Εικόνα που περιέχει γραμματοσειρά, κείμενο, λογότυπο, γραφικά

Περιγραφή που δημιουργήθηκε αυτόματα, Picture

Όνομα : Θεοδοσόπουλος Παναγιώτης

Μάθημα : Mobile & Distributed Systems (CN6035)

ΑΜ : 2471039

GitHub : pantheo-cell

**Πίνακας Περιεχομένων**

1. Περιγραφή Εφαρμογής ....................................................... 3

2. Λειτουργίες Χρήστη ........................................................ 5

3. Τεχνολογίες και Αρχιτεκτονική ......................................... 6

4. Δομή Βάσης Δεδομένων .................................................. 8

5. Προκλήσεις και Επίλυση ................................................ 9

6. Screenshots Εφαρμογής .................................................. 10

**ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ**

Η εφαρμογή 'Restaurant App' δημιουργήθηκε στα πλαίσια του μαθήματος 'Mobile & Distributed Systems' με στόχο την κατανόηση και υλοποίηση ενός πλήρους κατανεμημένου συστήματος μέσω της ανάπτυξης εφαρμογής κινητού τηλεφώνου.

Ο βασικός σκοπός της εφαρμογής είναι να δώσει στον τελικό χρήστη τη δυνατότητα να κάνει εύκολα κράτηση τραπεζιού σε διαθέσιμα εστιατόρια, με γρήγορη πρόσβαση από κινητό τηλέφωνο. Παράλληλα, ο χρήστης μπορεί να διαχειριστεί το ιστορικό των κρατήσεών του, να επεξεργαστεί ή να ακυρώσει μελλοντικές κρατήσεις, καθώς και να αποσυνδεθεί όποτε το επιθυμεί.

Η ανάπτυξη της εφαρμογής χωρίζεται σε τρία διακριτά μέρη: 1) Την εφαρμογή κινητού (frontend), 2) τον εξυπηρετητή (backend REST API) και 3) τη βάση δεδομένων (MariaDB).

Αυτή η διάσπαση αποτελεί χαρακτηριστικό γνώρισμα κατανεμημένων συστημάτων, καθώς κάθε κομμάτι λειτουργεί ανεξάρτητα αλλά συνεργάζεται άψογα με τα υπόλοιπα, προσφέροντας ένα συνεκτικό και αποδοτικό περιβάλλον χρήσης.

Η υλοποίηση της εφαρμογής έγινε με έμφαση στην εμπειρία χρήστη, την απλότητα χρήσης και τη modular αρχιτεκτονική του συστήματος. Μέσα από μια σύγχρονη και φιλική προς τον χρήστη διεπαφή, ο επισκέπτης μπορεί να πλοηγηθεί στη λίστα διαθέσιμων εστιατορίων, να επιλέξει το κατάλληλο για εκείνον και να πραγματοποιήσει κράτηση με μερικά απλά βήματα.

Στο backend υλοποιήθηκε RESTful API το οποίο χειρίζεται αιτήματα εγγραφής και σύνδεσης χρηστών, δημιουργία και ανάκτηση κρατήσεων, καθώς και προστασία μέσω token-based authentication. Η βάση δεδομένων αποθηκεύει χρήστες, κρατήσεις και εστιατόρια, με κανονικοποιημένη δομή για ευκολία και αποδοτικότητα.

Η εφαρμογή σχεδιάστηκε ώστε να μπορεί εύκολα να επεκταθεί με νέες λειτουργίες όπως προσθήκη αξιολογήσεων, ενσωμάτωση χάρτη με γεωγραφική αναζήτηση ή και σύστημα ειδοποιήσεων (push notifications).

Αποτελεί μια ρεαλιστική υλοποίηση κατανεμημένης αρχιτεκτονικής και προσφέρει μια πλήρη εμπειρία ανάπτυξης, συνδυάζοντας τεχνολογίες frontend, backend και βάσης δεδομένων με τρόπο που αντικατοπτρίζει πραγματικά σενάρια της βιομηχανίας.

**ΛΕΙΤΟΥΡΓΕΙΕΣ ΧΡΗΣΤΗ**

Η εφαρμογή παρέχει ένα πλήρες φάσμα λειτουργιών για τον χρήστη, σχεδιασμένων με γνώμονα την απλότητα και την αποτελεσματικότητα:

• Εγγραφή Χρήστη: Ο χρήστης δημιουργεί νέο λογαριασμό με όνομα, email και κωδικό πρόσβασης. Τα στοιχεία αποθηκεύονται με ασφάλεια, με χρήση hashing για τον κωδικό (bcrypt).

• Σύνδεση Χρήστη: Μετά την εγγραφή, ο χρήστης μπορεί να συνδεθεί χρησιμοποιώντας τα διαπιστευτήριά του. Εφόσον η σύνδεση είναι επιτυχής, λαμβάνει ένα JWT token που αποθηκεύεται τοπικά και χρησιμοποιείται για ασφαλή επικοινωνία με το backend.

• Προβολή Διαθέσιμων Εστιατορίων: Στην αρχική σελίδα εμφανίζονται όλα τα διαθέσιμα εστιατόρια, με εικόνα, όνομα και τοποθεσία. Ο χρήστης μπορεί να δει περισσότερες λεπτομέρειες και να ξεκινήσει διαδικασία κράτησης.

• Δημιουργία Κράτησης: Ο χρήστης επιλέγει εστιατόριο, ημερομηνία, ώρα και αριθμό ατόμων. Το σύστημα αποστέλλει αίτημα POST στο backend API για να δημιουργηθεί η κράτηση στη βάση.

• Προβολή Προφίλ και Κρατήσεων: Στην ενότητα προφίλ, ο χρήστης μπορεί να δει όλες τις κρατήσεις που έχει κάνει. Η λίστα παρουσιάζει τα στοιχεία κάθε κράτησης μαζί με δυνατότητα τροποποίησης ή διαγραφής.

• Τροποποίηση/Ακύρωση Κράτησης: Ο χρήστης έχει τη δυνατότητα να επεξεργαστεί την ημερομηνία, ώρα και αριθμό ατόμων μιας μελλοντικής κράτησης, ή να την ακυρώσει εντελώς μέσω API.

• Αποσύνδεση: Παρέχεται δυνατότητα αποσύνδεσης, η οποία καθαρίζει το αποθηκευμένο token και επιστρέφει τον χρήστη στην αρχική οθόνη.

**ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ**

Το σύστημα "Restaurant App" σχεδιάστηκε με βάση την αρχή της τριεπίπεδης αρχιτεκτονικής (3-tier architecture), ώστε κάθε μέρος (frontend, backend, database) να μπορεί να λειτουργεί και να εξελίσσεται ανεξάρτητα.

**Frontend: React Native + Expo**

Υλοποιεί το γραφικό περιβάλλον της εφαρμογής για κινητά (Android).

Χρησιμοποιεί το Expo για ταχύτερη ανάπτυξη και εύκολη εκτέλεση μέσω QR code.

Ο χρήστης μπορεί να:

Εγγραφεί / Συνδεθεί

Δει λίστα εστιατορίων (με εικόνες)

Κάνει κράτηση

Διαχειριστεί τις κρατήσεις του (μέσω Προφίλ)

**Backend: Node.js + Express**

Υλοποιεί REST API με endpoints όπως:

/api/register – Εγγραφή χρήστη

/api/login – Σύνδεση και επιστροφή token

/api/reservations – CRUD κρατήσεων

/api/user/reservations – Λήψη κρατήσεων χρήστη

Η λογική διαχωρίζεται σε routes, middleware (π.χ. JWT auth) και βάση δεδομένων.

Κωδικοποιεί τους κωδικούς με bcrypt και χρησιμοποιεί JWT για authentication.

**Database: MariaDB**

Υλοποιεί την αποθήκευση δεδομένων:

users: Πληροφορίες χρηστών

restaurants: Στατικά δεδομένα για διαθέσιμα εστιατόρια

reservations: Κρατήσεις με σχέση χρήστη-εστιατορίου

Σχεδιάστηκε με primary/foreign keys και σωστή κανονικοποίηση.

**Authentication με JWT**

Μετά τη σύνδεση, ο server δημιουργεί ένα JSON Web Token (JWT) που περιέχει το user\_id.

Το token στέλνεται στο frontend και αποθηκεύεται τοπικά.

Σε κάθε επόμενο request, προστίθεται στο header Authorization: Bearer <token>.

Ο backend ελέγχει την εγκυρότητα μέσω middleware.

**Επικοινωνία μέσω RESTful API**

Όλα τα επίπεδα επικοινωνούν με HTTP αιτήματα:

Το frontend στέλνει JSON μέσω fetch/axios.

Το backend απαντά με JSON (π.χ. { message: "...", token: "..." }).

Τα αιτήματα προστατεύονται με token-based authorization, για να αποκλείεται πρόσβαση σε μη εξουσιοδοτημένους χρήστες.

**ΔΟΜΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ**

Η εφαρμογή βασίζεται σε σχεσιακή βάση δεδομένων MariaDB. Έχουν δημιουργηθεί τρεις βασικοί πίνακες που σχετίζονται μεταξύ τους με primary και foreign keys:

**Users (Χρήστες):**  
 - user\_id (PK)  
 - name  
 - email  
 - password (hashed)

**Restaurants (Εστιατόρια):**  
 - restaurant\_id (PK)  
 - name  
 - location

**Reservations (Κρατήσεις):**  
 - reservation\_id (PK)  
 - user\_id (FK)  
 - restaurant\_id (FK)  
 - reservation\_date  
 - reservation\_time  
 - people\_count

Οι σχέσεις είναι οι εξής:  
- Ένας χρήστης μπορεί να έχει πολλές κρατήσεις  
- Ένα εστιατόριο μπορεί να συνδέεται με πολλές κρατήσεις  
- Μια κράτηση συνδέει έναν χρήστη με ένα εστιατόριο

**ΠΡΟΚΛΗΣΕΙΣ ΚΑΙ ΕΠΙΛΥΣΗ**

Κατά την ανάπτυξη της εφαρμογής εντοπίστηκαν οι παρακάτω τεχνικές και πρακτικές προκλήσεις:

Διαχείριση authentication state: Η διαχείριση του token στο frontend απαιτούσε σωστή αποθήκευση και ανάκτηση από το AsyncStorage.

Σφάλματα επικοινωνίας frontend-backend: Η διαφορετική IP ανά εκτέλεση απαιτούσε προσαρμογές στον κώδικα και .env.

MariaDB access: Η σύνδεση μέσω του pool έπρεπε να γίνεται με σωστό release για αποφυγή leaks.

Debugging σε κινητό με Expo: Το Expo Go είχε περιορισμούς SDK και debugging δικτύου, που ξεπεράστηκαν με console.log και Alert().

**SCREENSHOTS ΕΦΑΡΜΟΓΗΣ**

Στα παρακάτω screenshots παρουσιάζεται η ροή χρήσης της εφαρμογής:

• Αρχική οθόνη με λίστα εστιατορίων

• Φόρμα σύνδεσης και εγγραφής

• Δημιουργία νέας κράτησης

• Σελίδα προφίλ με προβολή, τροποποίηση και διαγραφή κρατήσεων











