Εφαρμογή Τουριστικού Οδηγού μίας περιοχής

Πανεπιστήμιο Πατρών  
Πολυτεχνική Σχολή  
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών  
Πρότζεκτ στο μάθημα Προγραμματισμός Διαδικτύου (ECE\_ΓΚ802)  
Ομάδα 28

Ελευθέριος Γκλιάτης, 1066548

up1066548@ac.upatras.gr

Πάνος Λελάκης, 1083712

up1083712@ac.upatras.gr

Στην παρούσα έκθεση περιγράφεται ο τρόπος με τον οποίο σχεδιάστηκε η εφαρμογή για την υλοποίηση μίας διαδικτυακής εφαρμογής τουριστικού οδηγού μίας περιοχής. Πιο συγκεκριμένα, αναγράφονται τα Use Cases που καλύπτει η εφαρμογή .............. Συνδυάζοντας το θεωρητικό με το πρακτικό τμήμα της εργασίας, σχεδιάστηκε μία βάση δεδομένων SQLite με χρήση κώδικα SQL για την αποθήκευση των στοιχείων τόσο των άρθρων, όσο και των χρηστών της βάσης. Επιπλέον, αναπτύχθηκε πρόγραμμα σε Python που υλοποιεί μία ευρεία συλλογή από εργαλεία για την αλληλεπίδραση μεταξύ χρήστη και βάσης. Αναλύονται τα σημαντικά σημεία του κώδικα SQL στην δημιουργία της βάσης δεδομένων, στην καταχώρηση εγγραφών σε αυτή και στην εκτέλεση SQL queries που προσομοιώνουν την προσδοκώμενη χρήση της, καθώς και τα σημεία του κώδικα Python για τη λειτουργία του προγράμματος. Σε καθεμία από τις παραπάνω περιπτώσεις, δίνονται σε μορφή screenshot τα αποτελέσματα του κώδικα.

Βασικοί Όροι: • Web programming • Full stack development • Database • Javascript • Fly.io • PostgreSQL • SQLite-async • NodeJS • Express

**Επιπλέον λέξεις-κλειδιά:** Τουριστικός οδηγός, Ιστοσελίδα,

Ελευθέριος Γκλιάτης, Πάνος Λελάκης. 2024. Εφαρμογή Τουριστικού Οδηγού μίας περιοχής: Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πρότζεκτ στο μάθημα Προγραμματισμός Διαδικτύου (ECE\_ΓΚ802), Ομάδα 28. Μάιος 26/5, 2024. Πάτρα, Ελλάδα, ................ σελίδες.

1. Περιληψη

Το θέμα της εργασίας είναι η ανάπτυξη μίας διαδικτυακής εφαρμογής τουριστικού οδηγού για μία περιοχή. Σκοπός της ιστοσελίδας της εφαρμογής είναι η πληροφόρηση των χρηστών της για τις υποδομές της περιοχής, τα σημεία και μαγαζιά που ενδέχεται να τους ενδιαφέρουν, καθώς και η διαφήμιση αυτών. Επιπλέον, δίνεται η δυνατότητα και για προγραμματισμό ταξιδιού, μέσω των bookmarks, εφόσον ο χρήστης έχει φτιάξει λογαριασμό και έχει κάνει login.

Όπως ζητήθηκε από την εκφώνηση της εργασίας, όλα τα δεδομένα της εφαρμογής, δηλαδή οι πληροφορίες των τοποθεσιών, των χρηστών και τα bookmarks, είναι αποθηκευμένα σε μία βάση δεδομένων τύπου PostgreSQL. Μετά από την full-stack ανάπτυξη της ιστοσελίδας, έγινε δυνατόν το front-end να επικοινωνεί άρτια με το back-end και την βάση δεδομένων, αλλά και το αντίστροφο, ώστε οι ιστοσελίδες να κατασκευάζονται δυναμικά.

Η αυθεντικοποίηση των χρηστών, επιπλέον, κρίθηκε απαραίτητη, ώστε να τους δίνεται η δυνατότητα να αποθηκεύουν τις τοποθεσίες που τους ενδιαφέρουν στα προσωπικά τους bookmarks. Για την ασφαλή επικοινωνία μεταξύ χρηστών και βάσης δεδομένων, καθώς και την προστασία των στοιχείων των λογαριασμών τους, οι sensitive πληροφορίες κωδικοποιούνται.

Καθώς η εκφώνηση της εργασίας θέτει ελάχιστους περιορισμούς όσον αφορά τον σχεδιασμό του μικρόκοσμού της, για τον σκοπό της επιλέχθηκε να χρησιμοποιηθεί ως διαφημιζόμενος τόπος ένα φανταστικό νησί. Έγινε προσπάθεια ώστε να περιέχει ρεαλιστικό πλήθος δεδομένων στην βάση, να προσφέρει πραγματικά χρήσιμες λειτουργίες αλλά και να έχει ελκυστική εμφάνιση.

Το μεγαλύτερο ενδιαφέρον στην εργασία έχει η επιλογή και υλοποίηση των λειτουργιών της εφαρμογής, με βάση την οπτική γωνία του χρήστη. Αποφασίστηκε συνειδητά να δοθεί περισσότερη έμφαση στην πληροφόρηση των χρηστών της ιστοσελίδας, καθώς οι χρήστες της αναμένεται να είναι επισκέπτες του νησιού. Για αυτό, υλοποιήθηκε responsive χάρτης της περιοχής που περιέχει markers με όλες τις τοποθεσίες της. Λόγω του ότι πιθανότατα οι επισκέπτες θα αναζητήσουν αυτές τις πληροφορίες με το smartphone τους όσο θα βρίσκονται στο νησί, επιλέχθηκε να δοθεί περισσότερη βαρύτητα στον mobile-first σχεδιασμό της από τον desktop-based.

Τέλος, η εφαρμογή ανέβηκε στην πλατφόρμα fly.io ώστε να είναι δημόσια προσβάσιμη. Οι ενδιαφερόμενοι χρήστες μπορούν να επισκεφθούν την ιστοσελίδα μέσω του link [1].

1. Τροποσ επιλυσησ

Οι παρακάτω ενότητες περιγράφουν τη μεθοδολογία, την κατανομή των εργασιών και το χρονοδιάγραμμα που ακολουθήθηκε, όπως επίσης και τον τρόπο με τον οποίο αξιολογήθηκε και τον τύπο των δεδομένων που χρησιμοποιήθηκαν.

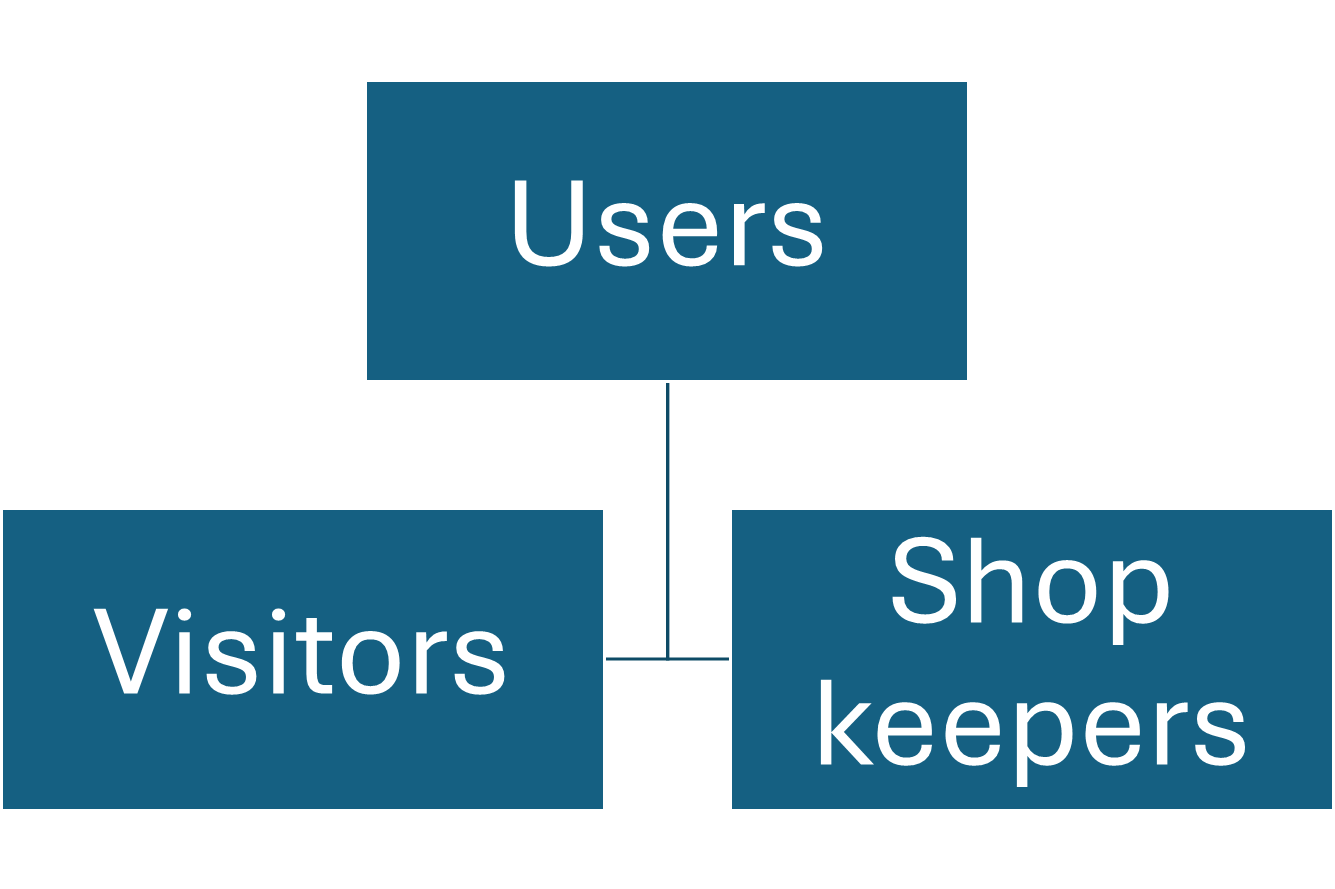
* 1. Μεθοδολογία

Για την επίλυση του προβλήματος, αρχικά έγινε αναζήτηση στο διαδίκτυο για ένα πρότυπο, πάνω στο οποίο θα μπορούσε κανείς να βασιστεί και να «φανταστεί» τι περιέχει μία ιστοσελίδα τουριστικού οδηγού. Για τον σκοπό αυτό, επιλέχθηκαν οι ιστοσελίδες <https://www.agistri-island.gr/> [2] και <https://www.visitgreece.gr/> [3], οι οποίες δείχνουν γραφικά τι λειτουργίες θα ήταν επιθυμητό να υποστηρίζονται.

Φυσικά, οι επαγγελματικές ιστοσελίδες τουριστικών οδηγών όπως το agistri-island και το visit-greece προσφέρουν μία ευρεία γκάμα λειτουργιών και εργαλείων, οπότε για την επιλογή των συγκεκριμένων λειτουργιών που υποστηρίζονται επιλέχθηκε η ανάπτυξη με βάση τις ανάγκες των χρηστών. Πιο συγκεκριμένα, οι χρήστες της εφαρμογής αναμένεται να είναι επισκέπτες του νησιού, οι οποίοι ψάχνουν για μέρη να επισκεφθούν, να γευματίσουν ή να διανυκτερεύσουν. Έτσι, η πρώτη σκέψη ήταν να δημιουργηθεί ένας χάρτης που θα περιέχει όλα τα ενδιαφέροντα μέρη του νησιού, με τα οποία θα μπορεί ο επισκέπτης να αλληλεπιδράσει. Μάλιστα, του δίνεται η δυνατότητα όχι μόνο να δει φωτογραφίες της τοποθεσίας και να διαβάσει πληροφορίες για αυτήν, αλλά και να την αποθηκεύσει στα bookmarks, εφόσον έχει φτιάξει λογαριασμό και έχει κάνει log-in.

Εφόσον καλύφθηκε η βασική ανάγκη της πλειοψηφίας των χρηστών, το επόμενο βήμα ήταν να αποφασιστούν ποιες θα είναι οι επιπρόσθετες λειτουργίες. Η πρώτη σκέψη ήταν η εφαρμογή να μπορεί να εξυπηρετήσει και τους μαγαζάτορες του νησιού. Έτσι, αποφασίστηκε να μπορεί να φτιάξει κανείς λογαριασμό μαγαζάτορα και να μπορεί να βλέπει πόσοι χρήστες έχουν κάνει bookmark το κάθε μαγαζί του. Ο κύριος λόγος που υλοποιήθηκε αυτό το εργαλείο είναι για να αναδειχθεί η δυνατότητα εύκολης προσθήκης επιπλέον λειτουργιών για την διαχείριση μαγαζιών.

Στην Εικ. 1 φαίνονται συνοπτικά όλες οι προαναφερόμενες λειτουργίες που υλοποιήθηκαν:



Εικ. 1: Διαγράμματα με τις κατηγορίες των χρηστών και τις λειτουργίες που υλοποιήθηκαν.

Μετά την διασαφήνιση των λειτουργιών που υποστηρίζονται, το επόμενο βήμα ήταν η ανάπτυξη των templates των απαραίτητων σελίδων, με συγγραφή κώδικα HTML και CSS, αφότου αυτές σχεδιάστηκαν γραφικά. Πιο συγκεκριμένα, οι σελίδες που κατασκευάστηκαν είναι οι εξής:

* Map (χάρτης με τοποθεσίες)
* Place info (φωτογραφίες και πληροφορίες για μία δεδομένη τοποθεσία)
* General info (γενικές πληροφορίες για το νησί)
* Contact (για την επικοινωνία με τους διαχειριστές της εφαρμογής)
* Bookmarks (όλα τα bookmarks ενός χρήστη)
* Owned places (όλα τα μαγαζιά ενός ιδιοκτήτη με τον αριθμό των bookmarks του καθενός)
* Login/Logout/Register (για την σύνδεση και αποσύνδεση των χρηστών, καθώς και για την δημιουργία νέου λογαριασμού)

Για την μόνιμη αποθήκευση όλων των παραπάνω πληροφοριών, τόσο για τους χρήστες όσο και για τις τοποθεσίες, σχεδιάστηκε μία βάση δεδομένων. Στην Εικ. 2 φαίνεται το ERD της:

A diagram of a diagram

Description automatically generated

Εικ. 2: Διάγραμμα ERD βάσης δεδομένων.

Έχοντας σχεδιάσει τον σκελετό της βάσης, κατασκευάστηκε το Schema, το οποίο φαίνεται στην Εικ. 3.

A diagram of a computer

Description automatically generated with medium confidence

Εικ. 3: Schema βάσης δεδομένων.

Ύστερα, το project αναδιαμορφώθηκε δομικά ώστε να ακολουθεί το μοντέλο MVC (Model Views Controller). Σύμφωνα με το μοντέλο αυτό, η εφαρμογή χωρίζεται σε 3 κύρια μέρη:

1. Το model, που είναι η βάση δεδομένων, ο κώδικας SQL που δημιουργεί τα απαραίτητα tables στην βάση και οι συναρτήσεις που αλληλεπιδρούν με αυτή μέσω SQL queries
2. Τα views, που είναι ο κώδικας HTML και τα templates των σελίδων
3. Ο controller, που καλεί τις συναρτήσεις του model, διαχειρίζεται την αυθεντικοποίηση των χρηστών και περνάει τα απαραίτητα δεδομένα στην κάθε σελίδα

Για λόγους ευκολίας, η εφαρμογή αρχικά είχε σχεδιαστεί ώστε να χρησιμοποιεί βάση δεδομένων τύπου SQLite-async. Αργότερα, έγιναν οι αναγκαίες αλλαγές στον κώδικα και στα αρχεία της εφαρμογής ώστε η βάση δεδομένων να είναι πλέον τύπου PostgreSQL, μιας και η PostgreSQL γίνεται ολοένα και πιο δημοφιλής σήμερα.

Το τελευταίο βήμα μετά την μετατροπή από SQLite-async σε PostgreSQL ήταν να ανεβεί η εφαρμογή σε δημόσια πλατφόρμα, ώστε να είναι προσβάσιμη από το κοινό. Για τον σκοπό αυτό, επιλέχθηκε η πλατφόρμα fly.io .

* 1. Αξιολόγηση

Αφότου ολοκληρώθηκε η ανάπτυξη του κώδικα της εφαρμογής, έγινε μία ανασκόπηση όλων των τμημάτων της, με σκοπό την αξιολόγησή της. Για την αξιολόγησή της, χρησιμοποιήθηκαν τα παρακάτω κριτήρια, ταξινομημένα με φθίνουσα σειρά προτεραιότητας (1 = υψηλότερη σημασία, 5 = χαμηλότερη σημασία):

1. Επίτευξη στόχου: ικανοποίηση μέσου χρήστη, κάλυψη αναγκών πληροφόρησης, υλοποίηση χρήσιμων εργαλείων και λειτουργιών, ελκυστική εμφάνιση ιστοσελίδας.

2. Χρηστικότητα: εύχρηστη εφαρμογή, με γρήγορη πρόσβαση στις πληροφορίες και στα εργαλεία, γρήγορη ανταπόκριση.

3. Bugs και κάλυψη ιδιαίτερων περιπτώσεων: η εφαρμογή δεν έχει bugs, κάθε περίπτωση σφάλματος διαχειρίζεται και ο χρήστης ενημερώνεται σχετικά.

4. Δομή και καθαρότητα κώδικα: τα αρχεία της εφαρμογής ακολουθούν την σωστή δομή και το μοντέλο MVC, ο κώδικας είναι καθαρογραμμένος με αρκετά σχόλια.

5. Χρήση βάσης δεδομένων PostgreSQL και ανέβασμα εφαρμογής σε δημόσια πλατφόρμα.

Η μεθοδολογία που επιλέχθηκε να ακολουθηθεί διαμορφώθηκε έτσι ώστε η εργασία να ικανοποιεί στο μέγιστο τα πρώτα 2 κριτήρια και όλα τα υπόλοιπα να τα πληροί έως έναν ικανοποιητικό βαθμό.

* 1. Κατανομή εργασιών

Η εργασία χωρίστηκε στα παρακάτω subtasks:

1. Κατανόηση προβλήματος και αναζήτηση για πηγές

2. Επιλογή λειτουργιών

3. Σχεδιασμός σελίδων

4. Σχεδιασμός βάσης δεδομένων

5. Ανακατασκευή σε μοντέλο MVC

6. Ανάπτυξη back-end

7. Χάρτης

8. Ανάπτυξη front-end

9. Αναπροσαρμογή σε βάση δεδομένων PostgreSQL

10. Ανέβασμα εφαρμογής στο fly.io

Και τα δύο μέλη της ομάδας πήραν μέρος σε όλα τα subtasks, ωστόσο ο καθένας ανέλαβε από ένα υποσύνολο αυτών ως «κύριες» ασχολίες του. Τα subtasks που ανέλαβε το κάθε μέλος είναι τα εξής:

Εξ’ ολοκλήρου από κοινού:

1) Κατανόηση προβλήματος και αναζήτηση για πηγές

2) Επιλογή λειτουργιών

5) Ανακατασκευή σε μοντέλο MVC

Ελευθέριος Γκλιάτης:

3) Σχεδιασμός σελίδων

7) Χάρτης

8) Ανάπτυξη front-end

Πάνος Λελάκης:

4) Σχεδιασμός βάσης δεδομένων

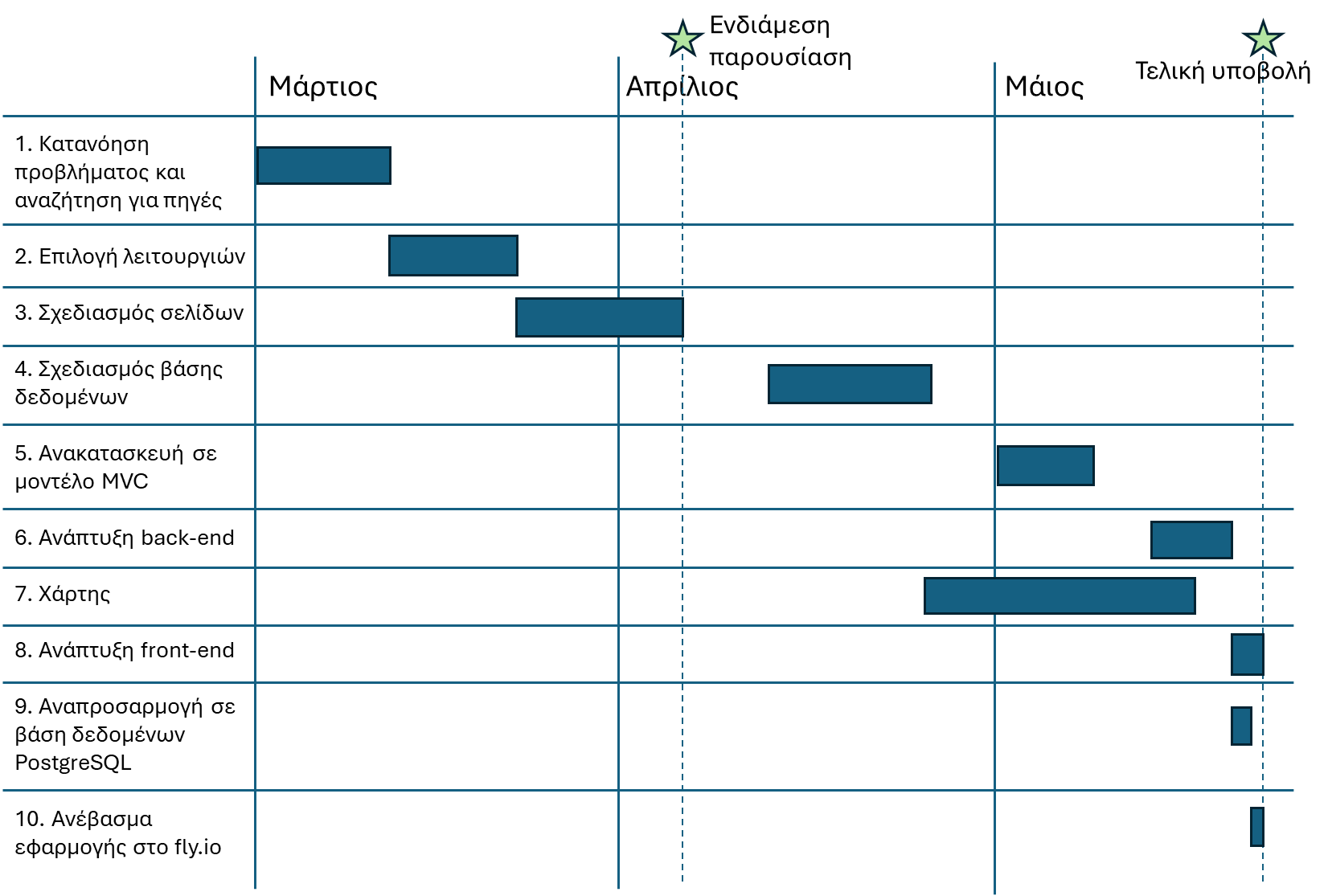
6) Ανάπτυξη back-end

9) Αναπροσαρμογή σε βάση δεδομένων PostgreSQL

10) Ανέβασμα εφαρμογής στο fly.io

* 1. Χρονοδιάγραμμα

Τα subtasks που αναφέρθηκαν στην ενότητα 2.3 μπορούν να τοποθετηθούν χρονικά ως εξής:



Εικ. 4: Χρονοδιάγραμμα

Λόγω του ότι οι απαραίτητες γνώσεις για την ανάπτυξη κώδικα και την διασύνδεση με την βάση δεδομένων καλύφθηκαν στις τελευταίες διαλέξεις του μαθήματος, η μεγαλύτερη πρόοδος του project έγινε εκείνη την περίοδο.

* 1. Δεδομένα

1. συνοψη

Η ανάπτυξη και υλοποίηση της Βάσης Δεδομένων για την αποθήκευση και ανάκτηση επιστημονικών άρθρων και περιοδικών αξιοποίησε τον συνδυασμό τεχνολογιών και εργαλείων ανάπτυξης: το ERD, το schema, τη βάση δεδομένων τύπου SQLite και την επικοινωνία αυτής με τη γλώσσα Python.

Παρόλο που είναι περιορισμένη σε εύρος, προσέφερε χρήσιμη και πολύτιμη εμπειρία με τις τεχνολογίες αυτές. Δεν δόθηκαν ακριβή ζητούμενα από την εκφώνηση της εργασίας, πράγμα που κατέστησε την αναζήτηση στο διαδίκτυο, για τον τρόπο με τον οποίο είναι δομημένες πραγματικές βάσεις που εκτελούν τις ίδιες λειτουργίες, απαραίτητη.

Τέλος, η βάση και το πρόγραμμα σε python σχεδιάστηκαν με απλές, μη-επαγγελματικές προδιαγραφές ως προς το εύρος των δυνατοτήτων που προσφέρουν, καθώς δόθηκε περισσότερη έμφαση στην σωστή λειτουργία τους. Φυσικά, τα όριά τους μπορούν να διευρυνθούν σημαντικά μετά από την ανάλυση των απαιτήσεων της εταιρείας που πρόκειται να τα χρησιμοποιήσει. Ορισμένα παραδείγματα επιπλέον λειτουργειών που θα μπορούσαν να προστεθούν είναι: η δυνατότητα για common (shared) libraries, public chat rooms, article recommendation engine, subscription/membership, εργαλεία για τους admins για την απευθείας τροποποίηση των στοιχείων ενός άρθρου κλπ.

ΠΗΓΕΣ

1. <https://santokriti.fly.dev/> (Link ιστοσελίδας εφαρμογής στην πλατφόρμα fly.io)