 A logo of a university

AI-generated content may be incorrect.

**Mobile & Distributed Systems**

(CN6035)

Λεκός Νικόλαος

(2471041)

**Περιεχόμενα**

1. **Εισαγωγή**
2. **Σκοπός και Στόχοι της Εργασίας**
3. **Χρησιμοποιημένες Τεχνολογίες**  
   3.1. Frontend – React  
   3.2. Backend – Node.js & Express  
   3.3. Βάση Δεδομένων – MariaDB  
   3.4. Εργαλεία και Βιβλιοθήκες
4. **Ανάλυση Λειτουργικότητας της Εφαρμογής**  
   4.1. Εγγραφή και Σύνδεση  
   4.2. Προβολή Εστιατορίων  
   4.3. Δημιουργία Νέας Κράτησης  
   4.4. Προβολή και Διαχείριση Κρατήσεων  
   4.5. Responsive και Φιλικό Περιβάλλον Χρήστη
5. **Σχεδιασμός και Υλοποίηση του Συστήματος**  
   5.1. Ανάπτυξη Frontend (React)  
   5.2. Ανάπτυξη Backend (Node.js & Express)  
   5.3. Βάση Δεδομένων (MariaDB)  
   5.4. CRUD Λειτουργικότητες  
   5.5. Αντιμετώπιση Σφαλμάτων
6. **Έλεγχος και Δοκιμές**  
   6.1. Έλεγχος Frontend  
   6.2. Έλεγχος Backend (API Testing)  
   6.3. Έλεγχος Βάσης Δεδομένων  
   6.4. Δοκιμές από Χρήστες
7. **Συμπεράσματα και Μελλοντικές Επεκτάσεις**
8. **Βιβλιογραφία**

# ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία επικεντρώνεται στην ανάπτυξη μιας **ολοκληρωμένης διαδικτυακής εφαρμογής** για κρατήσεις σε εστιατόρια, η οποία συνδυάζει σύγχρονες τεχνολογίες ανάπτυξης τόσο στο **frontend** όσο και στο **backend,** με διασύνδεση σε σχεσιακή **βάση δεδομένων**. Στόχος της εφαρμογής είναι να προσφέρει στους χρήστες μια απλή αλλά και λειτουργική εμπειρία. Ο χρήστης έχει τη δυνατότητα να δημιουργήσει λογαριασμό μέσω εγγραφής με email και κωδικό πρόσβασης, να πραγματοποιήσει ασφαλή είσοδο στην πλατφόρμα μέσω authentication token (JWT) και να περιηγηθεί στην λίστα των διαθέσιμων εστιατορίων. Μέσα από ένα φιλικό και καθαρό περιβάλλον, μπορεί να επιλέξει εστιατόριο, να καθορίσει ημερομηνία, ώρα και αριθμό ατόμων και να πραγματοποιήσει την κράτησή του. Επιπλέον, του παρέχεται η δυνατότητα διαχείρισης των κρατήσεών του π.χ. μπορεί να επεξεργαστεί ή να διαγράψει μελλοντικές κρατήσεις εύκολα και γρήγορα. Η υποστήριξη της εφαρμογής βασίζεται σε ένα backend server υλοποιημένο με Node.js και Express, ενώ η αποθήκευση των δεδομένων γίνεται με τη χρήση MariaDB. Η συνολική αρχιτεκτονική σχεδιάστηκε ώστε να εξασφαλίζει ασφάλεια, αξιοπιστία και επεκτασιμότητα, ενώ παράλληλα προσφέρει μια ευχάριστη εμπειρία χρήσης τόσο από επιτραπέζιες συσκευές όσο και από κινητά τηλέφωνα. Η εργασία αυτή δεν αποτελεί μόνο τεχνική υλοποίηση αλλά και μια πρακτική εφαρμογή των αρχών των Κατανεμημένων Συστημάτων και της Ανάπτυξης Εφαρμογών Κινητών Συσκευών.

# Κεφάλαιο 1 – Εισαγωγή

Η τεχνολογία εξελίσσεται με αλματώδεις ρυθμούς, φέρνοντας συνεχώς νέες λύσεις και εργαλεία που επιτρέπουν στις επιχειρήσεις και τους καταναλωτές να αλληλεπιδρούν με πιο γρήγορους και αποτελεσματικούς τρόπους. Ένα από τα πεδία που έχει βιώσει σημαντική πρόοδο είναι η βιομηχανία των εστιατορίων, η οποία πλέον βασίζεται όλο και περισσότερο σε ψηφιακές λύσεις για την εξυπηρέτηση των πελατών και την αποτελεσματική διαχείριση των κρατήσεων. Παράλληλα, οι κινητές συσκευές έχουν καθιερωθεί ως βασικό εργαλείο καθημερινότητας για εκατομμύρια χρήστες παγκοσμίως, προσφέροντας άμεση πρόσβαση σε μια πληθώρα υπηρεσιών. Στην εποχή αυτή, η δυνατότητα να πραγματοποιήσει κάποιος κρατήσεις σε εστιατόρια μέσω μιας εφαρμογής κινητού δεν είναι απλώς μια πολυτέλεια, αλλά αποτελεί μια αναγκαία υπηρεσία για τη βελτίωση της εξυπηρέτησης του πελάτη και την αναβάθμιση των λειτουργιών των επιχειρήσεων.

Με την ολοένα αυξανόμενη χρήση των ψηφιακών εφαρμογών στην καθημερινή ζωή, η ανάπτυξη και χρήση αυτών των εφαρμογών στοχεύει στην απλοποίηση των διαδικασιών που συνήθως απαιτούν περισσότερο χρόνο και προσπάθεια. Στην παρούσα εργασία, αναπτύχθηκε μια ολοκληρωμένη διαδικτυακή εφαρμογή για κρατήσεις σε εστιατόρια, η οποία προσφέρει τη δυνατότητα στους χρήστες να πραγματοποιούν κρατήσεις μέσω των κινητών τους συσκευών, με ευκολία και ασφάλεια. Ο κύριος σκοπός της εφαρμογής αυτής είναι να προσφέρει στον χρήστη την ευχέρεια να επιλέξει το κατάλληλο εστιατόριο με βάση τις προσωπικές του προτιμήσεις, όπως την τοποθεσία, τον τύπο κουζίνας και την ημερομηνία, ενώ παράλληλα να του επιτρέπει να διαχειρίζεται τις κρατήσεις του (π.χ., επεξεργασία, ακύρωση).

Η ανάπτυξη αυτής της εφαρμογής βασίζεται σε τρεις βασικούς πυλώνες, οι οποίοι συνεργάζονται αρμονικά για να προσφέρουν μια πλήρη και λειτουργική λύση:

Frontend – React: Η ανάπτυξη του frontend έγινε με τη βιβλιοθήκη React, η οποία επιτρέπει την δημιουργία διαδραστικών και δυναμικών user interfaces. Το React προσφέρει τον έλεγχο του state της εφαρμογής, κάτι που είναι απαραίτητο για να εμφανίζονται και να επεξεργάζονται τα δεδομένα των χρηστών σε πραγματικό χρόνο. Η εφαρμογή προσφέρει ένα responsive design, ώστε να είναι λειτουργική τόσο σε desktop όσο και σε κινητές συσκευές.

Backend – Node.js & Express: Το backend της εφαρμογής αναπτύχθηκε χρησιμοποιώντας το Node.js και το framework Express, τα οποία παρέχουν μια ισχυρή και αποδοτική λύση για τη διαχείριση HTTP αιτημάτων και την επικοινωνία με τη βάση δεδομένων. Η χρήση του RESTful API επιτρέπει την αποστολή και λήψη δεδομένων με ευκολία και αποδοτικότητα, ενώ το JWT (JSON Web Token) προσφέρει μια ασφαλή μέθοδο για την αυθεντικοποίηση των χρηστών.

Βάση Δεδομένων – MariaDB: Για την αποθήκευση των δεδομένων, χρησιμοποιήθηκε η MariaDB, η οποία είναι μια ανοιχτού κώδικα έκδοση του MySQL και προσφέρει ισχυρή απόδοση και επεκτασιμότητα. Η βάση δεδομένων περιλαμβάνει πίνακες για τους χρήστες, τα εστιατόρια και τις κρατήσεις, ενώ η σύνδεση με το backend πραγματοποιείται μέσω SQL queries για τη διαχείριση των δεδομένων.

Αυτή η εφαρμογή προσφέρει στους χρήστες τη δυνατότητα να δημιουργούν λογαριασμό και να συνδέονται με ασφαλή τρόπο, ενώ τους επιτρέπει να κάνουν κράτηση σε εστιατόρια μέσω μιας απλής φόρμας εισόδου. Η φόρμα κράτησης περιλαμβάνει πεδία για την επιλογή του εστιατορίου, την ημερομηνία, την ώρα και τον αριθμό των ατόμων, με τα δεδομένα να καταχωρούνται στη βάση δεδομένων και να διαχειρίζονται μέσω του backend.

Η εφαρμογή διασφαλίζει την ασφάλεια των χρηστών με τη χρήση του JWT για την αυθεντικοποίηση και την εξουσιοδότηση κατά την πρόσβαση στις προστατευμένες λειτουργίες, όπως η διαχείριση των κρατήσεων. Επιπλέον, η responsive σχεδίαση της εφαρμογής επιτρέπει στους χρήστες να έχουν πλήρη λειτουργικότητα σε οποιαδήποτε συσκευή, είτε πρόκειται για desktop είτε για κινητό τηλέφωνο.

Στην εποχή που οι χρήστες αναζητούν εφαρμογές που προσφέρουν εύκολη πλοήγηση, ταχύτητα και αξιοπιστία, η παρούσα εφαρμογή ικανοποιεί πλήρως αυτές τις ανάγκες. Εξασφαλίζει μία ευχάριστη εμπειρία χρήστη, η οποία περιλαμβάνει τη δυνατότητα εύκολης αναζήτησης και κράτησης, ενώ παρέχει παράλληλα τις λειτουργίες διαχείρισης των κρατήσεων για τη βελτίωση της συνολικής εμπειρίας του χρήστη.Η εργασία αυτή δεν αποτελεί απλά μια τεχνική υλοποίηση, αλλά και μια πρακτική εφαρμογή των αρχών των κατανεμημένων συστημάτων και της ανάπτυξης εφαρμογών για κινητές συσκευές, προσφέροντας μία ολοκληρωμένη λύση για το μέλλον των κρατήσεων σε εστιατόρια.

# Κεφάλαιο 2 – Σκοπός και Στόχοι της Εργασίας

Ο βασικός σκοπός αυτής της εργασίας είναι η ανάπτυξη μιας πλήρως λειτουργικής διαδικτυακής εφαρμογής για κρατήσεις σε εστιατόρια, η οποία θα επιτρέπει στους χρήστες να πραγματοποιούν κρατήσεις μέσω κινητών συσκευών και να διαχειρίζονται τις κρατήσεις τους με εύκολο και ασφαλή τρόπο. Στόχος της εφαρμογής είναι να προσφέρει στους χρήστες μια ευχάριστη και απλή εμπειρία χρήσης, ενώ ταυτόχρονα να ενισχύσει την απόδοση και τη λειτουργικότητα των επιχειρήσεων εστίασης. Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας σύγχρονες τεχνολογίες και ακολουθώντας τις βασικές αρχές των κατανεμημένων συστημάτων και της αρχιτεκτονικής πελάτη-διακομιστή (client-server).

Η εργασία αυτή, πέρα από την τεχνική υλοποίηση, προσφέρει μια πλήρη κατανόηση των διαδικασιών που απαιτούνται για την κατασκευή μιας **σύγχρονης web εφαρμογής**, συνδυάζοντας τις καλύτερες πρακτικές ανάπτυξης frontend και backend με ασφάλεια, επεκτασιμότητα και ευχρηστία. Στην εργασία επιγράφονται οι εξής στόχοι:

**1. Ανάπτυξη Δυναμικού Frontend με React**

Το frontend της εφαρμογής αναπτύχθηκε με την **React**, μία από τις πιο δημοφιλείς βιβλιοθήκες JavaScript για τη δημιουργία διαδραστικών user interfaces. Η React επιτρέπει την κατασκευή single-page applications (SPA), όπου οι χρήστες αλληλεπιδρούν με τη σελίδα χωρίς να χρειάζεται να ανανεώνουν ολόκληρη τη σελίδα, προσφέροντας έτσι μια ομαλή και γρήγορη εμπειρία χρήσης.

Ο κύριος στόχος του frontend είναι να προσφέρει φιλικό και διαδραστικό περιβάλλον για τον χρήστη, επιτρέποντας του να περιηγηθεί εύκολα σε διαθέσιμα εστιατόρια, να κάνει κράτηση, να δει τις κρατήσεις του και να επεξεργαστεί ή να διαγράψει κρατήσεις από το προσωπικό του προφίλ. Η χρήση της React βοηθά στην εύκολη διαχείριση της κατάστασης (state management) της εφαρμογής, εξασφαλίζοντας ότι τα δεδομένα ενημερώνονται σε πραγματικό χρόνο χωρίς αναγκαία ανανέωση της σελίδας.

Επιπλέον, το responsive design της εφαρμογής επιτρέπει στους χρήστες να τη χρησιμοποιούν εξίσου εύκολα τόσο σε επιτραπέζιους υπολογιστές όσο και σε κινητές συσκευές, προσφέροντας βελτιστοποιημένη εμπειρία χρήσης σε όλες τις πλατφόρμες.

**2. Δημιουργία RESTful Backend με Express και Node.js**

Το backend της εφαρμογής αναπτύχθηκε χρησιμοποιώντας Node.js και το framework Express. Το Node.js είναι ένα JavaScript runtime environment που επιτρέπει την εκτέλεση JavaScript κώδικα στο διακομιστή, και το Express είναι ένα minimalistic framework για την ανάπτυξη RESTful APIs. Η επιλογή αυτών των τεχνολογιών έγινε με στόχο την ευχρηστία και την επιδότηση στην ανάπτυξη, καθώς και την εύκολη επεκτασιμότητα της εφαρμογής.

Η χρήση RESTful API επιτρέπει την επικοινωνία του frontend με το backend μέσω HTTP αιτημάτων (GET, POST, PUT, DELETE), επιτρέποντας στους χρήστες να εκτελούν ενέργειες όπως η δημιουργία νέας κράτησης, η προβολή εστιατορίων και η διαχείριση κρατήσεων. Για την ασφάλεια, η εφαρμογή υλοποιεί JWT (JSON Web Tokens) για την αυθεντικοποίηση χρηστών, εξασφαλίζοντας ότι μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να εκτελούν τις προστατευμένες ενέργειες.

**3. Υλοποίηση Μηχανισμού Εγγραφής και Σύνδεσης Χρηστών με JWT**

Η εφαρμογή παρέχει τη δυνατότητα στους χρήστες να δημιουργήσουν λογαριασμό με τη χρήση του email και ενός ασφαλούς κωδικού πρόσβασης, ο οποίος αποθηκεύεται με hashing μέσω της βιβλιοθήκης bcryptjs. Η σύνδεση πραγματοποιείται με τη χρήση JWT (JSON Web Token), το οποίο εκδίδεται μετά από την επιτυχή είσοδο και χρησιμοποιείται για την ασφαλή επικοινωνία μεταξύ του frontend και του backend. Το JWT επιτρέπει την ασφαλή πρόσβαση στις προστατευμένες διαδρομές της εφαρμογής, διασφαλίζοντας την προστασία των προσωπικών δεδομένων του χρήστη.

Με αυτή τη διαδικασία, η εφαρμογή παρέχει μια ασφαλή και ευέλικτη λύση για την αυθεντικοποίηση χρηστών, ενώ παράλληλα επιτρέπει την ευκολία στην ανανέωση του token και στην παρακολούθηση των συνδέσεων των χρηστών.

**4. Ενσωμάτωση Λειτουργικότητας Προβολής Εστιατορίων**

Η εφαρμογή παρέχει τη δυνατότητα στους χρήστες να περιηγηθούν σε διαθέσιμα εστιατόρια και να επιλέξουν αυτό που ταιριάζει στις προτιμήσεις τους. Μέσω του frontend ανακτώνται τα δεδομένα των εστιατορίων από τη βάση δεδομένων και εμφανίζονται στην οθόνη, περιλαμβάνοντας βασικές πληροφορίες όπως το όνομα του εστιατορίου, την τοποθεσία του και τις ώρες λειτουργίας.

Η εφαρμογή ενσωματώνει λειτουργικότητα αναζήτησης και φίλτρων, επιτρέποντας στους χρήστες να βρουν εύκολα εστιατόρια βάσει τοποθεσίας ή άλλων κριτηρίων. Η οπτική παρουσίαση των εστιατορίων είναι εμπλουτισμένη με εικόνες που βελτιώνουν την εμπειρία του χρήστη.

**5. Καταχώρηση Νέας Κράτησης μέσω Φόρμας**

Οι χρήστες μπορούν να πραγματοποιήσουν νέες κρατήσεις μέσω μιας απλής φόρμας, η οποία επιτρέπει την επιλογή του εστιατορίου, την ημερομηνία, την ώρα και τον αριθμό ατόμων. Οι κρατήσεις αποστέλλονται στο backend μέσω POST αιτήματος και καταχωρούνται στη βάση δεδομένων, εξασφαλίζοντας τη σωστή καταγραφή τους. Η εφαρμογή παρέχει επίσης μια επιβεβαίωση στους χρήστες όταν η κράτηση πραγματοποιηθεί με επιτυχία.

**6. Διαχείριση Κρατήσεων από το Προσωπικό Προφίλ Χρήστη**

Ο χρήστης μπορεί να δει τις υπάρχουσες κρατήσεις του μέσα από το προσωπικό του προφίλ. Από εκεί, μπορεί να επεξεργαστεί ή να διαγράψει τις κρατήσεις του αν το επιθυμεί. Αυτή η δυνατότητα προσφέρει στον χρήστη πλήρη έλεγχο των κρατήσεών του και επιτρέπει την εύκολη τροποποίηση ή ακύρωση κρατήσεων όταν είναι απαραίτητο.

**7. Δομημένη Αποθήκευση Δεδομένων σε Βάση MariaDB**

Για την αποθήκευση των δεδομένων χρηστών, εστιατορίων και κρατήσεων, χρησιμοποιήθηκε η βάση δεδομένων MariaDB. Η βάση περιλαμβάνει πίνακες για χρήστες, εστιατόρια και κρατήσεις, με σχέσεις μεταξύ τους που διασφαλίζουν την ακεραιότητα των δεδομένων. Η σχεδίαση της βάσης δεδομένων εξασφαλίζει ότι τα δεδομένα αποθηκεύονται με ασφάλεια και ανακτώνται γρήγορα, ενώ οι σχέσεις foreign key αποτρέπουν την εμφάνιση εσφαλμένων ή ασυνεπών δεδομένων.

**8. Βελτιστοποίηση της Εφαρμογής για Κινητές Συσκευές**

Η εφαρμογή είναι πλήρως responsive, πράγμα που σημαίνει ότι προσαρμόζεται αυτόματα σε διαφορετικά μεγέθη οθόνης. Οι χρήστες μπορούν να έχουν την ίδια εμπειρία πλοήγησης, είτε χρησιμοποιούν κινητό τηλέφωνο, tablet ή υπολογιστή. Η βελτιστοποίηση για κινητές συσκευές είναι κρίσιμη για την επιτυχία της εφαρμογής, καθώς επιτρέπει στους χρήστες να πραγματοποιούν κρατήσεις οπουδήποτε και οποτεδήποτε, εξασφαλίζοντας έτσι μια άριστη εμπειρία χρήσης.

# Κεφάλαιο 3 – Χρησιμοποιημένες Τεχνολογίες

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν σύγχρονες τεχνολογίες που καλύπτουν όλες τις απαίτησης μιας κατανεμημένης εφαρμογής-διακοσμητή. Η επιλογή αυτών έγινε με γνώμονα την ευχρηστία της εφαρμογής την επεκτασιμότητα και την συμβατότητα των επιμέρους μερών του συστήματος .

**FRONTEND-REACT**

Το frontend της εφαρμογής αναπτύχθηκε με τη βιβλιοθήκη React, η οποία προσφέρει ευελιξία στην κατασκευή δια δραστικών user interfaces με state management. Οι βασικές λειτουργίες που υποστηρίζονται είναι οι εξής:

* Φόρμες εγγραφής και σύνδεσης χρηστών.
* Προβολή λίστα διαθέσιμων εστιατορίων
* Προβολή λεπτομερειών(ώρες-μέρες λειτουργίας)
* Υποβολή και επεξεργασία κράτησης
* Responsive σχεδιασμός με υποστήριξη για κινητές συσκευές.
* Ενημέρωση δεδομένων μέσω API

**Backend – Node.js & Express**

O διακοσμητής υλοποιήθηκε με χρήση Node.js και του framework Express, για την διαχείριση RESTFULL API endpoints. Βασικά χαρακτηριστικά :

* Routes για εγγραφή, σύνδεση, προβολή εστιατορίων, δημιουργία και διαχείριση κρατήσεων.
* Middleware για προστασία των endpoints μέσω **JWT Authentication**.
* Εύκολη επεκτασιμότητα και αναγνωσιμότητα του κώδικα.
* Αντιμετώπιση λαθών και αποστολή κατάλληλων μηνυμάτων στον client.

**Βάση Δεδομένων – MariaDB**

Για την αποθήκευση των δεδομένων χρησιμοποιήθηκε **MariaDB**, μια open-source έκδοση του MySQL. Η δομή περιλαμβάνει τα εξής:

* Πίνακας **users**: αποθήκευση στοιχείων των χρηστών με hashed κωδικούς πρόσβασης.
* Πίνακας **restaurants** : περιλαμβάνει στοιχεία κάθε εστιατορίου όπως όνομα, τοποθεσία και ώρες λειτουργίας.
* Πίνακας **reservations**: αποθηκεύονται οι κρατήσεις κάθε χρήστη με το αντίστοιχο εστιατόριο.

**Εργαλεία και Βιβλιοθήκες**

* **JWT** (jsonwebtoken): Χρήση token-based authentication.
* **bcryptjs**: Για hashing των passwords πριν αποθηκευτούν.
* **Axios**: Για επικοινωνία του frontend με το backend μέσω HTTP requests.
* **HeidiSQL**: Οπτικό εργαλείο διαχείρισης της MariaDB.
* **Postman**: Χρήση για δοκιμή και debugging των API endpoints.

# Κεφάλαιο 4 – Ανάλυση Λειτουργικότητας της Εφαρμογής

Η εφαρμογή προσφέρει μια πλήρως λειτουργική εμπειρία στον χρήστη, από την εγγραφή μέχρι τη διαχείριση των κρατήσεών του. Στο κεφάλαιο αυτό περιγράφονται αναλυτικά όλες οι κύριες λειτουργίες της.

**Εγγραφή και σύνδεση**

Η λειτουργία αυθεντικοποιήσης υλοποιείτε μέσω:

* Εγγραφής: Ο χρήστης καταχωρεί το email και τον κωδικό του. Ο κωδικός καταγράφεται με ασφάλεια (bcrypt hashing).
* Σύνδεσης: Ο χρήστης εισάγει τα στοιχεία του και λαμβάνει JWT token, με το οποίο μπορεί να έχει πρόσβαση στα προστατευμένα endpoints.
* Σύνδεσης: Ο χρήστης εισάγει τα στοιχεία του και λαμβάνει JWT token, με το οποίο μπορεί να έχει πρόσβαση στα προστατευμένα endpoints.

**Προβολή εστιατορίων**

* Κάθε κάρτα εστιατορίου περιλαμβάνει όνομα, τοποθεσία και φωτογραφία.
* Με κλικ πάνω στη φωτογραφία εμφανίζεται παράθυρο με ώρες λειτουργίας του καταστήματος.
* Υποστηρίζεται αναζήτηση εστιατορίων (τοποθεσία/όνομα).

Δημιουργία Νέας Κράτησης

Η φόρμα κράτησης επιτρέπει τα εξής :

* Επιλογή εστιατορίου από dropdown.
* Επιλογή ημερομηνίας και ώρας
* Επιλογή αριθμού ατόμων.

Πατώντας επιβεβαίωση αποστέλλεται αίτημα στο backend και καταχωρείται στην βάση

**Προβολή και Διαχείριση Κρατήσεων**

Ο χρήστης μπορεί να δει τις κρατήσεις του, με τις παρακάτω δυνατότητες:

* **Προβολή ιστορικού κρατήσεων**.
* **Διαγραφή κράτησης** με ένα κλικ.
* **Επεξεργασία κράτησης**: Δυνατότητα αλλαγής ημερομηνίας, ώρας ή αριθμού ατόμων μέσω αντίστοιχης φόρμας.

**Responsive και Φιλικό Περιβάλλον Χρήστη**

* Η διεπαφή χρήστη έχει σχεδιαστεί με responsiveness, ώστε να λειτουργεί σωστά σε κινητά και υπολογιστές.
* Χρησιμοποιούνται μοντέρνες κάρτες, αναδυόμενα παράθυρα, και διακριτικά animations.
* Ειδική μέριμνα έχει δοθεί στην ευκολία πλοήγησης, ώστε ο χρήστης να μπορεί να κάνει κράτηση με λίγα μόνο βήματα.

# Κεφάλαιο 5 – Σχεδιασμός και Υλοποίηση του Συστήματος

Σε αυτό το κεφάλαιο περιγράφεται αναλυτικά η διαδικασία σχεδίασης και υλοποίησης της εφαρμογής κρατήσεων, τόσο από πλευράς frontend όσο και backend, καθώς και η διασύνδεσή τους με τη βάση δεδομένων.

**Ανάπτυξη Frontend (React)**

Η διεπαφή χρήστη υλοποιήθηκε με React και αποτελείται από τις παρακάτω βασικές σελίδες:

* Σελίδα Σύνδεσης και Εγγραφής: Οι χρήστες μπορούν να δημιουργήσουν λογαριασμό ή να συνδεθούν. Μετά την επιτυχία, αποθηκεύεται token στο localStorage.
* Αρχική Σελίδα με Εστιατόρια: Εμφανίζει όλα τα εστιατόρια με εικόνες και βασικές πληροφορίες. Πατώντας πάνω σε κάποιο, εμφανίζεται αναδυόμενο παράθυρο με ωράριο λειτουργίας.
* Σελίδα Νέας Κράτησης: Επιλογή εστιατορίου, ημερομηνίας, ώρας και αριθμού ατόμων.
* Σελίδα Προβολής Κρατήσεων: Ο χρήστης βλέπει όλες τις κρατήσεις του, με δυνατότητα διαγραφής ή επεξεργασίας.

Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Εικόνα που περιέχει κείμενο, κολάζ, στιγμιότυπο οθόνης

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ορθογώνιο παραλληλόγραμμο

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

**Ανάπτυξη Backend (Node.js & Express)**

Το backend υλοποιήθηκε με Node.js και Express. Περιλαμβάνει:

* Routes:

1./register: Εγγραφή νέου χρήστη.

2./login: Επιστροφή JWT token.

3./restaurants: Επιστροφή λίστας εστιατορίων.

4./reservations: Δημιουργία, επεξεργασία, διαγραφή κράτησης.

5./user/reservations: Επιστροφή κρατήσεων χρήστη.

* Middleware: Το verifyToken.js ελέγχει την εγκυρότητα του JWT πριν επιτραπεί πρόσβαση στα προστατευμένα endpoints.
* Controllers: Κάθε route καλεί τον αντίστοιχο controller, όπου γίνεται η επεξεργασία των δεδομένων και η διασύνδεση με τη βάση.



**Βάση Δεδομένων (MariaDB)**

Χρησιμοποιήθηκε η HeidiSQL για τον σχεδιασμό και τη διαχείριση της βάσης. Οι βασικοί πίνακες είναι:

* users: user\_id, name, email, password (hashed)
* restaurants: restaurant\_id, name, location, description, opening\_hours
* reservations: reservation\_id, user\_id, restaurant\_id, date, time, people\_count

Οι σχέσεις μεταξύ των πινάκων διασφαλίζουν ακεραιότητα και σωστή αναφορά (foreign keys).

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

**Λειτουργικότητες CRUD**

* Οι κρατήσεις μπορούν να δημιουργηθούν, προβληθούν, τροποποιηθούν και διαγραφούν.
* Ο χρήστης βλέπει μόνο τις δικές του κρατήσεις που είναι προστατευμένες με token.
* Εφαρμόστηκε επιβεβαίωση πριν τη διαγραφή κράτησης και feedback μετά από κάθε ενέργεια.

**Αντιμετώπιση Σφαλμάτων**

* Στο backend: try/catch με επαλήθευση παραμέτρων και επιστροφή σαφών μηνυμάτων.
* Στο frontend: Προβλέπεται alert για σφάλματα όπως αποτυχία login ή αποτυχία σύνδεσης με API.

# Κεφάλαιο 6: Έλεγχος και Δοκιμές

Στο κεφάλαιο αυτό παρουσιάζονται οι ενέργειες που πραγματοποιήθηκαν για τον έλεγχο της σωστής λειτουργίας της εφαρμογής, τόσο σε επίπεδο frontend όσο και backend, με σκοπό την ανίχνευση σφαλμάτων και την επιβεβαίωση της αξιοπιστίας του συστήματος.

**6.1 Έλεγχος Frontend**

Η διεπαφή χρήστη δοκιμάστηκε σε πολλαπλές συσκευές και αναλύσεις οθόνης ώστε να διασφαλιστεί η συμβατότητα και η προσαρμοστικότητα (responsiveness). Πραγματοποιήθηκαν δοκιμές για:

* Την επιτυχία ή αποτυχία της εγγραφής και σύνδεσης χρήστη.
* Την ορθή εμφάνιση της λίστας εστιατορίων.
* Τη λειτουργία των κρατήσεων και την ορθή εμφάνιση στο ιστορικό του χρήστη.
* Τη διαγραφή και τροποποίηση κρατήσεων, μέσω UI και εμφάνιση μηνυμάτων επιβεβαίωσης.

**6.2 Έλεγχος Backend (API Testing)**

Τα REST API endpoints ελέγχθηκαν με τη χρήση του εργαλείου Postman. Πραγματοποιήθηκαν δοκιμές στα εξής σημεία:

* /register και /login με ορθά και εσφαλμένα δεδομένα.
* Επαλήθευση JWT authentication.
* Προσθήκη, ανάκτηση, τροποποίηση και διαγραφή κρατήσεων μέσω των endpoints /reservations, /user/reservations.
* Εμφάνιση εστιατορίων από το /restaurants.

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

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

**6.3 Έλεγχος Βάσης Δεδομένων**

Η MariaDB ελέγχθηκε ως προς:

* Τη σωστή αποθήκευση δεδομένων χρηστών και κρατήσεων.
* Την ακεραιότητα σχέσεων μεταξύ των πινάκων (μέσω ξένων κλειδιών).
* Την επιστροφή ορθών αποτελεσμάτων με SQL queries.

**6.4 Δοκιμές από Χρήστες**

Ζητήθηκε από 3 χρήστες να χρησιμοποιήσουν την εφαρμογή σε πραγματικές συνθήκες.

* Ευχρηστία UI
* Ταχύτητα και απόκριση
* Κατανόηση των λειτουργιών. Όλες οι λειτουργίες κρίθηκαν ικανοποιητικές, ενώ δόθηκαν μικρές προτάσεις για περαιτέρω βελτίωση της εμφάνισης.

# Κεφάλαιο 7: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Η εφαρμογή ολοκληρώθηκε με επιτυχία καλύπτοντας όλες τις βασικές απαίτησης της εργασίας ωστόσο υπάρχουν κάποιες μελλοντικές επεκτάσεις που θα μπορούσαν να «τελειοποιήσουν» την εφαρμογή .

Για τις μελλοντικές επεκτάσεις προτείνονται τα εξής:

* Προσθήκη αξιολογήσεων χρηστών για τα εστιατόρια.
* Υποστήριξη συστήματος ειδοποιήσεων για υπενθυμίσεις κράτησης.
* Περισσότερα φίλτρα και κατηγοριοποιήσεις στα εστιατόρια (κουζίνα, βαθμολογία κ.ά.).
* Υποστήριξη για κράτηση μέσω χάρτη (π.χ. με Google Maps API).
* Αποθήκευση φωτογραφιών από τους χρήστες για κάθε κράτηση.
* Περισσότερες επιλογές διαχείρισης προφίλ χρήστη**.**

# Κεφάλαιο 8 : Βιβλιογραφία

<https://blog.codeanalogies.com/2017/11/03/understanding-the-basics-of-express-js/>

<https://dev.to/code_2/how-to-build-a-full-stack-restaurant-delivery-app-using-nodejs-express-reactjs-and-css-54a0>

<https://mariadb.com/get-started-with-mariadb/>

<https://www.telerik.com/blogs/introduction-json-web-tokens-jwt>