



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Διαδικτυακές Κρατήσεις Αεροπορικών Εισιτηρίων - Online  
Air Ticket Reservation**

**Γιαννουκάκος Μιχαήλ**

**Εισηγητής: Κωνσταντίνος Κουκουλέτσος, Καθηγητής**

## Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένεια μου για τη στήριξή της και τις θυσίες της ώστε να μπορέσω να είμαι εδώ που είμαι σήμερα, σπουδάζοντας πάνω στο γνωστικό αντικείμενο το οποίο και επιθυμούσα. Στη συνέχεια, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου κ. Κουκουλέτσο Κωνσταντίνο για την εμπιστοσύνη που μου έδειξε στην ανάθεση της πτυχιακής μου εργασίας αλλά και για τις πολύτιμες συμβουλές του κατά την περίοδο εκπόνησής της και εν τέλει υλοποίησή της.

## Περίληψη

Στόχος της πτυχιακής μου εργασίας είναι η κατασκευή μιας δυναμικής ιστοσελίδας με κύρια λειτουργία την κράτηση αεροπορικών εισιτηρίων. Ο χρήστης εφόσον έχει κάνει εγγραφή με τα προσωπικά στοιχεία του στην ιστοσελίδα έχει τη δυνατότητα ως πελάτης να κάνει κράτηση εισιτηρίων για μια συγκεκριμένη πτήση συμπληρώνοντας τη φόρμα αναζήτησης η οποία εμφανίζεται στην αρχική σελίδα εφόσον τηρούνται ορισμένες προϋποθέσεις. Επίσης, είναι άμεσα δυνατή η ακύρωση της κράτησης εφόσον ο ίδιος ο χρήστης το επιθυμεί χωρίς να τίθεται θέμα στο οικονομικό κομμάτι. Επιπλέον, η ιστοσελίδα έχει διαθέσιμες πληροφορίες σχετικά με τις οδηγίες ασφαλείας αλλά και τις υπηρεσίες για μια πιο ευχάριστη πτήση. Η ιστοσελίδα υλοποιήθηκε με χρήση των γλωσσών προγραμματισμού HTML, PHP, CSS και Javascript. Χρησιμοποιήθηκε επίσης η MySQL για αποθήκευση και ανάκτηση δεδομένων από τη βάση.

## Abstract

The goal of my graduate thesis is to build a dynamic website with the main function of booking airline tickets. If the user has registered with his / her personal information on the website, he / she has the option as a customer to book tickets for a particular flight by filling out the search form displayed on the home page if certain conditions are met. In addition, it is immediately possible to cancel the reservation if the user wishes without questioning the financial part. Furthermore, the site has information regarding safety instructions and services available for a more enjoyable flight. The website was implemented using the HTML, PHP, CSS and Javascript programming languages. MySQL was also used to store and retrieve data from the database.

## Περιεχόμενα

Ευχαριστίες.....	1
Περίληψη.....	3
Abstract.....	4
Εικόνες.....	7
Κεφάλαιο 1 – Εισαγωγή .....	8
1.1 Διαδίκτυο (Internet) .....	8
1.2 Παγκόσμιος Ιστός (World Wide Web).....	8
1.3 Πρωτόκολλο Μεταφοράς Υπερκειμένου (Hypertext Transfer Protocol) .....	8
1.4 Ιστοσελίδα .....	9
1.4.1 Στατική Ιστοσελίδα .....	9
1.4.2 Δυναμική Ιστοσελίδα.....	10
1.5 Ιστότοπος.....	11
1.6 Περιηγητής Ιστού (Web Browser).....	12
1.7 Φιλοξενία Ιστοσελίδων (Web Hosting).....	13
1.8 Σύστημα Ονομάτων Τομέων (Domain Name System).....	14
1.9 Εξυπηρετητής Ιστού (Web Server).....	15
Κεφάλαιο 2 – Ηλεκτρονικές αγορές μέσω χρεωστικής-πιστωτικής κάρτας.....	17
2.1 Λογαριασμός εμπόρου.....	17
2.2 Τερματικό πληρωμής (Payment terminal).....	17
2.2.1 Ιστορία του τερματικού πληρωμής.....	19
2.3 Μονάδα αυτοματοποιημένης απόκρισης (ARU) .....	21
2.4 Πύλη πληρωμής (Payment gateway) .....	21
2.4.1 Διαδικασία συναλλαγών.....	21
Κεφάλαιο 3 – Εργαλεία Υλοποίησης.....	24
3.1 Apache HTTP Server .....	24
3.2 Λογισμικό ΧΑΜΡΡ .....	25
3.3 Notepad ++ .....	26
3.4 Γλώσσα Προγραμματισμού HTML (Hyper Text Markup Language).....	27
3.4.1 Έκδοση HTML5.....	28
3.5 Γλώσσα Προγραμματισμού PHP (Hypertext Preprocessor).....	29
3.5.1 Μεταβλητές Περιόδων Σύνδεσης (Sessions).....	29

3.5.2 Διαφορές HTTP GET & POST .....	31
3.5.5 Άλλες μέθοδοι αιτήσεων HTTP .....	32
3.6 MySQL .....	33
3.7 phpMyAdmin .....	34
3.7.1 Βασικές λειτουργίες phpMyAdmin .....	34
3.8 Γλώσσα προγραμματισμού CSS (Cascading Style Sheets) .....	35
3.9 Γλώσσα προγραμματισμού JavaScript .....	36
3.10 Πρωτόκολλο SMTP (Simple Mail Transfer Protocol) .....	37
3.10.1 Ιστορία του Πρωτοκόλλου SMTP .....	38
3.10.2 Test Mail Server Tool .....	39
Κεφάλαιο 4 – Ανάλυση λειτουργίας της εφαρμογής .....	40
4.1 Η βάση και οι πίνακες της .....	40
4.2 Αρχική σελίδα (index.php).....	44
4.3 Λογαριασμός χρήστη .....	46
4.3.1 Δημιουργία Λογαριασμού (fregister.php) .....	46
4.3.2 Σύνδεση Λογαριασμού (flogin.php) .....	47
4.3.3 Διαχείριση Λογαριασμού .....	49
4.4 Πραγματοποίηση κράτησης εισιτηρίων .....	50
Κεφάλαιο 5 – Συμπεράσματα .....	54
Παράρτημα .....	55
Παραπομπές .....	94
Βιβλιογραφία.....	96

## Εικόνες

Εικόνα 1.1: Παράδειγμα δημιουργίας στατικής ιστοσελίδας .....	9
Εικόνα 1.2: Παράδειγμα δημιουργίας δυναμικής ιστοσελίδας .....	10
Εικόνα 1.3: Παραδείγματα δημοφιλών web browsers .....	12
Εικόνα 3.1: Το λογότυπο του Apache HTTP Server .....	24
Εικόνα 3.2: Λογότυπο του XAMPP .....	25
Εικόνα 3.3: Λογότυπο της HTML5 .....	27
Εικόνα 3.4: Λογότυπο της PHP .....	29
Εικόνα 3.5: Λογότυπο της MySQL .....	33
Εικόνα 3.6: Λογότυπο του phpMyAdmin .....	34
Εικόνα 3.7: Λογότυπο της CSS3 .....	35
Εικόνα 3.8: Το βασικό μενού της εφαρμογής .....	39
Εικόνες 4.1: Η βάση δεδομένων swift και οι πίνακες. ....	40
Εικόνα 4.2: Ο πίνακας «booking_details» .....	41
Εικόνα 4.3: Ο πίνακας «flight_search» .....	42
Εικόνα 4.4: Ο πίνακας «flight_users» .....	43
Εικόνα 4.5: Ο πίνακας «passenger_details» .....	44
Εικόνα 4.6: Η εμφάνιση της αρχικής ιστοσελίδας.....	44
Εικόνα 4.7: Η εμφάνιση των επιλογών Register και Login.....	45
Εικόνα 4.8: Εμφάνιση της σελίδας fregister.php .....	46
Εικόνα 4.9: Εμφάνιση της σελίδας flogin.php .....	47
Εικόνα 4.10: Σελίδα ανάκτησης username ή password .....	48
Εικόνα 4.11: Σελίδα αντικατάστασης password.....	49
Εικόνα 4.12: Οι αλλαγές στο «Passenger Login» όταν ο χρήστης συνδέεται στο λογαριασμό του .....	49
Εικόνα 4.13: Η ιστοσελίδα settings.php όπου ο χρήστης κάνει αλλαγές στα προσωπικά του στοιχεία .....	50
Εικόνα 4.14: Η εμφάνιση των διαθέσιμων πτήσεων για τις συγκεκριμένες ημερομηνίες .....	51
Εικόνα 4.15: Η σελίδα book.php .....	51
Εικόνα 4.16: Το τελευταίο στάδιο της διαδικασίας για την κράτηση εισιτηρίων .....	52
Εικόνα 4.17: Η σελίδα bookings.php.....	53

## Κεφάλαιο 1 – Εισαγωγή

### 1.1 Διαδίκτυο (Internet)

Το Διαδίκτυο είναι το παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών που χρησιμοποιούν τη σουίτα πρωτοκόλλου Internet (TCP / IP) για τη σύνδεση συσκευών σε όλο τον κόσμο. Πρόκειται για ένα δίκτυο δικτύων που αποτελείται από ιδιωτικά, δημόσια, ακαδημαϊκά, επιχειρηματικά και κυβερνητικά δίκτυα τοπικού και παγκόσμιου πεδίου, που συνδέονται με ένα ευρύ φάσμα τεχνολογιών ηλεκτρονικής, ασύρματης και οπτικής δικτύωσης. Το Διαδίκτυο μεταφέρει ένα εύρος πληροφοριών και υπηρεσιών, όπως τα έγγραφα υπερκειμένου και εφαρμογές του Παγκόσμιου Ιστού (WWW), ηλεκτρονικό ταχυδρομείο, τηλεφωνία και κοινή χρήση αρχείων. (GCF Learnfree, 2018)

### 1.2 Παγκόσμιος Ιστός (World Wide Web)

Ο Παγκόσμιος Ιστός είναι χώρος πληροφόρησης όπου τα έγγραφα και άλλοι δικτυακοί πόροι εντοπίζονται από τους Ενιαίους Εντοπιστές Πόρων (URL), διασυνδεδεμένοι με συνδέσεις υπερκειμένου και προσβάσιμοι μέσω του Διαδικτύου χρησιμοποιώντας το πρωτόκολλο επικοινωνίας HTTP. Ο Παγκόσμιος Ιστός ήταν κεντρικός στην ανάπτυξη της εποχής της πληροφορίας και είναι το βασικό εργαλείο που δουλεύουν δισεκατομμύρια άνθρωποι για να αλληλεπιδράσουν στο Διαδίκτυο. (Technopedia, 2018)

### 1.3 Πρωτόκολλο Μεταφοράς Υπερκειμένου (Hypertext Transfer Protocol)

Το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) είναι ένα πρωτόκολλο εφαρμογής για καταναμημένα και συνεργατικά πληροφοριακά συστήματα. Το HTTP είναι το θεμέλιο της επικοινωνίας δεδομένων για τον Παγκόσμιο Ιστό. Το Hypertext είναι δομημένο κείμενο που χρησιμοποιεί λογικούς συνδέσμους (υπερσυνδέσμους) μεταξύ κόμβων που περιέχουν κείμενο. Το HTTP είναι το πρωτόκολλο για την ανταλλαγή ή τη μεταφορά υπερκειμένου. (Programming Notes, 2018)



## 1.4 Ιστοσελίδα

Μια ιστοσελίδα είναι ένα έγγραφο που είναι κατάλληλο για το World Wide Web και τα προγράμματα περιήγησης ιστού. Ένα πρόγραμμα περιήγησης ιστού εμφανίζει μια ιστοσελίδα σε μια οθόνη ή μια κινητή συσκευή. Οι περιηγητές ιστού συντονίζουν διάφορα στοιχεία πόρων ιστού για την γραπτή ιστοσελίδα, όπως φύλλα στυλ, σενάρια και εικόνες, για να παρουσιάσουν την ιστοσελίδα. Οι τυπικές ιστοσελίδες παρέχουν υπερκείμενο που περιλαμβάνει μια γραμμή πλοήγησης ή ένα μενού της πλαϊνής γραμμής που συνδέεται με άλλες ιστοσελίδες μέσω υπερσυνδέσμων, που συχνά αναφέρονται ως συνδέσεις. Σε ένα δίκτυο, ένα πρόγραμμα περιήγησης ιστού μπορεί να ανακτήσει μια ιστοσελίδα από έναν απομακρυσμένο διακομιστή ιστού. Ο διακομιστής ιστού ενδέχεται να περιορίσει την πρόσβαση σε ένα ιδιωτικό δίκτυο, όπως ένα εταιρικό intranet. Το πρόγραμμα περιήγησης ιστού χρησιμοποιεί το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) για να κάνει τέτοια αιτήματα. (Computerhope, 2018)

### 1.4.1 Στατική Ιστοσελίδα



Εικόνα 1.1: Παράδειγμα δημιουργίας στατικής ιστοσελίδας

Μια στατική ιστοσελίδα είναι μια ιστοσελίδα που παραδίδεται στον χρήστη ακριβώς όπως αποθηκεύεται, σε αντίθεση με τις δυναμικές ιστοσελίδες που δημιουργούνται από μια εφαρμογή ιστού.

Συνεπώς, μια στατική ιστοσελίδα εμφανίζει τις ίδιες πληροφορίες για όλους τους χρήστες, από όλα τα περιβάλλοντα, υπόκεινται στις σύγχρονες δυνατότητες ενός διακομιστή ιστού για να διαπραγματευτούν τον τύπο περιεχομένου ή τη γλώσσα του εγγράφου όπου υπάρχουν διαθέσιμες εκδόσεις και ο διακομιστής έχει ρυθμιστεί να το κάνει.

Οι στατικές ιστοσελίδες είναι κυρίως έγγραφα HTML που αποθηκεύονται ως αρχεία στο σύστημα αρχείων και διατίθενται από τον διακομιστή μέσω HTTP (παρόλα αυτά οι διευθύνσεις URL που τελειώνουν με ".html" δεν είναι πάντα στατικές). Ωστόσο, οι

ερμηνείες του όρου θα μπορούσαν να περιλαμβάνουν ιστοσελίδες που είναι αποθηκευμένες σε μια βάση δεδομένων και θα μπορούσαν ακόμη και να περιλαμβάνουν μορφοποιημένες σελίδες χρησιμοποιώντας ένα πρότυπο και να εξυπηρετούνται μέσω ενός διακομιστή εφαρμογών, εφόσον η σελίδα που εξυπηρετείται είναι αμετάβλητη και παρουσιάζεται ουσιαστικά ως αποθηκευμένη.

Οι στατικές ιστοσελίδες είναι κατάλληλες για τα περιεχόμενα που ποτέ ή σπάνια χρειάζεται να ενημερωθούν. Η διατήρηση μεγάλου αριθμού στατικών σελίδων ως αρχείων μπορεί να είναι ανέφικτη χωρίς αυτοματοποιημένα εργαλεία, όπως οι γεννήτριες στατικών τοποθεσιών που περιγράφονται στο σύστημα πρότυπων ιστού. Οποιαδήποτε εξατομίκευση ή διαδραστικότητα πρέπει να εκτελεστεί από την πλευρά του πελάτη.

Στα πλεονεκτήματα τους είναι το γεγονός ότι παρέχεται βελτιωμένη ασφάλεια σε σχέση με τους δυναμικούς ιστότοπους, βελτιωμένη απόδοση για τους τελικούς χρήστες σε σύγκριση με τους δυναμικούς ιστότοπους, δεν απαιτούνται προγραμματιστικές γνώσεις για τη δημιουργία τους και τέλος απαιτούνται λιγότερες ή καθόλου εξαρτήσεις σε συστήματα όπως βάσεις δεδομένων ή άλλους διακομιστές εφαρμογών για να λειτουργήσουν.

Μειονεκτήματα μιας στατικής ιστοσελίδας είναι η δυναμική λειτουργία πρέπει να προστεθεί ξεχωριστά και επίσης η διαδραστικότητά της με το χρήστη δεν είναι εύκολη (Bulkar, 2018).

#### 1.4.2 Δυναμική Ιστοσελίδα



Εικόνα 1.2: Παράδειγμα δημιουργίας δυναμικής ιστοσελίδας

Μια δυναμική ιστοσελίδα είναι μια ιστοσελίδα η κατασκευή της οποίας ελέγχεται από ένα σενάριο επεξεργασίας διακομιστή εφαρμογών. Στο scripting από την πλευρά του διακομιστή, οι παράμετροι καθορίζουν τον τρόπο με τον οποίο προχωράει η συναρμολόγηση κάθε νέας ιστοσελίδας, συμπεριλαμβανομένης της δημιουργίας περισσότερης επεξεργασίας από την πλευρά του πελάτη.

Μια δυναμική ιστοσελίδα από την πλευρά του πελάτη επεξεργάζεται την ιστοσελίδα χρησιμοποιώντας γλώσσα HTML που εκτελείται στο πρόγραμμα περιήγησης καθώς φορτώνεται. Η JavaScript και άλλες γλώσσες δέσμης ενεργειών καθορίζουν τον τρόπο με τον οποίο αναλύεται η HTML στην λαμβανόμενη σελίδα στο μοντέλο αντικειμένου εγγράφου ή αλλιώς DOM που αντιπροσωπεύει την φορτωμένη ιστοσελίδα. Οι ίδιες τεχνικές από την πλευρά του πελάτη μπορούν στη συνέχεια να ενημερώσουν δυναμικά ή να αλλάξουν το DOM με τον ίδιο τρόπο.

Στη συνέχεια, μια δυναμική ιστοσελίδα επαναφορτώνεται από το χρήστη ή από ένα πρόγραμμα υπολογιστή για να αλλάξει κάποιο μεταβλητό περιεχόμενο. Οι πληροφορίες ενημέρωσης θα μπορούσαν να προέρχονται από το διακομιστή ή από τις αλλαγές που έγιναν στο DOM της συγκεκριμένης σελίδας. Αυτό μπορεί να μειώσει ή να μην περιορίσει το ιστορικό περιήγησης ή να δημιουργήσει μια αποθηκευμένη έκδοση για να επιστρέψει, αλλά μια δυναμική ενημέρωση ιστοσελίδας που χρησιμοποιεί τεχνολογίες Ajax δεν θα δημιουργήσει ούτε μια σελίδα για να επιστρέψει ούτε να περικόψει το ιστορικό περιήγησης στον ιστό μπροστά από την εμφανιζόμενη σελίδα. Χρησιμοποιώντας τις τεχνολογίες Ajax, ο τελικός χρήστης αποκτά μία δυναμική σελίδα που διαχειρίζεται ως μία σελίδα στο πρόγραμμα περιήγησης ιστού, ενώ το πραγματικό περιεχόμενο ιστού που εμφανίζεται σε αυτήν τη σελίδα μπορεί να διαφέρει. Ο μηχανισμός Ajax βρίσκεται μόνο στο πρόγραμμα περιήγησης που ζητά τμήματα του DOM του, DOM, για τον πελάτη του, από διακομιστή εφαρμογών. (Bulgar, 2018)

## 1.5 Ιστότοπος

Ένας ιστότοπος είναι μια συλλογή σχετικών ιστοσελίδων, συμπεριλαμβανομένου του περιεχομένου πολυμέσων, που προσδιορίζεται με ένα κοινό όνομα τομέα και δημοσιεύεται σε τουλάχιστον έναν διακομιστή ιστού. Σημαντικά παραδείγματα είναι τα wikipedia.org, google.com και amazon.com.

Ένας δικτυακός τόπος μπορεί να είναι προσβάσιμος μέσω ενός δημόσιου δικτύου πρωτοκόλλου Internet (IP), όπως το Διαδίκτυο ή ένα ιδιωτικό τοπικό δίκτυο (LAN), αναφέροντας έναν ενιαίο εντοπιστή πόρων (URL) που προσδιορίζει τον ιστότοπο.

Οι ιστότοποι μπορούν να έχουν πολλές λειτουργίες και μπορούν να χρησιμοποιηθούν σε διάφορες μορφές. Ένας ιστότοπος μπορεί να είναι ένας προσωπικός ιστότοπος, ένας εταιρικός ιστότοπος για μια εταιρεία, ένας κυβερνητικός ιστότοπος, ένας ιστότοπος οργανισμού κλπ. Οι ιστότοποι είναι συνήθως αφιερωμένοι σε ένα συγκεκριμένο θέμα ή σκοπό, από την ψυχαγωγία και την κοινωνική δικτύωση μέχρι την παροχή ειδήσεων και εκπαίδευσης. Όλοι οι ιστοτόποι που είναι προσβάσιμοι από το κοινό συνιστούν το World Wide Web, ενώ ιδιωτικοί ιστότοποι, όπως ένας ιστότοπος εταιρείας για τους υπαλλήλους του, είναι συνήθως μέρος ενός ενδοδικτύου. (Freeservers, 2018)

## 1.6 Περιηγητής Ιστού (Web Browser)

Ένα πρόγραμμα περιήγησης ιστού (συνήθως αναφέρεται ως πρόγραμμα περιήγησης) είναι μια εφαρμογή λογισμικού για την πρόσβαση σε πληροφορίες στο World Wide Web. Κάθε μεμονωμένη ιστοσελίδα, εικόνα και βίντεο αναγνωρίζονται από μια ξεχωριστή διεύθυνση URL, επιτρέποντας στους browsers να τις ανακτήσουν και να τις εμφανίσουν στη συσκευή του χρήστη.

Το πρόγραμμα περιήγησης ιστού δεν είναι το ίδιο με μια μηχανή αναζήτησης, αν και οι δύο συχνά συγχέονται. Για ένα χρήστη, μια μηχανή αναζήτησης είναι απλά ένας ιστότοπος, όπως το google.com, που αποθηκεύει δεδομένα με δυνατότητα αναζήτησης για άλλους ιστότοπους. Αλλά για να συνδεθούν και να προβληθούν ιστότοποι στη συσκευή, ένας χρήστης πρέπει να έχει εγκατεστημένο ένα πρόγραμμα περιήγησης ιστού. (Orgera, 2018)



Εικόνα 1.3: Παραδείγματα δημοφιλών web browsers

Η συγκεκριμένη εργασία πραγματοποιήθηκε πάνω στον web browser Google Chrome.

## 1.7 Φιλοξενία Ιστοσελίδων (Web Hosting)

Μια υπηρεσία φιλοξενίας μέσω Διαδικτύου είναι μια υπηρεσία που εκτελεί διακομιστές Internet, επιτρέποντας σε οργανισμούς και άτομα να εκτελούν περιεχόμενο στο Διαδίκτυο. Υπάρχουν διάφορα επίπεδα υπηρεσιών και διάφορα είδη υπηρεσιών που προσφέρονται.

Ένα κοινό είδος φιλοξενίας είναι η φιλοξενία ιστοσελίδων. Οι περισσότεροι πάροχοι φιλοξενίας προσφέρουν έναν συνδυασμό υπηρεσιών. φιλοξενία ηλεκτρονικού ταχυδρομείου, για παράδειγμα. Η υπηρεσία φιλοξενίας DNS συνήθως συνοδεύεται από καταχώριση ονόματος τομέα.

Τα γενικά είδη φιλοξενίας μέσω Διαδικτύου παρέχουν ένα διακομιστή όπου οι πελάτες μπορούν να τρέξουν οτιδήποτε θέλουν (συμπεριλαμβανομένων των εξυπηρετητών ιστού και άλλων διακομιστών) και έχουν συνδέσεις στο Διαδίκτυο με καλό εύρος ζώνης. Τα πακέτα φιλοξενίας ταξινομούνται στις παρακάτω οικογένειες:

- Το Complex Managed Hosting, εφαρμόζεται τόσο σε φυσικούς αφοσιωμένους όσο και σε εικονικούς διακομιστές, με πολλές εταιρείες να επιλέγουν ένα υβριδικό (συνδυασμό φυσικής και εικονικής) φιλοξενίας. Υπάρχουν πολλές ομοιότητες μεταξύ της κανονικής και της πολύπλοκης διαχειριζόμενης φιλοξενίας, αλλά η βασική διαφορά είναι το επίπεδο διοικητικής και τεχνικής υποστήριξης που πληρώνει ο πελάτης - εξαιτίας τόσο του αυξημένου μεγέθους όσο και της πολυπλοκότητας της ανάπτυξης της υποδομής. Ο πάροχος προχωράει για να αναλάβει το μεγαλύτερο μέρος της διαχείρισης, συμπεριλαμβανομένης της ασφάλειας, της μνήμης, της αποθήκευσης και της υποστήριξης πληροφορικής. Η υπηρεσία είναι πρωτίστως ενεργή στη φύση.
- Αφιερωμένη υπηρεσία φιλοξενίας, η οποία ονομάζεται επίσης διαχειριζόμενη υπηρεσία φιλοξενίας, όπου ο πάροχος υπηρεσιών φιλοξενίας διαθέτει και διαχειρίζεται το μηχάνημα, νοικιάζει τον πλήρη έλεγχο στον πελάτη. Η διαχείριση του διακομιστή μπορεί να περιλαμβάνει παρακολούθηση ώστε ο διακομιστής να συνεχίσει να λειτουργεί αποτελεσματικά, υπηρεσίες δημιουργίας αντιγράφων ασφαλείας, εγκατάσταση επιδιορθώσεων ασφαλείας και διάφορα επίπεδα τεχνικής υποστήριξης.
- Εικονικός ιδιωτικός διακομιστής, στον οποίο χρησιμοποιείται τεχνολογία εικονικοποίησης για να επιτρέπεται η εκτέλεση πολλαπλών λογικών διακομιστών σε έναν μόνο φυσικό διακομιστή.
- Οι εγκαταστάσεις Colocation παρέχουν μόνο τη σύνδεση στο Internet, την αδιάλειπτη ισχύ και τον έλεγχο του κλίματος, αλλά αφήνουν τον πελάτη να κάνει τη δική του διαχείριση του συστήματος. το πιο ακριβό
- Η φιλοξενία Cloud, η οποία μπορεί επίσης να ονομάζεται φιλοξενία χρόνου ή κατ' απαίτηση φιλοξενίας, στην οποία ο χρήστης πληρώνει μόνο για το χρόνο και τον χρησιμοποιούμενο χώρο του συστήματος και η χωρητικότητα μπορεί

να κλιμακωθεί γρήγορα ή προς τα κάτω καθώς αλλάζουν οι απαιτήσεις πληροφορικής. (Techopedia, 2018)

## 1.8 Σύστημα Ονομάτων Τομέων (Domain Name System)

Το σύστημα ονομάτων τομέα (DNS) είναι ένα ιεραρχικό σύστημα ονομασίας για υπολογιστές, υπηρεσίες ή άλλους πόρους συνδεδεμένους στο Internet ή σε ένα ιδιωτικό δίκτυο. Συνδέει διάφορες πληροφορίες με ονόματα τομέα που έχουν εκχωρηθεί σε κάθε μία από τις συμμετέχουσες οντότητες. Πιο συγκεκριμένα, μεταφράζει πιο εύκολα αποθηκευμένα ονόματα τομέα στις αριθμητικές διευθύνσεις IP που απαιτούνται για τον εντοπισμό και τον εντοπισμό υπηρεσιών και συσκευών ηλεκτρονικών υπολογιστών με τα υποκείμενα πρωτόκολλα δικτύου. Παρέχοντας μια παγκόσμια, κατανεμημένη υπηρεσία καταλόγου, το σύστημα ονομάτων τομέα είναι ένα βασικό στοιχείο της λειτουργικότητας στο Internet, το οποίο χρησιμοποιείται από το 1985.

Το σύστημα ονομάτων τομέα αναθέτει την ευθύνη της εκχώρησης ονομάτων τομέα και της χαρτογράφησης αυτών των ονομάτων στους πόρους του Διαδικτύου, καθορίζοντας έγκυρους διακομιστές ονομάτων για κάθε τομέα. Οι διαχειριστές δικτύου μπορούν να μεταβιβάσουν εξουσία σε υποτομείς του χώρου ονομάτων που τους έχουν διατεθεί σε άλλους διακομιστές ονομάτων. Αυτός ο μηχανισμός παρέχει υπηρεσία διανομής και ανεκτικότητας σε σφάλματα και έχει σχεδιαστεί για να αποφεύγει μία μεγάλη κεντρική βάση δεδομένων.

Το Σύστημα Ονομάτων Τομέα (Domain Name System) καθορίζει επίσης την τεχνική λειτουργικότητα της υπηρεσίας βάσης δεδομένων που βρίσκεται στον πυρήνα της. Ορίζει το πρωτόκολλο DNS, μια λεπτομερή προδιαγραφή των δομών δεδομένων και των ανταλλαγών επικοινωνίας δεδομένων που χρησιμοποιούνται στο DNS, ως μέρος της σουίτας πρωτοκόλλου Internet. Ιστορικά, άλλες υπηρεσίες καταλόγου που προηγήθηκαν του DNS δεν ήταν κλιμακωτές σε μεγάλους ή παγκόσμιους καταλόγους, καθώς βασιζόνταν αρχικά σε αρχεία κειμένου, εμφανώς στο αρχείο hosts.

Το Διαδίκτυο διατηρεί δύο κύριους χώρους ονομάτων, τον ιεραρχικό χώρο του ονόματος τομέα και τους χώρους διευθύνσεων πρωτοκόλλου Internet (IP). Το σύστημα ονομάτων τομέα διατηρεί την ιεραρχία ονομάτων τομέα και παρέχει μεταφραστικές υπηρεσίες μεταξύ του και των χώρων διεύθυνσης. Οι διακομιστές ονομάτων δικτύου και ένα πρωτόκολλο επικοινωνίας εφαρμόζουν το σύστημα ονομάτων τομέα. Ένας διακομιστής ονομάτων DNS είναι ένας διακομιστής που αποθηκεύει τις εγγραφές DNS για έναν τομέα. Ένας διακομιστής ονομάτων DNS απαντά με απαντήσεις σε ερωτήματα κατά της βάσης δεδομένων του.

Οι πιο συνηθισμένοι τύποι αρχείων που είναι αποθηκευμένοι στη βάση δεδομένων DNS είναι οι Αρχές εκκίνησης (SOA), οι διευθύνσεις IP (A και AAAA), οι εναλλακτές αλληλογραφίας SMTP (MX), οι διακομιστές ονομάτων (NS), οι δείκτες για αντίστροφη αναζήτηση DNS (PTR) και ψευδώνυμα ονόματος τομέα (CNAME). Αν και δεν προορίζεται να είναι μια βάση δεδομένων γενικού σκοπού, το DNS μπορεί να αποθηκεύσει αρχεία για άλλους τύπους δεδομένων είτε για αυτόματες αναζητήσεις, όπως αρχεία DNSSEC, είτε για ανθρώπινα ερωτήματα όπως αρχεία υπεύθυνου προσώπου (RP). Ως βάση δεδομένων γενικού σκοπού, το DNS έχει χρησιμοποιηθεί επίσης για την καταπολέμηση ανεπιθύμητων ηλεκτρονικών μηνυμάτων (spam) αποθηκεύοντας μια λίστα μαύρων οπών σε πραγματικό χρόνο. Η βάση δεδομένων DNS αποθηκεύεται παραδοσιακά σε ένα δομημένο αρχείο ζώνης. (Wikipedia, 2018)

## 1.9 Εξυπηρετητής Ιστού (Web Server)

Ο εξυπηρετητής ιστού αναφέρεται σε λογισμικό διακομιστή ή σε υλικό που προορίζεται για την εκτέλεση του εν λόγω λογισμικού, το οποίο μπορεί να παρέχει περιεχόμενο στον Παγκόσμιο Ιστό. Ένας εξυπηρετητής ιστού επεξεργάζεται εισερχόμενα αιτήματα δικτύου μέσω του πρωτοκόλλου HTTP (και άλλων σχετικών πρωτοκόλλων).

Η κύρια λειτουργία του διακομιστή ιστού είναι η αποθήκευση, επεξεργασία και παράδοση ιστοσελίδων σε πελάτες. Η επικοινωνία μεταξύ πελάτη και διακομιστή πραγματοποιείται με τη χρήση του πρωτοκόλλου μεταφοράς HTTP (Hypertext Transfer Protocol). Οι σελίδες που παρέχονται είναι τα πιο συχνά έγγραφα HTML, τα οποία μπορεί να περιλαμβάνουν εικόνες, φύλλα στυλ και δέσμες ενεργειών εκτός από το περιεχόμενο κειμένου.

Ένας πράκτορας χρήστη, συνήθως ένα πρόγραμμα περιήγησης ιστού ή ανιχνευτής ιστού, ενεργοποιεί την επικοινωνία κάνοντας αίτηση για συγκεκριμένο πόρο χρησιμοποιώντας HTTP και ο διακομιστής ανταποκρίνεται με το περιεχόμενο αυτού του πόρου ή με μήνυμα σφάλματος αν δεν το κάνει. Ο πόρος είναι συνήθως ένα πραγματικό αρχείο στη δευτερεύουσα αποθήκευση του διακομιστή, αλλά αυτό δεν συμβαίνει απαραίτητα και εξαρτάται από τον τρόπο με τον οποίο υλοποιείται ο διακομιστής ιστού.

Ενώ η κύρια λειτουργία είναι η προβολή περιεχομένου, η πλήρης εφαρμογή του HTTP περιλαμβάνει επίσης τρόπους λήψης περιεχομένου από πελάτες. Αυτή η λειτουργία χρησιμοποιείται για την υποβολή φόρμας ιστού, συμπεριλαμβανομένης της μεταφόρτωσης αρχείων.

Πολλοί γενικοί διακομιστές ιστού υποστηρίζουν επίσης δέσμες ενεργειών διακομιστή χρησιμοποιώντας Active Pages Pages (ASP), PHP ή άλλες γλώσσες δέσμης ενεργειών. Αυτό σημαίνει ότι η συμπεριφορά του διακομιστή ιστού μπορεί να γραφτεί σε ξεχωριστά αρχεία, ενώ το πραγματικό λογισμικό διακομιστή παραμένει αμετάβλητο.

Συνήθως, αυτή η λειτουργία χρησιμοποιείται για τη δυναμική δημιουργία εγγράφων HTML σε αντίθεση με την επιστροφή στατικών εγγράφων. Το πρώτο χρησιμοποιείται κυρίως για την ανάκτηση ή την τροποποίηση πληροφοριών από βάσεις δεδομένων. Το τελευταίο είναι συνήθως πολύ πιο γρήγορα και πιο εύκολα αποθηκευμένο στην κρυφή μνήμη, αλλά δεν μπορεί να προσφέρει δυναμικό περιεχόμενο.

Οι διακομιστές Web δεν χρησιμοποιούνται μόνο για την εξυπηρέτηση του World Wide Web. Μπορούν επίσης να βρεθούν ενσωματωμένα σε συσκευές όπως εκτυπωτές, δρομολογητές, κάμερες web και εξυπηρετούν μόνο ένα τοπικό δίκτυο. Ο διακομιστής ιστού μπορεί στη συνέχεια να χρησιμοποιηθεί ως μέρος ενός συστήματος για την παρακολούθηση ή τη διαχείριση της εν λόγω συσκευής. Αυτό συνήθως σημαίνει ότι δεν χρειάζεται να εγκατασταθεί επιπλέον λογισμικό στον υπολογιστή-πελάτη, αφού απαιτείται μόνο ένα πρόγραμμα περιήγησης ιστού. (MDN web docs, 2018)



## Κεφάλαιο 2 – Ηλεκτρονικές αγορές μέσω χρεωστικής-πιστωτικής κάρτας

### 2.1 Λογαριασμός εμπόρου

Ένας λογαριασμός εμπόρου είναι ένας τύπος τραπεζικού λογαριασμού που επιτρέπει στις επιχειρήσεις να δέχονται πληρωμές με πολλούς τρόπους, συνήθως μέσω χρεωστικών ή πιστωτικών καρτών. Ο εμπορικός λογαριασμός δημιουργείται βάσει συμφωνίας μεταξύ αποδέκτη και τράπεζας εξαγοράς για τον διακανονισμό συναλλαγών με κάρτες πληρωμών. Σε ορισμένες περιπτώσεις ένας διαχειριστής πληρωμής, ένας ανεξάρτητος οργανισμός πωλήσεων (ISO) ή ένας πάροχος υπηρεσιών μέλους (MSP) είναι επίσης σημαντικό μέρος της εμπορικής συμφωνίας. Εάν ένας έμπορος συνάπτει μια εμπορική συμφωνία απευθείας με μια αποδέκτρια τράπεζα ή μέσω ενός website, η σύμβαση δεσμεύει συμβατικά τον έμπορο να τηρεί τους κανόνες λειτουργίας που έχουν θεσπίσει οι όμιλοι καρτών.

Σήμερα, η πλειοψηφία των συναλλαγών με πιστωτικές κάρτες αποστέλλονται ηλεκτρονικά στις τράπεζες επεξεργασίας εμπορικών συναλλαγών για εξουσιοδότηση, σύλληψη και κατάθεση. Υπάρχουν διάφορες μέθοδοι για την παρουσίαση της πώλησης πιστωτικών καρτών στο "σύστημα". Σε όλες τις περιπτώσεις, είτε ολόκληρη η μαγνητική ταινία διαβάζεται από ένα τράβηγμα μέσω ενός τερματικού / αναγνώστη πιστωτικής κάρτας, διαβάζεται ένα τσιπ υπολογιστή ή οι πληροφορίες πιστωτικής κάρτας εισάγονται χειροκίνητα σε ένα τερματικό πιστωτικής κάρτας, έναν υπολογιστή ή έναν ιστότοπο. Οι παλαιότερες μέθοδοι, που στέλνουν τα δελτία πιστωτικών καρτών σε μια τράπεζα επεξεργασίας εμπόρων ταχυδρομικώς ή με τηλεφωνική πρόσβαση σε μια μονάδα αυτόματης απόκρισης (ARU), εξακολουθούν να χρησιμοποιούνται σήμερα αλλά έχουν επισκιάσει εδώ και καιρό ηλεκτρονικές συσκευές. Αυτές οι πρώτες μέθοδοι χρησιμοποίησαν μορφές δύο τμημάτων και μια χειροκίνητη συσκευή για τη μηχανική αποτύπωση των πληροφοριών ανάγλυφου αριθμού κάρτας στις φόρμες. (Wikipedia, 2018)

### 2.2 Τερματικό πληρωμής (Payment terminal)

Ένα τερματικό πληρωμής, επίσης γνωστό ως τερματικό σημείο πώλησης, τερματικό πιστωτικής κάρτας, τερματικό EFTPOS (ή τερματικό PDQ στην Ανατολική Αφρική - το οποίο σημαίνει «Άμεσα Δεδομένα Διεργασιών»), είναι μια συσκευή που διασυνδέεται με κάρτες πληρωμών κάνουν ηλεκτρονικές μεταφορές κεφαλαίων. Υπάρχουν διάφοροι τύποι τερματικών που διατίθενται στους εμπόρους, αν και οι περισσότεροι έχουν τον ίδιο βασικό σκοπό και λειτουργίες. Επιτρέπουν σε έναν έμπορο να εισαγάγει, να σύρει ή να εισαγάγει με μη αυτόματο τρόπο τις απαιτούμενες πληροφορίες πιστωτικής / χρεωστικής κάρτας καθώς και να δεχθεί

συναλλαγές χωρίς επαφή NFC και να διαβιβάσει τα δεδομένα αυτά στον πάροχο υπηρεσιών εμπορίου για έγκριση και, τέλος, να μεταφέρει χρήματα στον έμπορο .

Τα μοντέλα υψηλότερου επιπέδου όχι μόνο επεξεργάζονται τις πιστωτικές και χρεωστικές κάρτες, αλλά χρησιμεύουν και ως ολοκληρωμένη οθόνη αφοσίωσης πελατών κατά την ολοκλήρωση της αγοράς. Δεν περιορίζονται στην ικανότητα επεξεργασίας καρτών δώρου, επιταγών, πληρωμών χωρίς επαφές και κινητών πορτοφολιών. Ορισμένα προγραμματίζονται επίσης να δέχονται κάρτες πίστης στο κατάστημα ή να επιτρέπουν στον πελάτη να χρησιμοποιεί το πληκτρολόγιο για να εισάγει τις πληροφορίες (π.χ. αριθμός τηλεφώνου) για εξαργύρωση σημείων. Οι οθόνες POS επιτρέπουν επίσης στους διαφημιζόμενους να διαφημίζουν κοντά στο μητρώο όταν δεν χρησιμοποιείται το τερματικό. Κατά τη διάρκεια της διαδικασίας πληρωμής, πολλά τερματικά έχουν οριστεί να εμφανίζουν μια λίστα στοιχείων που έχουν αγοραστεί μαζί με το τρέχον σύνολο. Η λειτουργία αυτή μπορεί να απενεργοποιηθεί ή μπορεί να χρησιμοποιηθεί για τη συμπλήρωση μιας αποκλειστικής οθόνης που αναφέρει τα στοιχεία που αγοράζονται. Ορισμένα καταστήματα χρησιμοποιούν επίσης το τερματικό για να βλέπουν οι πελάτες και να συμφωνούν με τους όρους μιας εγγύησης προϊόντος. Όπως τα ATMs, πολλά τερματικά POS διαθέτουν επίσης ανυψωμένα απτικά κουμπιά και υποδοχή ακουστικών που επιτρέπουν στον τυφλό να ολοκληρώσει ακουστικά τη διαδικασία πληρωμής.

Η πλειοψηφία των τερματικών καρτών μεταδίδει δεδομένα μέσω μιας τυπικής τηλεφωνικής γραμμής ή μέσω σύνδεσης στο διαδίκτυο. Μερικές επίσης έχουν τη δυνατότητα να αποθηκεύουν κρυφά τα δεδομένα συναλλαγών που πρέπει να μεταδίδονται στον επεξεργαστή πύλης όταν υπάρχει διαθέσιμη σύνδεση. Το κύριο μειονέκτημα για αυτό είναι ότι δεν υπάρχει άμεση εξουσιοδότηση κατά τη στιγμή της επεξεργασίας της κάρτας, η οποία μπορεί στη συνέχεια να οδηγήσει σε αποτυχημένες πληρωμές. Τα ασύρματα τερματικά μεταδίδουν δεδομένα κάρτας μέσω Bluetooth, Wi-Fi ή ακόμα και δορυφορικών δικτύων σε απομακρυσμένες περιοχές και σε αεροσκάφη επί του σκάφους. Ένας έμπορος μπορεί να αντικαταστήσει τη λειτουργικότητα του εξειδικευμένου υλικού τερματικών πιστωτικών καρτών χρησιμοποιώντας μια εφαρμογή τερματικού που εκτελείται σε έναν υπολογιστή ή μια κινητή συσκευή, όπως ένα smartphone. Συνήθως συνεργάζονται με ειδικούς αναγνώστες υλικού που μπορούν να μεταφέρουν δεδομένα μαγνητικής λωρίδας στην εφαρμογή, ενώ υπάρχουν και άλλες που λειτουργούν επίσης με έξυπνες κάρτες (χρησιμοποιώντας τεχνολογία όπως το EMV), παρόλο που σπάνια συμβαίνει σε αναγνώστες smartphone. Σε περίπτωση που δεν είναι διαθέσιμο το απαραίτητο υλικό, οι εφαρμογές αυτές συνήθως υποστηρίζουν τη μη αυτόματη εισαγωγή του αριθμού της κάρτας και άλλων δεδομένων. Επιπλέον, όλο και περισσότερες συσκευές αρχίζουν να προσφέρουν ενσωματωμένη τεχνολογία RFID ή NFC για να φιλοξενήσουν μεθόδους πληρωμής χωρίς επαφή ή κινητές συσκευές, συχνά χωρίς να χρειάζονται πρόσθετο εξωτερικό υλικό.

Με τη μετάβαση στη χρήση των τερματικών καρτών για την απευθείας λήψη πληροφοριών κάρτας αντί για τη μη αυτόματη εισαγωγή των στοιχείων της κάρτας, οι έμποροι επωφελούνται από την αποτελεσματικότητα των μειωμένων χρόνων επεξεργασίας συναλλαγών. Από την άποψη της ασφάλειας, οι μεγάλοι κατασκευαστές τερματικών καρτών προσφέρουν συνήθως λογισμικό που επιτρέπει την κρυπτογράφηση των δεδομένων από το τέλος μέχρι το τέλος. Παρ' όλα αυτά, υπήρξαν ορισμένες περιπτώσεις κακόβουλου λογισμικού POS pin pin. Σε χώρες όπως οι ΗΠΑ, όπου οι κάρτες μαγνητικής λωρίδας δεν έχουν καταργηθεί πλήρως, υπήρξε επίσης συχνότητα καθαρίσματος σε τερματικά καρτών. Για αυτό, τα αυτόνομα τερματικά πληρωμής θεωρούνται ανώτερα από τις καταγεγραμμένες μεθόδους πληρωμής επειδή δεν απαιτούν από τους κατόχους των καταστημάτων να πάρουν στην κατοχή τους την κάρτα του πελάτη. Το τερματικό πληρωμής προσφέρει τα εξής:

- Εισαγωγή κλειδιού (για την παραγγελία πελάτη που δεν υπάρχει ταχυδρομική και τηλεφωνική παραγγελία)
- Συμβουλές / φιλοδωρήματα
- Επιστροφές και προσαρμογές
- Διακανονισμοί (συμπεριλαμβανομένης της αυτόματης)
- Προ-εξουσιοδότηση
- Πληρωμές με συσκευές με near-field communication
- Απομακρυσμένη αρχικοποίηση και ενημέρωση λογισμικού
- Ενσωμάτωση POS
- Ημι-ολοκληρωμένο POS
- Πολλαπλές εμπορικές δυνατότητες
- Άδεια πέννας ή PIN από τον πελάτη
- Λειτουργία προσαύξησης
- Λειτουργία ασφαλούς κωδικού πρόσβασης (Wikipedia, 2018)

### 2.2.1 Ιστορία του τερματικού πληρωμής

Πριν από την ανάπτυξη των τερματικών πληρωμών, οι έμποροι χρησιμοποιούσαν χειροκίνητους εκτυπωτές (επίσης γνωστούς ως μηχανές ZipZap) για να συλλαμβάνουν τις πληροφορίες μιας πιστωτικής κάρτας σε χαρτί με χαρτί με αντίγραφο χαρτιού άνθρακα. Αυτοί οι χαρτοφύλακες έπρεπε να μεταφερθούν στην τράπεζα για επεξεργασία. Αυτή ήταν μια δυσκίνητη και χρονοβόρα διαδικασία.

Τα τερματικά σημείων πώλησης εμφανίστηκαν το 1979, όταν η Visa εισήγαγε ένα ογκώδες τερματικό λήψης ηλεκτρονικών δεδομένων, το οποίο ήταν το πρώτο τερματικό πληρωμής. Την ίδια χρονιά προστέθηκαν μαγνητικές ρίγες σε πιστωτικές κάρτες για πρώτη φορά. Αυτό επέτρεψε την ηλεκτρονική καταγραφή των πληροφοριών της κάρτας και οδήγησε στην ανάπτυξη τερματικών πληρωμών.

Μία από τις πρώτες εταιρείες που παρήγαγαν ειδικά τερματικά πληρωμής ήταν η Verifone. Ξεκίνησε το 1981 στη Χαβάη ως μικρή ηλεκτρονική εταιρεία. Το 1983 εισήγαγαν τη σειρά τερματικών ZON, η οποία θα γίνει το πρότυπο για τα σύγχρονα τερματικά πληρωμής.

Ο Ούγγρος George Wallner, γεννημένος στο Σίδνεϊ της Αυστραλίας ίδρυσε την αντίπαλη εταιρεία Hypercom το 1978 και το 1982 άρχισε να παράγει αποκλειστικά τερματικά πληρωμής και πήγε για να κυριαρχήσει στην περιοχή του νοτιο-ειρηνικού. Το 1988 η εταιρεία υπέγραψε συμφωνία με την American Express για την παροχή των τερματικών σταθμών σε αυτές στις ΗΠΑ. Για να εδραιώσει τη συμφωνία, η Hypercom μετέφερε την έδρα της από την Αυστραλία στην Αριζόνα των ΗΠΑ. Στη συνέχεια αντιμετώπισε ανταγωνισμό με την Verifone για την εγχώρια αγορά.

Ύστερα από μια δεκαετία το 1994 η εταιρεία Lipman Electronic Engineering, Ltd. ιδρύθηκε στο Ισραήλ. Η Lipman κατασκευάζει τη γραμμή Nurit των τερματικών επεξεργασίας. Λόγω της ήδη σταθερής θέσης της Verifone στη βιομηχανία επεξεργασίας πληρωμών, όταν ιδρύθηκε η Lipman, η Lipman στόχευσε μια αναξιόποιτη θέση στη βιομηχανία μεταποίησης. Ενώ η Lipman κατέχει περίπου 10% μερίδιο σε ενσύρματους τερματικούς σταθμούς πιστωτικών καρτών, είναι ο αδιαμφισβήτητος ηγέτης με περισσότερο από 95% μερίδιο στα τερματικά ασύρματης επεξεργασίας στα τέλη της δεκαετίας του 1990.

Η Verifone θα αποκτούσε αργότερα και τους δύο αυτούς σημαντικούς ανταγωνιστές, αποκτώντας την Lipman το 2006 και το τμήμα πληρωμών της επιχείρησης Hypercom, συμπεριλαμβανομένης της μάρκας της το 2011.

Το 1980 ο Jean-Jacques Poutrel και ο Michel Malhouitre ίδρυσαν την Ingenico στη Γαλλία και ανέπτυξαν το πρώτο τερματικό πληρωμών τους το 1984. Η μονάδα E & A της Βαρκελώνης θα οδηγήσει στην ανάπτυξη τερματικών πληρωμών για την επόμενη δεκαετία. Ingenico μέσω ορισμένων εξαγορών θα κυριαρχούσε στην ευρωπαϊκή αγορά των τερματικών πληρωμών. Απόκτησαν τη γαλλική βάση Bull και τη Βρετανία που βασίζονται De La Rue δραστηριότητα πληρωμής τερματικού σταθμού, καθώς και γερμανική Epos το 2001.

Αρχικά, συλλέχθηκαν πληροφορίες από τη μαγνητική ταινία στο πίσω μέρος της κάρτας, μεταφέροντας την κάρτα μέσω του τερματικού. Στα τέλη της δεκαετίας του 1990, αυτό άρχισε να αντικαθίσταται από έξυπνες κάρτες όπου το τσιπ ήταν ενσωματωμένο στην κάρτα. Αυτό έγινε για πρόσθετη ασφάλεια και απαιτούσε την εισαγωγή της κάρτας στο τερματικό της πιστωτικής κάρτας. Στα τέλη της δεκαετίας του 1990 και στις αρχές της δεκαετίας του 2000 εισήχθησαν συστήματα επικοινωνίας χωρίς επαφή και τα τερματικά πληρωμών ενημερώθηκαν ώστε να συμπεριλαμβάνουν τη δυνατότητα ανάγνωσης αυτών των καρτών χρησιμοποιώντας τεχνολογία near field communication (NFC). (Wikipedia, 2018)

## 2.3 Μονάδα αυτοματοποιημένης απόκρισης (ARU)

Μια ARU (γνωστή και ως εξουσιοδότηση φωνής, σύλληψη και κατάθεση) επιτρέπει τη μη αυτόματη είσοδο με κλειδί και την επακόλουθη εξουσιοδότηση μιας πιστωτικής κάρτας μέσω κυψελωτού ή χερσαίου τηλεφώνου. Με αυτήν τη μέθοδο, ένας έμπορος συνήθως αποτυπώνει την κάρτα του πελάτη του με ένα imprinter για να δημιουργήσει μια απόδειξη πελάτη και αντίγραφο εμπόρου, και στη συνέχεια επεξεργάζεται τη συναλλαγή ακαριαία μέσω του τηλεφώνου. (Wikipedia, 2018)

## 2.4 Πύλη πληρωμής (Payment gateway)

Μια πύλη πληρωμής είναι μια υπηρεσία εμπορίου που παρέχεται από έναν πάροχο υπηρεσιών εφαρμογών ηλεκτρονικού εμπορίου που εξουσιοδοτεί την επεξεργασία πιστωτικών καρτών ή άμεσων πληρωμών για ηλεκτρονικές επιχειρήσεις, ηλεκτρονικούς εμπόρους λιανικής πώλησης, επιχειρησιακά μοντέλα «bricks and clicks». Η πύλη πληρωμής μπορεί να παρέχεται από μια τράπεζα στους πελάτες της, αλλά μπορεί να παρέχεται από έναν εξειδικευμένο πάροχο χρηματοπιστωτικών υπηρεσιών ως ξεχωριστή υπηρεσία, όπως ένας πάροχος υπηρεσιών πληρωμών.

Μια πύλη πληρωμών διευκολύνει μια συναλλαγή πληρωμής με τη μεταφορά πληροφοριών μεταξύ μιας πύλης πληρωμής (όπως μια ιστοσελίδα, κινητό τηλέφωνο ή υπηρεσία αλληλεπιδραστικής φωνητικής απόκρισης) και τον επεξεργαστή του εμπρός ή την τράπεζα εξαγοράς. (Wikipedia, 2018)

### 2.4.1 Διαδικασία συναλλαγών

Όταν ένας πελάτης παραγγέλλει ένα προϊόν από έναν έμπορο με δυνατότητα πύλης πληρωμής, η πύλη πληρωμής εκτελεί μια ποικιλία εργασιών για τη διεκπεραίωση της συναλλαγής.

Ένας πελάτης τοποθετεί μια παραγγελία στον ιστότοπο πατώντας το κουμπί "Υποβολή παραγγελίας" ή αντίστοιχο κουμπί ή ίσως εισάγει τα στοιχεία της κάρτας του χρησιμοποιώντας μια υπηρεσία αυτόματου τηλεφωνητή.

Εάν η παραγγελία γίνεται μέσω ιστότοπου, ο φυλλομετρητής του πελάτη κρυπτογραφεί τις πληροφορίες που πρέπει να αποσταλούν μεταξύ του προγράμματος περιήγησης και του διακομιστή ιστού του εμπόρου. Μεταξύ άλλων μεθόδων, αυτό μπορεί να γίνει μέσω κρυπτογράφησης SSL (Secure Socket Layer). Η πύλη πληρωμής μπορεί να επιτρέπει την απευθείας αποστολή των δεδομένων συναλλαγών από το πρόγραμμα περιήγησης του πελάτη στην πύλη, παρακάμπτοντας τα συστήματα του εμπόρου. Αυτό μειώνει τις υποχρεώσεις συμμόρφωσης του

Κώδικα Ασφαλείας Δεδομένων Καρτών Πληρωμών (PCI DSS) του εμπόρου χωρίς να ανακατευθύνει τον πελάτη από τον ιστότοπο.

Ο έμπορος διαβιβάζει στη συνέχεια τις λεπτομέρειες της συναλλαγής στην πύλη πληρωμής. Αυτή είναι μια κρυπτογραφημένη σύνδεση (SSL) στο διακομιστή πληρωμών που φιλοξενείται από την πύλη πληρωμής.

Η πύλη πληρωμής μετατρέπει το μήνυμα από XML σε ISO 8583 ή σε μορφή παραλλαγής μηνύματος (μορφή που γίνεται κατανοητή από τους μεταγωγείς EFT) και στη συνέχεια διαβιβάζει τις πληροφορίες συναλλαγής στον επεξεργαστή πληρωμής που χρησιμοποιείται από την τράπεζα εξαγοράς του εμπόρου.

Ο επεξεργαστής πληρωμής διαβιβάζει τις πληροφορίες συναλλαγής στην ένωση καρτών (δηλ. : Visa / MasterCard / American Express). Εάν χρησιμοποιήθηκε μια κάρτα American Express ή Discover, τότε η ένωση καρτών ενεργεί επίσης ως τράπεζα έκδοσης και παρέχει άμεσα απάντηση εγκεκριμένου ή απορριφθέντος στην πύλη πληρωμής. Διαφορετικά (π.χ. κάρτα MasterCard ή Visa) χρησιμοποιήθηκε, η σύνδεση καρτών μεταφέρει τη συναλλαγή στη σωστή τράπεζα έκδοσης καρτών. Η τράπεζα έκδοσης πιστωτικών καρτών λαμβάνει την αίτηση εξουσιοδότησης, επαληθεύει τη διαθέσιμη πίστωση ή χρέωση και στη συνέχεια αποστέλλει μια απάντηση στον επεξεργαστή (με την ίδια διαδικασία με την αίτηση εξουσιοδότησης) με έναν κωδικό απάντησης δηλαδή έγκριση ή απόρριψη. Εκτός από την ανακοίνωση της τύχης της αίτησης εξουσιοδότησης, ο κωδικός απάντησης χρησιμοποιείται επίσης για τον ορισμό του λόγου για τον οποίο η συναλλαγή απέτυχε δηλαδή ανεπαρκής χρηματοδότηση ή τραπεζικός σύνδεσμος δεν είναι διαθέσιμος. Εν τω μεταξύ, ο εκδότης της πιστωτικής κάρτας κατέχει εξουσιοδότηση που συνδέεται με αυτόν τον έμπορο και τον καταναλωτή για το εγκεκριμένο ποσό. Αυτό μπορεί να επηρεάσει την ικανότητα του καταναλωτή να δαπανά περαιτέρω επειδή μειώνει τη διαθέσιμη πιστωτική γραμμή ή παύει να κατέχει ένα μέρος των κεφαλαίων σε έναν χρεωστικό λογαριασμό.

Ο επεξεργαστής προωθεί την απάντηση εξουσιοδότησης στην πύλη πληρωμής. Η πύλη πληρωμής λαμβάνει την απάντηση και την προωθεί στον ιστότοπο ή οποιαδήποτε διεπαφή χρησιμοποιήθηκε για την επεξεργασία της πληρωμής, όπου ερμηνεύεται ως σχετική απάντηση και μεταφέρεται ξανά στον έμπορο και τον κάτοχο της κάρτας. Αυτό είναι γνωστό ως Εξουσιοδότηση ή "Auth". Ολόκληρη η διαδικασία συνήθως διαρκεί 2-3 δευτερόλεπτα.

Ο έμπορος τότε εκπληρώνει την εντολή και η παραπάνω διαδικασία μπορεί να επαναληφθεί, αλλά αυτή τη φορά για να "Εκκαθαρίσει" την εξουσιοδότηση με την ολοκλήρωση της συναλλαγής. Τυπικά, το "Clear" ξεκινά μόνο αφού ο έμπορος έχει εκπληρώσει τη συναλλαγή δηλαδή αποστολή της παραγγελίας. Αυτό οδηγεί στην εκκαθάριση της τράπεζας από την τράπεζα της έκδοσης και τα προετοιμάζει για να τακτοποιήσει με την τράπεζα που αποκτά εμπόρους.

Ο έμπορος υποβάλλει όλες τις εγκεκριμένες εγκρίσεις τους, σε "παρτίδα" (τέλος της ημέρας), στην τράπεζα που τους απέκτησε για διακανονισμό μέσω του μεταποιητή

τους. Αυτό συνήθως μειώνει ή "εκκαθαρίζει" την αντίστοιχη "Auth", αν δεν έχει ρητά "Εκκαθαριστεί".

Η απορροφώσα τράπεζα υποβάλλει το αίτημα διακανονισμού παρτίδας του εκδότη της πιστωτικής κάρτας.

Ο εκδότης της πιστωτικής κάρτας καταβάλλει μια πληρωμή διακανονισμού στην απορροφώσα τράπεζα την επόμενη ημέρα στις περισσότερες περιπτώσεις.

Στη συνέχεια, η απορροφούσα τράπεζα καταθέτει το σύνολο των εγκεκριμένων κεφαλαίων στο λογαριασμό που ορίζει ο έμπορος την ίδια ή την επόμενη ημέρα. Αυτό θα μπορούσε να είναι ένας λογαριασμός με την απορροφούσα τράπεζα αν ο έμπορος κάνει τραπεζικές συναλλαγές με την ίδια τράπεζα ή λογαριασμό σε άλλη τράπεζα. Η όλη διαδικασία από την εξουσιοδότηση έως τη διευθέτηση σε χρηματοδότηση συνήθως διαρκεί 3 ημέρες.

Πολλές πύλες πληρωμών παρέχουν επίσης εργαλεία για την αυτόματη προβολή παραγγελιών για απάτη και τον υπολογισμό του φόρου σε πραγματικό χρόνο πριν από την αποστολή της αίτησης εξουσιοδότησης στον επεξεργαστή. Τα εργαλεία για την ανίχνευση απάτης περιλαμβάνουν τη γεωγραφική κατανομή, την ανάλυση ταχέων ταχύτητας, τις αναζητήσεις λίστας OFAC, τις αναζητήσεις μαύρης λίστας, την επαλήθευση διεύθυνσης παράδοσης, την τεχνολογία εκτύπωσης δακτυλικών αποτυπωμάτων, την ανίχνευση μορφοποίησης ταυτότητας και βασικούς ελέγχους AVS. (Wikipedia, 2018)

## Κεφάλαιο 3 – Εργαλεία Υλοποίησης

### 3.1 Apache HTTP Server



Εικόνα 3.1: Το λογότυπο του Apache HTTP Server

Όπως ειπώθηκε στο προηγούμενο κεφάλαιο, ο Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού καθώς λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X υποστηρίζοντας και γλώσσες προγραμματισμού όπως PHP, MySQL etc. Επίσης, χρησιμοποιείται και σε τοπικά δίκτυα ως διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσεων Δεδομένων όπως η Oracle και η MySQL.

Αρχικά βασισμένο στον εξυπηρετητή NCSA HTTPd, η ανάπτυξη του Apache ξεκίνησε στις αρχές του 1995, όταν οι εργασίες πάνω στον κώδικα του NCSA σταμάτησαν. Ο Apache έπαιξε καθοριστικό ρόλο στην αρχική ανάπτυξη του World Wide Web, ξεπερνώντας γρήγορα το NCSA HTTPd ως κυρίαρχο διακομιστή HTTP και παρέμεινε πιο δημοφιλής από τον Απρίλιο του 1996 μέχρι σήμερα. (Mitchell, 2018)



## 3.2 Λογισμικό ΧΑΜΡΡ



Εικόνα 3.2: Λογότυπο του ΧΑΜΡΡ

Το ΧΑΜΡΡ είναι ένα πακέτο στοίβας λύσεων διαδικτυακών διακομιστών ελεύθερου ανοικτού λογισμικού που αναπτύχθηκε από την Apache Friends και αποτελείται κυρίως από τον διακομιστή HTTP του Apache, τη βάση δεδομένων MariaDB και τους διερμηνείς για σενάρια γραμμένα στις γλώσσες προγραμματισμού PHP και Perl. Είναι μια απλή, ελαφριά διανομή Apache που καθιστά εξαιρετικά εύκολο για τους προγραμματιστές να δημιουργήσουν έναν τοπικό διακομιστή ιστού για δοκιμές και σκοπούς ανάπτυξης. Όλα τα απαραίτητα για τη δημιουργία ενός διακομιστή web - εφαρμογή διακομιστή (Apache), βάση δεδομένων (MariaDB), και γλώσσα δέσμης ενεργειών (PHP) - περιλαμβάνεται σε ένα εξαγωγίμο αρχείο. Το ΧΑΜΡΡ είναι επίσης cross-platform λογισμικό, πράγμα που σημαίνει ότι λειτουργεί εξίσου καλά σε Linux, Mac και Windows. Δεδομένου ότι οι περισσότερες πραγματικές εφαρμογές διακομιστή ιστού χρησιμοποιούν τα ίδια στοιχεία με το ΧΑΜΡΡ, κάνει την μετάβαση από έναν τοπικό διακομιστή δοκιμών σε ένα ζωντανό διακομιστή εξαιρετικά εύκολη επίσης. Τα αρχικά του σημαίνουν τα εξής:

- X: πηγάζει από το «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας
- A: Apache HTTP Server
- M: Βάση Δεδομένων MariaDB. Παλιότερα ήταν η Βάση Δεδομένων MySQL
- P: Γλώσσα Προγραμματισμού PHP
- P: Γλώσσα Προγραμματισμού Perl

Οι σχεδιαστές της ΧΑΜΡΡ σκόπευαν να το χρησιμοποιήσουν μόνο ως εργαλείο ανάπτυξης, ώστε οι σχεδιαστές ιστοσελίδων και οι προγραμματιστές να δοκιμάσουν τη δουλειά τους στους δικούς τους υπολογιστές χωρίς πρόσβαση στο Διαδίκτυο. Για να γίνει αυτό όσο το δυνατόν πιο εύκολο, πολλά σημαντικά χαρακτηριστικά ασφαλείας απενεργοποιούνται από προεπιλογή. Το ΧΑΜΡΡ έχει τη δυνατότητα να

εξυπηρετεί ιστοσελίδες στον Παγκόσμιο Ιστό. Παρέχεται ειδικό εργαλείο για την προστασία με κωδικό πρόσβασης των σημαντικότερων τμημάτων της συσκευασίας.

Το XAMPP παρέχει επίσης υποστήριξη για τη δημιουργία και τον χειρισμό βάσεων δεδομένων σε MariaDB και SQLite μεταξύ άλλων.

Μόλις εγκατασταθεί το XAMPP, είναι δυνατό να αντιμετωπιστεί ένας τοπικός ιστότοπος, όπως ένας απομακρυσμένος κεντρικός υπολογιστής, μέσω σύνδεσης μέσω ενός προγράμματος-πελάτη FTP. Χρησιμοποιώντας ένα πρόγραμμα όπως το FileZilla έχει πολλά πλεονεκτήματα κατά την εγκατάσταση ενός συστήματος διαχείρισης περιεχομένου (CMS) όπως το Joomla ή το WordPress. Είναι επίσης δυνατή η σύνδεση με το localhost μέσω FTP με έναν συντάκτη HTML. (Portableapps, 2018)

### 3.3 Notepad ++

Το Notepad ++ είναι ένας δωρεάν επεξεργαστής πηγαίου κώδικα και αντικατάσταση του Notepad που υποστηρίζει πολλές γλώσσες. Λειτουργώντας στο περιβάλλον MS Windows, η χρήση του διέπεται από την άδεια GPL.

Με βάση το ισχυρό στοιχείο επεξεργασίας Scintilla, το Notepad ++ είναι γραμμένο σε C ++ και χρησιμοποιεί καθαρό Win32 API και STL που εξασφαλίζει υψηλότερη ταχύτητα εκτέλεσης και μικρότερο μέγεθος προγράμματος. Διαθέτει εκτύπωση σύνταξης, αναδίπλωση κώδικα και περιορισμένη αυτόματη συμπλήρωση για προγραμματισμό, δέσμες ενεργειών και γλώσσες σήμανσης, αλλά όχι έξυπνη ολοκλήρωση κώδικα ή έλεγχος σύνταξης.

Το Notepad ++ έχει υποστήριξη για μακροεντολές και plugins και έχει επισημανθεί για την ισχυρή αρχιτεκτονική plugin, η οποία επέτρεψε την ενσωμάτωση διαφόρων νέων λειτουργιών στο πρόγραμμα. Επί του παρόντος, αναπτύσσονται πάνω από 140 συμβατά πρόσθετα για το Notepad ++, 10 από τα οποία περιλαμβάνονται από προεπιλογή στο πρόγραμμα. (Notepad ++, 2018)

### 3.4 Γλώσσα Προγραμματισμού HTML (Hyper Text Markup Language)



Εικόνα 3.3: Λογότυπο της HTML5

Η HTML είναι μια γλώσσα σήμανσης υπερκειμένου υπολογιστή σχεδιασμένη για να επιτρέψει τη δημιουργία ιστότοπου. Αυτές οι ιστοσελίδες μπορούν στη συνέχεια να προβληθούν από οποιονδήποτε άλλο συνδεδεμένο με το Διαδίκτυο. Είναι σχετικά εύκολο να μάθει, με τα βασικά να είναι προσβάσιμα στους περισσότερους ανθρώπους σε μια συνεδρίαση, και αρκετά ισχυρό σε αυτό που σας επιτρέπει να δημιουργήσετε. Υιοθετείται συνεχώς αναθεώρηση και εξέλιξη για να ανταποκριθεί στις απαιτήσεις και τις απαιτήσεις του αναπτυσσόμενου κοινού στο διαδίκτυο υπό την καθοδήγηση του «W3C, του οργανισμού που είναι επιφορτισμένος με το σχεδιασμό και τη διατήρηση της γλώσσας Τα προγράμματα περιήγησης Web λαμβάνουν έγγραφα HTML από έναν διακομιστή ιστού ή από τοπικό αποθηκευτικό χώρο και αποδίδουν τα έγγραφα σε ιστοσελίδες πολυμέσων. Η HTML περιγράφει τη δομή μιας ιστοσελίδας σημασιολογικά και αρχικά περιελάμβανε στοιχεία για την εμφάνιση του εγγράφου. (Shannon, 2018)

Τα στοιχεία HTML είναι τα δομικά στοιχεία των σελίδων HTML. Με κατασκευές HTML, εικόνες και άλλα αντικείμενα, όπως διαδραστικές φόρμες, μπορούν να ενσωματωθούν στην απόδοση της σελίδας. Η HTML παρέχει ένα μέσο για τη δημιουργία δομημένων εγγράφων με την ένδειξη διαρθρωτικής σημασιολογίας για κείμενο όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους, εισαγωγικά και άλλα στοιχεία. Τα στοιχεία HTML οριοθετούνται από ετικέτες, γραμμένες με γωνιακές παρενθέσεις. Ετικέτες όπως `<img />` και `<input />` εισάγουν άμεσα περιεχόμενο στη σελίδα. Άλλες ετικέτες όπως το `<p>` περιβάλλουν και παρέχουν πληροφορίες σχετικά με το κείμενο του εγγράφου και ενδέχεται να περιλαμβάνουν και άλλες ετικέτες ως υπο-στοιχεία. Τα προγράμματα περιήγησης δεν εμφανίζουν τις ετικέτες HTML, αλλά τα χρησιμοποιούν για την ερμηνεία του περιεχομένου της σελίδας.

Η HTML μπορεί να ενσωματώσει προγράμματα γραμμένα σε μια γλώσσα δέσμης ενεργειών όπως JavaScript, η οποία επηρεάζει τη συμπεριφορά και το περιεχόμενο των ιστοσελίδων. Η συμπερίληψη του CSS ορίζει την εμφάνιση και τη διάταξη του περιεχομένου. (Duckett, 2018)

### 3.4.1 Έκδοση HTML5

Πρόκειται για την πέμπτη και τρέχουσα σημαντική έκδοση του προτύπου HTML, η οποία και χρησιμοποιήθηκε για την πραγματοποίηση της πτυχιακής. Δημοσιεύθηκε τον Οκτώβριο του 2014 από την Κοινοπραξία World Wide Web (W3C) για τη βελτίωση της γλώσσας με την υποστήριξη των πιο πρόσφατων πολυμέσων, διατηρώντας ταυτόχρονα ευανάγνωστη από τους ανθρώπους και κατανοητά από υπολογιστές και συσκευές, browsers, parsers κ.λπ. Η HTML5 προορίζεται να περιλαμβάνει όχι μόνο HTML 4, αλλά και XHTML 1 και DOM Level 2.

Η HTML5 περιλαμβάνει λεπτομερή μοντέλα επεξεργασίας για την ενθάρρυνση περισσότερων διαλειτουργικών υλοποιήσεων. επεκτείνει, βελτιώνει και εξορθολογεί τη διαθέσιμη σήμανση για τα έγγραφα και εισάγει διεπαφές προγραμματισμού σήμανσης και εφαρμογών (API) για σύνθετες εφαρμογές ιστού. Για τους ίδιους λόγους, η HTML5 είναι επίσης υποψήφια για κινητές εφαρμογές πολλαπλών πλατφορμών, επειδή περιλαμβάνει χαρακτηριστικά γνωρίσματα σχεδιασμένα με συσκευές χαμηλής ισχύος.

Περιλαμβάνονται πολλά νέα συντακτικά χαρακτηριστικά. Για να συμπεριλάβει και να χειριστεί περιεχόμενο πολυμέσων και γραφικών, τα νέα στοιχεία <video>, <audio> και <canvas> προστέθηκαν και υποστήριξη για περιεχόμενο SVG και MathML για μαθηματικούς τύπους. Για να εμπλουτίσουν το σημασιολογικό περιεχόμενο των εγγράφων, προστίθενται νέα στοιχεία δομής σελίδας όπως <main>, <section>, <article>, <header>, <footer>, <aide>, <nav> και <figure>. Εισάγονται νέα χαρακτηριστικά, ορισμένα στοιχεία και ιδιότητες έχουν καταργηθεί και άλλα όπως <a>, <cite> και <menu> έχουν αλλάξει, επαναπροσδιοριστεί ή τυποποιηθεί.

Τα API και το μοντέλο αντικειμένου εγγράφου (DOM) αποτελούν πλέον θεμελιώδη μέρη της προδιαγραφής HTML5 και η HTML5 ορίζει επίσης καλύτερα την επεξεργασία για τυχόν μη έγκυρα έγγραφα. (Duckett, 2018)

### 3.5 Γλώσσα Προγραμματισμού PHP (Hypertext Preprocessor)



Εικόνα 3.4: Λογότυπο της PHP

Η PHP (Hypertext Preprocessor) είναι μια ευρέως χρησιμοποιούμενη γλώσσα γενικής χρήσης scripting ανοιχτού κώδικα που είναι ιδιαίτερα κατάλληλη για ανάπτυξη ιστοσελίδων και μπορεί να ενσωματωθεί σε HTML.

Αντί για πολλές εντολές για την εξαγωγή HTML (όπως φαίνεται στο C ή Perl), οι σελίδες PHP περιέχουν HTML με ενσωματωμένο. Ο κώδικας PHP περικλείεται σε ειδικές οδηγίες έναρξης και λήξης `<?Php και ?>` που επιτρέπουν την μετάβαση και έξοδο από τη λειτουργία της PHP.

Αυτό που διακρίνει την PHP από γλώσσες σαν την JavaScript από την πλευρά του πελάτη είναι ότι ο κώδικας εκτελείται στον εξυπηρετητή, δημιουργώντας HTML που στη συνέχεια αποστέλλεται στον πελάτη. Ο πελάτης θα λάβει τα αποτελέσματα της εκτέλεσης του σεναρίου, αλλά δεν θα ξέρει τι είναι ο υποκείμενος κώδικας.

Τα καλύτερα πράγματα στη χρήση της PHP είναι ότι είναι εξαιρετικά απλή για έναν νεοφερμένο, αλλά προσφέρει πολλά προηγμένα χαρακτηριστικά για έναν επαγγελματία προγραμματιστή. (PHP.net, 2018)

#### 3.5.1 Μεταβλητές Περιόδων Σύνδεσης (Sessions)

Όταν κάποιος εργάζεται με μια εφαρμογή, την ανοίγει, προχωρά σε κάποιες αλλαγές και στη συνέχεια την κλείνει. Αυτό μοιάζει πολύ με μια περίοδο λειτουργίας. Ο υπολογιστής αναγνωρίζει τον χρήστη και ξέρει πότε ξεκινάει και πότε τελειώνει η εφαρμογή. Αλλά στο διαδίκτυο ο διακομιστής ιστού δεν γνωρίζει τον χρήστη και τις πράξεις του επειδή η διεύθυνση HTTP δεν διατηρεί την κατάσταση. Οι μεταβλητές περιόδου σύνδεσης (Sessions) επιλύουν αυτό το πρόβλημα αποθηκεύοντας τις πληροφορίες χρηστών που χρησιμοποιούνται σε πολλές σελίδες (π.χ. όνομα χρήστη, αγαπημένο χρώμα κ.λπ.). Από προεπιλογή, οι μεταβλητές περιόδου λειτουργίας διαρκούν έως ότου ο χρήστης κλείσει το πρόγραμμα περιήγησης. Έτσι οι μεταβλητές

περιόδου σύνδεσης περιέχουν πληροφορίες για έναν μόνο χρήστη και είναι διαθέσιμες σε όλες τις σελίδες μιας εφαρμογής. Μία περίοδος σύνδεσης ξεκινά με τη συνάρτηση `session_start()` και τελειώνει μέσω της συνάρτησης `session_destroy()` ή `session_unset()`. Οι μεταβλητές περιόδου σύνδεσης ορίζονται με την παγκόσμια μεταβλητή PHP: `$_SESSION`. (W3Schools, PHP Sessions, 2018)

Δύο κοινώς χρησιμοποιούμενες μέθοδοι για απάντηση αιτήματος-απόκρισης μεταξύ πελάτη και διακομιστή είναι η GET (`$_GET`) και POST (`$_POST`).

- GET - Ζητά δεδομένα από συγκεκριμένο πόρο.
- POST - Υποβολή δεδομένων προς επεξεργασία σε συγκεκριμένο πόρο.

Στην περίπτωση αιτήματος HTTP GET το query string (ζεύγη ονόματος / τιμής) αποστέλλεται στη διεύθυνση URL ενός αιτήματος GET:

`/test/demo_form.php?name1=value1&name2=value2`

Ορισμένα πράγμα που ισχύουν για τα αιτήματα GET:

- Μπορούν να αποθηκευτούν προσωρινά.
- Παραμένουν στο ιστορικό του προγράμματος περιήγησης.
- Μπορούν να επισημανθούν με σελιδοδείκτη.
- Δεν θα πρέπει ποτέ να χρησιμοποιούνται όταν ασχολούνται με ευαίσθητα δεδομένα.
- Έχουν περιορισμένο μήκος.
- Θα πρέπει να χρησιμοποιούνται μόνο για την ανάκτηση δεδομένων.

Στην περίπτωση αιτήματος HTTP POST το query string (ζεύγη ονόματος / τιμής) αποστέλλεται στο σώμα μηνυμάτων HTTP ενός αιτήματος POST:

`POST /test/demo_form.php HTTP/1.1`

Host: w3schools.com

`name1=value1&name2=value2`

Ορισμένα πράγμα που ισχύουν για τα αιτήματα POST:

- Δεν αποθηκεύονται προσωρινά.
- Δεν παραμένουν στο ιστορικό του προγράμματος περιήγησης.
- Δεν μπορούν να τεθούν σε σελιδοδείκτη.
- Δεν έχουν περιορισμούς στο μήκος των δεδομένων. (W3Schools, 2018)

### 3.5.2 Διαφορές HTTP GET & POST

Οι διαφορές ανάμεσα στα αιτήματα GET και POST είναι οι εξής:

	GET	POST
BACK κουμπί/Επαναφόρτιση σελίδας	Ακίνδυνο	Τα δεδομένα θα υποβληθούν εκ νέου (το πρόγραμμα περιήγησης θα ειδοποιεί το χρήστη ότι τα δεδομένα πρόκειται να υποβληθούν εκ νέου)
Σελιδοδείκτης	Μπορεί να έχει σελιδοδείκτη	Δεν μπορεί να έχει σελιδοδείκτη
Προσωρινή Αποθήκευση	Μπορεί να αποθηκευτεί προσωρινά	Δεν μπορεί να αποθηκευτεί προσωρινά
Τύπος κωδικοποίησης	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Κωδικοποίηση multipart/form-data για δεδομένα του δυαδικού συστήματος
Ιστορικό	Οι παράμετροι παραμένουν στο ιστορικό του φυλλομετρητή	Οι παράμετροι δεν αποθηκεύονται στο ιστορικό του φυλλομετρητή
Περιορισμοί στο μήκος δεδομένων	Ναι, κατά την αποστολή δεδομένων, η μέθοδος GET προσθέτει τα δεδομένα στη διεύθυνση URL και το μήκος μιας διεύθυνσης URL είναι περιορισμένο (το μέγιστο μήκος URL είναι 2048 χαρακτήρες)	Δεν υπάρχουν περιορισμοί
Περιορισμοί στο είδος δεδομένων	Επιτρέπονται μόνο χαρακτήρες ASCII	Δεν υπάρχουν περιορισμοί. Επιτρέπονται επίσης τα δεδομένα του δυαδικού συστήματος
Ασφάλεια	Το GET είναι λιγότερο ασφαλές σε σύγκριση με το POST, επειδή τα	Το POST είναι λίγο πιο ασφαλές από το GET επειδή οι παράμετροι δεν

	δεδομένα που αποστέλλονται αποτελούν μέρος της διεύθυνσης URL. Δεν πρέπει μην χρησιμοποιείται το GET για αποστολή κωδικών πρόσβασης ή άλλες ευαίσθητες πληροφορίες!	αποθηκεύονται στο ιστορικό του προγράμματος περιήγησης ή στα αρχεία καταγραφής διακομιστή ιστού
Ορατότητα	Τα δεδομένα είναι ορατά σε όλους στη διεύθυνση URL	Τα δεδομένα δεν εμφανίζονται στη διεύθυνση URL

### 3.5.5 Άλλες μέθοδοι αιτήσεων HTTP

Άλλες μέθοδοι αιτήσεων HTTP αναφέρονται στον παρακάτω πίνακα:

Μέθοδος	Περιγραφή
HEAD	Ισχύει ότι και για το GET αλλά επιστρέφει μόνο HTTP επικεφαλίδες και κανένα σώμα εγγράφου
PUT	Ανεβάζει μια αναπαράσταση του καθορισμένου URI (Uniform Resource Identifier)
DELETE	Διαγράφει τον καθορισμένο πόρο
OPTIONS	Επιστρέφει τις μεθόδους HTTP που υποστηρίζει ο διακομιστής
CONNECT	Μετατρέπει τη σύνδεση αιτήματος σε μια διαφανή TCP / IP τούνελ



## 3.6 MySQL



Εικόνα 3.5: Λογότυπο της MySQL

Η MySQL είναι ένα σύστημα διαχείρισης βάσης δεδομένων ανοιχτού κώδικα. Το όνομά της είναι ένας συνδυασμός του "My", το όνομα της κόρης του συνιδρυτή Michael Widenius και της "SQL", η συντομογραφία της δομημένης γλώσσας ερωτημάτων. Το αναπτυξιακό πρόγραμμα MySQL κατέστησε διαθέσιμο τον πηγαίο κώδικα του υπό τους όρους της Γενικής Άδειας Δημόσιας Χρήσης του GNU, καθώς και βάσει ποικίλων ιδιοκτησιακών συμφωνιών. Η MySQL ανήκε και χρηματοδοτήθηκε από μια ενιαία κερδοσκοπική εταιρεία, τη σουηδική εταιρεία MySQL AB, που τώρα ανήκει στην Oracle Corporation. Για ιδιόκτητη χρήση, διατίθενται πολλές αμειβόμενες εκδόσεις και προσφέρουν επιπλέον λειτουργίες.

Η MySQL είναι ένα κεντρικό στοιχείο της στοίβας λογισμικού ανοιχτού κώδικα LAMP open source (και άλλων στοίβων "AMP"). Το LAMP είναι ένα αρκτικόλεξο για το "Linux, Apache, MySQL, Perl / PHP / Python". Οι εφαρμογές που χρησιμοποιούν τη βάση δεδομένων MySQL περιλαμβάνουν: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB και Drupal. Η MySQL χρησιμοποιείται επίσης σε πολλές ιστοσελίδες μεγάλης προβολής, όπως το Google, το Facebook, το Twitter και το YouTube. (Wikipedia, 2018)

## 3.7 phpMyAdmin



Εικόνα 3.6: Λογότυπο του phpMyAdmin

Το phpMyAdmin είναι ένα εργαλείο ελεύθερου λογισμικού γραμμένο στην PHP, το οποίο προορίζεται για τη διαχείριση της MySQL μέσω του Διαδικτύου. Το phpMyAdmin υποστηρίζει ένα ευρύ φάσμα λειτουργιών στις υπηρεσίες MySQL και MariaDB. Συχνά χρησιμοποιούμενες λειτουργίες όπως διαχείριση βάσεων δεδομένων, πίνακες, στήλες, σχέσεις, ευρετήρια, χρήστες, δικαιώματα κ.λπ. μπορούν να εκτελεστούν μέσω του περιβάλλοντος χρήστη, ενώ μπορεί να εκτελεστεί οποιαδήποτε δήλωση SQL και είναι διαθέσιμο σε 72 γλώσσες για τη διευκόλυνση χρήσης τους.

Το πρόγραμμα phpMyAdmin είναι μέλος της Conservancy Software Freedom Conservancy. Το SFC είναι ένας μη κερδοσκοπικός οργανισμός που συμβάλλει στην προώθηση, βελτίωση, ανάπτυξη και υπεράσπιση έργων Free, Libre και Open Source Software (FLOSS). (phpMyAdmin, 2018)

### 3.7.1 Βασικές λειτουργίες phpMyAdmin

- Δυναμική διαδικτυακή διασύνδεση
- Υποστήριξη για τις περισσότερες λειτουργίες MySQL:
  - περιήγηση και αποθήκευση βάσεων δεδομένων, πινάκων, προβολών, πεδίων και ευρετηρίων
  - δημιουργία, αντιγραφή, διαγραφή, μετονομασία και να αλλαγή βάσεων δεδομένων, πινάκων, πεδίων και ευρετηρίων
  - διακομιστή συντήρησης, βάσεις δεδομένων και πίνακες, με προτάσεις σχετικά με τη διαμόρφωση του διακομιστή
  - να εκτελεί, να επεξεργάζεται και να επισημαίνει οποιαδήποτε δήλωση SQL, ακόμη και παρτίδες ερωτημάτων

- διαχείριση των λογαριασμών και των προνομίων των χρηστών της MySQL
- διαχείριση αποθηκευμένων διαδικασιών και ενεργοποιήσεων
- Εισαγωγή δεδομένων από CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές: CSV, SQL, XML, PDF, ISO / IEC 26300 - Κείμενο και υπολογιστικό φύλλο OpenDocument, Word, LATEX και άλλοι
- Διαχείριση πολλών διακομιστών
- Δημιουργία γραφικών της διάταξης της βάσης δεδομένων σας σε διάφορες μορφές
- Δημιουργία πολύπλοκων ερωτημάτων χρησιμοποιώντας Query-by-example (QBE)
- Αναζήτηση σε παγκόσμιο επίπεδο σε μια βάση δεδομένων ή σε ένα υποσύνολο της
- Μετασχηματισμός αποθηκευμένων δεδομένων σε οποιαδήποτε μορφή χρησιμοποιώντας ένα σύνολο προκαθορισμένων λειτουργιών, όπως την εμφάνιση δεδομένων BLOB ως εικόνα ή λήψης. (phpMyAdmin, 2018)

### 3.8 Γλώσσα προγραμματισμού CSS (Cascading Style Sheets)



Εικόνα 3.7: Λογότυπο της CSS3

Η CSS είναι μια γλώσσα φύλλου στυλ που χρησιμοποιείται για την περιγραφή της παρουσίασης ενός εγγράφου γραμμένου σε μια γλώσσα σήμανσης. Παρόλο που χρησιμοποιείται συνήθως για τη ρύθμιση της εμφάνισης των ιστοσελίδων και των διεπαφών χρήστη που είναι γραμμένες σε HTML και XHTML, η γλώσσα μπορεί να εφαρμοστεί σε οποιοδήποτε έγγραφο XML, συμπεριλαμβανομένων των απλών XML, SVG και XUL, και εφαρμόζεται σε ομιλίες ή σε άλλο μέσο μαζικής ενημέρωσης. Μαζί με τη HTML και τη JavaScript, η CSS αποτελεί τεχνολογία ακρογωνιαίου λίθου που χρησιμοποιείται από τους περισσότερους ιστότοπους για τη δημιουργία οπτικών σελίδων, διεπαφών χρήστη για εφαρμογές ιστού και διεπαφών χρήστη για πολλές εφαρμογές κινητής τηλεφωνίας.

Η CSS σχεδιάστηκε κυρίως για να επιτρέψει τον διαχωρισμό της παρουσίασης και του περιεχομένου, συμπεριλαμβανομένων πτυχών όπως η διάταξη, τα χρώματα και οι γραμματοσειρές. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, να παρέχει μεγαλύτερη ευελιξία και έλεγχο στις προδιαγραφές των χαρακτηριστικών παρουσίασης, να επιτρέπει σε πολλαπλές σελίδες HTML, να μοιράζονται μορφοποίηση καθορίζοντας το σχετικό CSS σε ξεχωριστό αρχείο .css και μειώνοντας την πολυπλοκότητα και την επανάληψη στο δομικό περιεχόμενο.

Ο διαχωρισμός της μορφοποίησης και του περιεχομένου καθιστά δυνατή την παρουσίαση της ίδιας σελίδας σήμανσης σε διαφορετικά στυλ για διαφορετικές μεθόδους απόδοσης, όπως στην οθόνη, σε έντυπη μορφή, φωνητικά (μέσω προγράμματος περιήγησης με βάση την ομιλία ή αναγνώστη οθόνης) συσκευές. Μπορεί επίσης να εμφανίσει διαφορετικά την ιστοσελίδα ανάλογα με το μέγεθος της οθόνης ή τη συσκευή προβολής. Οι αναγνώστες μπορούν επίσης να καθορίσουν ένα διαφορετικό φύλλο στυλ, όπως ένα αρχείο CSS που είναι αποθηκευμένο στον δίσκο του υπολογιστή, για να αντικαταστήσει αυτό που έχει καθορίσει ο δημιουργός. Οι αλλαγές στον γραφικό σχεδιασμό ενός εγγράφου (ή εκατοντάδων εγγράφων) μπορούν να εφαρμοστούν γρήγορα και εύκολα, με την επεξεργασία μερικών γραμμών στο αρχείο CSS που χρησιμοποιούν, αντί να αλλάξουν σήμανση στα έγγραφα.

Η προδιαγραφή CSS περιγράφει ένα σχήμα προτεραιότητας για να καθορίσει ποιους κανόνες στυλ υπερισχύουν εάν περισσότεροι από ένας κανόνες συμφωνούν με ένα συγκεκριμένο στοιχείο. Οι προτεραιότητες υπολογίζονται και εκχωρούνται σε κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλέψιμα.

Οι προδιαγραφές CSS τηρούνται από την Κοινοπραξία World Wide Web (W3C). Τύπος μέσου διαδικτύου (τύπου MIME) κείμενο / css είναι καταχωρημένο για χρήση με CSS από RFC 2318 (Μάρτιος 1998). Το W3C λειτουργεί μια δωρεάν υπηρεσία επικύρωσης CSS για έγγραφα CSS. (Duckett, 2018)

### 3.9 Γλώσσα προγραμματισμού JavaScript

Η JavaScript(JS) είναι μια υψηλού επιπέδου, ερμηνευμένη γλώσσα προγραμματισμού. Είναι μια γλώσσα που χαρακτηρίζεται επίσης ως δυναμική, ασθενώς δακτυλογραφημένη, βασισμένη στο πρωτότυπο και πολυ-παραδειγματική.

Παράλληλα με η HTML και η CSS, η JavaScript είναι μία από τις τρεις βασικές τεχνολογίες του World Wide Web. Η JavaScript ενεργοποιεί διαδραστικές ιστοσελίδες και συνεπώς αποτελεί ουσιαστικό μέρος των web εφαρμογών. Η συντριπτική πλειοψηφία των ιστότοπων το χρησιμοποιούν και όλοι οι μεγάλοι φυλλομετρητές ιστού έχουν μια αποκλειστική μηχανή JavaScript για να την εκτελέσουν.

Ως γλώσσα πολλαπλών προτύπων, η JavaScript υποστηρίζει στυλ προγραμματισμού που βασίζονται σε γεγονότα, λειτουργικά και επιτακτικά (συμπεριλαμβανομένων αντικειμενοστραφών και πρωτότυπων). Διαθέτει API για εργασία με κείμενο, συστοιχίες, ημερομηνίες, κανονικές εκφράσεις και βασικούς χειρισμούς του DOM, αλλά η ίδια η γλώσσα δεν περιλαμβάνει I / O, όπως εγκαταστάσεις δικτύωσης, αποθήκευσης ή γραφικών, βασιζόμενη σε αυτές περιβάλλον στο οποίο είναι ενσωματωμένο.

Αρχικά εφαρμοζόταν μόνο από πλευράς πελάτη σε προγράμματα περιήγησης ιστού, οι μηχανές JavaScript ενσωματώνονται πλέον σε πολλούς άλλους τύπους λογισμικού κεντρικού υπολογιστή, συμπεριλαμβανομένου του διακομιστή σε διακομιστές και βάσεις δεδομένων ιστού, καθώς και σε προγράμματα εκτός του Διαδικτύου, όπως επεξεργαστές κειμένου και λογισμικό PDF, περιβάλλοντα που καθιστούν το JavaScript διαθέσιμο για τη σύνταξη εφαρμογών για κινητά και επιτραπέζιους υπολογιστές, συμπεριλαμβανομένων γραφικών.

Παρόλο που υπάρχουν ισχυρές εξωτερικές ομοιότητες μεταξύ της JavaScript και της Java, συμπεριλαμβανομένου του ονόματος γλώσσας, της σύνταξης και των αντίστοιχων τυπικών βιβλιοθηκών, οι δύο γλώσσες είναι διακριτές και διαφέρουν σημαντικά στο σχεδιασμό. Η JavaScript επηρεάστηκε από γλώσσες προγραμματισμού όπως Self και Scheme. (Haverbeke, 2018)

### 3.10 Πρωτόκολλο SMTP (Simple Mail Transfer Protocol)

Το πρωτόκολλο μεταφοράς απλής αλληλογραφίας (SMTP) είναι ένα πρότυπο διαδικτύου για τη μετάδοση ηλεκτρονικού ταχυδρομείου. Ορίστηκε για πρώτη φορά από το RFC 821 το 1982 και ενημερώθηκε για τελευταία φορά το 2008 με παρατεταμένες προσθήκες SMTP από το RFC 5321, το οποίο είναι σήμερα το πρωτόκολλο ευρείας χρήσης.

Παρόλο που οι διακομιστές ηλεκτρονικού ταχυδρομείου και άλλοι πράκτορες μεταφοράς αλληλογραφίας χρησιμοποιούν το SMTP για την αποστολή και λήψη μηνυμάτων ηλεκτρονικού ταχυδρομείου, οι εφαρμογές αλληλογραφίας πελάτη σε επίπεδο χρήστη συνήθως χρησιμοποιούν μόνο SMTP για την αποστολή μηνυμάτων σε διακομιστή αλληλογραφίας για αναμετάδοση. Για την ανάκτηση μηνυμάτων, οι εφαρμογές πελατών συνήθως χρησιμοποιούν είτε IMAP είτε POP3.

Η επικοινωνία SMTP μεταξύ των διακομιστών αλληλογραφίας χρησιμοποιεί τη θύρα TCP 25. Οι πελάτες αλληλογραφίας, από την άλλη πλευρά, υποβάλλουν συχνά τα εξερχόμενα μηνύματα ηλεκτρονικού ταχυδρομείου σε ένα διακομιστή αλληλογραφίας στη θύρα 587. Παρόλο που είναι παρωχημένοι, οι πάροχοι αλληλογραφίας εξακολουθούν να επιτρέπουν τη χρήση μη συνηθισμένης θύρας 465 για το σκοπό αυτό.

Οι συνδέσεις SMTP που έχουν ασφαλιστεί από το TLS, γνωστή ως SMTPS, μπορούν να γίνουν χρησιμοποιώντας το STARTTLS.

Αν και τα ιδιόκτητα συστήματα (όπως τα Microsoft Exchange και IBM Notes) και τα συστήματα webmail (όπως το Outlook.com, το Gmail και το Yahoo! Mail) χρησιμοποιούν τα δικά τους μη τυπικά πρωτόκολλα για πρόσβαση στους λογαριασμούς θυρίδων αλληλογραφίας στους δικούς τους διακομιστές αλληλογραφίας, όλοι χρησιμοποιούν SMTP όταν αποστολή ή λήψη μηνυμάτων ηλεκτρονικού ταχυδρομείου εκτός των δικών τους συστημάτων. (Simple Mail Transfer Protocol, 2018)

### 3.10.1 Ιστορία του Πρωτοκόλλου SMTP

Ήδη από τα τέλη της δεκαετίας του 1960 είχε αρχίσει η ανταλλαγή μηνυμάτων ηλεκτρονικού ταχυδρομείου μεταξύ χρηστών mainframe υπολογιστών. Καθώς ο αριθμός των χρηστών και των υπολογιστών αυξανόταν, έγινε φανερή η ανάγκη δημιουργίας ενός πρωτοκόλλου για την ανταλλαγή ηλεκτρονικών μηνυμάτων μεταξύ χρηστών που χρησιμοποιούσαν διαφορετικά συστήματα υπολογιστών. Εξέλιξη των αρχικών πρωτοκόλλων που αναπτύχθηκαν την εποχή (δεκαετία του 1970) αυτή αποτελεί το SMTP. Συγκεκριμένα οι ρίζες του SMTP εντοπίζονται στα πρωτόκολλα Mail Box Protocol (1971), FTP Mail (1973) και Mail Protocol. Όταν όμως άρχισε να σχηματίζεται το Διαδίκτυο (1980), ο Jon Postel πρότεινε την δημιουργία ενός νέου πρωτοκόλλου για την ανταλλαγή ηλεκτρονικών μηνυμάτων, το οποίο δεν θα βασιζόταν τόσο πολύ στο FTP όπως έκαναν οι πρόγονοί του. Έτσι λοιπόν το 1982 γεννήθηκε το SMTP.

Το πρόγραμμα Sendmail ήταν ένα από τα πρώτα προγράμματα που υλοποίησε το SMTP, ενώ σήμερα υπάρχει μία πληθώρα τέτοιων προγραμμάτων όπως για παράδειγμα τα Postfix, qmail, Novell GroupWise, Exim, Novell NetMail και άλλα. Σε μία μέτρηση που έγινε το 2001 βρέθηκαν τουλάχιστον 50 προγράμματα τα οποία υλοποιούσαν το πρωτόκολλο SMTP είτε ως client (δηλαδή αποστολείς ηλεκτρονικών μηνυμάτων) είτε ως server (δηλαδή παραλήπτες ηλεκτρονικών μηνυμάτων).

Το αρχικό SMTP υποστήριζε κατά βάση μόνο μηνύματα απλού κειμένου και όχι μετάδοση αρχείων (πχ. εικόνες, εκτελέσιμα, μουσική κοκ). Στην συνέχεια όμως αναπτύχθηκαν διάφορα standards που επέτρεπαν την εισαγωγή αρχείων στα ηλεκτρονικά μηνύματα. Ένα από αυτά τα standards είναι και το Multipurpose Internet Mail Extensions (MIME), το οποίο κωδικοποιεί τα αρχεία με τέτοιο τρόπο ούτως ώστε να μπορούν να μεταδοθούν σε απλά μηνύματα SMTP. (Simple Mail Transfer Protocol, 2018)

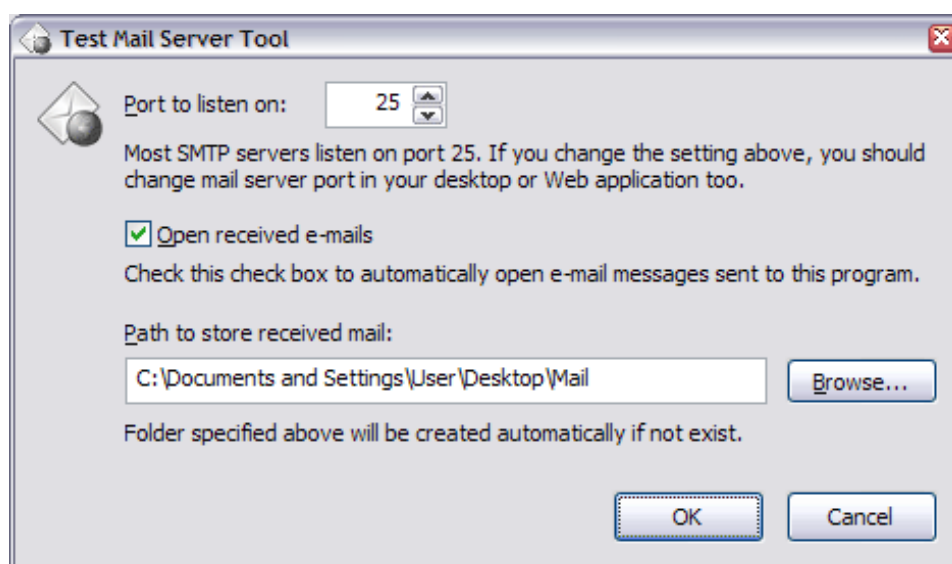
### 3.10.2 Test Mail Server Tool

Το Test Mail Server Tool είναι ένας διακομιστής πλήρους αλληλογραφίας (SMTP) για τη δοκιμή των δυνατοτήτων αποστολής μηνυμάτων μιας εφαρμογής ιστού ή επιφάνειας εργασίας ή για την προεπισκόπηση των μηνυμάτων πριν από την πραγματική αποστολή τους. Είναι ένα εξαιρετικό εργαλείο για προγραμματιστές ιστού ή για προγραμματιστές εφαρμογών που χρειάζονται διακομιστή αλληλογραφίας για δοκιμές.

Το πρόγραμμα ανταποκρίνεται στο localhost σε μια θύρα SMTP (συνήθως 25, μπορεί να αλλάξει). Όλα τα μηνύματα που αποστέλλονται σε αυτό το εργαλείο αποθηκεύονται αυτόματα και προαιρετικά ανοίγονται με τον πελάτη ηλεκτρονικού ταχυδρομείου.

Σημεία ενδιαφέροντος:

- Προγραμματιστές ιστού - δοκιμές και αποσφαλμάτωση μηνυμάτων αποστολής μηνυμάτων του ιστοτόπου.
- Προγραμματιστές εφαρμογών για επιτραπέζιους υπολογιστές - δοκιμή, εντοπισμός σφαλμάτων, προεπισκόπηση μηνυμάτων που αποστέλλονται από το πρόγραμμα
- Διαχειριστές γραφείου - προεπισκόπηση μηνυμάτων πριν την αποστολή τους σε πελάτες και συνεργάτες.
- Άλλοι χρήστες - προσθήκη δυνατότητα προεπισκόπησης αλληλογραφίας σε οποιοδήποτε πρόγραμμα. (Test Mail Server Tool, 2018)

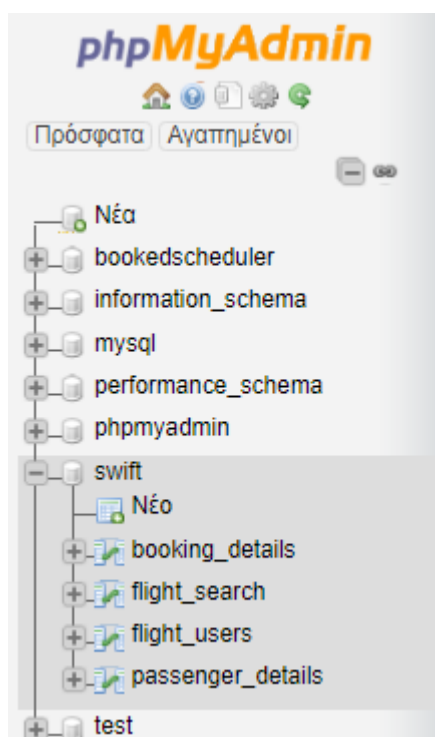


Εικόνα 3.8: Το βασικό μενού της εφαρμογής

## Κεφάλαιο 4 – Ανάλυση λειτουργίας της εφαρμογής

### 4.1 Η βάση και οι πίνακες της

Η βάση που χρησιμοποιήθηκε στο phpMyAdmin λέγεται swift και αποτελείται από τέσσερις πίνακες.



Εικόνες 4.1: Η βάση δεδομένων swift και οι πίνακες.

Ο πίνακας «booking\_details» χρησιμοποιείται για την αποθήκευση της κράτησης της πτήσης περιέχει τις εξής στήλες:

- ID της καταχώρησης στη βάση με primary key
- Ονοματεπώνυμο του πελάτη που κλίνει τα εισιτήρια
- ID της κράτησης
- Όνομα της πτήσης
- Email πελάτη
- Διεύθυνση κατοικίας πελάτη
- Συνολικό κόστος πτήσης
- Αριθμός ενήλικων επιβατών
- Αριθμός ανήλικων επιβατών



- Σύνολο επιβατών
- Έλεγχος πτήσης αν είναι επιστροφής ή όχι
- ID επιβάτη

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα	Ενέργεια
1	<b>b_id</b>	int(11)			Όχι	Καμία	AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
2	<b>b_name</b>	varchar(50)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
3	<b>b_fid</b>	varchar(5)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
4	<b>b_phno</b>	varchar(15)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
5	<b>b_mail</b>	varchar(50)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
6	<b>b_add</b>	varchar(100)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
7	<b>b_price</b>	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
8	<b>b_child</b>	varchar(3)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
9	<b>b_adults</b>	varchar(3)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
10	<b>b_total</b>	varchar(3)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
11	<b>b_type</b>	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
12	<b>b_pnr</b>	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα

Εικόνα 4.2: Ο πίνακας «booking\_details»

Ο πίνακας «flight\_search» περιέχει όλες τις διαθέσιμες πτήσεις τις οποίες μπορεί ο πελάτης να επιλέξει και περιέχει τις εξής στήλες:

- Εταιρεία της πτήσης
- Όνομα της πτήσης με primary key
- Πόλη στην οποία επιβαίνει ο πελάτης
- Πόλη στην οποία κατευθύνεται ο πελάτης
- Ημερομηνία πτήσης
- Ημερομηνία άφιξης
- Ώρα πτήσης
- Ώρα άφιξης
- Διαθέσιμες θέσεις για οικονομικό ταξίδι
- Διαθέσιμες θέσεις για επαγγελματικό ταξίδι
- Κόστος οικονομικού ταξιδιού
- Κόστος επαγγελματικού ταξιδιού

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα	Ενέργεια
1	company	varchar(30)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
2	fno	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
3	from_city	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
4	to_city	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
5	departure_date	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
6	arrival_date	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
7	departure_time	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
8	arrival_time	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
9	e_seats_left	int(3)			Όχι	20		Αλλαγή Διαγραφή Περισσότερα
10	b_seats_left	int(3)			Όχι	10		Αλλαγή Διαγραφή Περισσότερα
11	e_price	int(5)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
12	b_price	int(5)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα

Εικόνα 4.3: Ο πίνακας «flight\_search»

Ο πίνακας «flight\_users» περιέχει όλους τους λογαριασμούς χρηστών που έχουν εγγραφεί στην ιστοσελίδα και περιέχει τις εξής στήλες:

- ID της καταχώρησης στη βάση με primary key
- Όνομα χρήστη
- Επίθετο χρήστη
- Φύλο
- Username
- Κωδικός πρόσβασης
- Email
- Κωδικός για ενεργοποίηση του λογαριασμού που προστίθεται σε url έγκρισης
- Κατάσταση λογαριασμού (αν είναι ενεργοποιημένος ή όχι)
- Ημερομηνία και ώρα εγγραφής
- Έλεγχος για ανάκτηση κωδικού πρόσβασης
- Διεύθυνση κατοικίας
- Τηλέφωνο κατοικίας ή κινητό

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα	Ενέργεια
1	f_id	int(11)			Όχι	Καμία	AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
2	f_fname	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
3	f_lname	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
4	f_sex	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
5	f_uname	varchar(32)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
6	f_password	varchar(32)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
7	f_mailid	varchar(100)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
8	f_mailcode	varchar(100)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
9	f_active	int(11)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
10	f_regdate	datetime			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
11	f_passrec	int(11)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
12	f_address	varchar(500)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα
13	f_phone	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Περισσότερα

Εικόνα 4.4: Ο πίνακας «flight\_users»

Τέλος, ο πίνακας «passenger\_details» περιέχει όλες τις κρατήσεις εισιτηρίων με τα προσωπικά στοιχεία των επιβατών και περιέχει τις εξής στήλες:

- ID της καταχώρησης στη βάση με primary key
- ID του επιβάτη
- Ονοματεπώνυμο επιβάτη
- Ηλικία
- Φύλο
- Πτήση στην οποία θα επιβιβαστεί ο επιβάτης
- Πόλη επιβίβασης
- Πόλη άφιξης
- Ημερομηνία ταξιδιού
- Ημερομηνία άφιξης
- Ώρα ταξιδιού
- Ώρα άφιξης
- Έλεγχος πτήσης αν είναι επιστροφής ή όχι
- Είδος πτήσης
- Είδος επιβάτη (ενήλικος ή όχι)
- ID κράτησης

Διακομιστής: 127.0.0.1 > Βάση: swift > Πίνακας: passenger\_details

Περιήγηση Δομή Κώδικας SQL Αναζήτηση Προσθήκη Εξαγωγή Εισαγωγή Δικαιώματα Περισσότερα

Δομή πίνακα Εμφάνιση συσχετίσεων

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα	Ενέργεια
1	p_id	int(11)			Όχι	Καμία	AUTO_INCREMENT	Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
2	p_pnr	varchar(25)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
3	p_name	varchar(50)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
4	p_age	varchar(3)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
5	p_sex	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
6	p_fno	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
7	p_from	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
8	p_to	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
9	p_dedate	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
10	p_ardate	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
11	p_detime	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
12	p_artime	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
13	p_type	varchar(20)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
14	p_class	varchar(10)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
15	p_passtype	varchar(1)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα
16	p_fid	varchar(5)			Όχι	Καμία		Αλλαγή Διαγραφή Πρωτεύον Περισσότερα

Επιλογή όλων Με τους επιλεγμένους: Περιήγηση Αλλαγή Διαγραφή Πρωτεύον Μοναδικό Ευρετήριο

Προσθήκη στις κεντρικές στήλες Απομάκρυνση από κεντρικές στήλες Παράθυρο

Εικόνα 4.5: Ο πίνακας «passenger\_details»

## 4.2 Αρχική σελίδα (index.php)

Our Airlines Services Safety Passenger Login

Search Flights

One Way Return

From  
Please Select

To  
Please Select

Departure Date


Return Date

Number of Adults  
1

Number of Children  
0

Class  
Economy

Submit



Services offered by Our Airlines

Travel across the world through our flights which are getting updated in no time

- Free 3 course meals for every passenger
- Free Wi-Fi services offered during your flight
- Up to 25 Kg Baggage limit for every passenger
- Unlimited Food and Alcohol in Business Class
- Book hotels via Our Airlines and avail 10% discount per room
- Discounted travel coupons for every city
- Hire cabs or bus via Our Airlines and avail 10% discount per ticket
- Avail wheelchairs and emergency services for disabled at free of cost

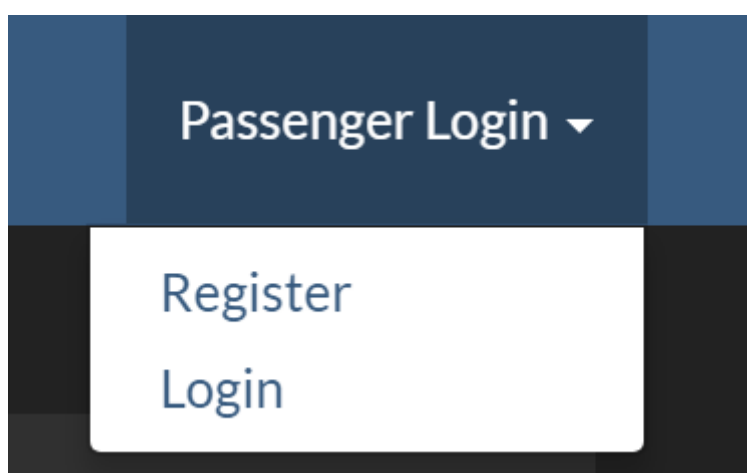
Εικόνα 4.6: Η εμφάνιση της αρχικής ιστοσελίδας

Η αρχική σελίδα όπως φαίνεται στην παραπάνω εικόνα έχει τη βασική φόρμα εύρεσης πτήσεων που μπορεί να χρησιμοποιεί ο πελάτης για να αναζητήσει την

επιθυμητή πτήση του. Υπάρχουν όμως ορισμένες προϋποθέσεις για την πραγματοποίηση της αναζήτησης:

- Επιλογή κράτησης εισιτηρίων επιστροφής ή όχι.
- Επιλογή πόλης από την οποία ο πελάτης θα ταξιδεύσει μέσω μιας dropdown list, όπου οι επιλογές καλούνται από τη βάση δεδομένων MySQL από έναν πίνακα που έχει αποθηκευμένες όλες τις διαθέσιμες πτήσεις.
- Επιλογή πόλης ως τελικός προορισμός του πελάτη μέσω ενός dropdown list και ισχύουν τα ίδια και εδώ.
- Ημερομηνία της πτήσης που επιθυμεί ο πελάτης.
- Ημερομηνία επιστροφής που επιθυμεί ο πελάτης με την προϋπόθεση ότι έχει επιλεγεί η επιλογή κράτησης εισιτηρίων επιστροφής αλλιώς η επιλογή αυτή είναι κλειδωμένη.
- Αριθμός ενήλικων επιβατών από 1 έως 5.
- Αριθμός παιδιών επιβατών από 0 έως 5.
- Επιλογή οικονομικού ή επαγγελματικού ταξιδιού.

Επιπλέον ο χρήστης μπορεί να επισκεφτεί τις σελίδες services.php και safety.php κουνώντας τον κέρσορα στις επιλογές «Services» και «Safety» στο πάνω μέρος της ιστοσελίδας αντίστοιχα. Η σελίδα services.php παρουσιάζει μέσω URL τις υπηρεσίες που προσφέρονται κατά τη διάρκεια των πτήσεων ενώ η σελίδα safety.php αναφέρεται στους κανόνες ασφαλείας για μια πιο ασφαλή και ευχάριστη πτήση. Τέλος, πάνω δεξιά υπάρχει το «Passenger Login» στο οποίο αν κάνουμε κλικ θα εμφανιστούν οι επιλογές «Register» και «Login» μέσω dropdown list.

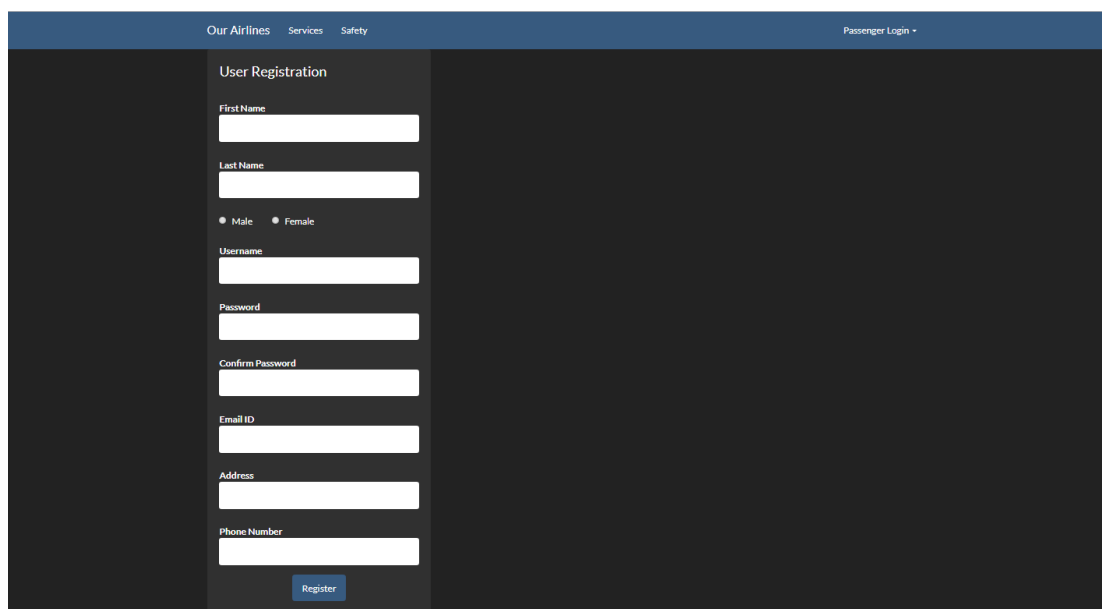


Εικόνα 4.7: Η εμφάνιση των επιλογών Register και Login

## 4.3 Λογαριασμός χρήστη

Για να μπορέσει ο πελάτης να κάνει την κράτηση που ο ίδιος επιθυμεί πρέπει να δώσει τα προσωπικά του στοιχεία στην ιστοσελίδα. Όπως είδαμε στην εικόνα 3.2 ο χρήστης μπορεί μέσω του «Passenger Login» μπορεί να συνδεθεί στο λογαριασμό του ή να δημιουργήσει έναν. Αν κάνουμε κλικ στο «Register» και «Login» θα βρεθούμε στις ιστοσελίδες fregister.php και flogin.php αντίστοιχα. Στην fregister.php ο χρήστης συμπληρώνει τα προσωπικά του στοιχεία προκειμένου να δημιουργήσει το λογαριασμό του. Στην flogin.php συμπληρώνει το username και password τα οποία υπέβαλε ο χρήστης στη δημιουργία του λογαριασμού του. Τα στοιχεία του πελάτη αποθηκεύονται στη βάση δεδομένων MySQL. Παρακάτω θα αναλυθεί η διαδικασία.

### 4.3.1 Δημιουργία Λογαριασμού (fregister.php)



The screenshot shows a web page with a dark blue header containing navigation links: 'Our Airlines', 'Services', 'Safety', and 'Passenger Login'. The main content area is a white box titled 'User Registration'. It contains the following fields and controls:

- First Name:
- Last Name:
- Gender:  Male  Female
- Username:
- Password:
- Confirm Password:
- Email ID:
- Address:
- Phone Number:
- Register:

Εικόνα 4.8: Εμφάνιση της σελίδας fregister.php

Όπως φαίνεται στην παραπάνω εικόνα η σελίδα fregister.php παρουσιάζει μια φόρμα όπου ο μελλοντικός πελάτης χρειάζεται για να εγγραφεί. Ο πελάτης πρέπει να συμπληρώσει στη φόρμα αυτή τα εξής:

- Το όνομα του πελάτη
- Το επίθετο του
- Το γένος του, το οποίο επιλέγεται μέσω radio buttons
- Το username του πελάτη με το οποίο ο ίδιος θα πρέπει να θυμάται ώστε να κάνει login στην ιστοσελίδα
- Το password για το οποίο ισχύει ότι ακριβώς για το username

- Το confirm password όπου για λόγους ασφαλείας ο χρήστης ξαναγράφει το password που επιθυμεί να έχει. Αν το περιεχόμενο των password και confirm password είναι διαφορετικό τότε η εγγραφή δεν θα πραγματοποιηθεί
- Το email ID όπου ο χρήστης δίνει το email του ώστε να προχωρήσει τη διαδικασία εγγραφής. Το σύστημα αναγνωρίζει αν ο χρήστης έχει δώσει πληκτρολογήσει περιεχόμενο το οποίο αντιστοιχεί σε email. Επίσης, ο χρήστης δεν μπορεί να δημιουργήσει περισσότερο από έναν λογαριασμούς με το ίδιο email
- Την διεύθυνση κατοικίας του
- Το τηλέφωνο κατοικίας ή κινητού του

Αν δεν συμπληρωθεί κάτι στην φόρμα η εγγραφή δεν πραγματοποιείται. Όλα τα στοιχεία του λογαριασμού του πελάτη αποθηκεύονται σε έναν πίνακα της βάσης δεδομένων που περιέχει όλους τους εγγεγραμμένους πελάτες.

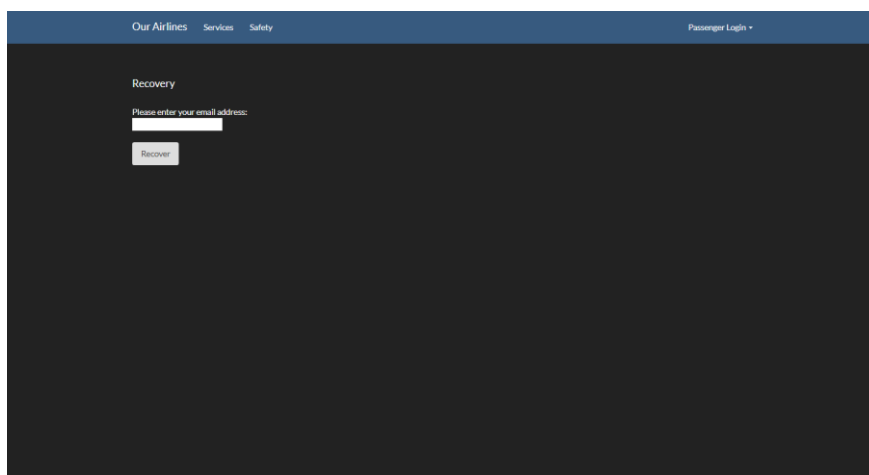
Επίσης ο λογαριασμός δεν ενεργοποιείται αμέσως μετά την εγγραφή. Ο χρήστης μόλις πραγματοποιήσει την εγγραφή του θα δεχθεί ένα μήνυμα στο email του από τον administrator του site το οποίο του στέλνει ένα URL στο οποίο θα πρέπει να μπει προκειμένου ο λογαριασμός του χρήστη να ενεργοποιηθεί. Το URL μετά την ενεργοποίηση του λογαριασμού θεωρείται άχρηστο. Η διαδικασία αυτή πραγματοποιείται με τη βοήθεια του Test Mail Server Tool το οποίο αναλύθηκε στο προηγούμενο κεφάλαιο.

#### 4.3.2 Σύνδεση Λογαριασμού (flogin.php)

Εικόνα 4.9: Εμφάνιση της σελίδας flogin.php

Όπως έχουμε δει σε πολλές διαδικτυακές εφαρμογές, forums, site αγορών κ.α. ο χρήστης συμπληρώνει στη φόρμα login το username και password προκειμένου να

συνδεθεί στο λογαριασμό του. Αν δεν έχει λογαριασμό επιλέγει την επιλογή «Register Here» όπου τον πηγαίνει στην σελίδα register.php που αναλύθηκε παραπάνω. Όμως έχει υπάρξει ουκ ολίγες φορές η περίπτωση να έχει ξεχάσει ο χρήστης είτε το username είτε το password του με αποτέλεσμα να μην μπορεί να συνδεθεί στο λογαριασμό του. Στην περίπτωση αυτή ο χρήστης θα πρέπει να επιλέξει τις επιλογές «Forgot Username?» ή «Forgot Password?» στις οποίες θα πρέπει ο χρήστης να δώσει το email με το οποίο πραγματοποίησε την εγγραφή προκειμένου να δεχθεί μήνυμα από τον admin για επίλυση του θέματος.

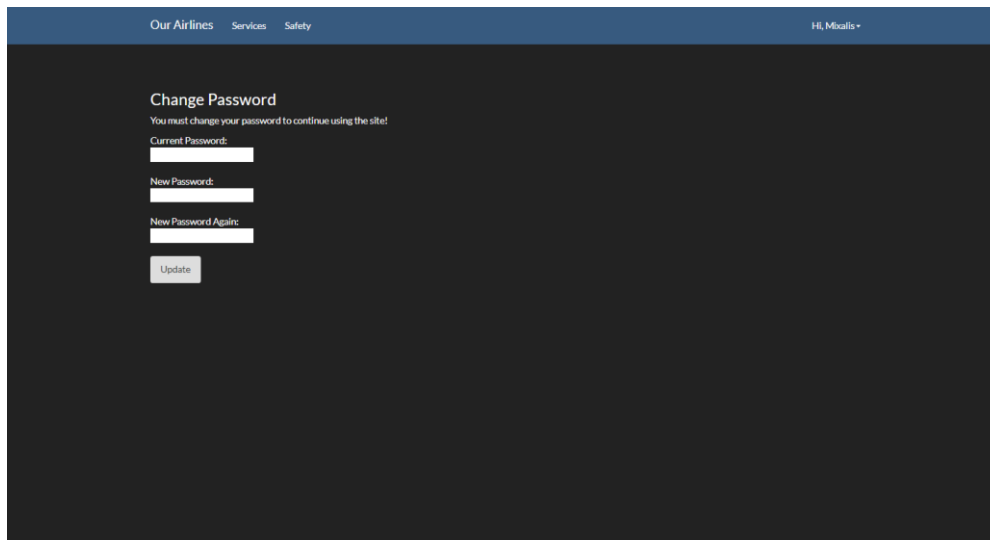


Εικόνα 4.10: Σελίδα ανάκτησης username ή password

Αμα ο χρήστης ζητήσει βοήθεια για το username του θα δεχθεί ένα μήνυμα από τον admin το οποίο θα του υπενθυμίσει το username του μέσω αναζήτησης στη βάση δεδομένων.

Στην περίπτωση όμως που ο χρήστης χρειάζεται βοήθεια για το password του θα δεχθεί έναν προσωρινό κωδικό από τον admin μέσω email με το οποίο μόλις συνδεθεί θα πρέπει να αντικαταστήσει με έναν προσωπικό δικό του για λόγους ασφαλείας. Ο χρήστης θα πρέπει να δώσει τον τωρινό κωδικό του και στην συνέχεια να πληκτρολογήσει έναν κωδικό τον οποίο θα πρέπει να δηλώσει δύο φορές όπως ακριβώς και στην διαδικασία εγγραφής προκειμένου η ενέργεια να είναι επιτυχής. Ο χρήστης μπορεί να αλλάξει τον κωδικό του λογαριασμού του όποτε ο ίδιος το επιθυμεί.

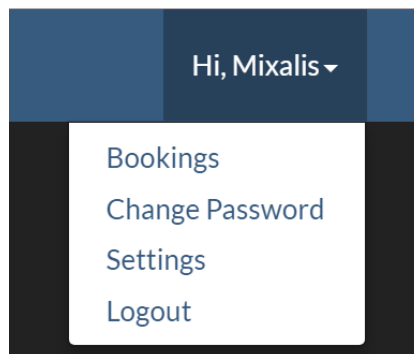




Εικόνα 4.11: Σελίδα αντικατάστασης password

### 4.3.3 Διαχείριση Λογαριασμού

Όταν ο πελάτης συνδεθεί με τα στοιχεία του στο λογαριασμό του θα δει το όνομά του στη θέση του «Passenger Login» και η dropdown list παρουσιάζει αλλαγές.

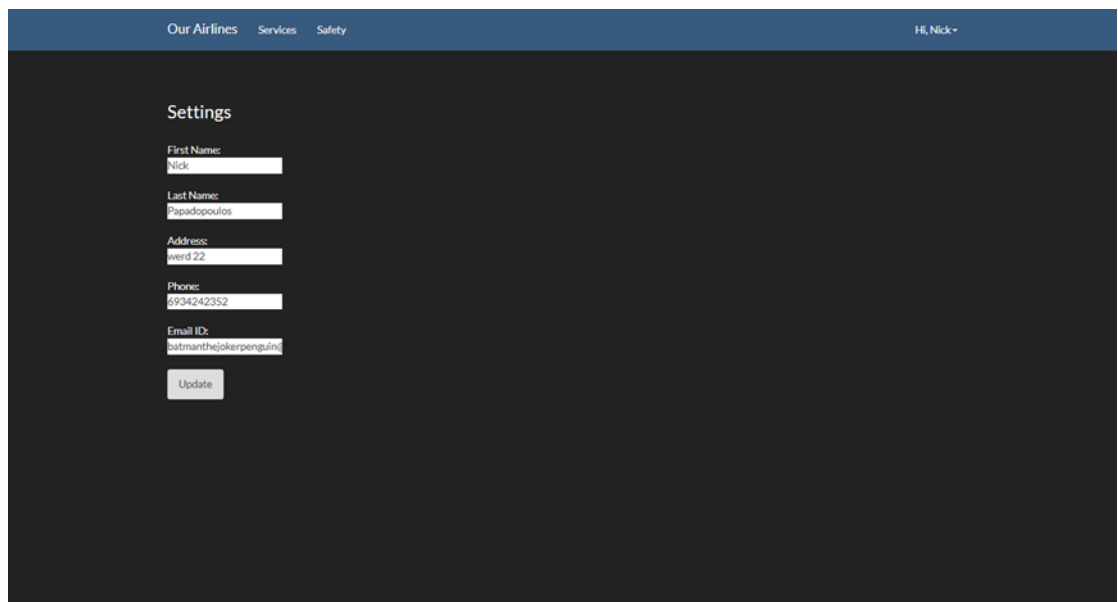


Εικόνα 4.12: Οι αλλαγές στο «Passenger Login» όταν ο χρήστης συνδέεται στο λογαριασμό του

Όπως παρατηρείται στην εικόνα 4.12, οι επιλογές «Login» και «Register» έχουν αντικατασταθεί από άλλες στη dropdown list. Οι επιλογές είναι οι εξής:

- Η επιλογή «Bookings» στην οποία ο χρήστης μπορεί να δει τα στοιχεία του μόλις πραγματοποιήσει την κράτηση των εισιτηρίων της πτήσης που επιθυμεί. Θα δούμε περισσότερα σχετικά με την επιλογή αυτή στην επόμενη ενότητα.
- Η επιλογή «Change Password» όπου ο χρήστης αλλάζει τον κωδικό πρόσβασης του και αναφέρθηκε παραπάνω.

- η επιλογή «Settings» όπου ο χρήστης αλλάζει ορισμένα από τα προσωπικά του στοιχεία. Συγκεκριμένα μπορεί να αλλάξει το ονοματεπώνυμό του, τη διεύθυνση κατοικίας, το τηλέφωνο και τέλος το email του.



The screenshot shows a web application interface. At the top, there is a dark blue navigation bar with the text 'Our Airlines', 'Services', and 'Safety'. On the right side of this bar, it says 'Hi, Nick'. Below the navigation bar, the main content area has a dark grey background. The title 'Settings' is centered at the top of this area. Below the title, there are several form fields, each with a label and a text input field: 'First Name' with the value 'Nick', 'Last Name' with 'Papadopoulos', 'Address' with 'Ierod 22', 'Phone' with '6934242352', and 'Email ID' with 'batmanthejokerpenguin'. At the bottom of the form, there is a light grey button labeled 'Update'.

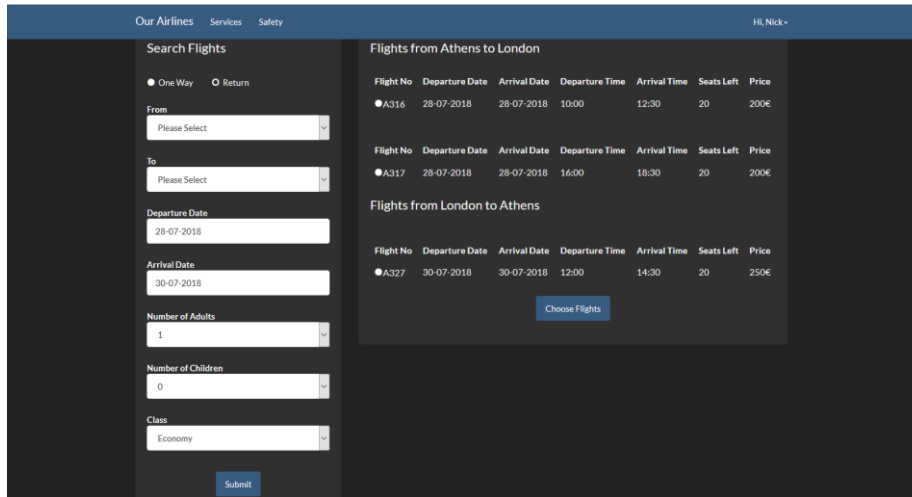
Εικόνα 4.13: Η ιστοσελίδα settings.php όπου ο χρήστης κάνει αλλαγές στα προσωπικά του στοιχεία

- Η επιλογή «Logout» όπου ο χρήστης μπορεί να απενεργοποιήσει το λογαριασμό του όποτε αυτός επιθυμεί απλά πατώντας πάνω στην επιλογή «Logout» πραγματοποιώντας ένα `session_destroy`.

#### 4.4 Πραγματοποίηση κράτησης εισιτηρίων

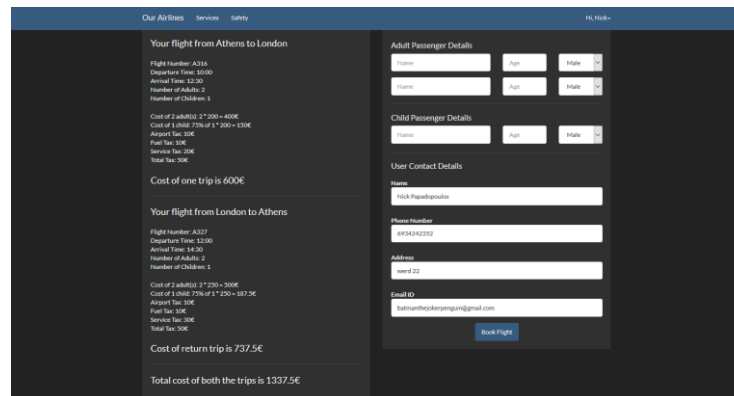
Τώρα θα μπορούμε στο κύριο μέρος της εργασίας: την κράτηση των αεροπορικών εισιτηρίων. Όπως αναφέρθηκε στην ενότητα 4.2, η αρχική σελίδα περιέχει τη φόρμα την οποία ο πελάτης χρησιμοποιεί προκειμένου να αναζητήσει την πτήση που επιθυμεί.

Ας πάρουμε ως παράδειγμα μια πτήση από Αθήνα σε Λονδίνο το οποίο θα είναι οικονομικό ταξίδι και επιστροφής. Στο παράδειγμα, ο πελάτης θα κλείσει εισιτήρια για δύο ενήλικες και ένα ανήλικο. Ο πελάτης δίνοντας τις ημερομηνίες που επιθυμεί να ταξιδέψει και να επιστρέψει από το ταξίδι του.



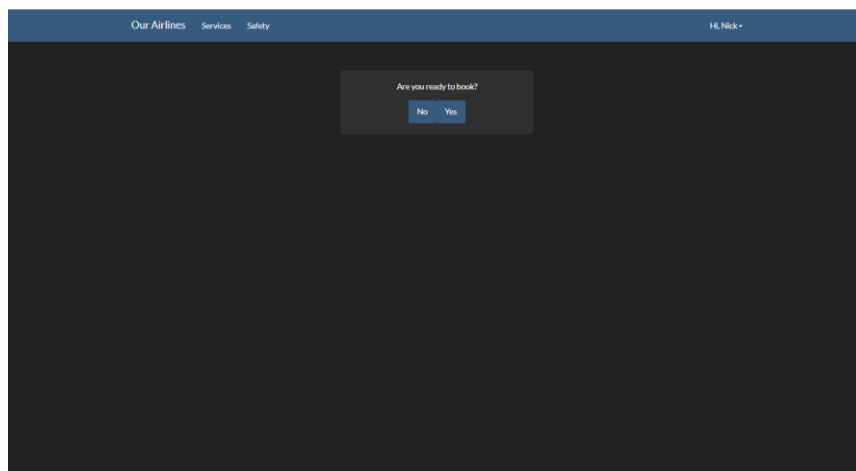
Εικόνα 4.14: Η εμφάνιση των διαθέσιμων πτήσεων για τις συγκεκριμένες ημερομηνίες

Όπως φαίνεται στην εικόνα 4.14, οι διαθέσιμες πτήσεις εμφανίζονται δεξιά στη σελίδα και μέσω radio buttons ο πελάτης επιλέγει την πτήση η οποία έχει τον επαρκή αριθμό διαθέσιμων θέσεων ή την προτεινόμενη για αυτόν ώρα. Στην περίπτωση που στον πίνακα της βάσης δεδομένων όπου γίνεται η αναζήτηση δεν υπάρχουν πτήσεις τη συγκεκριμένη ημερομηνία θα εμφανιστεί το μήνυμα «Not flights found». Μόλις ο πελάτης επιλέξει τις πτήσεις της προτίμησής του θα δει στην επόμενη σελίδα η οποία ονομάζεται book.php, στα αριστερά του αναλυτικά τα στοιχεία της πτήσεων αλλά το συνολικό κόστος τους. Στα δεξιά εμφανίζεται μια φόρμα συμπλήρωσης των στοιχείων των ατόμων (ονοματεπώνυμο, ηλικία, γένος) που θα είναι στις πτήσεις ενώ κάτω δεξιά στη σελίδα φαίνονται τα στοιχεία του πελάτη από το προσωπικό λογαριασμό του ο οποίος είναι και υπεύθυνος για την κράτηση. Επίσης, αν δοθούν λανθασμένα στοιχεία στη φόρμα όπως μια ηλικία κάτω των 18 για έναν ενήλικα, η διαδικασία δεν θα συνεχιστεί για προφανής λόγους.



Εικόνα 4.15: Η σελίδα book.php

Όταν ο πελάτης δώσει τα στοιχεία των επιβατών και πατήσει το κουμπί «Book Flight», τότε θα εμφανιστεί μια μικρή σελίδα η οποία ρωτάει μια τελευταία φορά τον χρήστη αν είναι σίγουρος για τις επιλογές του πριν κλείσει τα εισιτήριά του. Άμα επιλέξει «No» θα βρεθεί στην αρχική σελίδα index.php και η διαδικασία ακυρώνεται. Με την επιλογή «Yes» στην οθόνη εμφανίζεται μια επιβεβαίωση ότι τα εισιτήρια έκλεισαν και στην περίπτωση που ο χρήστης δεν πραγματοποιήσει καμιά ενέργεια, μεταφέρεται πίσω στην αρχική σελίδα μετά από 10 δευτερόλεπτα και πραγματοποιείται αυτόματη αποσύνδεση από το λογαριασμό για λόγους ασφαλείας. Τα στοιχεία των επιβατών αποθηκεύονται στη βάση MySQL στον πίνακα «passenger\_details». Επίσης, στον πίνακα «booking\_details» αποθηκεύονται τα στοιχεία του πελάτη-χρήστη.

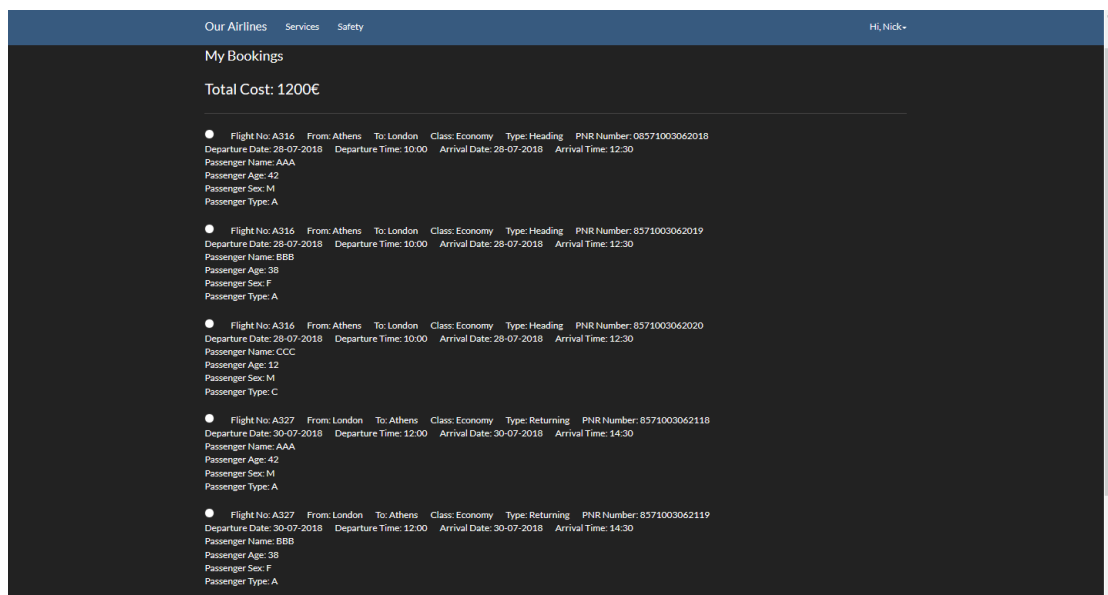


Εικόνα 4.16: Το τελευταίο στάδιο της διαδικασίας για την κράτηση εισιτηρίων

Μετά την ολοκλήρωση της κράτησης ο πελάτης-χρήστης μπορεί να δει τις πτήσεις του μέσω της επιλογής «Bookings» όπου θα μεταβεί σε μια σελίδα η οποία πριν την κράτηση ήταν άδεια εφόσον ξανασυνδεθεί στο λογαριασμό του καθώς, όπως αναφέρθηκε στην προηγούμενη ενότητα, πραγματοποιείται ένα αναγκαστικό logout για λόγους ασφαλείας. Σε αυτήν ο χρήστης μπορεί να ακυρώσει την πτήση ενός επιβάτη ο οποίος αποφάσισε τελικά να μην ταξιδέψει επιλέγοντας το αντίστοιχο radio button και στη συνέχεια πατώντας το κουμπί «Cancel Bookings». Όταν πατηθεί το κουμπί, εμφανίζεται μια άλλη σελίδα η οποία επιβεβαιώνει την ακύρωση της κράτησης για τον συγκεκριμένο επιβάτη για 5 δευτερόλεπτα πριν επιστρέψει στην σελίδα «bookings.php». Ταυτόχρονα, με την κίνηση αυτή συμβαίνουν τα εξής:

- Η κράτηση διαγράφεται από τον πίνακα «passenger\_details».
- Το κόστος του εισιτηρίου αφαιρείται από την κράτηση στον πίνακα «booking\_details».
- Το συνολικό κόστος των επιβατών μειώνεται κατά ένα στον πίνακα «booking\_details».

- Οι διαθέσιμες θέσεις της πτήσης αυξάνονται κατά μία καθώς απελευθερώνεται η θέση στον πίνακα «flight\_search».



Εικόνα 4.17: Η σελίδα bookings.php

## Κεφάλαιο 5 – Συμπεράσματα

Το ηλεκτρονικό εμπόριο τη σήμερον ημέρα επιτρέπει στις επιχειρήσεις να λειτουργούν χωρίς τα εμπόδια του χρόνου και της απόστασης. Οι χρήστες του διαδικτύου μπορούν να επισκεφτούν την ιστοσελίδα οποιαδήποτε στιγμή και να κάνουν κράτηση εισιτηρίων εύκολα και γρήγορα. Στα πλαίσια αυτής της πτυχιακής εργασίας, δημιουργήθηκε ένα σύστημα για online κρατήσεις. Βασικός στόχος ήταν η εύκολη πλοήγηση των υποψήφιων πελατών στην ιστοσελίδα μας μέσω της δημιουργίας προσωπικού λογαριασμού.

Η διαδικτυακή κράτηση αεροπορικών εισιτηρίων είναι γενικά μια πολύπλοκη εργασία, ακόμη και σε πλήρως λειτουργικές συσκευές πρόσβασης στο διαδίκτυο, όπως υπολογιστές, που αντιπροσωπεύοντας μια επαναλαμβανόμενη αναζήτηση στη βάση δεδομένων πτήσεων και στη συνέχεια περιήγηση σε λίστες πτήσεων, που αποτελούνται από διαφορετικούς μεταφορείς, τιμές, ημερομηνίες και τους χρόνους, ώστε να δημιουργηθεί ένας βέλτιστος συνδυασμός εξερχόμενων και εισερχόμενων πτήσεων.

Όπως παρατηρήθηκε κατά την ανάλυση της εφαρμογής η γλώσσα PHP και η βάση δεδομένων MySQL αποτελούν τον ακρογωνιαίο λίθο για την κατασκευή μιας δυναμικής ιστοσελίδας. Σε αντίθεση με τις γλώσσες προγραμματισμού HTML και CSS που βασίζονται στην κατασκευή του template μιας ιστοσελίδας και γενικότερα στη εξωτερική εμφάνισή της, η PHP και σ'ένα μικρότερο βαθμό η Javascript συμβάλλουν στη λειτουργικότητα της και στην ευκολότερη πλοήγηση από τη μια ιστοσελίδα του ιστοτόπου στην άλλη χωρίς να δημιουργούνται προβλήματα πρόσβασης.

## Παράρτημα

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της παρούσας εφαρμογής.

### connect.php

```
<?php
    $connect_error = 'Sorry, there was some connectivity issue!';
    @mysql_connect('localhost','root','') or die($connect_error);
    @mysql_select_db('swift') or die($connect_error);
?>
```

Κώδικας για σύνδεση της βάσης δεδομένων MySQL μέσω της γλώσσας PHP

### index.php

```
<div class="row">
    <div class="col-lg-4">
        <div class="well bs-component">
            <form class="form-horizontal" action="" method="GET">
                <legend>Search Flights</legend>
                <div class="form-group">
                    <div class="col-lg-10">
                        <div class="radio">
                            <label>
                                <input type="radio" name="path" value="oneway"
onclick="setReadOnly(this)">
                                    One Way
                                </label>
                            </div>
                            <br/>
                            <label>
                                <input type="radio" name="path" value="return"
checked onclick="setReadOnly(this)">
                                    Return
                                </label>
                            </div>
                        </div>
                    </div>
                    <div class="form-group">
                        <div class="col-lg-12">
                            <label class="control-label"
for="focusedInput">From</label>
                            <select class="form-control" name="from_city" id="from_city"><option
value="none">Please Select</option>
                                <option onmousedown="5"
                                </select>
                        </div>
                    </div>
                </form>
                <?php
                $query= "SELECT DISTINCT from_city FROM flight_search ORDER BY
from_city";
                $result = mysql_query($query);
                while($row = mysql_fetch_array($result)) {
```

```

        echo "<option value=" . $row["from_city"] . ">" .
$row["from_city"] . "</option>";
    }
    ?>
</select>
</div>
</div>
<div class="form-group">
    <div class="col-lg-12">
        <label
for="focusedInput">To</label>
        <select
class="form-control" name="to_city" id="to_city"><option
value="none" >Please Select</option>
<?php
$query= "SELECT DISTINCT to_city FROM flight_search ORDER BY
to city";
$result = mysql_query($query);
while($row = mysql_fetch_array($result)) {
    echo "<option value=" . $row["to_city"] . ">" . $row["to_city"]
. "</option>";
}
?>
</select>
</div>
</div>
<div class="form-group">
    <div class="col-lg-12">
        <label
for="focusedInput">Departure Date</label>
        <input class="form-control" name="departure_date"
id="departure_date" value="<?php if(isset($_GET['departure_date']))
{ echo htmlentities($_GET['departure_date']); } else echo '';?>"
required type="text">
        </div>
</div>
<div class="form-group">
    <div class="col-lg-12">
        <label
for="focusedInput">Arrival Date</label>
        <input class="form-control" name="return_date"
id="return_date" value="<?php if(isset($_GET['return_date'])) { echo
htmlentities($_GET['return_date']); } else echo '';?>" required
type="text">
        </div>
</div>
<div class="form-group">
    <div class="col-lg-12">
        <label for="select" class="control-label">Number of
Adults</label>
        <select class="form-control" name="count_a"
id="select">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
        </select>
    </div>
</div>
<div class="form-group">

```



```

        <div class="col-lg-12">
            <label for="select" class="control-label">Number of
Children</label>
            <select      class="form-control"      name="count_c"
id="select">
                <option value="0">0</option>
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option>
            </select>
        </div>
    </div>
    <div class="form-group">
        <div class="col-lg-12">
            <label      for="select"      class="control-
label">Class</label>
            <select      class="form-control"      name="class"
id="select">
                <option      name="economy"
value="Economy">Economy</option>
                <option      name="business"
value="Business">Business</option>
            </select>
            <br>
        </div>
    </div>
    <div class="form-group">
        <center><input      type="submit"      id="submit"
value="Submit" name="submit" class="btn btn-primary"></center>
    </div>
</form>
</div>
</div>

```

Κώδικας που δημιουργεί τη φόρμα αναζήτησης πτήσεων την οποία συμπληρώνει ο χρήστης.

```

<div class="col-lg-8">
    <div class="well bs-component">
        <?php if (f logged in()===true) {
            echo "<form id='form1' class='form-horizontal'
action='book.php' method='GET'>";
        } else {
            echo "<form id='form2' class='form-horizontal'
action='flogin.php' method='GET'>";
        } ?>

        <?php

            if (isset($_GET['path'])===true
                &&      isset($_GET['from_city'])===true      &&
isset($_GET['to_city'])===true
                &&      isset($_GET['departure_date'])===true
                &&      isset($_GET['count_a'])===true      &&
isset($_GET['count_c'])===true &&      isset($_GET['class'])===true) {

```

```

        $fro = $_GET['from_city'];
        $to = $_GET['to_city'];
        $departdate = $_GET['departure_date'];
        $class = $_GET['class'];
        $path = $_GET['path'];
        $counta = $_GET['count_a'];
        $countc = $_GET['count_c'];
        if($path==='oneway') {
            echo '<legend>Flights from ' . $fro . ' to ' . $to . '</legend>';
            $query = "SELECT * FROM `flight search` WHERE `from city` = '$fro' AND `to_city` = '$to' AND `departure_date` = '$departdate'";
            $result = mysql_query($query);
            if($result) {
                if(mysql_num_rows($result) > 0) {
                    while($row = mysql_fetch_assoc($result)) {
                        ?>
                        <table class="table">
                            <thead>
                                <tr>
                                    <th>Flight No</th>
                                    <th>Departure Date</th>
                                    <th>Arrival Date</th>
                                    <th>Departure Time</th>
                                    <th>Arrival Time</th>
                                    <th>Seats Left</th>
                                    <th>Price</th>
                                </tr>
                            </thead>
                            <tbody>
                                <tr>
                                    <?php if($class==='Economy') { ?>
                                        <td><input type='radio' required name='chose_to' value='<?php echo $row['fno']; ?>' /> <?php echo $row['fno']; ?></td>
                                        <td><?php echo $row['departure_date']; ?></td>
                                        <td><?php echo $row['arrival_date']; ?></td>
                                        <td><?php echo $row['departure_time']; ?></td>
                                        <td><?php echo $row['arrival_time']; ?></td>
                                        <td><?php echo $row['e_seats_left']; ?></td>
                                        <td><?php echo $row['e_price']; ?>€</td>
                                    <?php }

                                    <?php }
                                <td><input type="radio" required name="chose_to" value="<?php echo $row['fno']; ?>" /><?php echo $row['fno']; ?></td>
                                <td><?php echo $row['departure_date']; ?></td>
                                <td><?php echo $row['arrival_date']; ?></td>
                                <td><?php echo $row['departure time']; ?></td>
                                <td><?php echo $row['arrival time']; ?></td>
                                <td><?php echo $row['b_seats_left']; ?></td>
                                <td><?php echo $row['b_price']; ?>€</td>
                            <?php }
                            } else { 'Not enough seats left, please search again!'; }
                        }?>
                    </tr>
                </tbody>
            </table>

```

```



```

```

        <td><?php echo $row1['departure_time']; ?></td>
        <td><?php echo $row1['arrival_time']; ?></td>
        <td><?php echo $row1['b_seats_left']; ?></td>
        <td><?php echo $row1['b_price']; ?>€</td>
    </tr>
</table>
<?php
    }
}
} else { echo 'No flights found'; }
}
else { die(mysql_error()); }
echo '<legend>Flights from '.$to.' to
'.$fro.'</legend>';
if(isset($_GET['return_date'])===true) {
    $returndate = $_GET['return_date'];
    $query2 = "SELECT * FROM `flight search` WHERE
`from_city`= '$to' AND `to_city` = '$fro' AND `departure_date` =
'$returndate'";
    $result2 = mysql_query($query2);
    if($result2) {
        if(mysql_num_rows($result2) > 0) {
            while($row2 = mysql_fetch_assoc($result2)) {
                ?>
                <table class="table">
                <thead>
                <tr>
                <th>Flight No</th>
                <th>Departure Date</th>
                <th>Arrival Date</th>
                <th>Departure Time</th>
                <th>Arrival Time</th>
                <th>Seats Left</th>
                <th>Price</th>
                </tr>
                </thead>
                <tbody>
                <tr>
                <?php if($class==='Economy') { ?>
                <td><input type="radio" required name="chose_fro"
value="<?php echo $row2['fno']; ?>"/><?php echo $row2['fno'];
?></td>
                <td><?php echo $row2['departure_date']; ?></td>
                <td><?php echo $row2['arrival_date']; ?></td>
                <td><?php echo $row2['departure_time']; ?></td>
                <td><?php echo $row2['arrival_time']; ?></td>
                <td><?php echo $row2['e_seats_left']; ?></td>
                <td><?php echo $row2['e_price']; ?>€</td>
                </tr>
                </tbody>
                </table><table class="table"> <?php
            } else if($class==='Business') { ?>
            <tr>
            <td><input type="radio" required name="chose_fro"
value="<?php echo $row2['fno']; ?>"/><?php echo $row2['fno'];
?></td>
            <td><?php echo $row2['departure date']; ?></td>
            <td><?php echo $row2['arrival date']; ?></td>
            <td><?php echo $row2['departure_time']; ?></td>
            <td><?php echo $row2['arrival time']; ?></td>

```

```

        <td><?php echo $row2['b_seats_left']; ?></td>
        <td><?php echo $row2['b_price']; ?>€</td>
    </tr>
</table>
</table>
<?php }
}??
<input type="hidden" name="count_a" value="<?php echo
$counta; ?>"/>
<input type="hidden" name="count_c" value="<?php echo
$countc; ?>"/>
<input type="hidden" name="class" value="<?php echo
$class; ?>"/>
<center><input type="submit" id="submit2"
value="Choose Flights" name="submit2" class="btn btn-primary"
onsubmit="submitForms()"></center>
<?php
} else { echo 'No flights found'; }
}
else { die(mysql_error()); }
}
}
}
else { ?>

<h3>Services offered by Our Airlines</h3>
<h5>Travel across the world through our flights which
are getting updated in no time</h5>
<h5>Free 3 course meals for every passenger</h5>
<h5>Free Wi-Fi services offered during your
flight</h5>
<h5>Up to 25 Kg Baggage limit for every passenger</h5>
<h5>Unlimited Food and Alcohol in Business Class</h5>
<h5>Book hotels via Our Airlines and avail 10% discount
per room</h5>
<h5>Discounted travel coupons for every city</h5>
<h5>Hire cabs or bus via Our Airlines and avail 10%
discount per ticket</h5>
<h5>Avail wheelchairs and emergency services for
disabled at free of cost</h5>

<?php } ?>
<script text="text/javascript">
function submitForms() {
    "<?php if(isset($_POST["submit2"])){
        if(f_logged_in()===true){ ?>"
        document.getElementById("form1").submit();
    "<?php } else { ?>"
        document.getElementById("form2").submit();
    "<?php }
    }
    ?>"
}
}

```

Κώδικας ο οποίος κάνει αναζήτηση στη βάση δεδομένων MySQL για διαθέσιμες πτήσεις βάσει των στοιχείων της φόρμας αναζήτησης και τις τυπώνει στην οθόνη με όλες τις πληροφορίες τους δίνοντας την δυνατότητα στο χρήστη να επιλέξει αυτή που ο ίδιος επιθυμεί. Μετά την επιλογή του αποθηκεύονται μέσω μέθοδο αιτημάτων

HTTP GET μερικές από τις πληροφορίες σε μεταβλητές (όνομα πτήσης, αριθμός επιβατών, είδος πτήσης κλπ) ώστε να διατηρηθούν κατά τη διαδικασία της κράτησης.

### book.php

```
<?php if(f_logged_in()===true) {
    if(isset($_GET['class'])===true &&
    isset($_GET['count_a'])===true && isset($_GET['count_c'])===true) {
        $to = $_GET['chose_to'];
        $fro = $_GET['chose_fro'];
        $sql = "SELECT * FROM `flight_search` WHERE `fno`='\$to'";
        $rul = mysql_query($sql);
        while($rol = mysql_fetch_assoc($rul)) {
            $arr1 = $rol['arrival_time'];
        }
        $qu2 = "SELECT * FROM `flight_search` WHERE
`fno`='\$fro'";
        $ru2 = mysql_query($qu2);
        while($ro2 = mysql_fetch_assoc($ru2)) {
            $dep2 = $ro2['departure_time'];
        }
        if($dep2 < $arr1) {
            if(isset($_GET['chose_to'])===true) {
                if(isset($_GET['chose_fro'])===true) {
                    $to = $_GET['chose_to'];
                    $fro = $_GET['chose_fro'];
                    $counta = $_GET['count_a'];
                    $countc = $_GET['count_c'];
                    $class = $_GET['class'];
                    $q1 = "SELECT * FROM `flight_search` WHERE
`fno`='\$to'";
                    $r1 = mysql_query($q1);
                    while($row1 = mysql_fetch_assoc($r1)) {
                        echo '<legend>Your flight from
' . $row1['from_city'] . ' to ' . $row1['to_city'] . '</legend>';
                        echo 'Flight Number:
' . $row1['fno'] . '<br>';
                        if($class==='Economy') {
                            if (($counta+$countc)<=$row1['e_seats_left']) {
                                echo 'Departure Time:
' . $row1['departure time'] . '<br>';
                                echo 'Arrival Time:
' . $row1['arrival time'] . '<br>';
                                echo 'Number of Adults:
' . $counta . '<br>';
                                echo 'Number of Children:
' . $countc . '<br><br>';
                                $seat1 = $counta *
$row1['e_price'];
                                echo 'Cost of ' . $counta .
adult(s): ' . $counta . ' * ' . $row1['e_price'] . ' = ' . $seat1 . '€<br>';
                                $sect1 = ($countc *
$row1['e_price']) * 0.75;
                                echo 'Cost of ' . $countc . ' child:
75% of ' . $countc . ' * ' . $row1['e_price'] . ' = ' . $sect1 . '€<br>';
                                $etax1 = 50;
                                echo 'Airport Tax: 10€ <br>';
                                echo 'Fuel Tax: 10€ <br>';
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

                                echo 'Service Tax: 20€ <br>';
                                echo      'Total      Tax:
'. $setax1.'€<br><br>';
                                $tot1 = $eat1 + $ect1 + $etax1;
                                echo '<legend>Cost of one trip
is '. $setot1.'€</legend><hr>';
                                }
                                else {
                                echo      'Not      enough      available
seats. Sorry, please check some other flight!';
                                header("refresh:10;
url=index.php");
                                }
                                }
                                else if($class==='Business') {
if(($counta+$countc)<=$row1['b seats left']) {
                                echo      'Departure      Time:
'. $row1['departure time'].'<br>';
                                echo      'Arrival      Time:
'. $row1['arrival_time'].'<br>';
                                echo      'Number      of      Adults:
'. $counta.'<br>';
                                echo      'Number      of      Children:
'. $countc.'<br><br>';
                                $bat1      =      $counta      *
                                echo      'Cost      of      '. $counta.'
adult(s): '. $counta.' *      '. $row1['b_price'].' = '. $bat1.'€<br>';
                                $bct1      =      ($countc      *
                                echo      'Cost of '. $countc.' child:
75% of '. $countc.' *      '. $row1['b_price'].' = '. $bct1.'€<br>';
                                $btax1 = 150;
                                echo      'Airport Tax: 20€ <br>';
                                echo      'Fuel Tax: 20€ <br>';
                                echo      'Service Tax: 110€ <br>';
                                echo      'Total
Tax: '. $btax1.'€<br><br>';
                                $btot1 = $bat1 + $bct1 + $btax1;
                                echo '<legend>Cost of one trip
is '. $btot1.'€</legend><hr>';
                                }
                                else {
                                echo      'Not      enough      available
seats. Sorry, please check some other flight!';
                                header("refresh:10;
url=index.php");
                                }
                                }
                                }
                                $q2 = "SELECT * FROM `flight_search` WHERE
`fno`=' $fro'";
                                $r2 = mysql_query($q2);
                                while($row2 = mysql_fetch_assoc($r2)) {
                                echo      '<legend>Your      flight      from
'. $row2['from city'].' to '. $row2['to city'].'</legend>';
                                echo      'Flight      Number:
'. $row2['fno'].'<br>';
                                if($class==='Economy') {

```

```

if(( $\$counta+\$countc$ )<= $\$row2['e\_seats\_left']$ ) {
    echo      'Departure      Time:
'. $\$row2['departure\ time']$ .'  
';
    echo      'Arrival        Time:
'. $\$row2['arrival\ time']$ .'  
';
    echo      'Number      of      Adults:
'. $\$counta$ .'  
';
    echo      'Number      of      Children:
'. $\$countc$ .'  
<br>';
     $\$seat2$     =       $\$counta$     *
 $\$row2['e\ price']$ ;
    echo      'Cost      of      '. $\$counta$ .'
adult(s): '. $\$counta$ .' * '. $\$row2['e\_price']$ .' = '. $\$seat2$ .'  
';
     $\$sect2$     =      ( $\$countc$     *
 $\$row2['e\ price']$ ) * 0.75;
    echo      'Cost of '. $\$countc$ .' child:
75% of '. $\$countc$ .' * '. $\$row2['e\_price']$ .' = '. $\$sect2$ .'  
';
     $\$setax2$  = 50;
    echo      'Airport Tax: 10€ <br>';
    echo      'Fuel Tax: 10€ <br>';
    echo      'Service Tax: 30€ <br>';
    echo      'Total          Tax:
'. $\$setax2$ .'  
<br>';
     $\$setot2$  =  $\$seat2$  +  $\$sect2$  +  $\$setax2$ ;
    echo      '<legend>Cost of return
trip is '. $\$setot2$ .'  
</legend><hr>';
    }
    else {
        echo      'Not enough available
seats. Sorry, please check some other flight!';
        header("refresh:10;
url=index.php");
    }
}
else if( $\$class===$ 'Business') {
if(( $\$counta+\$countc$ )<= $\$row2['b\_seats\_left']$ ) {
    echo      'Departure      Time:
'. $\$row2['departure\ time']$ .'  
';
    echo      'Arrival        Time:
'. $\$row2['arrival\_time']$ .'  
';
    echo      'Number      of      Adults:
'. $\$counta$ .'  
';
    echo      'Number      of      Children:
'. $\$countc$ .'  
<br>';
     $\$bat2$     =       $\$counta$     *
 $\$row2['b\ price']$ ;
    echo      'Cost      of      '. $\$counta$ .'
adult(s): '. $\$counta$ .' * '. $\$row2['b\_price']$ .' = '. $\$bat2$ .'  
';
     $\$bct2$     =      ( $\$countc$     *
 $\$row2['b\_price']$ ) * 0.75;
    echo      'Cost of '. $\$countc$ .' child:
75% of '. $\$countc$ .' * '. $\$row2['b\_price']$ .' = '. $\$bct2$ .'  
';
     $\$btax2$  = 150;
    echo      'Airport Tax: 20€ <br>';
    echo      'Fuel Tax: 20€ <br>';
    echo      'Service Tax: 110€ <br>';
    echo      'Total          Tax:
'. $\$btax2$ .'  
<br>';
     $\$btot2$  =  $\$bat2$  +  $\$bct2$  +  $\$btax2$ ;

```



```

                                echo '€<legend>Cost of return
trip is ' . $btot2 . '€</legend><hr>';
                                }
                                else {
                                    echo 'Not enough available
seats. Sorry, please check some other flight!';
                                    header("refresh:10;
url=index.php");
                                }
                                }
                                }
                                if($class==='Economy') {
                                    $setotr = $setot1 + $setot2;
                                    echo '<legend>Total cost of both the
trips is ' . $setotr . '€</legend>';
                                }
                                else if($class==='Business') {
                                    $btotr = $btot1 + $btot2;
                                    echo '<legend>Total cost of both the
trips is ' . $btotr . '€</legend>';
                                }
                                }
                                else {
                                    $sto = $_GET['chose_to'];
                                    $counta = $_GET['count_a'];
                                    $countc = $_GET['count_c'];
                                    $class = $_GET['class'];
                                    $q1 = "SELECT * FROM `flight_search` WHERE
`fno`=' $sto ";
                                    $r1 = mysql_query($q1);
                                    while($row1 = mysql_fetch_assoc($r1)) {
                                        echo '<legend>Your flight from
' . $row1['from city'] . ' to ' . $row1['to city'] . '</legend>';
                                        echo 'Flight Number:
' . $row1['fno'] . '<br>';
                                        if($class==='Economy') {
                                            if (($counta+$countc) <= $row1['e_seats_left']) {
                                                echo 'Departure Time:
' . $row1['departure time'] . '<br>';
                                                echo 'Arrival Time:
' . $row1['arrival_time'] . '<br>';
                                                echo 'Number of Adults:
' . $counta . '<br>';
                                                echo 'Number of Children:
' . $countc . '<br><br>';
                                                $seat1 = $counta *
$row1['e_price'];
                                                echo 'Cost of ' . $counta .
adult(s): ' . $counta . ' * ' . $row1['e_price'] . ' = ' . $seat1 . '€<br>';
                                                $sect1 = ($countc *
$row1['e_price']) * 0.75;
                                                echo 'Cost of ' . $countc .
child:
75% of ' . $countc . ' * ' . $row1['e_price'] . ' = ' . $sect1 . '€<br>';
                                                $setax1 = 50;
                                                echo 'Airport Tax: 10€ <br>';
                                                echo 'Fuel Tax: 10€ <br>';
                                                echo 'Service Tax: 30€ <br>';
                                                echo 'Total Tax:
' . $setax1 . '€<br><br>';
                                                $setot1 = $seat1 + $sect1 + $setax1;

```

```

                                echo '<legend>Total cost of trip
is '.$tot1.'€</legend>';
                                }
                                else {
                                    echo 'Not enough available
seats. Sorry, please check some other flight!';
                                    header("refresh:10;
url=index.php");
                                }
                                }
                                else if($class==='Business') {
if (($counta+$countc) <= $row1['b_seats_left']) {
                                echo 'Departure Time:
'.$row1['departure time'].'<br>';
                                echo 'Arrival Time:
'.$row1['arrival time'].'<br>';
                                echo 'Number of Adults:
'.$counta.'<br>';
                                echo 'Number of Children:
'.$countc.'<br><br>';
                                $bat1 = $counta *
$row1['b_price'];
                                echo 'Cost of '.$counta.'
adult(s): '.$counta.' * '.$row1['b_price'].' = '.$bat1.'€<br>';
                                $bct1 = ($countc *
$row1['b_price']) * 0.75;
                                echo 'Cost of '.$countc.' child:
75% of '.$countc.' * '.$row1['b_price'].' = '.$bct1.'€<br>';
                                $btax1 = 150;
                                echo 'Airport Tax: 20€ <br>';
                                echo 'Fuel Tax: 20€ <br>';
                                echo 'Service Tax: 110€ <br>';
                                echo 'Total Tax:
'.$btax1.'<br><br>';
                                $btot1 = $bat1 + $bct1 + $btax1;
                                echo '<legend>Cost of one trip
is '.$btot1.'€</legend>';
                                }
                                else {
                                    echo 'Not enough available
seats. Sorry, please check some other flight!';
                                    header("refresh:10;
url=index.php");
                                }
                                }
                                }
                                }
                                }
                                }
                                } else { echo 'Sorry, but it appears your return flight
takes off before you land at your destination! Please select another
flight!';
                                header("refresh: 5; url=index.php");
                                }
                                }
                                } else if($class == $_SESSION['class'] && $counta ==
$_SESSION['count_a'] && $countc == $_SESSION['count_c']) {
                                    $to = $_SESSION['chose_to'];
                                    $fro = $_SESSION['chose fro'];
                                    $sql = "SELECT * FROM `flight_search` WHERE `fno`='".$to'";
                                    $rul = mysql_query($sql);

```

```

        while($ro1 = mysql_fetch_assoc($rul)) {
            $arr1 = $ro1['arrival_time'];
        }
        $qu2 = "SELECT * FROM `flight_search` WHERE
`fno`=' $fro'";
        $ru2 = mysql_query($qu2);
        while($ro2 = mysql_fetch_assoc($ru2)) {
            $dep2 = $ro2['departure_time'];
        }
        if($dep2 < $arr1) {
            if($to == $_SESSION['chose_to'] &&
$ SESSION['chose to'] != "") {
                if($fro == $_SESSION['chose_fro'] &&
$ _SESSION['chose_fro'] != "") {
                    $to = $_SESSION['chose_to'];
                    $fro = $_SESSION['chose_fro'];
                    $counta = $_SESSION['count_a'];
                    $countc = $_SESSION['count_c'];
                    $class = $_SESSION['class'];
                    $ql = "SELECT * FROM `flight_search` WHERE
`fno`=' $to'";
                    $r1 = mysql_query($ql);
                    while($row1 = mysql_fetch_assoc($r1)) {
                        echo '<legend>Your flight from
' . $row1['from_city'] . ' to ' . $row1['to_city'] . '</legend>';
                        echo 'Flight Number:
' . $row1['fno'] . '<br>';
                        if($class==='Economy') {
                            if (($counta+$countc)<=$row1['e_seats_left']) {
                                echo 'Departure Time:
' . $row1['departure_time'] . '<br>';
                                echo 'Arrival Time:
' . $row1['arrival time'] . '<br>';
                                echo 'Number of Adults:
' . $counta . '<br>';
                                echo 'Number of Children:
' . $countc . '<br><br>';
                                $seat1 = $counta *
$ row1['e price'];
                                echo 'Cost of ' . $counta .
adult(s): ' . $counta . ' * ' . $row1['e_price'] . ' = ' . $seat1 . '€<br>';
                                $sect1 = ($countc *
$ row1['e price']) * 0.75;
                                echo 'Cost of ' . $countc . ' child:
75% of ' . $countc . ' * ' . $row1['e_price'] . ' = ' . $sect1 . '€<br>';
                                $setax1 = 50;
                                echo 'Airport Tax: 10€ <br>';
                                echo 'Fuel Tax: 10€ <br>';
                                echo 'Service Tax: 20€ <br>';
                                echo 'Total Tax:
' . $setax1 . '€<br><br>';
                                $setot1 = $seat1 + $sect1 + $setax1;
                                echo '<legend>Cost of one trip
is ' . $setot1 . '€</legend><hr>';
                            }
                            else {
                                echo 'Not enough available
seats. Sorry, please check some other flight!';
                                header("refresh:10;
url=index.php");
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        }
        else if($class==='Business') {
if (($counta+$countc)<=$row1['b_seats_left']) {
                echo            'Departure           Time:
'. $row1['departure time'].'<br>';
                echo            'Arrival           Time:
'. $row1['arrival time'].'<br>';
                echo            'Number           of           Adults:
'. $counta.'<br>';
                echo            'Number           of           Children:
'. $countc.'<br><br>';
                $bat1            =           $counta            *
$row1['b price'];
                echo            'Cost           of           '. $counta.'
adult(s): '. $counta.' *           '. $row1['b_price'].' = '. $bat1.'€<br>';
                $bct1            =           ($countc            *
$row1['b price']) * 0.75;
                echo            'Cost           of           '. $countc.' child:
75%           of           '. $countc.' *           '. $row1['b_price'].' = '. $bct1.'€<br>';
                $btax1            =           150;
                echo            'Airport Tax: 20€ <br>';
                echo            'Fuel Tax: 20€ <br>';
                echo            'Service Tax: 110€ <br>';
                echo            'Total
Tax:'. $btax1.'€<br><br>';
                $btot1            =           $bat1            + $bct1            + $btax1;
                echo            '<legend>Cost           of           one           trip
is           '. $btot1.'€</legend><hr>';
        }
        else {
                echo            'Not           enough           available
seats. Sorry, please check some other flight!';
                header           ("refresh:10;
url=index.php");
        }
    }
}
}
}
    }
    $q2 = "SELECT * FROM `flight_search` WHERE
`fno`=' $fro'";
    $r2 = mysql_query($q2);
    while($row2 = mysql_fetch_assoc($r2)) {
        echo            '<legend>Your           flight           from
'. $row2['from city'].' to '. $row2['to city'].'</legend>';
        echo            'Flight           Number:
'. $row2['fno'].'<br>';
        if($class==='Economy') {
if (($counta+$countc)<=$row2['e_seats_left']) {
                echo            'Departure           Time:
'. $row2['departure_time'].'<br>';
                echo            'Arrival           Time:
'. $row2['arrival time'].'<br>';
                echo            'Number           of           Adults:
'. $counta.'<br>';
                echo            'Number           of           Children:
'. $countc.'<br><br>';
                $seat2            =           $counta            *
$row2['e price'];

```

```

                                echo 'Cost of '.$counta.'
adult(s): '.$counta.' * '.$row2['e_price'].' = '.$seat2.'  
';
                                $sect2 = ($countc *
$row2['e_price']) * 0.75;
                                echo 'Cost of '.$countc.' child:
75% of '.$countc.' * '.$row2['e_price'].' = '.$sect2.'  
';
                                $etax2 = 50;
                                echo 'Airport Tax: 10€ <br>';
                                echo 'Fuel Tax: 10€ <br>';
                                echo 'Service Tax: 30€ <br>';
                                echo 'Total Tax:
'.$etax2.'  
<br>';
                                $etot2 = $seat2 + $sect2 + $etax2;
                                echo '<legend>Cost of return
trip is '.$etot2.'  
</legend><hr>';
                                }
                                else {
                                echo 'Not enough available
seats. Sorry, please check some other flight!';
                                header("refresh:10;
url=index.php");
                                }
                                }
                                else if($class==='Business') {
if(($counta+$countc)<=$row2['b_seats_left']) {
                                echo 'Departure Time:
'.$row2['departure time'].'<br>';
                                echo 'Arrival Time:
'.$row2['arrival time'].'<br>';
                                echo 'Number of Adults:
'.$counta.'  
';
                                echo 'Number of Children:
'.$countc.'  
<br>';
                                $bat2 = $counta *
$row2['b_price'];
                                echo 'Cost of '.$counta.'
adult(s): '.$counta.' * '.$row2['b_price'].' = '.$bat2.'  
';
                                $bct2 = ($countc *
$row2['b_price']) * 0.75;
                                echo 'Cost of '.$countc.' child:
75% of '.$countc.' * '.$row2['b_price'].' = '.$bct2.'  
';
                                $btax2 = 150;
                                echo 'Airport Tax: 20€ <br>';
                                echo 'Fuel Tax: 20€ <br>';
                                echo 'Service Tax: 110€ <br>';
                                echo 'Total Tax:
'.$btax2.'  
<br>';
                                $btot2 = $bat2 + $bct2 + $btax2;
                                echo '<legend>Cost of return
trip is '.$btot2.'  
</legend><hr>';
                                }
                                else {
                                echo 'Not enough available
seats. Sorry, please check some other flight!';
                                header("refresh:10;
url=index.php");
                                }
                                }
                                }
                                if($class==='Economy') {

```

```

        $setotr = $setot1 + $setot2;
        echo '<legend>Total cost of both the
trips is '.$setotr.'€</legend>';
    }
    else if($class==='Business') {
        $bsetotr = $bsetot1 + $bsetot2;
        echo '<legend>Total cost of both the
trips is '.$bsetotr.'€</legend>';
    }
}
else {
    $sto = $_SESSION['chose_to'];
    $counta = $_SESSION['count_a'];
    $countc = $_SESSION['count_c'];
    $class = $_SESSION['class'];
    $q1 = "SELECT * FROM `flight_search` WHERE
`fno`='".$sto'";
    $r1 = mysql_query($q1);
    while($row1 = mysql_fetch_assoc($r1)) {
        echo '<legend>Your flight from
'.$row1['from_city'].' to '.$row1['to_city'].'</legend>';
        echo 'Flight Number:
'.$row1['fno'].'<br>';
        if($class==='Economy') {
if(($counta+$countc)<=$row1['e_seats_left']) {
            echo 'Departure Time:
'.$row1['departure time'].'<br>';
            echo 'Arrival Time:
'.$row1['arrival time'].'<br>';
            echo 'Number of Adults:
'.$counta.'<br>';
            echo 'Number of Children:
'.$countc.'<br><br>';
            $seat1 = $counta *
$row1['e_price'];
            echo 'Cost of '.$counta.'
adult(s): '.$counta.' * '.$row1['e_price'].' = '.$seat1.'€<br>';
            $sect1 = ($countc *
$row1['e_price']) * 0.75;
            echo 'Cost of '.$countc.' child:
75% of '.$countc.' * '.$row1['e_price'].' = '.$sect1.'€<br>';
            $setax1 = 50;
            echo 'Airport Tax: 10€ <br>';
            echo 'Fuel Tax: 10€ <br>';
            echo 'Service Tax: 30€ <br>';
            echo 'Total Tax:
'.$setax1.'€<br><br>';
            $setot1 = $seat1 + $sect1 + $setax1;
            echo '<legend>Total cost of trip
is '.$setot1.'€</legend>';
        }
        else {
            echo 'Not enough available
seats. Sorry, please check some other flight!';
            header("refresh:10;
url=index.php");
        }
    }
}
else if($class==='Business') {

```

```

if (($counta+$countc) <= $row1['b_seats_left']) {
    echo 'Departure Time:
'. $row1['departure time']. '<br>';
    echo 'Arrival Time:
'. $row1['arrival time']. '<br>';
    echo 'Number of Adults:
'. $counta. '<br>';
    echo 'Number of Children:
'. $countc. '<br><br>';
    $bat1 = $counta *
$row1['b_price'];
    echo 'Cost of '.$counta.'
adult(s): '.$counta.' * '.$row1['b_price'].' = '.$bat1.'€<br>';
    $bct1 = ($countc *
$row1['b_price']) * 0.75;
    echo 'Cost of '.$countc.' child:
75% of '.$countc.' * '.$row1['b_price'].' = '.$bct1.'€<br>';
    $btax1 = 150;
    echo 'Airport Tax: 20€ <br>';
    echo 'Fuel Tax: 20€ <br>';
    echo 'Service Tax: 110€ <br>';
    echo 'Total Tax:
'. $btax1. '<br><br>';
    $btot1 = $bat1 + $bct1 + $btax1;
    echo '<legend>Cost of one trip
is '.$btot1.'€</legend>';
}
else {
    echo 'Not enough available
seats. Sorry, please check some other flight!';
    header("refresh:10;
url=index.php");
}
}
}
}
}
} else { echo 'Sorry, but it appears your return flight
takes off before you land at your destination! Please select another
flight!';
    header("refresh: 5; url=index.php");
}
}
}
?>

```

Κώδικας που εμφανίζει στη σελίδα το συνολικό κόστος της πτήσης βάσει των στοιχείων που υπέβαλε ο χρήστης με την προϋπόθεση πάντα ότι είναι συνδεδεμένος στο λογαριασμό του. Επίσης, γίνεται έλεγχος για τον αριθμό διαθέσιμων θέσεων και για το αν οι ημερομηνίες επιτρέπουν την πραγματοποίηση της πτήσης.

```

if ($counta == 2 && $countc == 1) {?>
    <h4>Adult Passenger Details</h4>
    <div class="form-group">
        <div class="col-lg-6">

```

```

        <input type="text" class="form-control"
<?php if($counta==='1' || $counta==='2' || $counta==='3' ||
$counta==='4' || $counta==='5') echo 'required'; ?> name="aname1"
id="inputEmail" placeholder="Name">
        </div>
        <div class="col-lg-3">
            <input type="text" class="form-control"
<?php if($counta==='1' || $counta==='2' || $counta==='3' ||
$counta==='4' || $counta==='5') echo 'required'; ?> name="aage1"
id="inputEmail" placeholder="Age">
            </div>
            <div class="col-lg-3">
                <select class="form-control" name="asex1"
<?php if($counta==='1' || $counta==='2' || $counta==='3' ||
$counta==='4' || $counta==='5') echo 'required'; ?> id="select">
                    <option value="M">Male</option>
                    <option value="F">Female</option>
                </select>
            </div>
        </div>
        <div class="form-group">
            <div class="col-lg-6">
                <input type="text" class="form-control"
<?php if($counta==='1' || $counta==='2' || $counta==='3' ||
$counta==='4' || $counta==='5') echo 'required'; ?> name="aname2"
id="inputEmail" placeholder="Name">
            </div>
            <div class="col-lg-3">
                <input type="text" class="form-control"
<?php if($counta==='1' || $counta==='2' || $counta==='3' ||
$counta==='4' || $counta==='5') echo 'required'; ?> name="aage2"
id="inputEmail" placeholder="Age">
            </div>
            <div class="col-lg-3">
                <select class="form-control" name="asex2"
<?php if($counta==='1' || $counta==='2' || $counta==='3' ||
$counta==='4' || $counta==='5') echo 'required'; ?> id="select">
                    <option value="M">Male</option>
                    <option value="F">Female</option>
                </select>
            </div>
        </div>
        <hr>
        <h4>Child Passenger Details</h4>
        <div class="form-group">
            <div class="col-lg-6">
                <input type="text" class="form-control"
name="cname1" <?php if($countc==='1' || $countc==='2' ||
$countc==='3' || $countc==='4' || $countc==='5') echo 'required'; ?>
id="inputEmail" placeholder="Name">
            </div>
            <div class="col-lg-3">
                <input type="text" class="form-control"
name="cage1" <?php if($countc==='1' || $countc==='2' ||
$countc==='3' || $countc==='4' || $countc==='5') echo 'required'; ?>
id="inputEmail" placeholder="Age">
            </div>
            <div class="col-lg-3">
                <select class="form-control" name="csex1"
<?php if($countc==='1' || $countc==='2' || $countc==='3' ||
$countc==='4' || $countc==='5') echo 'required'; ?> id="select">

```



```

        <option value="M">Male</option>
        <option value="F">Female</option>
    </select>
</div>
</div>
<?php }

```

Κώδικας που εμφανίζει μια φόρμα όπου συμπληρώνονται τα ονόματα, οι ηλικίες και το φύλο των πελατών. Το συγκεκριμένο παράδειγμα βασίζεται στην κράτησης εισιτηρίων δύο ενηλίκων και ενός ανηλίκου.

```

<h4>User Contact Details</h4>
    <div class="form-group">
        <div class="col-lg-12">
            <label class="control-label"
for="focusedInput">Name</label>
            <input class="form-control" name="puname"
id="puname" value="<?php echo $f_data['f_fname'].
'. $f_data['f_lname']; ?>" required type="text">
        </div>
    </div>
    <div class="form-group">
        <div class="col-lg-12">
            <label class="control-label"
for="focusedInput">Phone Number</label>
            <input class="form-control" name="puphno"
id="puphno" value="<?php echo $f_data['f_phone']; ?>" required
type="text">
        </div>
    </div>
    <div class="form-group">
        <div class="col-lg-12">
            <label class="control-label"
for="focusedInput">Address</label>
            <input class="form-control" name="puadd"
id="puadd" value="<?php echo $f_data['f_address']; ?>" required
type="text">
        </div>
    </div>
    <div class="form-group">
        <div class="col-lg-12">
            <label class="control-label"
for="focusedInput">Email ID</label>
            <input class="form-control" name="pumail"
id="pumail" value="<?php echo $f_data['f_mailid']; ?>" required
type="text">
        </div>
    </div>

```

Κώδικας που τυπώνει τα στοιχεία του λογαριασμού του χρήστη όπως ονοματεπώνυμο, τηλέφωνο, email και διεύθυνση κατοικίας ώστε να αναγνωρίζει το σύστημα σε ποιανού χρήστη το λογαριασμό πρέπει να γίνει η κράτηση.

## confirm.php

```
if(f_logged_in()===true) {
    $uid = $f_data['f_id'];
    $pname = $_POST['pname'];
    $puphno = $_POST['puphno'];
    $puadd = $_POST['puadd'];
    $pumail = $_POST['pumail'];
    if(isset($_POST['chose_to'])===true) {
        if(isset($_POST['chose_fro'])=== true) {
            $counta = $_POST['count_a'];
            $countc = $_POST['count_c'];
            $class = $_POST['class'];
            $to = $_POST['chose_to'];
            $fro = $_POST['chose_fro'];
        }
        if($class==='Business' && $counta==='2' && $countc==='1') {
            $aname1 = $_POST['aname1'];
            $asex1 = $_POST['asex1'];
            $aage1 = $_POST['aage1'];
            $aname2 = $_POST['aname2'];
            $asex2 = $_POST['asex2'];
            $aage2 = $_POST['aage2'];
            $cname1 = $_POST['cname1'];
            $csex1 = $_POST['csex1'];
            $cage1 = $_POST['cage1'];

            $btotr = $_POST['btotr'];
            $totc = $counta + $countc;
            $type = 'Heading';
            $q1 = "SELECT * FROM `flight_search` WHERE `fno`='{$to}'";
            $r1 = mysql_query($q1);
            while($row1 = mysql_fetch_assoc($r1)) {
                $from_city = $row1['from_city'];
                $to_city = $row1['to_city'];
                $depart_date = $row1['departure_date'];
                $arr_date = $row1['arrival_date'];
                $depart_time = $row1['departure_time'];
                $arr_time = $row1['arrival_time'];
            }

            $q11 = "SELECT `b_seats_left` FROM `flight_search` WHERE `fno`='{$to}'";
            $r11 = mysql_query($q11);
            if($r11) {
                if(mysql_num_rows($r11)>0) {
                    while($row11 = mysql_fetch_assoc($r11)) {
                        $b_seats_left = $row11['b_seats_left'];
                        $dec_seats11 = $b_seats_left - $totc;

                        mysql_query("UPDATE `flight_search` SET `b_seats_left`='{$dec_seats11}' WHERE `fno`='{$to}');
                    }
                }

                $a1 = date('H:i:s');
                $e1 = str_replace(':', '', $a1);
                $a2 = date('d-m-Y');
                $e2 = str_replace('-', '', $a2);

                $pnr1 = $e1.$e2;
                $pnr2 = $e1.$e2+1;
            }
        }
    }
}
```

```

$pnr3 = $e1.$e2+2;
mysql_query("INSERT INTO `passenger_details` (`p_name`, `p_sex`,
`p_age`, `p_pnr`, `p_fno`, `p_from`, `p_to`, `p_dedate`, `p_ardate`,
`p_detime`, `p_artime`, `p_class`, `p_type`, `p_passtype`, `p_fid`)
VALUES ('$aname1', '$asex1', '$aage1', '$pnr1', '$to', '$from_city',
'$to_city', '$depart_date', '$arr_date', '$depart_time',
'$arr_time', '$class', '$type', 'A', '$uid'),('$aname2', '$asex2',
'$aage2', '$pnr2', '$to', '$from_city', '$to_city', '$depart_date',
'$arr_date', '$depart_time', '$arr_time', '$class', '$type', 'A',
'$uid'),('$cname1', '$csex1', '$cage1', '$pnr3', '$to',
'$from_city', '$to_city', '$depart_date', '$arr_date',
'$depart_time', '$arr_time', '$class', '$type', 'C', '$uid')");

mysql_query("INSERT INTO `booking_details` (`b_name`, `b_phno`,
`b_mail`, `b_add`, `b_price`, `b_child`, `b_adults`, `b_total`,
`b_type`, `b_pnr`, `b_fid`) VALUES ('$pname', '$puphno', '$pumail',
'$puadd', '$btotr', '$countc', '$counta', '$totc', '$type', '$pnr1',
'$uid')");
}
$type = 'Returning';
$q12 = "SELECT * FROM `flight_search` WHERE `fno`='$fro'";
$r12 = mysql_query($q12);
if($r12) {
if(mysql_num_rows($r12)>0) {
while($row12 = mysql_fetch_assoc($r12)) {
$from_city = $row12['from_city'];
$to_city = $row12['to_city'];
$depart_date = $row12['departure_date'];
$arr_date = $row12['arrival_date'];
$depart_time = $row12['departure_time'];
$arr_time = $row12['arrival_time'];
$b_seats_left = $row12['b_seats_left'];
$dec_seats12 = $b_seats_left - $totc;
mysql_query("UPDATE `flight_search` SET
`b_seats_left`='$dec_seats12' WHERE `fno`='$fro'");
}
}
}
$a5 = date('H:i:s');
$e5 = str_replace(':', '', $a5);
$a6 = date('d-m-Y');
$e6 = str_replace('-', '', $a6);
$pnr11 = $e5.$e6+100;
$pnr12 = $e5.$e6+101;
$pnr13 = $e5.$e6+102;
mysql_query("INSERT INTO `passenger_details` (`p_name`, `p_sex`,
`p_age`, `p_pnr`, `p_fno`, `p_from`, `p_to`, `p_dedate`, `p_ardate`,
`p_detime`, `p_artime`, `p_class`, `p_type`, `p_passtype`, `p_fid`)
VALUES ('$aname1', '$asex1', '$aage1', '$pnr11', '$fro',
'$from_city', '$to_city', '$depart_date', '$arr_date',
'$depart_time', '$arr_time', '$class', '$type', 'A',
'$uid'),('$aname2', '$asex2', '$aage2', '$pnr12', '$fro',
'$from_city', '$to_city', '$depart_date', '$arr_date',
'$depart_time', '$arr_time', '$class', '$type', 'A',
'$uid'),('$cname1', '$csex1', '$cage1', '$pnr13', '$fro',
'$from_city', '$to_city', '$depart_date', '$arr_date',
'$depart_time', '$arr_time', '$class', '$type', 'C', '$uid')");

```

```

mysql_query("INSERT INTO `booking_details` (`b_name`, `b_phno`,
`b mail`, `b add`, `b price`, `b child`, `b adults`, `b total`,
`b_type`, `b_pnr`, `b_fid`) VALUES ('$pname', '$puphno', '$pumail',
'$puadd', '$btotr', '$countc', '$counta', '$totc', '$type',
'$pnr11', '$uid')");

        session_destroy();
        echo 'Tickets Successfully Booked!
Please check your bookings page for details!';
        header('refresh: 10, url=index.php');
    }
    else {
        session_destroy();
        echo 'Invalid details provided, tickets
not booked';
        header('refresh: 10, url=index.php');
    }
}
}

```

Κώδικας που πραγματοποιεί την κράτηση των εισιτηρίων. Στο παράδειγμα αυτό, πραγματοποιείται κράτηση εισιτηρίων για επαγγελματικό ταξίδι για δύο ενήλικους και έναν ανήλικο.

## init.php

```

<?php
    ob_start();
    session_start();
    //error_reporting(0);

    require
$ SERVER["DOCUMENT_ROOT"].'/Airlines/core/database/connect.php';
    require
$ SERVER["DOCUMENT_ROOT"].'/Airlines/core/functions/general.php';
    require
$_SERVER["DOCUMENT_ROOT"].'/Airlines/core/functions/users.php';

    $current_file = $_SERVER['SCRIPT_NAME'];
    $current_file = explode('/', $current_file);
    $current_file = end($current_file);

    if(f_logged_in() === true) {
        $session_f_id = $_SESSION['f_id'];
        $f_data
=
f_data($session_f_id, 'f_id', 'f_uname', 'f_password', 'f_fname', 'f_lname', 'f_address', 'f_phone', 'f_mailid', 'f_sex', 'f_regdate', 'f_passrec');
        if(f_active($f_data['f_uname']) === false) {
            session_destroy();
            header('Location:
http://localhost/Airlines/index.php');
            exit();
        }

        if($current_file !== 'changepass.php' && $current_file !==
'logout.php' && $f_data['f_passrec'] == 1) {
            header('Location:
http://localhost/Airlines/changepass.php?force');
            exit();
        }
    }
}

```

```

    }
}

$errors = array();

?>

```

Κώδικας που διατηρεί τις προσωπικές πληροφορίες του χρήστη όσο είναι συνδεδεμένος στο λογαριασμό του. Στην περίπτωση που γίνει αποσύνδεση οι πληροφορίες παύουν να υφίστανται μέχρι την επόμενη σύνδεση. Τέλος, στην περίπτωση που ο χρήστης έχει δηλώσει πως έχει ξεχάσει τον κωδικό πρόσβασης του λογαριασμού του μεταφέρεται στη σελίδα όπου μπορεί να δώσει ένα καινούργιο εφόσον συνδεθεί με τον κωδικό που το σύστημα του έδωσε για να συνδεθεί.

## head.php

```

<!-- Website head placed here -->

<head>

    <meta charset="utf-8">

    <title><?php if (isset($title)) {echo $title;}
        else {echo "Airlines";} ?></title>

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

    <link rel="stylesheet" href="./vendor/css/bootstrap.min.css"
media="screen">
    <link rel="stylesheet" href="./vendor/css/bootswatch.min.css">

    <script src="https://code.jquery.com/jquery-
1.10.2.min.js"></script>
    <script src="//code.jquery.com/ui/1.11.0/jquery-
ui.js"></script>
    <link rel="stylesheet" type="text/css"
href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base
/jquery-ui.css" />
    <script src="./vendor/js/bootstrap.min.js"></script>
    <script src="./vendor/js/bootswatch.js"></script>

    <script type="text/javascript">
        $(document).ready(function() {
            $("#departure_date").datepicker({
                minDate: 0,
                dateFormat: "dd-mm-yy"
            });
            $("#return_date").datepicker({
                minDate: 1,
                dateFormat: "dd-mm-yy"
            });
        });
        function setReadOnly(obj){

```

```

        if(obj.value == "oneway"){
            $('#departure_date').datepicker('enable');
            $('#return_date').datepicker('disable');
        }
        else {
            $('#departure_date').datepicker('enable');
            $('#return_date').datepicker('enable');
        }
    }
</script>
</head>

```

Κώδικας που συνδέει αρχεία .css και .js που διαμορφώνουν το design της εφαρμογής, αλλά και ορίζει τις επιλογές του ημερολογίου στο οποίο ο χρήστης κάνει την αναζήτηση για την πτήση που επιθυμεί.

### flogin.php

```

<?php
    $title = 'Our Airlines | Login';
    include $_SERVER["DOCUMENT_ROOT"].'/Airlines/core/init.php';
    f_logged_in_redirect();
    include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/header.php';

if(isset($_GET['class'])===true && isset($_GET['count_a'])===true &&
isset($_GET['count_c'])===true) {
    $counta = $_GET['count_a'];
    $countc = $_GET['count_c'];
    $class = $_GET['class'];
    $_SESSION['count_a'] = $counta;
    $_SESSION['count_c'] = $countc;
    $_SESSION['class'] = $class;
    if(isset($_GET['chose_to'])===true) {
        $to = $_GET['chose_to'];
        $_SESSION['chose_to'] = $to;
        if(isset($_GET['chose_fro'])===true) {
            $fro = $_GET['chose_fro'];
            $_SESSION['chose_fro'] = $fro;
        }
    }
}

?>

    <form class="form-horizontal" action="loginact.php"
method="POST">
        <div class="row">
            <div class="col-lg-4">
                <div class="well bs-component">
                    <legend>Passenger Log In</legend>
                    <div class="form-group">
                        <label for="inputEmail" class="col-lg-2 control-
label">Username</label>
                        <div class="col-lg-12">
                            <input type="text" name="f_uname" class="form-
control" required placeholder="Username">
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </form>

```

```

        </div>
        <div class="form-group">
            <label for="inputPassword" class="col-lg-2
control-label">Password</label>
            <div class="col-lg-12">
                <input type="password" name="f_password"
class="form-control" required placeholder="Password">
            </div>
        </div>
        <div class="form-group">
            <center><button type="submit" id="submit"
value="submit" name="submit" class="btn btn-
success">Login</button></center>
        </div>
    </form>
    <strong><a
href="http://localhost/Airlines/fregister.php"
style="color:white">Register Here</a></strong><br/><br/>
    <strong><a
href="http://localhost/Airlines/recover.php?mode=f_uname"
style="color:white">Forgot Username?</a></strong><br/>
    <strong><a
href="http://localhost/Airlines/recover.php?mode=f_password"
style="color:white">Forgot Password?</a></strong><br/>
</form>
</div>
</div></div>

<?php include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/footer.php';
?>

```

Κώδικας που δημιουργεί τη φόρμα login για να συνδεθεί ο χρήστης με το λογαριασμό του. Στη περίπτωση που δεν θυμάται είτε το username ή το password υπάρχουν διαθέσιμες επιλογές ανάκτησής τους. Τέλος, αποθηκεύονται σε Sessions, πληροφορίες πτήσεων στην περίπτωση που ο χρήστης επιχειρήσει να κλείσει μια πτήση πριν συνδεθεί στο λογαριασμό του.

## loginact.php

```

<?php
    $title = 'Our Airlines | Login Error';
    include $_SERVER["DOCUMENT_ROOT"].'/Airlines/core/init.php';

    $to = $fro = $counta = $countc = $path = $class = "";

    if(isset($_SESSION['chose_fro'])) {
        $fro = $_SESSION['chose_fro'];
    }
    if(isset($_SESSION['chose_to'])) {
        $to = $_SESSION['chose_to'];
    }
    if(isset($_SESSION['class'])) {
        $class = $_SESSION['class'];
    }
}

```

```

if(isset($_SESSION['path'])) {
    $path = $_SESSION['path'];
}
if(isset($_SESSION['count_a'])) {
    $counta = $_SESSION['count_a'];
}
if(isset($_SESSION['count_c'])) {
    $countc = $_SESSION['count_c'];
}

if(empty($_POST) === false) {
    $f_username = $_POST['f_username'];
    $f_password = $_POST['f_password'];

    if(empty($f_username) === true || empty($f_password) === true){
        $errors[] = 'You need to enter both, the username and
the password!';
    }
    else if (f_exists($f_username)===false) {
        $errors[] = 'No such user exists! Please register!';
    }
    else if(f_active($f_username)===false) {
        $errors[] = 'Please activate your account!';
    }
    else {

        if(strlen($f_password)>32) {
            $errors[] = 'Password too long!';
        }

        $f_login = f_login($f_username, $f_password);
        if($f_login===false) {
            $errors[] = 'Username and Password do not match!';
        }
        else {
            $_SESSION['f_id'] = $f_login;
            if($to == $_SESSION['chose_to'] &&
$_SESSION['chose_to'] != "") {
                $counta = $_SESSION['count_a'];
                $countc = $_SESSION['count_c'];
                $class = $_SESSION['class'];
                $to = $_SESSION['chose_to'];
                $fro = $_SESSION['chose_fro'];
                $_SESSION['chose_to'] = $to;
                $_SESSION['chose_fro'] = $fro;
                $_SESSION['class'] = $class;
                $_SESSION['count_a'] = $counta;
                $_SESSION['count_c'] = $countc;
                header('Location:
http://localhost/Airlines/book.php');
                exit();
            }
            else {
                header('Location:
http://localhost/Airlines/index.php');
                exit();
            }
        }
    }
}
else {

```



```

        $errors[] = 'No Log In credentials received!';
    }

    include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/header.php';

    if(empty($errors) === false) {
?>
    <br/><h4>We tried to log you in, but : </h4><br/>

<?php
    echo output_errors($errors);
    }

    include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/footer.php';

?>

```

Κώδικας που ελέγχει για το αν στοιχεία που έδωσε ο χρήστης για να συνδεθεί είναι έγκυρα και πραγματοποιεί τη σύνδεση στο λογαριασμό του. Επίσης, στην περίπτωση που ο χρήστης επιχείρησε την κράτηση εισιτηρίων πριν συνδεθεί στο λογαριασμό του η εφαρμογή του διατηρεί τα δεδομένα και τον βοηθά να συνεχίσει τη διαδικασία κράτησης μετά τη σύνδεσή του.

## bookings.php

```

<?php
    include $_SERVER["DOCUMENT_ROOT"].'/Airlines/core/init.php';
    include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/header.php';
?>

    <legend>My Bookings</legend>
    <form action="cancel.php" method="GET">
    <?php
    if(f logged in()===false){
        header('Location:index.php');
    }

    $suid=$f_data['f_id']; //Assigns logged in id to a variable
$query1 = "SELECT b fid, sum(b price) as b price FROM
`booking_details` WHERE `b_fid` = '$suid' GROUP BY b_fid";
$result1=mysql_query($query1);
if($result1) {
    if(mysql_num_rows($result1) > 0) {
        while($row1=mysql_fetch_assoc($result1)) {
            echo "<font size='5'>Total Cost:
".$row1['b price']."€</font>";
            echo '<hr>';
        }
    }
}

    $query="SELECT * FROM `passenger_details` WHERE `p_fid` =
'$suid'"; //Sorts by date time
$result=mysql_query($query);
if($result) {

```



```

        </form>
<?php
include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/footer.php';
?>

```

Κώδικας που εμφανίζει τα στοιχεία της κράτησης του χρήστη μέσω των προσωπικών του στοιχείων. Ο χρήστης μπορεί να ακυρώσει την πτήση ή μερικά από τα εισιτήρια που επιθυμεί.

### cancel.php

```

<?php
include $_SERVER["DOCUMENT_ROOT"].'/Airlines/core/init.php';
include
$ SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/header.php';
if(f logged in()===true) {
    if(isset($_GET['delete'])===true) {
        $uid = $_f_data['f_id'];
        $pnr = $_GET['delete'];
        $query1="SELECT * FROM passenger_details WHERE p_pnr
=' $pnr ";
        $r1=mysql_query($query1);
        if($r1) {
            if(mysql_num_rows($r1)>0) {
                while($row1 = mysql_fetch_assoc($r1)) {
                    $fno = $row1['p_fno'];
                    $type = $row1['p_type'];
                    $class = $row1['p_class'];
                    $passtype = $row1['p_passtype'];
                    if($class==='Economy'){
                        $q2 = "SELECT e_seats_left FROM
flight_search WHERE fno=' $fno ";
                        $r2 = mysql_query($q2);
                        if($r2) {
                            if(mysql_num_rows($r2)>0) {
                                while($row2
mysql_fetch_assoc($r2)) {
                                    $e_seats_left =
$row2['e_seats_left'];
                                    $e_seats_left++;
                                    $q4 = "UPDATE flight_search
SET e_seats_left = ' $e_seats_left ' WHERE fno=' $fno ";
                                    mysql_query($q4);
                                }
                            }
                        }
                    }
                    else if($class==='Business'){
                        $q2 = "SELECT `b_seats_left` FROM
`flight search` WHERE `fno`=' $fno ";
                        $r2 = mysql_query($q2);
                        if($r2) {
                            if(mysql_num_rows($r2)>0) {
                                while($row2
mysql_fetch_assoc($r2)) {
                                    $b_seats_left =
$row2['b_seats_left'];
                                    $b_seats_left++;

```



```

mysql_query($q8);
}
}
    if($passtype=='C'){
        $q4 = "UPDATE booking_details SET b_child = '$schild'-1, b_total=
'$btot'-1 WHERE b_type='$stype'";
        mysql_query($q4);
        if($class=="Economy"){
            $q7 = "SELECT e_price from flight_search WHERE fno='$fno'";
            $r7 = mysql_query($q7);
            if($r7){
                if(mysql_num_rows($r7)>0) {
                    while($row7 = mysql_fetch_assoc($r7)) {
                        $eprice = $row7['e_price'];
                        $q8 = "UPDATE booking_details SET b_price = b_price - 0.75 *
'$eprice' WHERE b_type='$stype'";
                        mysql_query($q8);
                    }
                }
            }
            else
if($class=="Business"){
            $q7 = "SELECT b_price from flight_search WHERE fno='$fno'";
            $r7 = mysql_query($q7);
            if($r7){
                if(mysql_num_rows($r7)>0) {
                    while($row7 = mysql_fetch_assoc($r7)) {
                        $bprice = $row7['b_price'];
                        $q8 = "UPDATE booking_details SET b_price = b_price - 0.75 *
'$bprice' WHERE b_type='$stype'";
                        mysql_query($q8);
                    }
                }
            }
            $q8 = "UPDATE booking_details SET b price = b price - b_price /
('$adults' + 0.75 * '$schild') WHERE b_type='$stype'";
            mysql_query($q8);
        }
    }
}
}
}
}
}
}
}
    $q5 = "DELETE FROM booking_details WHERE b_total=0";
    mysql_query($q5);
    $q6 = "DELETE FROM `passenger_details` WHERE `p_pnr`='$pnr' AND
`p_fid`='$uid'";
    mysql_query($q6);
    echo 'Tickets Cancelled. You will be redirected to your bookings
page in 5 seconds!';
}
}
    //echo 'Tickets Cancelled. You will be redirected to your
bookings page in 5 seconds!';

```

```

header('refresh: 5, url=bookings.php');

include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/footer.php';
?>

```

Κώδικας που πραγματοποιεί την ακύρωση ενός εισιτηρίου από τον χρήστη και ταυτόχρονα απελευθερώνει τη θέση της πτήσης η οποία μέχρι πρότινος ήταν δεσμευμένη ενώ αφαιρείται και το χρηματικό ποσό του εισιτηρίου από την κράτηση.

### users.php

```

<?php

function f_recover($mode, $f_mailid) {
    $mode = sanitize($mode);
    $f_mailid = sanitize($f_mailid);
    $f_data = array();
    f_data(f_id from email($f_mailid), 'f_id', 'f_fname', 'f_uname');
    if ($mode == 'f_uname') {
        recovery_user_pass($f_mailid, 'Recovery: Your username',
"Hello " . $f_data['f_fname'] . ",\n\nYour username is: " .
$f_data['f_uname'] . "\n\n-Admin");
    }
    else if ($mode == 'f_password') {
        $generated_password = substr(md5(rand(999, 999999)), 0,
8);
        f_change_password($f_data['f_id'],
$generated_password);

        update_f($f_data['f_id'], array('f_passrec' => '1'));

        recovery_user_pass($f_mailid, 'Recovery: Your password',
"Hello " . $f_data['f_fname'] . ",\n\nYour new password is: " .
$generated_password . "\n\n-Admin");
    }
}

function f_activate($f_mailid, $f_mailcode) {
    $f_mailid = mysql_real_escape_string($f_mailid);
    $f_mailcode = mysql_real_escape_string($f_mailcode);
    if(mysql_result(mysql_query("SELECT COUNT(`f_id`) FROM
`flight_users` WHERE `f_mailid` = '$f_mailid' AND `f_mailcode` =
'$f_mailcode' AND `f_active` = 0"), 0) == 1) {
        mysql_query("UPDATE `flight_users` SET `f_active` = 1
WHERE `f_mailid` = '$f_mailid' ");
        return true;
    }
    else {
        return false;
    }
}

function update_f($f_id, $update_data) {
    $update = array();
    array_walk($update_data, 'array_sanitize');
    foreach ($update_data as $field => $data) {

```

```

        $update[] = '' . $field . ' = \'' . $data . '\'';
    }
    mysql_query("UPDATE `flight_users` SET " . implode(' ', $update) . "WHERE `f_id` = $f_id") or die(mysql_error());
}

function f_change_password($f_id, $f_password) {
    $f_id = (int)$f_id;
    $f_password = md5($f_password);
    mysql_query("UPDATE `flight_users` SET `f_password` = '$f_password', `f_passrec` = 0 WHERE `f_id` = $f_id");
}

function register_f($register_data) {
    array_walk($register_data, 'array_sanitize');
    $register_data['f_fname'] =
ucwords(strtolower($register_data['f_fname']));
    $register_data['f_lname'] =
ucwords(strtolower($register_data['f_lname']));
    $register_data['f_password'] =
md5($register_data['f_password']);
    $register_data['f_username'] =
strtolower($register_data['f_username']);
    $fields = '' . implode(' ', array_keys($register_data))
. ' ';
    $data = '' . implode(' ', $register_data) . ' ';

    mysql_query("INSERT INTO `flight_users` ($fields,
`f_regdate`) VALUES ($data, NOW())");
    activation($register_data['f_mailid'], 'Airlines: Activate
your account', "Hello " . $register_data['f_fname'] . ", \n\nYou need
to activate your account in order to use the features of Airlines.
Please click the link below: \n
http://localhost/Airlines/activate.php?f_mailid="
. $register_data['f_mailid'] . "&f_mailcode="
. $register_data['f_mailcode'] . " \n\n-Admin");
}

function f_data($f_id) {
    $data = array();
    $f_id = (int)$f_id;
    $func_num_args = func_num_args();
    $func_get_args = func_get_args();
    if($func_num_args > 1) {
        unset($func_get_args[0]);
        $fields = '' . implode(' ', $func_get_args) . ' ';
        $data = mysql_fetch_assoc(mysql_query("SELECT $fields
FROM `flight_users` WHERE `f_id` = $f_id"));
        return $data;
    }
}

function f_logged_in() {
    return (isset($_SESSION['f_id'])) ? true : false;
}

function f_exists($f_username) {
    $f_username = sanitize($f_username);
    $query = mysql_query("SELECT COUNT(`f_id`) FROM
`flight_users` WHERE `f_username` = '$f_username'");
    return (mysql_result($query, 0) == 1) ? true : false;
}

```

```

}

function f_email_exists($f_mailid) {
    $f_mailid = sanitize($f_mailid);
    $query = mysql_query("SELECT COUNT(`f_id`) FROM
`flight_users` WHERE `f_mailid`= '$f_mailid'");
    return (mysql_result($query, 0) == 1) ? true : false;
}

function f_regid_exists($f_regid) {
    $f_regid = sanitize($f_regid);
    $query = mysql_query("SELECT COUNT(`f_id`) FROM
`flight_users` WHERE `f_regid`= '$f_regid'");
    return (mysql_result($query, 0) == 1) ? true : false;
}

function f_active($f_username) {
    $f_username = sanitize($f_username);
    $query = mysql_query("SELECT COUNT(`f_id`) FROM
`flight_users` WHERE `f_username`= '$f_username' AND `f_active` = 1");
    return (mysql_result($query, 0) == 1) ? true : false;
}

function f_id_from_username($f_username) {
    $f_username = sanitize($f_username);
    $query = mysql_query("SELECT `f_id` FROM `flight_users`
WHERE `f_username` = '$f_username'");
    return mysql_result($query, 0, 'f_id');
}

function f_id_from_email($f_mailid) {
    $f_mailid = sanitize($f_mailid);
    $query = mysql_query("SELECT `f_id` FROM `flight_users`
WHERE `f_mailid` = '$f_mailid'");
    return mysql_result($query, 0, 'f_id');
}

function f_login($f_username, $f_password) {
    $f_id = f_id_from_username($f_username);
    $f_username = sanitize($f_username);
    $f_password = md5($f_password);
    $query = mysql_query("SELECT COUNT(`f_id`) FROM
`flight_users` WHERE `f_username`= '$f_username' AND `f_password` =
'$f_password'");
    return (mysql_result($query, 0) == 1) ? $f_id : false;
}

?>

```

Κώδικας με συναρτήσεις που πραγματοποιούν τις εξής λειτουργίες οι οποίες εκτελούνται στην εφαρμογή:

- Ανάκτηση username ή password στην περίπτωση που ζητηθεί από το χρήστη μέσω αποστολής email
- Ενεργοποίηση λογαριασμού μέσω αποστολής url verification σε email
- Αποθήκευση αλλαγών σε περίπτωση επεξεργασίας προσωπικών δεδομένων
- Αποθήκευση προσωπικών δεδομένων χρήστη σε μεταβλητή



- Έλεγχος για το αν ο χρήστης είναι συνδεδεμένος στον προσωπικό του λογαριασμό
- Έλεγχος για το αν το όνομα του χρήστη υπάρχει στη βάση δεδομένων
- Έλεγχος για το αν το email του χρήστη υπάρχει στη βάση δεδομένων
- Έλεγχος για το αν λογαριασμός έχει ενεργοποιηθεί
- Έλεγχος username και password για τη σύνδεση του χρήστη στο λογαριασμό του.

### activate.php

```

<?php
    $title = 'Our Airlines | Account Activation';
    include $_SERVER["DOCUMENT_ROOT"].'/Airlines/core/init.php';
    f_logged_in_redirect();
    include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/header.php
';

    if(isset($_GET['success']) === true &&
empty($_GET['success']) === true) {
?>
    <h4>You have successfully activated your account!</h4>
<?php
    }
    else if(isset($_GET['f_mailid'], $_GET['f_mailcode']) ===
true) {
        $f_mailid = trim($_GET['f_mailid']);
        $f_mailcode = trim($_GET['f_mailcode']);
        if(f_email_exists($f_mailid) === false) {
            $errors[] = 'Sorry, we couldn\'t find that Email
Address';
        }
        else if (f_activate($f_mailid, $f_mailcode) === false){
            $errors[] = 'Sorry, there was some problem activating
that account';
        }
        if(empty($errors) === false) {
?>
            <h4>Oops...</h4>
<?php

            echo output_errors($errors);
        }
        else {
            header('Location:
http://localhost/Airlines/activate.php?success');
            exit();
        }
    }
    else {
        header('Location: index.php');
        exit();
    }
}

```

```

include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/footer.php
';
?>

```

Κώδικας που πραγματοποιεί την ενεργοποίηση του λογαριασμού του χρήστη τη στιγμή που θα χρησιμοποιήσει το verification URL που έχει σταλεί στο email του.

### changeass.php

```

<?php
    $title = 'Our Airlines | Change Password';
    include $_SERVER["DOCUMENT_ROOT"].'/Airlines/core/init.php';
    f protect page();
    if(f_logged in()===false){
        header('Location:index.php');
    }
    if(empty($_POST)===false) {
        $required_fields
    array('current_password','password','password_again');
        foreach($_POST as $key=>$value) {
            if(empty($value) && in_array($key, $required_fields)
    === true ) {
                $errors[] = 'Fields marked with * are required!';
                break 1;
            }
        }

        if(md5($_POST['current_password']) ===
    $f data['f password']) {
            if(trim($_POST['password']) !==
    trim($_POST['password_again'])) {
                $errors[] = 'Your new passwords do not match';
            }
            else if(strlen($_POST['password']) < 6 ) {
                $errors[] = 'Your passwords must be atleast 6
characters!';
            }
        }
        else {
            $errors[] = 'Your current password is incorrect!';
        }
    }

    include
    $_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/header.php
    ';
    ?>

    <h3>Change Password</h3>

<?php

if(isset($_GET['success']) === true && empty($_GET['success'])
    === true ) {

```

```

    echo 'Your password has been changed successfully!';
}

else {

    if(isset($_GET['force']) === true && empty($_GET['force'])
    === true ) {
        ?>
        <p>You must change your password to continue using the
        site!</p>
    <?php
    }

    if(empty($_POST)===false && empty($errors)===true) {
        f change password($session_f_id, $_POST['password']);
        header('Location:
        http://localhost/Airlines/changepass.php?success');
    }
    else if(empty($errors) === false) {
        echo output_errors($errors);
    }
}
?>

    <form action="" method="POST">
        Current Password: <br/>
        <input type="password" name="current_password"
    /><br/><br/>
        New Password: <br/>
        <input type="password" name="password"
    /><br/><br/>
        New Password Again: <br/>
        <input type="password" name="password_again"
    /><br/><br/>
        <input type="submit" class="btn btn"
    value="Update" /><br/><br/>
    </form>

    <?php
    }
    include
    $_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/footer.php
    ';
    ?>

```

Κώδικας όπου γίνεται η αλλαγή του κωδικού πρόσβασης του λογαριασμού.

### settings.php

```

<?php
    $title = 'Our Airlines | Settings';
    include $_SERVER["DOCUMENT_ROOT"].'/Airlines/core/init.php';
    f protect_page();
    include
    $_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/header.php
    ';

```

```

if(f_logged_in()===false){
    header('Location:index.php');
}
if(empty($_POST)===false) {
    $required_fields =
array('f_fname','f_lname','f_mailid');
    foreach($_POST as $key=>$value) {
        if(empty($value) && in_array($key, $required_fields)
=== true ) {
            $errors[] = 'All the fields are required';
            break 1;
        }
    }

    if(empty($errors) === true ) {
        if(filter_var($_POST['f_mailid'],
FILTER_VALIDATE_EMAIL) === false) {
            $errors[] = 'Please enter a valid email address';
        }
        if(f_email_exists($_POST['f_mailid']) === true &&
$_POST['f_mailid'] !== $_POST['f_mailid']){
            $errors[] = 'Sorry, the email is already in use';
        }
        if(!preg_match('/^[a-z]{2,30}$/i',
$_POST['f_fname'])) {
            $errors[] = 'Your first name can contain only
alphabets';
        }
        if(!preg_match('/^[a-z]{2,30}$/i',
$_POST['f_lname'])) {
            $errors[] = 'Your last name can contain only
alphabets';
        }
    }
}
?>

<h3>Settings</h3>

<?php

    if(isset($_GET['success']) ===true &&
empty($_GET['success'])===true) {
        echo 'Updated!';
    }
    else
    {
        if(empty($_POST) ===false && empty($errors)===true) {
            $update_data = array(
                'f_fname' => $_POST['f_fname'],
                'f_lname' => $_POST['f_lname'],
                'f_mailid' => $_POST['f_mailid'],
            );
            update_f($session_f_id, $update_data);
            header('Location: settings.php?success');
            exit();
        }
        else if(empty($errors) === false ) {
            echo output_errors($errors);
        }
    }
}

```

```

?>
<form action="" method="POST">
  <br/>
  First Name: <br/>
  <input type="text" name="f_fname" value="<?php echo
$f_data['f_fname']; ?>"/><br/><br/>
  Last Name: <br/>
  <input type="text" name="f_lname" value="<?php echo
$f_data['f_lname']; ?>"/><br/><br/>
  Email ID: <br/>
  <input type="text" name="f_mailid" value="<?php echo
$f_data['f_mailid']; ?>"/><br/><br/>
  <input type="submit" class="btn btn" value="Update"
/><br/><br/>
</form>

<?php
}

  include
$_SERVER["DOCUMENT_ROOT"].'/Airlines/includes/overall/footer.php
';

?>

```

Κώδικας που εμφανίζει ορισμένες από τις πληροφορίες του χρήστη τις οποίες και μπορεί να επεξεργασθεί αποθηκεύοντας οποιαδήποτε αλλαγή στη βάση δεδομένων στη βάση της MySQL.

## Παραπομπές

- Bulkar, M. (2018). *Linkedin*. Ανάκτηση από <https://www.linkedin.com/pulse/difference-between-static-dynamic-websites-mangesh-bulkar>
- Computerhope*. (2018). Ανάκτηση από <https://www.computerhope.com/jargon/w/webpage.htm>
- Duckett, J. (2018). Στο J. Duckett, *HTML & CSS design and build websites*.
- Freeservers*. (2018). Ανάκτηση από <http://www.freeservers.com/WebHosting101/WhatIsAWebsite.html>
- GCF Learnfree*. (2018). Ανάκτηση από <https://www.gcflearnfree.org/internetbasics/what-is-the-internet/1/>
- Haverbeke, M. (2018). Στο M. Haverbeke, *Eloquent Javascript*.
- MDN web docs*. (2018). Ανάκτηση από [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server)
- Mitchell, B. (2018). *Lifewire*. Ανάκτηση από <https://www.lifewire.com/definition-of-apache-816509>
- Notepad ++*. (2018). Ανάκτηση από [https://notepad-plus-plus.org/Orgera, S. \(2018\).](https://notepad-plus-plus.org/Orgera, S. (2018).)
- Lifewire*. Ανάκτηση από <https://www.lifewire.com/what-is-a-browser-446234>
- PHP.net*. (2018). Ανάκτηση από <http://php.net/manual/en/intro-what-is.php>
- phpMyAdmin*. (2018). *phpMyAdmin*. Ανάκτηση από <https://www.phpmyadmin.net>
- Portableapps*. (2018). Ανάκτηση από <https://portableapps.com/apps/development/xampp>
- Programming Notes*. (2018). Ανάκτηση από [https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP\\_Basics.html](https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)
- Shannon, R. (2018). *HTML Source: HTML Tutorials*. Ανάκτηση από <http://www.yourhtmlsource.com/starthere/whatishtml.html>
- Simple Mail Transfer Protocol*. (2018). Ανάκτηση από [https://en.wikipedia.org/wiki/Simple\\_Mail\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol)
- Techopedia*. (2018). Ανάκτηση από <https://www.techopedia.com/definition/5217/world-wide-web-www>
- Techopedia*. (2018). Ανάκτηση από <https://www.techopedia.com/definition/2542/web-host>
- Test Mail Server Tool*. (2018). Ανάκτηση από <https://toolheap.com/test-mail-server-tool/>
- W3Schools*. (2018). Ανάκτηση από [https://www.w3schools.com/tags/ref\\_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp)
- W3Schools*. (2018). *PHP Sessions*. Ανάκτηση από [https://www.w3schools.com/php/php\\_sessions.asp](https://www.w3schools.com/php/php_sessions.asp)

*Wikipedia*. (2018). Ανάκτηση από <https://en.wikipedia.org/wiki/MySQL>

*Wikipedia*. (2018). Ανάκτηση από [https://en.wikipedia.org/wiki/Merchant\\_account](https://en.wikipedia.org/wiki/Merchant_account)

*Wikipedia*. (2018). Ανάκτηση από [https://en.wikipedia.org/wiki/Payment\\_gateway](https://en.wikipedia.org/wiki/Payment_gateway)

*Wikipedia*. (2018). Ανάκτηση από [https://en.wikipedia.org/wiki/Payment\\_terminal](https://en.wikipedia.org/wiki/Payment_terminal)

## Βιβλιογραφία

Επίσημη ιστοσελίδα Wikipedia: <https://en.wikipedia.org>

Tutorials for web technologies: <https://www.w3schools.com>

PHP Online Manual: <http://www.php.net>

Επίσημη ιστοσελίδα phpMyAdmin: <https://www.phpmyadmin.net>

Επίσημη ιστοσελίδα test mail server tool: <https://toolheap.com/test-mail-server-tool>

Επίσημη ιστοσελίδα Notepad++ : <https://notepad-plus-plus.org/>

Marijn Haverbeke, Eloquent Javascript, 2011.

Jon Duckett., HTML & CSS design and build websites, 2011.

Για την υλοποίηση της πτυχιακής επεξεργάσθηκαν κομμάτια κώδικα από τις εφαρμογές:

- 1) Online Airline Booking System: [https://osdn.net/projects/sfnet\\_oabs/](https://osdn.net/projects/sfnet_oabs/)
- 2) Online Flight Reservation System: <http://nevonprojects.com/online-flight-reservation-system/>
- 3) Swift Airline Reservation System: <https://github.com/ansin218/swift-airline-reservation-system>
- 4) Airlines Ticket Booking System: <https://www.freeprojectz.com/paid-projects/airlines-ticket-booking-system>
- 5) Flight Reservation System: <https://www.freeprojectz.com/php-projects-projects/flight-reservation-system>

ο οποίος αφορά τετριμμένες υλοποιήσεις ενός site και κατόπιν αναπτύχθηκε το κύριο μέρος της πτυχιακής ώστε να ανταποκρίνεται το site στις απαιτήσεις και προδιαγραφές που ανατέθηκαν εξαρχής.