A blue and black logo

AI-generated content may be incorrect.

A logo for a college

AI-generated content may be incorrect.

**UEL Number:2471052**

**Mobile and Distributed Systems**

**24/05/2025**

**Ανάπτυξη Εφαρμογής Κράτησης Θέσεων σε Εστιατόριο μέσω Κινητής Συσκευής**

1. **Εισαγωγή**

Πλέον, η ανάπτυξη εφαρμογών για κινητές συσκευές αποτελεί αναπόσπαστο κομμάτι της καθημερινής τεχνολογικής μας εμπειρίας, καθώς πολλές από τις λειτουργίες που κάποιος θα έψαχνε στον υπολογιστή του σκοπεύουν πλέον να μεταφερθούν στα κινητά και tablet. Στην εργασία αυτή λοιπόν, σχεδιάστηκε και υλοποιήθηκε μια εφαρμογή κρατήσεων τραπεζιών σε εστιατόρια, με στόχο να συνδυαστεί η εμπειρία προγραμματισμού frontend και backend μαζί με την κατανόηση της αρχιτεκτονικής κατανεμημένων συστημάτων. Η εφαρμογή που αναπτύχθηκε επιτρέπει στους χρήστες να δημιουργούν λογαριασμό, να συνδέονται με ασφάλεια, να βλέπουν διαθέσιμα εστιατόρια, να πραγματοποιούν κρατήσεις τραπεζιών, να βλέπουν το ιστορικό τους και να τροποποιούν ή να ακυρώνουν τις κρατήσεις τους. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν σύγχρονες τεχνολογίες του κόσμου της πληροφορικής, οι οποίες είναι:

* React Native για το frontend κομμάτι.
* Node.js με Express για το backend κομμάτι.
* MariaDB για τη βάση δεδομένων.

Ο βασικός στόχος της εφαρμογής είναι η κάλυψη μιας καθημερινής ανάγκης – η διαχείριση κρατήσεων – μέσα από μια εύχρηστη, λειτουργική και ευχάριστη προς τον χρήστη εμπειρία. Η εφαρμογή ακολουθεί τη λογική της πλήρους ροής δεδομένων, ξεκινώντας από την εισαγωγή μέσω του γραφικού περιβάλλοντος και καταλήγοντας στην ασφαλή αποθήκευση και ανάκτηση των πληροφοριών από τη βάση δεδομένων. Η αλληλεπίδραση μεταξύ frontend και backend υλοποιείται μέσω RESTful API, με στόχο την αξιόπιστη και αποδοτική ανταλλαγή πληροφοριών. Ιδιαίτερη έμφαση δόθηκε στην ασφάλεια των δεδομένων, μέσω της χρήσης τεχνολογιών, όπως η κρυπτογράφηση κωδικών πρόσβασης και η υλοποίηση authentication tokens. Η συνολική αρχιτεκτονική βασίστηκε σε σύγχρονες πρακτικές ανάπτυξης λογισμικού, ώστε να εξασφαλίζεται η σταθερότητα, η ευχρηστία και η αποδοτικότητα της εφαρμογής κατά τη χρήση της.

1. **Αρχιτεκτονική Συστήματος**

Η αρχιτεκτονική της εφαρμογής αυτής βασίζεται σε Three – Tier Architecture, που περιλαμβάνει το frontend, τo backend και το database. Η ξεκάθαρα διαχωρισμένη δομή συμβάλλει στην καλύτερη οργάνωση του κώδικα, στην ευκολότερη συντήρηση και στην επεκτασιμότητα της εφαρμογής. Το frontend υλοποιήθηκε με React Native, χρησιμοποιώντας Expo CLI για τη γρήγορη ανάπτυξη σε περιβάλλον Android και iOS. Μέσω της διεπαφής αυτής, ο χρήστης μπορεί να αλληλοεπιδρά με το σύστημα, πιο συγκεκριμένα να δημιουργήσει λογαριασμό, να συνδέεται, να περιηγείται στα διαθέσιμα εστιατόρια, να πραγματοποιεί και να επεξεργάζεται κρατήσεις και να αποσυνδέεται από το profile του. Όλα τα δεδομένα ανταλλάσσονται με το backend μέσω HTTP αιτημάτων. Το backend αναπτύχθηκε με Node.js και Express.js και λειτουργεί ως μεσάζοντας μεταξύ του frontend και του Database. Παρέχει RESTful API endpoints για όλες τις βασικές λειτουργίες, όπως εγγραφή και σύνδεση χρηστών, δημιουργία και ανάκτηση κρατήσεων, αναζήτηση εστιατορίων και διαχείριση δεδομένων. Η εσωτερική δομή του backend έχει οργανωθεί με τέτοιο τρόπο ώστε να διαχωρίζει τις βασικές λειτουργίες της εφαρμογής, όπως η επεξεργασία αιτημάτων, η επικύρωση χρηστών και η διαχείριση δεδομένων. Αυτό κάνει πιο εύκολη την ανάγνωση και την συντήρηση του κώδικα. Επιπλέον, για την προστασία των προσωπικών δεδομένων των χρηστών, εφαρμόζεται κρυπτογράφηση κωδικών με τη χρήση της βιβλιοθήκης bcrypt, ενώ η διαδικασία ταυτοποίησης χρηστών υλοποιείται μέσω tokens JWT, εξασφαλίζοντας ασφαλή επικοινωνία μεταξύ πελάτη και διακομιστή. Η βάση δεδομένων υλοποιήθηκε με MariaDB, και διαχειρίζεται τρεις βασικούς πίνακες: χρήστες (users), εστιατόρια (restaurants) και κρατήσεις (reservations).A screenshot of a menu

AI-generated content may be incorrect.

Οι σχέσεις ανάμεσα στους πίνακες υλοποιούνται με τη χρήση ξένων κλειδιών, διασφαλίζοντας την ακεραιότητα των δεδομένων. Η επικοινωνία του backend με τη βάση δεδομένων υλοποιείται με τη βοήθεια της βιβλιοθήκης mysql2 η οποία επιτρέπει την εκτέλεση ερωτημάτων SQL μέσα από τον Node.js server. Μέσω αυτής της σύνδεσης πραγματοποιείται η αποθήκευση, ανάκτηση και επεξεργασία όλων των δεδομένων της εφαρμογής, όπως χρήστες, εστιατόρια και κρατήσεις.



Συνολικά, η αρχιτεκτονική εξασφαλίζει την ανεξαρτησία των επιπέδων, τη Modular υλοποίηση και την αποδοτική ανταλλαγή δεδομένων μεταξύ client και server.

1. **Περιγραφή Λειτουργιών**

Η εφαρμογή περιλαμβάνει βασικές λειτουργίες που υποστηρίζουν την πλήρη ροή αλληλεπίδρασης του χρήστη, από την δυνατότητα δημιουργίας λογαριασμού μέχρι και την επεξεργασία κρατήσεων. Κάθε λειτουργία που είναι ενσωματωμένη στην εφαρμογή συνδυάζει την διεπαφή χρήστη, αιτήματα προς το backend και αλληλεπίδραση με τη βάση δεδομένων.

* 1. **Εγγραφή Χρήστη**

Ο χρήστης μπορεί να δημιουργήσει λογαριασμό χρησιμοποιώντας το όνομα του, το email του και έναν κωδικό πρόσβασης. Τα στοιχεία στέλνονται στο backend μέσω αιτήματος POST στο endpoint /register. Ο κωδικός κρυπτογραφείται με bcrypt και αποθηκεύεται στη βάση δεδομένων, πιο συγκεκριμένα στον πίνακα users. Αν το email αντιστοιχεί είδη σε κάποιον λογαριασμό, τότε το backend επιστρέφει αντίστοιχο μήνυμα σφάλματος.A screenshot of a phone

AI-generated content may be incorrect.

* 1. **Σύνδεση Χρήστη**

Στη φόρμα σύνδεσης ο χρήστης συμπληρώνει τα στοιχεία τα οποία χρησιμοποίησε για να δημιουργήσει τον λογαριασμό του. Με επιτυχές POST αίτημα στο /login, ο server επαληθεύει τα στοιχεία, δημιουργεί authentication token (JWT) και το επιστρέφει στον client. Το token αποθηκεύεται τοπικά στο frontend και χρησιμοποιείται για κάθε προστατευμένο endpoint.

A screenshot of a login box

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

* 1. **Προβολή Λίστας Εστιατορίων**

Όταν ο χρήστης συνδεθεί στην εφαρμογή, το πρώτο πράγμα που θα αντικρίσει είναι η λίστα των διαθέσιμων εστιατορίων, η οποία ανακτάται μέσω GET αιτήματος στο /restaurants. Το frontend επιτρέπει επίσης αναζήτηση με φίλτρα βάσει τοποθεσίας ή ονόματος. Τα εστιατόρια εμφανίζονται σε μορφή λίστας, με τα βασικά χαρακτηριστικά τους, όπως όνομα, τοποθεσία η ακόμα και το φαγητό το οποίο εμπεριέχουν.

A screenshot of a menu

AI-generated content may be incorrect.A close-up of a document

AI-generated content may be incorrect.A screen shot of a document

AI-generated content may be incorrect.

**3.4 Δημιουργία Κράτησης**

Ο χρήστης μπορεί να πραγματοποιήσει κράτηση σε οποίο από τα διαθέσιμα εστιατόρια θέλει, μόνο συμπληρώνοντας ημερομηνία, ώρα και αριθμό ατόμων. Τα στοιχεία αποστέλλονται στο endpoint /reservations με μέθοδο POST. Έπειτα, το backend αποθηκεύει την κράτηση στον πίνακα reservations, συνδέοντάς την με τον αντίστοιχο χρήστη και εστιατόριο μέσω των αντίστοιχων user\_id και restaurant\_id.

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a cell phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

* 1. **Προβολή Ιστορικού Κρατήσεων**

Ο χρήστης μπορεί να δει τις κρατήσεις που έχει κάνει πατώντας το κουμπί ‘Οι κρατήσεις μου’ ,το οποίο θα τον μεταφέρει στην αντίστοιχη σελίδα. Η λίστα ανακτάται μέσω του endpoint /user/reservations, το οποίο επιστρέφει κρατήσεις ταξινομημένες με βάση την ημερομηνία και την ώρα. Όσον αφορά την προστασία των δεδομένων, η πρόσβαση απαιτεί έγκυρο JWT token.

A white paper with black text

AI-generated content may be incorrect.

* 1. **Τροποποίηση ή Ακύρωση Κράτησης**

Τέλος, ο χρήστης έχει τη δυνατότητα να ακυρώσει ή να τροποποιήσει μια κράτηση από το ιστορικό του. Για την ακύρωση χρησιμοποιείται μέθοδος DELETE στο endpoint /reservations/:id, ενώ για την τροποποίηση χρησιμοποιείτε μέθοδος PUT στο ίδιο endpoint με ενημερωμένα δεδομένα. Το backend ελέγχει την εγκυρότητα του token και ενημερώνει ανάλογα τη βάση δεδομένων.

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

* 1. **Αποσύνδεση Χρήστη**

Η εφαρμογή παρέχει επίσης τη δυνατότητα αποσύνδεσης του χρήστη από τον λογαριασμό του. Η αποσύνδεση πραγματοποιείται αποκλειστικά στο frontend, καθώς το authentication token (JWT) αποθηκεύεται τοπικά στη συσκευή. Με την επιλογή "Αποσύνδεση", το token διαγράφεται από το AsyncStorage, με αποτέλεσμα ο χρήστης να χάνει την πρόσβαση σε προστατευμένα endpoints μέχρι να συνδεθεί ξανά. Μετά την αποσύνδεση, ο χρήστης μεταφέρεται αυτόματα στην οθόνη σύνδεσης εξασφαλίζοντας ότι δεν μπορεί να εκτελέσει ενέργειες χωρίς έγκυρη ταυτοποίηση. Το κουμπί αποσύνδεσης βρίσκετε μέσα από την οθόνη προφίλ.

A white background with red text

AI-generated content may be incorrect.

1. **Τεχνική Υλοποίηση**

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε με Open Source τεχνολογίες κώδικα, οι οποίες επιλέχθηκαν επειδή είναι εύχρηστες, συνεργάζονται καλά μεταξύ τους και διευκολύνουν την ανάπτυξη και τη δοκιμή της εφαρμογής.

* 1. **Frontend**

Το frontend υλοποιήθηκε με React Native και χρησιμοποιώντας το εργαλείο Expo CLI. Επιπλέον, η εφαρμογή υποστηρίζει preview σε κινητές συσκευές μέσω QR code και αναπτύχθηκε με Visual Studio Code. Η εφαρμογή ανταλλάσσει δεδομένα με τον server μέσω HTTP χρησιμοποιώντας fetch. Όταν ο χρήστης συνδεθεί, λαμβάνει ένα token (JWT), το οποίο αποθηκεύεται στη συσκευή ώστε να μπορεί να χρησιμοποιείται αυτόματα στις επόμενες ενέργειες.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a bar code

AI-generated content may be incorrect.

* 1. **Backend**

Το backend της εφαρμογής υλοποιήθηκε με Node.js και το πλαίσιο με Express.js . Παρέχει μια σειρά από API endpoints, όπως /register, /login, /restaurants και /reservations, τα οποία χειρίζονται τις βασικές λειτουργίες της εφαρμογής. Για τη διαδικασία ταυτοποίησης των χρηστών χρησιμοποιείται το πρότυπο JWT, ενώ οι κωδικοί πρόσβασης αποθηκεύονται με ασφάλεια μέσω κρυπτογράφησης, χρησιμοποιώντας τη βιβλιοθήκη bcrypt.js.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **Database**

Η βάση δεδομένων δημιουργήθηκε μέσω phpMyAdmin και περιλαμβάνει τρεις πίνακες: users, restaurants, reservations, με ξένους κλειδιά που διασφαλίζουν σχέσεις και ακεραιότητα. Ο πίνακας restaurants αποθηκεύει πληροφορίες για κάθε εστιατόριο, όπως όνομα, τοποθεσία και περιγραφή, ενώ ο πίνακας reservations καταγράφει κάθε κράτηση, συνδέοντάς την με τον αντίστοιχο χρήστη και εστιατόριο. Οι σχέσεις μεταξύ των πινάκων ορίζονται μέσω πρωτεύοντων και ξένων κλειδιών, ώστε να διασφαλίζεται η συνέπεια των δεδομένων και να αποτρέπεται η αποθήκευση μη έγκυρων ή ασύνδετων εγγραφών.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **Χρήση Εργαλείων και Testing**

Η εφαρμογή δοκιμάστηκε με το εργαλείο Postman, μέσω του οποίου ελέγχθηκαν όλα τα βασικά API endpoints, όπως η εγγραφή, η σύνδεση και η δημιουργία κρατήσεων. Επίσης, επιβεβαιώθηκε η σωστή λειτουργία του authentication με JWT tokens. Η ανάπτυξη έγινε στο Visual Studio Code, το οποίο χρησιμοποιήθηκε για τη συγγραφή και την οργάνωση του κώδικα της εφαρμογής. Για τη βάση δεδομένων χρησιμοποιήθηκε η MariaDB, η οποία εγκαταστάθηκε μέσω του πακέτου XAMPP. Η δημιουργία και η διαχείριση των πινάκων και των εγγραφών πραγματοποιήθηκαν εύκολα μέσω του περιβάλλοντος του phpMyAdmin.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Προβλήματα και Τρόποι Αντιμετώπισης**

Κατά την υλοποίηση της εφαρμογής προέκυψαν διάφορες τεχνικές δυσκολίες, τόσο σε επίπεδο συντονισμού των τεχνολογιών, όσο και σε λεπτομέρειες υλοποίησης. Μέσα από τη διαδικασία επίλυσης των προβλημάτων, έγινε πιο κατανοητός ο τρόπος με τον οποίο συνεργάζονται τα εργαλεία και τα επιμέρους τμήματα του συστήματος. Ένα από τα πρώτα προβλήματα που εμφανίστηκαν ήταν η αποτυχία εκκίνησης του backend μέσω Node.js, λόγω της αποτυχίας της προσθήκης του Node στο περιβάλλον PATH. Το πρόβλημα εντοπίστηκε κατά την προσπάθεια εκτέλεσης του αρχείου server.js, και επιλύθηκε μέσω προσθήκης του Path εγκατάστασης στις ρυθμίσεις του συστήματος. Παρόμοια δυσκολία παρουσιάστηκε με την διασύνδεση του backend και της βάσης δεδομένων. όπου εμφανίστηκε μήνυμα σφάλματος ‘Cannot connect: invalid settings.’ Το πρόβλημα αυτό παρουσιάστηκε διότι η MySQL στον Xampp δεν είχε ξεκινήσει. Ωστόσο, με την ενεργοποίηση της μέσω του XAMPP Control Panel, το πρόβλημα διορθώθηκε άμεσα. Μετέπειτα εντοπίστηκαν προβλήματα στο Postman κατά την διαδικασία δοκίμων, κυρίως λόγω λανθασμένης μορφοποίησης ή επειδή έλειπαν απαραίτητα δεδομένα από τις αιτήσεις. Σε ορισμένες περιπτώσεις, δεν είχε δηλωθεί σωστά ο τύπος του περιεχομένου (Content - Type: application/json), ενώ σε άλλες δεν είχαν συμπληρωθεί όλα τα απαιτούμενα πεδία στο σώμα της αίτησης. Τα προβλήματα λυθήκαν με ρυθμίσεις στους headers και με την αποστολή πλήρων και σωστά δομημένων δεδομένων στο backend. Ένα ακόμη σημαντικό θέμα ήταν να γίνει επαλήθευση του JWT Token στο frontend. Αρχικά, δεν γινόταν να προστεθεί σωστά στο Authorization header με το πρόθεμα Bearer, το οποίο είχε ως αποτέλεσμα οι προστατευμένες διαδρομές να απορρίπτονται. Το πρόβλημα λύθηκε με τον κατάλληλο έλεγχο και αποθήκευση του token στο AsyncStorage και χρήση του στο fetch με τα σωστά headers. Τέλος, κατά τη δημιουργία ή την τροποποίηση κρατήσεων υπήρξαν προβλήματα που σχετίζονταν είτε με ελλιπή δεδομένα είτε με λανθασμένη μορφοποίηση της ημερομηνίας και ώρας. Τα προβλήματα αυτά εντοπίστηκαν μέσω των μηνυμάτων του backend και επιλύθηκαν με κατάλληλους ελέγχους και επικυρώσεις τόσο κατά την αποστολή των δεδομένων από το frontend όσο και κατά την επεξεργασία τους από τον server. Η επίλυση αυτών των προβλημάτων βοήθησε να γίνει πιο κατανοητός ο τρόπος συνεργασίας μεταξύ του frontend, του backend και της βάσης δεδομένων μέσα στην εφαρμογή.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Συμπέρασμα**

Η ανάπτυξη της εφαρμογής αυτής βοήθησε στην κατανόηση του πώς συνεργάζονται το frontend, το backend και η βάση δεδομένων. Κατά τη διάρκεια της ανάπτυξης και της υλοποίησης, έγινε σαφές ο τρόπος που μεταφέρονται και επεξεργάζονται τα δεδομένα από τον χρήστη μέχρι τον server και τη βάση, καθώς και πόσο σημαντικός είναι ο σωστός έλεγχος σε κάθε στάδιο. Η χρήση εργαλείων όπως το Postman, το phpMyAdmin και το Visual Studio Code διευκόλυνε την ανάπτυξη και τις δοκιμές. Μέσα από τις δυσκολίες που υπήρξαν κατά την ανάπτυξη, υπήρξε η ευκαιρία να αντιμετωπιστούν προβλήματα στην πράξη και να αποκτηθεί καλύτερη εμπειρία στον έλεγχο σφαλμάτων, την αποστολή αιτημάτων και τη συνεργασία των τμημάτων της εφαρμογής. Ακολουθώντας την εκφώνηση, η εφαρμογή αυτή επιτρέπει στους χρήστες να δημιουργούν λογαριασμό, να συνδέονται, να βλέπουν διαθέσιμα εστιατόρια, να κάνουν κρατήσεις και να διαχειρίζονται το ιστορικό τους. Είναι αξιοσημείωτο κάπου εδώ ότι όλα αυτά λειτουργούν μέσα από ένα απλό και καθαρό περιβάλλον χρήστη, που ο οποιοδήποτε θα καταλάβαινε. Συνοπτικά, η εφαρμογή ολοκληρώθηκε με επιτυχία και κατάφερε να καλύψει όλους τους βασικούς στόχους που είχαν τεθεί από την αρχή.