



**Πανεπιστήμιο Ιωαννίνων**

Τμήμα Πληροφορικής & Τηλεπικοινωνιών

---

## **Πτυχιακή Εργασία**

---

Ανάπτυξη Διαδικτυακής Εφαρμογής σε γλώσσα PhP  
Υλοποίηση του Ηλεκτρονικού καταστήματος MotoRoda



**ΛΙΣΓΑΡΑΣ ΔΗΜΗΤΡΙΟΣ**

A.M. :1391

---

ΕΠΙΒΛΕΠΩΝ:ΛΙΑΡΟΚΑΠΗΣ ΔΗΜΗΤΡΙΟΣ, ΚΑΘΗΓΗΤΗΣ

## The Development of a Web Application in PHP

**Εγκρίθηκε από τριμελή εξεταστική επιτροπή**

Τόπος, Ημερομηνία

## **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1. Επιβλέπων καθηγητής

Όνομα Επίθετο,

2. Μέλος επιτροπής

Όνομα Επίθετο,

3. Μέλος επιτροπής

Όνομα Επίθετο

© Λισγάρας, Δημήτριος, 2024.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

## Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Λισγάρας, Δημήτριος

Υπογραφή

## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου στην οικογένειά μου, για τη στήριξη και την αγάπη τους καθ' όλη τη διάρκεια αυτής της πορείας. Ιδιαίτερα, ευχαριστώ τον πατέρα μου, Παύλο, για την αμέριστη πίστη και τις πολύτιμες συμβουλές του, καθώς και τη μητέρα μου, Κατερίνα, για την ανιδιοτελή της αγάπη και τις ατελείωτες ώρες ενθάρρυνσης. Τον αδελφό μου, Πέτρο, που στάθηκε πάντα δίπλα μου, ενισχύοντας την πίστη μου στις δυνατότητές μου. Καθώς επίσης και τον καθηγητή μου κ. Λιαροκάπη Δημήτριο, που με τον ενεργό του ρόλο και την καθοδήγησή του, διετέλεσε αξιοσημείωτο παράγοντα στην εκπόνηση αυτής της εργασίας. Τέλος, θα ήθελα να ευχαριστήσω την αγαπημένη μου, Αλεξάνδρα, για την αμέριστη υπομονή, την αγάπη και τη δύναμη που μου έδωσε, κάνοντάς με να νιώθω πάντα ότι δεν είμαι μόνος σε αυτή τη διαδρομή. Ευχαριστώ όλους σας για όλα όσα κάνατε και συνεχίζετε να κάνετε για μένα.

## ΠΕΡΙΛΗΨΗ

Η πτυχιακή εργασία αυτή πραγματεύεται την ανάπτυξη μιας ηλεκτρονικής εφαρμογής για την πώληση μεταχειρισμένων ποδηλάτων, με τίτλο "Ανάπτυξη Διαδικτυακής Εφαρμογής σε γλώσσα PHP". Σκοπός της εφαρμογής είναι να προσφέρει μια ολοκληρωμένη και αξιόπιστη λύση για τις πωλήσεις μεταχειρισμένων ποδηλάτων, προωθώντας την επαναχρησιμοποίηση παλαιών ποδηλάτων και συμβάλλοντας στη βιωσιμότητα. Στην εργασία αναλύονται οι λειτουργικές και μη λειτουργικές απαιτήσεις της εφαρμογής, όπως η δημιουργία λογαριασμών χρηστών, η διαχείριση προϊόντων, το καλάθι αγορών και η διαδικασία παραγγελιών. Επίσης, παρουσιάζονται η αρχιτεκτονική του συστήματος, ο σχεδιασμός της βάσης δεδομένων και η διεπαφή του χρήστη. Η υλοποίηση της εφαρμογής βασίστηκε στις γλώσσες προγραμματισμού PHP, MySQL για τη βάση δεδομένων, HTML και CSS για τη δημιουργία της διεπαφής, καθώς και JavaScript για τη διαδραστικότητα. Τέλος, προτείνονται μελλοντικές επεκτάσεις και βελτιώσεις για την εφαρμογή.

**Λέξεις-κλειδιά:** ηλεκτρονικό εμπόριο, PHP, MySQL, HTML, CSS, JavaScript

## Abstract

This thesis deals with the development of an electronic application for the sale of used bicycles, titled " The Development of a Web Application in PHP." The purpose of the application is to offer a complete and reliable solution for selling used bicycles, promoting reuse of old bicycles and contributing to sustainability. The thesis analyzes the functional and non-functional requirements of the application, such as user account creation, product management, shopping cart, and order processing. Additionally, the system architecture, database design, and user interface are presented. The implementation of the application was based on the PHP programming language, MySQL for the database, HTML and CSS for the interface creation, and JavaScript for interactivity. Finally, future extensions and improvements for the application are proposed.

**Keywords:** e-commerce, PHP, MySQL, HTML, CSS, JavaScript

|                                              |    |
|----------------------------------------------|----|
| Περιεχόμενα                                  |    |
| ΕΥΧΑΡΙΣΤΙΕΣ .....                            | 6  |
| ΠΕΡΙΛΗΨΗ .....                               | 7  |
| Abstract .....                               | 8  |
| Κεφάλαιο 1° .....                            | 11 |
| 1.1 Ορισμός ηλεκτρονικού εμπορίου .....      | 11 |
| 1.2 Πλεονεκτήματα και μειονεκτήματα .....    | 11 |
| 1.3 Τρόποι πληρωμής .....                    | 13 |
| 1.4 Γλώσσα Προγραμματισμού PHP .....         | 15 |
| 1.5 Βάσεις δεδομένων MySQL.....              | 16 |
| 1.6 PhpMyAdmin .....                         | 17 |
| 1.7 Τεχνολογίες Web:HTML,CSS,JavaScript..... | 18 |
| Κεφάλαιο 2°: .....                           | 21 |
| 2.1Λειτουργικές Απαιτήσεις .....             | 21 |
| 2.2 Μη Λειτουργικές Απαιτήσεις .....         | 22 |
| 2.3 Τοπική εγκατάσταση .....                 | 23 |
| 2.4 Online.....                              | 24 |
| Δομή Domain Name .....                       | 24 |
| Διαδικασία Κατοχύρωσης Domain Name .....     | 25 |
| Λειτουργία Domain Name .....                 | 25 |
| Σημασία του Domain Name .....                | 25 |
| Λειτουργία Web Server .....                  | 26 |
| Κύριοι Τύποι Web Servers .....               | 26 |
| Βασικές Λειτουργίες και Χαρακτηριστικά ..... | 27 |
| Κεφάλαιο 3° .....                            | 28 |
| Σχεδιασμός Συστήματος.....                   | 28 |
| 3.1 Αρχιτεκτονική Συστήματος .....           | 28 |
| 3.2 Σχεδιασμός Βάσης Δεδομένων.....          | 29 |
| 3.3 Σχεδιασμός Διεπαφής Χρήστη.....          | 33 |
| 3.4 Διαγράμματα UML.....                     | 35 |
| 3.5 Διαχείριση Έργου και Χρονοδιάγραμμα..... | 39 |

|                                            |    |
|--------------------------------------------|----|
| Κεφάλαιο 4° .....                          | 41 |
| Παρουσίαση Ηλεκτρονικού Καταστήματος ..... | 41 |
| 4.1 Πλοήγηση ως πελάτης.....               | 42 |
| 4.2 Πλοήγηση ως διαχειριστής .....         | 47 |
| Συμπεράσματα .....                         | 52 |
| Αναφορές.....                              | 53 |

# Κεφάλαιο 1<sup>ο</sup>

## 1.1 Ορισμός ηλεκτρονικού εμπορίου

Ηλεκτρονικό εμπόριο είναι ο ορισμός που χρησιμοποιούμε για να χαρακτηρίσουμε μια (συνήθως ανέπαφη) συναλλαγή, που λαμβάνει χώρο ανάμεσα σε δυο οι περισσότερους ανθρώπους. Κάνοντας χρήση των τηλεπικοινωνιακών υποδομών γνωστών άλλωστε και σαν το Διαδίκτυο. Επιτρέποντας την αγοραπωλησία αγαθών και υπηρεσιών χωρίς να είναι απαραίτητη η φυσική παρουσία ούτε των καταναλωτών ούτε των παροχών – προμηθευτών. (Carey, 2001)

Ο στόχος λοιπόν του ηλεκτρονικού εμπορίου είναι να διευκολύνει την αγοραπωλησία προϊόντων και υπηρεσιών μέσω του διαδικτύου. Αυτό περιλαμβάνει την απλοποίηση της διαδικασίας αγοράς για τους καταναλωτές, τη μείωση των δαπανών λειτουργίας για τις επιχειρήσεις και την επέκταση της αγοράς τους πέρα από τα γεωγραφικά όρια. Με τη χρήση ηλεκτρονικών πλατφορμών, είναι δυνατή η προσφορά ευέλικτων και εξατομικευμένων λύσεων, ενισχύοντας την εμπειρία του πελάτη και προωθώντας την καινοτομία. Τελικά, ο στόχος είναι να δημιουργηθεί ένα αποτελεσματικό, αποδοτικό και ανταγωνιστικό περιβάλλον εμπορίου.

Έχει επαναπροσδιορίσει τον τρόπο που οι άνθρωποι αγοράζουν και πωλούν προϊόντα, δημιουργώντας νέες δυνατότητες ανταλλαγής εμπορευμάτων. Αυτό έχει οδηγήσει σε αυξημένη διασυννοριακή εμπορική δραστηριότητα και πρόσβαση σε παγκόσμιες αγορές. Επιπλέον, έχει δημιουργήσει νέες επιχειρηματικές ευκαιρίες και έχει αλλάξει τον τομέα της λιανικής πώλησης. Ωστόσο, έχει επίσης δημιουργήσει προκλήσεις όπως η αύξηση του ανταγωνισμού και οι ανησυχίες σχετικά με την ασφάλεια και την προστασία των καταναλωτών.

## 1.2 Πλεονεκτήματα και μειονεκτήματα

Καθώς η τεχνολογία εξελίσσεται και επηρεάζει τον τρόπο που ζούμε, το ηλεκτρονικό εμπόριο αναδεικνύεται ως ένας ισχυρός παράγοντας, προσφέροντας ευκαιρίες για άνεση, προσαρμοστικότητα και διευρυμένη πρόσβαση σε προϊόντα και υπηρεσίες. Μερικά από τα πλεονεκτήματα που προσφέρονται μέσω αυτού είναι:

- Άνεση και ευκολία: Ο καταναλωτής μπορεί να αγοράσει οποιαδήποτε προϊόν από το σπίτι του, χωρίς να χρειαστεί να μετακινηθεί σε φυσικά καταστήματα.
- Ποικιλία προϊόντων: Οι καταναλωτές έχουν πρόσβαση σε μεγάλη ποικιλία προϊόντων από διάφορους προμηθευτές και κατασκευαστές, αυξάνοντας τις πιθανότητες να βρουν αυτό που ψάχνουν.
- Σύγκριση τιμών και επιλογών: Οι καταναλωτές μπορούν εύκολα να συγκρίνουν τιμές και χαρακτηριστικά προϊόντων από διαφορετικούς πωλητές, βοηθώντας τους να κάνουν πιο ενημερωτικές αγορές.

- Προσφορές και έκπτωση: Πολλές φορές, τα ηλεκτρονικά καταστήματα προσφέρουν ειδικές προσφορές και εκπτώσεις που μπορούν να εξοικονομήσουν χρήματα στους καταναλωτές.
- Άμεση παράδοση: Οι περισσότερες ηλεκτρονικές αγορές προσφέρουν επιλογές για γρήγορη παράδοση, ορισμένες φορές μέσα σε μερικές ώρες ή μέρες, μειώνοντας τον χρόνο αναμονής που απαιτείται με τις παραδοσιακές αγορές.

Από την πλευρά του πωλητή:

- Παγκόσμια αγορά: Μέσω του ηλεκτρονικού εμπορίου, ο πωλητής έχει πρόσβαση σε μια παγκόσμια αγορά, επιτρέποντας του να φτάσει πελάτες σε διάφορες γωνίες του κόσμου.
- Χαμηλότερο κόστος λειτουργίας: Συνήθως, το ηλεκτρονικό εμπόριο απαιτεί λιγότερα έξοδα λειτουργίας από τα φυσικά καταστήματα, όπως μίσθωση καταστημάτων και προσωπικό.
- Αυξημένη ορατότητα: Μέσω των διαδικτυακών διαφημίσεων και των μέσων κοινωνικής δικτύωσής, ο πωλητής μπορεί να αυξήσει την ορατότητα του και να φτάσει σε νέους πελάτες.
- Ανάλυση δεδομένων: Το ηλεκτρονικό εμπόριο παρέχει πολλές δυνατότητες ανάλυσης δεδομένων, επιτρέποντας στον πωλητή να κατανοήσει καλύτερα τις προτιμήσεις και τις συνήθειες των πελατών του.
- Ευελιξία και ευκολία στη διαχείριση του αποθέματος: Μέσω συστημάτων διαχείρισης αποθέματος και αυτοματοποιημένων διαδικασιών, ο πωλητής μπορεί να διαχειριστεί αποτελεσματικά το απόθεμά του.

Κατά τη διάρκεια της ψηφιακής εποχής, το ηλεκτρονικό εμπόριο έχει αναδειχθεί σε ισχυρό εργαλείο για επιχειρήσεις και καταναλωτές. Ωστόσο, πάρα τα πλεονεκτήματά του, υπάρχουν ορισμένα μειονεκτήματα που αξίζει να αναφερθούν.

- Ανάγκη για προσωπική επαφή: Σε ορισμένες αγορές, οι καταναλωτές επιθυμούν τη φυσική επαφή με το προϊόν πριν την αγορά του, κάτι που δεν είναι πάντα δυνατό με το ηλεκτρονικό εμπόριο.
- Καθυστερημένη παράδοση ή απώλεια αποστολών: Σε ορισμένες περιπτώσεις, οι παραδόσεις μπορεί να καθυστερήσουν ή ακόμα και να χαθούν, προκαλώντας απογοήτευση στους καταναλωτές.
- Έλλειψη προσωπικής εξυπηρέτησης: Σε περίπτωση προβλημάτων ή αποριών, οι καταναλωτές μπορεί να αισθανθούν ότι λείπει η προσωπική εξυπηρέτηση που παρέχεται σε παραδοσιακά καταστήματα.

- **Ασφάλεια:** Ο κίνδυνος της κλοπής προσωπικών δεδομένων των καταναλωτών, όπως πιστωτικών καρτών και προσωπικών πληροφοριών, από κακόβουλους χρήστες του διαδικτύου.
- **Αμφισβητήσιμη ποιότητα:** Έλλειψη εγγυήσεων ως προς την ποιότητα και την ακρίβεια των προϊόντων που αγοράζονται online. Ορισμένες φορές οι καταναλωτές μπορεί να λάβουν ένα προϊόν που δεν ανταποκρίνεται στις προσδοκίες τους, χωρίς να έχουν τη δυνατότητα να το ελέγξουν από κοντά πριν την αγορά.

Αναλόγως από την πλευρά του πωλητή

- **Ανταγωνισμός:** Λόγω της ευρείας χρήσης του διαδικτύου, ο ανταγωνισμός είναι έντονος, κάτι που μπορεί να δυσκολέψει τους πωλητές να ξεχωρίσουν.
- **Ανάγκη για εξειδίκευση:** Οι πωλητές πρέπει να έχουν εξειδικευμένες γνώσεις στο διαδίκτυο και στο ψηφιακό μάρκετινγκ για να επιτύχουν στο ηλεκτρονικό εμπόριο.
- **Ασφάλεια και προστασία δεδομένων:** Η ανησυχία για την ασφάλεια των συναλλαγών και την προστασία των δεδομένων των πελατών είναι μείον, καθώς μπορεί να αποθαρρύνει ορισμένους πελάτες από την αγορά.
- **Δυσκολία στην προσωπική επαφή:** Οι πωλητές μπορεί να δυσκολευτούν να προσφέρουν την ίδια επίπεδη προσωπική επαφή και εξυπηρέτηση που προσφέρουν τα παραδοσιακά καταστήματα.
- **Κόστος αποστολής:** Σε περιπτώσεις διεθνών αποστολών ή μεγάλων και βαριών αντικειμένων. Αυτό μπορεί να μειώσει τον ανταγωνισμό του πωλητή σε σχέση με τα τοπικά φυσικά καταστήματα και να επηρεάσει το τελικό κόστος που επιβαρύνει τον καταναλωτή. (Laudon & Guercio Traver, 2023)

### 1.3 Τρόποι πληρωμής

Στην εποχή της ψηφιακής επανάστασης, οι διαδικτυακές αγορές έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητας μας. Ένας κρίσιμος παράγοντας για την επιτυχία και την ασφάλεια των συναλλαγών στο ηλεκτρονικό εμπόριο είναι οι τρόποι πληρωμής που προφέρονται στους καταναλωτές. Οι μέθοδοι αυτές έχουν εξελιχθεί σημαντικά, προφέροντας μεγαλύτερη ευκολία και ασφάλεια.

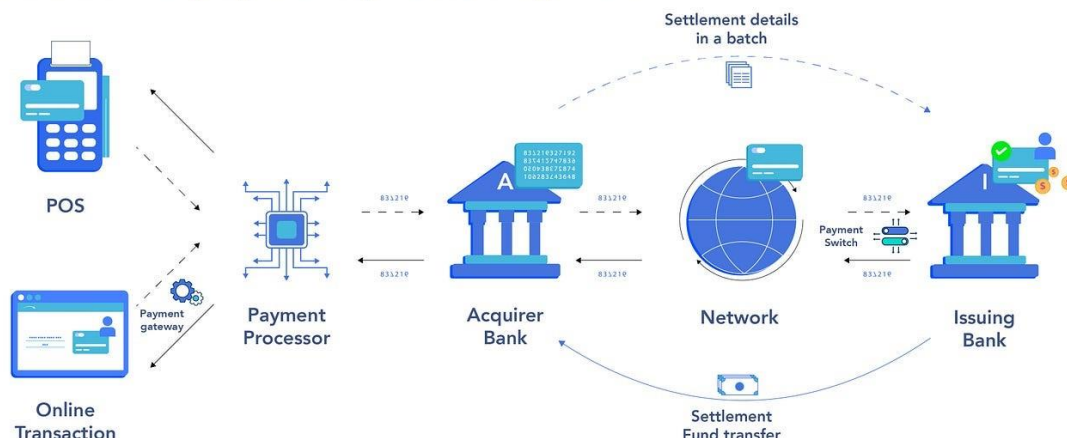
Από τους πιο διαδεδομένους τρόπους πληρωμής στο ηλεκτρονικό εμπόριο είναι οι εξής:

- **Πιστωτική ή χρεωστική κάρτα:** Ίσως ο πιο εδραιωμένος τρόπος πληρωμής, είναι σημαντικό να καταλάβουμε τον τρόπο με τον οποίο λειτουργούν για να μπορέσουμε να αναγνωρίσουμε τις δυνατότητες και τις αδυναμίες που έχουν ως τρόπος πληρωμής στο διαδίκτυο. Παρέχεται η δυνατότητα της άμεσης αποπληρωμής, είτε η πληρωμή σε δόσεις αν ο προμηθευτής παρέχει αυτή την

δυνατότητα. Ως επί τον πλείστον λειτουργεί με τον ίδιο τρόπο όπως σ' ένα κατάστημα με φυσική παρουσία. Στο διαδίκτυο όμως εγείρεται μια πολύ σημαντική διαφορά ο πωλητής δεν βλέπει ποτέ την κάρτα ούτε και υπάρχει κάποια υπογραφή. Έτσι για τον πωλητή υπάρχει ο κίνδυνος της αμφισβήτησης της συναλλαγής από την πλευρά του αγοραστή, παρόλο που τα αγαθά μπορεί να έχουν αποσταλεί ή ο χρήστης να έχει κάνει λήψη του ψηφιακού προϊόν. Στην εικόνα 1.1 παρουσιάζεται μια μέθοδος συναλλαγής με κάρτα στο διαδίκτυο.

## Credit card payment processing flow

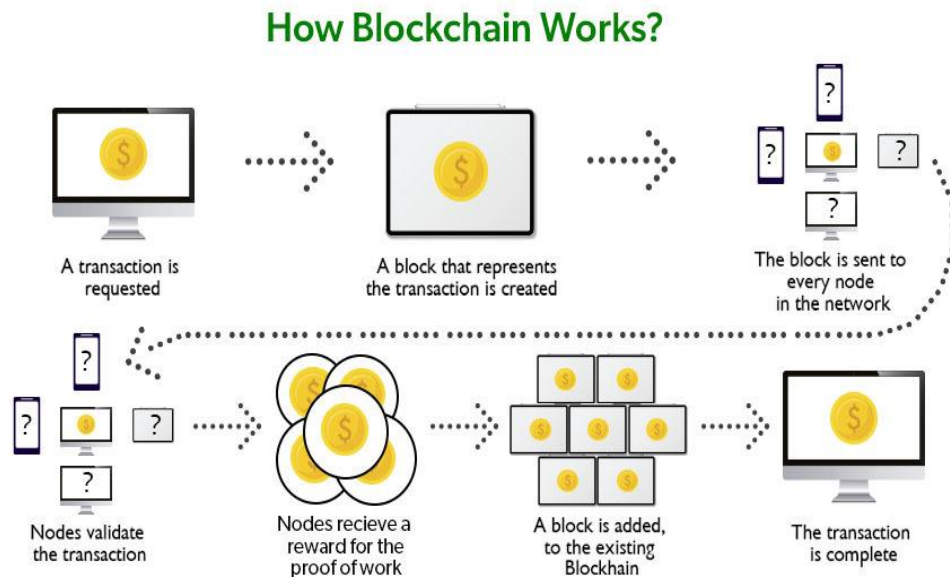
M2P



Εικόνα 1.1 Μεθοδολογία πληρωμής με κάρτα

- Διαδικτυακό σύστημα πληρωμών αποθηκευμένης αξίας: Ένας από τους πρωτοπόρους σε αυτόν τον τρόπο πληρωμής στο διαδίκτυο είναι το PAYPAL. Δημιουργήθηκε το 1998 και στην συνέχεια αποκτήθηκε το 2002 από το πιο αναγνωρισμένο ηλεκτρονικό καταστήματα, το eBay. Έπειτα το 2015 συνέχισε σαν ανεξάρτητη εταιρεία πάλι. Επιτρέπει τόσο στον αγοραστή όσο και στον πωλητή με μία ηλεκτρονική διεύθυνση να κάνουν και να παραλαμβάνουν πληρωμές μέχρι ένα συγκεκριμένο όριο, χρησιμοποιώντας τον τραπεζικό λογαριασμό τους ή την πιστωτική χρεωστική κάρτα τους. (Soni, 2022) Άλλες γνωστές εταιρείες είναι οι Amazon Pay, Dwolla.
- Ηλεκτρονικά πορτοφόλια: Τα τελευταία χρόνια αναπτύσσονται ταχύτατα οι πληρωμές μέσω των smartphone και των δυνατοτήτων που προσφέρουν αυτά. Εταιρείες όπως η Apple, η Samsung, η Google έχουν επενδύσει στην δημιουργία ηλεκτρονικών πορτοφολιών, όπου είναι εφαρμογές για το smartphone παρέχοντας στον χρήστη την δυνατότητα να μπορεί να αποθηκεύσει χρεωστικές κάρτες, κουπόνια επιβράβευσης, τιμολόγια κ.τ.λ.
- Blockchain: Από τις πιο νέες εμφανιζόμενες τεχνολογίες στον κλάδο του ηλεκτρονικού εμπορίου έχει δημιουργήσει την δυνατότητα σε χρήστες, οργανισμούς και εταιρείες να μπορούν να δημιουργήσουν, να εκτελέσουν και να επιβεβαιώσουν εμπορικές συναλλαγές στιγμιαία χωρίς την παρέμβαση μιας

τράπεζας με τον ρόλο της κεντρικής αρχής. Στην εικόνα 1.2 παρουσιάζεται πως ο τρόπος λειτουργίας του blockchain.



Εικόνα 1.2 Τρόπος λειτουργίας Blockchain

## 1.4 Γλώσσα Προγραμματισμού PHP

Η PHP(Hypertext Preprocessor) είναι μια γλώσσα σεναρίων η οποία σχεδιάστηκε κυρίως για την ανάπτυξη δυναμικών ιστοσελίδων και εφαρμογών ιστού. Δημιουργήθηκε το 1994 από τον Rasmus Lerdorf σαν μια συλλογή CGI scripts που σκοπό είχαν να βοηθήσουν στην καταμέτρηση των επισκεπτών στο διαδικτυακό του βιογραφικό. Ο Lerdorf είχε ονομάσει αυτά τα script “Personal Home Page Tools”, έτσι είχε πάρει και το όνομα η γλώσσα. Στην συνέχεια όμως για να συμμορφωθεί με τη σύμβαση ονομάτων GNU(Gnu’s Not Unix) άλλαξε το όνομα της σε αυτό που χρησιμοποιούμε μέχρι και σήμερα, το PHP Hypertext Preprocessor. Η PHP είναι ένα έργο ανοιχτού κώδικα, παρέχει στον χρήστη ελεύθερη πρόσβαση στον πηγαίο κώδικα της και δίνει πρακτικά την ελευθερία στην τροποποίηση αυτού. Σύμφωνα με τον Greg Michillie της Google, το Μάιο του 2015 η PHP χρησιμοποιούταν για τη λειτουργία περισσότερων από το 75%, των site όλου του κόσμου και αυτός ο αριθμός μάλιστα τείνει να αυξάνεται με την πάροδο των χρόνων.

### 1.4.1 Πλεονεκτήματα της PHP

Εύκολη Εκμάθηση και Χρήση:

- Η PHP είναι σχεδιασμένη για να είναι εύκολη στην εκμάθηση και χρήση, ειδικά για αρχάριους προγραμματιστές. Η σύνταξή της είναι παρόμοια με άλλες δημοφιλείς γλώσσες προγραμματισμού όπως η C και η Perl, καθιστώντας την προσιτή για όσους έχουν ήδη εμπειρία στον προγραμματισμό.

#### Διακομιστική Εκτέλεση:

- Η PHP εκτελείται στον διακομιστή, και το αποτέλεσμα της εκτέλεσης αποστέλλεται στον πελάτη ως HTML. Αυτό σημαίνει ότι ο κώδικας PHP δεν είναι ορατός στους τελικούς χρήστες, προσφέροντας ένα επίπεδο ασφάλειας.

#### Ενσωμάτωση με Βάσεις Δεδομένων:

- Η PHP υποστηρίζει τη σύνδεση με πολλές βάσεις δεδομένων, όπως MySQL, PostgreSQL, Oracle και SQLite. Αυτό επιτρέπει την εύκολη διαχείριση και ανάκτηση δεδομένων, καθιστώντας την PHP ιδανική για την ανάπτυξη εφαρμογών που βασίζονται σε βάσεις δεδομένων.

#### Ευελιξία και Επεκτασιμότητα:

- Η PHP είναι εξαιρετικά ευέλικτη και μπορεί να χρησιμοποιηθεί για την ανάπτυξη από μικρές ιστοσελίδες μέχρι μεγάλα, πολύπλοκα συστήματα διαχείρισης περιεχομένου (CMS) όπως το WordPress, το Joomla και το Drupal.

#### Εκτεταμένη Υποστήριξη και Κοινότητα:

- Η PHP έχει μια μεγάλη και ενεργή κοινότητα προγραμματιστών που προσφέρουν υποστήριξη και συνεισφέρουν σε πολλά έργα ανοικτού κώδικα. Υπάρχουν πολλοί διαθέσιμοι πόροι, όπως τεκμηρίωση, φόρουμ και βιβλία που μπορούν να βοηθήσουν τους προγραμματιστές να μάθουν και να βελτιώσουν τις δεξιότητές τους.

#### Υποστήριξη Πλατφορμών:

- Η PHP είναι διαλειτουργική και μπορεί να τρέξει σε πολλές πλατφόρμες, συμπεριλαμβανομένων των Windows, Linux και macOS, καθιστώντας την προσιτή για διαφορετικά περιβάλλοντα ανάπτυξης. (Lockhart, 2015) (Welling & Thomson, 2017)

## 1.5 Βάσεις δεδομένων MySQL

Η MySQL είναι ένα από τα πιο δημοφιλή συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ανοιχτού κώδικα στον κόσμο. Οι ρίζες της ξεκινάνε κάπου το 1979, με το εργαλείο για τις βάσεις δεδομένων UNIREG. Το συγκεκριμένο εργαλείο δημιουργήθηκε από τον Michael "Monty" Widenius για την σουηδική TcX. Το 1996 η MySQL 3.11.1 δημοσιεύτηκε στο κοινό για τις εκδόσεις των Linux και Solaris. Το 2008, η MySQL εξαγοράστηκε από τη Sun Microsystems, η οποία την συνέχισε εξαγοράστηκε από την Oracle Corporation το 2010. Έχει δημιουργηθεί μια μεγάλη κοινότητα γύρω απ' αυτή, γεγονός που καθιστά ευκολότερο την αντιμετώπιση πιθανών προβλημάτων.

### 1.5.1 Πλεονεκτήματα της MySQL:

- Ταχύτητα και απόδοση: Η MySQL είναι γνωστή για την υψηλή της απόδοση και την ικανότητα της να υποστηρίζει μεγάλες βάσεις δεδομένων και υψηλά φορτία εργασίας. Διαθέτει μεγάλη ταχύτητα στην ανάκτηση και την ενημέρωση δεδομένων, καθιστώντας την ιδανική για απαιτητικές εφαρμογές ιστού.
- Υποστήριξη σχεσιακών βάσεων δεδομένων: Η MySQL υποστηρίζει τη χρήση σχεσιακών βάσεων δεδομένων, επιτρέποντας στους χρήστες να οργανώσουν τα δεδομένα τους σε πίνακες και να ορίζουν σχέσεις μεταξύ αυτών. Η SQL (Structured Query Language) χρησιμοποιείται για τη διαχείριση και την ενημέρωση δεδομένων.
- Ασφάλεια: Η MySQL προσφέρει ισχυρά χαρακτηριστικά ασφάλειας, όπως έλεγχο πρόσβασης χρηστών, κρυπτογράφηση δεδομένων και υποστήριξη SSL (Secure Sockets Layer) για ασφαλείς συνδέσεις.
- Συμβατότητα: Η MySQL είναι συμβατή με πολλά λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, Linux, macOS. Επιπλέον συνεργάζεται καλά με διάφορες γλώσσες προγραμματισμού όπως η PHP.
- Κεντρική Διαχείριση Λογικής Επιχειρηματικών Διεργασιών: Η επιχειρηματική λογική μπορεί να διαχειριστεί κεντρικά στον διακομιστή βάσης δεδομένων, εξασφαλίζοντας συνέπεια και ευκολότερη συντήρηση του κώδικα. (DuBois, 2009)

### 1.6 PhpMyAdmin

Η PhpMyAdmin είναι μια εφαρμογή του ιστού γραμμένη σε PHP, περιέχει XHTML, CSS και JavaScript. Είναι ένα ελεύθερο εργαλείο λογισμικού ανοικτού κώδικα γραμμένο σε PHP, το οποίο παρέχει ένα ολοκληρωμένο περιβάλλον για την διαχείριση MySQL βάσεων δεδομένων. Αρχικά αναπτύχθηκε από τον Tobias Ratschiller το 1998 και έκτοτε έχει εξελιχθεί σε ένα από τα πιο δημοφιλή εργαλεία διαχείρισης βάσεων δεδομένων, χρησιμοποιούμενο ευρέως από διαχειριστές συστημάτων, προγραμματιστές και εταιρείες φιλοξενίας ιστοσελίδων. Το έχει πετύχει αυτό με κάποια από τα εξής χαρακτηριστικά που προσφέρει.

- **Διαχείριση Διακομιστή:** Παρέχει εργαλεία για την παρακολούθηση της απόδοσης του διακομιστή, τη διαχείριση των διαδικασιών και τη ρύθμιση των παραμέτρων του διακομιστή MySQL
- **Επέκταση και Παραμετροποίηση:** Η PhpMyAdmin διαθέτει επεκτάσιμο περιβάλλον, επιτρέποντας στους προγραμματιστές να προσθέσουν επιπλέον λειτουργίες μέσω πρόσθετων(plugins). Η διεπαφή του μπορεί επίσης να προσαρμοστεί σύμφωνα με τις προτιμήσεις του χρήστη.
- **Διαχείριση Δεδομένων:** Οι χρήστες μπορούν να προσθέτουν, να επεξεργάζονται, να διαγράφουν και να εξάγουν δεδομένα από τους πίνακες. Υπάρχουν επίσης εργαλεία για την εύκολη εισαγωγή δεδομένων από αρχεία CSV, SQL και άλλες μορφές.
- **Διαχείριση Βάσεων Δεδομένων:** Παρέχει την δυνατότητα εκτέλεσης ερωτημάτων SQL και την προβολή των αποτελεσμάτων τους. Επίσης μπορεί να διαχειριστεί και να τροποποιήσει πεδία πινάκων.
- **Δημιουργία και Διαχείριση Διαγραμμάτων:** Επιτρέπει στους χρήστες να δημιουργούν διαγράμματα ER(Entity-Relationship) που απεικονίζουν τις σχέσεις μεταξύ των πινάκων. Αυτό είναι ιδιαίτερα χρήσιμο για την κατανόηση των σύνθετων βάσεων δεδομένων και την ανάλυση της δομής του. (Delisle, 2012)

## 1.7 Τεχνολογίες Web:HTML,CSS,JavaScript

### 1.7.1 HTML

Η HTML είναι το ακρώνυμο του HyperText Markup Language είναι μια γλώσσα σήμανσης που χρησιμοποιείται για την δημιουργία και τη διόρθωση ιστοσελίδων και των περιεχομένων τους στο Διαδίκτυο. Δημιουργήθηκε από τον Tim Berners Lee το 1993 και έχει εξελιχθεί σε αυτό που ξέρουμε σήμερα. Στόχος της HTML είναι η επισήμανση των στοιχείων ενός εγγράφου, έτσι ώστε κάθε στοιχείο του εγγράφου να μορφοποιείται, να εμφανίζεται και να χρησιμοποιείται με κατάλληλο τρόπο. Το περιεχόμενο του εγγράφου μπορεί να είναι κείμενο, εικόνα ή οποιαδήποτε άλλη μορφή πολυμεσικής πληροφορίας. Δεν αποτελεί ωστόσο μια «ολοκληρωμένη» γλώσσα προγραμματισμού καθώς δεν είναι δυνατή η επίλυση υπολογιστικών προβλημάτων. Οι εντολές των γλωσσών σήμανσης είναι απλές και κατανοητές και ονομάζονται ετικέτες(tags). (DUCKETT, 2011)

Κάθε δομικό συστατικό (στοιχείο) ενός HTML εγγράφου αποτελείται από 3 μέρη:

- Την ετικέτα έναρξης(start tag), <...>
- Το περιεχόμενό του
- Την ετικέτα λήξης (end tag), </...>
- 

### 1.7.2 CSS

Η CSS είναι μια γλώσσα *στυλ* που χρησιμοποιείται για τον καθορισμό της εμφάνισης και της μορφοποίησης ενός εγγράφου γραμμένου σε HTML. Δημιουργήθηκε από τον Hakon Wium Lie το 1994, όπου εκείνη την περίοδο δούλευε μαζί με τον Tim Bernes-Lee τον δημιουργό της HTML στο Cern. Λειτουργεί σαν ένα αόρατο κουτί γύρω από κάθε στοιχείο της HTML. Λειτουργεί συσχετίζοντας τους κανόνες με τα στοιχεία της HTML. Αυτοί οι κανόνες καθορίζουν το πως θα εμφανίζονται συγκεκριμένα στοιχεία. Η CSS χρησιμοποιείται στους περισσότερους ιστότοπους για των διαχωρισμό του περιεχομένου HTML από την παρουσίαση. Η εφαρμογή της στις ιστοσελίδες γίνεται με τρεις τρόπους.

- Internal CSS: Ο κώδικας γράφεται στο ίδιο αρχείο όπου γράφουμε και τα στοιχεία του HTML.
- External CSS: Ο κώδικας γράφεται σε ένα διαφορετικό αρχείο από το .html αρχείο μας. Αυτός ο τρόπος είναι ο πιο ενδεδειγμένος καθώς μας βοηθάει να υπαρχή καλύτερη οργάνωση στην σελίδα που δημιουργούμε.
- Inline CSS: Ο κώδικας εφαρμόζεται απευθείας στα στοιχεία HTML μέσω του χαρακτηριστικού 'style'.

(DUCKETT, 2011)

### 1.7.3 JavaScript

Η JavaScript γενικά, είναι μια ολοκληρωμένη, υψηλού επιπέδου γλώσσα προγραμματισμού γενικής χρήσης σε αντίθεση με την HTML και την CSS. Δημιουργήθηκε το 1995 από το Brendan Eich, όταν αυτός εργαζόταν στην εταιρεία Netscape Communications με τη συνδρομή και της εταιρείας Sun Microsystems. Σε αρκετές περιπτώσεις δημιουργείται η ανάγκη να γράψει κανείς ευκολά και γρήγορα ένα σενάριο(script) για μια συγκεκριμένη εργασία που αφορά κάποια ή κάποιες σελίδες του διαδικτύου. Στόχος της JavaScript είναι να προσφέρει βοήθεια στο προγραμματιστή εκείνον ο οποίος επιθυμεί να κατασκευάσει αλληλεπιδραστικές και λειτουργικές ιστοσελίδες στο διαδίκτυο. Το μεγάλο πλεονέκτημα της JavaScript είναι η άμεση απόκριση στην αλληλεπίδραση με το χρήστη, σταματάει δηλαδή την αναγκαστική επαναφόρτιση των ιστοσελίδων. Τα πιο δημοφιλή προγράμματα περιήγησης ενσωματώνουν διερμηνευτή JavaScript. (Edge, Mozilla, Firefox, Google Chrome, Safari, Opera). (Λιακέας, 2015)

Δυνατότητες και περιορισμοί της JavaScript

- Έλεγχος της συμπεριφοράς και των λειτουργιών του προγράμματος περιήγησης.
  - Έλεγχος εγκυρότητας στοιχείων τα οποία εισάγονται με φόρμες.
  - Ανάγνωση ή καταγραφή της κατάστασης του υπολογιστή του χρήστη μέσω cookies
- Η JavaScript δεν μπορεί να προσπελάσει βάσεις δεδομένων, εκτός και αν χρησιμοποιήσει την τεχνολογία Ajax ενεργοποιώντας ένα κατάλληλο σενάριο στο διακομιστή.
  - Η JavaScript δεν μπορεί να προστατεύσει το περιεχόμενο μια ιστοσελίδας (κώδικα, εικόνες , κ.λ.π.) σε περίπτωση που κάποιος θελήσει να το αντιγράψει.
  - Για λόγους ασφάλειας η JavaScript δεν παρέχει τη δυνατότητα προσπέλασης ή χειρισμού αρχείων στον υπολογιστή του χρήστη. Η εγγραφή δεδομένων είναι επίσης περιοσμένη αποκλειστικά και μόνο στη δημιουργία των cookies. Τα cookies αποτελούνται μόνο από κείμενο και δε μπορούν να έχουν κάποια επίδραση στον υπολογιστή του χρήστη. Αυτή η δυνατότητα `προφυλάσσει το χρήστη από σενάρια τα οποία θα μπορούσαν να προξενήσουν βλάβες στον υπολογιστή του.

## Κεφάλαιο 2<sup>ο</sup>:

### 2.1 Λειτουργικές Απαιτήσεις

Μια λειτουργική απαίτηση περιγράφει πώς πρέπει να λειτουργεί ένα σύστημα για να ικανοποιήσει τις ανάγκες του χρήστη. Καθορίζει τις συγκεκριμένες ενέργειες που πρέπει να εκτελεί το σύστημα ώστε να ανταποκρίνεται στις προσδοκίες του χρήστη. Αυτές οι απαιτήσεις αντιπροσωπεύουν τα χαρακτηριστικά που είναι άμεσα αντιληπτά από τον χρήστη και διαφέρουν από τις μη λειτουργικές απαιτήσεις, οι οποίες επικεντρώνονται στην εσωτερική λειτουργία του συστήματος, όπως η απόδοση και η ασφάλεια.

Οι λειτουργικές απαιτήσεις αποτελούνται από δύο στοιχεία: τη λειτουργία και τη συμπεριφορά. Η λειτουργία αναφέρεται στο τι κάνει το σύστημα (π.χ. «επεξεργασία παραγγελίας πελάτη»), ενώ η συμπεριφορά περιγράφει πώς το σύστημα εκτελεί αυτή τη λειτουργία (π.χ. «Το σύστημα θα επεξεργαστεί την παραγγελία επικυρώνοντας τα στοιχεία του πελάτη και ενημερώνοντας το απόθεμα προϊόντων»). Ακολουθούν οι κύριες λειτουργικές απαιτήσεις του ηλεκτρονικού καταστήματος:

#### 2.1.1 Κατάλογος Προϊόντων

- **Περιήγηση Προϊόντων:** Οι χρήστες πρέπει να μπορούν να περιηγούνται σε κατηγορίες προϊόντων και να βλέπουν λεπτομέρειες προϊόντων όπως όνομα, περιγραφή και τιμή.
- **Φιλτράρισμα και Ταξινόμηση:** Δυνατότητα φιλτραρίσματος προϊόντων με βάση διάφορα κριτήρια (π.χ., ανδρικό, γυναικείο) και ταξινόμησης των αποτελεσμάτων.

#### 2.1.2 Διαχείριση Καλαθιού Αγορών

- **Προσθήκη στο Καλάθι:** Οι χρήστες πρέπει να μπορούν να προσθέτουν προϊόντα στο καλάθι αγορών τους.
- **Ενημέρωση Ποσοτήτων:** Οι χρήστες πρέπει να μπορούν να αλλάζουν τις ποσότητες των προϊόντων στο καλάθι αγορών τους.
- **Αφαίρεση Προϊόντων:** Δυνατότητα αφαίρεσης προϊόντων από το καλάθι.
- **Προβολή Καλαθιού:** Οι χρήστες πρέπει να μπορούν να βλέπουν τα προϊόντα που έχουν προσθέσει στο καλάθι τους και το συνολικό κόστος.

### 2.1.3 Διαδικασία Παραγγελίας

- **Στοιχεία Πληρωμής:** Οι χρήστες πρέπει να εισάγουν τα στοιχεία πληρωμής για την ολοκλήρωση της παραγγελίας.
- **Επιβεβαίωση Παραγγελίας:** Οι χρήστες πρέπει να λαμβάνουν επιβεβαίωση της παραγγελίας τους με πληροφορίες όπως ο αριθμός παραγγελίας και η αναμενόμενη ημερομηνία παράδοσης.
- **Παρακολούθηση Παραγγελίας:** Δυνατότητα παρακολούθησης της κατάστασης της παραγγελίας από την αποστολή μέχρι την παράδοση.

### 2.1.4 Διαχείριση Χρηστών και Αυθεντικοποίηση

- **Εγγραφή Χρηστών:** Οι χρήστες πρέπει να μπορούν να δημιουργούν λογαριασμούς με προσωπικές πληροφορίες.
- **Σύνδεση και Αποσύνδεση:** Οι χρήστες πρέπει να μπορούν να συνδέονται και να αποσυνδέονται από το σύστημα.
- **Ανάκτηση Κωδικού Πρόσβασης:** Δυνατότητα ανάκτησης του κωδικού πρόσβασης σε περίπτωση που ο χρήστης τον ξεχάσει.

## 2.2 Μη Λειτουργικές Απαιτήσεις

Μια μη λειτουργική απαίτηση περιγράφει τον τρόπο με τον οποίο λειτουργεί ένα σύστημα, αντί να επικεντρώνεται σε συγκεκριμένες λειτουργίες που εκτελεί. Αφορά κυρίως την απόδοση, την ασφάλεια, την αξιοπιστία και τη χρηστικότητα του συστήματος, δηλαδή τα χαρακτηριστικά που καθορίζουν την ποιότητα της λειτουργίας του συστήματος. Οι μη λειτουργικές απαιτήσεις δεν σχετίζονται με το τι κάνει το σύστημα, αλλά με το πώς το κάνει.

### 2.2.1 Ασφάλεια

- **Κρυπτογράφηση Δεδομένων:** Όλα τα ευαίσθητα δεδομένα, όπως οι κωδικοί πρόσβασης και οι πληροφορίες πληρωμών, πρέπει να είναι κρυπτογραφημένα.
- **Προστασία από Επιθέσεις:** Το σύστημα πρέπει να προστατεύεται από SQL injections, cross-site scripting (XSS) και άλλες κακόβουλες επιθέσεις.
- **Διαχείριση Συνεδριών:** Οι συνεδρίες χρηστών πρέπει να είναι ασφαλείς και να αποτρέπεται η μη εξουσιοδοτημένη πρόσβαση.

### 2.2.2 Απόδοση και Κλιμάκωση

- **Αποδοτικότητα:** Το σύστημα πρέπει να είναι αποδοτικό και να ανταποκρίνεται γρήγορα στις ενέργειες των χρηστών.

- **Κλιμακωσιμότητα:** Το σύστημα πρέπει να μπορεί να κλιμακωθεί για να υποστηρίζει αυξημένο αριθμό χρηστών και συναλλαγών χωρίς να επηρεάζεται η απόδοση.

### 2.2.3 Ευχρηστία

- **Φιλικότητα προς τον Χρήστη:** Η διεπαφή χρήστη πρέπει να είναι απλή, διαισθητική και εύκολη στη χρήση.

## 2.3 Τοπική εγκατάσταση

### 2.3.1 XAMPP

Το XAMPP είναι μια δωρεάν και ανοικτού κώδικα πλατφόρμα ανάπτυξης διακομιστών web που περιλαμβάνει το Apache, MySQL (ή MariaDB), PHP και Perl. Σχεδιασμένο για να διευκολύνει τους προγραμματιστές, το XAMPP επιτρέπει την εύκολη εγκατάσταση και διαχείριση ενός πλήρους διακομιστή web σε τοπικό υπολογιστή. Αυτό καθιστά την ανάπτυξη και τον εντοπισμό σφαλμάτων ιστοσελίδων και εφαρμογών πιο προσιτή και αποδοτική, πριν από την ανάπτυξή τους σε ζωντανούς διακομιστές.

Η χρήση του XAMPP παρέχει ένα ολοκληρωμένο περιβάλλον εργασίας που υποστηρίζει πολλαπλές πλατφόρμες, όπως Windows, Linux και macOS. Το εργαλείο αυτό είναι ιδιαίτερα δημοφιλές λόγω της ευκολίας χρήσης του, περιλαμβάνοντας εργαλεία όπως το phpMyAdmin για τη διαχείριση βάσεων δεδομένων MySQL μέσω γραφικού περιβάλλοντος. Με το XAMPP, οι προγραμματιστές μπορούν να αναπτύσσουν δυναμικές ιστοσελίδες και να δοκιμάζουν νέες λειτουργίες χωρίς την ανάγκη συνεχούς σύνδεσης στο διαδίκτυο.

### 2.3.2 Visual Studio Code

Το Visual Studio Code, κοινώς γνωστό ως VS Code, είναι ένας δωρεάν και ανοικτού κώδικα κειμενογράφος (code editor) που αναπτύχθηκε από τη Microsoft. Χρησιμοποιείται ευρέως για την ανάπτυξη λογισμικού και παρέχει πολλές δυνατότητες που διευκολύνουν τους προγραμματιστές. Ακολουθούν μερικά από τα βασικά χαρακτηριστικά του VS Code:

- **Υποστήριξη Πολλών Γλωσσών Προγραμματισμού:** Το VS Code υποστηρίζει διάφορες γλώσσες προγραμματισμού όπως JavaScript, Python, Java, C++, HTML, CSS, και πολλές άλλες.
- **Επεκτάσεις (Extensions):** Μπορείς να προσθέσεις νέες λειτουργίες μέσω των επεκτάσεων, όπως υποστήριξη για περισσότερες γλώσσες προγραμματισμού, εργαλεία debugging, θεματικά πακέτα και άλλα.

- **Ενσωματωμένο Terminal:** Περιλαμβάνει ενσωματωμένο terminal, επιτρέποντάς σου να εκτελείς εντολές από τον κειμενογράφο χωρίς να αλλάζεις παράθυρα.
- **Αυτόματη Συμπλήρωση Κώδικα (IntelliSense):** Προσφέρει λειτουργία IntelliSense, η οποία παρέχει έξυπνες προτάσεις κώδικα και αυτόματη συμπλήρωση καθώς γράφεις.
- **Δυνατότητες Debugging:** Έχει ενσωματωμένα εργαλεία debugging που επιτρέπουν την εύκολη ανάλυση και διόρθωση σφαλμάτων στον κώδικά σου.
- **Εύκολη Πλοήγηση και Αναζήτηση:** Παρέχει ισχυρά εργαλεία αναζήτησης και πλοήγησης στον κώδικα, διευκολύνοντας την εύρεση συγκεκριμένων γραμμών ή αρχείων.
- **Git Integration:** Έχει ενσωματωμένη υποστήριξη για Git, επιτρέποντας τη διαχείριση αποθετηρίων (repositories) απευθείας από το VS Code.
- **Προσαρμοσμένα Θέματα και Εμφάνιση:** Επιτρέπει την προσαρμογή της εμφάνισης του editor με διάφορα θέματα και προσαρμοσμένες επιλογές.

Το VS Code είναι ιδιαίτερα δημοφιλές λόγω της ευελιξίας και της ισχυρής υποστήριξης της κοινότητας, καθιστώντας το ένα από τα πιο χρησιμοποιημένα εργαλεία για προγραμματιστές παγκοσμίως.

## 2.4 Online

### 2.4.1 Domain Name

Ένα domain name είναι μια μοναδική διεύθυνση που χρησιμοποιείται για την ταυτοποίηση ενός ιστότοπου στο διαδίκτυο. Τα domain names παρέχουν έναν εύκολο τρόπο για να θυμόμαστε και να βρίσκουμε ιστότοπους χωρίς να χρειάζεται να χρησιμοποιούμε τις δύσκολες στη μνήμη διευθύνσεις IP (Internet Protocol). Ακολουθούν βασικές πληροφορίες για τα domain names:

#### Δομή Domain Name

- **Top-Level Domain (TLD):** Είναι η τελευταία κατάληξη του domain name, όπως .com, .org, .net, .gov, .edu, και πολλές άλλες. Υπάρχουν γενικά TLDs (gTLDs) και κωδικοί χωρών (ccTLDs), όπως .gr για την Ελλάδα.

- **Second-Level Domain:** Είναι το κύριο τμήμα του domain name, όπως "motoroda" στο motoroda.com. Είναι αυτό που επιλέγεις όταν κατοχυρώνεις ένα domain name.
- **Subdomain:** Είναι προαιρετικό και αποτελείται από ένα τμήμα που προηγείται του second-level domain, όπως "blog" στο blog.motoroda.com.

### Διαδικασία Κατοχύρωσης Domain Name

- **Επιλογή Ονόματος:** Επιλέγεις ένα μοναδικό όνομα που αντιπροσωπεύει την επιχείρησή σου, τον ιστότοπό σου ή την προσωπική σου μάρκα.
- **Έλεγχος Διαθεσιμότητας:** Χρησιμοποιείς μια υπηρεσία καταχώρησης domain (domain register) για να ελέγξεις αν το επιλεγμένο όνομα είναι διαθέσιμο.
- **Κατοχύρωση:** Εφόσον το όνομα είναι διαθέσιμο, το κατοχυρώνεις μέσω ενός καταχωρητή domain, όπως το GoDaddy, το Namecheap, ή το Google Domains.
- **Διαχείριση DNS:** Ρυθμίζεις τις εγγραφές DNS (Domain Name System) για να κατευθύνεις το domain name σου στον διακομιστή (server) που φιλοξενεί τον ιστότοπό σου.

### Λειτουργία Domain Name

Τα domain names λειτουργούν μέσω του συστήματος DNS (Domain Name System), το οποίο μετατρέπει τα domain names σε διευθύνσεις IP που κατανοούν οι υπολογιστές. Όταν πληκτρολογείς ένα domain name στον περιηγητή σου, το DNS server αναλαμβάνει να βρει την αντίστοιχη διεύθυνση IP και να σε συνδέσει στον σωστό ιστότοπο.

### Σημασία του Domain Name

- **Ευκολία Χρήσης:** Διευκολύνει τους χρήστες να θυμούνται και να βρίσκουν έναν ιστότοπο.
- **Επαγγελματισμός:** Ένα καλό domain name προσδίδει επαγγελματικό κύρος και αξιοπιστία.
- **Μάρκετινγκ και Branding:** Βοηθά στην προώθηση της μάρκας και μπορεί να επηρεάσει θετικά το SEO (Search Engine Optimization).

Τα domain names αποτελούν βασικό στοιχείο του διαδικτύου και είναι απαραίτητα για τη δημιουργία και τη διαχείριση ενός επιτυχημένου ιστότοπου.

### 2.4.2 Web Server

Οι web servers είναι θεμελιώδεις για τη λειτουργία του διαδικτύου, καθώς αποτελούν τον πυρήνα κάθε ιστότοπου. Είναι υπεύθυνοι για την αποθήκευση, την επεξεργασία και την παράδοση περιεχομένου στους χρήστες με ασφάλεια και αποτελεσματικότητα. Η επιλογή του κατάλληλου web server και η σωστή διαχείρισή του είναι κρίσιμα στοιχεία για την επιτυχία ενός ιστότοπου. Ακολουθούν βασικές πληροφορίες σχετικά με τους web servers:

#### Λειτουργία Web Server

- **Αποθήκευση Περιεχομένου:** Ο web server αποθηκεύει αρχεία ιστού, όπως HTML, CSS, JavaScript, εικόνες και βίντεο. Αυτά τα αρχεία συνθέτουν τις ιστοσελίδες που βλέπουν οι χρήστες.
- **Απάντηση σε Αιτήματα:** Όταν ένας χρήστης εισάγει μια διεύθυνση URL στον περιηγητή του, το αίτημα αποστέλλεται στον αντίστοιχο web server. Ο διακομιστής αναζητά το ζητούμενο αρχείο και το αποστέλλει πίσω στον περιηγητή, ο οποίος το εμφανίζει ως ιστοσελίδα.
- **Επεξεργασία Δεδομένων:** Σε δυναμικούς ιστότοπους, οι web servers επεξεργάζονται δεδομένα μέσω server-side scripts, όπως PHP, Python, Ruby ή άλλες γλώσσες προγραμματισμού. Αυτά τα scripts μπορεί να αλληλεπιδρούν με βάσεις δεδομένων για την παραγωγή δυναμικού περιεχομένου.
- **Ασφάλεια:** Οι web servers παρέχουν επίσης διάφορα επίπεδα ασφαλείας για την προστασία του περιεχομένου και των δεδομένων χρηστών. Αυτό περιλαμβάνει τη χρήση SSL/TLS για κρυπτογράφηση δεδομένων και μηχανισμούς προστασίας από επιθέσεις, όπως το DDoS.

#### Κύριοι Τύποι Web Servers

- **Apache HTTP Server:** Ένας από τους πιο δημοφιλείς web servers. Είναι ανοικτού κώδικα και χρησιμοποιείται ευρέως για τη φιλοξενία ιστοσελίδων σε Linux και Unix-based συστήματα.
- **Nginx:** Ένας άλλος δημοφιλής web server, γνωστός για την υψηλή του απόδοση και την ικανότητά του να διαχειρίζεται μεγάλο όγκο ταυτόχρονων συνδέσεων. Χρησιμοποιείται τόσο ως web server όσο και ως reverse proxy server.

- **Microsoft Internet Information Services (IIS):** Ένας web server που αναπτύσσεται από τη Microsoft και είναι ενσωματωμένος στα Windows Server λειτουργικά συστήματα. Χρησιμοποιείται κυρίως σε περιβάλλοντα Windows.
- **LiteSpeed:** Ένας υψηλής απόδοσης web server που είναι γνωστός για την ταχύτητά του και την αποτελεσματική διαχείριση πόρων. Χρησιμοποιείται συχνά ως εναλλακτική λύση του Apache.

#### Βασικές Λειτουργίες και Χαρακτηριστικά

- **Φιλοξενία Ιστοσελίδων:** Ο κύριος ρόλος του web server είναι η φιλοξενία και η παράδοση ιστοσελίδων στους χρήστες.
- **Διαχείριση Συνεδριών και Cookies:** Οι web servers διαχειρίζονται συνεδρίες χρηστών και αποθηκεύουν πληροφορίες μέσω cookies.
- **Logging και Παρακολούθηση:** Καταγράφουν και παρακολουθούν τη δραστηριότητα των χρηστών και την απόδοση του διακομιστή.
- **Επέκταση και Ενσωμάτωση:** Δυνατότητα προσθήκης modules και plugins για την επέκταση των λειτουργιών του web server. (Gourley & Totty, 2002) (Soni, 2016)

# Κεφάλαιο 3<sup>ο</sup>

## Σχεδιασμός Συστήματος

Το κεφάλαιο αυτό περιγράφει τον σχεδιασμό του ηλεκτρονικού καταστήματος, περιλαμβάνοντας την αρχιτεκτονική του συστήματος, τον σχεδιασμό της βάσης δεδομένων, την σχεδίαση της διεπαφής χρήστη και τη δημιουργία διαγραμμάτων UML για την οπτική αναπαράσταση του συστήματος και των λειτουργιών του.

### 3.1 Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική του συστήματος καθορίζει τη δομή και τη λειτουργία του ηλεκτρονικού καταστήματος. Το σύστημα βασίζεται σε μια πολυεπίπεδη αρχιτεκτονική που περιλαμβάνει τα εξής επίπεδα:

#### 3.1.1 Επίπεδο Παρουσίασης (*Presentation Layer*)

Το επίπεδο παρουσίασης είναι υπεύθυνο για την αλληλεπίδραση με τους χρήστες. Περιλαμβάνει τις ιστοσελίδες που προβάλλονται στους χρήστες και χρησιμοποιεί HTML, CSS και JavaScript για τη δημιουργία μιας φιλικής και διαδραστικής διεπαφής χρήστη.

#### 3.1.2 Επίπεδο Επιχειρηματικής Λογικής (*Business Logic Layer*)

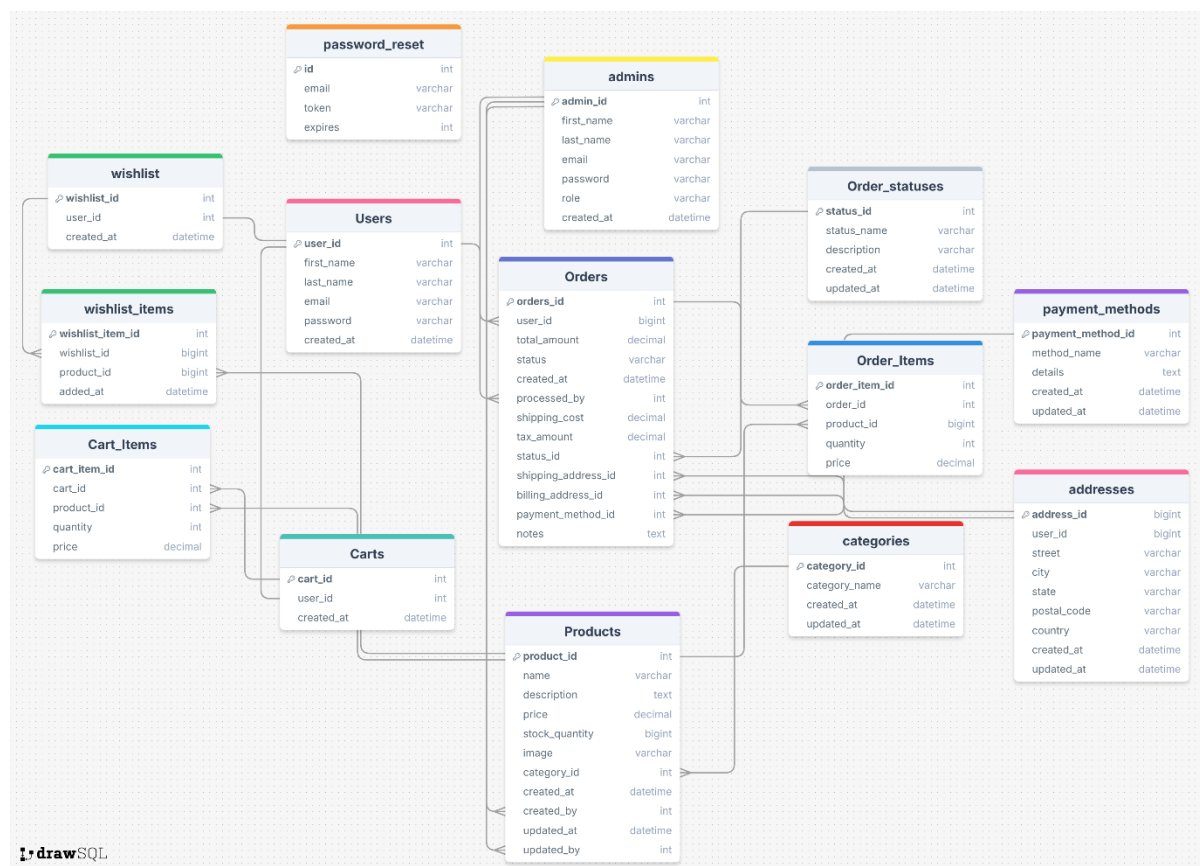
Το επίπεδο επιχειρηματικής λογικής περιέχει τη λογική της εφαρμογής και τις λειτουργίες που εκτελούνται από το σύστημα. Χρησιμοποιεί PHP για την επεξεργασία των αιτημάτων των χρηστών, την επαλήθευση των δεδομένων και την εφαρμογή των επιχειρηματικών κανόνων.

#### 3.1.3 Επίπεδο Δεδομένων (*Data Layer*)

Το επίπεδο δεδομένων είναι υπεύθυνο για την αποθήκευση και την ανάκτηση δεδομένων. Χρησιμοποιεί MySQL για τη διαχείριση της βάσης δεδομένων και την εκτέλεση ερωτημάτων για την αποθήκευση και την ανάκτηση πληροφοριών σχετικά με προϊόντα, χρήστες και παραγγελίες.

## 3.2 Σχεδιασμός Βάσης Δεδομένων

Ο σχεδιασμός της βάσης δεδομένων είναι κρίσιμος για την αποδοτική λειτουργία του ηλεκτρονικού καταστήματος. Περιλαμβάνει τη δημιουργία πινάκων για την αποθήκευση πληροφοριών προϊόντων, χρηστών, παραγγελιών και άλλων δεδομένων. Ακολουθεί μια εικόνα της συνολικής βάσης καθώς και η περιγραφή των κύριων πινάκων:



Εικόνα 3 1 Βάση δεδομένων

### 3.2.1 Πίνακας Χρηστών(Users)

- **User\_id(int):** Primary key, μοναδικό για κάθε χρήστη.
- **First\_name(varchar):** Το όνομα του χρήστη.
- **Last\_name(varchar):** Το επώνυμο του χρήστη.
- **Email(varchar):** Το email του χρήστη, μοναδικό.
- **Password(varchar):** Hashed κωδικός για την αυθεντικοποίηση του χρήστη.
- **Created\_at(datetime):** Η στιγμή που δημιουργήθηκε ο χρήστης.

**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για τον χρήστη, όπως προσωπικές πληροφορίες και στοιχεία για την σύνδεση του.

### 3.2.2 Πίνακας Διαχειριστών(Admins)

- **admin\_id(int):** Primary key, μοναδικό για κάθε διαχειριστή.
- **First\_name(varchar):** Το όνομα του διαχειριστή.
- **Last\_name(varchar):** Το επώνυμο του διαχειριστή.
- **Email(varchar):** Το email του διαχειριστή, μοναδικό.
- **Password(varchar):** Hashed κωδικός για την αυθεντικοποίηση του διαχειριστή.
- **Role(varchar):** Ρόλος για τον διαχειριστή (π.χ. superadmin, moderator).
- **Created\_at(datetime):** Η στιγμή που δημιουργήθηκε ο διαχειριστή.

**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για τον διαχειριστή, όπως προσωπικές πληροφορίες και στοιχεία για την σύνδεση του καθώς επίσης και τις δυνατότητες του.

### 3.2.3 Πίνακας Επαναφορά Κωδικών(Users)

- **id(int):** Primary key, μοναδικό για κάθε κωδικό επαναφοράς.
- **Email(varchar):** Το email του χρήστη που αιτείται επαναφορά κωδικού.
- **Token(varchar):** Χρησιμοποιείται για την επικαιροποίηση του κωδικού επαναφοράς.
- **expires (int):** Ο χρόνος λήξης για το token επαναφοράς.

**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για το token και τον χρόνο λήξης αυτών, για την επαναφορά των κωδικών.

### 3.2.4 Πίνακας Διευθύνσεων (Addresses)

- **address\_id(int):** Primary key, μοναδικό για κάθε διεύθυνση.
- **User\_id(bigint):** Ξένο κλειδί όπου αναφέρεται στην διεύθυνση που ανήκει στον χρήστη.
- **Street, city, state, postal\_code, country(varchar):** Στοιχεία της διεύθυνσης.
- **Created\_at(datetime):** Η στιγμή που δημιουργήθηκε ο χρήστης.
- **Updated\_at(datetime):** Η στιγμή που η διεύθυνση ανανεώθηκε τελευταία φορά.

**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για την διεύθυνση του χρήστη.

### 3.2.5 Πίνακας Παραγγελιών(Orders)

- **Orders\_id(int):** Primary key, μοναδικό για κάθε παραγγελία.
- **User\_id(int):** Ξένο κλειδί αναφερόμενο στους χρήστες, δείχνοντας ποιος έκανε την παραγγελία.
- **Total\_amount(decimal):** Το συνολικό ποσό της παραγγελίας.
- **Status(varchar):** Η κατάσταση της παραγγελίας (π.χ. ολοκληρωμένη, σε αναμονή).
- **Created\_at(datetime):** Η στιγμή που δημιουργήθηκε η παραγγελία.
- **Processed\_by(int):** Το ID του διαχειριστή υποδεικνύοντας ποιος διαχειριστικό την παραγγελία.

- **Shipping\_cost(decimal):** Το κόστος δρομολόγησης.
- **Tax\_amount(decimal):** Το ποσό των φόρων.
- **Status\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα order\_statuses.
- **Shipping\_address\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα addresses για την δρομολόγηση της παραγγελίας.
- **Billing\_address\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα addresses για την κοστολόγηση της παραγγελίας.
- **Payment\_method\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα payment\_methods.
- **Notes(text):** Επιπλέον σημειώσεις για την παραγγελία.

**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για την παραγγελία, για τον χρήστη, τον τρόπο πληρωμής, δρομολόγησης και την συνολική κατάσταση της παραγγελίας.

### 3.2.6 Πίνακας Παραγγελίας αντικειμένων(Order\_Items)

- **Order\_item\_id(int):** Primary key, μοναδικό για κάθε αντικείμενο παραγγελίας.
- **Order\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα Orders, δείχνοντας σε πια παραγγελία ανήκει το αντικείμενο.
- **Product\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα Products, δείχνοντας πιο προϊόν είναι στην παραγγελία.
- **quantity(int):** Η ποσότητα των προϊόντων που παραγγέλθηκαν.
- **Price(decimal):** Η τιμή των προϊόντων την στιγμή της παραγγελίας.

**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για τα αντικείμενα των παραγγελιών, δημιουργεί σύνδεση των παραγγελιών με τα προϊόντα που αγοράστηκαν.

### 3.2.7 Πίνακας Προϊόντων (Products)

- **product\_id(int):** Primary key, μοναδικό για κάθε προϊόν.
- **name(varchar):** Το όνομα του προϊόντος.
- **description(text):** Περιγραφή του προϊόντος.
- **price(decimal):** Η τιμή του προϊόντος.
- **Stock\_quantity(varchar):** Διαθέσιμο απόθεμα του προϊόντος.
- **Image(varchar):** Διευθυνσιοδότηση (URL) της εικόνας του προϊόντος.
- **Category\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα κατηγορίες.
- **Created\_at(datetime):** Η χρονική στιγμή που δημιουργήθηκε το προϊόν.
- **Created\_by(int):** Το ID του διαχειριστή υποδεικνύοντας ποιος δημιούργησε την εισαγωγή του προϊόντος
- **Updated\_by(int):** Το ID του διαχειριστή υποδεικνύοντας ποιος ανανέωσε την εισαγωγή του προϊόντος τελευταία φορά.

**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για κάθε διαθέσιμο προϊόν στην βάση δεδομένων μας. Περιλαμβάνοντας απόθεμα και κατηγορία.

### 3.2.8 Πίνακας Κατηγοριών(Categories)

- **category\_id(int):** Primary key, μοναδικό για κάθε κατηγορία.
- **category\_name(varchar):** Το όνομα της κάθε κατηγορίας
- **Created\_at(datetime):** Η χρονική στιγμή που δημιουργήθηκε η κατηγορία.
- **Updated\_at(datetime):** Η χρονική στιγμή που η κατηγορία ανανεώθηκε τελευταία φορά. **Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει πληροφορίες για τον χρήστη, όπως προσωπικές πληροφορίες και στοιχεία για την σύνδεση του.

### 3.2.9 Πίνακας Κατάστασης Παραγγελίας(Order\_Statues)

- **status\_id(int):** Primary key, μοναδικό για κάθε την κατάσταση παραγγελίας.
- **status\_name(varchar):** Η κατάσταση της παραγγελίας (π.χ. Pending, Shipped, Delivered).
- **description(varchar):** Περιγραφή της κατάστασης.
- **Created\_at(datetime):** Η χρονική στιγμή που δημιουργήθηκε η κατάσταση.
- **Updated\_at(datetime):** Η χρονική στιγμή που η κατάσταση ανανεώθηκε τελευταία φορά.  
**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει τις καταστάσεις μιας παραγγελίας.

### 3.2.10 Πίνακας Τρόπος Πληρωμής(Payment\_Methods)

- **Payment\_method\_id(int):** Primary key, μοναδικό για κάθε την κατάσταση παραγγελίας.
- **Method\_name(varchar):** Το όνομα του τρόπου πληρωμής (π.χ. Πιστωτική Κάρτα, Paypal)
- **details(text):** Λεπτομέρειες για τον τρόπο πληρωμής.
- **Created\_at(datetime):** Η χρονική στιγμή που ο τρόπος πληρωμής προστέθηκε.
- **Updated\_at(datetime):** Η χρονική στιγμή που η κατάσταση ανανεώθηκε τελευταία φορά.  
**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει διάφορους τρόπους πληρωμής της παραγγελίας.

### 3.2.11 Πίνακας Καλαθιών (Carts)

- **cart\_id(int):** Primary key, μοναδικό για κάθε την κατάσταση παραγγελίας.
- **User\_id(int):** Ξένο κλειδί αναφερόμενο στους χρήστες, υποδεικνύοντας σε ποιόν χρήστη ανήκει το καλάθι.
- **Created\_at(datetime):** Η χρονική στιγμή που το καλάθι δημιουργήθηκε.  
**Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει προϊόντα τα οποία ο χρήστης σκοπεύει να αγοράσει.

### 3.2.12 Πίνακας Αντικείμενα καλαθιών(Cart\_Items)

- **Cart\_item\_id(int):** Primary key, μοναδικό για κάθε αντικείμενο καλαθιού.

- **Cart\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα Carts, υποδεικνύοντας σε πιο καλάθι ανήκει το αντικείμενο.
  - **Product\_id(int):** Ξένο κλειδί αναφερόμενο στον πίνακα Products, υποδεικνύοντάς πιο προϊόν είναι στο καλάθι.
  - **quantity(int):** Η ποσότητα του προϊόντας στο καλάθι.
  - **price(decimal):** Η τιμή του προϊόντος την στιγμή που αυτό μπήκε στο καλάθι.
- Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει αντικείμενα που ανήκουν στο καλάθι ενός χρήστη, που είναι πριν την αγορά.

### 3.2.13 Πίνακας Λίστας Επιθυμιών(Wishlist)

- **wishlist\_id(int):** Primary key, μοναδικό για κάθε λίστα επιθυμιών.
  - **User\_id(int):** Ξένο κλειδί που αναφέρεται στο πίνακα Users, υποδεικνύοντας σε ποιον χρήστη ανήκει η λίστα επιθυμιών.
  - **Created\_at(datetime):** Η χρονική στιγμή που δημιουργήθηκε η λίστα επιθυμιών.
- Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει τις λίστες επιθυμιών των χρηστών.

### 3.2.14 Πίνακας Αντικείμενα Λίστας Επιθυμιών(Wishlist\_items)

- **Wishlist\_item\_id(int):** Primary key, μοναδικό για κάθε την κατάσταση παραγγελίας.
  - **Wishlist\_id(int):** Ξένο κλειδί που αναφέρεται στον πίνακα Wishlist, υποδεικνύοντας σε πια λίστα επιθυμιών ανήκει το κλειδί.
  - **Product\_id(int):** Ξένο κλειδί που αναφέρεται στον πίνακα Products, υποδεικνύοντας πιο προϊόν είναι στην λίστα επιθυμιών.
  - **Added\_at(datetime):** Η χρονική στιγμή που το προϊόν τοποθετήθηκε στην λίστα επιθυμιών.
- Σκοπός:** Ο πίνακας δεδομένων αποθηκεύει τις λεπτομέρειες ενός προϊόντος που είναι στην λίστα επιθυμιών του χρήστη.

## 3.3 Σχεδιασμός Διεπαφής Χρήστη

Η διεπαφή χρήστη (UI) είναι το τμήμα της εφαρμογής που αλληλοεπιδρά άμεσα με τον χρήστη. Ο σχεδιασμός του UI πρέπει να είναι φιλικός και εύχρηστος, διευκολύνοντας την πλοήγηση και τη χρήση του ηλεκτρονικού καταστήματος.

### 3.3.1 Αρχική Σελίδα

Η αρχική σελίδα περιλαμβάνει την κατηγορία προϊόντων, μια φόρμα επικοινωνίας καθώς επίσης και τον τρόπο με τον οποίο ο χρήστης μπορεί να συνδεθεί ή να δημιουργήσει καινούργιο λογαριασμό. Οφείλει να είναι εύκολα προσβάσιμη και να προσφέρει συνδέσμους για την πλοήγηση σε άλλες σελίδες του καταστήματος.

### 3.3.2 Σελίδα Κατηγορίες

Η σελίδα κατηγοριών επιτρέπει στους χρήστες να περιηγούνται στις διάφορες κατηγορίες προϊόντων του ηλεκτρονικού καταστήματος. Κάθε κατηγορία παρουσιάζει μια συλλογή προϊόντων που ανήκουν στην ίδια ομάδα, διευκολύνοντας την αναζήτηση και την επιλογή.

### 3.3.3 Σελίδα Προϊόντος

Η σελίδα προϊόντος παρέχει λεπτομερείς πληροφορίες για κάθε προϊόν, όπως περιγραφή, τιμή και φωτογραφίες. Οι χρήστες μπορούν να προσθέτουν προϊόντα στο καλάθι αγορών από αυτή τη σελίδα.

### 3.3.4 Καλάθι Αγορών

Η σελίδα καλαθιού αγορών εμφανίζει τα προϊόντα που έχουν προστεθεί στο καλάθι αγορών, επιτρέποντας στους χρήστες να ενημερώνουν τις ποσότητες, να αφαιρούν προϊόντα και να προχωρούν στην ολοκλήρωση της παραγγελίας.

### 3.3.5 Σελίδα Ολοκλήρωσης Παραγγελίας (Checkout)

Η σελίδα ολοκλήρωσης παραγγελίας συλλέγει τις απαραίτητες πληροφορίες αποστολής και πληρωμής από τον χρήστη και επιβεβαιώνει την παραγγελία.

### 3.3.6 Σελίδα Σύνδεσης

Η σελίδα σύνδεσης επιτρέπει στους χρήστες να εισέλθουν στον λογαριασμό τους για να έχουν πρόσβαση στις προσωπικές τους πληροφορίες και στις παραγγελίες τους.

### 3.3.7 Σελίδα Εγγραφής

Η σελίδα εγγραφής επιτρέπει στους νέους χρήστες να δημιουργήσουν έναν λογαριασμό, παρέχοντας τις απαραίτητες πληροφορίες για να έχουν πρόσβαση στις υπηρεσίες του ηλεκτρονικού καταστήματος.

### 3.3.8 Σελίδα Επικοινωνίας

Η σελίδα επικοινωνίας παρέχει στους χρήστες τα απαραίτητα μέσα για να επικοινωνήσουν με την ομάδα υποστήριξης του καταστήματος, είτε για απορίες είτε για υποστήριξη μετά την πώληση.

### 3.3.9 Σελίδα Παραγγελιών

Η σελίδα παραγγελιών επιτρέπει στους χρήστες να δουν το ιστορικό των παραγγελιών τους και την κατάσταση κάθε παραγγελίας.

### 3.3.10 Σελίδα Αποσύνδεσης

Η σελίδα αποσύνδεσης επιτρέπει στους χρήστες να αποσυνδέονται με ασφάλεια από τον λογαριασμό τους. Πρέπει να επιβεβαιώνει την αποσύνδεση και να παρέχει μια σαφή διαδρομή για περαιτέρω ενέργειες.

### 3.3.11 Σελίδα Λίστα Επιθυμιών

Η σελίδα λίστας επιθυμιών επιτρέπει στους χρήστες να αποθηκεύουν προϊόντα που τους ενδιαφέρουν για μελλοντική αγορά.

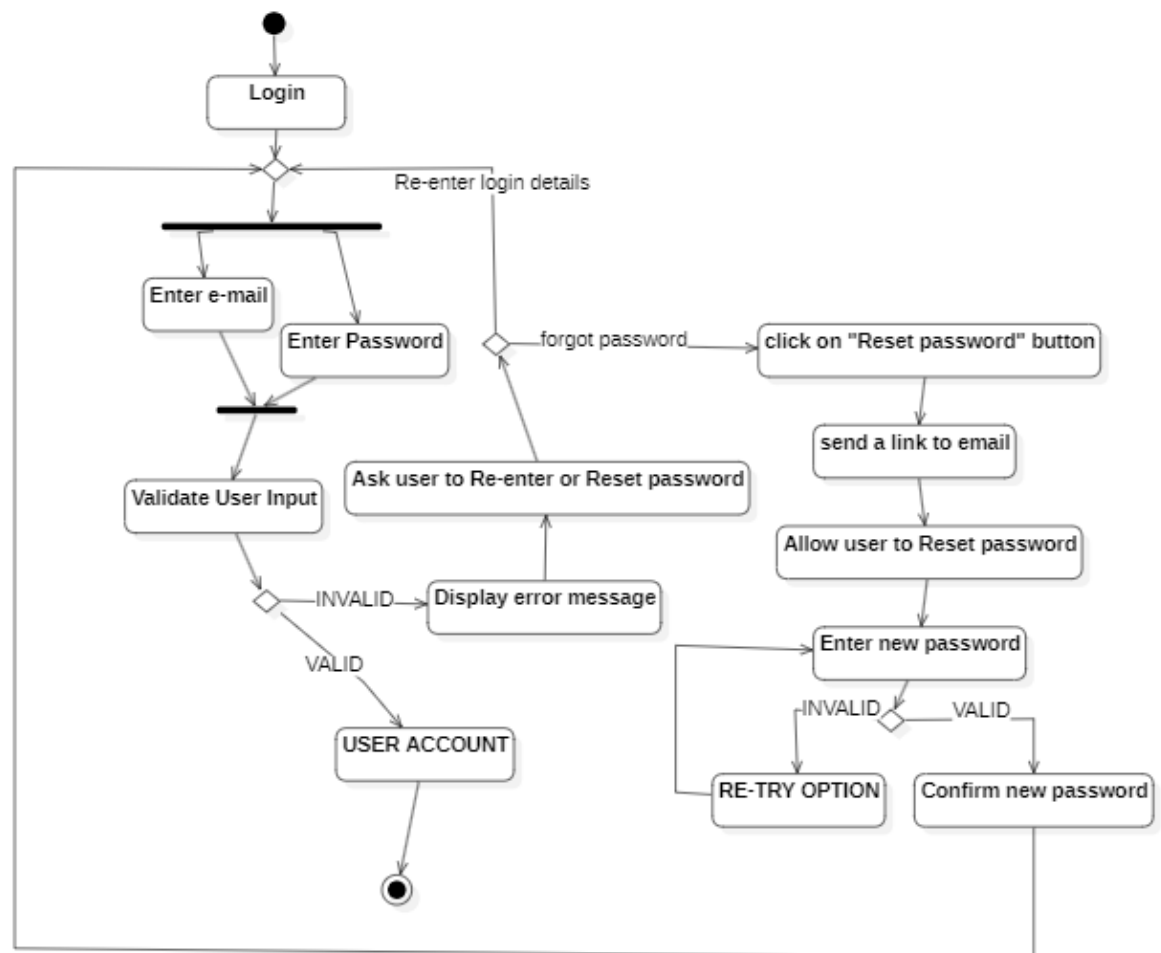
## 3.4 Διαγράμματα UML

Τα διαγράμματα UML (Unified Modeling Language) χρησιμοποιούνται για την οπτική αναπαράσταση του συστήματος και των λειτουργιών του. Ακολουθούν τα κύρια διαγράμματα UML που χρησιμοποιούνται για τον σχεδιασμό του ηλεκτρονικού καταστήματος:

### 3.4.1 Διαγράμματα Δραστηριοτήτων (Activity Diagram)

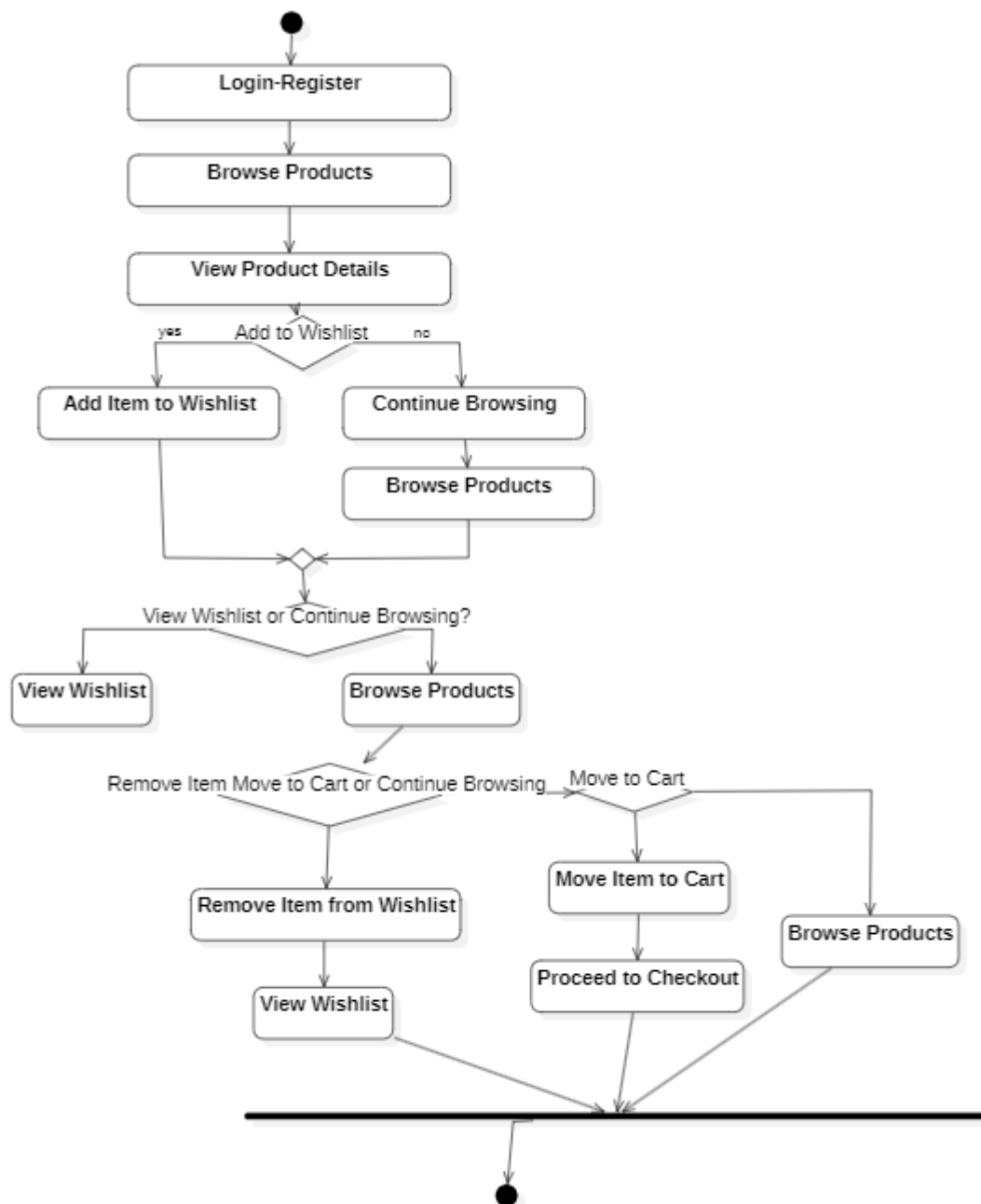
Το διάγραμμα δραστηριοτήτων απεικονίζει τη ροή εργασιών και τις διαδικασίες του συστήματος. Για την δημιουργία των διαγραμμάτων χρησιμοποιήθηκε το εργαλείο

StarUML. Ακολουθεί το διάγραμμα σύνδεσης και επαναφοράς κωδικού χρήστη.



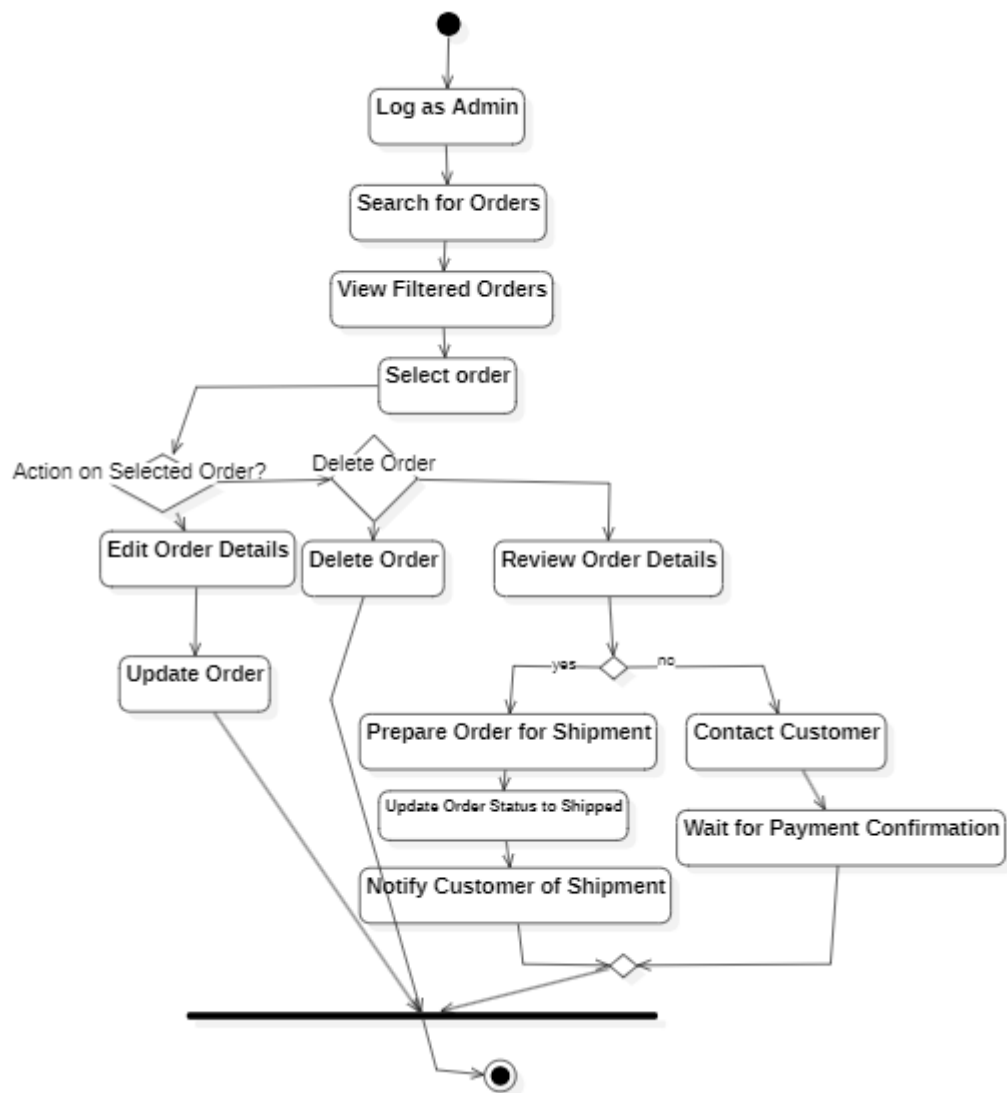
Εικόνα 3 2 Σύνδεση και επαναφορά κωδικού.

### 3.4.2 Διάγραμμα Διαχείρισης Λίστας Επιθυμιών



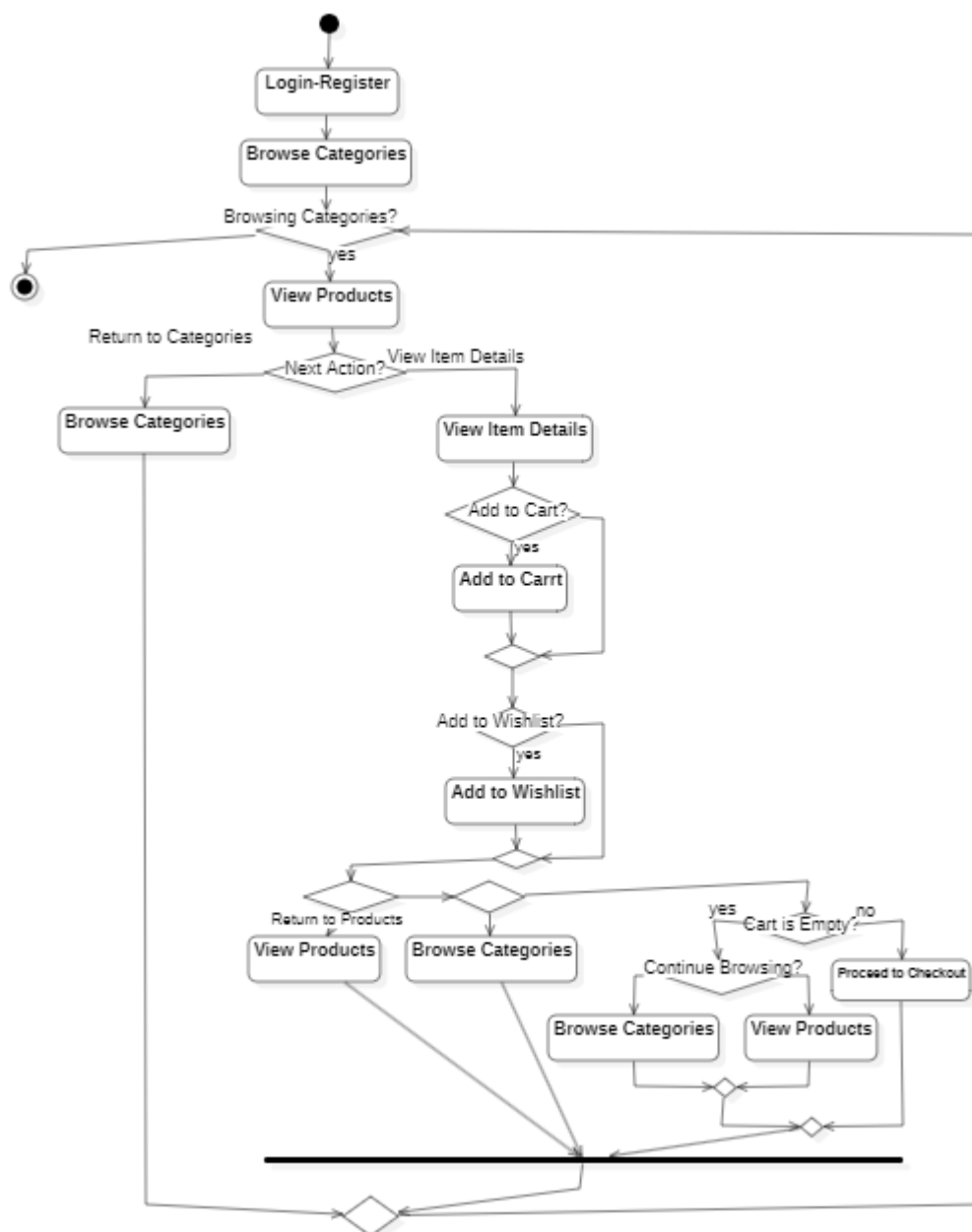
Εικόνα 3 3 Διαχείριση Λίστας Επιθυμιών

### 3.4.3 Διάγραμμα Διαχείρισης Παραγγελίας από τον Διαχειριστή.



Εικόνα 3 4 Διαχείριση Παραγγελίας από Διαχειριστή.

### 3.4.4 Διάγραμμα Περιήγησης Προϊόντων και Χρήση Καλαθιού.



Εικόνα 3 5 Περιήγησης Προϊόντων και Χρήση Καλαθιού

### 3.5 Διαχείριση Έργου και Χρονοδιάγραμμα

Η επιτυχής ανάπτυξη του ηλεκτρονικού καταστήματος απαιτεί την αποτελεσματική διαχείριση του έργου και την τήρηση του χρονοδιαγράμματος. Περιλαμβάνει τον καθορισμό φάσεων ανάπτυξης, τον καθορισμό προθεσμιών και την παρακολούθηση της προόδου του έργου.

### 3.5.1 Φάσεις Ανάπτυξης

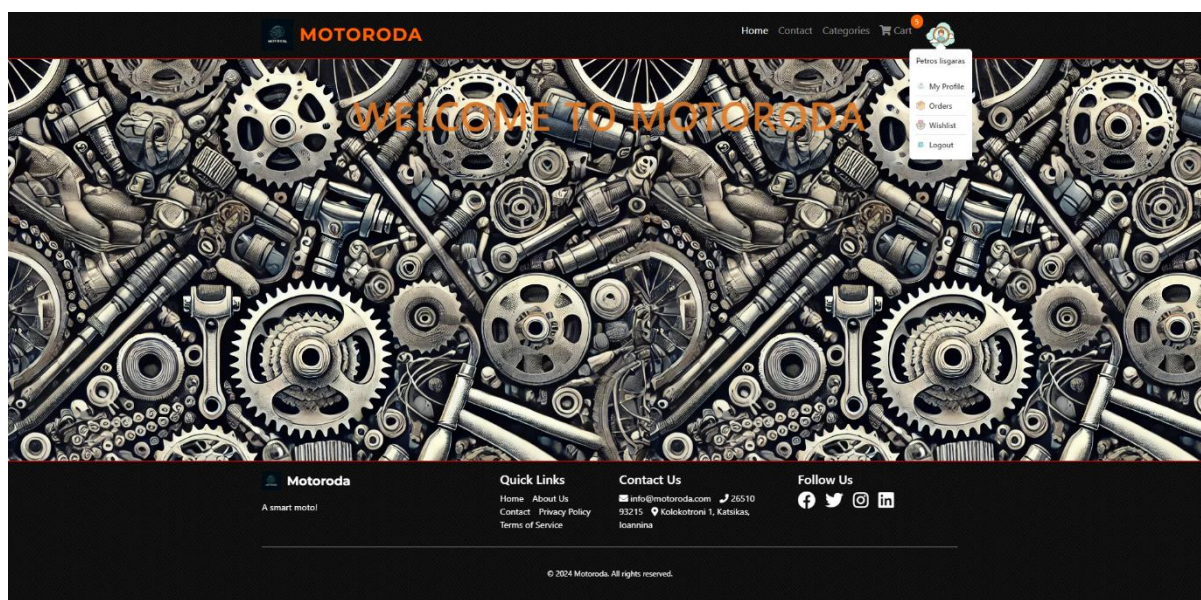
- **Συλλογή Απαιτήσεων:** Καθορισμός των αναγκών και των προσδοκιών των χρηστών.
- **Σχεδιασμός:** Σχεδιασμός της αρχιτεκτονικής, της βάσης δεδομένων και της διεπαφής χρήστη.
- **Υλοποίηση:** Ανάπτυξη των λειτουργιών και της λογικής του συστήματος.
- **Δοκιμές:** Έλεγχος του συστήματος για την εξασφάλιση της ορθής λειτουργίας.

**Ανάπτυξη και Συντήρηση:** Ανάπτυξη του συστήματος και συνεχής υποστήριξη και βελτιώσεις

## Κεφάλαιο 4<sup>ο</sup>

### Παρουσίαση Ηλεκτρονικού Καταστήματος

Το ηλεκτρονικό κατάστημα δημιουργήθηκε τοπικά στον υπολογιστή μας, παρέχοντας μια ασφαλή και ελεγχόμενη πλατφόρμα ανάπτυξης και δοκιμών. Στο πλαίσιο αυτής της εργασίας, θα πραγματοποιηθεί μια λεπτομερή πλοήγηση του καταστήματος, εστιάζοντας στη φιλική προς τον χρήστη σχεδίαση και τη λειτουργικότητα της ιστοσελίδας, ανεξαρτήτως του backend που χρησιμοποιούμε. Η αρχική σελίδα είναι καθαρή και οργανωμένη, προσφέροντας εύκολη πλοήγηση στις κατηγορίες προϊόντων. Η σελίδα κάθε προϊόντος παρέχει λεπτομερείς πληροφορίες. Η διαδικασία αγοράς είναι απλοποιημένη και ασφαλής.

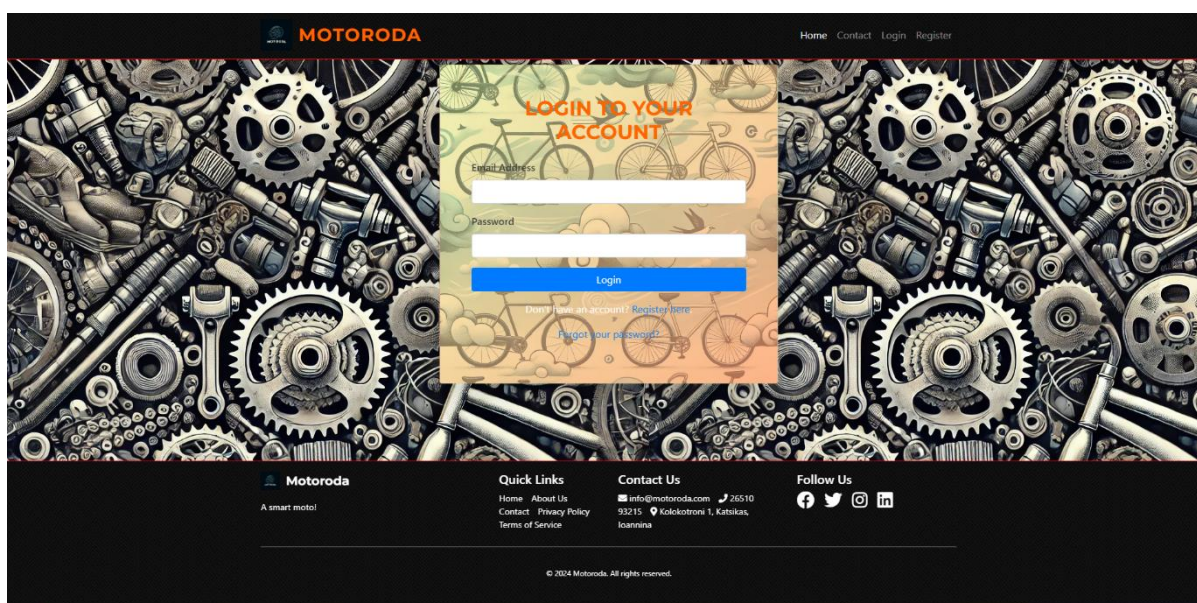


Εικόνα 4.1 Αρχική σελίδα

Στην παραπάνω εικόνα βλέπουμε την αρχική σελίδα του ηλεκτρονικού καταστήματος. Το πάνω μέρος της σελίδας αποτελείται από το header το οποίο περιλαμβάνει το λογότυπο της εταιρείας μας, μια σελίδα επικοινωνίας(Contact). Επιπλέον βλέπουμε την καρτέλα (Login) που παρέχει την δυνατότητα σύνδεσης του χρήστη στο ηλεκτρονικό κατάστημα, αλλά και την καρτέλα εγγραφής του νέου χρήστη (Register). Ο χρήστης όταν συνδεθεί έχει την δυνατότητα να περιηγηθεί στις διάφορες κατηγορίες. Επίσης μπορεί να δει το καλάθι με τα προϊόντα του, να δει τις παραγγελίες του καθώς επίσης και την λίστα των επιθυμιών του. Το κάτω μέρος της σελίδας αποτελείται από το footer το οποίο περιλαμβάνει πληροφορίες για την εταιρεία μας. Επίσης υπάρχουν και σύνδεσμοι για να μπορεί ο χρήστης να μας βρεί στα μέσα κοινωνικής δικτύωσης.

## 4.1 Πλοήγηση ως πελάτης

Ως πελάτης στο ηλεκτρονικό κατάστημα, η εμπειρία πλοήγησης είναι σχεδιασμένη να είναι απλή, ευχάριστη και αποτελεσματική. Κάθε σελίδα προϊόντος προσφέρει αναλυτικές πληροφορίες. Η διαδικασία αγοράς είναι απλοποιημένη και ασφαλής, διασφαλίζοντας ότι η εμπειρία στο ηλεκτρονικό κατάστημα είναι απόλυτα ικανοποιητική. Στο σημείο αυτό μπορούμε να κάνουμε σύνδεση στο κατάστημα. Πατώντας Login ο χρήστης μεταφέρεται αμέσως στην σελίδα Login όπου του ζητείτε το email και ο κωδικός του.



Εικόνα 4. 2 Σύνδεση χρήστη

Αν δεν έχει λογαριασμό και επιθυμεί να εγγραφεί, μέσω της σελίδας(register) εμφανίζεται μια φόρμα υποβολής στοιχείων, η οποία ζητάει από το χρήστη να βάλει το email, έναν κρυφό κωδικό καθώς το όνομα και το επώνυμο του. Αν ο χρήστης δεν συμπληρώσει όλα τα πεδία, θα εμφανιστεί ένα μήνυμα σφάλματος υποδεικνύοντάς του τι είναι λάθος. Αν τα πεδία συμπληρωθούν σωστά, ο λογαριασμός δημιουργείται και ο χρήστης μεταφέρεται στην σελίδα Login για να βάλει τα στοιχεία του. Ολοκληρώνοντας την διαδικασία ο χρήστης εισέρχεται στο ηλεκτρονικό κατάστημα συμπληρώνοντας τα στοιχεία του και πατώντας Login.

**MOTORODA** Home Contact Login Register

### CREATE AN ACCOUNT

First Name  Last Name

Email Address

Password

Confirm Password

[Forgot your password?](#) [Login here](#)

**Register**

© 2024 Motoroda. All rights reserved.

Εικόνα 4. 3 Εγγραφή χρήστη

**MOTORODA** Home Contact Categories Cart

### CONTACT US

Name

Email

Message

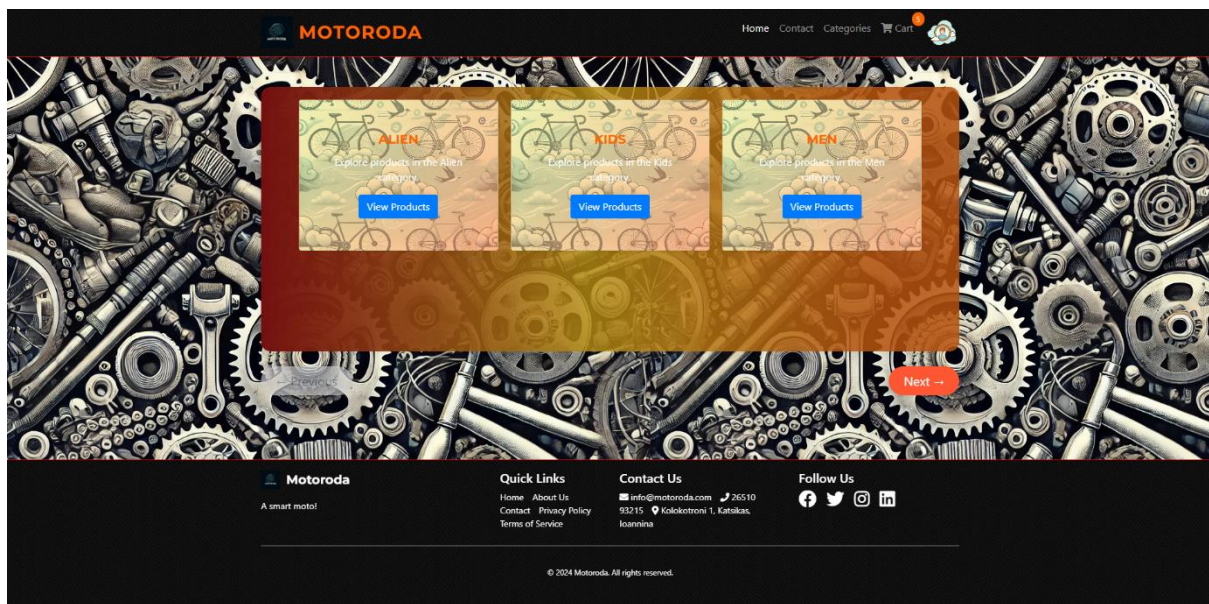
**Send Message**

[Back to Login](#)

© 2024 Motoroda. All rights reserved.

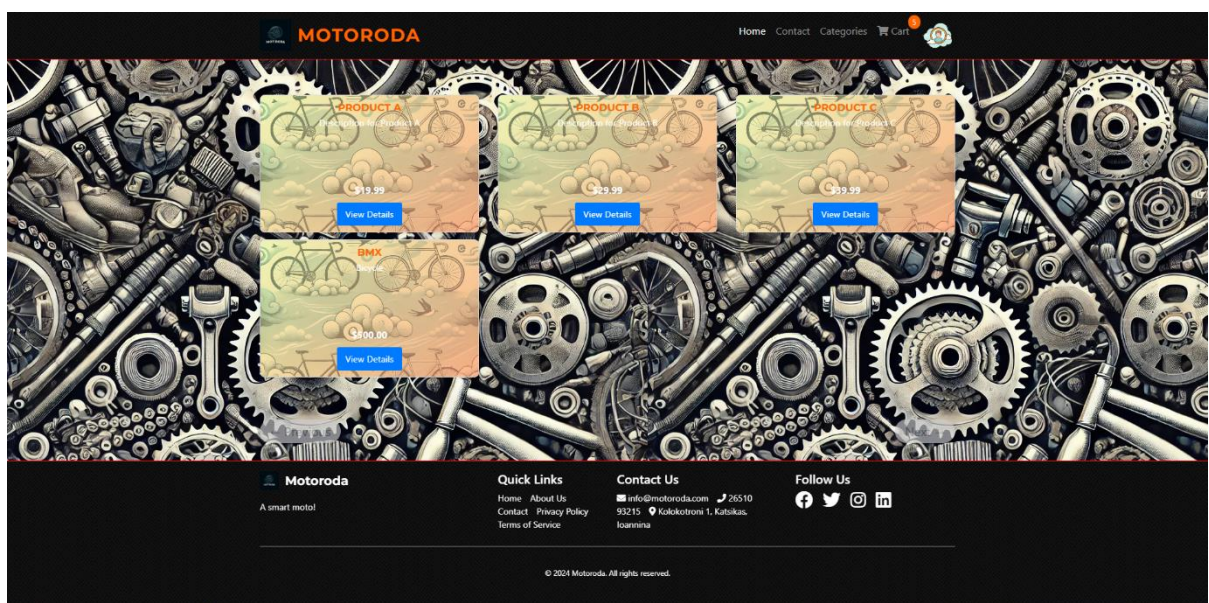
Εικόνα 4 .4 Επικοινωνία

Από την επιλογή “Επικοινωνία”, ο χρήστης έχει τη δυνατότητα να έρθει σε επαφή με την εταιρεία εύκολα και γρήγορα. Οι πελάτες μπορούν να στείλουν τα ερωτήματά τους και να λάβουν άμεσες απαντήσεις μέσω email ή να βρουν τα τηλέφωνα επικοινωνίας για άμεση τηλεφωνική υποστήριξη. Με αυτόν τον τρόπο, διασφαλίζουμε ότι κάθε χρήστης μπορεί να επιλέξει τον πιο βολικό τρόπο επικοινωνίας, λαμβάνοντας την υποστήριξη που χρειάζεται άμεσα και αποτελεσματικά.

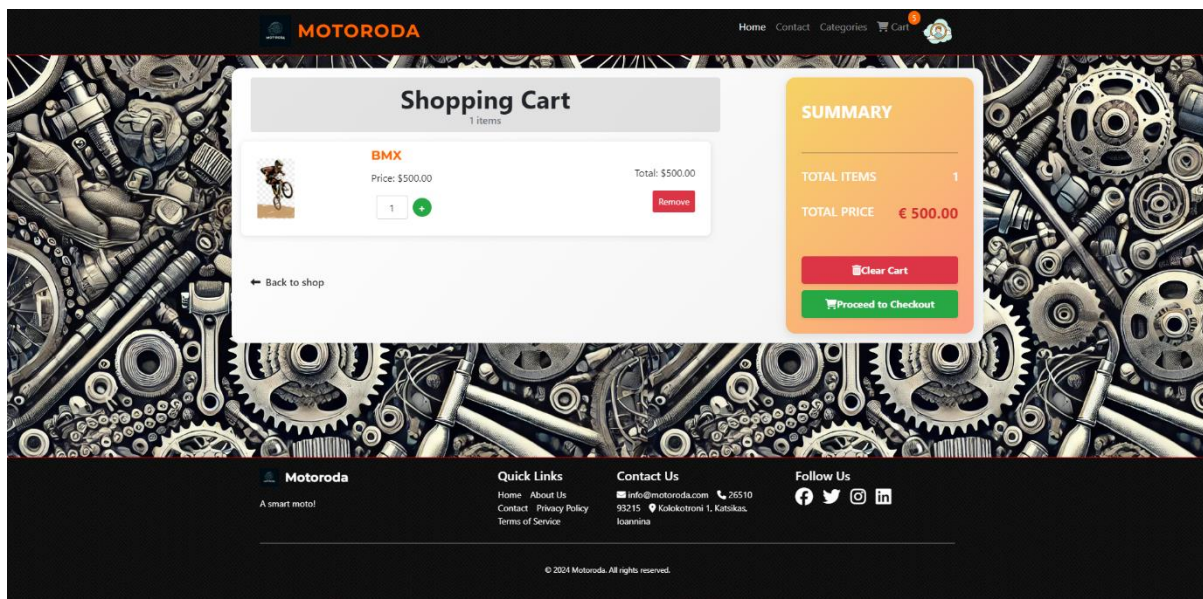


Εικόνα 4. 5 Κατηγορίες

Από την επιλογή “Κατηγορίες” στην μπάρα του μενού, ο χρήστης μπορεί να πλοηγηθεί στις διάφορες κατηγορίες προϊόντων που υπάρχουν στο ηλεκτρονικό κατάστημα. Αυτή η επιλογή προσφέρει μια οργανωμένη και εύκολη στην χρήση πλοήγηση, επιτρέποντας στους πελάτες να βρίσκουν γρήγορα τα προϊόντα που τους ενδιαφέρουν. Κάθε κατηγορία ανοίγει μια νέα σελίδα που παρουσιάζει τα διαθέσιμα προϊόντα με περιγραφές, διευκολύνοντας ακόμη περισσότερο την εύρεση του επιθυμητού προϊόντος. Μέσω αυτής της λειτουργίας, οι χρήστες μπορούν να περιηγηθούν γρήγορα και αποτελεσματικά στα προϊόντα, απολαμβάνοντας μια ευχάριστη και άνετη εμπειρία αγορών.

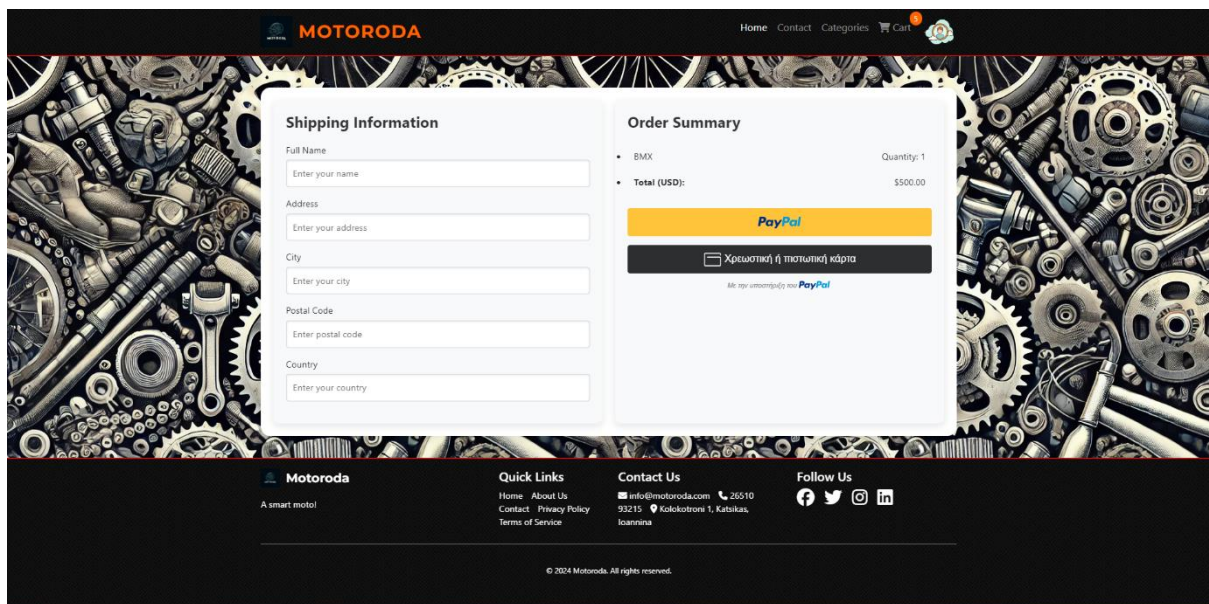


Εικόνα 4.6 Προϊόντα



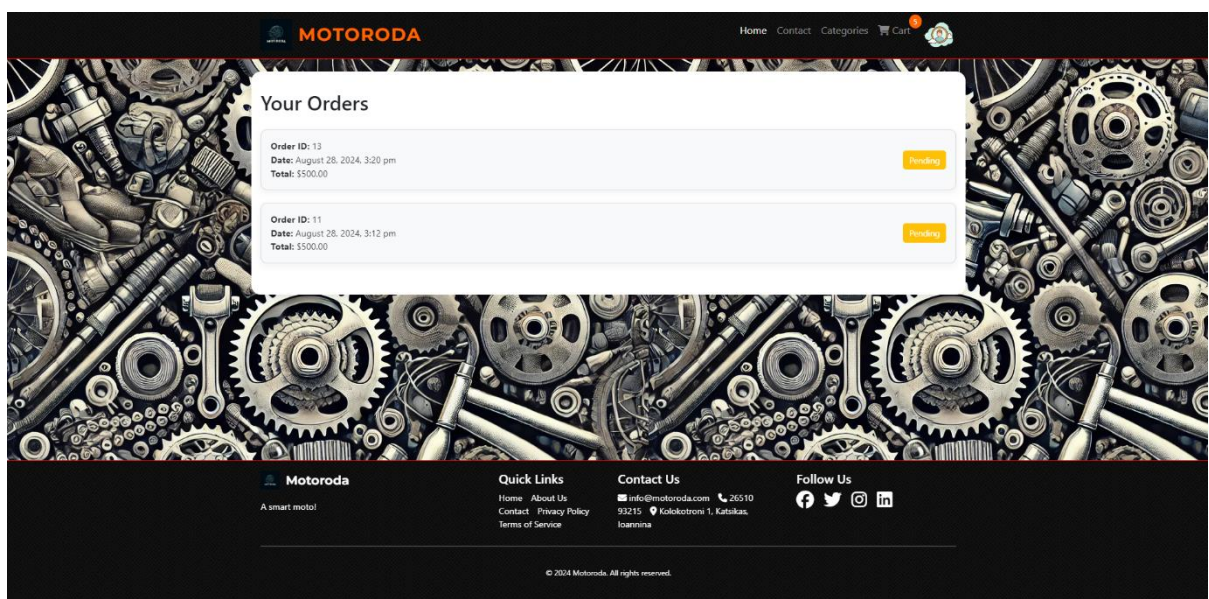
Εικόνα 4.7 Καλάθι αγορών

Από την επιλογή “Καλάθι Αγορών” στην μπάρα του μενού μας, ο χρήστης μπορεί να ελέγξει και να διαχειριστεί τα προϊόντα που έχει επιλέξει για αγορά. Αυτή η επιλογή προσφέρει μια σαφή και εύχρηστη επισκόπηση των επιλεγμένων προϊόντων, επιτρέποντας στους πελάτες να δουν τα είδη που έχουν προσθέσει στο καλάθι τους, τις ποσότητες, τις τιμές και το συνολικό κόστος. Οι χρήστες έχουν τη δυνατότητα να ενημερώνουν τις ποσότητες, να αφαιρούν προϊόντα ή να προσθέτουν νέα, διασφαλίζοντας ότι το καλάθι τους ανταποκρίνεται πλήρως στις ανάγκες τους. Επιπλέον, από την επιλογή “Proceed to Checkout” μπορούν να προχωρήσουν στη διαδικασία ολοκλήρωσης της αγοράς με μερικά απλά βήματα, εισάγοντας τα στοιχεία αποστολής και πληρωμής. Η διαδικασία είναι σχεδιασμένη να είναι γρήγορη και ασφαλής, εξασφαλίζοντας μια ομαλή και ευχάριστη εμπειρία αγορών.



Εικόνα 4.8 “Checkout”

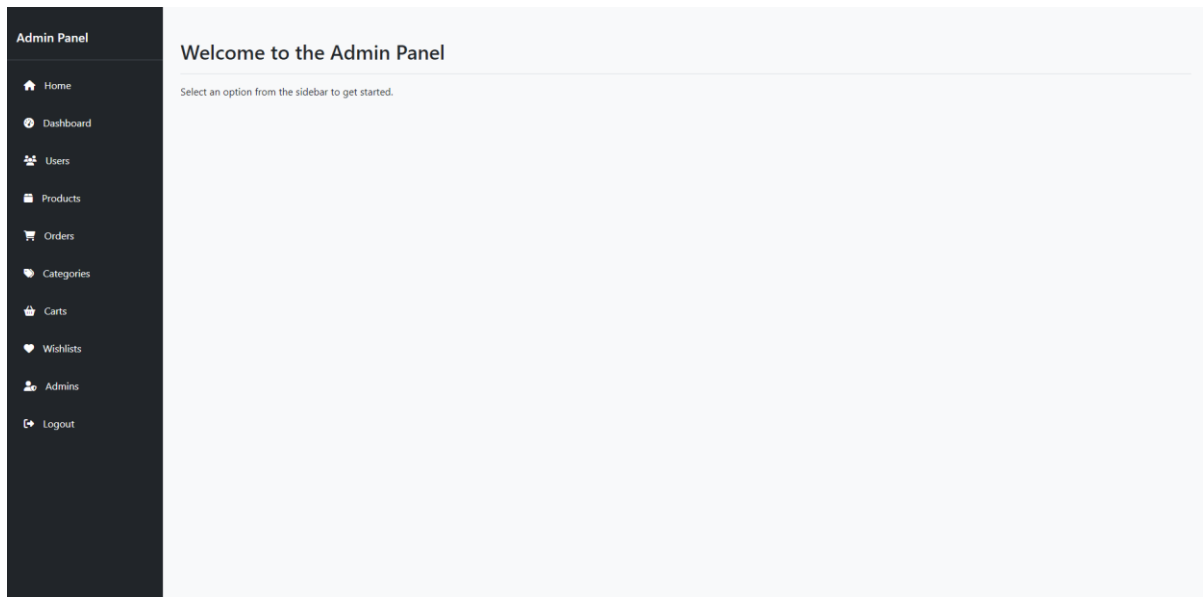
Στην σελίδα checkout του καταστήματος, ο χρήστης μπορεί να εκπονήσει την αγορά του, συμπληρώνοντας τα απαραίτητα στοιχεία. Μέσω του τρόπου πληρωμής PayPal μπορεί με ασφάλεια να πληρώσει για τα προϊόντα που επιθυμεί να αγοράσει.



Εικόνα 4.9 Παραγγελίες

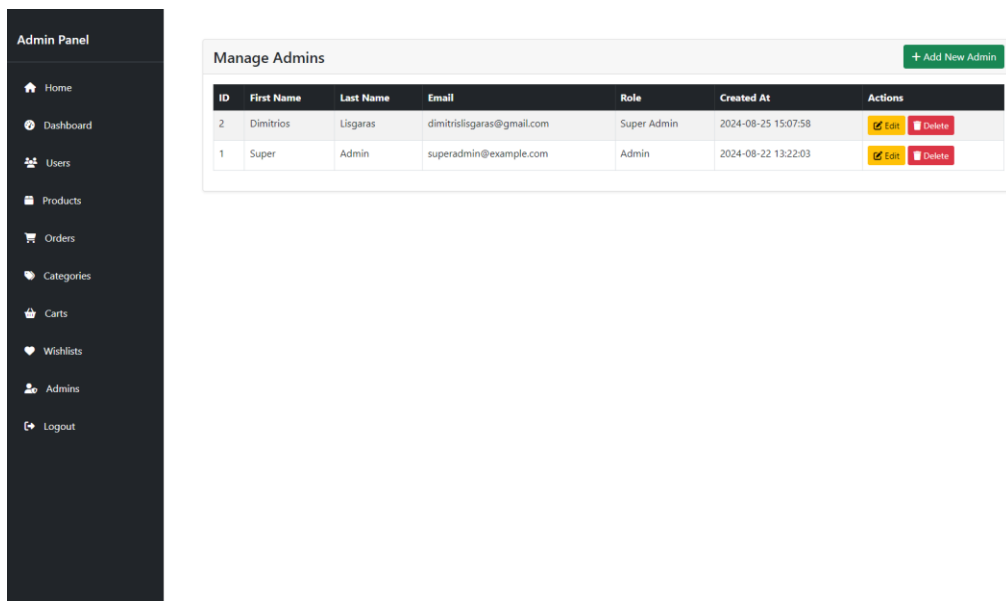
Στην σελίδα παραγγελίες του καταστήματος, ο χρήστης μπορεί να παρακολουθεί την κατάσταση της παραγγελίας του. Παρέχονται πληροφορίες όπως το σύνολο της παραγγελίας, το πότε αυτή δημιουργήθηκε καθώς επίσης σε τι κατάσταση βρίσκεται.

## 4.2 Πλοήγηση ως διαχειριστής



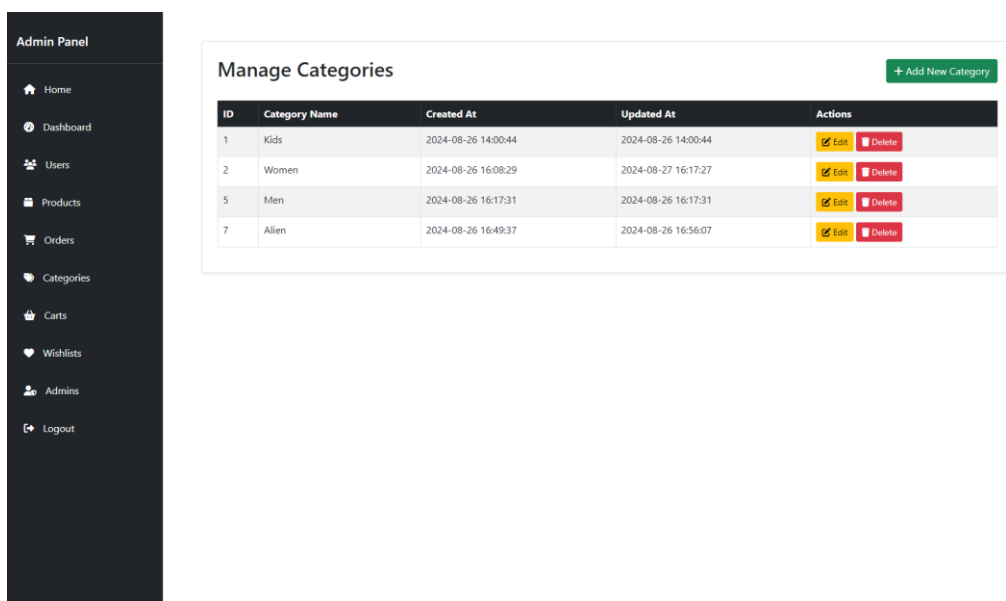
Εικόνα 4.10 Admin Panel

Από το “Admin Panel”, οι διαχειριστές του ηλεκτρονικού καταστήματος μπορούν να έχουν πλήρη έλεγχο και εποπτεία της λειτουργίας του. Το admin panel προσφέρει μια ολοκληρωμένη και εύχρηστη διεπαφή, μέσω της οποίας οι διαχειριστές μπορούν να διαχειρίζονται τα προϊόντα, τις κατηγορίες, τις παραγγελίες και τους χρήστες. Οι διαχειριστές έχουν τη δυνατότητα να προσθέτουν, να ενημερώνουν και να διαγράφουν προϊόντα.



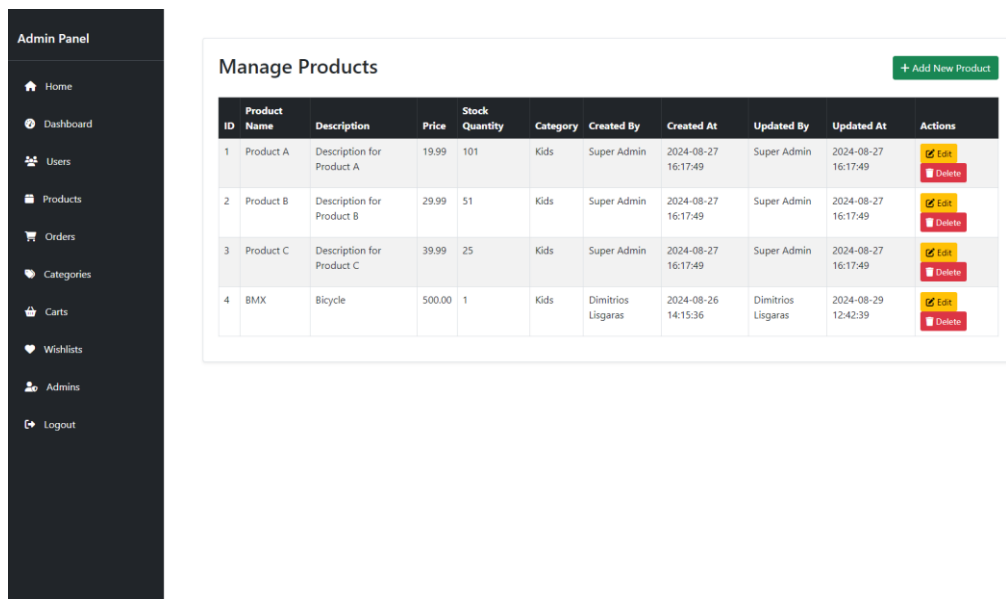
Εικόνα 4.11 Διαχειριστές

Από την επιλογή “Διαχειριστές” στο μενού μας, οι υπεύθυνοι του ηλεκτρονικού καταστήματος μπορούν να διαχειρίζονται αποτελεσματικά τους χρήστες με δικαιώματα διαχειριστή. Η ενότητα αυτή επιτρέπει την προσθήκη λογαριασμών διαχειριστών.



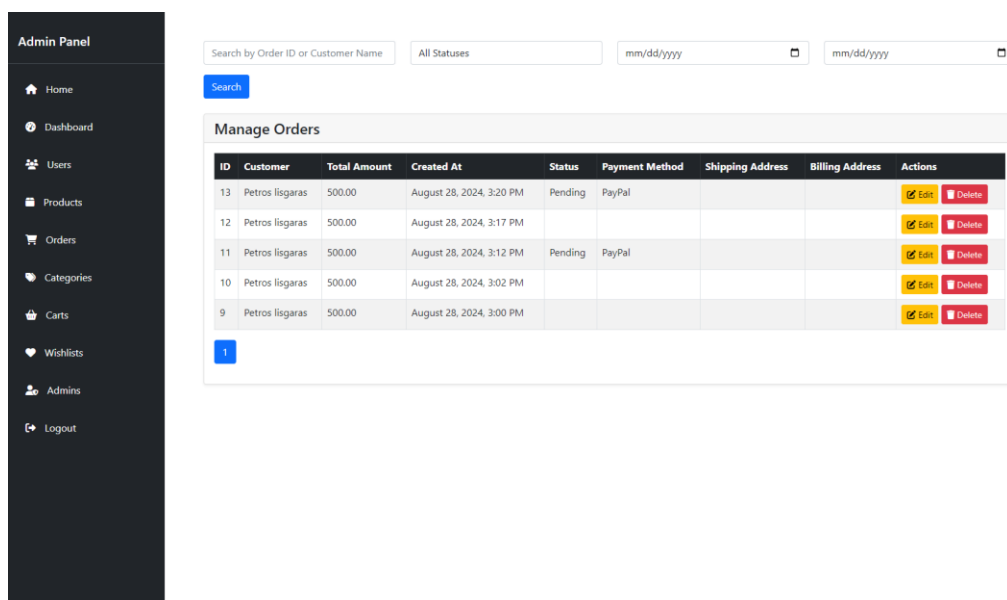
Εικόνα 4.12 Διαχειριστές-Κατηγορίες

Από την επιλογή “Categories” στο μενού, οι υπεύθυνοι του ηλεκτρονικού καταστήματος μπορούν να διαχειρίζονται αποτελεσματικά τις κατηγορίες των προϊόντων. Η ενότητα αυτή επιτρέπει την προσθήκη, επεξεργασία και διαγραφή κατηγοριών, διευκολύνοντας την οργάνωση και την ταξινόμηση. Οι διαχειριστές μπορούν να δημιουργούν νέες κατηγορίες, να ενημερώνουν υπάρχουσες και να διασφαλίζουν ότι κάθε προϊόν τοποθετείται στη σωστή κατηγορία για εύκολη πλοήγηση από τους χρήστες. Με αυτόν τον τρόπο, η επιλογή “Κατηγορίες” συμβάλλει στην αποδοτική διαχείριση και παρουσίαση των προϊόντων του καταστήματος.



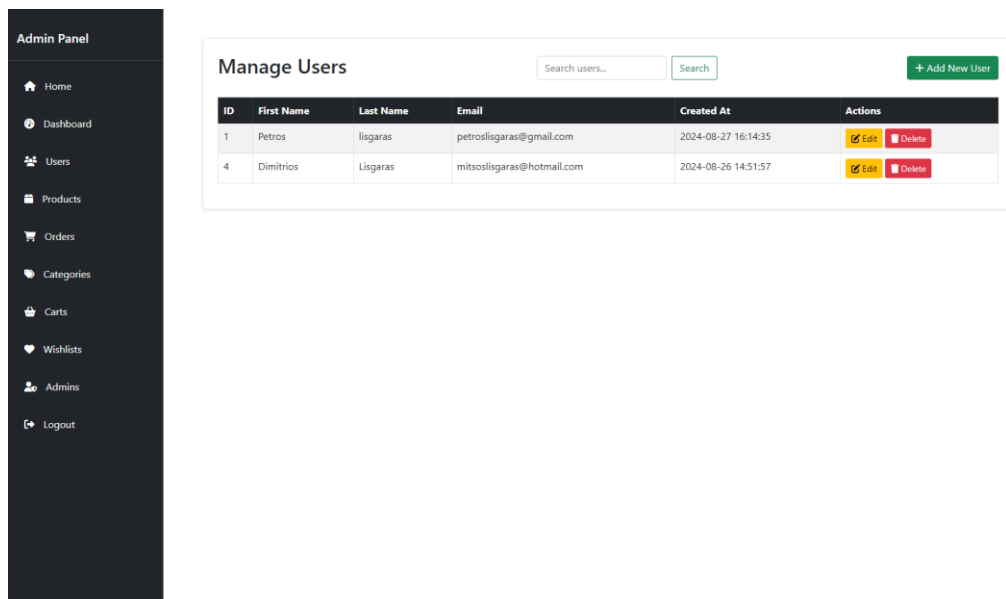
Εικόνα 4.13 Διαχειριστές-Προϊόντα

Από την επιλογή “Products” στο μενού, οι υπεύθυνοι του ηλεκτρονικού καταστήματος μπορούν να διαχειρίζονται αποτελεσματικά τα προϊόντα. Η ενότητα αυτή επιτρέπει την προσθήκη, επεξεργασία και διαγραφή προϊόντων, εξασφαλίζοντας ότι οι πληροφορίες είναι πάντα ενημερωμένες και ακριβείς. Με αυτόν τον τρόπο, γίνεται αποδοτική η διαχείριση και παρουσίαση των προϊόντων, ενισχύοντας τη συνολική εμπειρία των πελατών.



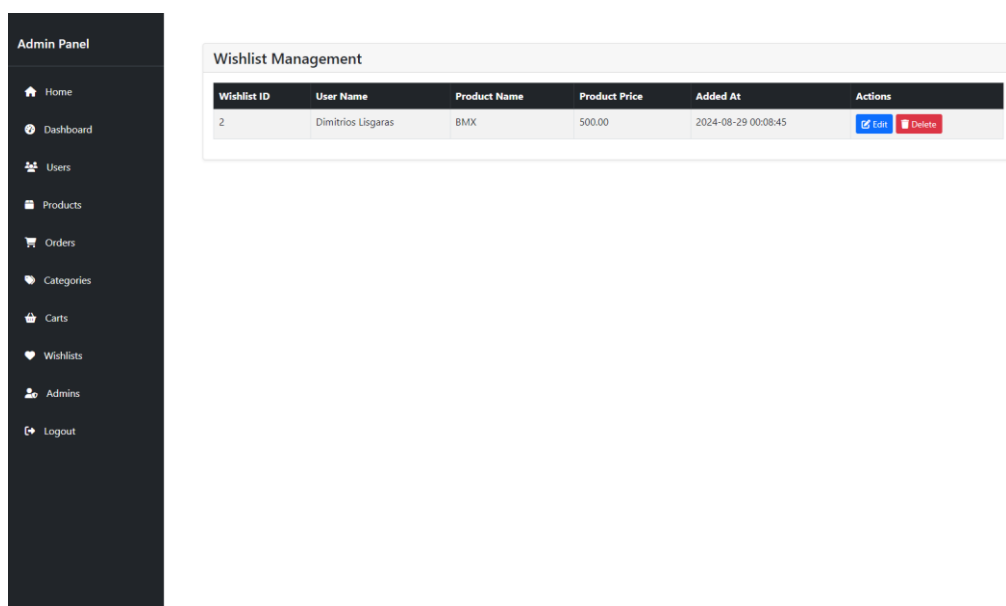
Εικόνα 4.14 Διαχειριστές-Παραγγελίες

Από την επιλογή “Orders” στο μενού, οι υπεύθυνοι του ηλεκτρονικού καταστήματος μπορούν να διαχειρίζονται αποτελεσματικά τις παραγγελίες. Η ενότητα αυτή επιτρέπει την προσθήκη, επεξεργασία και διαγραφή παραγγελιών, εξασφαλίζοντας ότι οι πληροφορίες είναι πάντα ενημερωμένες και ακριβείς.



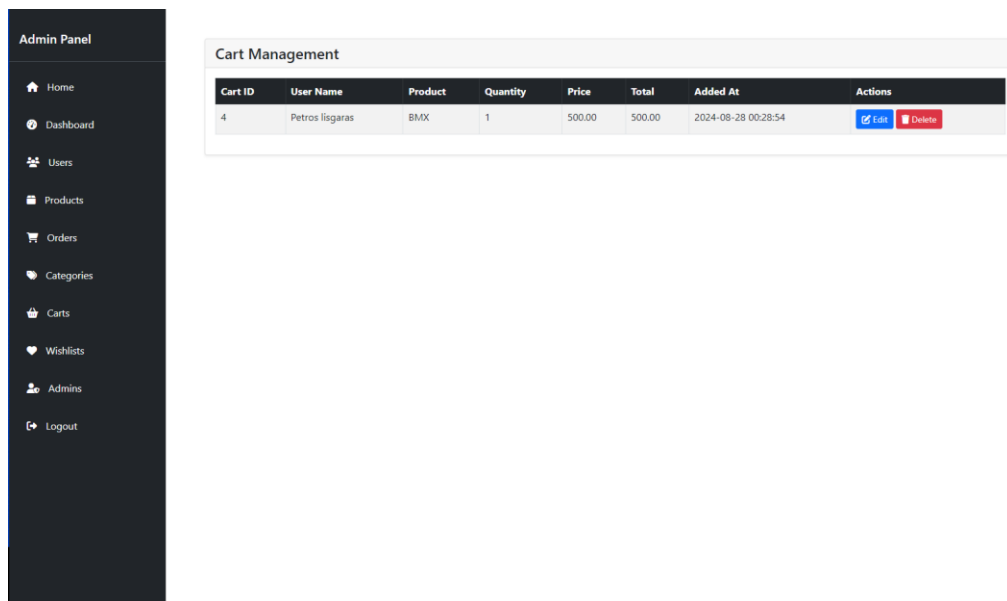
Εικόνα 4.15 Διαχειριστές-Χρήστες.

Από την επιλογή “Users” στο μενού, οι υπεύθυνοι του ηλεκτρονικού καταστήματος μπορούν να διαχειρίζονται αποτελεσματικά τους χρήστες. Η ενότητα αυτή επιτρέπει την προσθήκη, επεξεργασία και διαγραφή χρηστών.



Εικόνα 4.16 Διαχειριστές-Λίστα Επιθυμιών.

Από την επιλογή “WishList” στο μενού, οι υπεύθυνοι του ηλεκτρονικού καταστήματος μπορούν να διαχειρίζονται αποτελεσματικά τις παραγγελίες. Η ενότητα αυτή επιτρέπει την επεξεργασία και διαγραφή «λιστών επιθυμιών».



Εικόνα 4.17 Διαχειριστές-Καλάθι αγορών.

Από την επιλογή “Cart” στο μενού, οι υπεύθυνοι του ηλεκτρονικού καταστήματος μπορούν να διαχειρίζονται αποτελεσματικά τα καλάθια των χρηστών. Η ενότητα αυτή επιτρέπει την επεξεργασία και διαγραφή καλαθιών.

## Συμπεράσματα

Η ανάπτυξη της ηλεκτρονικής εφαρμογής "MotoRoda" για την πώληση μεταχειρισμένων ποδηλάτων αποδεικνύει τη δυνατότητα δημιουργίας μιας αξιόπιστης και λειτουργικής πλατφόρμας που διευκολύνει τις αγοραπωλησίες μεταχειρισμένων προϊόντων. Η εφαρμογή αξιοποιεί σύγχρονες τεχνολογίες, όπως PHP, MySQL, HTML, CSS και JavaScript, για την υλοποίηση μιας ολοκληρωμένης λύσης που προσφέρει ευκολία στην χρήση και ασφάλεια στις συναλλαγές.

Η εφαρμογή επιτυγχάνει τους στόχους της, παρέχοντας δυνατότητες όπως η δημιουργία και διαχείριση λογαριασμών χρηστών, η προβολή και διαχείριση προϊόντων, η χρήση καλαθιού αγορών και η ολοκλήρωση παραγγελιών.

Μέσω της ανάπτυξης της ηλεκτρονικής εφαρμογής, διαπιστώθηκε ότι οι ηλεκτρονικές πλατφόρμες μπορούν να προσφέρουν οικονομικά προσιτές λύσεις στους καταναλωτές.

Για μελλοντικές επεκτάσεις, προτείνονται η ενσωμάτωση περισσότερων τρόπων πληρωμής, η ανάπτυξη εφαρμογών για κινητές συσκευές, καθώς και η προσθήκη λειτουργιών κοινωνικής δικτύωσης για τη δημιουργία μιας κοινότητας χρηστών. Επίσης, η προσθήκη των αλγορίθμων αναζήτησης και σύστασης προϊόντων μπορεί να ενισχύσει την εμπειρία του χρήστη και να αυξήσει τις πωλήσεις.

Συνοψίζοντας, η εφαρμογή που αναπτύχθηκε ανταποκρίνεται στις ανάγκες της σύγχρονης αγοράς και προάγει την περιβαλλοντική συνείδηση μέσω της επαναχρησιμοποίησης ποδηλάτων.

## Αναφορές

- Barker, D. (2010). *Web Content Management*. O'Reilly Media Inc.
- Carey, P. (2001). *The internet and e-commerce*. London: Thorogood.
- Delisle, M. (2012). *Mastering phpMyAdmin 3.4 for Effective MySQL*. Birmingham: Packt Publishing Ltd.
- DuBois, P. (2009). *MySQL*. Pearson.
- DUCKETT, J. (2011). *HTML & CSS Design and Build Websites*. Indianapolis: John Wiley & Sons.
- Gourley, D., & Totty, B. (2002). *HTTP: The Definitive Guide*. Sebastopol: O'Reilly Media.
- Kurose, J. F., & Ross, K. (2017). *Computer Networking*. Essex: Pearson.
- Laudon, K. C., & Guercio Traver, C. (2023). *E-commerce business, technology, society*. Pearson.
- Lockhart, J. (2015). *Modern PHP*. O'Reilly Media.
- Soni, R. (2016). *Nginx: From Beginner to Pro*. Kolkata: Apress.
- Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development*. Pearson.
- Λιακέας, Γ. (2015). *Η γλώσσα JavaScript*. Αθήνα: Κλειδάριθμος.

## Παράρτημα: Ανάλυση κώδικα back-end

Η ανάλυση κώδικα back-end είναι μια κρίσιμη διαδικασία για την εξασφάλιση της σωστής λειτουργίας και της βελτιστοποίησης των εφαρμογών web. Αυτή η διαδικασία περιλαμβάνει την αξιολόγηση της αρχιτεκτονικής του κώδικα, τη διαχείριση της βάσης δεδομένων, την απόδοση των διακομιστών, καθώς και την ασφάλεια και την επεκτασιμότητα του συστήματος. Η ανάλυση κώδικα back-end συμβάλλει στην αναγνώριση και διόρθωση αδυναμιών που θα μπορούσαν να επηρεάσουν την ασφάλεια και την απόδοση της εφαρμογής, διασφαλίζοντας έτσι την παροχή αξιόπιστων και αποδοτικών υπηρεσιών στους χρήστες.

### Το αρχείο Header.php

```
<?php

require_once '../config/config.php';

// Start session if not already started
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Favicon Links -->

    <link type="image/png" sizes="96x96" rel="icon" href="<?php echo BASE_URL;
?>/assets/images/faviconBike.png">

    <!-- Font -->

    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@700&display=swap">

    <!-- Bootstrap CSS -->

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
```

```

integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="<?php echo BASE_URL;
?>/assets/css/style.css">

    <script src="https://kit.fontawesome.com/5c5946fe44.js" crossorigin="anonymous"></script>

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
rel="stylesheet">

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">

    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>

    <!-- AJAX -->

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

    <title>Motoroda</title>
</head>

<body>

    <!-- Navigation Bar -->

    <nav class="navbar navbar-expand-lg navbar-dark sticky-top">

        <div class="container">

            <!-- Logo -->

            <a class="navbar-brand d-flex align-items-center" href="<?php echo BASE_URL;
?>/public" aria-label="Motoroda Home">

                <span class="logo-text">Motoroda</span>

            </a>

            <!-- Navbar Links -->

            <div class="collapse navbar-collapse" id="navbarNav">

                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="<?php echo
BASE_URL; ?>/public">Home</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="<?php echo BASE_URL;
?>/public/contact.php">Contact</a>

                    </li>

```

```

        <!-- Categories Link, visible only when user is logged in -->

        <?php if (isset($_SESSION['username'])) : ?>

            <li class="nav-item">

                <a class="nav-link" href="<?php echo BASE_URL;
?>/categories/index.php">Categories</a>

            </li>

        <!-- Cart Link with Badge -->

        <li class="nav-item cart-icon">

            <a class="nav-link" href="<?php echo BASE_URL;
?>/shopping/cart.php">

                <i class="fas fa-shopping-cart"></i> Cart

                <span class="cart-badge">5</span>

            </a>

        </li>

        <!-- User Account Dropdown -->

        <div class="action">

            <div class="profile" onclick="menuToggle();">

            </div>

            <div class="menu">

                <h3><?php echo $_SESSION['username']; ?></h3>

                <ul>

                    <li>

                        <a href="#">My Profile</a>

                    </li>

                    <li>

                        <a href="<?php echo BASE_URL;
?>/shopping/order.php?id=<?php echo $_SESSION['user_id']; ?>">Orders</a>

                    </li>

                    <li>

                        <a href="<?php echo BASE_URL;
?>/shopping/wishlist.php?id=<?php echo $_SESSION['user_id']; ?>">Wishlist</a>

                    </li>

                    <li>

                        <a href="<?php echo BASE_URL; ?>/auth/logout.php">Logout</a>

```

```

        </li>

    </ul>

</div>

</div>

<?php else : ?>

    <!-- Login and Register Links -->

    <li class="nav-item">

        <a class="nav-link" href="<?php echo BASE_URL;
?>/auth/login.php">Login</a>

    </li>

    <li class="nav-item">

        <a class="nav-link" href="<?php echo BASE_URL;
?>/auth/register.php">Register</a>

    </li>

    <?php endif; ?>

</ul>

</div>

</div>

</nav>

<!-- Main Content Container -->

<div class="container"></div>

<script>

    function menuToggle() {

        const toggleMenu = document.querySelector('.menu');

        toggleMenu.classList.toggle('active');

    }

</script>

</body>

</html>

```

## To αρχείο footer.php

```
<!-- Footer -->

<footer class="footer mt-auto py-3 bg-dark text-white">

    <div class="container">

        <div class="row">

            <!-- Branding Section -->

            <div class="col-md-4">

                <a class="footer-brand d-flex align-items-center" href="<?php echo BASE_URL;
?>/public">

                    <span class="footer-brand-text">Motoroda</span>

                </a>

                <p class="mt-3">A smart moto!</p>

            </div>

            <!-- Navigation Links -->

            <div class="col-md-2">

                <h5>Quick Links</h5>

                <ul class="list-unstyled">

                    <li><a href="<?php echo BASE_URL; ?>/public" class="text-
white">Home</a></li>

                    <li><a href="<?php echo BASE_URL; ?>/about.php" class="text-white">About
Us</a></li>

                    <li><a href="<?php echo BASE_URL; ?>/contact.php" class="text-
white">Contact</a></li>

                    <li><a href="<?php echo BASE_URL; ?>/privacy-policy.php" class="text-
white">Privacy Policy</a></li>

                    <li><a href="<?php echo BASE_URL; ?>/terms.php" class="text-white">Terms
of Service</a></li>

                </ul>

            </div>

            <!-- Contact Information -->

            <div class="col-md-3">

                <h5>Contact Us</h5>

                <ul class="list-unstyled">

                    <li><i class="fas fa-envelope"></i> info@motoroda.com</li>

                    <li><i class="fas fa-phone"></i> 26510 93215</li>

                </ul>

            </div>

        </div>

    </div>

</footer>
```

```

                <li><i class="fas fa-map-marker-alt"></i> Kolokotroni 1, Katsikas,
Ioannina</li>

            </ul>

        </div>

        <!-- Social Media Links -->

        <div class="col-md-3">

            <h5>Follow Us</h5>

            <div class="d-flex">

                <a href="#" class="text-white me-3"><i class="fab fa-facebook fa-
2x"></i></a>

                <a href="#" class="text-white me-3"><i class="fab fa-twitter fa-
2x"></i></a>

                <a href="#" class="text-white me-3"><i class="fab fa-instagram fa-
2x"></i></a>

                <a href="#" class="text-white"><i class="fab fa-linkedin fa-2x"></i></a>

            </div>

        </div>

    </div>

</div>

<hr class="bg-white mt-4">

<!-- Legal Information -->

<div class="text-center py-3">

    <p>&copy; 2024 Motoroda. All rights reserved.</p>

</div>

</div>

</footer>

<!-- JavaScript to auto-hide the alert and adjust form position -->

<script>

document.addEventListener('DOMContentLoaded', (event) => {

    const alertElement = document.querySelector('.alert');

    if (alertElement) {

        setTimeout(() => {

            alertElement.style.display = 'none'; // Hide the alert

            document.querySelector('.card').style.marginTop = '0'; // Adjust the form
position

        }, 5000); // Alert will disappear after 5 seconds

    }

}

```

```
});  
  
</script>  
  
<!-- Bootstrap JS and dependencies -->  
  
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"  
integrity="sha384-JVR1DA1Ac1B/Cz5HBZrT/P4f0nX0BBBcMdrDs4ZIjzF4BRHPfEJoZpVNE+k7Sk36"  
crossorigin="anonymous"></script>  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"  
integrity="sha384-cVKIPhGJCUZZ2zFL1qwn2FbCBU7zO8X1H91f1QrX3zF4AOhT0c4tjPviApub0GM"  
crossorigin="anonymous"></script></body></html>
```

## To αρχείο Product.php

```
<?php

require_once '../config/config.php';

// Start session if not already started
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

// Check if product ID is provided
if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
    die("Invalid product ID.");
}

$productId = $_GET['id'];

// Fetch product details
try {
    $stmt = $conn->prepare("SELECT * FROM products WHERE product_id = :product_id");
    $stmt->bindParam(':product_id', $productId);
    $stmt->execute();
    $product = $stmt->fetch(PDO::FETCH_ASSOC);

    if (!$product) {
        die("Product not found.");
    }
} catch (Exception $e) {
    die("Error fetching product: " . $e->getMessage());
}

?>

<?php include '../templates/header.php'; ?>

<!-- Link to the new product.css for enhanced styling -->
<link rel="stylesheet" href="../assets/css/product.css">

<div class="container mt-5 product-container">
```

```

<div class="row">

    <!-- Product Image -->

    <div class="col-md-6 product-image">

        "

    </div>

    <!-- Product Details -->

    <div class="col-md-6 product-details">

        <h3 class="product-name"><?php echo htmlspecialchars($product['name']); ?></h3>

        <p class="product-price">$<?php echo number_format($product['price'], 2); ?></p>

        <p class="product-description"><?php echo
htmlspecialchars($product['description']); ?></p>

        <!-- Stock Availability -->

        <p class="product-stock text-info">Available Stock: <span id="stock-count"><?php
echo $product['stock_quantity']; ?></span></p>

        <!-- Add to Cart Form -->

        <form id="add-to-cart-form">

            <input type="hidden" name="product_id" value="<?php echo
$product['product_id']; ?>">

            <div class="form-group">

                <label for="quantity">Quantity</label>

                <input type="number" name="quantity" id="quantity" class="form-control"
value="1" min="1" max="<?php echo $product['stock_quantity']; ?>">

            </div>

            <button type="button" id="add-to-cart-btn" class="btn btn-success btn-lg mt-
3">Add to Cart</button>

        </form>

        <!-- Add to Wishlist Button -->

        <button type="button" id="add-to-wishlist-btn" class="btn btn-warning btn-lg mt-
3">Add to Wishlist</button>

        <!-- Success or Error Message -->

        <div id="message" class="mt-3"></div>

    </div>

</div>

```

```

<?php include '../templates/footer.php'; ?>

<!-- Include jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>
$(document).ready(function() {
    // Update stock availability dynamically
    $('#quantity').on('input', function() {
        var maxStock = <?php echo $product['stock_quantity']; ?>;
        var selectedQuantity = parseInt($(this).val());

        if (selectedQuantity > maxStock) {
            $(this).val(maxStock);
        }

        $('#stock-count').text(maxStock - selectedQuantity);
    });

    // AJAX request to add item to cart
    $('#add-to-cart-btn').click(function() {
        var productId = $('#input[name="product_id"]').val();
        var quantity = $('#quantity').val();

        $.ajax({
            url: 'add_to_cart.php',
            type: 'POST',
            data: {
                product_id: productId,
                quantity: quantity
            },
            success: function(response) {
                $('#message').html('<div class="alert alert-success">Product added to cart successfully!</div>');
                $('#stock-count').text(<?php echo $product['stock_quantity']; ?> - quantity);
            },
            error: function() {
                $('#message').html('<div class="alert alert-danger">Failed to add product to cart. Please try again.</div>');
            }
        });
    });
});

```

```

        }

    });

});

// AJAX request to add item to wishlist
$('#add-to-wishlist-btn').click(function() {
    var productId = $('#input[name="product_id"]').val();

    $.ajax({
        url: '../shopping/add_to_wishlist.php',
        type: 'POST',
        data: {
            product_id: productId
        },
        success: function(response) {
            $('#message').html('<div class="alert alert-success">Product added to wishlist successfully!</div>');
        },
        error: function() {
            $('#message').html('<div class="alert alert-danger">Failed to add product to wishlist. Please try again.</div>');
        }
    });
});

});
</script>
</body>
</html>

```

## To αρχείο `Category.php`

```
<?php

require_once '../config/config.php';

// Start session if not already started
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

// Check if category ID is provided
if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
    die("Invalid category ID.");
}

$categoryId = $_GET['id'];

// Fetch category details
try {
    $stmt = $conn->prepare("SELECT category_name FROM categories WHERE category_id = :category_id");
    $stmt->bindParam(':category_id', $categoryId);
    $stmt->execute();
    $category = $stmt->fetch(PDO::FETCH_ASSOC);

    if (!$category) {
        die("Category not found.");
    }
} catch (Exception $e) {
    die("Error fetching category: " . $e->getMessage());
}

// Fetch products for the category
try {
    $stmt = $conn->prepare("SELECT * FROM products WHERE category_id = :category_id");
    $stmt->bindParam(':category_id', $categoryId);
    $stmt->execute();
    $products = $stmt->fetchAll(PDO::FETCH_ASSOC);
} catch (Exception $e) {
```

```

        die("Error fetching products: " . $e->getMessage());
    }

    // Pagination logic

    $productsPerPage = 6; // Show 6 products per page

    $totalProducts = count($products);

    $totalPages = ceil($totalProducts / $productsPerPage);

    // Get current page from query parameter, default to 1

    $currentPage = isset($_GET['page']) ? (int)$_GET['page'] : 1;

    $currentPage = max(1, min($currentPage, $totalPages)); // Ensure current page is within valid
    range

    $startIndex = ($currentPage - 1) * $productsPerPage;

    $endIndex = min($startIndex + $productsPerPage, $totalProducts);

    ?>

<?php include '../templates/header.php'; ?>

<div class="container mt-5">

    <h2><?php echo htmlspecialchars($category['category_name']); ?> Products</h2>

    <div class="row product-row"> <!-- Adjusted margin and padding -->

        <!-- Display Products -->

        <?php for ($i = $startIndex; $i < $endIndex; $i++): ?>

            <?php $product = $products[$i]; ?>

            <div class="col-md-4 mb-3"> <!-- Adjusted bottom margin for gap -->

                <div class="card text-center shadow-sm product-card">

                    <div class="card-body d-flex flex-column justify-content-between">

                        <div>

                            <h5 class="card-title"><?php echo
htmlspecialchars($product['name']); ?></h5>

                            <p class="card-text product-description"><?php echo
htmlspecialchars($product['description']); ?></p>

                        </div>

                        <div class="mt-auto">

                            <p class="card-text product-price mb-2"><?php echo
number_format($product['price'], 2); ?></p>

                            <a href="product.php?id=<?php echo $product['product_id']; ?>"
class="btn btn-primary btn-sm">View Details</a>

```

```

        </div>

    </div>

</div>

</div>

<?php endfor; ?>

</div>

<!-- Navigation Buttons -->

<div class="d-flex justify-content-between align-items-center mb-3 mt-3">

    <div>

        <a href="?id=<?php echo $categoryId; ?>&page=<?php echo $currentPage - 1; ?>"

            class="btn custom-prev-btn btn-lg rounded-pill px-3 py-1 shadow-sm <?php echo
$currentPage <= 1 ? 'disabled' : ''; ?>"

            &larr; Previous

        </a>

    </div>

    <div>

        <a href="?id=<?php echo $categoryId; ?>&page=<?php echo $currentPage + 1; ?>"

            class="btn custom-next-btn btn-lg rounded-pill px-3 py-1 shadow-sm <?php echo
$currentPage >= $totalPages ? 'disabled' : ''; ?>"

            Next &rarr;

        </a>

    </div>

</div>

</div>

<!-- Custom CSS for Button and Card Styling -->

<style>

    .custom-prev-btn, .custom-next-btn {

        background-color: #FF5733; /* Custom orange color */

        color: #fff; /* White text */

        border: none; /* Remove border */

        transition: background-color 0.3s ease; /* Smooth transition */

        cursor: pointer; /* Add pointer cursor for clickable buttons */

    }

    .custom-prev-btn:hover, .custom-next-btn:hover {

        background-color: #C70039; /* Darker shade on hover */

    }

</style>

```

```

}

.custom-prev-btn.disabled, .custom-next-btn.disabled {
    background-color: #ccc; /* Gray color for disabled state */
    color: #6c757d; /* Light gray text for disabled state */
    cursor: not-allowed; /* Not allowed cursor for disabled buttons */
    pointer-events: none; /* Disable all click events */
}

/* Adjusted height and spacing for the product row */
.product-row {
    margin-top: 10px; /* Small margin at the top */
    margin-bottom: 0px; /* No bottom margin */
    padding-bottom: 0; /* No padding at the bottom */
}

.product-card {
    background-color: #f8f9fa; /* Background color for product cards */
    border: 1px solid #ddd; /* Border around cards */
    border-radius: 8px; /* Rounded corners */
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Subtle shadow for depth */
    transition: transform 0.2s ease, box-shadow 0.2s ease; /* Smooth transition for hover effects */
    height: 220px; /* Slightly smaller height for the product card */
    padding: 5px; /* Padding inside cards */
    margin-bottom: 35px; /* Increased space below each card for better separation */
}

.product-card:hover {
    transform: translateY(-2px); /* Slightly elevate on hover */
    box-shadow: 0 5px 10px rgba(0, 0, 0, 0.15); /* Deeper shadow on hover */
}

.product-card .card-body {
    padding: 5px; /* Reduced padding inside card body */
    display: flex; /* Make card-body a flex container */
    flex-direction: column; /* Arrange children in a column */
    justify-content: space-between; /* Space between elements */
}

```

```

        height: 100%; /* Fill the entire height of the card */
    }

    .product-card .card-title {
        font-size: 1em; /* Slightly smaller font size for title */
        margin-bottom: 3px; /* Reduced margin below the title */
    }

    .product-card .product-description {
        font-size: 0.9em; /* Standard font size for description */
        margin-bottom: 3px; /* Reduced margin below the text */
    }

    .product-card .product-price {
        font-size: 0.95em; /* Standard font size for price */
        margin-bottom: 3px; /* Reduced margin below the price */
        font-weight: bold; /* Make price bold for emphasis */
    }

    .btn-primary.btn-sm {
        padding: 7px 12px; /* Adjusted padding for button size */
        font-size: 0.9em; /* Reduced font size for button */
        margin-top: 0; /* Remove margin-top to bring button closer to the price */
    }

    /* Adjust Pagination Buttons Position */
    .d-flex.justify-content-between.align-items-center {
        margin-top: -20px; /* Slight negative margin to move buttons up */
        margin-bottom: 10px; /* Reduce bottom margin for better spacing */
    }
</style>

<?php include '../templates/footer.php'; ?>

```

## To αρχείο Order.php

```
<?php

// Start session and include necessary files

session_start();

require_once '../config/config.php'; // Correct path to config.php


// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header('Location: ../auth/login.php'); // Redirect to login page if not logged in
    exit();
}


// Get user ID from session
$user_id = $_SESSION['user_id'];


// Initialize messages array for dynamic feedback
$messages = ['error' => '', 'success' => ''];


// Handle AJAX requests
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $response = ['status' => 'error', 'message' => 'Invalid request'];

    if (isset($_POST['action'])) {
        switch ($_POST['action']) {
            case 'update':
                if (isset($_POST['cart_item_id'], $_POST['quantity'])) {
                    $cart_item_id = (int)$_POST['cart_item_id'];
                    $quantity = (int)$_POST['quantity'];

                    // Fetch the current product ID and available stock
                    $query = "SELECT ci.product_id, ci.quantity, p.stock_quantity FROM
cart_items ci JOIN products p ON ci.product_id = p.product_id WHERE ci.cart_item_id =
:cart_item_id";
```

```

$stmt = $conn->prepare($query);

$stmt->bindValue(':cart_item_id', $cart_item_id, PDO::PARAM_INT);

$stmt->execute();

$cart_item = $stmt->fetch(PDO::FETCH_ASSOC);

if ($cart_item) {

    $product_id = $cart_item['product_id'];

    $current_quantity_in_cart = $cart_item['quantity'];

    $available_stock = $cart_item['stock_quantity'];

    // Check if requested quantity exceeds available stock

    if ($quantity > ($available_stock + $current_quantity_in_cart)) {

        $response['message'] = 'Not enough stock available.';

        $response['current_quantity'] = $current_quantity_in_cart; // Send
current quantity to revert

    } else {

        // Calculate new stock quantity

        $new_stock_quantity = $available_stock + $current_quantity_in_cart
- $quantity;

        // Update the quantity in the database

        $query = "UPDATE cart_items SET quantity = :quantity WHERE
cart_item_id = :cart_item_id";

        $stmt = $conn->prepare($query);

        $stmt->bindValue(':quantity', $quantity, PDO::PARAM_INT);

        $stmt->bindValue(':cart_item_id', $cart_item_id, PDO::PARAM_INT);

        if ($stmt->execute()) {

            // Update the stock in the products table

            $query = "UPDATE products SET stock_quantity =
:new_stock_quantity WHERE product_id = :product_id";

            $stmt = $conn->prepare($query);

            $stmt->bindValue(':new_stock_quantity', $new_stock_quantity,
PDO::PARAM_INT);

            $stmt->bindValue(':product_id', $product_id, PDO::PARAM_INT);

            $stmt->execute();

            $response = ['status' => 'success', 'message' => 'Cart updated
successfully.'];

        } else {

            $response['message'] = 'Failed to update cart.';

```

```

        }

    }
    } else {

        $response['message'] = 'Item not found in cart.';

    }
}

break;

case 'remove':

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

        $cart_item_id = (int)$_POST['cart_item_id'];

        // Get the product ID and quantity of the item to be removed
        $query = "SELECT product_id, quantity FROM cart_items WHERE cart_item_id =
:cart_item_id";

        $stmt = $conn->prepare($query);

        $stmt->bindValue(':cart_item_id', $cart_item_id, PDO::PARAM_INT);

        $stmt->execute();

        $cart_item = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($cart_item) {

            // Remove the item from the cart

            $query = "DELETE FROM cart_items WHERE cart_item_id = :cart_item_id";

            $stmt = $conn->prepare($query);

            $stmt->bindValue(':cart_item_id', $cart_item_id, PDO::PARAM_INT);

            if ($stmt->execute()) {

                // Increase the stock quantity in the products table

                $query = "UPDATE products SET stock_quantity = stock_quantity +
:quantity WHERE product_id = :product_id";

                $stmt = $conn->prepare($query);

                $stmt->bindValue(':quantity', $cart_item['quantity'],
PDO::PARAM_INT);

                $stmt->bindValue(':product_id', $cart_item['product_id'],
PDO::PARAM_INT);

                $stmt->execute();

                $response = ['status' => 'success', 'message' => 'Item removed
successfully.'];

            } else {

                $response['message'] = 'Failed to remove item from cart.';

```

```

        }

        } else {

            $response['message'] = 'Item not found in cart.';

        }

    }

    break;

case 'clear':

    // Fetch all cart items to adjust stock quantities

    $query = "SELECT ci.product_id, ci.quantity FROM cart_items ci JOIN carts c ON
ci.cart_id = c.cart_id WHERE c.user_id = :user_id";

    $stmt = $conn->prepare($query);

    $stmt->bindValue(':user_id', $user_id, PDO::PARAM_INT);

    $stmt->execute();

    $cart_items = $stmt->fetchAll(PDO::FETCH_ASSOC);

    // Clear the entire cart

    $query = "DELETE FROM cart_items WHERE cart_id IN (SELECT cart_id FROM carts
WHERE user_id = :user_id)";

    $stmt = $conn->prepare($query);

    $stmt->bindValue(':user_id', $user_id, PDO::PARAM_INT);

    if ($stmt->execute()) {

        // Restore the stock for all removed items

        foreach ($cart_items as $item) {

            $query = "UPDATE products SET stock_quantity = stock_quantity +
:quantity WHERE product_id = :product_id";

            $stmt = $conn->prepare($query);

            $stmt->bindValue(':quantity', $item['quantity'], PDO::PARAM_INT);

            $stmt->bindValue(':product_id', $item['product_id'], PDO::PARAM_INT);

            $stmt->execute();

        }

        $response = ['status' => 'success', 'message' => 'Cart cleared
successfully.'];

    } else {

        $response['message'] = 'Failed to clear cart.';

    }

    break;

}

```

```

    }

    echo json_encode($response);

    exit();
}

// Fetch the user's cart ID
$query = "SELECT cart_id FROM carts WHERE user_id = :user_id";
$stmt = $conn->prepare($query);
$stmt->bindValue(':user_id', $user_id, PDO::PARAM_INT);
$stmt->execute();
$cart = $stmt->fetch(PDO::FETCH_ASSOC);

// If no cart exists, create a new one
if (!$cart) {
    $query = "INSERT INTO carts (user_id) VALUES (:user_id)";
    $stmt = $conn->prepare($query);
    $stmt->bindValue(':user_id', $user_id, PDO::PARAM_INT);
    $stmt->execute();
    $cart_id = $conn->lastInsertId();
} else {
    $cart_id = $cart['cart_id'];
}

// Initialize total variable to avoid undefined variable warning
$total = 0;

// Fetch cart items
$query = "SELECT ci.cart_item_id, ci.product_id, ci.quantity, ci.price, p.name, p.image
        FROM cart_items ci
        JOIN products p ON ci.product_id = p.product_id
        WHERE ci.cart_id = :cart_id";
$stmt = $conn->prepare($query);
$stmt->bindValue(':cart_id', $cart_id, PDO::PARAM_INT);
$stmt->execute();
$cart_items = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

```

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Your Cart</title>

    <!-- Include Bootstrap and Font Awesome for icons -->

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
rel="stylesheet">

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">

    <link rel="stylesheet" href="../../assets/css/style.css"> <!-- Global site styles -->

    <link rel="stylesheet" href="../../assets/css/cart.css"> <!-- Cart-specific styles -->

    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

    <!-- Include Bootstrap JS after jQuery -->

    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>

    <!-- Custom CSS for styling -->

    <style>

        /* Custom styling for quantity buttons and input */

        .quantity-input {

            text-align: center;

            max-width: 50px;

            -moz-appearance: textfield; /* For Firefox */

        }

        .quantity-input::-webkit-outer-spin-button,

        .quantity-input::-webkit-inner-spin-button {

            -webkit-appearance: none; /* For Chrome, Safari, Edge, Opera */

            margin: 0;

        }

        .quantity-btn {

            width: 30px;

            height: 30px;

            line-height: 30px;

            padding: 0;

            border: none;

            font-weight: bold;

            border-radius: 50%;

            cursor: pointer;

            transition: background-color 0.3s ease;


```

```

}

.quantity-btn.decrease {
    background-color: #dc3545; /* Red for decrease button */
    color: #fff;
}

.quantity-btn.decrease:hover {
    background-color: #c82333; /* Darker red on hover */
}

.quantity-btn.increase {
    background-color: #28a745; /* Green for increase button */
    color: #fff;
}

.quantity-btn.increase:hover {
    background-color: #218838; /* Darker green on hover */
}

/* Gradient background for summary container */
.cart-summary {
    background: linear-gradient(135deg, #f6d365 0%, #fda085 100%); /* Gradient from
yellowish to orange */
    border-radius: 15px; /* Rounded corners */
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1); /* Subtle shadow for depth */
    padding: 25px;
    width: 300px;
    position: sticky;
    top: 20px;
    margin-left: auto; /* Pushes the summary container to the far right */
}

.cart-summary h3, .cart-summary h5 {
    color: #fff; /* White text for better contrast */
}

.total-price {
    font-size: 1.5em; /* Larger font size for emphasis */
    color: #fff; /* White color for better contrast */
    font-weight: bold;
}

.btn-clear-cart, .btn-checkout {
    display: flex;
    align-items: center;

```

```

    justify-content: center;

    font-size: 1em; /* Smaller font size */

    font-weight: bold;

    padding: 10px 15px; /* Reduced padding for smaller size */

    border: none;

    border-radius: 5px;

    cursor: pointer;

    width: 100%;

    margin-top: 10px; /* Adjusted margin */

    transition: background-color 0.3s ease, transform 0.2s ease;
}

.btn-clear-cart {

    background-color: #dc3545; /* Red color for "Clear Cart" */

    color: #fff;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.btn-clear-cart:hover {

    background-color: #c82333;

    transform: scale(1.03); /* Slightly smaller scale on hover */
}

.btn-checkout {

    background-color: #28a745; /* Green color for "Proceed to Checkout" */

    color: #fff;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.btn-checkout:hover {

    background-color: #218838;

    transform: scale(1.03); /* Slightly smaller scale on hover */
}

/* Enhanced alert styling */

.custom-alert {

    padding: 15px;

    margin-bottom: 20px;

    border-radius: 5px;

    font-size: 16px;

    font-weight: bold;

    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);

    opacity: 0;

```

```

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

    .custom-alert.show {
        opacity: 1;
    }
</style>
</head>
<body>
<?php include '../templates/header.php'; ?>

<section class="h-100 h-custom">
    <div class="card-container">
        <div class="col-lg-8 col-12">
            <!-- Cart Items -->
            <div class="card-title-container">
                <h1 class="fw-bold mb-0 text-black">Shopping Cart</h1>
                <h6 class="mb-0 text-muted"><?php echo count($cart_items); ?> items</h6>
            </div>

            <!-- Display alert messages -->
            <div id="alert-container"></div>

            <?php if (count($cart_items) > 0): ?>
                <?php $total = 0; ?>
                <?php foreach ($cart_items as $item): ?>
                    <div class="card-item row mb-3">
                        <div class="col-3">
                            ">
                        </div>
                        <div class="col-6">
                            <h5 class="card-title"><?php echo htmlspecialchars($item['name']); ?></h5>
                            <p class="card-text">Price: $<?php echo number_format($item['price'], 2);
?></p>

                            <!-- Input group with minus and plus buttons -->
                            <div class="d-flex align-items-center mb-3">
                                <?php if ($item['quantity'] > 1): ?>
                                    <button class="quantity-btn decrease" type="button" data-id="<?php
echo $item['cart_item_id']; ?>" data-action="decrease"></button>

```

```

        <?php endif; ?>

        <input type="number" class="form-control quantity-input mx-2" data-
id="<?php echo $item['cart_item_id']; ?>" value="<?php echo $item['quantity']; ?>" min="1">

        <button class="quantity-btn increase" type="button" data-id="<?php
echo $item['cart_item_id']; ?>" data-action="increase">+</button>

        </div>

    </div>

    <div class="col-3 text-right">

        <p class="card-text">Total: $<?php echo number_format($item['price'] *
$item['quantity'], 2); ?></p>

        <button class="btn btn-danger btn-sm remove-cart" data-id="<?php echo
$item['cart_item_id']; ?>">Remove</button>

    </div>

</div>

    <?php $total += $item['price'] * $item['quantity']; ?>

    <?php endforeach; ?>

    <?php else: ?>

        <p>Your cart is empty.</p>

    <?php endif; ?>

    <div class="pt-5">

        <h6 class="mb-0"><a href="../categories/index.php" class="text-body"><i class="fas fa-
long-arrow-alt-left me-2"></i>Back to shop</a></h6>

    </div>

</div>

<!-- Summary Container -->

<div class="cart-summary">

    <h3 class="fw-bold mb-5 mt-2 pt-1">Summary</h3>

    <hr class="my-4">

    <div class="d-flex justify-content-between mb-4">

        <h5 class="text-uppercase">Total Items</h5>

        <h5><?php echo count($cart_items); ?></h5>

    </div>

    <div class="d-flex justify-content-between mb-5">

        <h5 class="text-uppercase">Total Price</h5>

        <h5 class="total-price">€ <?php echo number_format($total, 2); ?></h5>

    </div>

```

```

<button type="button" class="btn-clear-cart">
    <i class="fas fa-trash-alt"></i> Clear Cart
</button>

<button type="button" class="btn-checkout" onclick="location.href='checkout.php'">
    <i class="fas fa-shopping-cart"></i> Proceed to Checkout
</button>
</div>
</div>
</section>

<?php include '../templates/footer.php'; ?>

<!-- Ensure the script is after the HTML content -->
<script>
$(document).ready(function() {
    // Function to show alerts dynamically without close button
    function showAlert(type, message) {
        var alertHtml = '<div class="custom-alert alert-' + type + '">' +
            message +
            '</div>';

        $('#alert-container').html(alertHtml);

        $('#alert-container .custom-alert').addClass('show');

        // Auto-dismiss alert after 5 seconds
        setTimeout(function() {
            $('#alert-container .custom-alert').fadeOut(500, function() {
                $(this).remove(); // Remove the alert from DOM
            });
        }, 5000);
    }

    // Function to clear cart via AJAX
    function clearCart() {
        if (confirm('Are you sure you want to clear your cart?')) {
            $.ajax({
                url: 'cart.php',
                method: 'POST',
            });
        }
    }
});

```

```

        data: {action: 'clear'},
        dataType: 'json',
        success: function(response) {
            if (response.status === 'success') {
                showAlert('success', response.message);
                location.reload(); // Reload the page or update the cart dynamically
            } else {
                showAlert('danger', response.message);
            }
        }
    });
}

// Attach click event handler to the clear cart button
$('.btn-clear-cart').on('click', function() {
    clearCart();
});

// Handle quantity change with AJAX
$('.quantity-input').on('change', function() {
    updateQuantity($(this).data('id'), $(this).val());
});

// Handle minus and plus buttons click
$('.quantity-btn').on('click', function() {
    var cart_item_id = $(this).data('id');
    var action = $(this).data('action');
    var $inputField = $('.quantity-input[data-id="' + cart_item_id + '"]');
    var currentQuantity = parseInt($inputField.val());

    if (action === 'decrease' && currentQuantity > 1) {
        $inputField.val(currentQuantity - 1);
    } else if (action === 'increase') {
        $inputField.val(currentQuantity + 1);
    }

    updateQuantity(cart_item_id, $inputField.val());
}

```

```

});

// Function to update quantity via AJAX
function updateQuantity(cart_item_id, new_quantity) {
    if (new_quantity > 0) {
        $.ajax({
            url: 'cart.php',
            method: 'POST',
            data: {action: 'update', cart_item_id: cart_item_id, quantity: new_quantity},
            dataType: 'json',
            success: function(response) {
                if (response.status === 'success') {
                    showAlert('success', response.message);
                    location.reload(); // Reload the page or update the cart dynamically
                } else {
                    showAlert('danger', response.message);
                    $('.quantity-input[data-id="' + cart_item_id +
                    '"']).val(response.current_quantity); // Revert to the correct quantity
                }
            }
        });
    } else {
        showAlert('danger', "Quantity must be at least 1.");
        $('.quantity-input[data-id="' + cart_item_id + '"']).val(1); // Revert to minimum
        value if below 1
    }
}

// Handle item removal with AJAX
$('.remove-cart').on('click', function(e) {
    e.preventDefault();
    var cart_item_id = $(this).data('id');

    $.ajax({
        url: 'cart.php',
        method: 'POST',
        data: {action: 'remove', cart_item_id: cart_item_id},
        dataType: 'json',
        success: function(response) {

```

```
        if (response.status === 'success') {  
            showAlert('success', response.message);  
            location.reload(); // Reload the page or update the cart dynamically  
        } else {  
            showAlert('danger', response.message);  
        }  
    }  
    });  
});  
</script>  
</body>  
</html>
```

## To αρχείο cart.php

```
<?php

session_start();

require_once '../config/config.php'; // Correct path to config.php

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header('Location: ../auth/login.php'); // Redirect to login page if not logged in
    exit();
}

// Get user ID from session
$user_id = $_SESSION['user_id'];

// Fetch orders for the user
$query = "SELECT o.order_id, o.total_amount, o.created_at, s.status_name
          FROM orders o
          JOIN order_statuses s ON o.status_id = s.status_id
          WHERE o.user_id = :user_id
          ORDER BY o.created_at DESC";

$stmt = $conn->prepare($query);
$stmt->bindValue(':user_id', $user_id, PDO::PARAM_INT);
$stmt->execute();

$orders = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Your Orders</title>
    <!-- Include Bootstrap and Font Awesome for icons -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
rel="stylesheet">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
    <link rel="stylesheet" href="../assets/css/orders.css"> <!-- Orders-specific styles -->
```

```

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>
</head>

<body>

<?php include '../templates/header.php'; ?>

<div class="container orders-container">

    <h2 class="mb-4">Your Orders</h2>

    <?php if (count($orders) > 0): ?>

        <ul class="orders-list">

            <?php foreach ($orders as $order): ?>

                <li class="order-item">

                    <div class="order-details">

                        <div>

                            <strong>Order ID:</strong> <?php echo
htmlspecialchars($order['order_id']); ?><br>

                            <strong>Date:</strong> <?php echo htmlspecialchars(date("F j, Y,
g:i a", strtotime($order['created_at']))); ?><br>

                            <strong>Total:</strong> $<?php echo
number_format($order['total_amount'], 2); ?>

                        </div>

                        <div class="order-status <?php echo strtolower($order['status_name']);
?>"

                            title="<?php echo ucfirst($order['status_name']); ?>"

                            <?php echo htmlspecialchars($order['status_name']); ?>

                        </div>

                    </div>

                </li>

            <?php endforeach; ?>

        </ul>

        <?php else: ?>

            <p>You have no orders yet.</p>

        <?php endif; ?>

    </div>

<?php include '../templates/footer.php'; ?>

```

```
<script>  
    // Optional: Add JavaScript for enhancing UX like collapsible order details, etc.  
</script>  
  
</body>  
</html>
```