

Εργασία στο μάθημα Προγραμματισμός Διαδικτύου 2024/2025

Δημιουργία εφαρμογής αναζήτησης αεροπορικών εισιτηρίων

Μαρία - Νίκη Ζωγράφου

Προπτυχιακή φοιτήτρια τμήματος ΗΜΤΥ, Πανεπιστημίου Πατρών, up1096060@upnet.gr

Νικόλαος Γέροντας

Προπτυχιακός φοιτητής τμήματος ΗΜΤΥ, Πανεπιστημίου Πατρών, up1092813@upnet.gr

Σε αυτήν την εργασία περιγράφουμε τη σχεδίαση και υλοποίηση μιας διαδικτυακής εφαρμογής αναζήτησης εισιτηρίων, αξιοποιώντας την αρχιτεκτονική MVC, στα πλαίσια της διδασκαλίας του μαθήματος «Προγραμματισμός Διαδικτύου» για το ακαδημαϊκό έτος 2024-2025.

1 ΠΕΡΙΛΗΨΗ

Σε αυτή την εργασία περιγράφουμε τη σχεδίαση και την υλοποίηση μιας διαδικτυακής εφαρμογής με τίτλο FlyExpress, η οποία επιτρέπει την αναζήτηση αεροπορικών εισιτηρίων με διαδραστικό και φιλικό προς τον χρήστη τρόπο. Η εφαρμογή συνδυάζει μοντέρνο frontend ενώ παράλληλα προσφέρει δυνατότητες φιλτραρίσματος, εύρεσης τιμών ανά ημέρα, και αποθήκευσης αγαπημένων δρομολογίων. Ιδιαίτερη έμφαση δόθηκε στην εμπειρία χρήστη, τόσο σε desktop όσο και σε κινητές συσκευές, καθώς και στην αποδοτική παρουσίαση σύνθετων πληροφοριών όπως διάρκεια, τιμή και ημερολόγιο τιμών. Η εφαρμογή δημιουργήθηκε στο πλαίσιο του μαθήματος «Προγραμματισμός Διαδικτύου» του ακαδημαϊκού έτους 2024–2025 και αξιοποιεί σύγχρονες τεχνολογίες όπως Express.js, SQLite, Handlebars, και globe.gl για 3D γραφικά.

2 ΜΕΘΟΔΟΛΟΓΙΑ

Παρακάτω ανατίθενται τα βήματα που ακολουθήσαμε για την υλοποίηση της εφαρμογής μας:

2.1 ERD:

Μετά την ανάλυση του μικρόκοσμου ενός εκδοτικού οίκου, προχωρήσαμε στην κατάλληλη σχεδίαση της μορφής της βάσης δεδομένων μας. Η σχεδίαση αυτή ήταν εφικτή μέσω του web εργαλείου [ERD Maker](#) του τμήματός μας:

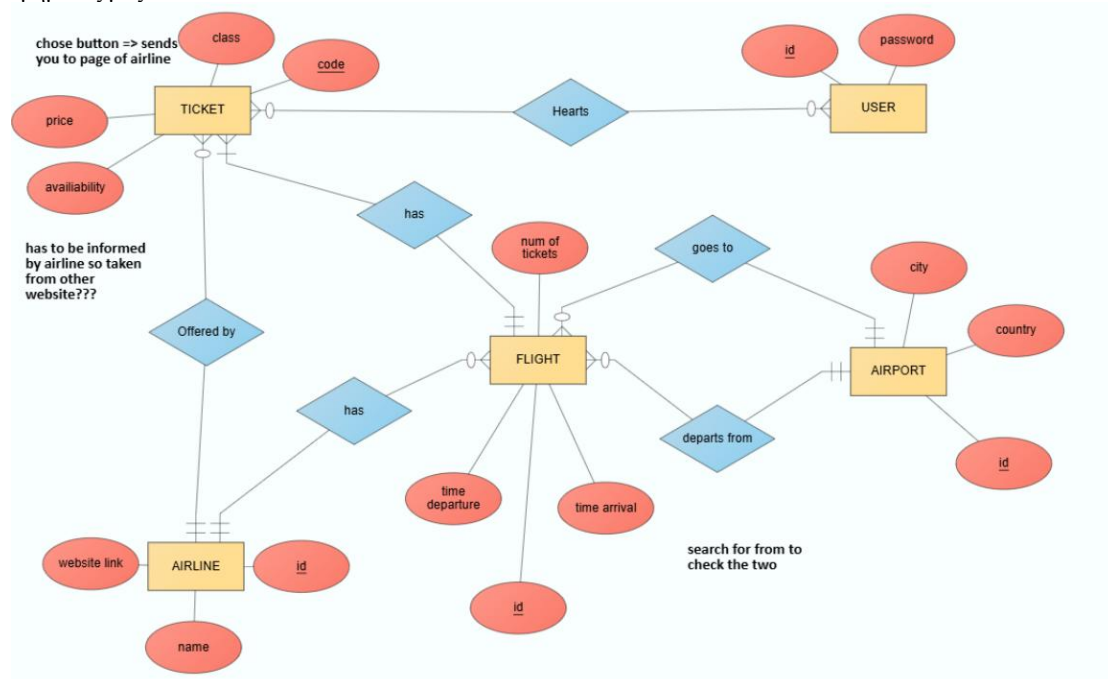


Figure 1: ERD της βάσης δεδομένων μας.

2.2 Schema:

Στη συνέχεια, βάση του παραπάνω ERD καταλήξαμε στο σχεσιακό μοντέλο που ακολουθεί. Η σχεδίαση του ήταν εφικτή μέσω του web εργαλείου [Schema Builder](#) του τμήματός μας:

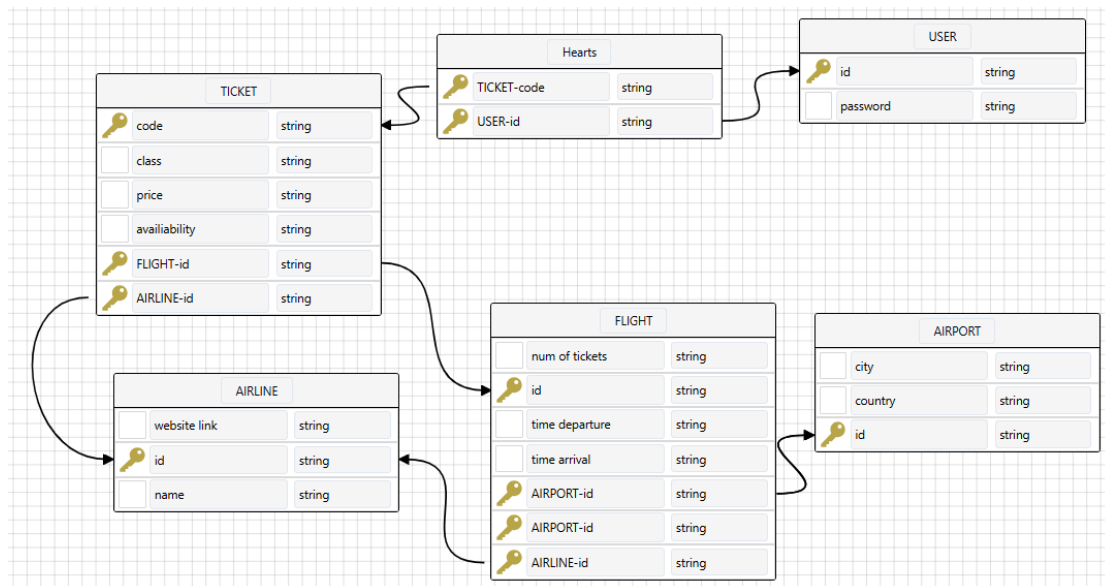


Figure 2: Schema της βάσης δεδομένων μας.

2.3 Σχεδιασμός Όψης της Εφαρμογής

Στο στάδιο της σχεδίασης του περιβάλλοντος χρήστη, δημιουργήσαμε ένα αρχικό πρωτότυπο της ιστοσελίδας χρησιμοποιώντας HTML, CSS και JavaScript. Η εμφάνιση και η λειτουργικότητα της πλατφόρμας σχεδιάστηκαν εμπνευσμένες από σύγχρονες ιστοσελίδες κρατήσεων, όπως το Google Flights/Sky express, με στόχο τη διασφάλιση οικειότητας και ευχρηστίας για τον τελικό χρήστη.

2.4 Δημιουργία Βάσης & Data Generation:

Αποφασίσαμε να χρησιμοποιήσουμε την SQLite για την βάση μας, καθώς αποτελεί βασική βιβλιοθήκη της Python. Έτσι, μετατρέψαμε το παραπάνω Schema σε εντολές δημιουργίας της βάσης δεδομένων μας.

2.5 Οργάνωση Εφαρμογής – Ανάπτυξη Δρομολογητή – Αρχιτεκτονική MVC:

Η εφαρμογή σχεδιάστηκε σύμφωνα με την αρχιτεκτονική MVC (Model–View–Controller), με σκοπό τη σαφή διαχωριστικότητα των ρόλων του συστήματος. Η δομή των αρχείων περιλαμβάνει:

Model: Περιλαμβάνει τον χειρισμό της βάσης δεδομένων. Περιέχει συναρτήσεις για την ανάκτηση, εισαγωγή και διαγραφή δεδομένων μέσω SQL queries, χρησιμοποιώντας τη βιβλιοθήκη better-sqlite3. Οι συναρτήσεις αυτές συγκεντρώνονται στο αρχείο model-betterSqlite3.mjs.

View: Το επίπεδο παρουσίασης υλοποιείται με χρήση του template engine Handlebars. Τα .hbs αρχεία περιέχουν το δυναμικό περιεχόμενο της εφαρμογής, το οποίο εμπλουτίζεται με δεδομένα που αποστέλλονται από τον controller μέσω της μεθόδου res.render.

Controller: Αποτελεί τον ενδιάμεσο κρίκο μεταξύ μοντέλου και προβολής. Περιέχει τη λογική ελέγχου της εφαρμογής (στο αρχείο controller.mjs), δρομολογεί τα αιτήματα χρηστών μέσω του Express Router, ελέγχει sessions και αυθεντικοποίηση, διαχειρίζεται την επικοινωνία με το μοντέλο και προωθεί τις κατάλληλες απόψεις (views) για απόδοση στο frontend.

3 ΑΞΙΟΛΟΓΗΣΗ

Η αξιολόγηση της εφαρμογής πραγματοποιήθηκε βάσει προκαθορισμένων κριτηρίων επιτυχίας, που εστιάζουν στην φιλικότητα προς τον χρήστη, την αισθητική παρουσίαση και τη λειτουργική απόδοση του συστήματος. Ασφάλεια Δεδομένων:

3.1 Ευκολία χρήσης:

Η εφαρμογή σχεδιάστηκε με στόχο τη μέγιστη απλότητα στη χρήση και την καθοδηγούμενη εμπειρία για τον χρήστη. Ενδεικτικά:

1. Διαδραστικό ημερολόγιο με τιμές: Ο χρήστης μπορεί να επιλέξει ημερομηνίες με προβολή ενδεικτικών τιμών ανά ημέρα, σε μορφή hoverable calendar, όπως στο Google Flights.
2. Widget Date Grid: Δυναμικό πλέγμα που εμφανίζει με μια ματιά τις φθηνότερες επιλογές για συχνές πτήσεις βάσει εξερχόμενης και επιστροφής ημερομηνίας.
3. Αποφυγή υπερφόρτωσης πληροφορίας: Εμφάνιση ενδεικτικών αποτελεσμάτων και επιλογή show more flights, ώστε να μην κατακλύζεται ο χρήστης από άχρηστα αποτελέσματα.
4. Αποδοτική Αναζήτηση: Δίνεται η επιλογή χρήσης φίλτρων (π.χ. μέγιστη τιμή, διάρκεια) ώστε ο χρήστης να βρίσκει γρηγορότερα αυτό που ψάχνει.
5. Σύστημα ταξινόμησης: Ο χρήστης μπορεί να ταξινομήσει τις πτήσεις κατά τιμή ή διάρκεια (αύξουσα/φθίνουσα), ενισχύοντας την αποτελεσματικότητα της αναζήτησης.
6. Σύστημα σύνδεσης/εγγραφής: Καθαρή και απλή διεπαφή με οπτική ανατροφοδότηση σε αποτυχία ή επιτυχία, καθώς και αυτόματη αποστολή email καλωσορίσματος μετά την εγγραφή.
7. Σελίδα αγαπημένων: Ο χρήστης μπορεί να προσθέτει ή να αφαιρεί αγαπημένες πτήσεις με ένα μόνο κλικ

3.2 Λειτουργία για κινητά:

Με χρήση των developer tools επιβεβαιώσαμε ότι η εφαρμογή, χάρη στη css και σε δυναμικές αναθέσεις μεγεθών, είναι λειτουργική και ευπαρουσίαστη και σε κινητά τηλέφωνα.

3.3 Αξιολόγηση με Google LightHouse:

Το Google Lighthouse είναι ένα αυτόματο εργαλείο open source που επιτρέπει την αξιολόγηση της ποιότητας μιας ιστοσελίδας με βάση πέντε βασικούς άξονες: Απόδοση, Προσβασιμότητα, Best Practices και Search Engine Optimization.

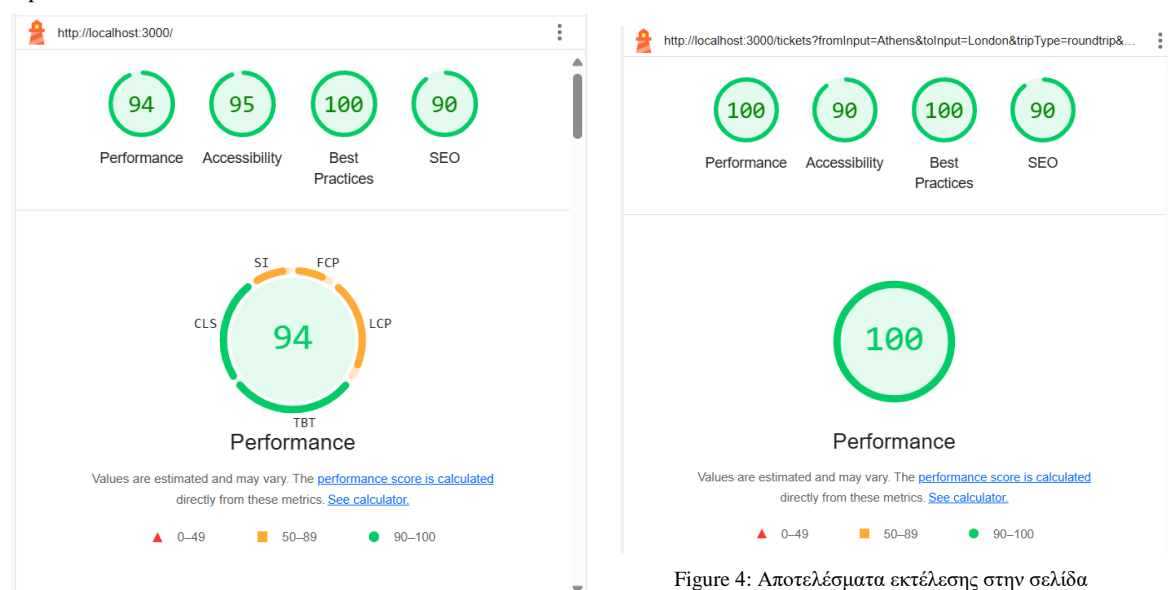


Figure 4: Αποτελέσματα εκτέλεσης στην σελίδα αναζήτησης

Figure 3: Αποτελέσματα εκτέλεσης στην αρχική σελίδα

4 ΔΕΔΟΜΕΝΑ

Για την αξιολόγηση της εφαρμογής ήταν απαραίτητο να δημιουργηθεί ένα ρεαλιστικό αλλά ελαφρύ dataset, το οποίο να προσομοιώνει συχνές και σπάνιες πτήσεις χωρίς να επιβαρύνει σημαντικά τη μνήμη του συστήματος. Η παραγωγή των δεδομένων δεν βασίστηκε σε εξωτερικές πηγές αλλά σε έξυπνη γεννήτρια τυχαίων πτήσεων, που αναπτύξαμε με στόχο την ισορροπία μεταξύ ρεαλισμού και αποδοτικότητας. Συγκεκριμένα:

Χρησιμοποιήθηκε σταθμισμένη δειγματοληψία ώστε να εμφανίζονται δημοφιλείς προορισμοί (π.χ. Αθήνα–Λονδίνο, Νέα Υόρκη –Τόκιο) με μεγαλύτερη συχνότητα, ενώ αραιές πτήσεις να παράγονται πιο σπάνια (όπως Κάριο-Μπανγκοκ).

Παράχθηκε επίσης πλήθος χρηστών με αγαπημένα, ώστε να μπορέσουμε να δείξουμε στατιστικά των **Top Destinations**, ως προτάσεις στους χρήστες.

5 ΔΙΑΜΟΙΡΑΣΜΟΣ:

Η εργασία αποτελείται από πολλά μέρη, απαιτώντας στενή συνεργασία για τη δημιουργία και τη σύνδεση των επιμέρους τμημάτων της. Αν και χρειάστηκε στενή συνεργασία για την δημιουργία όλων των τμημάτων του project, ο καθένας επικεντρώθηκε περισσότερο σε συγκεκριμένα μέρη του έργου, με τον εξής διαμοιρασμό:

- ERD: από κοινού
- Schema: από κοινού
- Δημιουργία Βάσης- Παραγωγή Δεδομένων: Μαρία-Νίκη Ζωγράφου
- Controller module, Router : Από κοινού
- Δημιουργία Home Page: Από κοινού
- Calendar Date Picker: Νίκος Γέροντας
- Επιλογή Show More, Max price Filter: Μαρία-Νίκη Ζωγράφου
- Date Grid Widget: Μαρία-Νίκη Ζωγράφου
- Sorting, Max Duration filter: Νίκος Γέροντας
- Top Destinations: Νίκος Γέροντας
- About us page: Μαρία-Νίκη Ζωγράφου
- Login και Register: Νίκος Γέροντας
- Favourites: Από κοινού
- Deployment: Νίκος Γέροντας

6 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Όπως και προηγουμένως, οι προαναφερθέντες στόχοι θα αποτελέσουν τη βάση του χρονοδιαγράμματος που ακολουθεί:

- 2.1 ERD: τέλη Μαρτίου
- 2.2 Schema: τέλη Μαρτίου
- 2.3 Σχεδιασμός αρχικής όψης εφαρμογής: τέλη Μαρτίου
- 2.4 Αρχιτεκτονική MVC προσχέδιο: Τέλη Απριλίου
- 2.5 Δημιουργία Tickets Page με Max Price και Sorting Filter: Αρχές Μαΐου
- 2.6 Δημιουργία Show More και Sorting Tickets: Αρχές Μαΐου
- 2.7 Δημιουργία Βάσης & Data Generation: Αρχές Μαΐου
- 2.8 Favourites: Μέσα Μαΐου
- 2.9 Date Grid/Calendar Widgets: Μέσα Μαΐου
- 2.10 Authentication, Log in: Τέλη Μαΐου

7 ΟΔΗΓΙΕΣ

7.1 Οδηγίες Εγκατάστασης

Για την εκτέλεση της εφαρμογής απαιτείται η εγκατάσταση Node.js και, προαιρετικά, Python για δημιουργία νέας βάσης δεδομένων.

Απαιτήσεις:

Python (προαιρετικό): Εγκαταστήστε την Python από τον σύνδεσμο <https://www.python.org/downloads/>.

Git Bash (προαιρετικό): Για εύκολη λήψη του repository.

Node.js v22.15.0

Βήματα Εγκατάστασης:

Ακολουθήστε τον σύνδεσμο: https://github.com/Nick-744/web_dev_project

Κάντε κλικ στο Code (πράσινο κουμπί) → Download ZIP. (ή με χρήση git clone)

Αποσυμπίστε το αρχείο και ανοίξτε ένα τερματικό στο φάκελο.

(Προαιρετικό) Ρύθμιση αποστολής email: Για αποστολή email δημιουργείτε .env αρχείο με τις μεταβλητές WEBSITE_EMAIL και WEBSITE_EMAIL_PASSWORD και SESSION_SECRET.

(Προαιρετικό) Δημιουργία νέας βάσης δεδομένων: Χρησιμοποιήστε το final_base_generation.py

Εκτέλεση στο terminal:

```
python final_base_generation.py
```

Τέλος, εκτελέστε τις εξής εντολές στο τερματικό:

```
npm install
```

```
npm run watch
```

Η εφαρμογή θα εκκινήσει στη διεύθυνση: <http://localhost:3000>

7.2 Public Website

Η ιστοσελίδα βρίσκεται επίσης και σε δημόσιο URL, στο παρακάτω σύνδεσμο: <https://fly-express.onrender.com/>

8 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ

Παράδειγμα Home Page:

Εδώ ο χρήστης μπορεί να επιλέξει τον προορισμό που τον ενδιαφέρει. Φυσικά, οι τιμές του ημερολογίου ενημερώνονται δυναμικά βάση της κατάστασης επιλογής.

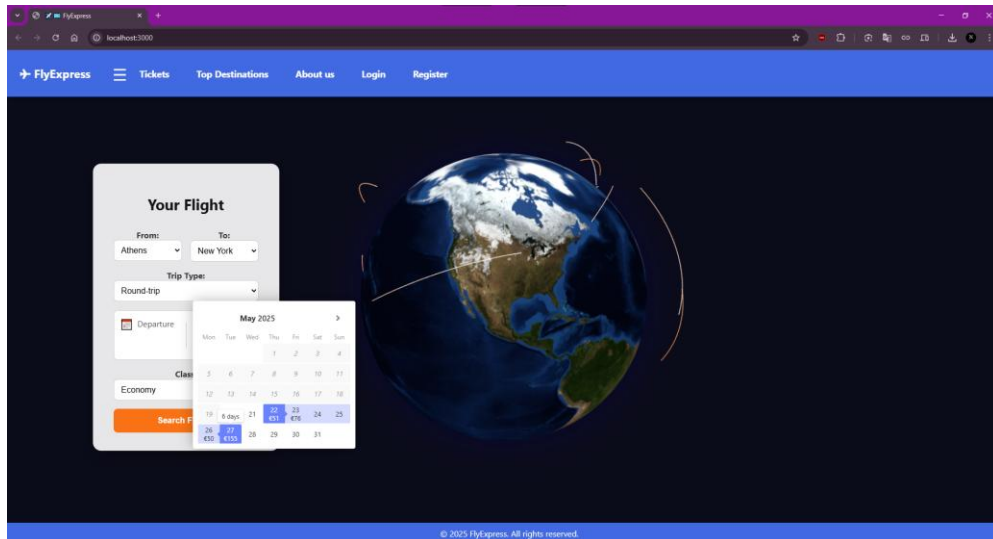


Figure 4: Παράδειγμα χρήσης αναζήτησης στην αρχική σελίδα

Παράδειγμα αποτελέσματος από την home page:

Όπως φαίνεται και στην παρακάτω εικόνα, η σελίδα tickets επέστρεψε τα αποτελέσματα της βάσης που ικανοποιούν τους κανόνες αναζήτησής μας:

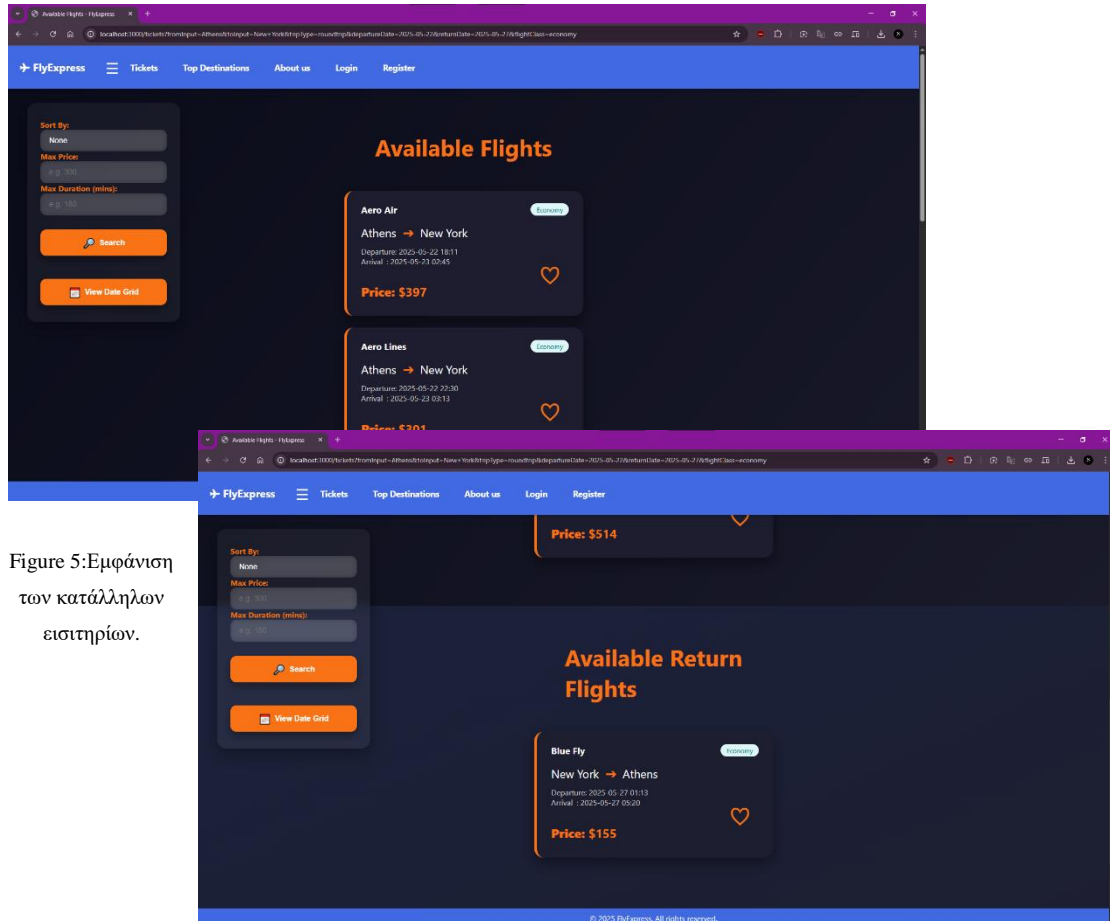


Figure 5: Εμφάνιση των κατάλληλων εισιτηρίων.

Παράδειγμα εφαρμογής φίλτρων:

Ο χρήστης, εφαρμόζοντας ως μέγιστη τιμή τα 200, η εφαρμογή κάνει το κατάλληλο SQL query στην βάση και γυρνάει τα νέα κατάλληλα φιλτραρισμένα αποτελέσματα. Δίνονται επίσης οι επιλογές διάταξης κατά αύξουσα και φθίνουσα σειρά, καθώς και διάρκειας πτήσης.

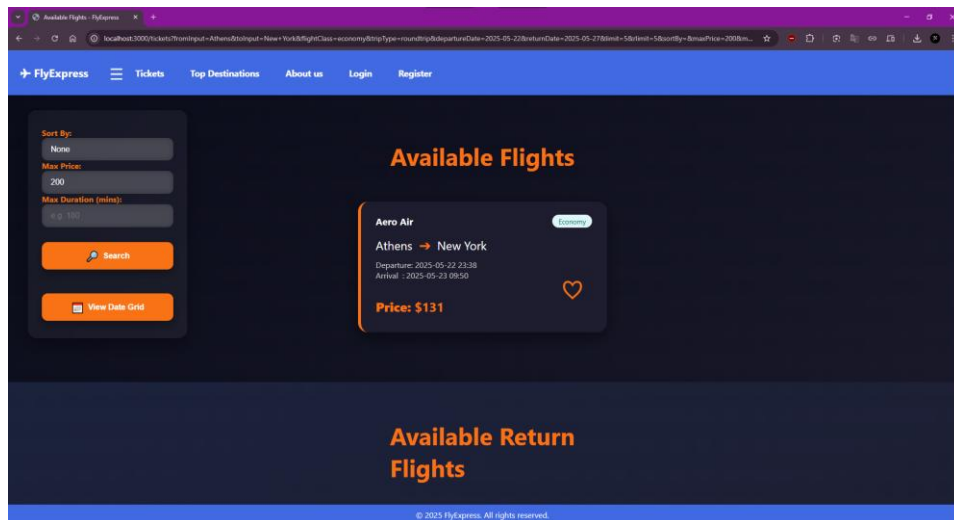


Figure 6: Το φίλτρο για την τιμή εισιτηρίου 200.

Παράδειγμα Date Grid:

Φυσικά, δεν θα μπορούσε να λείπει η λειτουργία του Date Grid:

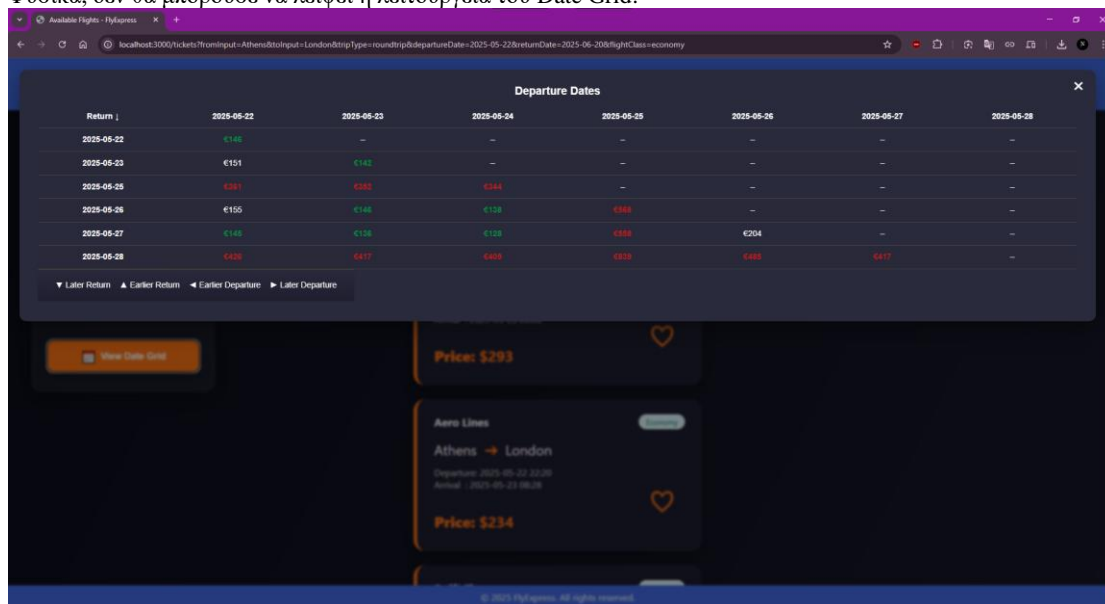


Figure 7: Η συνολική αξία του ταξιδιού.

Παράδειγμα Add in Favorites:

Φυσικά, πατώντας την καρδούλα στο εισιτήριο που θέλει να παρακολουθήσει ο χρήστης, αυτό προστίθεται στα αγαπημένα του, εφόσον έχει συνδεθεί:

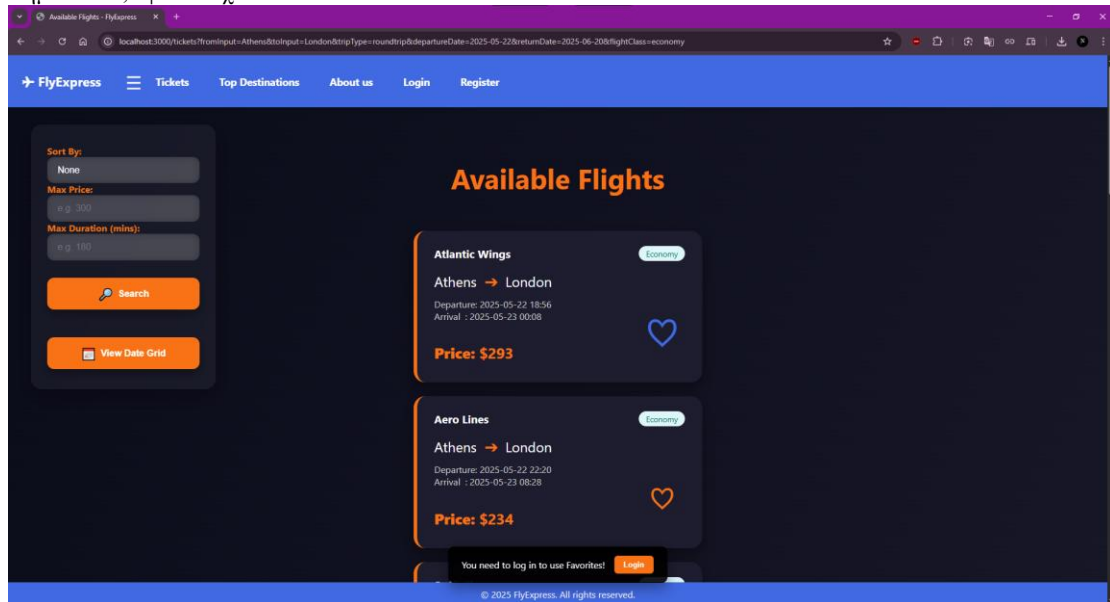


Figure 8: Prompt σύνδεσης.

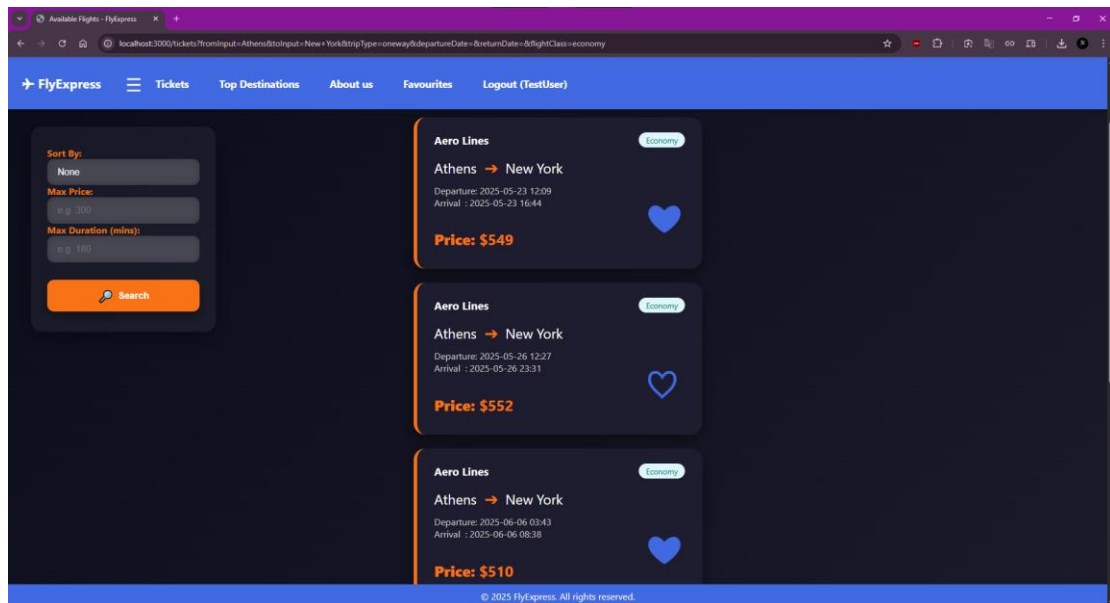


Figure 9: Η λειτουργία καρδούλα.

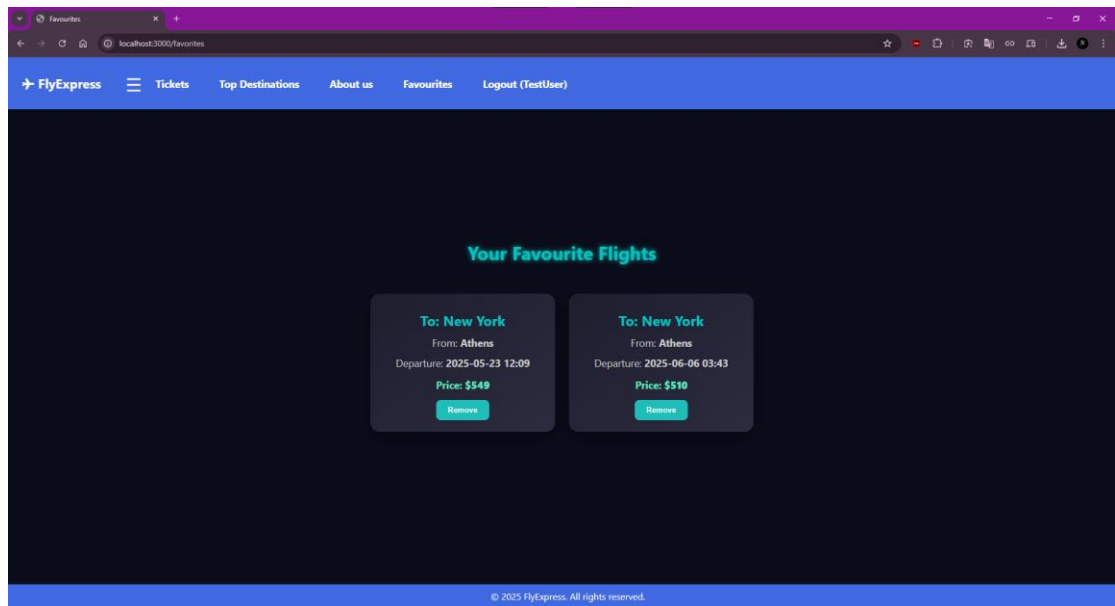


Figure 10: Η σελίδα αγαπημένα του χρήστη.

Παράδειγμα Top destinations:

Τα στατιστικά αποτελέσματα των πιο δημοφιλών προορισμών φαίνονται στην σελίδα Top destinations, η οποία ανανεώνεται δυναμικά βάση των like των χρηστών, ενώ οι εικόνες παρέχονται από το Wikipedia:

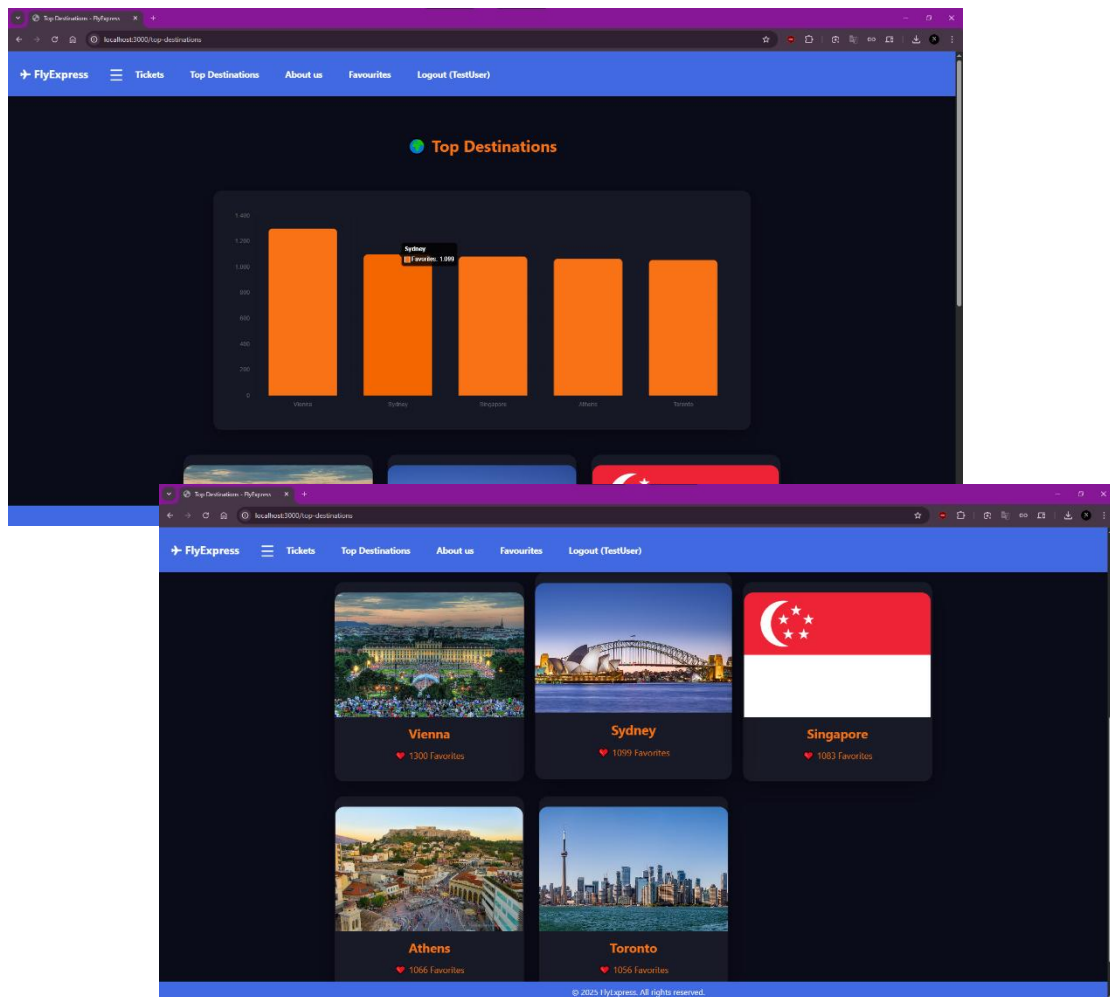


Figure 11: Οι top 5 liked προορισμοί από τους χρήστες μας.

Παράδειγμα Login/Register:

Το site, λόγω της δυνατότητας καρδούλα, δίνει στον χρήστη την επιλογή εγγραφής (με αποστολή mail καλωσορίσματος από την ομάδα) και σύνδεσης μέσω του λογαριασμού του:

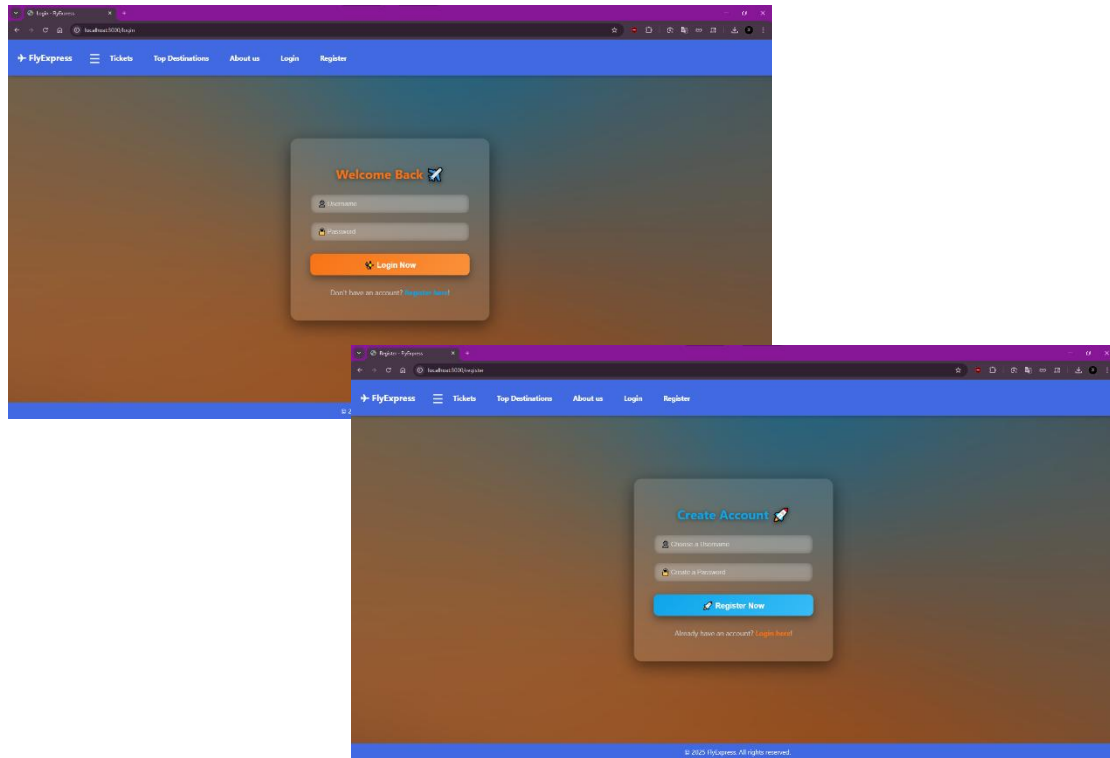


Figure 12: Σελίδες χρήστη.

Παράδειγμα χρήσης από κινητό:

Η σελίδα έχει δημιουργεί με τέτοιο τρόπο ώστε να λαμβάνει επίσης υπόψιν την χρήση από κινητό τηλέφωνο:

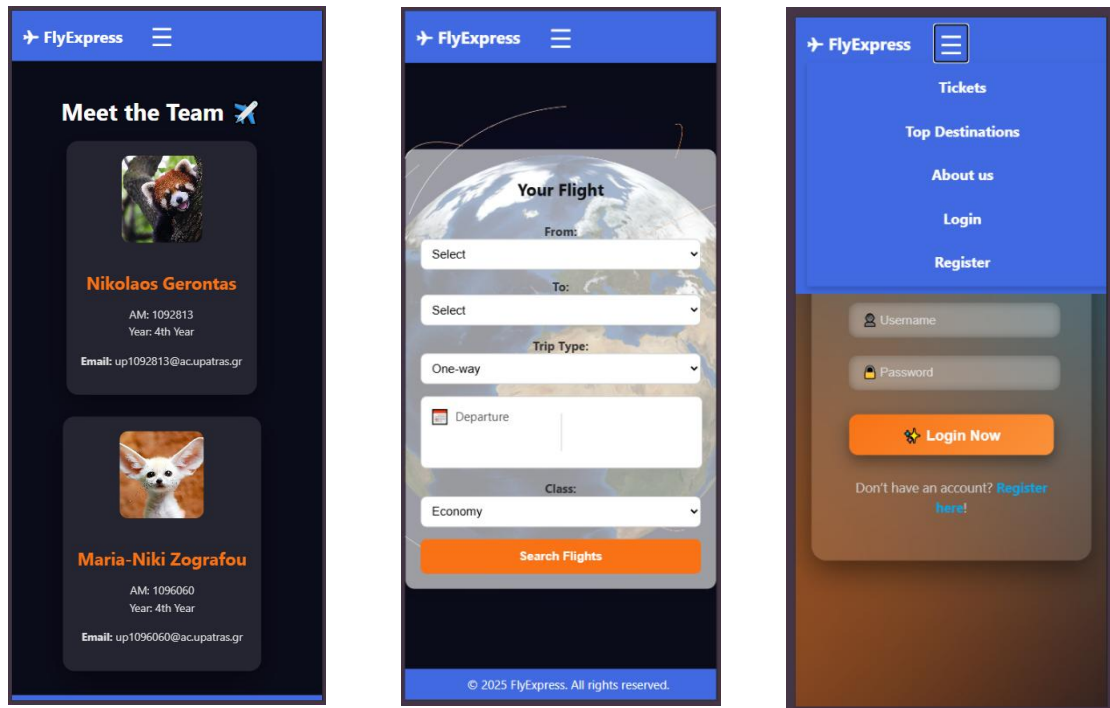


Figure 13: Προβολή σε οθόνη κινητού τηλεφώνου.