)
VALUES(?,?)");
        $insert_admin->execute([$name, $cpass]);
        $message[] = 'new admin registered successfully!';
    }
}
// Είσοδος Διαχειριστή
$select_admin = $conn->prepare("SELECT * FROM `admins` WHERE name = ? AND password = ?");
$select_admin->execute([$name, $pass]);
```

Εικόνα 8: Κώδικας εισόδου-εγγραφής-εξόδου διαχειριστή

3.2 Γρήγορη Προβολή

Σε αυτό το κομμάτι του κώδικα μπορούμε να δούμε γενικότερα πιο λεπτομερείς το προϊόν το οποίο θα είναι σε πώληση καθώς θα έχει τις εικόνες και τα στοιχεία που διαθέτει το προϊόν. Αρχικά, για να κάνω γρήγορη προβολή του προϊόντος θα πρέπει να το επιλέξω με βάση το προϊόν του ID. Στη συνέχεια, δημιουργούμε μια φόρμα όπου θα έχει τις εικόνες, τα περιεχόμενα, όνομα, τιμή και ποσότητα του προϊόντος.

```
<?php

    $pid = $_GET['pid'];
    $select_products = $conn->prepare("SELECT * FROM `products` WHERE id = ?");
    $select_products->execute([$pid]);
    if($select_products->rowCount() > 0){
        while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
            ?>
<div class="main-image">
    
</div>
```

```

<div class="sub-image">
  
  
  
</div>

<div class="content">

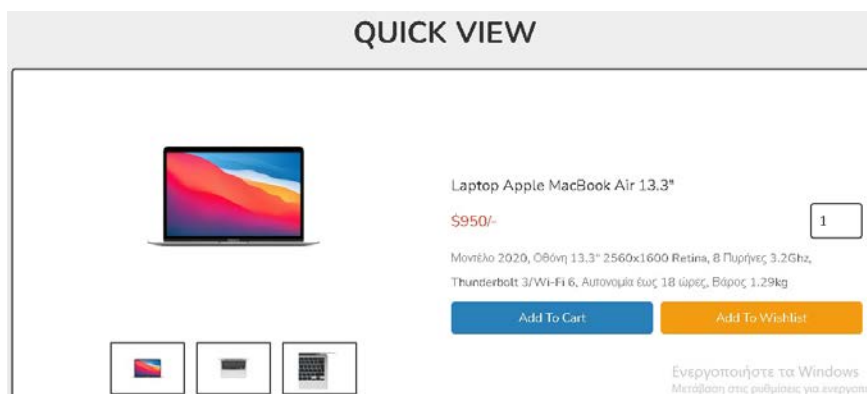
  <div class="name"><?= $fetch_product['name']; ?></div>
  <div class="flex">
    <div class="price"><span>$</span><?= $fetch_product['price']; ?><span>/-
</span></div>

    <input type="number" name="qty" class="qty" min="1" max="99" onkey-
press="if(this.value.length == 2) return false;" value="1">
  </div>

  <div class="details"><?= $fetch_product['details']; ?></div>

```

Εικόνα 9: κώδικας γρήγορης προβολής



Εικόνα 10: γρήγορη προβολή

3.3 Καλάθι αγορών και λίστα επιθυμιών

Σε αυτήν την ενότητα θα αναφερθούμε για το πώς δημιουργήσαμε το καλάθι αγορών και τη λίστα επιθυμιών αλλά και πως μετακινούμε τα προϊόντα από την απλή αγορά μέσα σε αυτό το καλάθι ή μέσα στην λίστα επιθυμιών.

3.3.1 Καλάθι αγορών και λίστα επιθυμιών

Αρχικά, όσον αφορά για το αν έχει μεταφερθεί ένα προϊόν στη λίστα επιθυμιών ελέγχει εάν το προϊόν είναι στη λίστα επιθυμιών ή στο καλάθι αγορών και αν είναι ή όχι εμφανίζει

ένα αντίστοιχο μήνυμα και τέλος κάνουμε εισαγωγή του προϊόντος στην λίστα επιθυμιών αν δεν είναι εκεί.

```
$check_wishlist_numbers = $conn->prepare("SELECT * FROM `wishlist` WHERE name = ? AND user_id = ?");
$check_wishlist_numbers->execute([$name, $user_id]);

$check_cart_numbers = $conn->prepare("SELECT * FROM `cart` WHERE name = ? AND user_id = ?");
$check_cart_numbers->execute([$name, $user_id]);

if($check_wishlist_numbers->rowCount() > 0){
    $message[] = 'already added to wishlist!';
}elseif($check_cart_numbers->rowCount() > 0){
    $message[] = 'already added to cart!';
}else{

    $insert_wishlist = $conn->prepare("INSERT INTO `wishlist`(user_id, pid, name, price, image) VALUES(?,?,?,?,?)");
    $insert_wishlist->execute([$user_id, $pid, $name, $price, $image]);
    $message[] = 'added to wishlist!';
}
```

Εικόνα 11: Κώδικας μεταφοράς προϊόντων στην λίστα επιθυμιών

Στην περίπτωση που μεταφέρουμε το προϊόν στο καλάθι αγορών ελέγχουμε εάν το προϊόν είναι στο καλάθι αγορών, εάν είναι να βγάλει ένα μήνυμα ότι υπάρχει και τέλος να εισάγει το προϊόν στο καλάθι αγορών αν δεν υπάρχει εκεί.

```
$check_cart_numbers = $conn->prepare("SELECT * FROM `cart` WHERE name = ? AND user_id = ?");
$check_cart_numbers->execute([$name, $user_id]);

if($check_cart_numbers->rowCount() > 0){
    $message[] = 'already added to cart!';
}else{

    $insert_cart = $conn->prepare("INSERT INTO `cart`(user_id, pid, name, price, quantity, image) VALUES(?,?,?,?,?,?)");
    $insert_cart->execute([$user_id, $pid, $name, $price, $qty, $image]);
    $message[] = 'added to cart!';
}
```

Εικόνα 12: Κώδικας μεταφοράς προϊόντων στο καλάθι αγορών

3.3.2 Λίστα επιθυμιών

Μέσα στη σελίδα της λίστας επιθυμιών υπάρχει μια φόρμα για την διαγραφή ενός συγκεκριμένου στοιχείου ώστε να γίνει διαγραφή και από την βάση δεδομένων. Επιπλέον, γί-

νεται διαγραφή όλων των αντικειμένων από την λίστα επιθυμιών του συγκεκριμένου χρήστη. Τέλος, επιλέγει όλα τα στοιχεία στην λίστα επιθυμιών του συγκεκριμένου χρήστη και βγάζει το συνολικό κόστος τους.

```
if(isset($_POST['delete'])){
    $wishlist_id = $_POST['wishlist_id'];

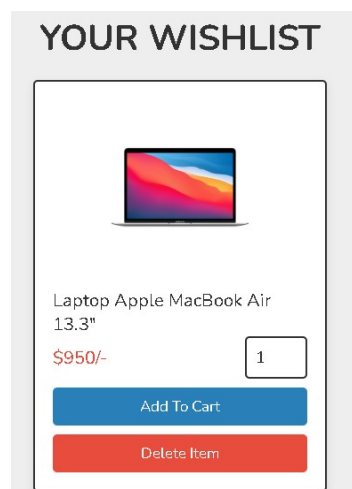
    $delete_wishlist_item = $conn->prepare("DELETE FROM `wishlist` WHERE id = ?");
    $delete_wishlist_item->execute([$wishlist_id]);
}

if(isset($_GET['delete_all'])){

    $delete_wishlist_item = $conn->prepare("DELETE FROM `wishlist` WHERE user_id = ?");
    $delete_wishlist_item->execute([$user_id]);
    header('location:wishlist.php');
}

$grand_total = 0;
$select_wishlist = $conn->prepare("SELECT * FROM `wishlist` WHERE user_id = ?");
$select_wishlist->execute([$user_id]);
if($select_wishlist->rowCount() > 0){
    while($fetch_wishlist = $select_wishlist->fetch(PDO::FETCH_ASSOC)){
        $grand_total += $fetch_wishlist['price'];
    }
}
```

Εικόνα 13: Κώδικας σελίδας λίστας επιθυμιών



Εικόνα 14: σελίδα λίστας επιθυμιών

3.3.3 Καλάθι αγορών

Όσον αφορά για την σελίδα εμφάνισης του καλαθιού αγορών μέσα σε αυτό υπάρχει ένα κουμπί για την διαγραφεί ενός στοιχείου από το καλάθι, υπάρχει επιπλέον και ένα κουμπί για την διαγραφή όλων των στοιχείων από το καλάθι και ένα κουμπί ενημέρωσης της

ποσότητας ενός στοιχείου στο καλάθι. Τέλος, επιλέγουμε όλα τα στοιχεία από τον πίνακα καλαθιού για τον συγκεκριμένο χρήστη ώστε να βρούμε το συνολικό κόστος τους.

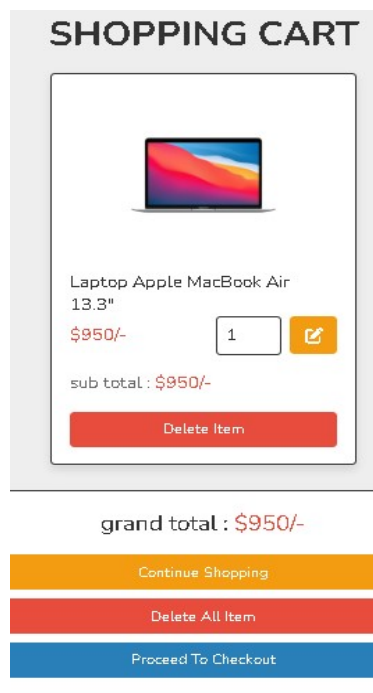
```

if(isset($_POST['delete'])){
    $cart_id = $_POST['cart_id'];
    $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE id = ?");
    $delete_cart_item->execute([$cart_id]);
}
if(isset($_GET['delete_all'])){
    $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE user_id = ?");
    $delete_cart_item->execute([$user_id]);
    header('location:cart.php');
}
if(isset($_POST['update_qty'])){
    $cart_id = $_POST['cart_id'];
    $qty = $_POST['qty'];
    $qty = filter_var($qty, FILTER_SANITIZE_STRING);
    $update_qty = $conn->prepare("UPDATE `cart` SET quantity = ? WHERE id = ?");
    $update_qty->execute([$qty, $cart_id]);
    $message[] = 'cart quantity updated';
}
<?php
    $grand_total = 0;

    $select_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
    $select_cart->execute([$user_id]);
    if($select_cart->rowCount() > 0){
        while($fetch_cart = $select_cart->fetch(PDO::FETCH_ASSOC)){
            <?
            <form action="" method="post" class="box">
            <input type="hidden" name="cart_id" value="<?= $fetch_cart['id']; ?>">
            <a href="quick_view.php?pid=<?= $fetch_cart['pid']; ?>" class="fas fa-eye"></a>
            
            <div class="name"><?= $fetch_cart['name']; ?></div>
            <div class="flex">
                <div class="price"><?= $fetch_cart['price']; ?></div>
                <input type="number" name="qty" class="qty" min="1" max="99" onkey-
press="if(this.value.length == 2) return false;" value="<?= $fetch_cart['quantity'];
?>">
                <button type="submit" class="fas fa-edit" name="update_qty"></button>
            </div>
            <div class="sub-total"> sub total : <span><?= $sub_total = ($fetch_cart['price']
* $fetch_cart['quantity']); ?></span> </div>
            <input type="submit" value="delete item" onclick="return confirm('delete this from
cart?');" class="delete-btn" name="delete">
            </form>
            <?php
                $grand_total += $sub_total;
            }

```

Εικόνα 15: Κώδικας σελίδας καλαθιού αγορών



Εικόνα 16: Σελίδα καλαθιού αγορών

3.3.4 Από το καλάθι αγορών στην αγορά

Σε αυτήν την περίπτωση μεταφέρουμε όλα τα προϊόντα που έχουμε τοποθετήσει στο καλάθι για μετέπειτα αγορά αυτών. Επίσης, όσον αφορά τον κώδικα κάνουμε έλεγχο του καλαθιού για προϊόντα, αν δεν είναι άδειο το καλάθι τότε τοποθετούμε την παραγγελία στη βάση δεδομένων και έτσι διαγράφουμε όλα τα προϊόντα από το καλάθι του χρήστη. Τέλος, εμφανίζουμε και ένα μήνυμα ότι η παραγγελία ολοκληρώθηκε αλλιώς θα εμφανίσουμε ένα μήνυμα ότι το καλάθι είναι άδειο.

```
$check_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
$check_cart->execute([$user_id]);

if($check_cart->rowCount() > 0){

    $insert_order = $conn->prepare("INSERT INTO `orders`(user_id, name, number, email,
method, address, total_products, total_price) VALUES(?,?,?,?,?,?,?,?)");
    $insert_order->execute([$user_id, $name, $number, $email, $method, $address, $total_products, $total_price]);

    $delete_cart = $conn->prepare("DELETE FROM `cart` WHERE user_id = ?");
    $delete_cart->execute([$user_id]);

    $message[] = 'order placed successfully!';
}else{
    $message[] = 'your cart is empty';
}
```

Εικόνα 17: κώδικας αγοράς προϊόντων

PLACE YOUR ORDERS

your name :

enter your name

your number :

enter your number

your email :

enter your email

payment method :

cash on delivery

address line 01 :

e.g. flat number

address line 02 :

e.g. street name

city :

e.g. Larissa

state :

e.g. thessaly

country :

e.g. Greece

pin code :

e.g. 123456

Place Order

Εικόνα 18: Στοιχεία παραγγελίας προϊόντων

3.4 Σελίδα αναζήτησης και αγοράς προϊόντων

3.4.1 Σελίδα αναζήτησης

Σε αυτήν την περίπτωση μπορούμε να κάνουμε αναζήτηση προϊόντων τα οποία μπορεί να υπάρχουν στο eshop. Αρχικά, δημιουργούμε μια φόρμα αναζήτησης προϊόντων, μετά ελέγχουμε αν έχει γίνει υποβολή της φόρμας αναζήτησης, κάνουμε λήψη του όρου αναζήτησης από την φόρμα, εκτελούμε το ερώτημα SQL για αναζήτηση των προϊόντων με βάση το όνομα και τέλος ελέγχουμε αν βρέθηκαν αποτελέσματα.

```

<section class="search-form">
  <form action="" method="post">
    <input type="text" name="search_box" placeholder="search here..." maxlength="100"
class="box" required>
    <button type="submit" class="fas fa-search" name="search_btn"></button>
  </form>
</section>
<?php
  if(isset($_POST['search_box']) OR isset($_POST['search_btn']))){

    $search_box = $_POST['search_box'];

    $select_products = $conn->prepare("SELECT * FROM `products` WHERE name LIKE
'${$search_box}%'");
    $select_products->execute();

    if($select_products->rowCount() > 0){
      while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
        ?>

```

Εικόνα 19: κώδικας σελίδας αναζήτησης

3.4.2 Αγορά προϊόντων

Σε αυτήν την σελίδα του eshop έχουμε όλα τα προϊόντα που έχουμε προσθέσει. Πρώτα, εκτελούμε ερώτημα SQL για ανάκτηση των πιο πρόσφατων προϊόντων, ελέγχουμε εάν υπάρχουν αποτελέσματα και εμφανίζουμε κάθε προϊόν με ένα loop. Τέλος, δημιουργούμε και μια φόρμα προσθήκης προϊόντων στο καλάθι το οποίο περιέχει κουμπί για λίστα επιθυμιών και γρήγορη προβολή, εικόνα προϊόντος, όνομα προϊόντος και τιμή προϊόντος.

```
<?php

$select_products = $conn->prepare("SELECT * FROM `products`");
$select_products->execute();

if($select_products->rowCount() > 0){

    while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
        ?>

        <form action="" method="post" class="box">
            <input type="hidden" name="pid" value="<?= $fetch_product['id']; ?>">
            <input type="hidden" name="name" value="<?= $fetch_product['name']; ?>">
            <input type="hidden" name="price" value="<?= $fetch_product['price']; ?>">
            <input type="hidden" name="image" value="<?= $fetch_product['image_01']; ?>">

            <button class="fas fa-heart" type="submit" name="add_to_wishlist"></button>

            <a href="quick_view.php?pid=<?= $fetch_product['id']; ?>" class="fas fa-eye"></a>

            <div class="name"><?= $fetch_product['name']; ?></div>
            <!-- Τιμή προϊόντος -->
            <div class="flex">
                <div class="price"><span>$</span><?= $fetch_product['price']; ?><span>/-
</span></div>

                <input type="number" name="qty" class="qty" min="1" max="99" onkey-
press="if(this.value.length == 2) return false;" value="1">
            </div>

            <input type="submit" value="add to cart" class="btn" name="add_to_cart">
        </form>
        <?php
        }
        }else{

            echo '<p class="empty">no products found!</p>';
        }
    ?>
</div>
```

Εικόνα 20: κώδικας αγοράς προϊόντων

3.5 Επικοινωνία - Μηνύματα

3.5.1 Επικοινωνία των χρηστών με τους διαχειριστές

Για να μπορούν οι χρήστες να επικοινωνήσουν με τους διαχειριστές δημιουργήσα ένα κώδικα οπου ελέγχει εάν έχει γίνει υποβολή της φόρμας επικοινωνίας μέσα σε αυτό κάνουμε λήψη των δεδομένων από την φόρμα και τα φιλτράρουμε για να έχουμε ασφάλεια, μετά κάνουμε έλεγχο αν το μήνυμα έχει αποσταλεί από τον χρήστη, δημιουργούμε πάλι ένα μήνυμα για το είδη απεσταλμένο μήνυμα, εισάγουμε το μήνυμα στην βάση δεδομένων και τέλος δημιουργούμε και μια φόρμα αναζήτησης.

```
if(isset($_POST['send'])){\n\n    $name = $_POST['name'];\n    $name = filter_var($name, FILTER_SANITIZE_STRING);\n    $email = $_POST['email'];\n    $email = filter_var($email, FILTER_SANITIZE_STRING);\n    $number = $_POST['number'];\n    $number = filter_var($number, FILTER_SANITIZE_STRING);\n    $msg = $_POST['msg'];\n    $msg = filter_var($msg, FILTER_SANITIZE_STRING);\n\n    $select_message = $conn->prepare("SELECT * FROM `messages` WHERE name = ? AND email\n= ? AND number = ? AND message = ?");\n    $select_message->execute([$name, $email, $number, $msg]);\n\n    if($select_message->rowCount() > 0){\n        $message[] = 'already sent message!';\n    }else{\n\n        $insert_message = $conn->prepare("INSERT INTO `messages`(user_id, name, email,\nnumber, message) VALUES(?,?,?,?,?)");\n        $insert_message->execute([$user_id, $name, $email, $number, $msg]);\n\n        $message[] = 'sent message successfully!';\n    }\n}\n<form action="" method="post">\n    <h3>get in touch</h3>\n    <input type="text" name="name" placeholder="enter your name" required\nmaxlength="20" class="box">\n    <input type="email" name="email" placeholder="enter your email" required\nmaxlength="50" class="box">\n    <input type="number" name="number" min="0" max="999999999" placeholder="enter\nyour number" required onkeypress="if(this.value.length == 10) return false;"\nclass="box">\n    <textarea name="msg" class="box" placeholder="enter your message" cols="30"\nrows="10"></textarea>\n    <input type="submit" value="send message" name="send" class="btn">\n</form>
```

Εικόνα 21: κώδικας επικοινωνίας χρηστών με διαχειριστές

Εικόνα 22: σελίδα μηνύματος προς διαχειριστές

3.5.2 Μηνύματα που παίρνουν οι διαχειριστές από τους χρήστες

Σε αυτήν την περίπτωση για να μπορούμε να βλέπουμε τα μηνύματα που μας στέλνει ο χρήστης αρχικά κάνουμε ανάκτηση του ID του διαχειριστή από τη συνεδρία και αν δεν έχει οριστεί τον ανακατευθύνουμε προς την σελίδα σύνδεσης του διαχειριστή. Στη συνέχεια, εάν υπάρχει παράμετρο διαγραφής στο URL τότε κάνουμε ανάκτηση του ID που πρέπει να διαγραφεί, μετά εκτελούμε ερώτημα για διαγραφή μηνύματος με βάση το ID και αφού γίνει η διαγραφή τον πηγαίνουμε στην σελίδα μηνυμάτων. Τέλος, εκτελούμε ένα ερώτημα για να ανακτήσουμε όλα τα μηνύματα και φτιάχνουμε έναν σύνδεσμο διαγραφής μηνύματος με παράμετρο delete το ID του μηνύματος

```
$admin_id = $_SESSION['admin_id'];

if(!isset($admin_id)){
    header('location:admin_login.php');
};

if(isset($_GET['delete'])){
    $delete_id = $_GET['delete'];
    $delete_message = $conn->prepare("DELETE FROM `messages` WHERE id = ?");
    $delete_message->execute([$delete_id]);
    header('location:messages.php');
}

<?php

$select_messages = $conn->prepare("SELECT * FROM `messages`");
$select_messages->execute();
if($select_messages->rowCount() > 0){
    while($fetch_message = $select_messages->fetch(PDO::FETCH_ASSOC)){
        ?>
        <div class="box">
            <p> user id : <span><?= $fetch_message['user_id']; ?></span></p>
```

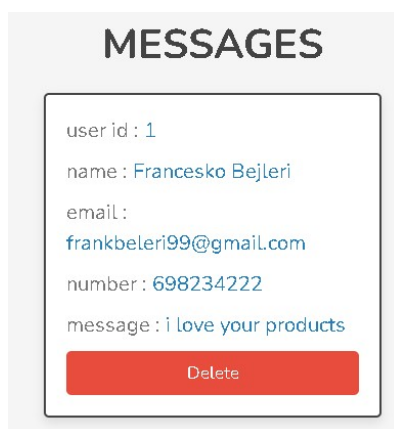
```

<p> name : <span><?= $fetch_message['name']; ?></span></p>
<p> email : <span><?= $fetch_message['email']; ?></span></p>
<p> number : <span><?= $fetch_message['number']; ?></span></p>
<p> message : <span><?= $fetch_message['message']; ?></span></p>

<a href="messages.php?delete=<?= $fetch_message['id']; ?>" onclick="return con-
firm('delete this message?');" class="delete-btn">delete</a>
</div>
<?php
}
}else{
    echo '<p class="empty">you have no messages</p>';
}
?>

```

Εικόνα 23: κώδικας μηνυμάτων προς διαχειριστές



Εικόνα 24: σελίδα μηνυμάτων προς διαχειριστές

3.6 Αρχική Σελίδα - Javascript κώδικας

3.6.1 Αρχική σελίδα

Τα πράγματα που έχουμε τοποθετήσει για την αρχική σελίδα είναι ότι προσθήσαμε ένα στοιχείο slider όπου μας δείχνουν τα πιο πρόσφατα προϊόντα. Στη συνέχεια, τοποθετήσαμε παρακάτω και τις κατηγορίες των προϊόντων όπου μπορούμε να τις επιλέξουμε κάνοντας slide. Επιπλέον, έχουμε και ένα τμήμα όπου έχουμε τοποθετήσει τα πιο πρόσφατα προϊόντα και επιλέγοντας κυρίως τα 6 πιο πρόσφατα από τη βάση δεδομένων όπως και να κάνουμε έλεγχο εάν υπάρχουν εγγραφές στην επιλογή των προϊόντων. Τέλος, μέσω javascript κώδικα δημιούργησα ένα swiper για τις κατηγορίες προϊόντων, για τα τελευταία προϊόντα και για την κύρια εικόνα στο homepage.

```
<div class="swiper-slide slide">
```

```

        <div class="image">
            
        </div>
        <div class="content">
            <span>upto 50% off</span>
            <h3>latest smartphones</h3>
            <a href="shop.php" class="btn">shop now</a>
        </div>
    </div>
    <a href="category.php?category=laptop" class="swiper-slide slide">
        
        <h3>laptop</h3>
    </a>
    <a href="category.php?category=tv" class="swiper-slide slide">
        
        <h3>tv</h3>
    </a>
    <a href="category.php?category=camera" class="swiper-slide slide">
        
        <h3>camera</h3>
    </a>
    <a href="category.php?category=mouse" class="swiper-slide slide">
        
        <h3>mouse</h3>
    </a>
<?php
    $select_products = $conn->prepare("SELECT * FROM `products` LIMIT 6");
    $select_products->execute();
    if($select_products->rowCount() > 0){
        while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
            ?>
var swiper = new Swiper(".category-slider", {
    loop:true,
    spaceBetween: 20,
    pagination: {
        el: ".swiper-pagination",
        clickable:true,
    },
    navigation: {
        nextEl: '.swiper-button-next',
        prevEl: '.swiper-button-prev',
    },
    breakpoints: {
        0: {
            slidesPerView: 2,
        },
        650: {
            slidesPerView: 3,
        },
        768: {
            slidesPerView: 4,
        },
        1024: {
            slidesPerView: 5,
        },
    },
});
});

```

Εικόνα 25: κώδικας αρχικής σελίδας

3.6.2 Javascript κώδικας

Σε αυτό το javascript κώδικα έχω τοποθετήσει να κάνουμε την επιλογή του μενού πλοήγησης και του προφίλ, όταν κάνουμε κλικ σε ένα κουμπί να κάνει εναλλαγή της κλάσης active προσθέτοντας ή αφαιρώντας την κλάση active από το μενού πλοήγησης και αφαίρεση της κλάσης active από το προφίλ. Στη συνέχεια, όσον αφορά όταν κάνουμε κλικ στο κουμπί του χρήστη γίνεται προσθήκη ή αφαίρεση της κλάσης active από το προφίλ και αφαίρεση της κλάσης active από το προφίλ. Επίσης, κάνουμε και αφαίρεση της κλάσης active κατά την κύλιση της σελίδας αφαιρώντας την κλάση από το μένου πλοήγησης και από το προφίλ. Επιπλέον, γράφουμε κώδικα για επιλογή της κύριας εικόνας και των υποεικόνων στην γρήγορη προβολή. Τέλος, γίνεται εναλλαγή της κύριας εικόνας με το κλικ σε υποεικόνες κάνοντας λήψη της διεύθυνσης της εικόνας και ορισμό της διεύθυνσης ως την νέα κύρια εικόνα.

```
let navbar = document.querySelector('.header .flex .navbar');
let profile = document.querySelector('.header .flex .profile');

document.querySelector('#menu-btn').onclick = () =>{
  navbar.classList.toggle('active');
  profile.classList.remove('active');
}

document.querySelector('#user-btn').onclick = () =>{
  profile.classList.toggle('active');
  navbar.classList.remove('active');
}

window.onscroll = () =>{
  navbar.classList.remove('active');    profile.classList.remove('active');
}

let mainImage = document.querySelector('.quick-view .box .row .image-container .main-image img');
let subImages = document.querySelectorAll('.quick-view .box .row .image-container .sub-image img');

subImages.forEach(images =>{
  images.onclick = () =>{
    src = images.getAttribute('src');
    mainImage.src = src;
  }
});
```

Εικόνα 26: κώδικας javascript

3.7 Παραγγελίες

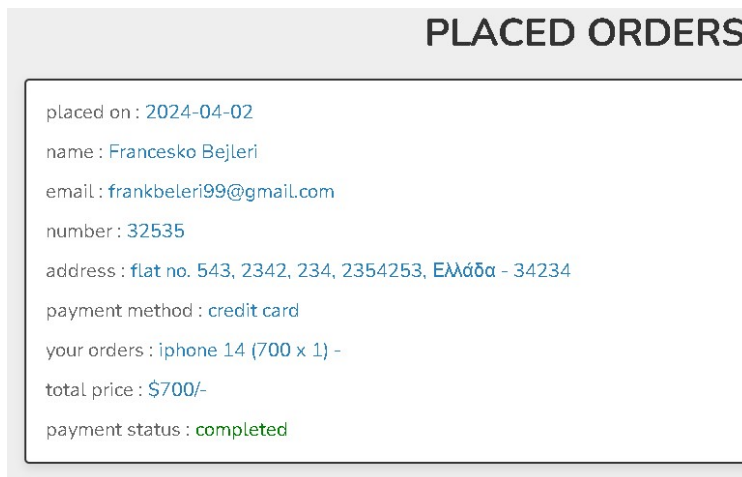
3.7.1 Παραγγελίες χρηστών

Αρχικά, δημιούργησα μια σελίδα στο eshop μου ώστε ο χρήστης να μπορεί να βλέπει τις παραγγελίες που κάνει. Μέσα στον κώδικα κάνω έλεγχο αν ο χρήστης είναι συνδεδεμένος και να του εμφανιστεί ένα μήνυμα ότι πρέπει να συνδεθεί, μετά προετοιμάζουμε και εκτελούμε το SQL query για την ανάκτηση των παραγγελιών του χρήστη, ελέγχουμε αν υπάρχουν αποτελέσματα από το query και τέλος το επαναλαμβάνουμε για κάθε εγγραφή που επιστρέφεται από το query και εμφανίζουμε τις λεπτομέρειες της παραγγελίας. Στη συνέχεια, χρησιμοποιούμε το fetch για να μπορέσει ο χρήστης να δει τα στοιχεία της παραγγελίας του. Τέλος, αν δεν υπάρχουν παραγγελίες να εμφανιστεί το αντίστοιχο μήνυμα.

```
<div class="box-container">

    <?php
        if($user_id == ''){
            echo '<p class="empty">please login to see your orders</p>';
        }else{
            $select_orders = $conn->prepare("SELECT * FROM `orders` WHERE user_id = ?");
            $select_orders->execute([$user_id]);
            if($select_orders->rowCount() > 0){
                while($fetch_orders = $select_orders->fetch(PDO::FETCH_ASSOC)){
                    ?>
                    <div class="box">
                        <p>placed on : <span><?= $fetch_orders['placed_on']; ?></span></p>
                        <p>name : <span><?= $fetch_orders['name']; ?></span></p>
                        <p>email : <span><?= $fetch_orders['email']; ?></span></p>
                        <p>number : <span><?= $fetch_orders['number']; ?></span></p>
                        <p>address : <span><?= $fetch_orders['address']; ?></span></p>
                        <p>payment method : <span><?= $fetch_orders['method']; ?></span></p>
                        <p>your orders : <span><?= $fetch_orders['total_products']; ?></span></p>
                        <p>total price : <span><?= $fetch_orders['total_price']; ?></span></p>
                        <p> payment status : <span style="color:<?php if($fetch_orders['payment_status']
= 'pending'){ echo 'red'; }else{ echo 'green'; }; ?>"><?= $fetch_orders['payment_sta-
tus']; ?></span> </p>
                    </div>
                <?php
                    }
                }else{
                    echo '<p class="empty">no orders placed yet!</p>';
                }
            }
        }
    <?php
    </div>
```

Εικόνα 27: κώδικας παραγγελίας χρηστών



Εικόνα 28: σελίδα παρακολούθησης παραγγελίας

3.7.2 Παραγγελίες που παρακολουθούν οι διαχειριστές

Σε αυτήν την σελίδα οι διαχειριστές ελέγχουν γενικότερα αν η παραγγελία ολοκληρώθηκε ώστε να την ενημερώσουν και να την διαγράψουν. Αρχικά, δημιουργούμε μια φόρμα για ενημέρωση της κατάστασης πληρωμής, μέσα σε αυτό κάνουμε ανάκτηση του ID παραγγελίας και της νέας κατάστασης πληρωμής από την φόρμα, καθαρισμό της μεταβλητής \$payment_status από ενδεχόμενα επικίνδυνα δεδομένα, εκτέλεση ενημέρωσης της κατάστασης πληρωμής της παραγγελίας στη βάση δεδομένων και μια προσθήκη μηνύματος επιτυχούς ενημέρωσης κατάστασης πληρωμής. Επίσης, για να διαγράψουμε μια παραγγελία κάνουμε ανάκτηση του ID παραγγελίας προς διαγραφή και εκτελούμε ερώτημα για διαγραφή παραγγελίας από την βάση δεδομένων. Μέσα στην σελίδα του διαχειριστή θα γίνει μια εκτέλεση ερωτήματος όλων των παραγγελιών και θα εμφανίζονται οι πληροφορίες των παραγγελιών με κουμπί ενημέρωσης και διαγραφής.

```
if(isset($_POST['update_payment'])){
    $order_id = $_POST['order_id'];
    $payment_status = $_POST['payment_status'];
    $payment_status = filter_var($payment_status, FILTER_SANITIZE_STRING);
    $update_payment = $conn->prepare("UPDATE `orders` SET payment_status = ? WHERE id = ?");
    $update_payment->execute([$payment_status, $order_id]);
    $message[] = 'payment status updated!';
}

if(isset($_GET['delete'])){
    $delete_id = $_GET['delete'];
    $delete_order = $conn->prepare("DELETE FROM `orders` WHERE id = ?");
    $delete_order->execute([$delete_id]);
    header('location:placed_orders.php');
}
<?php
$select_orders = $conn->prepare("SELECT * FROM `orders`");
```

```

$select_orders->execute();
if($select_orders->rowCount() > 0){
    while($fetch_orders = $select_orders->fetch(PDO::FETCH_ASSOC)){
        ?>
        <div class="box">
            <p> placed on : <span><?= $fetch_orders['placed_on']; ?></span> </p>
            <p> name : <span><?= $fetch_orders['name']; ?></span> </p>
            <p> number : <span><?= $fetch_orders['number']; ?></span> </p>
            <p> address : <span><?= $fetch_orders['address']; ?></span> </p>
            <p> total products : <span><?= $fetch_orders['total_products']; ?></span> </p>
            <p> total price : <span><?= $fetch_orders['total_price']; ?></span> </p>
            <p> payment method : <span><?= $fetch_orders['method']; ?></span> </p>
            <form action="" method="post">
                <input type="hidden" name="order_id" value="<?= $fetch_orders['id']; ?>">
                <select name="payment_status" class="select">

                    <option selected disabled><?= $fetch_orders['payment_status']; ?></option>
                    <option value="pending">pending</option>
                    <option value="completed">completed</option>
                </select>
                <div class="flex-btn">
                    <input type="submit" value="update" class="option-btn" name="update_pay-
ment">
                    <a href="placed_orders.php?delete=<?= $fetch_orders['id']; ?>" class="de-
lete-btn" onclick="return confirm('delete this order?');">delete</a>
                </div>
            </form>
        </div>
    }
}

```

Εικόνα 29: κώδικας παρακολούθησης παραγγελιών από διαχειριστές

PLACED ORDERS

placed on : 2024-04-02

name : Francesko Bejleri

number : 32535

address : flat no. 543, 2342, 234, 2354253, Ελλάδα - 34234

total products : iphone 14 (700 x 1) -

total price : \$700/-

payment method : credit card

completed

Update Delete

Εικόνα 30: σελίδα παρακολούθησης παραγγελιών του διαχειριστή

3.8 Καταχώρηση προϊόντων στο eshop

Σε αυτήν την υποενότητα θα αναφέρουμε ένα τρόπο με τον οποίο τοποθετούμε τα προϊόντα στο eshop μέσω μια φόρμας όπου τοποθετούμε εικόνες, τιμή και περιγραφή από την

σελίδα του διαχειριστεί. Αρχικά, κάνουμε ανάκτηση δεδομένων από την φόρμα, ανάκτηση εικόνων από την φόρμα και ελέγχουμε το μέγεθος. Στη συνέχεια, ελέγχουμε για ύπαρξη προϊόντος με το ίδιο όνομα και αν δεν υπάρχει εισάγουμε νέο προϊόν στη βάση δεδομένων. Επιπλέον, για την διαγραφή ενός προϊόντος κάνουμε ανάκτηση του ID, ελέγχουμε τις εικόνες των προϊόντων από τη βάση δεδομένων και διαγράφουμε τα προϊόντα από τη βάση δεδομένων, το καλάθι αγορών και την λίστα επιθυμιών. Επίσης, για να μπορούμε να προσθέσουμε το όνομα του προϊόντος, την τιμή και τις εικόνες φτιάχνουμε μια φόρμα. Τέλος, δημιουργούμε μια ακόμη φόρμα για να εμφανιστούν αυτά τα ονόματα, οι εικόνες και οι τιμές των προϊόντων.

```
$name = $_POST['name'];
$name = filter_var($name, FILTER_SANITIZE_STRING);
$price = $_POST['price'];
$price = filter_var($price, FILTER_SANITIZE_STRING);
$details = $_POST['details'];
$details = filter_var($details, FILTER_SANITIZE_STRING);

$image_01 = $_FILES['image_01']['name'];
$image_01 = filter_var($image_01, FILTER_SANITIZE_STRING);
$image_size_01 = $_FILES['image_01']['size'];
$image_tmp_name_01 = $_FILES['image_01']['tmp_name'];
$image_folder_01 = '../uploaded_img/'.$image_01;
$select_products = $conn->prepare("SELECT * FROM `products` WHERE name = ?");
$select_products->execute([$name]);
if($select_products->rowCount() > 0){
    $message[] = 'product name already exist!';
}else{
    $insert_products = $conn->prepare("INSERT INTO `products` (name, details, price, image_01, image_02, image_03) VALUES(?,?,?,?,?,?)");
    $insert_products->execute([$name, $details, $price, $image_01, $image_02, $image_03]);
    if(isset($_GET['delete'])){
        $delete_id = $_GET['delete'];
        $delete_product_image = $conn->prepare("SELECT * FROM `products` WHERE id = ?");
        $delete_product_image->execute([$delete_id]);
        $fetch_delete_image = $delete_product_image->fetch(PDO::FETCH_ASSOC);
        $delete_product = $conn->prepare("DELETE FROM `products` WHERE id = ?");
        $delete_product->execute([$delete_id]);

        $delete_cart = $conn->prepare("DELETE FROM `cart` WHERE pid = ?");
        $delete_cart->execute([$delete_id]);
        $delete_wishlist = $conn->prepare("DELETE FROM `wishlist` WHERE pid = ?");
        $delete_wishlist->execute([$delete_id]);

        header('location:products.php');
    }
}
<div class="inputBox">
    <span>product name (required)</span>
    <input type="text" class="box" required maxlength="100" placeholder="enter product name" name="name">
</div>
<div class="inputBox">
    <span>product price (required)</span>
```

```

        <input type="number" min="0" class="box" required max="9999999999" place-
holder="enter product price" onkeypress="if(this.value.length == 10) return false;"
name="price">
    </div>
    <div class="inputBox">
        <span>image 01 (required)</span>
        <input type="file" name="image_01" accept="image/jpg, image/jpeg, image/png,
image/webp" class="box" required>
    </div>
    <div class="box">
        
        <div class="name"><?=$fetch_products['name']; ?></div>
        <div class="price"><span><?=$fetch_products['price']; ?></span>/-</div>
        <div class="details"><span><?=$fetch_products['details']; ?></span></div>

    <div class="flex-btn">
        <a href="update_product.php?update=<?=$fetch_products['id']; ?>" class="op-
tion-btn">update</a>
        <a href="products.php?delete=<?=$fetch_products['id']; ?>" class="delete-btn"
onClick="return confirm('delete this product?');">delete</a>
    </div>

```

Εικόνα 31: κώδικας εισαγωγής προϊόντων

Εικόνα 32: σελίδα εισαγωγής προϊόντων

3.9 Ενημέρωση προφίλ χρήστη & διαχειριστή

3.9.1 Ενημέρωση προφίλ χρήστη

Σε αυτήν την περίπτωση ο χρήστης μπορεί να ενημερώνει το προφίλ του αλλάζοντας τον κωδικό πρόσβασης. Αρχικά, κάνουμε λήψη των τιμών από την φόρμα και φιλτράρουμε τα δεδομένα, στη συνέχεια ενημερώνουμε το προφίλ του χρήστη στη βάση δεδομένων και κάνουμε και έλεγχο για το αν θα γίνει ή αν δεν θα γίνει η ενημέρωση του κωδικού

πρόσβασης. Τέλος, δημιουργούμε και μια φόρμα ενημέρωσης του χρήστη για αλλαγή του κωδικού.

```
if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $name = filter_var($name, FILTER_SANITIZE_STRING);
    $email = $_POST['email'];
    $email = filter_var($email, FILTER_SANITIZE_STRING);

    $update_profile = $conn->prepare("UPDATE `users` SET name = ?, email = ? WHERE id = ?");
    $update_profile->execute([$name, $email, $user_id]);

    $empty_pass = 'da39a3ee5e6b4b0d3255bfef95601890afd80709';
    $prev_pass = $_POST['prev_pass'];
    $old_pass = sha1($_POST['old_pass']);
    $old_pass = filter_var($old_pass, FILTER_SANITIZE_STRING);
    $new_pass = sha1($_POST['new_pass']);
    $new_pass = filter_var($new_pass, FILTER_SANITIZE_STRING);
    $cpass = sha1($_POST['cpass']);
    $cpass = filter_var($cpass, FILTER_SANITIZE_STRING);

    if($old_pass == $empty_pass){
        $message[] = 'please enter old password!';
    }elseif($old_pass != $prev_pass){
        $message[] = 'old password not matched!';
    }elseif($new_pass != $cpass){
        $message[] = 'confirm password not matched!';
    }else{
        if($new_pass != $empty_pass){
            $update_admin_pass = $conn->prepare("UPDATE `users` SET password = ? WHERE id = ?");
            $update_admin_pass->execute([$cpass, $user_id]);
            $message[] = 'password updated successfully!';
        }else{
            $message[] = 'please enter a new password!';
        }
    }
}

<form action="" method="post">
    <h3>update now</h3>
    <input type="hidden" name="prev_pass" value="<?= $fetch_profile["password"]; ?>">
    <input type="text" name="name" required placeholder="enter your username"
maxlength="20" class="box" value="<?= $fetch_profile["name"]; ?>">
    <input type="email" name="email" required placeholder="enter your email"
maxlength="50" class="box" oninput="this.value = this.value.replace(/\s/g, '')"
value="<?= $fetch_profile["email"]; ?>">
    <input type="password" name="old_pass" placeholder="enter your old password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
    <input type="password" name="new_pass" placeholder="enter your new password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
    <input type="password" name="cpass" placeholder="confirm your new password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
    <input type="submit" value="update now" class="btn" name="submit">
</form>
```

Εικόνα 33: κώδικας ενημέρωσης προφίλ χρήστη

Εικόνα 34: σελίδα ενημέρωσης προφίλ χρήστη

3.9.2 Ενημέρωση του προφίλ του διαχειριστή

Έχει σχεδόν την ίδια λογική με αυτό του χρήστη απλά υπάρχουν μικρό αλλαγές. Η διαφορά είναι ότι κάνουμε μόνο λήψη του ονόματος από τη φόρμα και φιλτράρισμα και μετά κάνουμε ενημέρωση του ονόματος του διαχειριστή στη βάση δεδομένων ενώ όλα τα προηγούμενα για την αλλαγή του κωδικού είναι ίδια εκτός ότι γίνεται ενημέρωση του κωδικού πρόσβασης του διαχειριστή στη βάση δεδομένων. Τέλος, δημιουργούμε μια φόρμα για να μπορεί να γίνει η αλλαγή του κωδικού του διαχειριστή.

```
$name = $_POST['name'];
$name = filter_var($name, FILTER_SANITIZE_STRING);
$update_profile_name = $conn->prepare("UPDATE `admins` SET name = ? WHERE id = ?");
$update_profile_name->execute([$name, $admin_id]);
$update_admin_pass = $conn->prepare("UPDATE `admins` SET password = ? WHERE id = ?");
$update_admin_pass->execute([$confirm_pass, $admin_id]);
$message[] = 'password updated successfully!';
<form action="" method="post">
  <h3>update profile</h3>
  <!-- Κρυφό πεδίο για τον προηγούμενο κωδικό πρόσβασης -->
  <input type="hidden" name="prev_pass" value="<?=$fetch_profile['password']; ?>">
  <input type="text" name="name" value="<?=$fetch_profile['name']; ?>" required
  placeholder="enter your username" maxlength="20" class="box" oninput="this.value =
  this.value.replace(/\s/g, '')">
  <input type="password" name="old_pass" placeholder="enter old password"
  maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
  <input type="password" name="new_pass" placeholder="enter new password"
  maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
  <input type="password" name="confirm_pass" placeholder="confirm new password"
  maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, '')">
  <input type="submit" value="update now" class="btn" name="submit">
</form>
```

Εικόνα 35: Ενημέρωση προφίλ διαχειριστή

3.10 Λογαριασμοί Χρηστών - Διαχειριστών

3.10.1 Λογαριασμοί Χρηστών

Σε αυτό το μέρος οι διαχειριστές έχουν την δυνατότητα να μπορούν να διαγράψουν πράγματα που αφορούν τους χρήστες όπως να διαγράψουν το χρήστη από το πίνακα “users”, να διαγράψει την παραγγελία του χρήστη από το πίνακα “orders”, να κάνει διαγραφεί τον μηνυμάτων του χρήστη από τον πίνακα “messages”, να διαγράψει το καλάθι του χρήστη από τον πίνακα “cart” και διαγραφής λίστας επιθυμιών του χρήστη από τον πίνακα “wishlist”. Τέλος, στο php επιλέγουμε όλους τους λογαριασμούς χρηστών, κάνουμε έλεγχο αν υπάρχουν οι λογαριασμοί και δείχνουμε τις πληροφορίες του χρήστη και το κουμπί διαγραφής.

```
if(isset($_GET['delete'])){
    $delete_id = $_GET['delete'];
    $delete_user = $conn->prepare("DELETE FROM `users` WHERE id = ?");
    $delete_user->execute([$delete_id]);

    $delete_orders = $conn->prepare("DELETE FROM `orders` WHERE user_id = ?");
    $delete_orders->execute([$delete_id]);
    $delete_messages = $conn->prepare("DELETE FROM `messages` WHERE user_id = ?");
    $delete_messages->execute([$delete_id]);
    $delete_cart = $conn->prepare("DELETE FROM `cart` WHERE user_id = ?");
    $delete_cart->execute([$delete_id]);

    $delete_wishlist = $conn->prepare("DELETE FROM `wishlist` WHERE user_id = ?");
    $delete_wishlist->execute([$delete_id]);

    header('location:users_accounts.php');
}
<?php
$select_accounts = $conn->prepare("SELECT * FROM `users`");
$select_accounts->execute();

if($select_accounts->rowCount() > 0){
    while($fetch_accounts = $select_accounts->fetch(PDO::FETCH_ASSOC)){
        ?>
        <div class="box">
            <p> user id : <span><?= $fetch_accounts['id']; ?></span> </p>
            <p> username : <span><?= $fetch_accounts['name']; ?></span> </p>
            <p> email : <span><?= $fetch_accounts['email']; ?></span> </p>
            <a href="users_accounts.php?delete=<?= $fetch_accounts['id']; ?>" onclick="return
confirm('delete this account? the user related information will also be delete!')"
class="delete-btn">delete</a>
        </div>
    <?php
    }
    }else{
        echo '<p class="empty">no accounts available!</p>';
    }
}
```

Εικόνα 36: κώδικας κινήσεων διαχειριστών προς τους χρήστες

3.10.2 Λογαριασμοί Διαχειριστών

Σε αυτήν την περίπτωση είναι πιο απλά τα πράγματα καθώς ο διαχειριστής μπορεί να κάνει προετοιμασία και εκτέλεση εντολής διαγραφής διαχειριστή από την βάση δεδομένων. Στη συνέχεια, εκτελούμε ερωτήματα για ανάκτηση όλων των λογαριασμών των διαχειριστών από τη βάση δεδομένων, μετά κάνουμε έλεγχο αν υπάρχουν οι λογαριασμοί και φτιάχνουμε και έναν σύνδεσμο για διαγραφή του λογαριασμού μέσω javascript. Τέλος, εάν ο λογαριασμός είναι του συνδεδεμένου χρήστη, εμφανίζουμε την επιλογή ενημέρωσης.

```
if(isset($_GET['delete'])){
    $delete_id = $_GET['delete'];

    $delete_admins = $conn->prepare("DELETE FROM `admins` WHERE id = ?");
    $delete_admins->execute([$delete_id]);

    header('location:admin_accounts.php');
}

<?php
    $select_accounts = $conn->prepare("SELECT * FROM `admins`");
    $select_accounts->execute();

    if($select_accounts->rowCount() > 0){
        while($fetch_accounts = $select_accounts->fetch(PDO::FETCH_ASSOC)){
            ?>
            <div class="box">
                <p> admin id : <span><?= $fetch_accounts['id']; ?></span> </p>
                <p> admin name : <span><?= $fetch_accounts['name']; ?></span> </p>
                <div class="flex-btn">
                    <a href="admin_accounts.php?delete=<?= $fetch_accounts['id']; ?>" onclick="return confirm('delete this account?')" class="delete-btn">delete</a>

                    <?php
                        if($fetch_accounts['id'] == $admin_id){
                            echo '<a href="update_profile.php" class="option-btn">update</a>';
                        }
                    ?>
                </div>
            </div>
```

Εικόνα 37: κώδικας κινήσεων από τους διαχειριστές προς τους διαχειριστές

3.11 Ενημέρωση Προϊόντων - Κεφαλίδα Χρήστη

3.11.1 Ενημέρωση των Προϊόντων

Σε αυτήν την σελίδα δημιούργησα έναν κώδικα όπου μπορώ να ενημερώνω το παλιό μου προϊόν και να αλλάξω τις εικόνες, τις πληροφορίες και οτιδήποτε γύρω από αυτό. Αρχικά,

κάνουμε ανάκτηση δεδομένων από την φόρμα, ενημερώνουμε το προϊόν στη βάση δεδομένων, ανάκτηση των παλαιών ονομάτων των εικόνων, ανάκτηση νέων ονομάτων και αποθήκευση των νέων εικόνων, φιλτράρισμα των ονομάτων των νέων εικόνων, ανάκτηση των μεγεθών των νέων εικόνων και ανάκτηση των προσωρινών ονομάτων των νέων εικόνων. Στη συνέχεια, όταν επιλέγουμε μια εικόνα για ενημέρωση, ενημερώνουμε το όνομα της εικόνας στη βάση δεδομένων, αποθηκεύουμε την εικόνα και διαγράφουμε την παλιά εικόνα από το φάκελο αποθήκευσης. Τέλος, φτιάχνουμε και μια φόρμα που περιέχουν τα πεδία εισαγωγής για την ενημέρωση των πληροφοριών του προϊόντος και για την ενημέρωση των εικόνων του προϊόντος.

```
$pid = $_POST['pid'];
$name = $_POST['name'];
$name = filter_var($name, FILTER_SANITIZE_STRING);
$price = $_POST['price'];
$price = filter_var($price, FILTER_SANITIZE_STRING);
$details = $_POST['details'];
$details = filter_var($details, FILTER_SANITIZE_STRING);

$update_product = $conn->prepare("UPDATE `products` SET name = ?, price = ?, details
= ? WHERE id = ?");
$update_product->execute([$name, $price, $details, $pid]);

$message[] = 'product updated successfully!';

$sold_image_01 = $_POST['old_image_01'];
$sold_image_02 = $_POST['old_image_02'];
$sold_image_03 = $_POST['old_image_03'];

$image_01 = $_FILES['image_01']['name'];
$image_02 = $_FILES['image_02']['name'];
$image_03 = $_FILES['image_03']['name'];

$image_01 = filter_var($image_01, FILTER_SANITIZE_STRING);
$image_02 = filter_var($image_02, FILTER_SANITIZE_STRING);
$image_03 = filter_var($image_03, FILTER_SANITIZE_STRING);

$image_size_01 = $_FILES['image_01']['size'];
$image_size_02 = $_FILES['image_02']['size'];
$image_size_03 = $_FILES['image_03']['size'];

$image_tmp_name_01 = $_FILES['image_01']['tmp_name'];
$image_tmp_name_02 = $_FILES['image_02']['tmp_name'];
$image_tmp_name_03 = $_FILES['image_03']['tmp_name'];

if(!empty($image_01)){
    if($image_size_01 > 2000000){
        $message[] = 'image size is too large!';
    }else{
        $update_image_01 = $conn->prepare("UPDATE `products` SET image_01 = ? WHERE id
= ?");
        $update_image_01->execute([$image_01, $pid]);
        move_uploaded_file($image_tmp_name_01, $image_folder_01);
        unlink('../uploaded_img/'.$sold_image_01);
        $message[] = 'image 01 updated successfully!';
    }
}
```

```

    }
    <span>update name</span>
    <input type="text" name="name" required class="box" maxlength="100" place-
holder="enter product name" value="<?= $fetch_products['name']; ?>">
    <span>update price</span>
    <input type="number" name="price" required class="box" min="0" max="9999999999"
placeholder="enter product price" onkeypress="if(this.value.length == 10) return false;"
value="<?= $fetch_products['price']; ?>">
    <span>update details</span>
    <textarea name="details" class="box" required cols="30" rows="10"><?= $fetch_prod-
ucts['details']; ?></textarea>
    <span>update image 01</span>
    <input type="file" name="image_01" accept="image/jpg, image/jpeg, image/png, im-
age/webp" class="box">
    <span>update image 02</span>
    <input type="file" name="image_02" accept="image/jpg, image/jpeg, image/png, im-
age/webp" class="box">
    <span>update image 03</span>
    <input type="file" name="image_03" accept="image/jpg, image/jpeg, image/png, im-
age/webp" class="box">
    <div class="flex-btn">
    <input type="submit" name="update" class="btn" value="update">
    <a href="products.php" class="option-btn">go back</a>

```

Εικόνα 38: κώδικας ενημέρωσης προϊόντων

Εικόνα 39: σελίδα ενημέρωσης προϊόντων

3.11.2 Κεφαλίδα Χρήστη

Στο πάνω μέρος του eshop μου έχω δημιουργήσει πλοήγηση με συνδέσμους προς τις σελίδες που έχω τοποθετήσει, όσον αφορά τον php κώδικα, ελέγχουμε τον αριθμό των

προϊόντων στη λίστα επιθυμιών και στο καλάθι αγορών. Επιπλέον, τοποθετούμε και ένα εικονίδιο της λίστας επιθυμιών και του καλάθι αγορών όπου θα δείχνουν το καθένα τον αριθμό των προϊόντων. Τέλος, επιλέγουμε τα στοιχεία του χρήστη με βάση το id του και αν αυτά υπάρχουν να εμφανίζονται τα στοιχεία του προφίλ και οι επιλογές, μετά να εμφανίζεται το όνομα του χρήστη και επιλογή για ενημέρωση του προφίλ και εμφανίζονται επιλογές για εγγραφή, σύνδεση και αποσύνδεση.

```
<nav class="navbar">
    <a href="home.php">home</a>
    <a href="about.php">about</a>
    <a href="orders.php">orders</a>
    <a href="shop.php">shop</a>
    <a href="contact.php">contact</a>
</nav>
<?php
    $count_wishlist_items = $conn->prepare("SELECT * FROM `wishlist` WHERE user_id = ?");
    $count_wishlist_items->execute([$user_id]);
    $total_wishlist_counts = $count_wishlist_items->rowCount();

    $count_cart_items = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
    $count_cart_items->execute([$user_id]);
    $total_cart_counts = $count_cart_items->rowCount();
    ?>
<div id="menu-btn" class="fas fa-bars"></div>
    <a href="search_page.php"><i class="fas fa-search"></i></a>
    <a href="wishlist.php"><i class="fas fa-heart"></i><span>(= $total_wish-
list_counts; ?&gt;)&lt;/span&gt;&lt;/a&gt;
    &lt;a href="cart.php"&gt;&lt;i class="fas fa-shopping-cart"&gt;&lt;/i&gt;&lt;span&gt;(<?= $to-
tal_cart_counts; ?&gt;)&lt;/span&gt;&lt;/a&gt;
    &lt;div id="user-btn" class="fas fa-user"&gt;&lt;/div&gt;
&lt;?php
    $select_profile = $conn-&gt;prepare("SELECT * FROM `users` WHERE id = ?");
    $select_profile-&gt;execute([$user_id]);

    if($select_profile-&gt;rowCount() &gt; 0){
        $fetch_profile = $select_profile-&gt;fetch(PDO::FETCH_ASSOC);
        ?&gt;
        &lt;p&gt;<?= $fetch_profile["name"]; ?&gt;&lt;/p&gt;
        &lt;a href="update_user.php" class="btn"&gt;update profile&lt;/a&gt;

        &lt;div class="flex-btn"&gt;
            &lt;a href="user_register.php" class="option-btn"&gt;register&lt;/a&gt;
            &lt;a href="user_login.php" class="option-btn"&gt;login&lt;/a&gt;
        &lt;/div&gt;

        &lt;a href="components/user_logout.php" class="delete-btn" onClick="return con-
firm('logout from the website?');"&gt;logout&lt;/a&gt;</pre

```

Εικόνα 40: κώδικας επικεφαλίδας χρήστη

IShop.

home about orders shop contact

🔍 ❤️(1) 🛒(0) 👤

Εικόνα 41: επικεφαλίδα χρήστη

3.12 Υποσέλιδο, Πληροφορίες και Αξιολογήσεις

3.12.1 Υποσέλιδο σελίδας χρήστη

Σε αυτό το κομμάτι της ιστοσελίδας τοποθετήσαμε κάποιους συνδέσμους όπως συνδέσμους με τους οποίους μπορούμε να ανακατευθυνόμαστε πιο γρήγορα για αγορά προϊόντων και για να συνομιλήσουμε με τον διαχειριστή. Επιπλέον, έχουμε και άλλους συνδέσμους όπου υπάρχει η εγγραφή και η είσοδος του χρήστη. Επίσης, έχω τοποθετήσει και πληροφορίες επικοινωνίας όπως τηλέφωνα, ηλεκτρονικό ταχυδρομείο και τοποθεσία. Στη συνέχεια, έχω βάλει και συνδέσμους κοινωνικών δικτύων όπως Facebook και Instagram. Τέλος, έχω προσθέσει και ένα κείμενο πιστωτικών πληροφοριών.

```
<div class="box">
  <h3>quick links</h3>
  <a href="home.php"> <i class="fas fa-angle-right"></i> home</a>
  <a href="about.php"> <i class="fas fa-angle-right"></i> about</a>
  <a href="shop.php"> <i class="fas fa-angle-right"></i> shop</a>
  <a href="contact.php"> <i class="fas fa-angle-right"></i> contact</a>
</div>

<div class="box">
  <h3>contact us</h3>
  <a href="tel:241067452"><i class="fas fa-phone"></i> 241067452</a>
  <a href="tel:241045321"><i class="fas fa-phone"></i> 241045321</a>
  <a href="mailto:frankbeleri99@gmail.com"><i class="fas fa-envelope"></i> frank-
beleri99@gmail.com</a>
  <a href="https://www.google.com/myplace"><i class="fas fa-map-marker-alt"></i>
Larissa, Greece </a>
</div>

<div class="credit">&copy; copyright @ <?= date('Y'); ?> by <span>Francesko Be-
jleri</span> | all rights reserved!</div>
```

Εικόνα 42: κώδικας υποσέλιδου του χρήστη

Quick Links	Extra Links	Contact Us	Follow Us
> home	> login	☎ 241067452	f facebook
> about	> register	☎ 241045321	t twitter
> shop	> cart	✉ frankbeleri99@gmail.com	@ instagram
> contact	> orders	📍 Larissa, Greece	in linkedin
			Ευρωπαϊκή Μετάβαση στις Windows.

© copyright @ 2024 by Francesco Bejleri | all rights reserved!

Εικόνα 43: υποσέλιδο του χρήστη

3.12.2 Πληροφορίες και αξιολογήσεις του eshop

Σε αυτήν την σελίδα έχω προσθέσει ένα κείμενο που εξηγεί για ποιόν λόγο πρέπει ο πελάτης να επιλέξει το ηλεκτρονικό κατάστημά μας καθώς και διάφορες αξιολογήσεις που έχουν γίνει από διάφορους πελάτες όπου δίνουν την άποψη τους για το προϊόντα, για την ιστοσελίδα γενικότερα και για την εξυπηρέτηση. Τέλος, έχω τοποθετήσει και ένα δρομέα ώστε να μπορούμε να μετακινούμε τις αξιολογήσεις των χρηστών.

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

<div class="content">
  <h3>why choose us?</h3>
  <p>Discover why IShop is your ultimate destination for all things electronic. With a handpicked selection of top-tier products and a knowledgeable staff ready to assist, we ensure quality and expertise with every purchase. Our extensive inventory caters to all preferences, offering competitive pricing without sacrificing quality. Plus, our dedicated customer support team is here to assist with any inquiries, making your shopping experience seamless and stress-free. Join our community of satisfied customers and experience the convenience of our streamlined ordering process. Choose IShop today and elevate your electronic experience to new heights!</p>
  <a href="contact.php" class="btn">contact us</a>
</div>
<div class="swiper-wrapper">
  <div class="swiper-slide slide">
    
    <p>IShop stands out as an exceptional e-commerce platform for electronic gadgets. Their extensive selection, intuitive website, and prompt customer service make shopping a breeze. From browsing to checkout, every step is seamless, ensuring a hassle-free experience. With speedy delivery and reliable support, IShop exceeds expectations. Whether you're a tech enthusiast or casual shopper, IShop earns top marks for their outstanding service and product offerings.</p>
    <div class="stars">
      <i class="fas fa-star"></i>
      <i class="fas fa-star"></i>
      <i class="fas fa-star"></i>
      <i class="fas fa-star"></i>
      <i class="fas fa-star-half-alt"></i>
    </div>
    <h3>Giannis Papadopoulos</h3>
  </div>

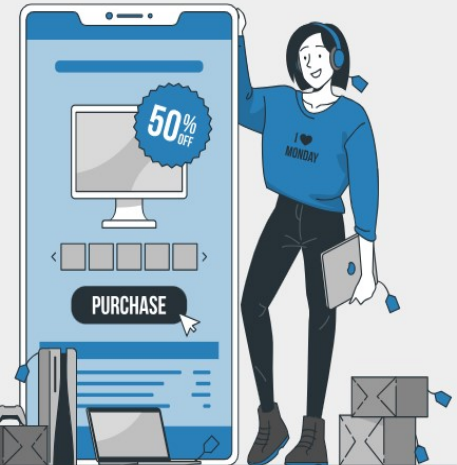
  var swiper = new Swiper(".reviews-slider", {
    loop:true,
    spaceBetween: 20,
    pagination: {
      el: ".swiper-pagination",
      clickable:true,
    },
    breakpoints: {
      0: {
        slidesPerView:1,
      },
      768: {
        slidesPerView: 2,
      },
    },
  });
```

```

991: {
    slidesPerView: 3,
  },
},
});

```

Εικόνα 44: κώδικας πληροφοριών και αξιολογήσεων του χρήστη



why choose us?

Discover why IShop is your ultimate destination for all things electronic. With a handpicked selection of top-tier products and a knowledgeable staff ready to assist, we ensure quality and expertise with every purchase. Our extensive inventory caters to all preferences, offering competitive pricing without sacrificing quality. Plus, our dedicated customer support team is here to assist with any inquiries, making your shopping experience seamless and stress-free. Join our community of satisfied customers and experience the convenience of our streamlined ordering process. Choose IShop today and elevate your electronic experience to new heights!

Contact Us

Ενεργοποιήστε τα Windows
Μετάβαση στις ρυθμίσεις για ενεργοποίηση Windows.

CLIENT'S REVIEWS



IShop stands out as an exceptional e-commerce platform for electronic gadgets. Their extensive selection, intuitive website, and prompt customer service make shopping a breeze. From browsing to checkout, every step is seamless, ensuring a hassle-free experience. With speedy delivery and reliable support, IShop exceeds expectations. Whether you're a tech enthusiast or casual shopper, IShop earns top marks for their outstanding service and product offerings.

★★★★★



IShop impresses with its extensive selection of electronic products, competitive pricing, and user-friendly interface. Navigating the site is a breeze, and their responsive customer service adds to the positive shopping experience. While some product descriptions could be more detailed, overall, Ishop is a reliable destination for electronic enthusiasts looking for great deals and a hassle-free shopping experience.

★★★★★

Anita May



While Ishop offers a decent range of electronic products, the overall shopping experience leaves something to be desired. Navigation can be clunky at times, with slow loading pages and occasional glitches. Pricing is competitive, but the lack of detailed product descriptions makes it difficult to make informed purchasing decisions. Customer service responses have been hit or miss, with some inquiries going unanswered for extended periods. While Ishop has potential, it falls short of providing a consistently satisfactory online shopping experience.

Ενεργοποιήστε τα Windows
Μετάβαση στις ρυθμίσεις για ενεργοποίηση Windows.

Εικόνα 45: σελίδα πληροφοριών και αξιολογήσεων του χρήστη

4 Η Δυσκολίες του Κώδικα

Σε αυτό την ενότητα θα αναφερθούμε κυρίως για τα κομμάτια του κώδικα στα οποία υπήρχε μια ιδιαίτερη δυσκολία για να υλοποιηθεί σωστά ώστε ο πελάτης να μην αντιμετωπίζει προβλήματα κατά τις αγορές των προϊόντων.

4.1 Καλάθι αγορών και λίστα επιθυμιών

Το κυρίως πράγμα το οποίο με δυσκόλεψε ήταν να μπορώ να κάνω τον κώδικα ασφαλή για να μπορώ να αποφεύγω SQL Injections οπότε σε αυτήν την περίπτωση έκανα απολύμανση εισόδων χρησιμοποιώντας `'filter_var'`, το `'$_POST['pid']'` και το `'$_POST['name']'` όπως επίσης έκανα έλεγχο ύπαρξης προϊόντων με το (`'$conn->prepare'`). Στη συνέχεια, έκανα και έλεγχο αν ο χρήστης είναι συνδεδεμένος με το (`'user_id==''` ') ώστε να αποφύγω μη εξουσιοδοτημένες ενέργειες. Επιπλέον, ένα άλλο δύσκολο κομμάτι ήταν ο συγχρονισμός και η ακεραιότητα των δεδομένων δηλαδή να κάνουμε έλεγχο αν υπάρχει προϊόν στο καλάθι αγορών ή στην λίστα επιθυμιών ώστε να αποφεύγουμε διπλές εγγραφές και να διατηρηθεί η ακεραιότητα των δεδομένων. Τώρα, όσον αφορά στο κυρίως κομμάτι του κώδικα του καλαθιού ένα δύσκολο σημείο ήταν να μπορεί ο πελάτης να διαγράψει ένα προϊόν οπότε σε αυτήν την περίπτωση χρησιμοποιούμε το `'$_POST['cart_id']'` ενώ για να διαγραφούν όλα τα προϊόντα με το `user_id`. Τέλος, για το κομμάτι που περιέχει το λίστα επιθυμιών έχει σχεδόν τα ίδια προβλήματα με το καλάθι απλά όσον αφορά τα μηνύματα χρησιμοποιούμε μηνύματα ανατροφοδότησης για κάθε ενέργεια του χρήστη και για την αποφυγή διπλών αιτημάτων χρησιμοποίησα το `'return confirm('delete all from wishlist?');'` για να επιβεβαιώσει την ενέργεια διαγραφής όλων των στοιχείων

4.2 Τοποθέτηση προϊόντων από τον διαχειριστή

Σε αυτή τη σελίδα του κώδικα είχα δυσκολία στο να γίνει η σωστή τοποθέτηση των προϊόντων ώστε να φανούν στην κεντρική σελίδα του eshop. Πρώτα, για να μπορέσω να προστατέψω τον κώδικα από SQL Injection πρόσθεσα προετοιμασμένες δηλώσεις όπως και να ελέγξω ότι ο διαχειριστής είναι συνδεδεμένος. Ένα άλλο δύσκολο σημείο ήταν να διαχειριστώ την μεταφόρτωση των εικόνων του προϊόντος, ώστε να ελέγξουμε το μέγεθος και να καθαρίζουμε τα ονόματα των αρχείων και για να μπορώ να μετακινήσω τα αρχεία χρησιμοποιήσα την `'move_upload_file'` προς το `'upload_image'`. Ένα άλλο πρόβλημα που αντιμετώπισα είναι το μέγεθος της εικόνας, το οποίο έβγαине πολύ μεγάλο, οπότε για να το αποφύγω αυτό τοποθέτησα έναν κώδικα που ελέγχει αν η εικόνα είναι πολύ μεγάλη και έτσι βγάζει ένα μήνυμα ότι υπάρχει σφάλμα. Τέλος, υπήρχε και μια είδους δυσκολίας ως προς την διαγραφή των προϊόντων καθώς εκτός από το id τους που πρέπει να χρησιμοποιήσω για να το διαγράψω θα έπρεπε να ψάξω και έναν τρόπο για να διαγράψω επίσης και τις εικόνες όποτε χρησιμοποιώ το `'fetch_delete_image'` και unlink από το φάκελο που τα έχω τοποθετήσει για να διαγραφούν τελείως.

4.3 Ενημέρωση των προϊόντων από τον διαχειριστή

Σε αυτό το κομμάτι του κώδικα υπήρξε περισσότερο δυσκολία στο πως θα ενημερώνω τις παλιές εικόνες, να τις διαγράψω και να τοποθετήσω τις νέες. Αρχικά, δημιούργησα την φόρμα ενημέρωσης του προϊόντος για να μπορέσω να κάνω την διαδικασία, ενημερώνω τις βασικές πληροφορίες του προϊόντος στη βάση δεδομένων και χρησιμοποιώ και το `'prepare'` για SQL Injection. Στη συνέχεια, ελέγχω και εδώ αν η εικόνα ξεπερνάει τα 2MB, ενημερώνω την βάση δεδομένων, μετακινώ τα αρχεία στο φάκελο που έχουμε τις εικόνες και διαγράφουμε τις παλιές εικόνες για την αποφυγή άχρηστων δεδομένων στον διακομιστή. Τέλος, ανακτούμε τις πληροφορίες του προϊόντος που πρόκειται να ενημερωθεί από την βάση δεδομένων και τις εμφανίζει στην φόρμα για επεξεργασία, όπως επίσης χρησιμοποιώ προετοιμασμένη δήλωση και `'filter_var'` για την ασφάλεια.

4.4 Δυσκολίες στην πληρωμή των προϊόντων

Γενικότερα, μία από τις δυσκολίες που προέκυψαν σε αυτό το σημείο ήταν να γίνει ένας συνολικός υπολογισμός διάφορων προϊόντων που θα τοποθετήσουμε στο καλάθι αγορών

όποτε με διάφορους τρόπους κατάφερα να το υλοποιήσω, όμως υπήρξε και μια επιπλέον δυσκολία στο γίνει σωστά ο υπολογισμός και έτσι τελικά έκανα ‘fetch’ το καλάθι μαζί με την ποσότητα του προϊόντος ώστε να μου βγάλει το συνολικό κόστος. Πριν από αυτά, επίσης φίλτραρα όνομα, αριθμό τηλεφώνου, email, διεύθυνση και μέθοδο πληρωμής για να είναι ασφαλή από κακόβουλο κώδικα. Στη συνέχεια, επειδή υπήρχε μια άλλη δυσκολία για την σωστή διαχείριση συναλλαγών στη βάση δεδομένων ώστε να αντιμετωπιστούν σωστά, έλεγχα αν υπήρχαν αντικείμενα στο καλάθι του χρήστη, ώστε να εισαχθεί στην παραγγελία του χρήστη και να καθαρίζει το καλάθι. Τέλος, δημιουργούμε και μια φόρμα για να μπορούν να περαστούν τα στοιχεία του πελάτη.

4.5 Δυσκολίες στην γρήγορη προβολή & αναζήτηση

Σε αυτήν την περίπτωση οι δυσκολίες που υπήρξαν ήταν να μπορέσω από σελίδα όπου ο πελάτης αγοράζει να δημιουργεί μια νέα σελίδα που να έχει τις λεπτομέρειες του προϊόντος. Αρχικά, για να μπορεί να φαίνεται το προϊόν το επιλέγαμε από την βάση δεδομένων, βάζουμε ‘prepare’ για να αποφύγουμε SQL Injection. Επίσης, ένα θέμα που υπήρχε είναι ότι δεν μου εμφανιζόντουσαν τα προϊόντα παρόλο που τα πέρασα όμως βρήκα την λύση τοποθετώντας κρυφά πεδία για το κωδικό του προϊόντος, το όνομα, η τιμή και οι εικόνες. Τέλος, δημιούργω μια φόρμα για να εμφανιστεί το προϊόν όμως δεν υπήρχε ασφάλεια οπότε ο οποιοσδήποτε θα μπορούσε να βάλει κακόβουλο κώδικα και να καταστραφεί, άρα κάνουμε fetch για να έχουμε ασφάλεια. Στην συνέχεια, όσον αφορά την αναζήτηση εκτός από την φόρμα που δημιουργούμε για να κάνουμε αναζήτηση με δυσκόλεψε το σημείο ώστε να γίνει η αναζήτηση με το όνομα του προϊόντος, μετά αφού βρήκα την λύση δεν λειτουργούσε με παραπλήσιο όνομα χωρίς δηλαδή να βάλω όλο το όνομα. Τελικά, κατάφερα να λύσω και αυτό το πρόβλημα τοποθετώντας το ‘LIKE’ ώστε αν γράψω όλο ή το παραπλήσιο όνομα να βρίσκει κατευθείαν το προϊόν.

4.6 Δυσκολίες στην ενημέρωση προφίλ

Σε αυτήν την τροποποίηση του ονόματος και του κωδικού δυσκολεύτηκα μέχρι να βρω τον τρόπο ώστε να λειτουργεί σωστά. Αρχικά, για να μπορεί να γίνει η αλλαγή χρειάστηκε να κάνω ενημέρωση της βάσης δεδομένων ώστε από τον παλιό κωδικό να μπορώ να εισχωρήσω καινούργιο, όμως η ασφάλεια σε αυτήν την περίπτωση δεν ήταν καλή οπότε φίλτραρα και κρυπτογράφησα τις εισόδους με ‘filter_var’ και ‘sha1’ αντίστοιχα

για αποφυγή SQL Injection και XSS. Επιπλέον, ένα άλλο πρόβλημα που υπήρχε ήταν ότι είχα πολλούς παλιούς κωδικούς οπότε εξασφάλισα ότι ο παλιός κωδικός πρόσβασης να ταιριάζει με το αποθηκευμένο, επίσης εκτός από τον νέο κωδικό έπρεπε να δημιουργήσω και την επιβεβαίωση αυτού και έπρεπε να δείξω ότι ταιριάζουν. Τέλος, δημιουργούμε μια φόρμα ώστε να περαστούν τα νέα στοιχεία και όσον αφορά τη διαφορά του διαχειριστή με τον χρήστη είναι ότι ο χρήστης χρειάζεται το όνομα και το email ενώ ο διαχειριστής μόνο το όνομα.

4.7 Δυσκολίες σε αρχική σελίδα & κατηγορίες χρηστών

Σε αυτήν την περίπτωση που βρισκόμαστε στην αρχική σελίδα υπήρξαν κάποια δύσκολα σημεία. Πρώτα, ένα πρόβλημα που υπήρχε είναι ότι δεν μπορούσα να εμφανίσω τα προϊόντα σε κάθε κατηγορία που άνηκε οπότε για να μπορέσω να το εμφανίσω πήρα από τη βάση δεδομένων τα προϊόντα που είχαν όνομα σαν την κατηγορία που θα ορίσω εγώ και αυτό το έκανα με το 'LIKE' ώστε να μπορέσω να το προσαρμόσω, όμως στην αρχική σελίδα δεν μπορούσα να το εμφανίσω οπότε χρησιμοποίησα το 'href' για να καλέσω την σελίδα της κατηγορίας και μέσα σε αυτό να βάλω το όνομα τις κάθε κατηγορίας, με αυτόν τον τρόπο όλα λειτουργούσαν κανονικά. Επιπλέον, ένα άλλο που με δυσκόλεψε ήταν όταν φαίνονται καινούργια προϊόντα να μην φαίνονται όλα τα προϊόντα που υπάρχουν στο eshop, οπότε κάλεσα από τη βάση δεδομένων τα προϊόντα αλλά χρησιμοποίησα το 'LIMIT 6' για να φαίνονται μόνο τα 6 καινούργια προϊόντα. Τέλος, υπήρξε και μια δυσκολία στο πως να κυλίσουν τα μικρά εικονίδια στις κατηγορίες των προϊόντων όμως κατάφερα και τοποθέτησα έναν 'swiper' μέσα στην κλάση τις κατηγορίας για να μπορέσει να γίνει η κύλιση.

4.8 Δυσκολίες στην εγγραφή και login

Σε αυτό το σημείο του eshop γενικότερα δεν υπήρξαν δυσκολίες, όμως υπήρχαν κάποια μικροπροβλήματα. Αρχικά, το όνομα, το email, ο κωδικός και επαναλαμβανόμενος κωδικός ήταν πολλή ελεύθερα δηλαδή δεν υπήρχε ασφάλεια οπότε πρόσθεσα σε καθένα από αυτά ασφάλεια με το 'filter_var' και 'FILTER_SANITIZE_STRING' για προστασία από κακόβουλο κώδικα αλλά και το 'sha_1' για κρυπτογράφηση των δεδομένων. Στη συνέχεια, ένα άλλο πρόβλημα που υπήρχε είναι ότι παιρνούσαν τα δεδομένα για να γίνει η εγγραφή αλλά δεν γνωρίζαμε αν αυτό το email υπήρχε πιο πριν ή ήταν καινούργιος

οπότε πρόσθεσα και έλεγχο μέσα στη βάση δεδομένων των ‘users’ αν υπάρχει το ‘email’. Επιπλέον, η εγγραφή γινόταν κανονικά και μπορεί ο χρήστης ή ο διαχειριστής στη συνέχεια να κάνει είσοδο. Επίσης, η μόνη διαφορά που υπάρχει στην είσοδο σε σχέση με την εγγραφή του χρήστη ή του διαχειριστή είναι ότι χρησιμοποιούμε μόνο το όνομα και των κωδικό και για το αν υπάρχει ο διαχειριστής ή ο χρήστης στη βάση δεδομένων να ελέγχεται το ‘email’ και ο κωδικός. Τέλος, για καθένα από την εγγραφή και την είσοδο του χρήστη ή του διαχειριστή φτιάχνουμε μια φόρμα.

5 Συμπεράσματα

Σε αυτήν την πτυχιακή μιλήσαμε για το πως εταιρίες eshop κατάφεραν από μικρά και ανοργάνωτα που ήταν, να γίνουν μεγάλα και οργανωμένα πλέον και να αλλάξουν την ιδεολογία των καταναλωτών από τα καταστημάτων του δρόμου να περιηγηθούν στον παγκόσμιο ιστό και να βρίσκουν τα προϊόντα τους φθηνότερα και γρηγορότερα, αυτό το κατάφεραν εταιρίες όπως το skroutz, το public και άλλα.

Στη συνέχεια, μιλήσαμε για την html το πόσο βασικό είναι για να μπορεί να γίνει η δόμηση μιας ιστοσελίδας αλλά και ενός ολόκληρου eshop, την css η οποία βοηθάει στην εμφάνιση τις ιστοσελίδας ώστε να δώσει μια ομοιομορφία μαζί με την html, την javascript η οποία έχει διαδραστική και δυναμική χρήση ως προς κάποιες λειτουργίες των ιστοσελίδων και του eshop και η php η οποία όπως είπαμε καλύπτει το πίσω λειτουργία του eshop και η οποία συνδέοντας με την βάση δεδομένων μπορεί να εισχωρήσει και διαγράψει διάφορα δεδομένα στο eshop.

Επιπρόσθετα, αναφέραμε και την ανάλυση του κώδικα δηλαδή το πως δημιουργήθηκε η είσοδος, έξοδος, εγγραφή του χρήστη ή του διαχειριστεί, το καλάθι αγορών, την λίστα επιθυμιών, την παραγγελία των προϊόντων, τις κατηγορίες των προϊόντων και την εισαγωγή των προϊόντων.

Επιπλέον, όσον αφορά τις δυσκολίες που υπήρχαν το ποιο βασικό ήταν η προστασία και η ασφάλεια του eshop που αρχικά δεν υπήρχε και έτσι ο οποιοσδήποτε μπορούσε να τοποθετήσει κακόβουλο κώδικα , ένα άλλο ήταν και η εισαγωγή των προϊόντων από τον διαχειριστή στο κύριο eshop διότι στην αρχή δεν μπορούσα να βρω τον τρόπο να περάσω τα δεδομένα αλλά στην τελική το κατάφερα.

Συμπερασματικά, αυτή η πτυχιακή με βοήθησε αρκετά να δω τις διαφορετικές πτυχές ενός eshop διότι αρχικά γνώριζα μόνο πολύ βασικά πράγματα όπως μια είσοδο του χρήστη ενώ τώρα κατάφερα να μετακινώ προϊόντα στο καλάθι αγορών για να τα αγοράσω, την λίστα επιθυμιών για οτιδήποτε αρέσει τον πελάτη και την εισαγωγή προϊόντων στο κύριο eshop από την ιστοσελίδα του διαχειριστεί και τελικά να ασχοληθώ ποιο πολύ με αυτό το κομμάτι και να εξελιχθώ πιο πολύ.