



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΓΙΑ ΤΗΝ ΔΙΑΧΕΙΡΙΣΗ ΚΡΑΤΗΣΕΩΝ ΕΣΤΙΑΤΟΡΙΩΝ

ΠΑΡΑΣΚΕΥΟΠΟΥΛΟΥ ΧΡΙΣΤΙΝΑ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΤΖΙΑΛΛΑΣ ΓΡΗΓΟΡΙΟΣ
ΚΑΘΗΓΗΤΗΣ

Λαμία 22/06 έτος 2025



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

ONLINE APPLICATION FOR MANAGING RESERVATIONS FOR RESTAURANTS

PARASKEVOPOULOU CHRISTINA

FINAL THESIS

ADVISOR

TZIALLAS GRIGORIOS

Lamia 22/06 year 2025

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων Συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 22/06/2025

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως αντικείμενο την ανάπτυξη μιας διαδικτυακής εφαρμογής για τη διαχείριση κρατήσεων σε εστιατόρια. Σκοπός της εφαρμογής είναι να παρέχει έναν εύκολο, γρήγορο και φιλικό τρόπο για τους πελάτες να πραγματοποιούν κρατήσεις τραπεζιών μέσω διαδικτύου, ενώ ταυτόχρονα προσφέρει στους ιδιοκτήτες τη δυνατότητα παρακολούθησης και διαχείρισης των κρατήσεων.

Η εφαρμογή αναπτύχθηκε με τη χρήση τεχνολογιών frontend (HTML, CSS, JavaScript) και backend (Node.js με Express.js), ενώ για την αποθήκευση των δεδομένων χρησιμοποιήθηκε η βάση δεδομένων PostgreSQL. Ο χρήστης μπορεί να επιλέξει πόλη, εστιατόριο, ημερομηνία και ώρα, και να συμπληρώσει προσωπικά στοιχεία για την ολοκλήρωση της κράτησης. Οι κρατήσεις αποθηκεύονται στη βάση και συσχετίζονται με διαθέσιμα τραπέζια, ανάλογα με τον αριθμό ατόμων. Οι ιδιοκτήτες μπορούν να συνδεθούν με ειδικό λογαριασμό και να δουν συγκεντρωτικά όλες τις κρατήσεις, μαζί με πληροφορίες για τη διαθεσιμότητα των τραπεζιών.

Η εργασία αναλύει τα στάδια υλοποίησης της εφαρμογής, τις τεχνολογίες που αξιοποιήθηκαν, τις προκλήσεις που προέκυψαν κατά την ανάπτυξη και τις δυνατότητες μελλοντικής εξέλιξης, όπως η ενσωμάτωση τεχνητής νοημοσύνης για έξυπνες προτάσεις και προβλέψεις ζήτησης.

ABSTRACT

This thesis focuses on the development of a web-based application for managing restaurant reservations. The aim of the application is to provide an easy, fast, and user-friendly way for customers to book tables online, while also offering restaurant owners the ability to monitor and manage their reservations effectively.

The application was developed using modern frontend technologies (HTML, CSS, JavaScript) and a backend based on Node.js with Express.js. Data is stored in a PostgreSQL database. Users can select a city, restaurant, date, and time, and complete their reservation by filling in their personal details. Each reservation is stored in the database and linked to available tables according to the number of guests. Restaurant owners can log in with a dedicated account and view all reservations, along with table availability.

This thesis analyzes the development stages of the application, the technologies used, the challenges encountered during the implementation, and future improvement opportunities such as the integration of artificial intelligence for intelligent suggestions and demand prediction.

Table of Contents

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	III
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....	2
(1.1 ΓΕΝΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΘΕΜΑΤΟΣ).....	2
(1.2 ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΤΗΣ ΠΤΥΧΙΑΚΗΣ).....	2
ΚΕΦΑΛΑΙΟ 2 ΕΦΑΡΜΟΓΕΣ ΚΡΑΤΗΣΕΩΝ.....	3
(2.1 ΕΙΔΗ ΕΦΑΡΜΟΓΩΝ ΚΡΑΤΗΣΕΩΝ).....	3
(2.2 ΛΕΙΤΟΥΡΓΙΕΣ ΚΑΙ ΔΥΝΑΤΟΤΗΤΕΣ ΤΩΝ ΕΦΑΡΜΟΓΩΝ ΚΡΑΤΗΣΕΩΝ).....	4
(2.3 ΑΝΑΛΥΣΗ ΑΝΑΓΚΩΝ ΧΡΗΣΤΩΝ ΚΑΙ ΙΔΙΟΚΤΗΤΩΝ ΕΣΤΙΑΤΟΡΙΩΝ).....	6
(2.4 ΧΡΗΣΗ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ & ΠΡΟΤΑΣΕΙΣ ΜΕ ΒΑΣΗ ΣΥΜΠΕΡΙΦΟΡΑ ΧΡΗΣΤΗ)	8
ΚΕΦΑΛΑΙΟ 3 ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ ΓΙΑ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ	10
(3.1 ΤΕΧΝΟΛΟΓΙΕΣ FRONTEND & BACKEND ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ)	10
(3.2 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ).....	11
(3.3 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ).....	11
(3.4 ΔΙΑΧΕΙΡΙΣΗ ΤΑΥΤΟΤΗΤΑΣ ΚΑΙ ΕΞΟΥΣΙΟΔΟΤΗΣΗ ΧΡΗΣΤΩΝ)	12
ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	12
(4.1 ΣΚΟΠΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ).....	12
(4.2 ΔΙΑΔΙΚΑΣΙΑ ΚΡΑΤΗΣΕΩΝ)	13
(4.3 ΠΑΡΟΥΣΙΑΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟΥ ΚΩΔΙΚΑ).....	16
ΚΕΦΑΛΑΙΟ 5 ΕΠΙΔΕΙΞΗ ΕΦΑΡΜΟΓΗΣ.....	23
(5.1 ΠΕΡΙΓΡΑΦΗ ΔΙΑΔΙΚΑΣΙΑΣ)	23
(5.2 ΔΥΣΚΟΛΙΕΣ & ΛΥΣΕΙΣ)	23
(5.3 ΣΧΟΛΙΑ ΓΙΑ ΤΗΝ ΑΠΟΔΟΣΗ ΚΑΙ ΕΠΕΚΤΑΣΙΜΟΤΗΤΑ)	23
ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	24

(6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ)	
.....	24
(6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ)	
.....	25
ΒΙΒΛΙΟΓΡΑΦΙΑ	26

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 Γενική Παρουσίαση του Θέματος

Τα τελευταία χρόνια, η ραγδαία ανάπτυξη των τεχνολογιών πληροφορικής και επικοινωνιών έχει επηρεάσει σημαντικά κάθε πτυχή της καθημερινότητας. Ο τομέας της εστίασης δεν θα μπορούσε να μείνει ανεπηρέαστος από τη μετάβαση στην ψηφιακή εποχή, η οποία επιτάσσει την ανάγκη για ευέλικτα, εύχρηστα και προσβάσιμα συστήματα κρατήσεων. Η χρήση διαδικτυακών εργαλείων για τη διαχείριση τραπεζιών, την καταγραφή κρατήσεων και την αλληλεπίδραση με τους πελάτες αποτελεί πλέον αναπόσπαστο μέρος της σύγχρονης επιχειρηματικότητας στον χώρο της εστίασης.

Η παρούσα πτυχιακή εργασία επικεντρώνεται στην ανάπτυξη μιας ολοκληρωμένης διαδικτυακής εφαρμογής κράτησης τραπεζιών για εστιατόρια. Στόχος είναι η παροχή μιας εύχρηστης και λειτουργικής πλατφόρμας η οποία εξυπηρετεί τόσο τους πελάτες όσο και τους ιδιοκτήτες των επιχειρήσεων. Μέσω της εφαρμογής, ο πελάτης έχει τη δυνατότητα να πραγματοποιεί κράτηση online, χωρίς την ανάγκη τηλεφωνικής επικοινωνίας ή φυσικής παρουσίας στο κατάστημα. Ταυτόχρονα, ο ιδιοκτήτης του εστιατορίου μπορεί να έχει πρόσβαση σε συγκεντρωτικά δεδομένα κρατήσεων, να διαχειρίζεται τη διαθεσιμότητα των τραπεζιών και να βελτιστοποιεί τη λειτουργία της επιχείρησής του.

Η εφαρμογή αναπτύχθηκε με τη χρήση μοντέρνων τεχνολογιών frontend (HTML, CSS, JavaScript) και backend (Node.js με Express.js), ενώ για την αποθήκευση των δεδομένων χρησιμοποιείται σύστημα βάσης δεδομένων PostgreSQL. Το περιβάλλον χρήστη σχεδιάστηκε με έμφαση στη φιλικότητα, την ευκολία χρήσης και την αισθητική απλότητα, ώστε να είναι προσιτό σε κάθε είδους χρήστη.

Μέσα από την παρούσα εργασία αναλύεται βήμα προς βήμα η διαδικασία σχεδιασμού και υλοποίησης της εφαρμογής, παρουσιάζονται οι τεχνολογικές επιλογές που υιοθετήθηκαν, και καταγράφονται οι δυνατότητες μελλοντικής επέκτασης του συστήματος, όπως η εισαγωγή τεχνητής νοημοσύνης για εξατομικευμένες προτάσεις ή η δημιουργία mobile εφαρμογής.

Ο βασικός σκοπός της πτυχιακής εργασίας είναι η δημιουργία μιας σύγχρονης, λειτουργικής και αξιόπιστης διαδικτυακής εφαρμογής η οποία θα καλύπτει τις ανάγκες των πελατών και των ιδιοκτητών εστιατορίων στον τομέα των κρατήσεων. Η εφαρμογή αυτή φιλοδοξεί να αποτελέσει μια αποτελεσματική εναλλακτική στις παραδοσιακές μεθόδους διαχείρισης κρατήσεων (όπως το τηλέφωνο ή το χειρόγραφο ημερολόγιο), προάγοντας τη χρήση τεχνολογίας στην καθημερινή λειτουργία των επιχειρήσεων εστίασης.

Οι επιμέρους στόχοι της εργασίας περιλαμβάνουν:

Καταγραφή και ανάλυση των αναγκών που έχουν οι χρήστες και οι επιχειρήσεις στον τομέα των κρατήσεων.

Σχεδίαση της αρχιτεκτονικής του συστήματος και δημιουργία σχεσιακής βάσης δεδομένων.

Υλοποίηση του frontend και του backend, με έμφαση στη σωστή ροή πληροφοριών και την εμπειρία χρήστη.

Ανάπτυξη διεπαφής διαχείρισης κρατήσεων από την πλευρά των ιδιοκτητών, με έλεγχο προσβασιμότητας μέσω login.

Διασφάλιση ευελιξίας και επεκτασιμότητας, ώστε να μπορούν να προστεθούν νέα χαρακτηριστικά στο μέλλον.

Τεκμηρίωση, παρουσίαση και αξιολόγηση της λειτουργίας της εφαρμογής μέσα από πραγματικά σενάρια χρήσης.

Μέσα από την ολοκλήρωση της εργασίας, επιδιώκεται όχι μόνο η τεχνική υλοποίηση ενός συστήματος, αλλά και η απόκτηση εμπειρίας στη σχεδίαση διαδικτυακών εφαρμογών, στη συνεργασία μεταξύ client και server, και στη δημιουργία ενός ψηφιακού προϊόντος με πρακτική αξία.

ΚΕΦΑΛΑΙΟ 2 ΕΦΑΡΜΟΓΕΣ ΚΡΑΤΗΣΕΩΝ

2.1 Είδη εφαρμογών κρατήσεων

Οι εφαρμογές κρατήσεων αποτελούν αναπόσπαστο κομμάτι της σύγχρονης ψηφιακής καθημερινότητας, καθώς διευκολύνουν την πρόσβαση των χρηστών σε υπηρεσίες και βελτιστοποιούν τη διαχείριση της διαθεσιμότητας για τις επιχειρήσεις. Η χρήση τους έχει εδραιωθεί τόσο σε τομείς εξειδικευμένων υπηρεσιών όσο και στον χώρο της μαζικής εστίασης.

Τα βασικότερα είδη εφαρμογών κρατήσεων περιλαμβάνουν:

- **Εφαρμογές εστίασης:** Επιτρέπουν την κράτηση τραπεζιού σε συγκεκριμένο χώρο, ημερομηνία και ώρα. Συχνά συνδέονται με πραγματικό αριθμό διαθέσιμων τραπεζιών, δυνατότητες πολλαπλών εστιατορίων και ενοποιημένες πλατφόρμες (π.χ. e-table, OpenTable).
- **Εφαρμογές διαμονής (ξενοδοχεία, Airbnb):** Αφορούν κυρίως την κράτηση δωματίων με δυνατότητα προβολής τιμών, παροχών, αξιολογήσεων και άμεσης πληρωμής.
- **Εφαρμογές υγείας και ευεξίας:** Διαχείριση ραντεβού σε ιατρικά κέντρα, κομμωτήρια, spa, κ.ά., με χρονοπρογραμματισμό και υπενθυμίσεις.
- **Εφαρμογές μετακίνησης:** Όπως κρατήσεις αεροπορικών, ακτοπλοϊκών και σιδηροδρομικών εισιτηρίων.

Η εφαρμογή που αναπτύσσεται στο πλαίσιο αυτής της πτυχιακής ανήκει στην κατηγορία **κρατήσεων εστίασης**, με έμφαση στη λειτουργικότητα, την απλότητα και την αμεσότητα τόσο για τον πελάτη όσο και για τον επιχειρηματία.

2.2 Λειτουργίες και δυνατότητες των εφαρμογών κρατήσεων

Οι εφαρμογές κρατήσεων, ανεξαρτήτως κλάδου, έχουν ως κύριο στόχο τη **διευκόλυνση της επικοινωνίας** μεταξύ πελάτη και παρόχου υπηρεσιών, μέσω μιας αυτοματοποιημένης και φιλικής προς τον χρήστη διεπαφής. Η αποτελεσματικότητα μιας τέτοιας εφαρμογής εξαρτάται σε μεγάλο βαθμό από το σύνολο των λειτουργιών που προσφέρει, καθώς και από τη δυνατότητά της να καλύψει τις ανάγκες και των δύο πλευρών: του τελικού χρήστη (πελάτη) και του ιδιοκτήτη/διαχειριστή της επιχείρησης.

1. Δυνατότητα επιλογής περιοχής/τοποθεσίας

Μια βασική λειτουργία των εφαρμογών κρατήσεων είναι η δυνατότητα **φιλτραρίσματος ανά γεωγραφική περιοχή**, επιτρέποντας στον χρήστη να επιλέγει πρώτα την πόλη ή περιοχή που τον ενδιαφέρει και στη συνέχεια να βλέπει μόνο τα διαθέσιμα εστιατόρια που αντιστοιχούν σε αυτήν. Αυτό αυξάνει σημαντικά τη χρηστικότητα και περιορίζει το εύρος των επιλογών, διευκολύνοντας την πλοήγηση.

Στην παρούσα εφαρμογή Book&Bite, ο χρήστης μπορεί να επιλέξει μεταξύ τεσσάρων βασικών πόλεων (Πάτρα, Αθήνα, Θεσσαλονίκη, Λάρισα), με τα εστιατόρια να εμφανίζονται δυναμικά ανάλογα με την επιλογή.

2. Εμφάνιση διαθέσιμων εστιατορίων με πληροφορίες

Η κάθε επιλογή εστιατορίου συνοδεύεται από **αναλυτική προεπισκόπηση** που περιλαμβάνει:

- Εικόνα του χώρου
- Τύπο κουζίνας (π.χ. Ιταλική, Μεσογειακή)
- Ώρες λειτουργίας
- Διεύθυνση
- Μέσο εύρος τιμών (€ έως €€€)
- Αξιολόγηση σε μορφή αστεριών (★)

Η πληροφορία αυτή βοηθά τον χρήστη να κάνει πιο ενημερωμένη επιλογή πριν προχωρήσει στην κράτηση.

3. Επιλογή ημερομηνίας & ώρας βάσει διαθεσιμότητας

Η λειτουργία επιλογής ημερομηνίας είναι δυναμική και **περιορίζεται από το παρόν** (δεν μπορεί να επιλεγεί ημερομηνία παρελθόντος). Η επιλογή ώρας γίνεται με δύο dropdown (ώρες και λεπτά) και προσαρμόζεται **ανάλογα με το ωράριο του κάθε εστιατορίου**.

Για παράδειγμα, ένα εστιατόριο με ωράριο 14:00 – 23:00 εμφανίζει μόνο αντίστοιχες επιλογές στο dropdown της ώρας, ενώ απενεργοποιούνται οι υπόλοιπες.

4. Καταχώρηση στοιχείων κράτησης (φόρμα εισαγωγής)

Ο χρήστης καλείται να συμπληρώσει φόρμα με:

- Ονοματεπώνυμο
- Email επικοινωνίας
- Αριθμό ατόμων
- Προαιρετική περίπτωση (γενέθλια, επέτειος κ.λπ.)

Η φόρμα είναι υποχρεωτική για την ολοκλήρωση της κράτησης και εφαρμόζονται **έλεγχοι εγκυρότητας** (validation) στα πεδία, όπως απαιτούμενα στοιχεία και έλεγχος για email.

5. Υποβολή και καταγραφή κράτησης

Κατά την υποβολή της φόρμας:

- Τα δεδομένα αποστέλλονται στο backend μέσω HTTP POST αιτήματος (fetch).

- Ο server ελέγχει τα στοιχεία, αναζητά κατάλληλο τραπέζι με βάση τη χωρητικότητα και **καταχωρεί την κράτηση στη βάση δεδομένων.**
- Δημιουργείται **μοναδικός κωδικός κράτησης** (π.χ. AG49X7TF), ο οποίος αποστέλλεται ως επιβεβαίωση στον χρήστη.

Αυτή η διαδικασία εξασφαλίζει οργανωμένη αποθήκευση και εύκολη αναγνώριση κρατήσεων στο σύστημα.

6. Προβολή κράτησης στον χρήστη – Επιβεβαίωση

Ο χρήστης βλέπει άμεσα μήνυμα επιβεβαίωσης που περιλαμβάνει:

- Ημερομηνία & ώρα κράτησης
- Εστιατόριο
- Περίσταση (αν έχει δηλωθεί)
- Αριθμός ατόμων
- Μοναδικός κωδικός

Αυτό προσφέρει αίσθημα ασφάλειας και επαγγελματισμού, ενώ μπορεί να προστεθεί στο μέλλον αποστολή μέσω email ή SMS.

7. Δυνατότητα εισόδου ιδιοκτήτη (login)

Ο διαχειριστής/ιδιοκτήτης του εστιατορίου έχει τη δυνατότητα να **συνδεθεί μέσω ξεχωριστής φόρμας login**. Στην τρέχουσα υλοποίηση, υπάρχει ένας προκαθορισμένος λογαριασμός με username και password, ο οποίος ενεργοποιεί την πρόσβαση στο **πίνακα διαχείρισης κρατήσεων**.

8. Προβολή πινάκων κρατήσεων με στοιχεία

Αφού ο ιδιοκτήτης συνδεθεί επιτυχώς, του εμφανίζεται **πίνακας κρατήσεων** με τις εξής πληροφορίες:

- Ονοματεπώνυμο πελάτη
- Ημερομηνία & ώρα
- Αριθμός ατόμων
- Περίσταση
- Κωδικός κράτησης
- Όνομα εστιατορίου
- Πόσα τραπέζια με αυτή τη χωρητικότητα είναι διαθέσιμα

Η πληροφορία αυτή βοηθά τον επιχειρηματία να γνωρίζει την πληρότητα, να κάνει προγραμματισμό προσωπικού και να εντοπίζει εύκολα τις ώρες αιχμής.

Για τη σωστή ανάπτυξη μιας διαδικτυακής εφαρμογής κρατήσεων, είναι θεμελιώδες να προηγηθεί η **ανάλυση των αναγκών των τελικών χρηστών και των επιχειρηματιών**. Η κατανόηση του τρόπου με τον οποίο χρησιμοποιούν την υπηρεσία, των δυσκολιών που αντιμετωπίζουν και των προσδοκιών που έχουν, αποτελεί το βασικό υπόβαθρο για την υλοποίηση μιας λειτουργικής και αποδοτικής εφαρμογής.

Η εφαρμογή **Book&Bite** σχεδιάστηκε με διπλή στόχευση: την κάλυψη των αναγκών των **πελατών** που θέλουν να πραγματοποιήσουν μια κράτηση και των **ιδιοκτητών εστιατορίων** που επιθυμούν να διαχειρίζονται αποτελεσματικά τη λειτουργία του καταστήματός τους.

2.3.1 Ανάγκες Χρηστών

Οι πελάτες που χρησιμοποιούν μια εφαρμογή κρατήσεων αναζητούν ευκολία, διαφάνεια και άμεση ανταπόκριση. Συγκεκριμένα:

- Ευκολία στη χρήση
Οι χρήστες θέλουν να ολοκληρώνουν την κράτηση γρήγορα και χωρίς περιττά βήματα. Η διεπαφή πρέπει να είναι ξεκάθαρη, κατανοητή και να μη δημιουργεί σύγχυση ή απογοήτευση κατά την πλοήγηση.
- Διαθεσιμότητα πληροφοριών πριν την κράτηση
Προτιμούν να βλέπουν βασικά χαρακτηριστικά του εστιατορίου: τύπο κουζίνας, ωράριο, τιμές, τοποθεσία και αξιολογήσεις, προτού προχωρήσουν.
- Δυνατότητα επιλογής ημερομηνίας και ώρας βάσει διαθεσιμότητας
Ο χρήστης επιθυμεί να γνωρίζει πότε μπορεί να πάει στο εστιατόριο και ποιες ώρες είναι διαθέσιμες, αποφεύγοντας απορρίψεις κρατήσεων σε δεύτερο χρόνο.
- Επιβεβαίωση κράτησης
Μία άμεση και σαφής επιβεβαίωση (ιδανικά με μοναδικό κωδικό κράτησης) ενισχύει την εμπιστοσύνη στην πλατφόρμα και δημιουργεί επαγγελματική εικόνα.
- Χρήση από κινητά και tablets
Η εφαρμογή πρέπει να είναι προσβάσιμη από φορητές συσκευές, καθώς ένα μεγάλο ποσοστό των χρηστών χρησιμοποιεί smartphone για αναζήτηση και κράτηση.
- Ασφάλεια προσωπικών δεδομένων
Οι χρήστες πρέπει να αισθάνονται ότι τα στοιχεία τους (email, όνομα κ.λπ.) είναι ασφαλή και δεν θα χρησιμοποιηθούν καταχρηστικά.

Οι ιδιοκτήτες ή διαχειριστές εστιατορίων αντιμετωπίζουν καθημερινά την πρόκληση του **προγραμματισμού, της οργάνωσης τραπεζιών και της αποφυγής λαθών**, ειδικά σε ώρες αιχμής. Μια ψηφιακή εφαρμογή κρατήσεων μπορεί να συμβάλει καθοριστικά στην οργάνωση του καταστήματος, υπό την προϋπόθεση ότι καλύπτει συγκεκριμένες ανάγκες:

- **Κεντρική προβολή όλων των κρατήσεων**
Ο επιχειρηματίας χρειάζεται άμεση εικόνα για το ποιες κρατήσεις έχουν γίνει, ποιοι πελάτες αναμένονται, πόσα άτομα έρχονται, σε ποια ώρα και ποιο τραπέζι θα χρησιμοποιήσουν.
- **Διαχείριση τραπεζιών με βάση διαθεσιμότητα και χωρητικότητα**
Η εφαρμογή πρέπει να μπορεί να εντοπίζει αυτόματα τραπέζια με επαρκή αριθμό θέσεων για κάθε κράτηση, αποφεύγοντας υπερπλήρωση ή διπλοκρατήσεις.
- **Αναγνώριση φορτισμένων ωρών**
Μέσα από την παρακολούθηση των κρατήσεων, ο ιδιοκτήτης αποκτά εικόνα για τις ώρες αιχμής, γεγονός που τον βοηθά να κατανείμει το προσωπικό πιο αποτελεσματικά ή να προτείνει διαφορετικές ώρες στους πελάτες.
- **Απλό και ασφαλές login**
Ο ιδιοκτήτης πρέπει να έχει δυνατότητα εισόδου σε διαχειριστικό περιβάλλον μόνο με έγκυρα διαπιστευτήρια (username/password), προστατεύοντας τις πληροφορίες από μη εξουσιοδοτημένη πρόσβαση.
- **Ιστορικό κρατήσεων**
Η δυνατότητα αναδρομής στο παρελθόν και η εξέταση του προφίλ των πελατών επιτρέπει καλύτερη εξυπηρέτηση (π.χ. επαναλαμβανόμενοι πελάτες, δημοφιλείς μέρες/ώρες).
- **Εύκολη προσαρμογή σε αλλαγές**
Η εφαρμογή πρέπει να υποστηρίζει ενδεχόμενες μελλοντικές αλλαγές στο ωράριο, στον αριθμό τραπεζιών ή στη διαθεσιμότητα.
- **Οικονομία χρόνου και μείωση λαθών**
Η αυτοματοποίηση των κρατήσεων μειώνει την ανάγκη για τηλεφωνική εξυπηρέτηση και χειρόγραφες καταχωρήσεις, ελαχιστοποιώντας ανθρώπινα λάθη και καθυστερήσεις.

2.3.3 Κοινά Οφέλη και Ανάγκη Ψηφιοποίησης

Και οι δύο πλευρές (χρήστης – επιχείρηση) επωφελούνται από την εφαρμογή:

Πελάτης	Ιδιοκτήτης
Δεν χρειάζεται να τηλεφωνήσει	Δεν αποσπάται το προσωπικό
Κάνει κράτηση όποτε τον εξυπηρετεί	Έχει διαθέσιμο ιστορικό κρατήσεων
Λαμβάνει επιβεβαίωση άμεσα	Οργανώνει τραπέζια βάσει πληρότητας
Νιώθει ασφαλής με τα δεδομένα του	Έχει πρόσβαση από οπουδήποτε

Η χρήση ενός τέτοιου συστήματος βελτιώνει την εμπειρία εξυπηρέτησης, αυξάνει την επαγγελματική εικόνα της επιχείρησης και αποτελεί πλέον αναγκαίο εργαλείο για κάθε σύγχρονο εστιατόριο που επιθυμεί να παραμένει ανταγωνιστικό.

Η **τεχνητή νοημοσύνη** (Artificial Intelligence – AI) έχει εισχωρήσει δυναμικά στον τομέα των διαδικτυακών εφαρμογών και ειδικά στο πεδίο της **εξατομικευμένης εξυπηρέτησης χρηστών**. Η χρήση τεχνικών AI μπορεί να προσδώσει σημαντικά πλεονεκτήματα σε μια εφαρμογή κρατήσεων, μετατρέποντάς την από ένα απλό εργαλείο κράτησης τραπεζιού σε έναν **έξυπνο ψηφιακό βοηθό**.

Αν και στην παρούσα πτυχιακή εργασία η τεχνητή νοημοσύνη δεν έχει ακόμα ενσωματωθεί ενεργά, αναγνωρίζεται η αξία της και διατυπώνονται συγκεκριμένες δυνατότητες μελλοντικής ενσωμάτωσής της, προκειμένου να ενισχυθεί η **λειτουργικότητα, η προσωποποίηση και η εμπειρία χρήστη**.

2.4.1 Εξατομικευμένες Προτάσεις Εστιατορίων

Μια από τις βασικές εφαρμογές της ΑΙ σε τέτοια συστήματα είναι η προσωποποιημένη πρόταση καταστημάτων. Η εφαρμογή, αφού αναλύσει:

- Το ιστορικό των κρατήσεων του κάθε χρήστη (συχνότητα, τύπος κουζίνας, περιοχές),
- Τα δημοφιλέστερα εστιατόρια,
- Τις προτιμώμενες ημέρες και ώρες,

μπορεί να προτείνει δυναμικά εστιατόρια που ταιριάζουν στο προσωπικό του προφίλ.

Για παράδειγμα, αν ένας χρήστης κάνει συχνά κρατήσεις για 2 άτομα σε ιταλική κουζίνα, η εφαρμογή θα εμφανίζει πρώτα τα σχετικά εστιατόρια όταν ο χρήστης επισκέπτεται την πλατφόρμα.

2.4.2 Πρόβλεψη Πληρότητας & Διαχείριση Ζήτησης

Ένα άλλο εξαιρετικά χρήσιμο χαρακτηριστικό της τεχνητής νοημοσύνης είναι η **πρόβλεψη πληρότητας** ανά ημέρα, ώρα και εστιατόριο. Χρησιμοποιώντας δεδομένα όπως:

- Το ποσοστό πληρότητας ανά εβδομάδα ή περίοδο,
- Τις εποχιακές τάσεις (π.χ. αυξημένη ζήτηση τα Σαββατοκύριακα ή σε γιορτές),
- Τον αριθμό επισκέψεων χρηστών σε σελίδες συγκεκριμένων εστιατορίων,

η εφαρμογή μπορεί να "προβλέπει" πότε ένα εστιατόριο θα είναι πιθανόν πλήρες και να προτείνει εναλλακτικές ώρες ή εστιατόρια.

Επιπλέον, με τέτοια δεδομένα μπορεί να δημιουργηθεί **δυναμική στρατηγική προώθησης**, π.χ. να προσφέρονται εκπτώσεις σε ώρες μη αιχμής για καλύτερη κατανομή κρατήσεων

2.4.3 Αυτόματες Υπενθυμίσεις & Διαχείριση No-shows

Η τεχνητή νοημοσύνη μπορεί επίσης να βελτιώσει τη **συμπεριφορά των χρηστών μετά την κράτηση**:

- **Αυτόματες ειδοποιήσεις** (email ή SMS reminders) την ημέρα της κράτησης.
- **Ανάλυση του ιστορικού εμφάνισης** (εμφανίστηκε / δεν εμφανίστηκε).
- Σε περίπτωση συχνών "no-shows", το σύστημα θα μπορούσε να απαιτεί προκαταβολή ή να περιορίζει κρατήσεις.

Μια τέτοια λειτουργία αυξάνει την **αξιοπιστία της βάσης δεδομένων κρατήσεων** και μειώνει τα λειτουργικά κόστη για την επιχείρηση.

2.4.4 Ανάλυση συναισθήματος (Sentiment Analysis)

Εφόσον η εφαρμογή μελλοντικά εμπλουτιστεί με κριτικές χρηστών ή αξιολογήσεις, μπορεί να εφαρμοστεί ανάλυση συναισθήματος (NLP – Natural Language Processing) στα σχόλια, προκειμένου να κατηγοριοποιηθούν οι εμπειρίες σε θετικές, αρνητικές ή ουδέτερες.

Αυτό επιτρέπει:

- Πιο αντικειμενική κατάταξη εστιατορίων
- Ειδοποιήσεις προς τον ιδιοκτήτη σε περίπτωση πολλών αρνητικών εμπειριών
- Στατιστικά ανάλυσης φήμης της επιχείρησης

2.4.5 Προσαρμοσμένη εμπειρία περιήγησης (Smart UI)

Η διεπαφή χρήστη (UI) μπορεί να μετασχηματιστεί με βάση την **προσωπικότητα και τις συνήθειες του χρήστη**:

- Αυτόματη συμπλήρωση πεδίων με βάση παλιές κρατήσεις
- Προτεινόμενες ώρες βάσει προηγούμενων επισκέψεων
- Προσαρμοσμένο layout με «αγαπημένα εστιατόρια»

Αυτό οδηγεί σε **σημαντική αύξηση της ικανοποίησης χρήστη** και δημιουργεί αίσθηση εξατομικευμένης εξυπηρέτησης.

2.4.6 Τεχνικές Εφαρμογής AI

Η ενσωμάτωση τεχνητής νοημοσύνης μπορεί να υλοποιηθεί με εργαλεία όπως:

- **Python scripts σε background** με βιβλιοθήκες όπως scikit-learn ή TensorFlow
- **Machine Learning APIs** (π.χ. Google AI, Microsoft Azure, HuggingFace)
- **Ανάλυση δεδομένων** μέσω SQL και AI συνδυαστικά

Συμπερασματικά, η τεχνητή νοημοσύνη μπορεί να **μεταμορφώσει πλήρως** την εμπειρία χρήσης μιας εφαρμογής κρατήσεων, από ένα απλό εργαλείο σε ένα **έξυπνο σύστημα υποστήριξης αποφάσεων**. Αν και δεν υλοποιείται στο παρόν στάδιο, η επέκταση της εφαρμογής Book&Bite προς αυτήν την κατεύθυνση αποτελεί μια **ρεαλιστική και ισχυρή προοπτική για το μέλλον**.

ΚΕΦΑΛΑΙΟ 3

3.1 Τεχνολογίες Frontend & Backend για την Ανάπτυξη Εφαρμογών

Ο διαχωρισμός της εφαρμογής σε frontend και backend επιτρέπει την ανεξαρτησία μεταξύ διεπαφής χρήστη και διαχείρισης δεδομένων. Αυτό οδηγεί σε καλύτερη οργάνωση, επεκτασιμότητα και συντήρηση του λογισμικού.

✓ **Frontend (πλευρά πελάτη):**

Το κομμάτι της εφαρμογής που αλληλεπιδρά απευθείας με τον χρήστη και αφορά την εμφάνιση και λειτουργικότητα της διεπαφής.

Τεχνολογίες:

- **HTML5:** Ορίζει τη δομή του περιεχομένου.
- **CSS3:** Καθορίζει την εμφάνιση και την αισθητική του περιβάλλοντος.
- **JavaScript (Vanilla JS):** Χρησιμοποιείται για διαδραστικές λειτουργίες (φόρμες, επιλογές, εμφάνιση δεδομένων).

✓ **Backend (πλευρά διακομιστή):**

Το backend διαχειρίζεται τα αιτήματα του χρήστη, εκτελεί λογική, αλληλεπιδρά με τη βάση δεδομένων και επιστρέφει απαντήσεις.

Τεχνολογίες:

- **Node.js:** Πλατφόρμα εκτέλεσης JavaScript στον server.
- **Express.js:** Framework για εύκολη δημιουργία RESTful API endpoints.
-

Ο διακομιστής χειρίζεται όλα τα αιτήματα κρατήσεων και τραπεζιών, κάνει ελέγχους εγκυρότητας, δημιουργεί μοναδικούς κωδικούς κράτησης και αποθηκεύει τα δεδομένα στην PostgreSQL.

3.2 Γλώσσες Προγραμματισμού (HTML, CSS, JavaScript)

Η υλοποίηση της διεπαφής έγινε με τις εξής τεχνολογίες:

✓ **HTML (HyperText Markup Language)**

Χρησιμοποιήθηκε για τη δομή της σελίδας. Περιλαμβάνει:

- Τίτλους (<h1>, <h2>)
- Φόρμες (<form>)
- Ετικέτες (<label>)
- Πεδία εισόδου (<input>, <select>)
- Κουμπιά (<button>)

✓ CSS (Cascading Style Sheets)

Ορίζει την αισθητική της εφαρμογής:

- Χρήση χρωματικής παλέτας με φόντο, κουμπιά και πλαίσια.
- Responsive σχεδίαση για υποστήριξη κινητών συσκευών.
- Χρήση σκιών, περιθωρίων, στρογγυλεμένων γωνιών και animation (@keyframes) για αισθητική εμπειρία χρήστη.

✓ JavaScript

Η γλώσσα που δίνει ζωή στην εφαρμογή. Στην εφαρμογή Book&Bite χρησιμοποιείται για:

- Την **απόκρυψη/εμφάνιση** περιοχών του UI ανάλογα με την επιλογή του χρήστη.
- Την **επικοινωνία με το backend** μέσω fetch() (AJAX).

Τον **δυναμικό έλεγχο στοιχείων** όπως ώρες λειτουργίας, φίλτρα, preview εστιατορίων, αποστολή φόρμας.

3.3 Βάσεις Δεδομένων

Η βάση δεδομένων αποτελεί το θεμέλιο της εφαρμογής, καθώς σε αυτή αποθηκεύονται όλες οι σχετικές πληροφορίες (κρατήσεις, τραπέζια, πελάτες κ.λπ.).

✓ PostgreSQL

Επιλέχθηκε ως **ισχυρή σχεσιακή βάση δεδομένων**, με δυνατότητα επεκτάσεων, ασφάλειας και αποδοτικής διαχείρισης δεδομένων.

Στην παρούσα εφαρμογή δημιουργήθηκαν δύο βασικοί πίνακες:

1. **reservations**
 - id (πρωτεύον κλειδί)
 - name, guests, date, time, occasion, bookingcode, restaurant_name
2. **tables**
 - id, restaurant_name, available_tables, people_count

Η βάση λειτουργεί κάτω από τον PostgreSQL server και συνδέεται με το backend μέσω της βιβλιοθήκης pg του Node.js

3.4 Διαχείριση Ταυτότητας και Εξουσιοδότηση Χρηστών (Login, Roles)

Μια λειτουργική και ασφαλής εφαρμογή κρατήσεων πρέπει να υποστηρίζει **μηχανισμούς ταυτοποίησης** (authentication) και **εξουσιοδότησης** (authorization), διαχωρίζοντας τους ρόλους.

✓ Τύποι Χρηστών:

- **Απλός χρήστης (πελάτης):** Μπορεί να κάνει κράτηση, να συμπληρώσει φόρμα και να δει την επιβεβαίωση.

- **Ιδιοκτήτης εστιατορίου (admin):** Έχει πρόσβαση στον πίνακα κρατήσεων και μπορεί να δει όλα τα στοιχεία.

✓ **Διαδικασία Login:**

- Ο ιδιοκτήτης εισάγει όνομα χρήστη και κωδικό.
- Γίνεται απλός έλεγχος στο frontend και ενεργοποιείται η εμφάνιση του πίνακα κρατήσεων.
- Στο backend μπορεί να ενσωματωθεί στο μέλλον σύστημα role-based access ή token-based authentication (JWT).

Παρότι προς το παρόν ο έλεγχος είναι τοπικός (hardcoded credentials), η δομή της εφαρμογής επιτρέπει στο μέλλον την **επέκταση με σύστημα πολλαπλών χρηστών** με εγγραφή και διαχείριση ρόλων.

ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

4.1 Σκοπός της Εφαρμογής και Βασικές Λειτουργίες

Η εφαρμογή **Book&Bite** αποτελεί ένα διαδικτυακό σύστημα διαχείρισης κρατήσεων εστιατορίων, σχεδιασμένο για να εξυπηρετεί τόσο τους πελάτες που επιθυμούν να κάνουν εύκολα και άμεσα κράτηση τραπέζιού, όσο και τους ιδιοκτήτες/διαχειριστές των εστιατορίων, οι οποίοι μπορούν να βλέπουν συγκεντρωτικά τις κρατήσεις και τη διαθεσιμότητα των τραπεζιών.

Ο βασικός **σκοπός** της εφαρμογής είναι η **απλοποίηση και αυτοματοποίηση της διαδικασίας κράτησης**, με τρόπο που να μειώνει τα ανθρώπινα λάθη, να εξοικονομεί χρόνο και να αυξάνει την αποδοτικότητα των επιχειρήσεων εστίασης.

✓ Κύριες λειτουργίες της εφαρμογής:

Για τον τελικό χρήστη:

- Επιλογή πόλης και εστιατορίου.
 - Προβολή πληροφοριών για κάθε εστιατόριο.
 - Επιλογή ημερομηνίας, ώρας και αριθμού ατόμων.
 - Συμπλήρωση στοιχείων κράτησης και υποβολή.
- Εμφάνιση επιβεβαίωσης με μοναδικό κωδικό κράτησης.

Για τον ιδιοκτήτη:

- Σύνδεση στο σύστημα μέσω login.
- Προβολή λίστας κρατήσεων σε πίνακα.
- Ενημέρωση για διαθέσιμα τραπέζια ανά κράτηση.
- Δυνατότητα εποπτείας της πληρότητας.

Η εφαρμογή παρέχει **βασική αλλά πλήρη λειτουργικότητα**, με δυνατότητα μελλοντικής επέκτασης (π.χ. πολλαπλοί χρήστες, ειδοποιήσεις, online πληρωμές, στατιστικά).

4.2 Διαδικασία Κρατήσεων

Η ροή της εφαρμογής ακολουθεί τα παρακάτω βήματα:

• Βήμα 1: Επιλογή πόλης

Ο χρήστης επιλέγει την πόλη στην οποία επιθυμεί να πραγματοποιήσει κράτηση (π.χ. Πάτρα, Αθήνα, Θεσσαλονίκη, Λάρισα). Η επιλογή αυτή καθορίζει ποια εστιατόρια θα εμφανιστούν στη συνέχεια.

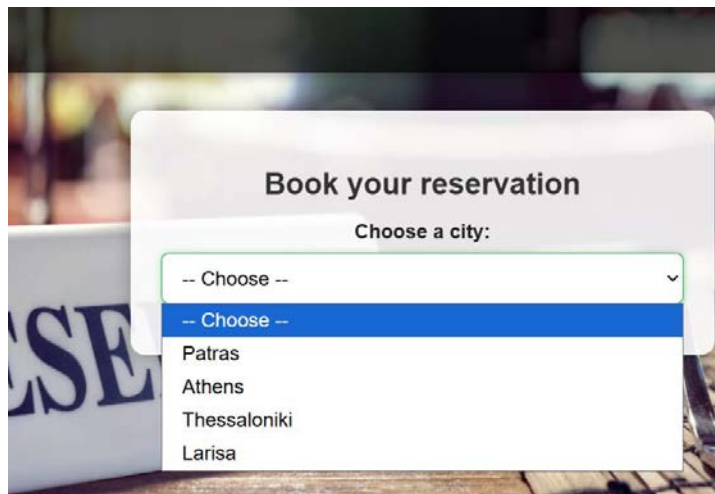
<!-- Επιλογή πόλης -->

<label for="city">Choose a city:</label>

```

<select id="city" >
  <option value="">-- Choose --</option>
  <option value="patra">Patras</option>
  <option value="athens">Athens</option>
  <option value="thessaloniki">Thessaloniki</option>
  <option value="larisa">Larisa</option>
</select>

```



- **Βήμα 2: Επιλογή εστιατορίου**

Με βάση την επιλεγμένη πόλη, φορτώνονται αυτόματα τα διαθέσιμα εστιατόρια. Κάθε επιλογή συνοδεύεται από προεπισκόπηση (φωτογραφία, ώρες λειτουργίας, κουζίνα, διεύθυνση, αξιολόγηση).

```

const restaurantDetails = {
  "Salumeria": {
    cuisine: "Greek, Italian",
    hours: "12:00 - 23:55 ",
    address: "Agίου Andreou 25, Patra",
    price: "€€€",
    rating: "4.5/5",
    image: "images/salumeria.jpg"
  },
  "Rodon": {
    cuisine: "Mexican",
    hours: "18:00 - 23:55 ",
    address: "Platamonos 32, Patra",
    price: "€€",
    rating: "4.2/5",
    image: "images/rodon.jpg"
  },
  "Black Duck": {
    cuisine: "Mediterranean",

```

```

    hours: "12:00 - 23:55 ",
    address: "Ermou 5, Athens",
    price: "€€",
    rating: "4.7/5",
    image: "images/blackduck.jpg"
  },
  "Taratsa Restaurant": {
    cuisine: "Italian",
    hours: "15:00 - 23:55 ",
    address: "Korinthou 89, Psyrh",
    price: "€€€",
    rating: "4.1/5",
    image: "images/taratsa.jpg"
  },
  "Ergon Agora": {
    cuisine: "Chinese",
    hours: "14:00 - 23:55 ",
    address: "Praxitelous 22, Thessaloniki ",
    price: "€€€",
    rating: "4.8/5",
    image: "images/ergon.jpg"
  },
  "Spitiko": {
    cuisine: "Italian",
    hours: "13:00 - 23:55 ",
    address: "Aristotelous 58, Thessaloniki",
    price: "€€",
    rating: "4.5/5",
    image: "images/spitiko.jpg"
  },
  "Brusko": {
    cuisine: "Italian",
    hours: "14:00 - 23:55 ",
    address: "Botsi 89, Larisa",
    price: "€€€",
    rating: "4.4/5",
    image: "images/brusko.jpg"
  },
  "Gooseman": {
    cuisine: "Lebanese",
    hours: "13:00 - 23:00 ",
    address: "Arsaki 5, Larisa",
    price: "€€",
    rating: "4.0/5",
    image: "images/gooseman.jpg"
  },
};

```

Κάποια παραδείγματα εστιατορίων ανάλογα με την πόλη που έχουμε διαλέξει:

Book your reservation

Choose a city:

Patras

Choose a restaurant:

-- Choose --

-- Choose --

Salumeria

Rodon

Choose a city:

Athens

Choose a restaurant:

-- Choose --

-- Choose --

Black Duck

Taratsa Restaurant

Choose a city:

Thessaloniki

Choose a restaurant:

-- Choose --

-- Choose --

Ergon Agora

Spitiko

Book your reservation

Choose a city:

Larisa

Choose a restaurant:

-- Choose --

-- Choose --

Brusko

Gooseman

- **Βήμα 3: Επιλογή ημερομηνίας**

Ο χρήστης μπορεί να επιλέξει ημερομηνία από ημερολόγιο (<input type="date">) — δεν επιτρέπονται προηγούμενες ημέρες. Εμφανίζεται μόνο αν έχει επιλεγεί εστιατόριο.

Book your reservation

Choose a city:

Choose a restaurant:

Choose a date:

ηη/μμ/εεεε

July 2025

Δε	Τρ	Τε	Πε	Πα	Σα	Κυ
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Διαγραφή Σήμερα

- **Βήμα 4: Επιλογή ώρας και λεπτών**

Ανάλογα με τις ώρες λειτουργίας του εστιατορίου, εμφανίζονται δυναμικά μόνο οι επιτρεπόμενες ώρες και λεπτά (ανά 30 λεπτά). Η επιλογή προσαρμόζεται αυτόματα αν η ημερομηνία είναι η σημερινή (ώστε να μην επιτρέπεται κράτηση σε παρελθούσα ώρα).

The screenshot shows a reservation form with the following fields:

- Name:** A text input field.
- Email:** A text input field.
- Time:** A dropdown menu for hours (12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23) and a dropdown for minutes (00, 30). The selected time is 12:00.
- People:** A dropdown menu showing 30.

- **Βήμα 5: Συμπλήρωση στοιχείων κράτησης**

Ο χρήστης συμπληρώνει:

- Ονοματεπώνυμο
- Email
- Αριθμό ατόμων (1–10)
- Προαιρετική περίσταση (π.χ. Γενέθλια, Επέτειος)

The screenshot shows a reservation form with the following fields:

- Name:** A text input field.
- Email:** A text input field.
- Time:** A dropdown menu.

People:

Occasion (optional):

None

- **Βήμα 6: Υποβολή φόρμας**

Με το πάτημα του κουμπιού **Submit**:

- Στέλνεται POST αίτημα στο backend (/api/reservations).
- Ο server εντοπίζει το κατάλληλο τραπέζι με βάση τη χωρητικότητα και καταχωρεί την κράτηση στη βάση.
- Δημιουργείται **μοναδικός κωδικός κράτησης** (π.χ. A92D4FHT) και εμφανίζεται στον χρήστη σε παράθυρο επιβεβαίωσης.

Occasion (optional):

None

Submit

🎉 **Your reservation is confirmed!**

👤 **Guests:** 2

📅 **Date:** 2025-07-26

🕒 **Time:** 12:00

🍽️ **Restaurant:** Brusko

🔒 **Booking Code:** VQXBE2UW

It will be sent to you a confirmation message at your email address! Thank you for your reservation!

- **Βήμα 7: Εμφάνιση πίνακα κρατήσεων (για ιδιοκτήτη)**

Αν ο χρήστης συνδεθεί ως **ιδιοκτήτης** (π.χ. με username: est2025, password: Est12346!), εμφανίζεται **πίνακας με όλες τις κρατήσεις** από όλα τα εστιατόρια. Περιλαμβάνει:

- Όνομα πελάτη
- Αριθμό ατόμων
- Ημερομηνία και ώρα
- Περίσταση
- Κωδικό κράτησης
- Εστιατόριο
- Διαθέσιμα τραπέζια της ίδιας χωρητικότητας

Ο πίνακας φορτώνεται δυναμικά μέσω αιτήματος στο API (GET /api/reservations και GET /api/tables).

Σύνδεση:

Πίνακας κρατήσεων:

Reservations Overview							
Name	Guests	Date	Time	Occasion	Booking Code	Restaurant	Available tables
ap	2	2025-05-28T21:00:00.000Z	22:00:00	-	86RWK4PX	Gooseman	10
fd dhf	4	2025-07-11T21:00:00.000Z	20:00:00	-	K24RLD0P	Spitiko	7
vagg	2	2025-08-06T21:00:00.000Z	20:00:00	-	H2OG346B	Taratsa Restaurant	10
eirini papad	2	2025-06-05T21:00:00.000Z	22:00:00	-	YYPB1HDV	Taratsa Restaurant	10

Τεχνικά Σημεία Αρχιτεκτονικής:

Στοιχείο	Περιγραφή
Frontend	HTML5, CSS3, JavaScript (Vanilla JS)
Backend	Node.js με Express.js
API Endpoints	/api/reservations, /api/tables
Βάση δεδομένων	PostgreSQL
Αντιστοίχιση τραπεζιού	SQL query με LIMIT 1 για το μικρότερο επαρκές τραπέζι
Μοναδικός Κωδικός	Γεννάται με συνάρτηση τυχαίων χαρακτήρων (A1B2C3D4)
Ασφάλεια διαχειριστή	Έλεγχος login με username/password
Προεπισκόπηση εστιατορίων	HTML section με image + rating + κουζίνα + διεύθυνση

4.3 Παρουσίαση Προγραμματιστικού Κώδικα

Στην παρούσα ενότητα παρουσιάζονται τα βασικά σημεία του προγραμματιστικού κώδικα της εφαρμογής **Book&Bite**, χωρισμένα σε τρία επίπεδα: **Frontend (HTML/CSS/JS)**, **Backend (Node.js με Express)** και **Σύνδεση με Βάση Δεδομένων (PostgreSQL)**.

Η ανάπτυξη πραγματοποιήθηκε με απλές και καθαρές τεχνικές, με στόχο τη σωστή ροή της πληροφορίας και τη βελτιωμένη εμπειρία χρήστη.

4.3.1 Κώδικας Διεπαφής Χρήστη (HTML/JavaScript)

Η διεπαφή της εφαρμογής σχεδιάστηκε με HTML5 και εμπλουτίστηκε με JavaScript για την υλοποίηση δυναμικής συμπεριφοράς. Παρακάτω φαίνεται το βασικό τμήμα HTML για την επιλογή πόλης και εστιατορίου:

```

<!-- Επιλογή πόλης -->
<label for="city">Choose a city:</label>
<select id="city" >
  <option value="">-- Choose --</option>
  <option value="patra">Patras</option>
  <option value="athens">Athens</option>
  <option value="thessaloniki">Thessaloniki</option>
  <option value="larisa">Larisa</option>
</select>

<!-- Επιλογή εστιατορίου -->
<div id="restaurant-selection" style="display: none;">
  <label for="restaurant">Choose a restaurant:</label>
  <select id="restaurant"></select>
</div>

```

Με JavaScript υλοποιείται ο δυναμικός έλεγχος των πεδίων και η ανάκτηση των πληροφοριών για κάθε εστιατόριο:

```

// Όταν επιλέγεται πόλη, εμφανίζονται τα αντίστοιχα εστιατόρια
document.getElementById("city").addEventListener("change", function() {
  const city = this.value;
  const restaurantSelect = document.getElementById("restaurant");
  const restaurantDiv = document.getElementById("restaurant-selection");

  // Καθαρισμός παλιών επιλογών
  restaurantSelect.innerHTML = "<option value=''>-- Choose --</option>";

  if (city) {
    restaurants[city].forEach(rest => {
      let option = document.createElement("option");
      option.value = rest;
      option.textContent = rest;
      restaurantSelect.appendChild(option);
    });
    restaurantDiv.style.display = "block";
  } else {
    restaurantDiv.style.display = "none";
  }
});

```

Η επιλογή εστιατορίου ενεργοποιεί την προβολή πληροφοριών και την εμφάνιση ημερολογίου και φόρμας κράτησης, με βάση το ωράριο λειτουργίας.

4.3.2 Υποβολή Κράτησης και Fetch προς Backend

Η αποστολή της κράτησης υλοποιείται με χρήση fetch() για αποστολή POST αιτήματος στον server:

```
// Υποβολή κράτησης
document.getElementById("booking-form").addEventListener("submit", async function(event) {
    event.preventDefault();
    console.log("Submit event triggered");

    // Εμφάνιση loading μηνύματος
    document.getElementById("confirmation").innerText = "Submitting your reservation...";
    document.getElementById("confirmation-box").style.display = "block";

    const name = document.getElementById("name").value;
    //const email = document.getElementById("email").value;
    const date = document.getElementById("date").value;
    const hour = document.getElementById("hour").value;
    const minute = document.getElementById("minute").value;
    const time = `${hour}:${minute}`;
```

```
const guests = document.getElementById("guests").value;
const occasion = document.getElementById("occasion").value;

let occasionText = "";
if (occasion) {
    const occasionLabel = document.querySelector(`#occasion option[value="${occasion}"]`).textContent;
    occasionText = ` for the occasion: ${occasionLabel}`;
}
//const bookingCode = generateBookingCode();
try {
    const response = await fetch('http://localhost:3000/api/reservations', {
```

```
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({
            name,
            guests,
            date,
            time,
            occasion,
            restaurant_name: document.getElementById("restaurant").value
        })),
    });

const data = await response.json();
```

4.3.3 Κώδικας Backend – Node.js (server.js)

Το backend έχει υλοποιηθεί σε Node.js με Express. Η κύρια διαδρομή αποθήκευσης κράτησης είναι η εξής:

```
app.post('/api/reservations', async (req, res) => {
  const { name, guests, date, time, occasion, restaurant_name } = req.body;
  const guestsCount = Number(guests);
  const bookingcode = generateBookingCode();

  if (!name || !guests || !date || !time || !restaurant_name) {
    return res.status(400).json({ error: 'Όλα τα πεδία είναι υποχρεωτικά.' });
  }

  const occasionValue = occasion || '';

  const client = await pool.connect();

  try {
    await client.query('BEGIN');
```

```
// Βρίσκουμε κατάλληλο τραπέζι με βάση τον αριθμό ατόμων (ή ίσο ή μεγαλύτερο)
const tableResult = await client.query(
  `SELECT * FROM tables
   WHERE restaurant_name = $1
   AND people_count >= $2
   ORDER BY people_count ASC
   LIMIT 1`,
  [restaurant_name, guestsCount]
);

if (tableResult.rowCount === 0) {
  throw new Error('Δεν υπάρχουν διαθέσιμα τραπέζια για τον αριθμό ατόμων.');
```

```
await client.query('COMMIT');

res.status(200).json({ message: 'Κράτηση αποθηκεύτηκε με επιτυχία!', bookingcode });
} catch (err) {
  await client.query('ROLLBACK');
  console.error('Σφάλμα στην κράτηση:', err);
  res.status(500).json({ error: 'Σφάλμα αποθήκευσης κράτησης' });
} finally {
  client.release();
}
});
```

Η βάση δεδομένων διαχειρίζεται επίσης τον πίνακα tables, ώστε να καταγράφονται οι διαθέσιμες θέσεις ανά τραπέζι και ανά εστιατόριο.

4.3.4 Παραλαβή Κωδικού Κράτησης από τον Server

Η δημιουργία του κωδικού κράτησης (bookingcode) δεν πραγματοποιείται από τον πελάτη (browser), αλλά αποτελεί ευθύνη του διακομιστή. Κατά την υποβολή της φόρμας κράτησης από τον χρήστη, ο server (Node.js backend) επεξεργάζεται τα δεδομένα, δημιουργεί έναν μοναδικό κωδικό και τον επιστρέφει στον browser.

Στο frontend JavaScript αρχείο, αυτός ο κωδικός παραλαμβάνεται μέσω της παρακάτω εντολής:

```
const bookingCode = data.bookingcode || 'Δεν παρέχεται κωδικός';
```

Στη συνέχεια, αυτός εμφανίζεται στον χρήστη ως μέρος του μηνύματος επιβεβαίωσης:

```
const confirmationMessage = `
  <p>🎉 <strong> Your reservation is confirmed!</strong></p>

  <p>👥 <strong>Guests:</strong> ${guests}</p>
  <p>📅 <strong>Date:</strong> ${date}</p>
  <p>🕒 <strong>Time:</strong> ${time}</p>
  <p>🍽️ <strong>Restaurant:</strong> ${document.getElementById("restaurant").value}</p>
  <p>🌸 <strong>Occasion:</strong> ${occasionText.trim()}</p>` : ""

  <p>🔑 <strong>Booking Code: </strong>${bookingCode}</p>

  <p>It will be sent to you a confirmation message at your email address!Thank you for your reservation!</p>
`;

document.getElementById("confirmation").innerHTML = confirmationMessage;
```

Ο κωδικός αυτός είναι μοναδικός για κάθε κράτηση και αποθηκεύεται στη βάση δεδομένων. Η δημιουργία του πραγματοποιείται με JavaScript κώδικα στον backend, όπως φαίνεται στο αντίστοιχο αρχείο Node.js (στο route /api/reservations).

4.3.5 Προβολή Κρατήσεων από Ιδιοκτήτη

Ο ιδιοκτήτης κάνει είσοδο μέσω login φόρμας, η οποία ελέγχει τα credentials τοπικά (για λόγους απλότητας):

```
document.getElementById('loginForm').addEventListener('submit', function (event) {
    event.preventDefault();

    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    if( username === 'est2025' && password === 'Est12346!') {
        loadReservations(); // <--- κάλεσέ την εδώ
        document.getElementById('loginForm').style.display = 'none'
    } else{
        alert('Invalid login credentials!')
    }
});
```

Ανάκτηση και Εμφάνιση Κρατήσεων

Μόλις γίνει επιτυχής σύνδεση, καλείται η συνάρτηση loadReservations(), η οποία εκτελεί δύο fetch αιτήματα:

1. Για να πάρει όλες τις κρατήσεις από το endpoint /api/reservations
2. Για να πάρει τις πληροφορίες των τραπεζιών από /api/tables

Ο κώδικας έχει ως εξής:

```
function loadReservations() {
    fetch('http://localhost:3000/api/reservations')
        .then(response => response.json())
        .then(reservationsData => {
            fetch('http://localhost:3000/api/tables')
                .then(response => response.json())
                .then(tablesData => {
                    const tableBody = document.querySelector('#reservationsTable tbody');
                    tableBody.innerHTML = ''; // Καθαρισμός παλιών

                    reservationsData.forEach(reservation => {
                        const matchingTable = tablesData.find(table =>
                            table.restaurant_name.trim().toLowerCase() === reservation.restaurant_name.trim().toLowerCase() &&
                            table.people_count === reservation.guests
                        );

                        const availableTables = matchingTable ? matchingTable.available_tables : 'Unknown';

                        const row = document.createElement('tr');
                        row.innerHTML = `
                            <td>${reservation.name}</td>
                            <td>${reservation.guests}</td>
                            <td>${reservation.date}</td>
                            <td>${reservation.time}</td>
```

```

        <td>${reservation.occasion || '-'}</td>
        <td>${reservation.bookingcode}</td>
        <td>${reservation.restaurant_name}</td>
        <td>${availableTables}</td>
    `;
    tableBody.appendChild(row);
  });

  // Εμφάνιση του section
  document.getElementById('reservationsSection').style.display = 'block';
})
.catch(error => {
  console.error('✖ Σφάλμα φόρτωσης κρατήσεων:', error);
  alert('Σφάλμα φόρτωσης κρατήσεων');
})
.catch(error => {
  console.error('✖ Σφάλμα φόρτωσης reservations:', error);
  alert('Σφάλμα φόρτωσης κρατήσεων');
});

```

Εμφάνιση στον Χρήστη

Ο παραπάνω κώδικας γεμίζει δυναμικά έναν HTML πίνακα (#reservationsTable tbody) με όλα τα στοιχεία της κάθε κράτησης, όπως:

- Όνομα πελάτη
- Αριθμός ατόμων
- Ημερομηνία / Ώρα
- Περίσταση
- Κωδικός κράτησης
- Όνομα εστιατορίου
- Διαθέσιμα τραπέζια

Ο πίνακας εμφανίζεται μόνο αφού φορτωθούν όλα τα δεδομένα με επιτυχία.

ΚΕΦΑΛΑΙΟ 5 ΕΠΙΔΕΙΞΗ ΕΦΑΡΜΟΓΗΣ

5.1 Περιγραφή Διαδικασίας

Η δημιουργία της εφαρμογής Book&Bite ακολούθησε μια τυπική διαδικασία ανάπτυξης λογισμικού, η οποία περιλάμβανε διαδοχικά στάδια σχεδίασης, υλοποίησης, δοκιμών και βελτιστοποίησης. Αναλυτικά:

- **Ανάλυση Απαιτήσεων:** Καταγράφηκαν οι ανάγκες τόσο των τελικών χρηστών (πελατών) όσο και των διαχειριστών (ιδιοκτητών εστιατορίων). Ιδιαίτερη έμφαση δόθηκε στη φιλικότητα της διεπαφής και στην αυτοματοποίηση της διαδικασίας κράτησης.

- **Αρχιτεκτονικός Σχεδιασμός:** Η εφαρμογή διαχωρίστηκε σε frontend και backend. Ο frontend είναι υπεύθυνος για την παρουσίαση και την αλληλεπίδραση με τον χρήστη, ενώ το backend επεξεργάζεται τα δεδομένα και επικοινωνεί με τη βάση.
- **Υλοποίηση Backend:** Δημιουργήθηκαν RESTful API endpoints με χρήση του Express.js, για την αποστολή, αποθήκευση και ανάκτηση δεδομένων κρατήσεων και τραπεζιών από τη βάση PostgreSQL.
- **Υλοποίηση Frontend:** Χρησιμοποιήθηκαν HTML, CSS και JavaScript για τον δυναμικό χειρισμό της διεπαφής. Ενσωματώθηκαν φόρμες, διαδραστικά dropdowns και δυναμική προεπισκόπηση εστιατορίων.
- **Δοκιμές και Debugging:** Πραγματοποιήθηκαν λειτουργικοί και αισθητικοί έλεγχοι σε διάφορα σενάρια (π.χ. ώρες μη λειτουργίας, πλήθος κρατήσεων, κενή φόρμα).
- **Τελική Βελτιστοποίηση:** Προστέθηκαν στοιχεία όπως emojis, responsive λειτουργία για κινητές συσκευές και ελαφρύ animation για καλύτερη εμπειρία χρήστη.

5.2 Δυσκολίες & Λύσεις

Κατά τη διάρκεια ανάπτυξης της εφαρμογής, παρουσιάστηκαν αρκετές τεχνικές προκλήσεις:

- **Διαχείριση Διαθεσιμότητας Ώρας Κράτησης:** Απαιτήθηκε δυναμική ενεργοποίηση/απενεργοποίηση των ωρών και λεπτών ανάλογα με το ωράριο κάθε εστιατορίου. Η λύση δόθηκε μέσω JavaScript, με παραμετροποίηση του DOM κατά την επιλογή του χρήστη.
- **Αναζήτηση Κατάλληλου Τραπεζιού:** Έπρεπε να συσχετιστεί κάθε κράτηση με το κατάλληλο τραπέζι (βάσει χωρητικότητας). Η υλοποίηση έγινε μέσω SQL ερωτήματος με LIMIT 1, ώστε να επιλέγεται το πρώτο διαθέσιμο τραπέζι που καλύπτει τις ανάγκες.
- **Διαχείριση και Εμφάνιση Κρατήσεων Ιδιοκτήτη:** Η ταυτόχρονη ανάκτηση δεδομένων από δύο πίνακες (reservations, tables) απαιτούσε συνδυασμό fetch requests και JavaScript rendering. Η λύση βασίστηκε σε Promise chains και DOM manipulation.
- **Επικοινωνία Frontend–Backend:** Χρειάστηκε λεπτομερής έλεγχος των JSON payloads και ορθή διαχείριση των HTTP status responses.

5.3 Σχόλια για την Απόδοση και Επεκτασιμότητα

Η τελική εφαρμογή παρουσιάζει:

- **Καλή ταχύτητα απόκρισης:** Η απλή αρχιτεκτονική (vanilla JS + Node.js) χωρίς βαριά frameworks εξασφαλίζει χαμηλό χρόνο φόρτωσης και άμεση ανταπόκριση σε ενέργειες του χρήστη.
- **Οργανωμένη Δομή Κώδικα:** Η κατηγοριοποίηση των λειτουργιών σε modules (frontend – backend – βάση) επιτρέπει ευκολότερη συντήρηση και debugging.
- **Επεκτασιμότητα:** Η υπάρχουσα δομή μπορεί εύκολα να επεκταθεί με νέες λειτουργίες, όπως:
- Προσθήκη email/sms επιβεβαίωσης.
- Δημιουργία mobile έκδοσης.
- Εμπλουτισμένο login σύστημα (JWT, OAuth).

- **Πίνακας στατιστικών για ιδιοκτήτες.**
- **Ασφάλεια:** Αν και η αυθεντικοποίηση του διαχειριστή είναι σε βασικό επίπεδο (hardcoded credentials), το backend υποστηρίζει περαιτέρω ενσωμάτωση role-based συστημάτων πρόσβασης.
- **User Experience:** Η εφαρμογή έχει σχεδιαστεί ώστε να είναι κατανοητή από χρήστες κάθε επιπέδου. Η χρήση εικόνων, καθαρής τυπογραφίας και απλών οδηγιών καθιστά τη διαδικασία κράτησης ευχάριστη.

ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ

6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η παρούσα πτυχιακή εργασία είχε ως στόχο τη μελέτη, σχεδίαση και υλοποίηση μιας διαδικτυακής εφαρμογής για τη **διαχείριση κρατήσεων σε εστιατόρια**, η οποία να καλύπτει με απλό και λειτουργικό τρόπο τις βασικές ανάγκες τόσο των πελατών όσο και των ιδιοκτητών επιχειρήσεων εστίασης.

Η υλοποίηση βασίστηκε σε τεχνολογίες **ανοικτού κώδικα**, με έμφαση στην **αρχιτεκτονική client-server** και σε σύγχρονες μεθόδους διασύνδεσης χρηστών και διαχείρισης δεδομένων. Το frontend (HTML/CSS/JS) προσφέρει ένα φιλικό και κατανοητό περιβάλλον στον επισκέπτη, ενώ το backend (Node.js/Express) αναλαμβάνει τη λειτουργική επικοινωνία με τη βάση δεδομένων (PostgreSQL), αποθηκεύοντας με ασφάλεια τις πληροφορίες κράτησης.

Μέσα από την αναλυτική ανάπτυξη των λειτουργικών ενοτήτων της εφαρμογής επιτεύχθηκε:

- ✓ Η **δυναμική επιλογή πόλης και εστιατορίου**, ανάλογα με την τοποθεσία του χρήστη.
- ✓ Η **παρουσίαση πληροφοριών εστιατορίου**, όπως ώρες λειτουργίας, τιμολογιακή κατηγορία, είδος κουζίνας και αξιολόγηση.
- ✓ Η **έξυπνη διαχείριση κρατήσεων**, με χρονικούς περιορισμούς ανάλογα με την ημερομηνία και ώρα, καθώς και εμφάνιση επιβεβαίωσης κράτησης με μοναδικό κωδικό.
- ✓ Η **πρόσβαση του ιδιοκτήτη σε πίνακα κρατήσεων**, με πληροφορίες για τραπέζια και αριθμό διαθέσιμων θέσεων.

Η εφαρμογή αποδείχθηκε **σταθερή, επεκτάσιμη και πλήρως λειτουργική**, καλύπτοντας επιτυχώς τα κριτήρια που τέθηκαν κατά τον σχεδιασμό της. Παράλληλα, αποτέλεσε μία χρήσιμη άσκηση πάνω σε βασικές τεχνολογίες ανάπτυξης web εφαρμογών και αρχιτεκτονικές μοντέλων δεδομένων.

Παρότι η παρούσα εφαρμογή ανταποκρίνεται στις βασικές ανάγκες ενός εστιατορίου, υπάρχουν σημαντικά περιθώρια **εξέλιξης και εμπλουτισμού των λειτουργιών** της. Ορισμένες προτεινόμενες κατευθύνσεις είναι:

✓ **Επαλήθευση και Αυθεντικοποίηση Χρηστών**

Η προσθήκη ενός πλήρους συστήματος εγγραφής και σύνδεσης χρηστών, με έλεγχο ταυτότητας και ρόλων (π.χ. πελάτης, ιδιοκτήτης), θα επιτρέψει τη δημιουργία **προσωπικών λογαριασμών** και ιστορικού κρατήσεων.

✓ **Ειδοποιήσεις μέσω Email και SMS**

Η αποστολή **αυτόματων επιβεβαιώσεων μέσω email ή SMS** μετά την κράτηση αποτελεί σημαντική λειτουργικότητα για τη βελτίωση της εμπειρίας χρήστη και την αποτροπή λαθών ή απουσιών.

✓ **Διαχείριση Διαθεσιμότητας σε Πραγματικό Χρόνο**

Η ανάπτυξη ενός **προηγμένου συστήματος κρατήσεων με live ενημέρωση** διαθέσιμων τραπεζιών ανά ημέρα, ώρα και αριθμό ατόμων, θα επιτρέψει καλύτερη εκμετάλλευση των διαθέσιμων πόρων.

✓ **Dashboard και Στατιστικά για Ιδιοκτήτες**

Η δημιουργία ενός **διαχειριστικού πίνακα** με στατιστικά για αριθμό κρατήσεων ανά ημέρα/εβδομάδα/μήνα, ποσοστά πληρότητας, είδος περιστάσεων, κ.λπ., θα προσφέρει πολύτιμα επιχειρηματικά δεδομένα στους ιδιοκτήτες.

✓ **Βελτιστοποίηση για Κινητές Συσκευές (Responsive Design / Mobile App)**

Η μετατροπή της εφαρμογής σε πλήρως **responsive** ή ακόμα και σε **mobile app** μέσω εργαλείων όπως το React Native, θα ενισχύσει τη χρηστικότητα και την προσβασιμότητα από κάθε είδους συσκευή.

✓ **Επεκτασιμότητα για Δίκτυο Εστιατορίων**

Μια ενδεχόμενη μελλοντική επέκταση θα μπορούσε να περιλαμβάνει την ενσωμάτωση **πολλαπλών εστιατορίων**, τα οποία θα μπορούν να δημιουργούν λογαριασμό, να διαχειρίζονται τις κρατήσεις τους και να τροποποιούν τις πληροφορίες τους (μενού, ωράριο, τραπεζία).

✓ **Πολυγλωσσική Υποστήριξη**

Η δυνατότητα εναλλαγής γλώσσας σε **Ελληνικά, Αγγλικά και άλλες γλώσσες**, θα καθιστούσε την εφαρμογή πιο φιλική σε αλλοδαπούς χρήστες και τουρίστες, ειδικά σε τουριστικές πόλεις.

Συνοψίζοντας, η παρούσα εργασία αποτελεί τη **βάση για ένα πραγματικό, παραγωγικό σύστημα κρατήσεων**, το οποίο με τη συνεχή βελτίωση και εξέλιξή του, μπορεί να μετατραπεί σε **ένα πλήρως εμπορικό εργαλείο**, προσαρμόσιμο στις ανάγκες κάθε μονάδας εστίασης.

1. Welling, L., & Thomson, L. (2017). **PHP and MySQL Web Development** (5th Edition). Addison-Wesley.
2. Flanagan, D. (2020). **JavaScript: The Definitive Guide** (7th Edition). O'Reilly Media.
3. Holtsnider, B., & Jaffe, B. D. (2012). **IT Manager's Handbook: Getting your new job done**. Morgan Kaufmann.
4. Sebesta, R. W. (2015). **Programming the World Wide Web** (8th Edition). Pearson.
5. Moisés Macero. (2021). *Building REST APIs with Node.js*. Leanpub. <https://leanpub.com/building-rest-apis-with-nodejs>
6. PostgreSQL Global Development Group. (n.d.). *PostgreSQL 15 Documentation*. <https://www.postgresql.org/docs/>
7. MDN Web Docs. (n.d.). *HTML, CSS, JavaScript documentation*. Mozilla Foundation. <https://developer.mozilla.org/>
8. Express.js. (n.d.). *Express Web Framework Documentation*. <https://expressjs.com/>
9. Node.js Foundation. (n.d.). *Node.js Documentation*. <https://nodejs.org/en/docs/>
10. W3Schools. (n.d.). *Web Development Tutorials*. <https://www.w3schools.com/>