Εικόνα που περιέχει γραμματοσειρά, σχεδίαση

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο, λογότυπο, γραφικά, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**(ΣΤ’ εξάμηνο)**

**MOBILE & DISTRIBUTED SYSTEMS**

**Εικόνα που περιέχει σκίτσο/σχέδιο, ζωγραφιά, διάγραμμα, παράλληλα

Περιγραφή που δημιουργήθηκε αυτόματαΑΤΟΜΙΚΗ ΕΡΓΑΣΙΑ**

ΒΑΣΙΛΗΣ ΠΕΡΔΙΚΟΥΛΗΣ

[GitHub repo](https://github.com/VPERD03/myrestaurantapp)

UEL: 2471046

**Επόπτρια Καθηγήτρια: Αικατερίνη Ζόντου**

**ΠΑΤΡΑ**

**2024 - 2025**

Εικόνα που περιέχει γραμματοσειρά, σχεδίαση

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο, λογότυπο, γραφικά, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**(ΣΤ’ εξάμηνο)**

**MOBILE & DISTRIBUTED SYSTEMS**

**Ανάπτυξη Εφαρμογής Κράτησης Θέσεων σε Εστιατόριο**

**ΒΑΣΙΛΗΣ ΠΕΡΔΙΚΟΥΛΗΣ**

**Επόπτρια Καθηγήτρια: Αικατερίνη Ζόντου**

**ΠΑΤΡΑ**

**2024 – 2025**

**Περιεχόμενα**

* Εισαγωγή
* Τεχνολογίες που χρησιμοποιήθηκαν
* Ανάλυση Λειτουργικότητας / Χρήσης
* Ανάλυση Ασφάλειας και Authentication
* Πλεονεκτήματα και Περιορισμοί της Εφαρμογής
* Έλεγχος με Postman
* Εισαγωγή

Η τεχνολογία έχει εισχωρήσει σε κάθε πτυχή της καθημερινής μας ζωής, μεταμορφώνοντας τον τρόπο με τον οποίο εργαζόμαστε, επικοινωνούμε και εξυπηρετούμαστε. Ένας από τους τομείς όπου αυτή η μετάβαση γίνεται έντονα αισθητή είναι η εστίαση. Οι σύγχρονες ανάγκες των πελατών απαιτούν ταχύτητα, απλότητα και ευελιξία, ειδικά όταν πρόκειται για κρατήσεις σε εστιατόρια. Αντί να απαιτείται τηλεφωνική επικοινωνία ή φυσική παρουσία, όλο και περισσότερες επιχειρήσεις εστίασης στρέφονται σε ψηφιακές λύσεις. Σε αυτό το πλαίσιο εντάσσεται και η εφαρμογή **MyRestaurantApp**.

Η **MyRestaurantApp** αποτελεί μια web εφαρμογή πλήρους στοίβας (full-stack) που προσφέρει δυνατότητες **διαχείρισης κρατήσεων σε εστιατόρια**. Ο βασικός της σκοπός είναι να επιτρέψει στους πελάτες να διαχειρίζονται εύκολα τις κρατήσεις τους από τον υπολογιστή ή το κινητό τους παρέχοντας ταυτόχρονα έναν ασφαλή και αποδοτικό τρόπο αποθήκευσης και προβολής των δεδομένων για τον ιδιοκτήτη ή διαχειριστή του εστιατορίου.

Η εφαρμογή απευθύνεται κυρίως σε πελάτες προσφέροντάς τους:

* τη δυνατότητα εγγραφής και σύνδεσης,
* τη δημιουργία νέας κράτησης σε συγκεκριμένο εστιατόριο, ημερομηνία και ώρα,
* την προβολή των υπαρχουσών κρατήσεων τους,
* καθώς και τη δυνατότητα να τροποποιήσουν ή να διαγράψουν μια κράτηση εφόσον χρειάζεται.

Στο backend, η εφαρμογή έχει σχεδιαστεί με γνώμονα την ασφάλεια και επεκτασιμότητα, χρησιμοποιώντας Node.js και Express. Η επικοινωνία με τη βάση δεδομένων MariaDB γίνεται με χρήση SQL queries, ενώ για την αυθεντικοποίηση χρηστών χρησιμοποιούνται JSON Web Tokens (JWT), εξασφαλίζοντας ασφαλείς διαδρομές (routes) για τα ευαίσθητα δεδομένα.

Το frontend είναι υλοποιημένο με React.js, ένα από τα πιο διαδεδομένα frameworks για κατασκευή μοντέρνων web εφαρμογών. Η χρήση React προσφέρει δυναμικότητα και ταχύτητα στην εμπειρία χρήστη, ενώ επιτρέπει την εύκολη συντήρηση και επέκταση της εφαρμογής στο μέλλον.

Η MyRestaurantApp αποτελεί ένα ολοκληρωμένο παράδειγμα συγχρονισμένης συνεργασίας frontend και backend τεχνολογιών με στόχο την βελτίωση της εμπειρίας του χρήστη στην εξυπηρέτηση του κλάδου της εστίασης. Μέσα από αυτή την εργασία επιδιώκεται όχι μόνο η επίδειξη τεχνικών δεξιοτήτων, αλλά και η εφαρμογή αρχών καλής σχεδίασης, ασφαλούς ανάπτυξης και φιλικής εμπειρίας χρήστη (UX).

* Τεχνολογίες που χρησιμοποιήθηκαν

Η ανάπτυξη της εφαρμογής MyRestaurantApp βασίστηκε σε ένα σύγχρονο τεχνολογικό οικοσύστημα το οποίο εξασφαλίζει **ταχύτητα, ασφάλεια, ευελιξία** και **επέκταση στο μέλλον**. Παρακάτω αναλύονται αναλυτικά οι τεχνολογίες που χρησιμοποιήθηκαν τόσο στο frontend όσο και στο backend της εφαρμογής.

**Frontend-React.js**

Η διεπαφή χρήστη (User Interface) κατασκευάστηκε χρησιμοποιώντας το React.js, μια δημοφιλή JavaScript βιβλιοθήκη που αναπτύχθηκε από το Facebook και προορίζεται για τη δημιουργία **δυναμικών και διαδραστικών web εφαρμογών**.

**Πλεονεκτήματα χρήσης React**:

* **Component-based αρχιτεκτονική**: Ο κώδικας οργανώνεται σε επαναχρησιμοποιήσιμα components (π.χ. Navbar, ReservationForm), διευκολύνοντας τη συντήρηση και επέκταση της εφαρμογής.
* **Virtual Dom**: Βελτιώνει σημαντικά την απόδοση της εφαρμογής, καθώς μόνο τα απαραίτητα τμήματα του Dom ανανεώνονται όταν αλλάζει η κατάσταση (state).
* **Χρήση hooks (useState, useEffect)**: Επιτρέπουν την εύκολη διαχείριση κατάστασης και side-effects των components.
* **Rooting με React Router**: Χρησιμοποιείται για την πλοήγησή μεταξύ των σελίδων (Login, Register, Profile, Reservation), προσφέροντας εμπειρία Single Page Application (SPA).

Επιπλέον δόθηκε έμφαση στην εμπειρία χρήστη (UX), με χρήση dropdowns για επιλογή ώρας, περιορισμούς σε ημερομηνίες και responsive σχεδιασμό.

**Backend-Node.js & Express**

Το backend της εφαρμογής υλοποιήθηκε με Node.js ένα runtime περιβάλλον για JavaScript που επιτρέπει την ανάπτυξη server-side εφαρμογών. Πάνω από αυτό χρησιμοποιήθηκε το Express.js ένα ελαφρύ και ευέλικτο web framework που απλοποιεί τη δημιουργία RESTful APIs.

**Κύριες λειτουργίες στο backend:**

* **Routes για Authentication** ( /api/auth ): login/register με bcrypt και JWT.
* **Routes για κρατήσεις** ( /api/reservations ): δημιουργία, προβολή, επεξεργασία, διαγραφή.
* **Middleware για έλεγχο ταυτότητας (authMiddlewar.js)**: Επιβεβαιώνει το token πριν πριν επιτραπεί πρόσβαση σε προστατευμένες διαδρομές.
* **Έλεγχοι εγκυρότητας εισόδου**: Εξασφαλίζεται ότι οι χρήστες δεν μπορούν να κάνουν κράτηση σε παρελθόντα χρόνο ή εκτός ωραρίου.

Η επιλογή του Node.js ενισχύει τη συνοχή της εφαρμογής, αφού τόσο το frontend όσο και το backend βασίζονται στη JavaScript.

**Βάση Δεδομένων-MariaDB**

Για την αποθήκευση των δεδομένων (χρήστες, κράτησης, εστιατόρια), χρησιμοποιήθηκε η MariaDB, ένα αξιόπιστο και ανοιχτού κώδικα σύστημα σχεσιακής βάσης δεδομένων (RDBMS) που βασίζεται στο MySQL.

Δομή της βάσης:

* users: Πληροφορίες για χρήστες (όνομα, email, hashed password).
* restaurants: Στατικά δεδομένα για διαθέσιμα εστιατόρια.
* reservations: Κρατήσεις συνδεδεμένες με τον χρήστη και το εστιατόριο.

Η σύνδεση με τον backend πραγματοποιείται μέσω του πακέτου mysql2 (Promise-based), το οποίο επιτρέπει την εκτέλεση ασφαλών, παραμετροποιημένων SQL queries.

**Ασφάλεια-JWT & Bcrypt**

Η εφαρμογή ενσωματώνει μηχανισμούς ασφαλείας, εξασφαλίζοντας ότι τα δεδομένα των χρηστών προστατεύονται κατάλληλα.

* JWT (JSON Web Tokens): Μετά το login εκδίδεται ένα token το οποίο αποθηκεύεται στο localstorage και χρησιμοποιείται σε κάθε προστατευμένο API call μέσω του header Authorization. Το token περιέχει πληροφορίες για τον χρήστη (email, id) και έχει διάρκεια ζωής 1 ώρα.
* Bcrypt: Χρησιμοποιείται για το hashing των κωδικών πρόσβασης πριν αποθηκευτούν στη βάση. Κατά τη σύνδεση, ο κωδικός ελέγχεται με χρήση της μεθόδου bcrypt.compare().

**Εργαλεία και περιβάλλον**

* Postman: Για τη δοκιμή των API endpoints (CRUD στις κρατήσεις, εγγραφή, σύνδεση).
* VS Code: Κύριο εργαλείο ανάπτυξης.
* GitHub: Διαχείριση έκδοσης και αποθήκευση του κώδικα.
* npm: για τη διαχείριση εξαρτήσεων (π.χ. express, bcrypt, jsonwebtoken, cors, react-router-dom).
* Ανάλυση Λειτουργικότητας / Χρήσης

Η εφαρμογή MyRestaurantApp στοχεύει στη διευκόλυνση των πελατών ενός εστιατορίου ή μιας αλυσίδας εστιατορίων στην κράτηση τραπεζιών, μέσα από έναν απλό και φιλικό προς το χρήστη ιστότοπο. Η λειτουργικότητα καλύπτει πλήρως τον κύκλο ζωής μιας κράτησης (CRUD – Create, Read, Update, Delete), καθώς και τις βασικές ενέργειες ενός χρήστη όπως εγγραφή και σύνδεση.

**Εγγραφή και Σύνδεση Χρήστη**

Ο χρήστης έχει τη δυνατότητα να δημιουργήσει έναν λογαριασμό με βασικά στοιχεία: όνομα,email και κωδικό πρόσβασης. Η σύνδεση γίνεται με χρήση email/κωδικού και μετά την επιτυχή ταυτοποίηση, ο χρήστης λαμβάνει ένα JWT token το οποίο αποθηκεύεται στο localstorage. Το token αυτό επιτρέπει την πρόσβαση σε προστατευμένες διαδρομές (routes) της εφαρμογής, όπως η σελίδα προφίλ ή η φόρμα κράτησης.

**Δημιουργία Κράτησης**

Ο αυθεντικοποιημένος χρήστης μπορεί να προχωρήσει σε κράτηση επιλέγοντας:

* Εστιατόριο (από λίστα)
* Ημερομηνία (>= σημερινή)
* Ώρα (12:00 έως 23:00 σε 30 λεπτά διαστήματα)
* Αριθμό ατόμων

Η κράτηση αποστέλλεται στον server μέσω POST /api/reservations, και αποθηκεύεται σε πίνακα της βάσης δεδομένων. Σε περίπτωση επιτυχίας, ο χρήστης ενημερώνεται με μήνυμα και ανακατευθύνετε στο προφίλ του.

**Εμφάνιση Κρατήσεων (Read)**

Στη σελίδα προφίλ εμφανίζονται όλες οι κρατήσεις του χρήστη οι οποίες φορτώνονται δυναμικά από το endpoint GET /api/reservations/user. Για κάθε κράτηση εμφανίζονται:

* Όνομα εστιατορίου
* Ημερομηνία και ώρα
* Αριθμό ατόμων

**Τροποποίηση Κράτησης (Update)**

Κάθε κράτηση μπορεί να τροποποιηθεί από τον ίδιο τον χρήστη μέσω ειδικής φόρμας που εμφανίζεται με το πάτημα ενός κουμπιού “τροποποίηση”. Ο χρήστης μπορεί να αλλάξει ημερομηνία, ώρα ή αριθμό ατόμων αρκεί η νέα ώρα να βρίσκεται εντός ωραρίου και να μην είναι παρελθοντική. Το σχετικό endpoint είναι PUT /api/reservations/:id.

**Διαγραφή Κράτησης (Delete)**

Ο χρήστης έχει τη δυνατότητα να διαγράψει μια κράτηση μέσα από το προφίλ του με το πάτημα ενός κουμπιού “Διαγραφή”. Η διαγραφή επιβεβαιώνεται πρώτα από τον χρήστη και εκτελείται με DELETE /api/reservations/:id. Μετά την επιτυχή διαγραφή, η κράτηση αφαιρείται από τη λίστα εμφανίσεων.

**Μηνύματα Επιβεβαίωσης και Χειρισμός Σφαλμάτων**

Η εφαρμογή παρέχει φιλικά μηνύματα επιτυχίας ή σφάλματος για κάθε ενέργεια,ώστε ο χρήστης να έχει σαφή εικόνα για την κατάσταση των ενεργειών του. Επιπλέον γίνεται έλεγχος εξουσιοδότησης (authorization) για κάθε ευαίσθητη ενέργεια όπως π.χ. τροποποίηση ή διαγραφή

* Ανάλυση Ασφάλειας και Authentication

Η ασφάλεια αποτελεί βασικό πυλώνα της εφαρμογής MyRestaurantApp, καθώς περιλαμβάνει πληροφορίες χρηστών όπως email και κρατήσεις. Σε αυτό το κεφάλαιο περιγράφονται οι βασικοί μηχανισμοί authentication (ταυτοποίηση), authorization (εξουσιοδότηση), καθώς και τα μέτρα που εφαρμόζονται για την ασφαλή επικοινωνία μεταξύ frontend και backend.

**Χρήση JWT για Authentication**

Η ταυτοποίηση χρηστών γίνεται με χρήση JSON Web Tokens (JWT):

* Κατά τη σύνδεση, ο server δημιουργεί ένα υπογεγραμμένο token με το μυστικό JWT\_SECRET και το στέλνει στον cilent.
* Το token περιέχει πληροφορίες όπως το email και το id του χρήστη, καθώς και χρόνο λήξης (exp).
* Το token αποθηκεύεται στο localStorage του backend και χρησιμοποιείται για την εξουσιοδοτημένη πρόσβαση στις προστατευμένες διαδρομές της εφαρμογής.

**Middleware Ελέγχου Token**

Στο backend, υπάρχει custom middleware (authMiddleware.js) το οποίο:

1. Διαβάζει το token από το header Authorization.
2. Επαληθεύει την εγκυρότητα και τη διάρκεια ζωής του.
3. Αν το token είναι έγκυρο, προσθέτει τα στοιχεία του χρήστη στο req.user .
4. Αν είναι άκυρο ή ληγμένο επιστρέφει 401 (Unauthorized).

Αυτό εξασφαλίζει ότι μόνο αυθεντικοί χρήστες μπορούν να αποκτήσουν πρόσβαση στα endpoints κράτησης ή διαγραφής.

**Προστατευμένα Routes στο Frontend**

Στην πλευρά του client, χρησιμοποιείται το component ProtectedRoute, το οποίο:

* Ελέγχει αν υπάρχει JWT στο localStorage.
* Κάνει decode του token και ελέγχει αν έχει λήξει.
* Αν όλα είναι έγκυρα, επιτρέπει την πρόσβαση στον χρήστη.
* Διαφορετικά τον ανακατευθύνει αυτόματα στη σελίδα σύνδεσης.

Αυτό ενισχύει τη χρηστική ασφάλεια, εξασφαλίζοντας ότι κανείς δεν μπορεί να εισέλθει σε σελίδες χωρίς αυθεντικοποίηση.

**Ασφαλής επικοινωνία Frontend - Backend**

* Όλες οι ευαίσθητες αιτήσεις προς τον server (POST, PUT, DELETE) γίνονται με το Authorization header.
* Η React χρησιμοποιεί τη βοηθητική συνάρτηση secureFetch ώστε να διασφαλίζεται ότι:

1. Το token περιλαμβάνεται σε κάθε αίτηση.
2. Αν ο server επιστρέφει 401, γίνεται αυτόματη αποσύνδεση και επαναπροώθηση.

**Έλεγχοι Server-Side για Ενίσχυση Ασφάλειας**

Ο backend εκτελεί επιπλέον ελέγχους όπως:

* Πιστοποίηση ιδιοκτησίας κράτησης π.χ. ο χρήστης δεν μπορεί να διαγράψει κράτηση άλλου χρήστη.
* Επιβεβαίωση στοιχείων εισόδου π.χ. το email είναι μοναδικό και ο κωδικός αποθηκεύεται με χρήση bcrypt hashing.

Με αυτή την προσέγγιση η εφαρμογή παρέχει ισχυρό επίπεδο ασφάλειας για τα δεδομένα των χρηστών και προστατεύει από κοινές απειλές όπως εξουσιοδότηση μη εξουσιοδοτημένων αιτημάτων και κλοπή ταυτότητας.

* Πλεονεκτήματα και Περιορισμοί της Εφαρμογής

**Πλεονεκτήματα**

Η εφαρμογή MyRestaurantApp προσφέρει μια σειρά από πλεονεκτήματα τόσο σε επίπεδο χρήσης όσο και σε επίπεδο υλοποίησης:

* **Ευκολία Χρήσης**

Η διεπαφή χρήστη (UI) είναι καθαρή και φιλική προς τον χρήστη. Οι βασικές λειτουργίες (κράτηση, προβολή, διαγραφή, επεξεργασία) ολοκληρώνονται με λίγα κλικ.

* **Πλήρες CRUD για κρατήσεις**

Ο χρήστης μπορεί να:

* Δημιουργεί κράτηση (Create),
* Βλέπει τις κρατήσεις του (Read),
* Τις επεξεργάζεται (Update),
* Και τις διαγράφει (Delete).
* **Ασφαλής Αυθεντικοποίηση**

Με την χρήση JWT και middleware, διασφαλίζεται ότι κάθε χρήστης έχει πρόσβαση μόνο στα δικά του δεδομένα.

* **Αντιμετώπιση edge cases**

Η εφαρμογή ελέγχει:

* Ημερομηνίες στο παρελθόν,
* Ώρες εκτός ωραρίου,
* πλήθος ατόμων, με τρόπο που ενισχύει την αξιοπιστία της.
* **Εύκολη επέκταση**

Ο κώδικας είναι modular τόσο στο frontend όσο και στο backend, επιτρέποντας εύκολη προσθήκη νέων δυνατοτήτων στο μέλλον (π.χ. βαθμολογίες, κράτηση με προκαταβολή, διαχείριση από ιδιοκτήτες).

**Περιορισμοί**

Παρότι η εφαρμογή είναι πλήρως λειτουργική, παρουσιάζει κάποιους αναγνωρισμένους περιορισμούς:

* **Έλλειψη ρόλων χρηστών (π.χ. admin)**

Προς το παρόν, όλοι οι χρήστες έχουν ίσα δικαιώματα και δεν υπάρχει δυνατότητα διαχείρισης των κρατήσεων από πλευράς εστιατορίου.

* **Μη χρήση SSL (https)**

Η τοπική ανάπτυξη γίνεται χωρίς SSL, κάτι που είναι κρίσιμο για production περιβάλλον.

* **Απουσία email verification / reset password**

Η διαδικασία εγγραφής είναι άμεση και δεν υπάρχει έλεγχος εγκυρότητας email ή δυνατότητα επαναφοράς κωδικού.

* Έλεγχος με Postman

Για να διασφαλιστεί ότι το backend της εφαρμογής λειτουργεί σωστά, πραγματοποιήθηκαν δοκιμές μέσω του εργαλείου Postman. Οι δοκιμές επικεντρώθηκαν στη λειτουργία σύνδεσης και εγγραφής χρηστών.

* Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο, λογισμικό, λογισμικό πολυμέσων

  Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.**Εγγραφή χρήστη (Post /api/auth/register)**

Μέσω του Postman στέλνεται JSON payload με πεδία name, email και password. Η επιτυχής απόκριση επιστρέφει μήνυμα επιτυχίας και αποθηκεύει τον χρήστη στη βάση.

* Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο, λογισμικό, λογισμικό πολυμέσων

  Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.**Σύνδεση χρήστη (POST /api/auth/login)**
* Αποστέλλονται email και password. Αν είναι έγκυρα, ο server επιστρέφει JWT token, που χρησιμοποιείται για προστατευμένα endpoints.

[GitHub repo](https://github.com/VPERD03/myrestaurantapp)