



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος Πτυχιακής Εργασίας	(Ελληνικά) Smart online εξατομικευμένο τουριστικό γραφείο (Αγγλικά) Smart online personalized travel agency web app
Όνοματεπώνυμο Φοιτητή	ΦΩΤΟΠΟΥΛΟΥ ΜΑΡΙΑΜ
Πατρώνυμο	ΕΥΑΓΓΕΛΟΣ
Αριθμός Μητρώου	Π20201
Επιβλέπων	ΕΥΘΥΜΙΟΣ ΑΛΕΠΗΣ, Καθηγητής

Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους συνέβαλαν στην ολοκλήρωση της πτυχιακής μου εργασίας. Πρώτα απ' όλα, ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου, κύριο Ευθύμιο Αλέπη, για την καθοδήγηση που μου προσέφερε σε όλη τη διάρκεια της εκπόνησης της εργασίας αυτής. Επίσης, θέλω να ευχαριστήσω την οικογένειά μου για την αμέριστη υποστήριξη και ψυχολογική ενθάρρυνση που μου έδωσε. Τέλος, ευχαριστώ όλους όσους συνέβαλαν έμμεσα στην ολοκλήρωση αυτής της εργασίας και μοιράστηκαν τις γνώσεις και την εμπειρία τους μαζί μου.

Περίληψη (Abstract)

(English)

The subject of my thesis was to develop a smart online personalized travel agency as a web application, using Spring Boot, specifically Spring Security and Data JPA for the back-end utilization, React.js for the front-end and a MySQL database.

The aim is to create a platform that allows users to book travel packages or hotels, leave reviews, view their personal statistics, and receive personalized recommendations based on their interests. Additionally, the system distinguishes between user roles, with both user and admin functionalities. The admin has access to extra functionalities, such as adding or deleting packages and users, as well as viewing statistics on a dashboard.

Key Words: Web Application, Spring Boot, React.js MySQL, Personalization, User Roles, User Statistics

(Ελληνικά)

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη ενός smart online εξατομικευμένου τουριστικού γραφείου ως web εφαρμογή, χρησιμοποιώντας Spring Boot, και συγκεκριμένα Spring Security και Data JPA για την υλοποίηση του back-end, React.js για το front-end και μία βάση MySQL.

Ο στόχος είναι η δημιουργία μιας πλατφόρμας που επιτρέπει στους χρήστες να κλείνουν ταξιδιωτικά πακέτα ή ξενοδοχεία, να κάνουν αξιολογήσεις, να βλέπουν τα προσωπικά τους στατιστικά και να λαμβάνουν εξατομικευμένες προτάσεις βάσει των ενδιαφερόντων τους. Επιπλέον, υπάρχει διάκριση των ρόλων χρηστών, με λειτουργίες τόσο για χρήστες όσο και για διαχειριστές. Ο διαχειριστής έχει επιπλέον πρόσβαση σε λειτουργίες, όπως η προσθήκη ή διαγραφή πακέτων και χρηστών, καθώς και η προβολή στατιστικών σε ένα πίνακα ελέγχου.

Λέξεις Κλειδιά: Web Εφαρμογή, Spring Boot, React.js MySQL, Εξατομίκευση, Ρόλοι Χρηστών, Στατιστικά Χρηστών

Πίνακας περιεχομένων

Copyright ©.....	2
Ευχαριστίες.....	3
Περίληψη (Abstract)	4
Εισαγωγή.....	7
Ανάλυση Αρχιτεκτονικής	8
Στήσιμο Εφαρμογής.....	10
Σχεδιασμός του back-end.....	10
Σχεδιασμός της βάσης δεδομένων	10
Spring Data JPA	11
Υλοποίηση Ασφάλειας.....	12
Spring Security	12
JSON Web Token (JWT)	12
Εφαρμογή του JWT στο Spring Security.....	12
Κλάση User.....	13
Κλάση Role	13
Κλάση JwtTokenProvider.....	14
Κλάση JwtAuthenticationFilter	14
Κλάση JwtAuthenticationEntryPoint.....	15
Κλάση CustomUserDetailsService.....	15
Κλάση SpringSecurityConfig.....	16
Κλάση AuthServiceImpl.....	16
Κλάση AuthController	17
Κλάση JwtAuthResponse.....	18
Διαγραμματική Αναπαράσταση του Authentication.....	18
Διάγραμμα βάσης δεδομένων	19
Επεξήγηση των Entities.....	20
Δομή του back-end.....	21
Επεξήγηση των annotations	22
Spring Boot Annotations	22
Spring Data JPA Annotations	24
Security Annotations	24
Lombok Annotations.....	24
JavaMailSender	25
Collaborative filtering	25
Σχεδιασμός του front-end	28

Βασικές έννοιες της React	28
Δημιουργία του React Project.....	28
Δομή του React Project	29
Δεδομένα από RapidAPI	30
Σχεδιασμός UI.....	31
Εγχειρίδιο χρήσης (για τον user)	32
Εγγραφή και σύνδεση	32
Αναζήτηση ξενοδοχείων.....	33
Κράτηση ταξιδιωτικών πακέτων	36
Προβολή δημοφιλών προορισμών.....	39
Εξατομικευμένα προτεινόμενα ταξιδιωτικά πακέτα.....	41
Αξιολογήσεις της εφαρμογής	41
Προβολή λεπτομερειών και στατιστικών χρήστη	42
Εγχειρίδιο χρήσης (για τον administrator)	46
Διαχείριση χρηστών.....	48
Διαχείριση ταξιδιωτικών πακέτων	49
Διαχείριση ιστορικού συναλλαγών	50
Επικοινωνία με subscribed users.....	50
Προβολή στατιστικών	51
Μελλοντικές επεκτάσεις της εφαρμογής	54
Συμπεράσματα.....	55
Βιβλιογραφία.....	56

Εισαγωγή

Στη σύγχρονη εποχή, η ψηφιακή τεχνολογία έχει επηρεάσει όλους τους τομείς της καθημερινής ζωής, συμπεριλαμβανομένου του τουρισμού. Οι ταξιδιώτες επιδιώκουν πλέον εξατομικευμένες εμπειρίες και εύκολη πρόσβαση σε υπηρεσίες που ανταποκρίνονται στα ενδιαφέροντα και τις προτιμήσεις τους.

Ο σκοπός της παρούσας εργασίας είναι να δημιουργηθεί μια πλατφόρμα που θα προσφέρει προσωποποιημένες υπηρεσίες στους χρήστες, βελτιώνοντας την εμπειρία τους και διευκολύνοντας τη διαδικασία κράτησης. Μέσα από την εφαρμογή, οι χρήστες μπορούν να κλείνουν ταξιδιωτικά πακέτα ή ξενοδοχεία και να λαμβάνουν προσωποποιημένες προτάσεις, ανάλογα με τα ενδιαφέροντά τους. Μπορούν ακόμα να δημοσιεύσουν αξιολογήσεις και κριτικές για την εφαρμογή και τα ταξιδιωτικά πακέτα που έχουν κλείσει, βοηθώντας άλλους χρήστες στην επιλογή των προσφερόμενων υπηρεσιών. Ακόμα, τους παρέχεται η δυνατότητα να δουν στατιστικά των κρατήσεων και των συναλλαγών τους.

Σημαντικό χαρακτηριστικό της εφαρμογής είναι η διάκριση των ρόλων χρηστών σε “user” και “admin”. Ο απλός χρήστης έχει πρόσβαση στις λειτουργίες που προαναφέρθηκαν. Από την πλευρά του ο διαχειριστής, ωστόσο, μπορεί να προσθέσει, να τροποποιήσει ή να διαγράψει πακέτα και χρήστες, να στείλει email σε κάποιον χρήστη, ή σε πολλούς χρήστες μαζί, καθώς και να παρακολουθεί στατιστικά που αφορούν την κατανομή της ηλικίας και του φύλου των χρηστών, τη δραστηριότητα τους στην εφαρμογή, καθώς και στατιστικά που αφορούν τον ρυθμό ανάπτυξης των χρηστών.

Η εφαρμογή υλοποιήθηκε ενσωματώνοντας έναν συνδυασμό τεχνολογιών. Από την πλευρά του back-end, χρησιμοποιήθηκε η γλώσσα Java, και συγκεκριμένα το framework Spring Boot με το Spring Data JPA για την αυτοματοποιημένη οργάνωση και διαχείριση των δεδομένων. Η χρήση, παράλληλα, του Spring Security, με το JWT (JSON Web Token) πρότυπο, επέτρεψαν την υλοποίηση μηχανισμών ασφάλειας στην εφαρμογή, όπως η αυθεντικοποίηση των χρηστών, η διαχείριση των ρόλων και ο έλεγχος πρόσβασης σε λειτουργίες. Για τον σχεδιασμό του front-end, χρησιμοποιήθηκε η βιβλιοθήκη React της JavaScript, για τη δημιουργία ενός πιο ευέλικτου και φιλικού προς τον χρήστη περιβάλλοντος, ενώ ως βάση δεδομένων χρησιμοποιήθηκε η MySQL για την αποδοτική αποθήκευση, ανάκτηση και διαχείριση των δεδομένων της εφαρμογής.

Η παρούσα εργασία αναλύει την ανάπτυξη της εφαρμογής, την υλοποίηση των επιμέρους τεχνολογιών και τον τρόπο με τον οποίο αυτές συνεργάζονται για την επίτευξη ενός αποδοτικού και φιλικού προς το χρήστη συστήματος.

Ανάλυση Αρχιτεκτονικής

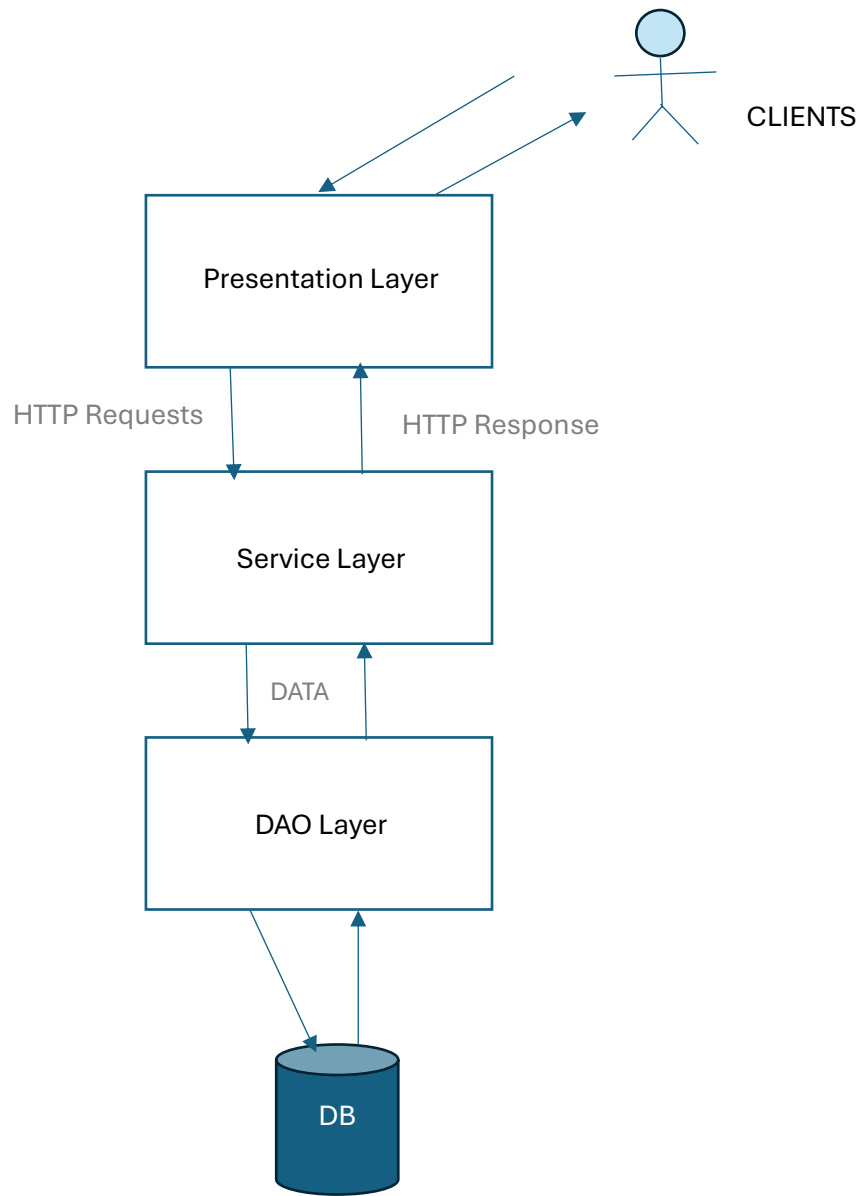
Η εφαρμογή έχει υλοποιηθεί χρησιμοποιώντας μία Three-Layer αρχιτεκτονική, με τα εξής βασικά επίπεδα: Το Presentation Layer, που αφορά το front-end της εφαρμογής με τη χρήση της React.js, και τα Service και DAO (Data Access Object) Layers, που αφορούν το back-end με τη χρήση του Spring Boot.

Το Presentation Layer είναι υπεύθυνο για την αλληλεπίδραση με τον τελικό χρήστη. Με τη χρήση της βιβλιοθήκης React της JavaScript παρέχεται ένα δυναμικό UI στον browser, επιτρέποντας στους χρήστες να αλληλεπιδρούν με την εφαρμογή με φυσικό και εύκολο τρόπο. Το επίπεδο αυτό επικοινωνεί με το Service Layer στέλνοντας στο back-end με τη χρήση της βιβλιοθήκης Axios HTTP αιτήματα.

Το Service Layer είναι το επίπεδο, το οποίο περιλαμβάνει τη business λογική της εφαρμογής. Διαχειρίζεται τα αιτήματα από το Presentation Layer με τη χρήση του Spring Boot server. Το Spring Security διασφαλίζει την ασφάλεια της εφαρμογής και η χρήση του JWT (JSON Web Token) επιτρέπει το authentication και authorization των χρηστών. Στο επίπεδο αυτό υλοποιούνται λειτουργίες που σχετίζονται με την επιτυχή σύνδεση των χρηστών, τη δημιουργία και διαχείριση των ταξιδιωτικών πακέτων και αξιολογήσεων, την παρακολούθηση της δραστηριότητας του χρήστη και άλλα. Το Service Layer επικοινωνεί με το Presentation Layer μέσω RESTful APIs και με το DAO Layer για την αποθήκευση και την ανάκτηση δεδομένων από τη βάση.

Το DAO Layer είναι υπεύθυνο για την πρόσβαση στα δεδομένα. Περιλαμβάνει τη σύνδεση με τη MySQL βάση και την εκτέλεση CRUD operations, με τη χρήση του Spring Data JPA. Η βάση δεδομένων περιλαμβάνει τα δεδομένα για τους χρήστες, τα ταξιδιωτικά πακέτα, τις κρατήσεις και αξιολογήσεις των χρηστών, καθώς και δεδομένα που αφορούν τη δραστηριότητά τους στην εφαρμογή.

Παρατίθεται, ακολούθως, η διαγραμματική αναπαράσταση της χρήσης της 3-Layer αρχιτεκτονικής στην εφαρμογή.



ΔΙΑΓΡΑΜΜΑ 1: Three-Layer Αρχιτεκτονική

Στήσιμο Εφαρμογής

Σχεδιασμός του back-end

Χρησιμοποιώντας την Ultimate Edition του IntelliJ IDEA, δημιουργήθηκε το νέο Spring Boot project. Επιλέχθηκαν τα εξής dependencies για την υλοποίηση της εργασίας:

- Spring Web: Παρέχει τα εργαλεία για τη δημιουργία των RESTful APIs.
- Spring Data JPA: Υποστηρίζει την αλληλεπίδραση με τη βάση δεδομένων μέσω της ORM τεχνολογίας.
- MySQL Driver: Επιτρέπει τη σύνδεση με τη βάση δεδομένων MySQL.
- Spring Security: Διαχειρίζεται την ασφάλεια της εφαρμογής, περιλαμβάνοντας αυθεντικοποίηση και εξουσιοδότηση.
- Lombok: Παρέχει μία απλοποίηση του κώδικα, με αυτοματοποιημένους getters, setters και constructors
- JWT: Υποστηρίζει τη δημιουργία και επαλήθευση JSON Web Tokens για την ασφάλεια της εφαρμογής.
- ModelMapper: Διευκολύνει τη μετατροπή δεδομένων μεταξύ διαφορετικών αντικειμένων (DTOs και Entities).
- Spring Boot Starter Mail: Υποστηρίζει την αποστολή email μέσω SMTP.

Σχεδιασμός της βάσης δεδομένων

Για τη δημιουργία της βάσης δεδομένων χρησιμοποιήθηκε η MySQL.

Χρησιμοποιώντας το MySQL Workbench, ένα δημοφιλές εργαλείο γραφικής διεπαφής για τη διαχείριση των βάσεων δεδομένων MySQL, δημιουργείται η βάση δεδομένων της εφαρμογής με το όνομα `travel_agency`, μέσα από το query `“create database travel_agency”`.

Έπειτα, στο αρχείο `application.properties` καθορίστηκαν οι βασικές ιδιότητες για τη σύνδεση με τη βάση δεδομένων, όπως το URL της βάσης, το όνομα χρήστη και ο κωδικός πρόσβασης, καθώς και ρυθμίσεις για την αυτόματη ενημέρωση των πινάκων της βάσης δεδομένων.

Έτσι, η εφαρμογή είναι πλέον έτοιμη να συνδεθεί με τη βάση δεδομένων επιτυχώς, εξασφαλίζοντας την ομαλή λειτουργία και υλοποίηση των ενεργειών δημιουργίας, αποθήκευσης, επεξεργασίας και ανάκτησης των δεδομένων.

Spring Data JPA

Το Spring Data JPA, μέρος της ευρύτερης οικογένειας Spring Data, μια ευκολότερη και πιο συνεκτική προσέγγιση στη διαχείριση της βάσης δεδομένων με τη χρήση του JPA (Java Persistence API). Διευκολύνει τη δημιουργία εφαρμογών με τροφοδοσία Spring που χρησιμοποιούν τεχνολογίες πρόσβασης δεδομένων.

Η εφαρμογή ενός DAO Layer για μια εφαρμογή μπορεί να είναι αρκετά περίπλοκη. Πρέπει να γραφτεί πολύς boilerplate κώδικας για να εκτελεστούν ακόμα και τα πιο απλά queries. Το Spring Data JPA επιτρέπει την αλληλεπίδραση με τη βάση δεδομένων μέσω απλών interface χωρίς να χρειάζεται να γραφτούν μεγάλες ποσότητες κώδικα SQL.

Κύρια Στοιχεία του Spring Data JPA

Entity Classes (Modal Layer)

Οι entity classes αντιστοιχούν στους πίνακες της βάσης δεδομένων. Κάθε entity κλάση αντιπροσωπεύει έναν πίνακα και κάθε πεδίο της κλάσης αντιπροσωπεύει μια στήλη του πίνακα. Αυτές οι κλάσεις ορίζονται με διάφορα annotations της JPA, όπως τα @Entity, @Table και @GeneratedValue, τα οποία επιτρέπουν στο Spring Data JPA να κάνει το σωστό mapping μεταξύ των κλάσεων και των δεδομένων της βάσης.

Repository Interfaces (DAO Layer)

Τα repository interfaces παρέχουν τις βασικές λειτουργίες CRUD (Create, Read, Update, Delete) χωρίς την ανάγκη για γραφή SQL κώδικα. Το Spring Data JPA παρέχει έτοιμες μεθόδους για τις πιο συνηθισμένες λειτουργίες, όπως η εύρεση, η αποθήκευση και η διαγραφή δεδομένων. Δύο βασικά στοιχεία για την υλοποίηση των παραπάνω είναι το annotation @Repository και η χρήση του JpaRepository. Το @Repository είναι ένα από τα annotations του Spring, που δηλώνει ότι η κλάση είναι ένα repository που διαχειρίζεται την πρόσβαση στη βάση δεδομένων. Το JpaRepository είναι μία ιδιαίτερη επέκταση του Repository, που περιέχει το πλήρες API των CrudRepository και PagingAndSortingRepository, υλοποιώντας τις βασικές CRUD λειτουργίες, αλλά και επιπλέον λειτουργίες ταξινόμησης ή προηγμένης αναζήτησης (findById(), findBy(), findAll() και άλλες παρόμοιες μέθοδοι που θα απαιτούσαν χρήση πολύπλοκων queries).

Υλοποίηση Ασφάλειας

Spring Security

Το Spring Security αποτελεί ένα framework, το οποίο παρέχει ολοκληρωμένη προστασία για τις εφαρμογές Java, εστιάζοντας στην αυθεντικοποίηση και εξουσιοδότηση των χρηστών. Με τον όρο αυθεντικοποίηση (authentication), ορίζουμε τη διαδικασία επιβεβαίωσης της ταυτότητας ενός χρήστη, δηλαδή η επαλήθευση των διαπιστευτηρίων του. Η εξουσιοδότηση (authorization) είναι η διαδικασία καθορισμού αν ένας χρήστης έχει τα κατάλληλα δικαιώματα για να εκτελέσει μια συγκεκριμένη ενέργεια ή να έχει πρόσβαση σε έναν συγκεκριμένο πόρο.

JSON Web Token (JWT)

Το JSON Web Token (JWT) είναι ένα ανοικτό πρότυπο (RFC 7519) που ορίζει έναν τρόπο για την ασφαλή μετάδοση πληροφοριών μεταξύ δύο μερών ως ένα JSON αντικείμενο. Οι πληροφορίες μπορούν να είναι επαληθευμένες και εμπιστευτικές με τη χρήση ψηφιακών υπογραφών. Ένα JWT αποτελείται από τρία μέρη, τα οποία διαχωρίζονται με τελείες (.) μεταξύ τους.

Το πρώτο μέρος είναι το Header, το οποίο περιέχει πληροφορίες για τον τύπο του token και τον αλγόριθμο υπογραφής που χρησιμοποιείται.

Το δεύτερο μέρος είναι το Payload, το οποίο περιέχει τα claims, δηλαδή πληροφορίες για τον χρήστη. Υπάρχουν τρεις τύποι claims: Τα registered claims, τα οποία δεν είναι υποχρεωτικά αλλά συνιστώνται, όπως ο iss (issuer), exp (expiration time), sub (subject), και aud (audience), τα public claims για την αποθήκευση δημόσιων πληροφοριών και τα private claims που δημιουργούνται για να μεταφέρουν πληροφορίες μεταξύ των μερών που συμφωνούν στη χρήση τους.

Το τελευταίο μέρος του JWT token είναι το signature, η υπογραφή δηλαδή. Το Signature χρησιμοποιείται για να διασφαλίσει ότι το token δεν έχει αλλοιωθεί. Δημιουργείται παίρνοντας το κωδικοποιημένο Header, το κωδικοποιημένο Payload, ένα secret key και τον αλγόριθμο που καθορίζεται στο Header, και τα υπογράφει.

Εφαρμογή του JWT στο Spring Security

Η εφαρμογή μου περιλαμβάνει την αυθεντικοποίηση των χρηστών βάσει ρόλων, συνδυάζοντας το Spring Security με JSON Web Tokens. Η διαδικασία που ακολουθείται θεωρητικά είναι η εξής: ο χρήστης υποβάλλει το username (ή το email) και το password του μέσω μίας φόρμας login. Το backend (χρησιμοποιώντας το Spring Security) επαληθεύει τα διαπιστευτήρια του χρήστη,

και εάν αυτά είναι έγκυρα, το σύστημα δημιουργεί ένα JWT token, το οποίο περιλαμβάνει τις πληροφορίες για τον χρήστη και τα δικαιώματά του. Το token, στη συνέχεια, αποστέλλεται στον πελάτη και αποθηκεύεται στο local storage. Έτσι, κάθε επόμενο αίτημα που κάνει ο πελάτης περιλαμβάνει το JWT token στο Authorization header (Bearer token). Το backend επαληθεύει το JWT token για να διασφαλίσει την αυθεντικότητα και την ακεραιότητα του. Εάν το token είναι έγκυρο, το σύστημα ανακτά τις πληροφορίες του χρήστη από το token, και ανάλογα με τις πληροφορίες που αυτό περιλαμβάνει, το σύστημα αποφασίζει αν ο χρήστης έχει τα απαραίτητα δικαιώματα για να εκτελέσει τη ζητούμενη ενέργεια ή αν έχει πρόσβαση στον ζητούμενο πόρο.

Κλάση User

Η κλάση User αναπαριστά τους χρήστες της εφαρμογής και περιέχει όλες τις απαραίτητες πληροφορίες για τον χρήστη, όπως το username, το email, το password και τον ρόλο του. Αυτές οι πληροφορίες χρησιμοποιούνται από το Spring Security για την αυθεντικοποίηση και την εξουσιοδότηση των χρηστών. Υπάρχουν, επίσης, και τα πεδία firstName, lastName, age, gender, registrationDate και subscribed, τα οποία δεν χρησιμοποιούνται στη διαδικασία του authentication, είναι ωστόσο χρήσιμα για την εξασφάλιση εξατομικευμένης εμπειρίας του χρήστη στην εφαρμογή και για την εξαγωγή στατιστικών.

Για την εγγραφή και σύνδεση του χρήστη χρησιμοποιείται η κλάση AuthServiceImpl, η οποία κάνει implement τη διεπαφή AuthService, και χρησιμοποιεί το UserRepository. Η κλάση AuthServiceImpl θα αναλυθεί περαιτέρω στη συνέχεια. Για τις υπόλοιπες CRUD λειτουργίες που αφορούν τα δεδομένα των χρηστών, χρησιμοποιείται η κλάση UserServiceImpl, η οποία υλοποιεί το interface UserService και χρησιμοποιεί, πάλι το UserRepository.

Κλάση Role

Η κλάση Role αναπαριστά τους ρόλους που μπορούν να αποδοθούν στους χρήστες, παίζοντας σημαντικό ρόλο στην εξουσιοδότηση των χρηστών. Το πεδίο name αντιπροσωπεύει το όνομα του ρόλου, το οποίο μπορεί να είναι είτε ROLE_USER, ή ROLE_ADMIN.

Χρησιμοποιείται στη διαδικασία της αυθεντικοποίησης μέσα από την κλάση AuthServiceImpl, η οποία υλοποιεί τη διεπαφή AuthService και χρησιμοποιεί το RoleRepository, ένα JpaRepository με τη μέθοδο findByName(), για την απόδοση του ρόλου ROLE_USER σε έναν καινούργιο χρήστη. Η κλάση AuthServiceImpl αναλύεται παρακάτω.

Κλάση JwtTokenProvider

Η κλάση JwtTokenProvider είναι υπεύθυνη για τη δημιουργία, επεξεργασία και επικύρωση των JWT tokens που χρησιμοποιούνται για την αυθεντικοποίηση των χρηστών στην εφαρμογή. Έχει σαν πεδία το jwtSecret, που είναι το secret key για την υπογραφή των JWT tokens, και το jwtExpirationDate που αποθηκεύει τη διάρκεια ζωής του token σε χιλιοστά του δευτερολέπτου.

Μέσα από τη μέθοδο generateToken() δημιουργείται ένα νέο JWT token. Η μέθοδος παίρνει το αντικείμενο αυθεντικοποίησης, το οποίο περιέχει τα στοιχεία του χρήστη, εξάγει το username του, καθορίζει μία ημερομηνία λήξης του token χρησιμοποιώντας την τρέχουσα ημερομηνία και δημιουργεί το token με τη χρήση του builder Jwts, υπογράφοντάς το με το secret key.

Η μέθοδος key() δημιουργεί το κλειδί που χρησιμοποιείται για την υπογραφή του token. Χρησιμοποιεί το jwtSecret και δημιουργεί το Key αντικείμενο με τον αλγόριθμο HMAC SHA-256. Η χρήση του αλγορίθμου αυτού προσφέρει υψηλό επίπεδο ασφάλειας, αποτρέποντας την παραποίηση των tokens

Η μέθοδος getUsername() εξάγει το όνομα χρήστη από το JWT token, χρησιμοποιώντας τον parser του Jwts για να αποκωδικοποιήσει το token και να πάρει τα claims. Η μέθοδος αυτή χρησιμοποιείται στο Authorization του χρήστη, επιτρέποντας στο σύστημα να αποφασίσει αν ο χρήστης έχει τα απαραίτητα permissions για να προσπελάσει έναν συγκεκριμένο πόρο.

Η μέθοδος, τέλος, validateToken() επικυρώνει το JWT token για να εξασφαλίσει ότι είναι έγκυρο. Χρησιμοποιεί τον parser του Jwts για να αποκωδικοποιήσει και να επικυρώσει το token. Αν το token δεν είναι έγκυρο ή έχει αλλοιωθεί, αυτή η μέθοδος θα αποτύχει και θα πετάξει εξαιρέσεις, διαφορετικά θα επιστρέψει true.

Κλάση JwtAuthenticationFilter

Η κλάση JwtAuthenticationFilter είναι ένα φίλτρο που χρησιμοποιείται για την αυθεντικοποίηση των αιτημάτων HTTP μέσω JWT tokens. Αυτή η κλάση εκτελείται μία φορά ανά αίτημα και είναι υπεύθυνη για την εξαγωγή, την επαλήθευση και την επεξεργασία των JWT tokens από τις HTTP κεφαλίδες των αιτημάτων.

Ελέγχει κάθε εισερχόμενο HTTP αίτημα για να βρει ένα JWT token στην κεφαλίδα Authorization. Αν βρεθεί και είναι έγκυρο, το φίλτρο επαληθεύει το token και θέτει τον χρήστη στο SecurityContext του Spring Security, επιτρέποντας έτσι την πρόσβαση στους προστατευμένους πόρους της εφαρμογής.

Χρησιμοποιεί την κλάση JwtTokenProvider, που αναλύθηκε παραπάνω και την UserDetailsService, η οποία χρησιμοποιείται για τη φόρτωση των λεπτομερειών του χρήστη βάσει του ονόματος χρήστη που εξάγεται από το JWT token.

Μέσω της μεθόδου `doFilterInternal()` εξάγεται το JWT token από την κεφαλίδα `Authorization` του αιτήματος, και αν είναι έγκυρο και έχει κείμενο, επικυρώνεται με τη βοήθεια της μεθόδου `validateToken()` του `JwtTokenProvider()` που περιεγράφηκε παραπάνω. Στη συνέχεια, αν το token είναι έγκυρο, εξάγεται το `username` του χρήστη και οι λεπτομέρειές του, και δημιουργείται ένα `UsernamePasswordAuthenticationToken` με τις λεπτομέρειες του χρήστη και τις αρχές εξουσιοδότησης (`authorities`). Τέλος, θέτει το `UsernamePasswordAuthenticationToken` στο `SecurityContextHolder`, το οποίο είναι το κεντρικό σημείο αποθήκευσης της κατάστασης της ασφάλειας (`authentication`) στο Spring Security, και καλείται η μέθοδος `doFilter` του `FilterChain` για να συνεχιστεί η επεξεργασία του αιτήματος από τα υπόλοιπα φίλτρα της αλυσίδας.

Η εξαγωγή του token από την κεφαλίδα του αιτήματος γίνεται μέσω της `getTokenFromRequest()`, η οποία ελέγχει αν η κεφαλίδα `Authorization` περιέχει κείμενο και αν ξεκινάει με το πρόθεμα "Bearer " και αν αυτή είναι σωστή, επιστρέφει το token αφαιρώντας το πρόθεμα "Bearer ".

Κλάση `JwtAuthenticationEntryPoint`

Η κλάση `JwtAuthenticationEntryPoint` είναι υπεύθυνη για τη διαχείριση των περιπτώσεων όπου ένας μη εξουσιοδοτημένος χρήστης προσπαθεί να προσπελάσει προστατευμένους πόρους στην εφαρμογή σας. Υλοποιεί τη διεπαφή `AuthenticationEntryPoint` και χρησιμοποιείται από το Spring Security για να στείλει μια κατάλληλη απόκριση όταν απαιτείται αυθεντικοποίηση.

Κλάση `CustomUserDetailsService`

Η `CustomUserDetailsService` είναι υπεύθυνη για την αναζήτηση των χρηστών στη βάση δεδομένων και την παροχή των στοιχείων αυτών στο Spring Security για αυθεντικοποίηση και εξουσιοδότηση. Αυτό επιτρέπει στο σύστημα να επαληθεύει τα διαπιστευτήρια του χρήστη και να φορτώνει τις αρμοδιότητες (`authorities`) που σχετίζονται με τον χρήστη.

Χρησιμοποιεί το `UserRepository` για την αναζήτηση των χρηστών βάσει `username` ή `email`, και αν βρει τον χρήστη, συλλέγει από το αντικείμενο τους ρόλους και τους μετατρέπει σε ένα `set` από `GrantedAuthority` αντικείμενα. Κάθε ρόλος αντιπροσωπεύεται από ένα `SimpleGrantedAuthority` που περιέχει το όνομα του ρόλου. Τέλος, δημιουργείται ένα αντικείμενο `User` του `org.springframework.security.core.userdetails.User`, το οποίο περιέχει το όνομα χρήστη, τον κωδικό πρόσβασης και τις αρμοδιότητες (`authorities`) του χρήστη. Αυτό το αντικείμενο είναι που χρησιμοποιεί το Spring Security για να επιβεβαιώσει την αυθεντικότητα του χρήστη.

Η κλάση `CustomUserDetails` χρησιμοποιείται στην κλάση `JwtAuthenticationFilter` που αναλύθηκε παραπάνω, για τη φόρτωση των λεπτομερειών του χρήστη από το token.

Κλάση `SpringSecurityConfig`

Η κλάση `SpringSecurityConfig` είναι η κεντρική κλάση για τη ρύθμιση της αυθεντικοποίησης και της εξουσιοδότησης των χρηστών. Χρησιμοποιεί τη μέθοδο `SecurityFilterChain`, η οποία ξεκινά με την απενεργοποίηση της προστασίας CSRF (Cross-Site Request Forgery). Αυτή η ρύθμιση απενεργοποιεί έναν επιπλέον μηχανισμό προστασίας που συνήθως ενσωματώνεται για την αποφυγή επιθέσεων τύπου CSRF, επιλέγοντας να εστιάσει στη χρήση JWT tokens για την αυθεντικοποίηση και όχι σε αυτήν την προστασία. Ενώ η προστασία CSRF είναι σημαντική για τις εφαρμογές web με φόρμες, η απενεργοποίησή της σε περιβάλλοντα API που χρησιμοποιούν JWT είναι αποδεκτή.

Στη συνέχεια, διαμορφώνεται η πολιτική εξουσιοδότησης με τη μέθοδο `AuthorizeHttpRequests`. Για κάθε endpoint ορίζεται στη μέθοδο `requestMatchers` το url του και ο τύπος πρόσβασης, ορίζοντας αν όλοι οι χρήστες ή μόνο οι αυθεντικοποιημένοι χρήστες με κάποιον ρόλο μπορούν να έχουν πρόσβαση.

Σημαντικό στοιχείο της κλάσης είναι η ενσωμάτωση του `JwtAuthenticationFilter`, το οποίο τοποθετείται πριν από το `UsernamePasswordAuthenticationFilter` στο φίλτρο αλυσίδας ασφαλείας. Με αυτόν τον τρόπο, το σύστημα ασφαλείας μπορεί να διαχειριστεί την αυθεντικοποίηση και τις αρμοδιότητες του χρήστη χωρίς επιπλέον αιτήματα προς τη βάση δεδομένων. Επιπλέον χρησιμοποιείται το `JwtAuthenticationEntryPoint` για τον χειρισμό εξαιρέσεων και ο `BCryptPasswordEncoder`, ο οποίος είναι υπεύθυνος για την κρυπτογράφηση των κωδικών πρόσβασης πριν την αποθήκευσή τους στη βάση δεδομένων. Τέλος, ορίζεται το `Bean AuthenticationManager`, το οποίο επιτρέπει την επικύρωση των στοιχείων σύνδεσης των χρηστών και την επιστροφή ενός `Authentication` αντικειμένου που περιγράφει την κατάσταση της αυθεντικοποίησης.

Κλάση `AuthServiceImpl`

Η κλάση αυτή, η οποία κάνει implement το interface `AuthService`, είναι αυτή η οποία χειρίζεται τις διαδικασίες εγγραφής και σύνδεσης του χρήστη.

Κατά την εγγραφή (`register`) νέου χρήστη, η κλάση ελέγχει αν το όνομα χρήστη ή το email που παρέχεται από τον χρήστη κατά την εγγραφή είναι ήδη καταχωρημένα στη βάση δεδομένων. Εάν εντοπιστεί ότι υπάρχουν ήδη, γίνεται throw ένα `APIException` με κωδικό κατάστασης 400. Αν τα δεδομένα είναι μοναδικά, δημιουργείται ένα νέο αντικείμενο `User`, κρυπτογραφείται ο κωδικός πρόσβασης με τη βοήθεια του `PasswordEncoder`, και αποδίδεται στον χρήστη ο

ρόλος ROLE_USER από το RoleRepository. Τέλος, το νέο αντικείμενο χρήστη αποθηκεύεται στη βάση δεδομένων και επιστρέφεται μήνυμα επιβεβαίωσης.

Στη διαδικασία της σύνδεσης (login), η κλάση αναλαμβάνει την αυθεντικοποίηση του χρήστη μέσω της μεθόδου authenticate() του AuthenticationManager, η οποία ελέγχει τα στοιχεία σύνδεσης που παρέχονται από τον χρήστη μέσω του LoginDto. Εφόσον τα στοιχεία είναι σωστά, το AuthenticationManager δημιουργεί ένα αντικείμενο Authentication, το οποίο αποθηκεύεται στο SecurityContextHolder για να καταγραφούν οι λεπτομέρειες της τρέχουσας αυθεντικοποίησης. Έπειτα, χρησιμοποιείται ο JwtTokenProvider για τη δημιουργία ενός JWT token που περιέχει το όνομα χρήστη και άλλες σχετικές πληροφορίες. Αυτό το token επιστρέφεται μαζί με το ρόλο του χρήστη μέσω ενός αντικειμένου JwtAuthResponse, το οποίο περιλαμβάνει το JWT token και τον ρόλο του χρήστη για περαιτέρω χρήση στην εφαρμογή.

Κλάση AuthController

Η κλάση AuthController είναι αυτή που διαχειρίζεται τα HTTP αιτήματα από τον χρήστη για την εγγραφή και τη σύνδεσή του. Χρησιμοποιεί την AuthService, η οποία με τη σειρά της χρησιμοποιεί τα αντικείμενα και τις μεθόδους του Spring Security για την υλοποίηση της ασφάλειας.

Στη μέθοδο register(), η οποία είναι προσβάσιμη μέσω του endpoint /api/auth/register, λαμβάνεται ένα αντικείμενο RegisterDto από το σώμα της αίτησης. Αυτό το DTO (Data Transfer Object) περιέχει τις πληροφορίες που είναι απαραίτητες για την εγγραφή ενός νέου χρήστη, όπως το όνομα, το επώνυμο, το email, το όνομα χρήστη και τον κωδικό πρόσβασης. Η μέθοδος καλεί την register μέθοδο της AuthService, η οποία δημιουργεί μια νέα εγγραφή στη βάση δεδομένων. Αν η εγγραφή είναι επιτυχής, επιστρέφεται ένα μήνυμα επιβεβαίωσης με HTTP status 201 Created, το οποίο ενημερώνει τον χρήστη ότι η εγγραφή του πραγματοποιήθηκε με επιτυχία.

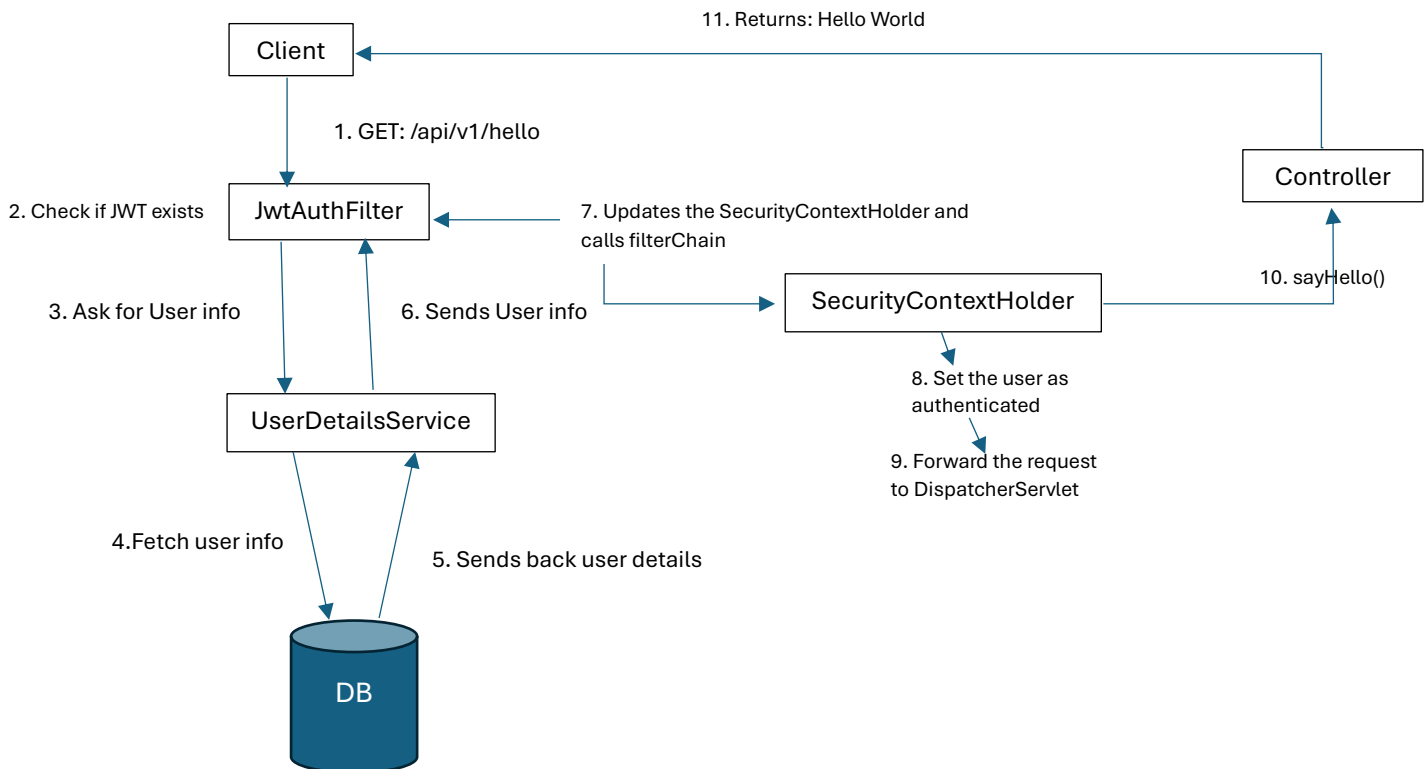
Στη μέθοδο login(), μέσω του endpoint /api/auth/login, λαμβάνεται ένα αντικείμενο LoginDto από το σώμα της αίτησης. Αυτό το DTO περιλαμβάνει το όνομα χρήστη ή το email και τον κωδικό πρόσβασης για την αυθεντικοποίηση του. Η μέθοδος καλεί τη login μέθοδο της AuthService, η οποία ελέγχει τα στοιχεία σύνδεσης και, εάν αυτά είναι σωστά, δημιουργεί ένα JWT token. Το token και ο ρόλος του χρήστη επιστρέφονται ως JwtAuthResponse με HTTP status 200 OK.

Κλάση JwtAuthResponse

Είναι ένα απλό Data Transfer Object (DTO) για την σύνδεση στην εφαρμογή. Παρέχει το JWT token που επιτρέπει την αυθεντικοποίηση των αιτήσεων του χρήστη και περιλαμβάνει επιπλέον πληροφορίες για τον ρόλο του χρήστη και τον τύπο του token. Ο ρόλος της είναι να μεταφέρει τα απαραίτητα δεδομένα στον πελάτη για τη συνεχή αλληλεπίδραση με το backend μέσω του JWT token.

Η κλάση JwtAuthResponse μεταφέρει δεδομένα μεταξύ του AuthController και του πελάτη (client) μετά την επιτυχή αυθεντικοποίηση του τελευταίου. Περιέχει τα πεδία accesstoken, δηλαδή το JWT token που παράγεται κατά τη σύνδεση του χρήστη, το tokenType, με προκαθορισμένη τιμή “Bearer“ και role, με την τιμή ROLE_USER ή ROLE_ADMIN, ανάλογα με τον ρόλο του χρήστη.

Διαγραμματική Αναπαράσταση του Authentication

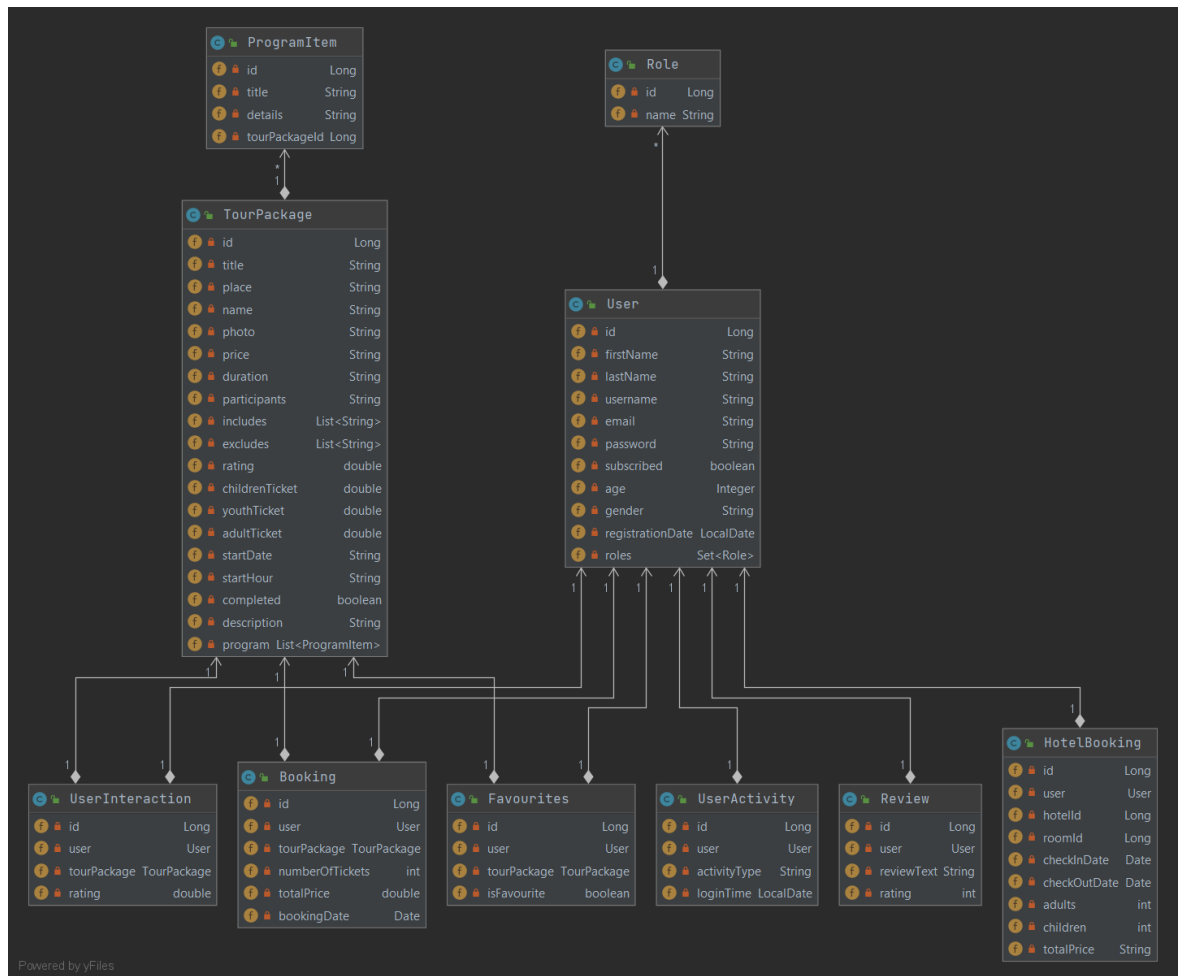


ΔΙΑΓΡΑΜΜΑ 2: Διαδικασία Authentication

Διάγραμμα βάσης δεδομένων

Για τον σχεδιασμό του διαγράμματος της βάσης δεδομένων, επιλέχθηκε το διάγραμμα τάξεων UML. Τα UML διαγράμματα χρησιμοποιούνται ευρέως για τη μοντελοποίηση συστημάτων και βοηθούν στην κατανόηση της βάσης δεδομένων.

Το παρακάτω διάγραμμα UML δημιουργήθηκε αυτόματα από το IntelliJ IDEA. Περιλαμβάνει τους πίνακες της βάσης, τα πεδία τους, καθώς και τις μεταξύ τους σχέσεις, ενώ δεν επιλέχθηκε να φαίνονται οι getters, οι setters και οι constructors :



ΔΙΑΓΡΑΜΜΑ 3: Διάγραμμα UML βάσης δεδομένων

Επεξήγηση των Entities

Στο παραπάνω διάγραμμα απεικονίζονται όλες οι οντότητες της εφαρμογής. Ακολουθεί μία συνοπτική επεξήγηση της κάθε μίας:

User: αναπαριστά τους χρήστες της εφαρμογής.

Role: αναπαριστά τον ρόλο ενός χρήστη.

TourPackage: αναπαριστά τα ταξιδιωτικά πακέτα.

ProgramItem: Αναπαριστά τα επιμέρους στοιχεία ενός προγράμματος για ένα ταξιδιωτικό πακέτο.

Booking: Αναπαριστά τις κρατήσεις που κάνουν οι χρήστες για ένα ταξιδιωτικό πακέτο.

Favorites: Αναπαριστά τα αγαπημένα ταξιδιωτικά πακέτα των χρηστών.

UserInteraction: Αναπαριστά τις αλληλεπιδράσεις των χρηστών με τα ταξιδιωτικά πακέτα. Χρησιμοποιείται για την προβολή προτεινόμενων ταξιδιωτικών πακέτων σε κάθε χρήστη, με τη χρήση του collaborative filtering (θα εξηγηθεί παρακάτω).

HotelBooking: Αναπαριστά τις κρατήσεις που κάνουν οι χρήστες για ένα ξενοδοχείο.

UserActivity: Αναπαριστά τις δραστηριότητες των χρηστών στην εφαρμογή. Χρησιμοποιείται για την εξαγωγή στατιστικών στοιχείων των χρηστών.

Review: Αναπαριστά τις αξιολογήσεις των χρηστών για την εφαρμογή.

Δομή του back-end

Το Spring Boot project της εφαρμογής ακολουθεί μία clean αρχιτεκτονική, διαχωρίζοντας τα components σε packages. Ακολουθεί μία αναλυτική περιγραφή των packages που δομούν το project και η επεξήγηση του ρόλου του καθενός.

Το package Entity, περιέχει τα entities, που όπως εξηγήθηκε παραπάνω, αντιπροσωπεύουν τα μοντέλα δεδομένων της εφαρμογής. Αυτές οι κλάσεις συνδέονται με τους πίνακες της βάσης δεδομένων και ο ρόλος τους είναι η αναπαράσταση των δεδομένων.

Το package dto περιέχει τις κλάσεις Data Transfer Object, οι οποίες χρησιμοποιούνται για τη μεταφορά δεδομένων μεταξύ των διαφόρων επιπέδων της εφαρμογής και για την ανταλλαγή δεδομένων με τον πελάτη. Ο ρόλος τους είναι η μεταφορά δεδομένων.

Το package controller περιέχει τις κλάσεις που χειρίζονται τα αιτήματα HTTP και τις αποκρίσεις. Αυτές οι κλάσεις είναι υπεύθυνες για την αλληλεπίδραση με τον χρήστη της εφαρμογής μέσω API endpoints. Ο ρόλος τους είναι η επικοινωνία με τον πελάτη (client), η διαχείριση των εισερχομένων αιτημάτων και η επιστροφή των κατάλληλων απαντήσεων. Οι controller χρησιμοποιούν τις υπηρεσίες (services) για να εκτελούν τις λειτουργίες της εφαρμογής και να αλληλεπιδρούν με τη βάση δεδομένων μέσω των repository interfaces. Σημαντικά annotations που χρησιμοποιούνται στους controllers είναι τα @RestController, @Autowired, @PostMapping, @GetMapping, @DeleteMapping και άλλα, τα οποία είναι απαραίτητα για τη διαχείριση των HTTP requests και την επιστροφή των δεδομένων σε JSON μορφή.

Το package service περιέχει τις κλάσεις που είναι υπεύθυνες για τη λογική της εφαρμογής και συνδυάζουν τις λειτουργίες των repository interfaces. Το service layer χρησιμοποιείται για την εκτέλεση κανόνων και σύνθετων λειτουργιών, χρησιμεύοντας ως γέφυρα μεταξύ των controllers και των repositories. Για παράδειγμα, στο service layer μπορεί να ληφθούν οι αποφάσεις για το πώς θα επεξεργαστούν τα δεδομένα, πότε θα αποθηκευτούν ή θα ενημερωθούν, και ποια δεδομένα θα ανακτηθούν από τη βάση. Το package service περιέχει τις διεπαφές για τα services, ενώ η υλοποίηση τους γίνεται στις κλάσεις που περιέχονται στο υπο-package serviceImpl.

Το package repository περιέχει τις κλάσεις που αλληλεπιδρούν με τη βάση δεδομένων. Αυτές οι κλάσεις επεκτείνουν το JpaRepository και έχουν ως ρόλο την πρόσβαση και διαχείριση δεδομένων από τη βάση δεδομένων. Το JpaRepository προσφέρει έτοιμες μεθόδους για βασικές CRUD λειτουργίες και υποστηρίζει τη δημιουργία προσαρμοσμένων queries, διευκολύνοντας έτσι την αλληλεπίδραση με τη βάση δεδομένων και μειώνοντας τον κώδικα που πρέπει να γραφτεί χειροκίνητα.

Το `package config` περιέχει τις κλάσεις που είναι υπεύθυνες για τη διαμόρφωση της ασφάλειας της εφαρμογής περιλαμβάνοντας την κλάση `SpringSecurityConfig`.

Το `package security` περιέχει τις κλάσεις που σχετίζονται με την ασφάλεια της εφαρμογής, όπως η διαχείριση της αυθεντικοποίησης και της εξουσιοδότησης. Περιέχει τις κλάσεις `JwtTokenProvider`, `JwtAuthenticationFilter`, `CustomUserDetailsService` και `JwtAuthenticationEntryPoint`.

Το `package exception` περιέχει τις κλάσεις που χειρίζονται τις εξαιρέσεις που μπορεί να προκύψουν κατά τη διάρκεια της εκτέλεσης της εφαρμογής.

Το `package mapper` περιέχει τις κλάσεις που μετατρέπουν τα `entities` σε `DTO` και αντίστροφα. Αυτό βοηθά στον διαχωρισμό των μοντέλων δεδομένων από τα δεδομένα που μεταφέρονται.

Το `package utils`, τέλος, περιλαμβάνει τη βοηθητική κλάση `PasswordEncoderImpl` που χειρίζεται την κωδικοποίηση των `passwords`.

Επεξήγηση των annotations

Τα `annotations` στο `Spring Boot` παρέχουν έναν τρόπο για να δηλώνουμε μεταδεδομένα στο επίπεδο της κλάσης ή της μεθόδου, διευκολύνοντας τη διαμόρφωση και τη διαχείριση των εφαρμογών. Μέσα από τα `annotations`, μπορούμε να καθορίσουμε πώς θα συμπεριφέρονται τα διάφορα μέρη της εφαρμογής χωρίς να χρειάζεται να γράφουμε πολύπλοκο και επαναλαμβανόμενο κώδικα διαμόρφωσης. Ακολουθεί η επεξήγηση των σημαντικότερων `annotations` της εφαρμογής.

Spring Boot Annotations

@SpringBootApplication

Χρησιμοποιείται στην κύρια κλάση της εφαρμογής για να δηλώσει ότι πρόκειται για μια `Spring Boot` εφαρμογή. Περιλαμβάνει τα `annotations@Configuration`, `@EnableAutoConfiguration`, και `@ComponentScan` συνδυασμένα.

@Component

Δηλώνει μια κλάση ως `Spring Component`. Αυτή η κλάση ανιχνεύεται αυτόματα και προστίθεται στο `application context` του `Spring`.

@Service

Ειδική μορφή του `@Component` που χρησιμοποιείται για τις `service classes`.

@Repository

Ειδική μορφή του @Component που χρησιμοποιείται για τις repository classes. Δηλώνει ότι η κλάση είναι υπεύθυνη για την αλληλεπίδραση με τη βάση δεδομένων.

@RestController

Συνδυασμός των @Controller και @ResponseBody. Δηλώνει ότι οι μέθοδοι μιας κλάσης θα επιστρέφουν JSON δεδομένα.

@RequestMapping

Καθορίζει τη διαδρομή (URL path) για τα αιτήματα που θα αντιστοιχούν στις μεθόδους της κλάσης ή της μεθόδου. Χρησιμοποιείται για την αντιστοίχιση αιτημάτων HTTP σε συγκεκριμένες μεθόδους του controller και μπορεί να ορίσει τη μέθοδο του αιτήματος (GET, POST, PUT, DELETE).

@GetMapping

Είναι μία εξειδικευμένη μορφή του @RequestMapping που δηλώνει ότι η μέθοδος του controller πρέπει να επεξεργαστεί HTTP GET αιτήματα. Χρησιμοποιείται για την ανάκτηση δεδομένων από τον server.

@PostMapping

Είναι μια εξειδικευμένη μορφή του @RequestMapping που δηλώνει ότι η μέθοδος του controller πρέπει να επεξεργαστεί HTTP POST αιτήματα. Χρησιμοποιείται για τη δημιουργία νέων δεδομένων.

@PutMapping

Αντίστοιχα με τα προηγούμενα, για την ενημέρωση ή αντικατάσταση δεδομένων.

@DeleteMapping

Αντίστοιχα με τα προηγούμενα, για τη διαγραφή δεδομένων.

@PathVariable

Δηλώνει ότι μία παράμετρος της μεθόδου αντιστοιχεί σε ένα μέρος του URL path. Χρησιμοποιείται για την εξαγωγή δεδομένων από URL path.

@RequestParam

Χρησιμοποιείται κυρίως για την ανάκτηση δεδομένων που αποστέλλονται ως παράμετροι σε αιτήματα GET.

@RequestBody

Χρησιμοποιείται για να δηλώσει ότι το σώμα της αίτησης (request body) θα πρέπει να μετατραπεί σε αντικείμενο της κλάσης που καθορίζεται ως παράμετρος της μεθόδου.

Spring Data JPA Annotations

@Entity

Δηλώνει μια κλάση ως οντότητα που θα αντιπροσωπεύει έναν πίνακα στη βάση δεδομένων.

@Table

Χρησιμοποιείται για να καθορίσει το όνομα του πίνακα στη βάση δεδομένων που αντιστοιχεί σε μια οντότητα.

@Id

Δηλώνει το primary key μιας οντότητας.

@GeneratedValue

Καθορίζει τη στρατηγική για την αυτόματη δημιουργία τιμών για το primary key.

@Column

Χρησιμοποιείται για να καθορίσει τις ιδιότητες μιας στήλης στον πίνακα της βάσης δεδομένων.

@ManyToMany, @OneToMany, @ManyToOne, @OneToOne

Καθορίζουν τις σχέσεις μεταξύ των οντοτήτων.

@JoinTable

Χρησιμοποιείται για να καθορίσει τον πίνακα ενδιάμεσων συνδέσεων σε μια σχέση many-to-many.

Security Annotations

@EnableMethodSecurity

Ενεργοποιεί την ασφάλεια σε επίπεδο μεθόδου, επιτρέποντας τη χρήση annotations όπως @PreAuthorize για τον έλεγχο της πρόσβασης σε μεθόδους.

@PreAuthorize

Χρησιμοποιείται για να καθορίσει ποιος έχει πρόσβαση σε μια μέθοδο βάσει ρόλων ή άλλων χαρακτηριστικών.

Lombok Annotations

@Getter

Δημιουργεί αυτόματα getters για όλα τα πεδία της κλάσης.

@Setter

Δημιουργεί αυτόματα setters για όλα τα πεδία της κλάσης.

@NoArgsConstructor

Δημιουργεί έναν κατασκευαστή χωρίς παραμέτρους.

@AllArgsConstructor

Δημιουργεί έναν κατασκευαστή με παραμέτρους για όλα τα πεδία της κλάσης.

JavaMailSender

Στην εφαρμογή έχει προστεθεί λειτουργία αποστολής αυτοματοποιημένων email, για παράδειγμα όταν ένας χρήστης κάνει subscribe, ή κλείνει ένα ταξιδιωτικό πακέτο ή ένα ξενοδοχείο. Επίσης ο administrator της εφαρμογής μπορεί να επιλέξει να στείλει ομαδικά ή ατομικά email σε χρήστες.

Για τον λόγο αυτό χρησιμοποιείται η υπηρεσία JavaMailSender του Spring Boot. Αφού προστέθηκε το κατάλληλο dependency “spring-boot-starter-mail” στο αρχείο pom.xml, πρέπει να προστεθούν και οι απαραίτητες ρυθμίσεις στο αρχείο application.properties για τον email server που θα χρησιμοποιηθεί. Αυτές οι ρυθμίσεις περιλαμβάνουν τις πληροφορίες για τον SMTP server, το username, το password, καθώς και τις απαιτούμενες ιδιότητες για την ασφάλεια και τον έλεγχο ταυτότητας.

Έπειτα, δημιουργήθηκε η κλάση EmailSenderServiceImpl, η οποία υλοποιεί τη διεπαφή EmailSenderService, και περιέχει την μέθοδο sendEmail(), η οποία δημιουργεί ένα αντικείμενο SimpleMailMessage, θέτοντας τον αποστολέα, τον παραλήπτη, το θέμα και το κυρίως κείμενο.

Η υπηρεσία EmailSenderService χρησιμοποιείται, από τον EmailController, για την επιτυχή αποστολή, τελικά, του email.

Collaborative filtering

Το Collaborative filtering είναι μια δημοφιλής μέθοδος προτάσεων που χρησιμοποιείται σε πολλές εφαρμογές για την εξατομίκευση περιεχομένου, όπως προτάσεις προϊόντων, ταινιών, και ταξιδιωτικών πακέτων. Στην περίπτωση της travel agency εφαρμογής μου, το collaborative filtering χρησιμοποιείται για την εμφάνιση προτεινόμενων ταξιδιωτικών πακέτων σε κάθε χρήστη, βασισμένο στις προτιμήσεις και τις ενέργειες άλλων χρηστών. Τι είναι, όμως, το collaborative filtering και πώς χρησιμοποιήθηκε στην εφαρμογή?

Το Collaborative Filtering είναι μια τεχνική προτάσεων που βασίζεται στις αλληλεπιδράσεις των χρηστών με τα αντικείμενα του συστήματος. Η βασική ιδέα είναι ότι αν δύο χρήστες έχουν κοινές προτιμήσεις ή ενέργειες στο παρελθόν, είναι πιθανό να έχουν παρόμοιες προτιμήσεις και στο μέλλον. Υπάρχουν κυρίως δύο τύποι collaborative filtering, το user-based collaborative filtering και το item-based collaborative filtering.

Όσον αφορά το user-based collaborative filtering, το σύστημα προτάσεων βασίζεται στην ομοιότητα μεταξύ χρηστών. Ο βασικός κανόνας είναι ότι αν δύο χρήστες έχουν δείξει παρόμοιες προτιμήσεις για αντικείμενα στο παρελθόν, τότε η προτίμηση ενός χρήστη για ένα νέο αντικείμενο μπορεί να προβλεφθεί με βάση τις προτιμήσεις του άλλου χρήστη. Δεν θα αναλυθεί περαιτέρω αυτός ο τύπος, καθώς στην εφαρμογή μου έχει υλοποιηθεί το item-based collaborative filtering.

Σε αυτή την προσέγγιση, η έμφαση δίνεται στην ομοιότητα μεταξύ αντικειμένων, δηλαδή ταξιδιωτικών πακέτων. Το σύστημα προτάσεων βασίζεται στην υποθετική παραδοχή ότι αν δύο αντικείμενα έχουν εκτιμηθεί με παρόμοιο τρόπο από τους χρήστες, τότε η προτίμηση του χρήστη για ένα αντικείμενο μπορεί να επηρεάσει την προτίμησή του για το άλλο αντικείμενο. Η λειτουργία του αλγορίθμου αυτού βασίζεται στον υπολογισμό, αρχικά, της ομοιότητας μεταξύ αντικειμένων, χρησιμοποιώντας Pearson Correlation. Έπειτα γίνεται η ανάλυση του ποια αντικείμενα είναι παρόμοια με αυτά που έχουν ήδη αρέσει στον χρήστη, και, τέλος, χρησιμοποιώντας τις υπολογισμένες ομοιότητες, το σύστημα προτείνει νέα ταξιδιωτικά πακέτα σε χρήστες με βάση τις προτιμήσεις άλλων χρηστών.

Από άποψη κώδικα, η διαδικασία ξεκινάει με μία προ-επεξεργασία των δεδομένων. Η μέθοδος `preprocessData()` δέχεται τα δεδομένα των αξιολογήσεων των χρηστών για τα ταξιδιωτικά πακέτα, δημιουργώντας έναν πίνακα αλληλεπιδράσεων (`userPackageMatrix`) που αναπαριστά τις βαθμολογίες των πακέτων από τους χρήστες. Στη συνέχεια, ακολουθεί η κανονικοποίηση των δεδομένων για τη σύγκριση των προτιμήσεων των χρηστών, χωρίς την επιρροή από την προσωπική κλίμακα αξιολόγησης του καθενός.

Ακολουθεί ο υπολογισμός ομοιότητας των πακέτων. Μέσω της μεθόδου `calculatePackageSimilarityPearson()` δημιουργείται ένας πίνακας που καταγράφει πόσο παρόμοια είναι ζεύγη πακέτων με βάση τις αλληλεπιδράσεις των χρηστών. Ο υπολογισμός αυτός γίνεται με τη χρήση του Pearson Correlation.

Τέλος, μέσω της μεθόδου `calculatePredictedRating()` υπολογίζεται η προβλεπόμενη βαθμολογία ενός πακέτου για έναν χρήστη με βάση την ομοιότητα μεταξύ πακέτων και τις αλληλεπιδράσεις του χρήστη με παρόμοια πακέτα, και, μέσω της `recommendPackages()` μεθόδου, επιστρέφονται τα καλύτερα πακέτα ως προτεινόμενα στον εκάστοτε χρήστη.

Ο λόγος που επιλέχθηκε να εφαρμοστεί το item-based collaborative filtering έναντι του user-based collaborative filtering, έγκειται κυρίως στη σταθερότητα

των συστάσεων που μπορεί να παρέχει ο πρώτος τρόπος, καθώς αυτές βασίζονται στη σύγκριση των ταξιδιωτικών πακέτων και παραμένουν συνεπείς, ακόμα και αν νέοι χρήστες ενταχθούν στο σύστημα. Έτσι, το μέγεθος των χρηστών και οι αυξομειώσεις τους δεν επηρεάζουν το σύστημα, περιπλέκοντας τον αλγόριθμο, όπως θα γινόταν με τη χρήση του user-based collaborative filtering.

Σχεδιασμός του front-end

Βασικές έννοιες της React

Για το front-end της εφαρμογής επιλέχθηκε η χρήση της React, μίας δημοφιλούς JavaScript βιβλιοθήκης για την κατασκευή δυναμικών και επαναχρησιμοποιήσιμων διεπαφών χρήστη. Η React επικεντρώνεται στη δημιουργία UI components, τα οποία είναι επαναχρησιμοποιήσιμα, συνθετικά και αυτόνομα. Κάθε component είναι ένα μπλοκ κατασκευής του UI, το οποίο διαχειρίζεται τη δική του κατάσταση (state), το δικό του UI, και μπορεί να περιλαμβάνει άλλα components ή να χρησιμοποιείται σε άλλα μέρη της εφαρμογής.

Μία άλλη βασική αρχή της React είναι το Virtual Dom για την αποτελεσματική ανανέωση του UI. Αντί να τροποποιεί απευθείας το Document Object Model (DOM) του browser, η React δημιουργεί και διαχειρίζεται μια εικονική αναπαράσταση του DOM, η οποία συγκρίνεται με την πραγματική και επιτρέπει την μεγιστοποίηση της απόδοσης μέσω στοχευμένων ενημερώσεων.

Μία επιπλέον σημαντική έννοια της React είναι η χρήση JSX (JavaScript XML), μίας ευανάγνωστης σύνταξης που επιτρέπει την αναπαράσταση του UI, χρησιμοποιώντας συνδυασμό JavaScript και HTML στον ίδιο κώδικα.

Η μονοδιάστατη ροή δεδομένων είναι μια ακόμα θεμελιώδης έννοια στη React. Τα δεδομένα “κινούνται” από τα components υψηλότερου επιπέδου προς τα χαμηλότερα, μέσω των props (properties). Τέλος, κεντρικό σημείο στη λειτουργία των React components είναι η διαχείριση της κατάστασης (state) τους. Το state αναπαριστά την εσωτερική κατάσταση ενός component και μπορεί να αλλάξει με την αλληλεπίδραση του χρήστη ή άλλες ενέργειες.

Δημιουργία του React Project

Η διαδικασία δημιουργίας του React Project πραγματοποιήθηκε με τη χρήση του Visual Studio Code (VSCode). Αρχικά, εγκαταστάθηκαν το Node.js και το npm (Node Package Manager). Το Node.js είναι μια πλατφόρμα JavaScript για την κατασκευή server-side εφαρμογών, και το npm είναι το εργαλείο διαχείρισης πακέτων που επιτρέπει την εγκατάσταση εξαρτήσεων και βιβλιοθηκών για το project. Η εγκατάσταση του Node.js περιλαμβάνει αυτόματα και το npm.

Με την εγκατεστημένη έκδοση του Node.js και του npm, δημιούργησα ένα νέο React project χρησιμοποιώντας το εργαλείο Create React App και την εντολή στο terminal `npx create-react-app my-react-app`. Η εντολή αυτή διαμορφώνει το αρχικό react project με μία δομή αρχικών φακέλων και αρχείων που θα χρειαστούν.

Η εκκίνηση της web εφαρμογής γίνεται με την εντολή `npm start`. Στο πλαίσιο ανάπτυξης της εφαρμογής, απαιτείται η εγκατάσταση επιπλέον βιβλιοθηκών και

εργαλείων, όπως η react-router-dom, για την πλοήγηση μεταξύ των σελίδων και η βιβλιοθήκη axios για την εκτέλεση των HTTP αιτημάτων και την επικοινωνία με το back-end. Οι βιβλιοθήκες αυτές, όπως και άλλες που χρειάστηκαν, εγκαταστάθηκαν με την εντολή npm install.

Δομή του React Project

Η δομή του project αποτελεί βασικό στοιχείο για την οργάνωση και την ευκολία ανάγνωσης του κώδικα. Έτσι, χώρισα τον φάκελο src σε 4 κύριους υπο-φακέλους: components, images, routes και services.

Ο φάκελος components περιέχει όλα τα επαναχρησιμοποιήσιμα React components της εφαρμογής. Είναι ένας αρκετά σύνθετος φάκελος, καθώς περιέχει και άλλους φακέλους και υπο-φακέλους με όλα τα απαιτούμενα components, καθώς και τα αρχεία css της εφαρμογής. Οι φάκελοι αυτοί περιλαμβάνουν τα components που αφορούν το UI για τα ταξιδιωτικά πακέτα, τις φόρμες login και register, το navbar, τα διάφορα ξενοδοχεία, δωμάτια ξενοδοχείων, τις φόρμες για τις κρατήσεις και τις σελίδες πληρωμής, τις καρτέλες κριτικής των χρηστών, τις φόρμες επικοινωνίας και αξιολόγησης και άλλα, καθώς και το UI για το dashboard του admin, με τα διάφορα στατιστικά στοιχεία για τους χρήστες, τα ταξιδιωτικά πακέτα, τις κρατήσεις και τις πληρωμές.

Ο φάκελος images περιλαμβάνει όλα τα στατικά αρχεία εικόνων που χρησιμοποιούνται στην εφαρμογή, όπως background εικόνες, εικόνες αξιοθέατων, ξενοδοχείων και άλλα.

Ο φάκελος routes περιέχει όλα τα αρχεία που καθορίζουν τις διαδρομές (routes) της εφαρμογής. Κάθε route αντιπροσωπεύει μια διαφορετική σελίδα. Οι διαδρομές ρυθμίζονται με τη βοήθεια της βιβλιοθήκης react-router-dom.

Ο φάκελος services περιέχει όλα τα αρχεία που σχετίζονται με τις υπηρεσίες της εφαρμογής. Οι υπηρεσίες αυτές είναι υπεύθυνες για την επικοινωνία με το back-end, την ανάκτηση δεδομένων από API, την αποστολή αιτημάτων και την επεξεργασία των αποκρίσεων. Στα αρχεία js του φακέλου αυτού χρησιμοποιείται η βιβλιοθήκη axios, ορίζεται κάθε φορά το κατάλληλο REST_API_URL και έπειτα οι μέθοδοι για την επικοινωνία με τα endpoints και την ανάκτηση ή αποστολή δεδομένων.

Πρέπει να αναφερθεί ότι η εφαρμογή React “τρέχει” by default στο port 3000, ενώ το Spring Boot Project στο port 8080. Για τη σύνδεση, επομένως, του front-end με το back-end και την αποστολή των αιτημάτων από διαφορετικά domains, χρησιμοποιείται το annotation @CrossOrigin του Spring Boot.

Το Cross-Origin Resource Sharing (CORS) είναι ένας μηχανισμός ασφαλείας που επιτρέπει στους servers να ελέγχουν ποια domains μπορούν να έχουν πρόσβαση στους πόρους τους. Έτσι, σε κάθε Controller του Spring Boot έχει προστεθεί το

annotation @CrossOrigin("*"), το οποίο δηλώνει ότι επιτρέπονται αιτήματα από οποιοδήποτε domain προς τους πόρους της εφαρμογής, καθιστώντας δυνατή πλέον τη σύνδεση του front-end με το back-end.

Δεδομένα από RapidAPI

Στο πλαίσιο της εργασίας, έχω αξιοποιήσει διάφορα δεδομένα από δωρεάν APIs για να εμπλουτίσω της λειτουργίες της εφαρμογής μου. Έτσι, πέρα από τα δεδομένα για τα ταξιδιωτικά πακέτα, τα οποία έχω εγώ δημιουργήσει και εξάγονται από τη βάση δεδομένων, υπάρχουν δεδομένα για ξενοδοχεία, αξιοθέατα και εστιατόρια, τα οποία αντλήθηκαν από το RapidAPI, έναν ιστότοπο με δωρεάν και επί πληρωμή APIs.

Για την πρόσβαση σε αυτά τα δεδομένα, απαιτείται, αρχικά, η δημιουργία λογαριασμού στο RapidAPI. Έπειτα, αφού επιλέχθηκαν τα APIs που ενσωμάτωσα στην εφαρμογή, απέκτησα ένα API key για την πρόσβαση σε αυτά. Το αίτημα γίνεται με τη βιβλιοθήκη axios, προσθέτοντας το API key στα headers του αιτήματος. Μόλις επιστραφεί η απάντηση, τα δεδομένα είναι έτοιμα προς χρήση για την ενημέρωση του UI της εφαρμογής.

Χρησιμοποιήθηκαν τα ακόλουθα APIs:

Hotels.com Provider: Αξιοποιήθηκε για να εξαχθούν τα ξενοδοχεία για συγκεκριμένες πόλεις που υπάρχουν διαθέσιμες στην κεντρική σελίδα της εφαρμογής, καθώς και για την εξαγωγή διαθέσιμων δωματίων σε συγκεκριμένες ημερομηνίες και με ορισμένο αριθμό ατόμων.

Travel Advisor: Αξιοποιήθηκε για να εξαχθούν τα εστιατόρια και τα αξιοθέατα για συγκεκριμένες πόλεις που υπάρχουν διαθέσιμες στην κεντρική σελίδα της εφαρμογής.

Google API: Αξιοποιήθηκε για να εξαχθούν οι συντεταγμένες των πόλεων που υπάρχουν διαθέσιμες στην κεντρική σελίδα της εφαρμογής, προκειμένου να χρησιμοποιηθούν σαν παράμετροι στα παραπάνω APIs για τα αξιοθέατα και τα εστιατόρια.

Aridojito Booking: Αξιοποιήθηκε για να εξαχθούν τα ξενοδοχεία ύστερα από αναζήτηση των χρηστών για κάποια περιοχή, με ορισμένες ημερομηνίες και αριθμό ατόμων.

Σχεδιασμός UI

Το styling της εφαρμογής υλοποιήθηκε, στο μεγαλύτερο μέρος της, με χρήση απλής CSS. Ωστόσο, για τον σχεδιασμό του dashboard του administrator, το οποίο περιέχει πίνακες, γραφήματα και πίτες με στατιστικά στοιχεία χρησιμοποιήθηκε η Material-UI, μία δημοφιλής βιβλιοθήκη που ακολουθεί της αρχές σχεδίασης του Material Design της Google.

Έτσι, αξιοποιήθηκε το DataGrid component που δημιουργεί έναν πλήρως λειτουργικό πίνακα δεδομένων, υποστηρίζοντας sorting, filtering, pagination και πολλές άλλες λειτουργίες, καθώς και τα Table components που περιλαμβάνουν τα TableBody, TableCell, TableContainer, TableHead, TableRow, και Paper, και χρησιμοποιούνται επίσης για τον σχεδιασμό πινάκων δεδομένων.

Για την οπτικοποίηση δεδομένων με τη χρήση γραφημάτων, χρησιμοποιήθηκαν και οι βιβλιοθήκες Chart.js, Recharts και react-circular-progressbar, οι οποίες, σε συνδυασμό με το Material-UI, επιτυγχάνουν έναν επαγγελματικό και συνεκτικό σχεδιασμό.

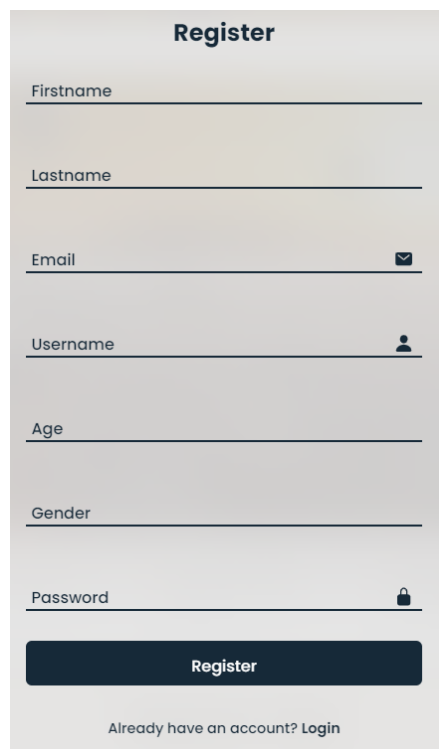
Εγχειρίδιο χρήσης (για τον user)

Εγγραφή και σύνδεση

Η ενότητα αυτή θα σας καθοδηγήσει στα πρώτα βήματα της εφαρμογής, την εγγραφή και τη σύνδεση.

Όταν επισκεφθείτε για πρώτη φορά τη σελίδα μας, θα δείτε στο πάνω δεξιά μέρος τις επιλογές Sign up και Sign in. Εάν είστε καινούργιος χρήστης, μπορείτε να ολοκληρώσετε την εγγραφή σας, πατώντας στο κουμπί Sign up, το οποίο θα σας εμφανίσει τη φόρμα εγγραφής. Συμπληρώστε το όνομα, το επίθετο και το email σας και έπειτα να δημιουργήσετε ένα username. Στη συνέχεια επιλέγετε το φύλο και την ηλικία σας και δημιουργείτε έναν ισχυρό κωδικό πρόσβασης. Μόλις πατήσετε το κουμπί Register, ολοκληρώνεται η εγγραφή σας και μεταβαίνετε αυτομάτως στη φόρμα Login, για να επιτευχθεί η σύνδεσή σας.

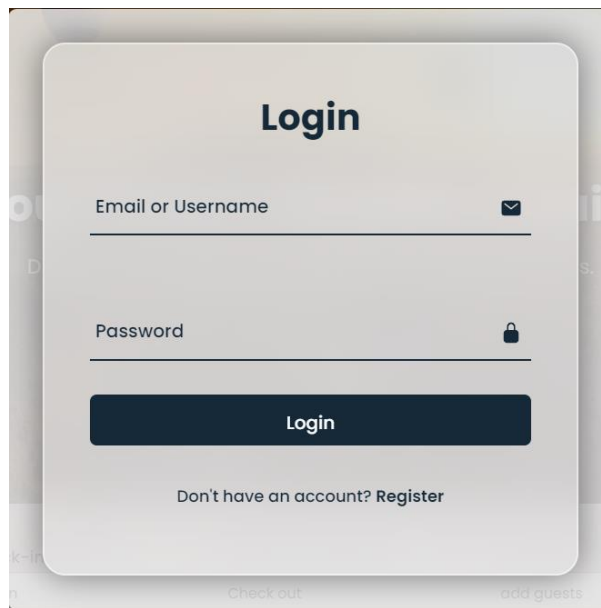
Εάν έχετε ήδη λογαριασμό, μπορείτε να μεταβείτε στη φόρμα σύνδεσης από τη φόρμα εγγραφής επιλέγοντας το κουμπί: “Already have an account? Login”, το οποίο θα σας ανακατευθύνει στη φόρμα σύνδεσης.

The image shows a registration form titled "Register". It contains several input fields: "Firstname", "Lastname", "Email" (with an envelope icon), "Username" (with a person icon), "Age", "Gender", and "Password" (with a lock icon). Below the fields is a dark blue "Register" button. At the bottom, there is a link that says "Already have an account? Login".

ΕΙΚΟΝΑ 1: Φόρμα εγγραφής

Στη φόρμα σύνδεσης, θα πρέπει να εισάγετε το email ή το username σας και το password. Μόλις εισάγετε τα στοιχεία σας, πατήστε το κουμπί Login και η σύνδεσή σας ολοκληρώνεται με επιτυχία.

Μπορείτε πάλι να μεταβείτε στη φόρμα εγγραφής από τη φόρμα σύνδεσης, επιλέγοντας το κουμπί “Don’t have an account? Register”, το οποίο θα σας ανακατευθύνει στη φόρμα εγγραφής για να δημιουργήσετε τον λογαριασμό σας.



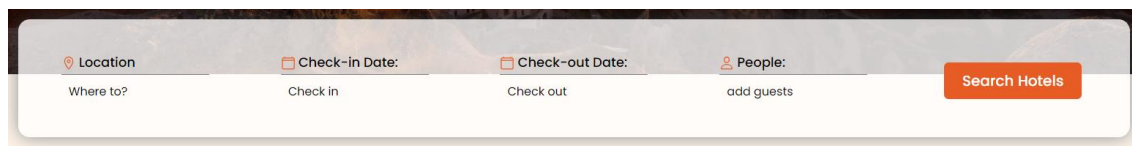
ΕΙΚΟΝΑ 2: Φόρμα Σύνδεσης

Αφού ολοκληρωθεί επιτυχώς η σύνδεσή σας, θα μεταβείτε στην κεντρική σελίδα της εφαρμογής, τις επιλογές της οποίας θα περιγράψουμε παρακάτω.

Αναζήτηση ξενοδοχείων

Στην κεντρική σελίδα της εφαρμογής, υπάρχει η δυνατότητα να αναζητήσετε διαμονή για ένα μέρος, στο οποίο θέλετε να ταξιδέψετε.

Έτσι, στη φόρμα που υπάρχει στην αρχή της κεντρικής σελίδας, εισάγετε το μέρος που θέλετε να επισκεφτείτε, τις ημερομηνίες άφιξης και αναχώρησης και τον αριθμό των ατόμων, και πατήστε το κουμπί Search Hotels.



ΕΙΚΟΝΑ 3: Φόρμα Αναζήτησης ξενοδοχείων

Θα ανακατευθυνθείτε, έπειτα, στη σελίδα με όλα τα διαθέσιμα για τις ημερομηνίες που ορίσατε ξενοδοχεία. Υπάρχει η δυνατότητα να φιλτράρετε τα αποτελέσματα με κριτήρια τις τιμές, την απόσταση από το κέντρο, ή τις

αξιολογήσεις και τις πολιτικές ακύρωσης, ώστε να λάβετε πιο στοχευμένα αποτελέσματα.

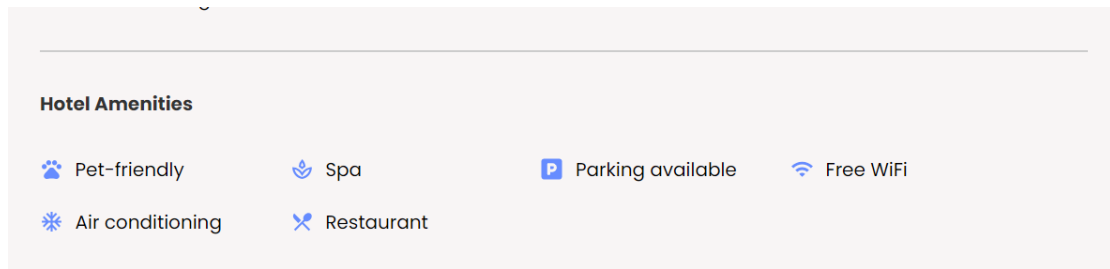
The screenshot shows a search results interface. On the left, there are filter sections: 'Filters' with a price range slider from \$0 to \$2000, 'Rating' with checkboxes for 5 Stars, 4+ Stars, 3+ Stars, 2+ Stars, and 1+ Star, 'Cancellation Policies' with checkboxes for Free Cancellation and Non-Refundable, 'Distance from center' with checkboxes for Less than 1 km, Less than 3 km, and Less than 5 km, and 'Feedback rating' with checkboxes for Excellent: 9+ and others. The main area displays two hotel cards. The first card is for 'Sea Containers London', located 1.44 km from the center, with a 5-star rating, \$483 per night, and 9.4 rating from 1097 reviews. The second card is for 'The Selwyn Hotel', located 12.88 km from London, with a 4.5-star rating, \$361 per night, and 9.4 rating from 184 reviews.

ΕΙΚΟΝΑ 4: Αποτελέσματα αναζήτησης ξενοδοχείων

Μπορείτε να μεταβείτε στις λεπτομέρειες κάποιου ξενοδοχείου, πατώντας πάνω στην αντίστοιχη καρτέλα. Θα οδηγηθείτε σε μία σελίδα με πληροφορίες για το συγκεκριμένο ξενοδοχείο, όπως τις παροχές του, φωτογραφίες του, καθώς και μία φόρμα όπου μπορείτε να αναζητήσετε δωμάτιο και να κάνετε, αν επιλέξετε, κράτηση.

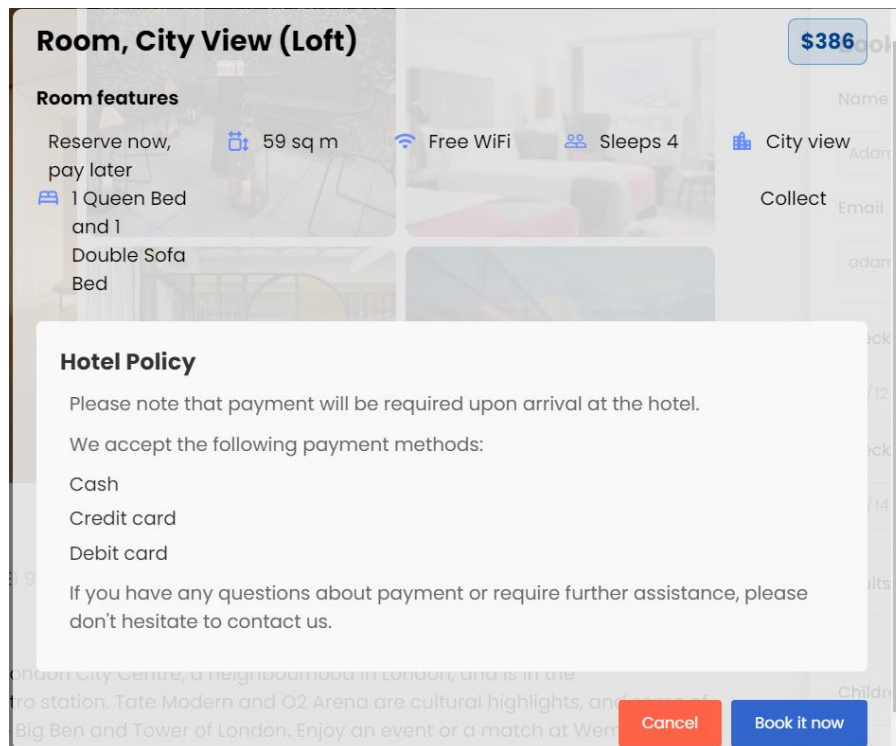
The screenshot shows the details page for 'Sea Containers London'. It features a gallery of images including the hotel's interior with a large orange sculpture, an outdoor terrace with a piano, a modern room with a desk, a lounge area with a green sofa, and a rooftop view. Below the gallery is the hotel name 'Sea Containers London' and its address '20 Upper Ground, London, England, SE1 9PD'. A 'Description' section provides information about the hotel's location in London City Centre and its proximity to cultural landmarks and public transportation. On the right side, there is a 'Book A Room' form with fields for Name, Email, Check In Date, Check Out Date, Adults (set to 1), and Children (set to 0), and a 'Search a Room' button.

ΕΙΚΟΝΑ 5: Σελίδα λεπτομερειών ξενοδοχείου

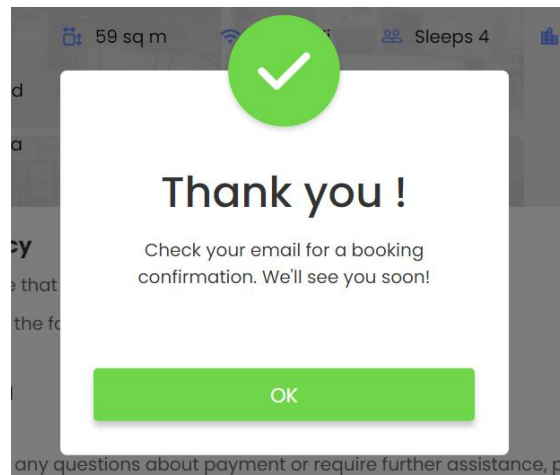


ΕΙΚΟΝΑ 6: Παροχές ξενοδοχείου

Συμπληρώνοντας τη φόρμα Book A Room, εμφανίζονται τα αποτελέσματα με τα χαρακτηριστικά του δωματίου και τις μεθόδους πληρωμών. Επιλέγοντας το κουμπί Book It now, κάνετε κράτηση του δωματίου και εμφανίζεται στην οθόνη σας το αντίστοιχο μήνυμα επιβεβαίωσης. Παράλληλα, θα σας έχει αποσταλεί και με email η επιβεβαίωση της κράτησης με επιπλέον πληροφορίες.



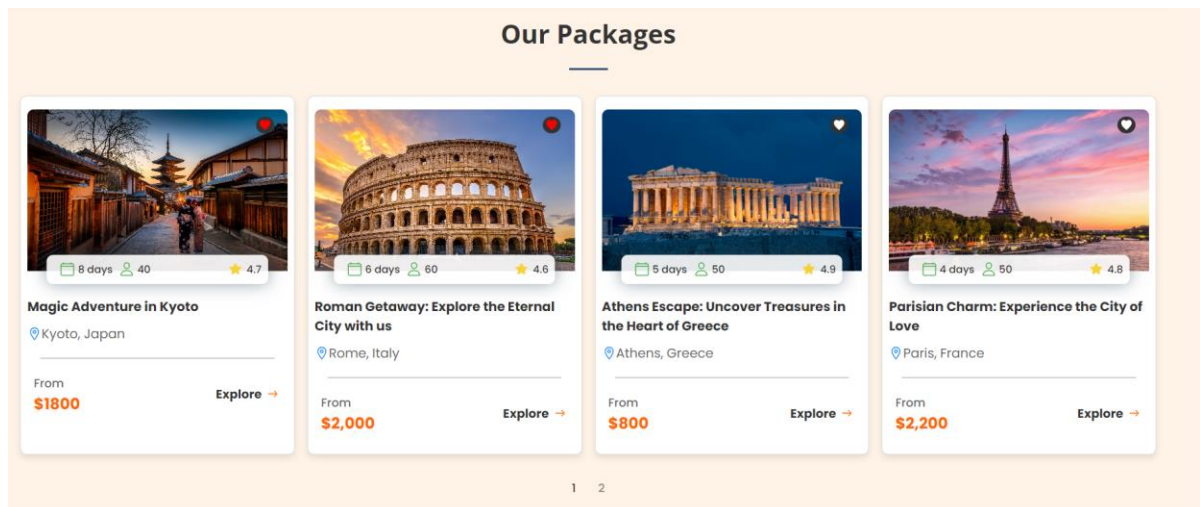
ΕΙΚΟΝΑ 7: Διαθεσιμότητα και χαρακτηριστικά δωματίου



ΕΙΚΟΝΑ 8: Επιβεβαίωση κράτησης δωματίου

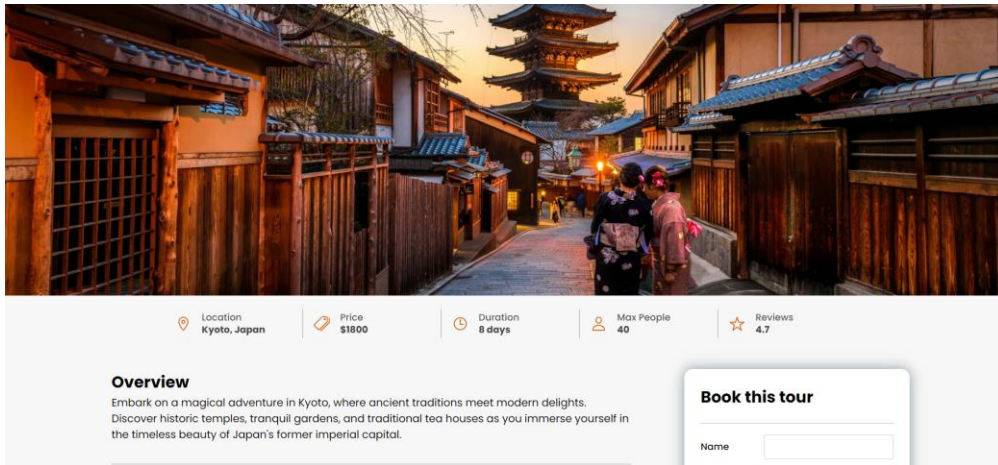
Κράτηση ταξιδιωτικών πακέτων

Στην κεντρική σελίδα μπορείτε επίσης να δείτε τα διαθέσιμα ταξιδιωτικά πακέτα που υπάρχουν και να επιλέξετε αν κάποιο από αυτά σας ενδιαφέρει, προσθέτοντάς το στα αγαπημένα σας, ή να δείτε περισσότερες πληροφορίες γι' αυτό, πατώντας το κουμπί “Explore” στην αντίστοιχη καρτέλα.

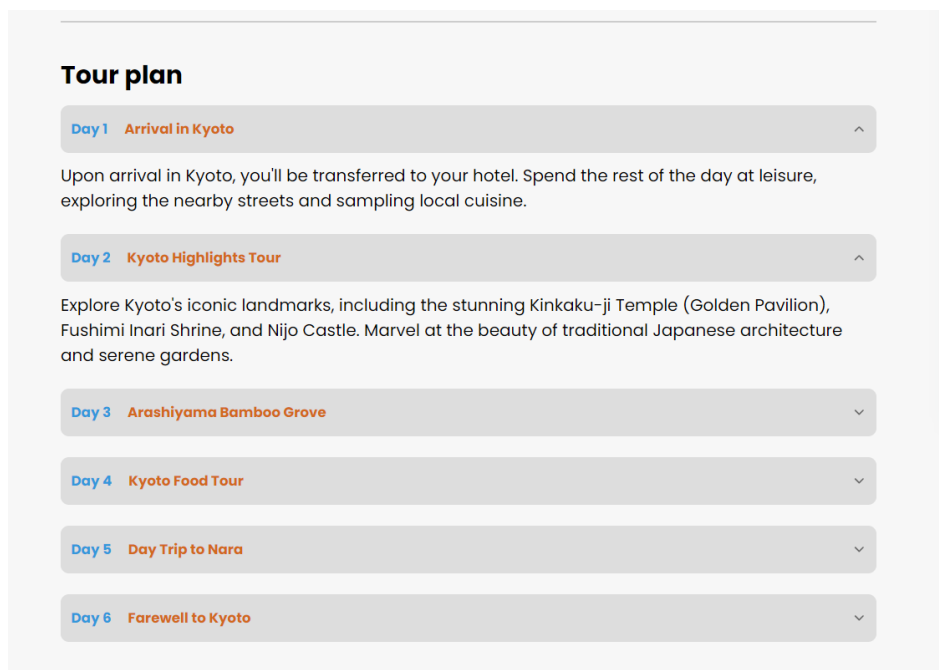


ΕΙΚΟΝΑ 9: Διαθέσιμα ταξιδιωτικά πακέτα

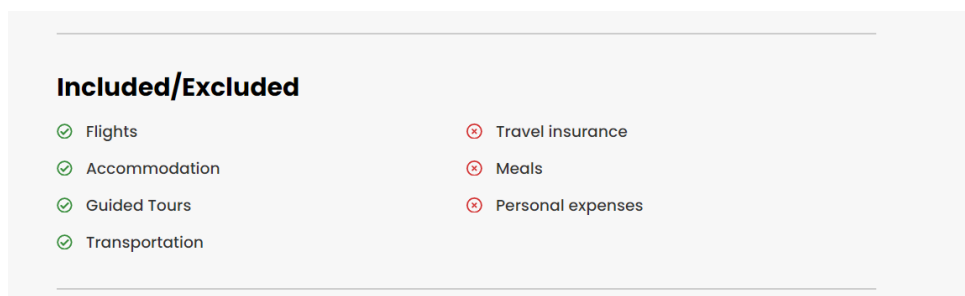
Επιλέγοντας να δείτε τις λεπτομέρειες κάποιου πακέτου, μεταβαίνετε στην αντίστοιχη σελίδα που περιέχει πληροφορίες όπως το αναλυτικό πρόγραμμα του ταξιδιού, οι τιμές του για τις διάφορες ηλικίες, η διάρκεια, ο μέγιστος αριθμός των ατόμων, καθώς και τι περιλαμβάνεται και τι όχι στο ταξίδι. Επίσης, υπάρχει διαθέσιμη η φόρμα κράτησης του ταξιδιού, την οποία μπορείτε να συμπληρώσετε για κάνετε book το πακέτο.



EIKONA 10: Overview ταξιδιωτικού πακέτου



EIKONA 11: Πρόγραμμα ταξιδιωτικού πακέτου



EIKONA 11: Included/Excluded ταξιδιωτικού πακέτου

Book this tour

Name

Email

Date

Time 10:00

Tickets +

Total **\$0**

[Book now](#)

ΕΙΚΟΝΑ 12: Φόρμα κράτησης του ταξιδιωτικού πακέτου

Αφού συμπληρώσετε τη φόρμα κράτησης του πακέτου, θα μεταβείτε σε ένα ασφαλές περιβάλλον για τη συμπλήρωση των στοιχείων σας και την πληρωμή. Μόλις πατήσετε το κουμπί Pay Now, θα έχετε ολοκληρώσει επιτυχώς την κράτηση του ταξιδιωτικού πακέτου, και ένα email επιβεβαίωσης με περαιτέρω πληροφορίες θα σας έχει αποσταλεί.

Payment Details

Credit Card Information

Card Number
 ✓

Name on Card
 ✓

Email
 ✓

Expiration Date
 ✓

CVV
 ✓

[Pay Now](#)

Package summary

DESTINATION
Kyoto, Japan

NUMBER OF TICKETS
111

START DATE
2024-09-15, 10:00

DURATION
8 days

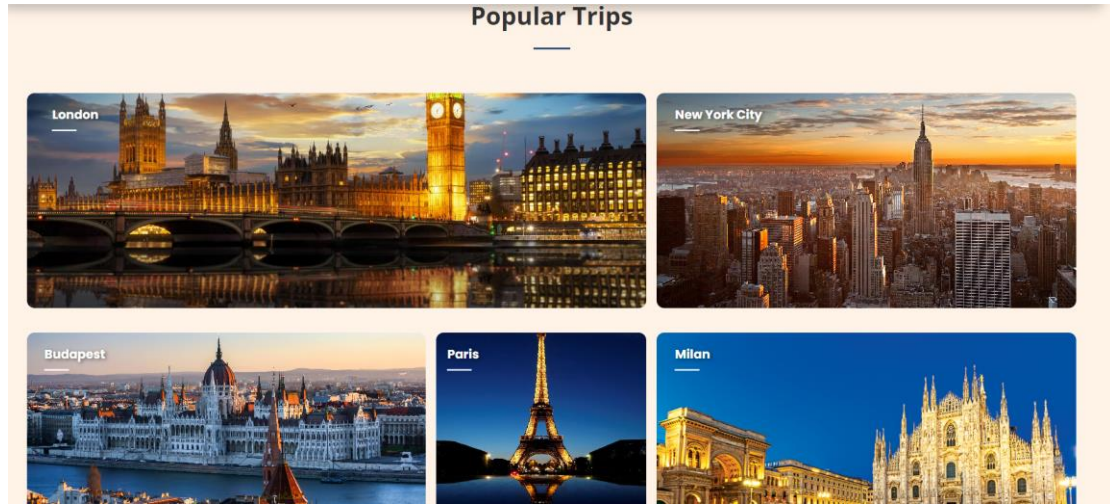
EMAIL
adams@gmail.com

Subtotal	\$4100
Tax	\$30
Total	\$4130

ΕΙΚΟΝΑ 13: Περιβάλλον κράτησης και πληρωμής του ταξιδιωτικού πακέτου

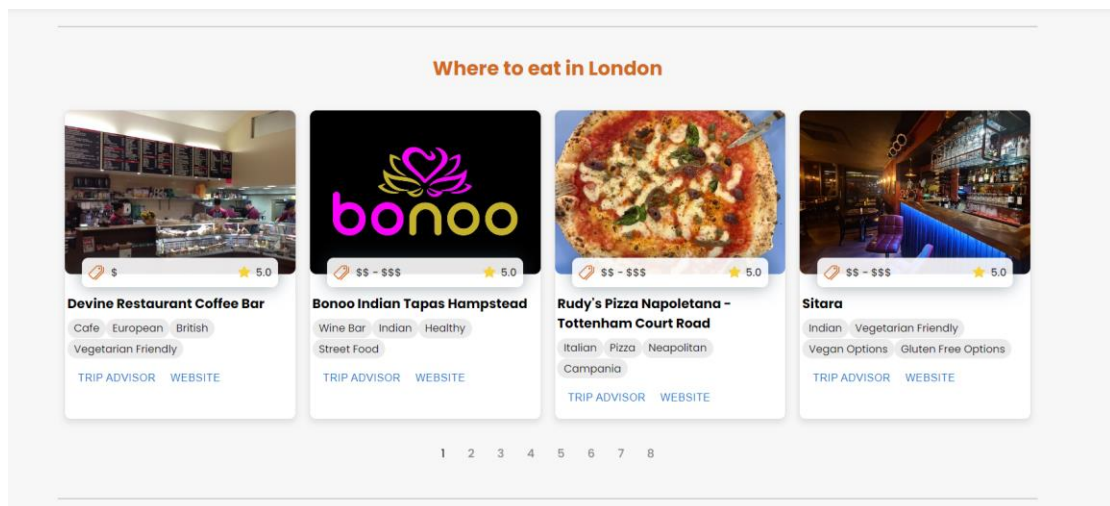
Προβολή δημοφιλών προορισμών

Στην κεντρική σελίδα υπάρχει και το section με δημοφιλείς προορισμούς, για τους οποίους μπορείτε να δείτε περισσότερες λεπτομέρειες, πατώντας στην αντίστοιχη καρτέλα.

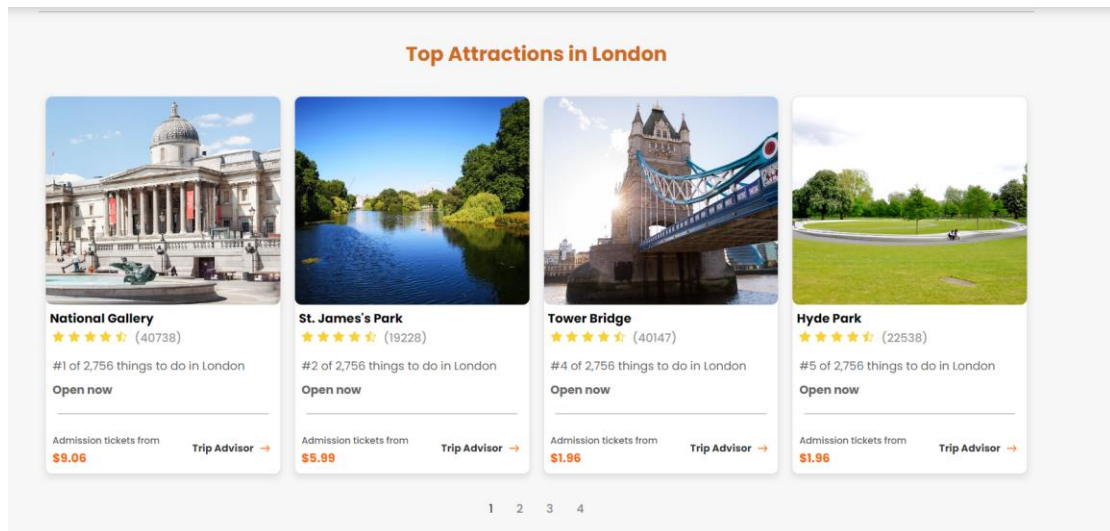


ΕΙΚΟΝΑ 14: Δημοφιλείς προορισμοί

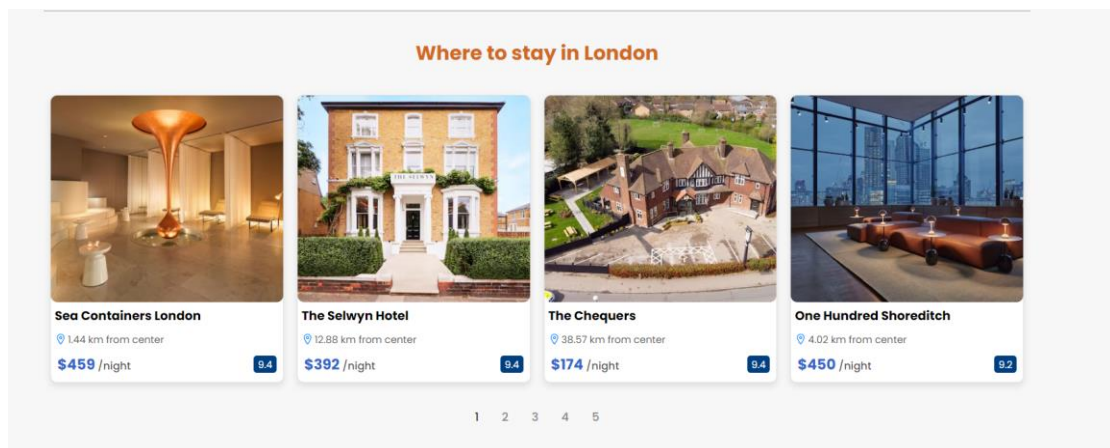
Επιλέγοντας κάποιον από τους παραπάνω προορισμούς, μεταβαίνετε σε μία σελίδα με πληροφορίες γι' αυτόν, όπως εστιατόρια, αξιοθέατα και ξενοδοχεία, καθώς και, αν υπάρχουν, τα διαθέσιμα μας ταξιδιωτικά πακέτα προς αυτόν. Μπορείτε, έπειτα, να επισκεφτείτε την ιστοσελίδα των διάφορων εστιατορίων ή αξιοθέατων, ή να μεταβείτε, πάλι, στη σελίδα με λεπτομέρειες για τα ξενοδοχεία που περιγράψαμε προηγουμένως.



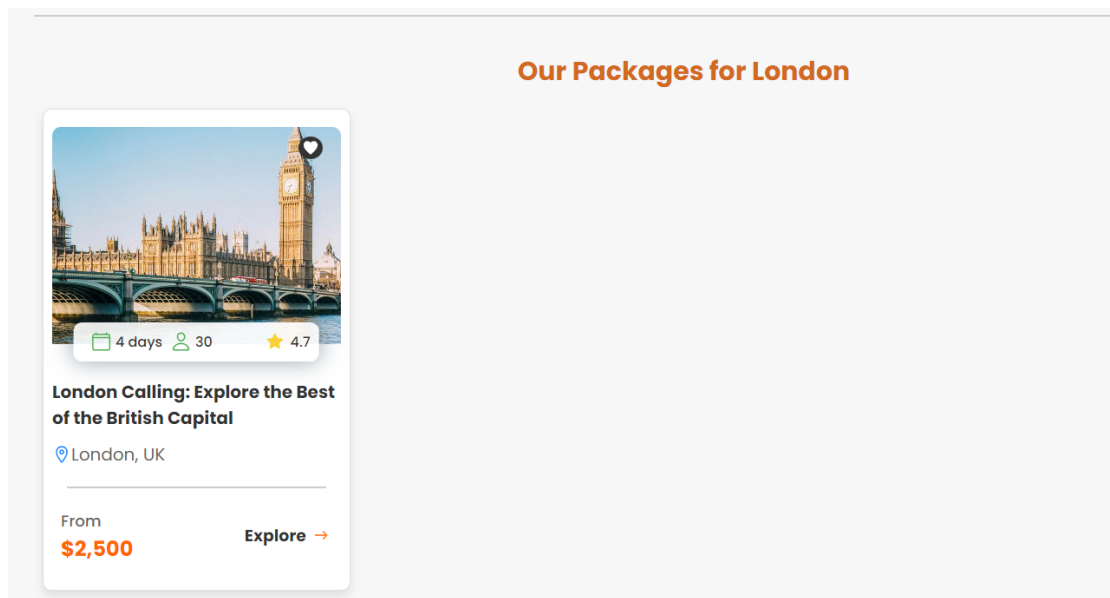
ΕΙΚΟΝΑ 15: Εστιατόρια επιλεγμένου προορισμού



ΕΙΚΟΝΑ 16: Αξιοθέατα επιλεγμένου προορισμού



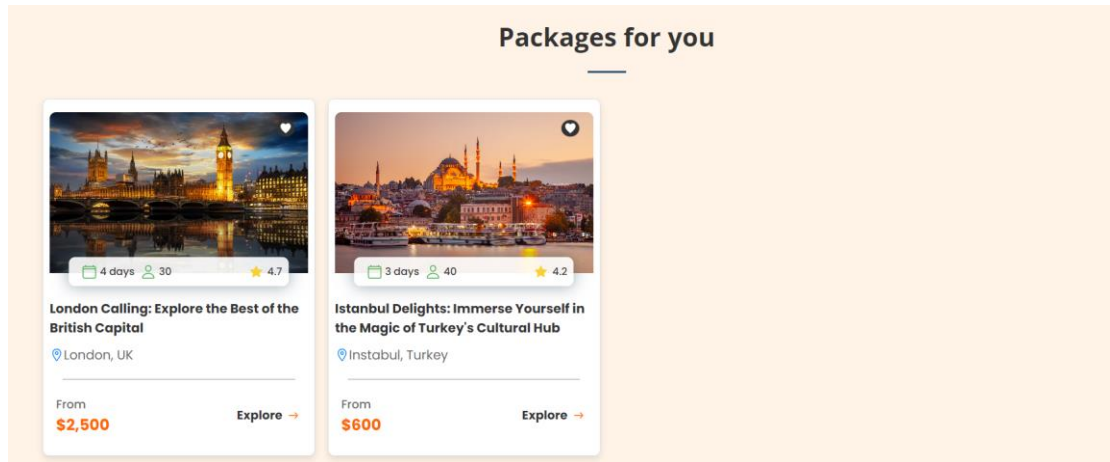
ΕΙΚΟΝΑ 17: Ξενοδοχεία επιλεγμένου προορισμού



ΕΙΚΟΝΑ 18: Ταξιδιωτικά πακέτα επιλεγμένου προορισμού

Εξατομικευμένα προτεινόμενα ταξιδιωτικά πακέτα

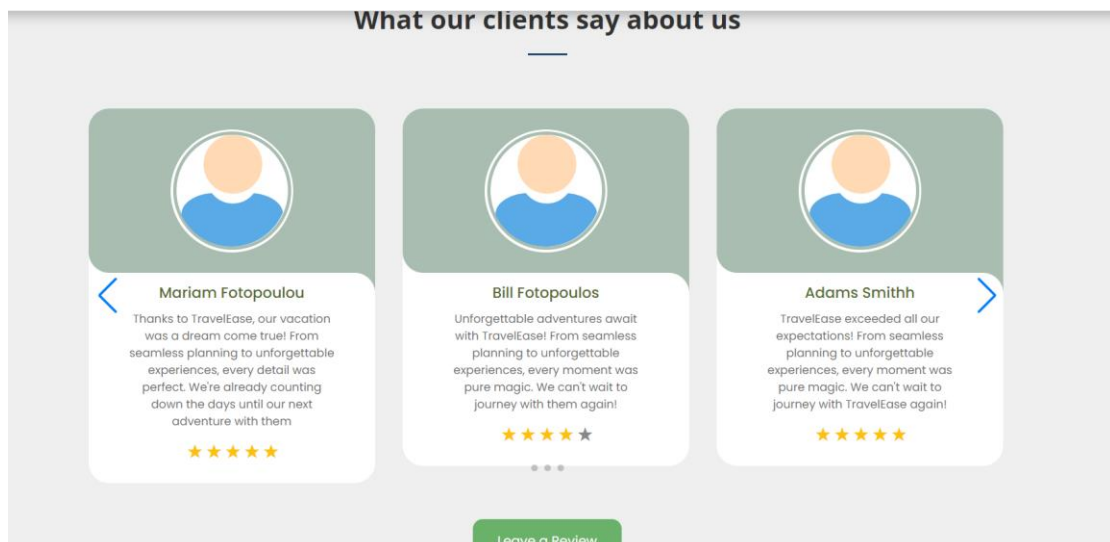
Αν έχετε αλληλεπιδράσει με την εφαρμογή μας και στο παρελθόν και έχετε αξιολογήσει ή χρησιμοποιήσει τις υπηρεσίες μας, στην ενότητα Packages for you θα βρείτε προτεινόμενα ταξιδιωτικά πακέτα που πιστεύουμε ότι θα σας ενδιαφέρουν.



ΕΙΚΟΝΑ 19: Προτεινόμενα εξατομικευμένα ταξιδιωτικά πακέτα

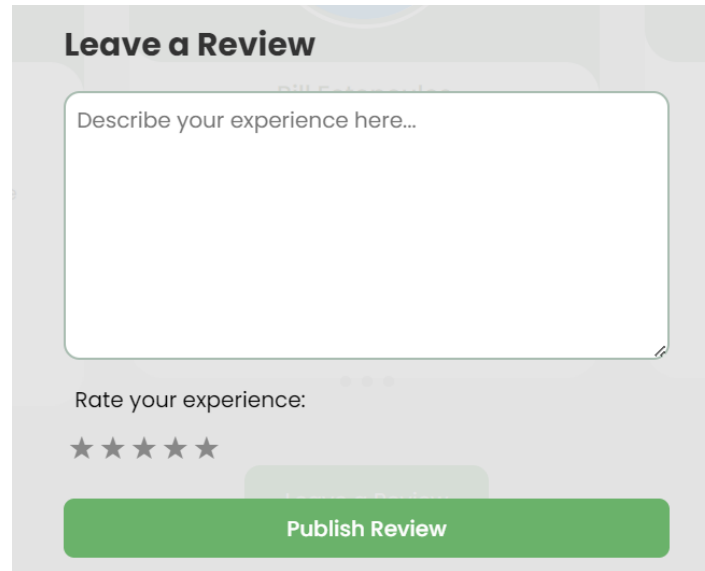
Αξιολογήσεις της εφαρμογής

Στο κάτω μέρος της κεντρικής σελίδας μπορείτε να δείτε τις αξιολογήσεις και τις κριτικές που έχουν υποβάλει διάφοροι χρήστες για την εφαρμογή, καθώς και να περιγράψετε και εσείς την εμπειρία και τις εντυπώσεις σας από τη χρήση των υπηρεσιών μας, βοηθώντας μας να γίνουμε καλύτεροι.



ΕΙΚΟΝΑ 20: Αξιολογήσεις της εφαρμογής

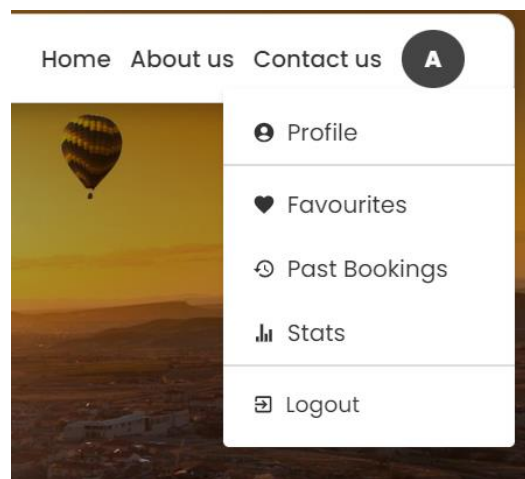
Πατώντας το κουμπί “Leave a Review”, εμφανίζεται μία φόρμα όπου μπορείτε να περιγράψετε την εμπειρία σας και να βαθμολογήσετε τις υπηρεσίες που σας παρήχθησαν. Με το κουμπί Publish Review, η κριτική σας δημοσιεύεται και μπορούν να τη δουν όλοι οι χρήστες.



ΕΙΚΟΝΑ 21: Φόρμα αξιολόγησης της εφαρμογής

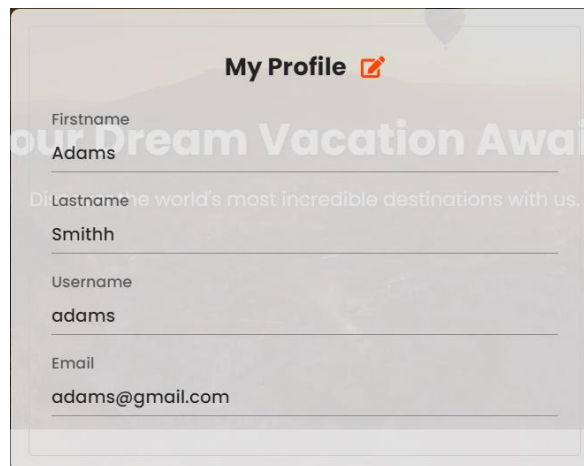
Προβολή λεπτομερειών και στατιστικών χρήστη

Στο πανθαρ της εφαρμογής, αν πατήσετε στο εικονίδιο του προφίλ σας, θα εμφανιστεί ένα drop-down menu με διάφορες επιλογές. Από εκεί μπορείτε να διαχειριστείτε το προφίλ σας, να δείτε τα αγαπημένα σας, το ιστορικό κρατήσεων σας και στατιστικά, καθώς και να αποσυνδεθείτε.



ΕΙΚΟΝΑ 22: Μενού λεπτομερειών χρήστη

Επιλέγοντας την επιλογή Profile, εμφανίζεται μία φόρμα με προ-συμπληρωμένα τα στοιχεία σας. Μπορείτε να τα επεξεργαστείτε, πατώντας το εικονίδιο επεξεργασίας δίπλα από το “My Profile”, όπως φαίνεται παρακάτω:

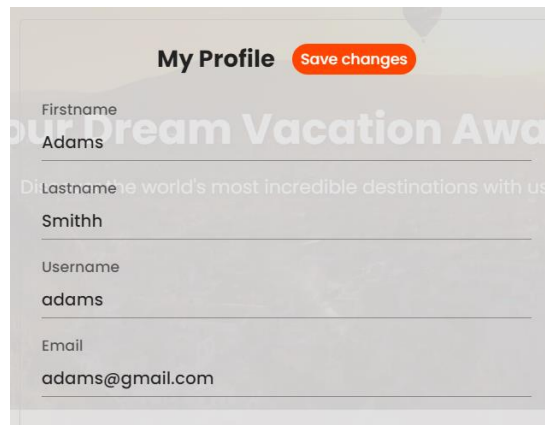


The screenshot shows a user profile form titled "My Profile" with an edit icon. The form contains the following fields and values:

Field	Value
Firstname	Adams
Lastname	Smithh
Username	adams
Email	adams@gmail.com

ΕΙΚΟΝΑ 23: Φόρμα ανανέωσης στοιχείων χρήστη

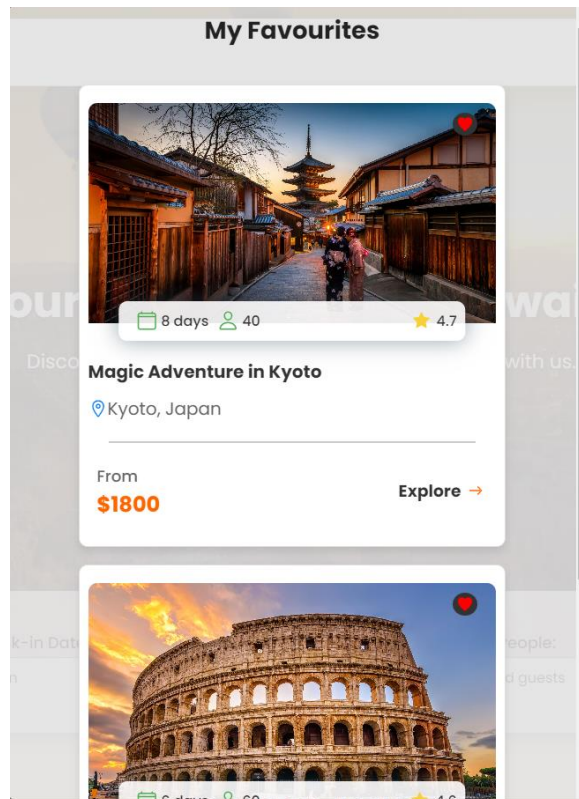
Μόλις τελειώσετε με την επεξεργασία των στοιχείων σας, πατάτε το κουμπί “Save Changes”, και τα ανανεωμένα στοιχεία σας αποθηκεύονται επιτυχώς.



The screenshot shows the same user profile form as in Figure 23, but with a red "Save changes" button highlighted next to the "My Profile" title.

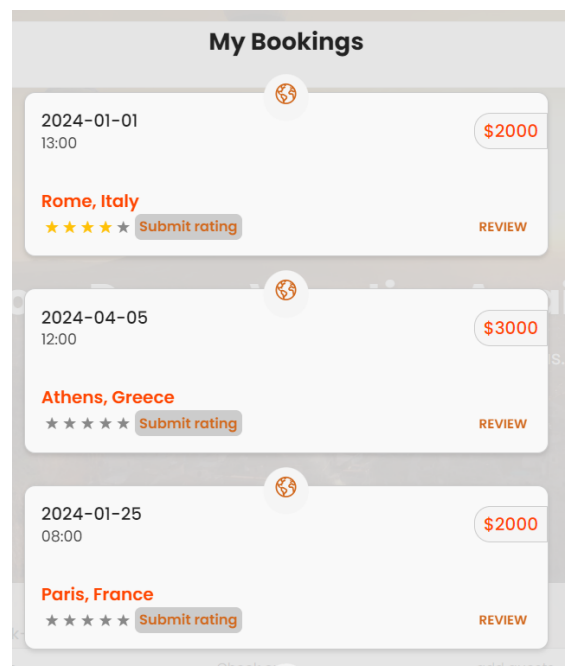
ΕΙΚΟΝΑ 24: Αποθήκευση ανανεωμένων στοιχείων χρήστη

Στο μενού των λεπτομερειών του χρήστη, αν επιλέξετε το κουμπί “Favorites”, θα εμφανιστεί μία λίστα με τα αγαπημένα σας ταξιδιωτικά πακέτα.



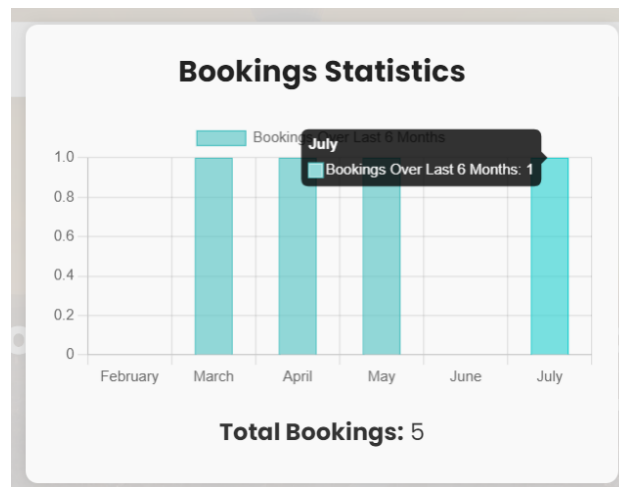
ΕΙΚΟΝΑ 25: Αγαπημένα ταξιδιωτικά πακέτα χρήστη

Το κουμπί “Past bookings” θα σας ανακατευθύνει σε μία λίστα με όλες τις παρελθοντικές σας κρατήσεις ταξιδιωτικών πακέτων, τις οποίες μπορείτε και να βαθμολογήσετε, επιλέγοντας τον αριθμό των αστεριών που θέλετε και πατώντας το κουμπί “Submit Rating”:

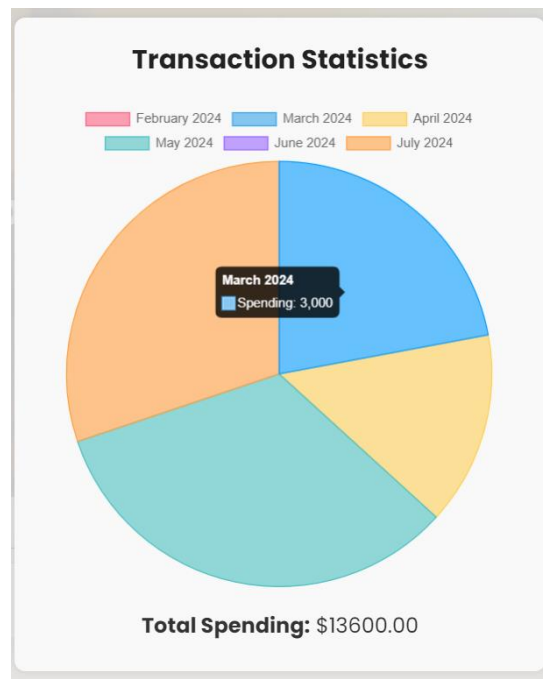


ΕΙΚΟΝΑ 26: Ιστορικό κρατήσεων ταξιδιωτικών πακέτων του χρήστη

Το κουμπί “Stats” θα σας εμφανίσει μία λίστα με τα στατιστικά που αφορούν τις κρατήσεις και τις συναλλαγές που έχετε πραγματοποιήσει στο παρελθόν χρησιμοποιώντας τις υπηρεσίες μας:



ΕΙΚΟΝΑ 27: Στατιστικά κρατήσεων ταξιδιωτικών πακέτων του χρήστη



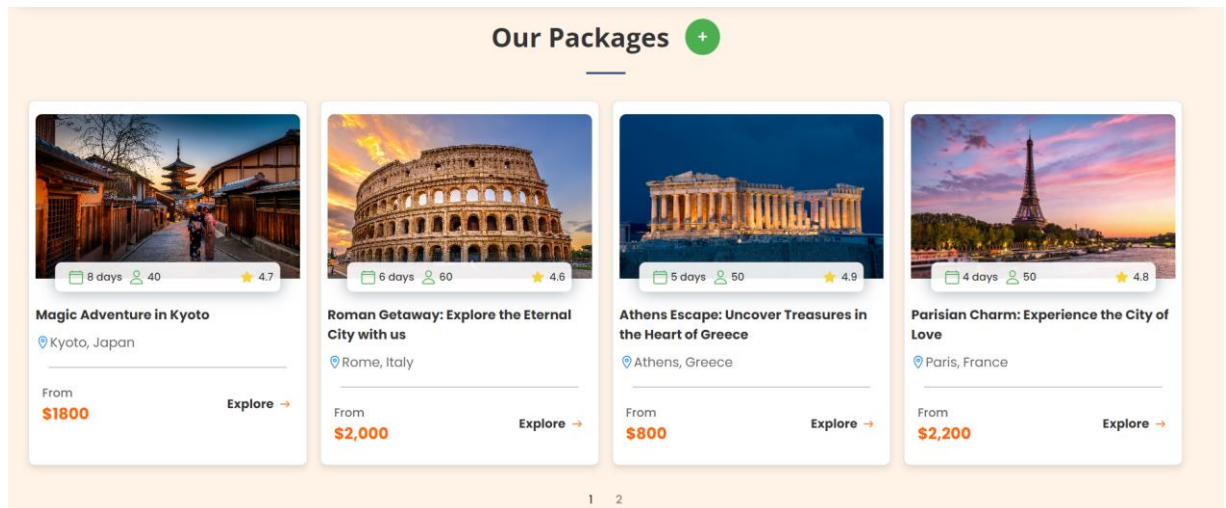
ΕΙΚΟΝΑ 28: Στατιστικά συναλλαγών του χρήστη

Το κουμπί “Logout”, τέλος, θα πραγματοποιήσει την επιτυχή αποσύνδεσή σας από την εφαρμογή, και θα σας ανακατευθύνει στην κεντρική σελίδα, με διαθέσιμες ξανά τις επιλογές Sign out και Sign In στο Navbar.

Εγχειρίδιο χρήσης (για τον administrator)

Ως διαχειριστής της εφαρμογής, έχετε πρόσβαση σε συγκεκριμένες λειτουργίες για την εξασφάλιση της ομαλής λειτουργίας του συστήματος.

Μετά την επιτυχή σύνδεσή σας στην εφαρμογή, θα μεταβείτε στην κεντρική σελίδα. Εκεί, στο section “Our Packages”, μπορείτε να δείτε τα ταξιδιωτικά πακέτα και να τα επεξεργαστείτε, να τα διαγράψετε, ή να προσθέσετε ένα καινούργιο πακέτο, πατώντας στο αντίστοιχο πράσινο εικονίδιο.

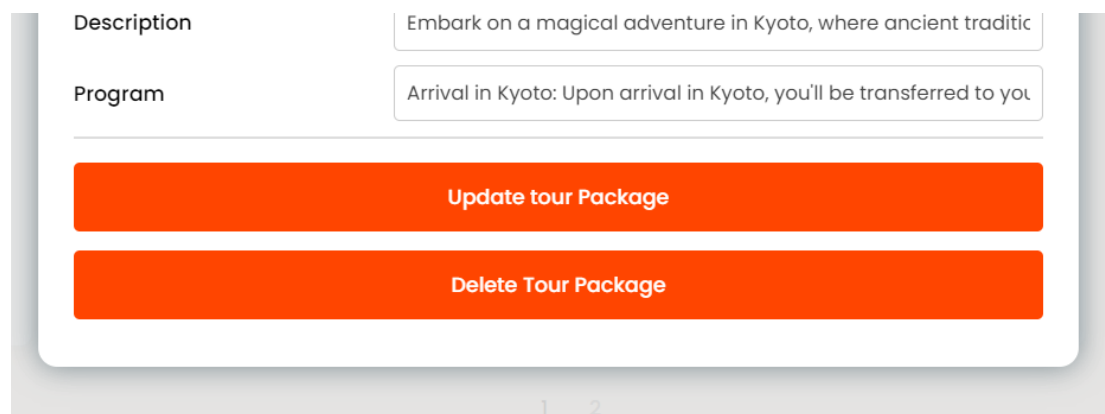


ΕΙΚΟΝΑ 29: Λίστα ταξιδιωτικών πακέτων με δυνατότητα τροποποίησης

ΕΙΚΟΝΑ 30: Φόρμα προσθήκης ταξιδιωτικού πακέτου

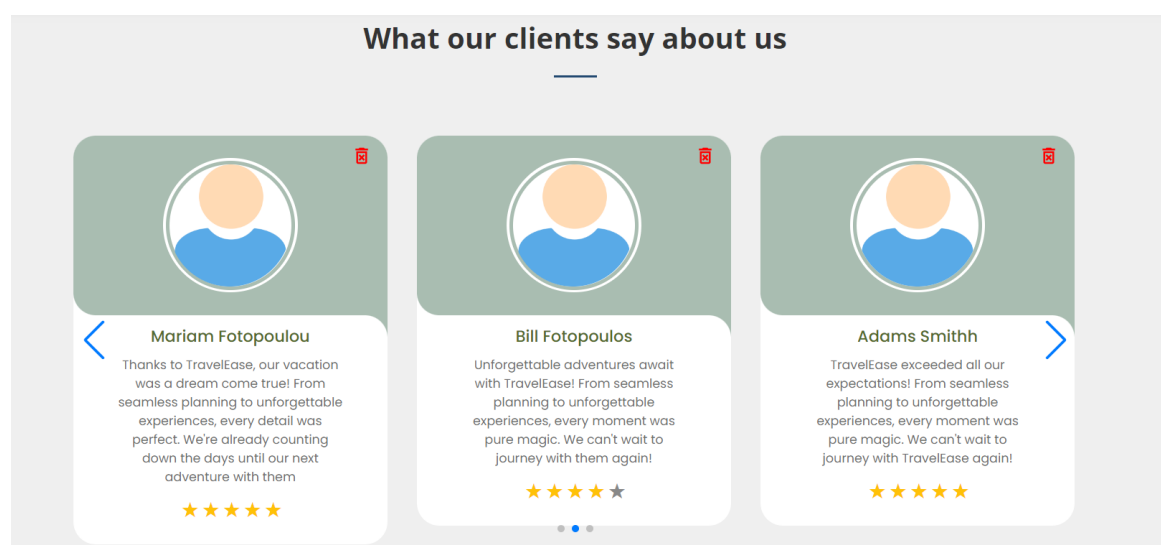
Πατώντας πάνω σε κάποια καρτέλα ταξιδιωτικού πακέτου, παράλληλα, εμφανίζεται μία φόρμα με προ-συμπληρωμένα τα πεδία των λεπτομερειών του πακέτου. Μπορείτε να τα τροποποιήσετε και να ολοκληρώσετε την τροποποίηση

πατώντας το κουμπί “Update Tour Package”, ή να το διαγράψετε, πατώντας το κουμπί “Delete Tour Package”.



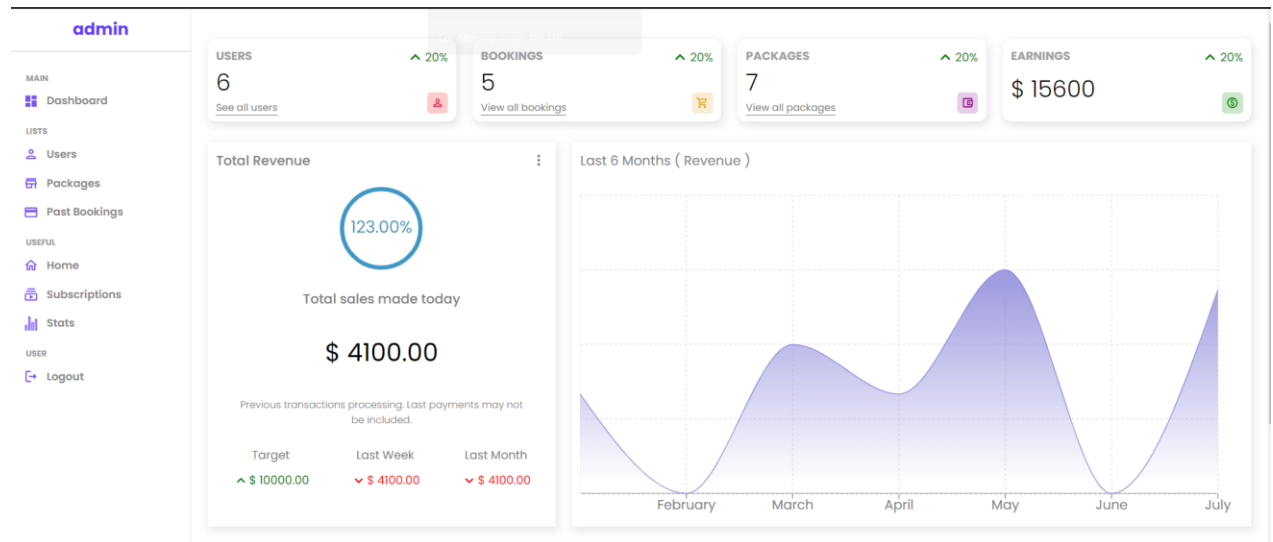
ΕΙΚΟΝΑ 31: Κουμπιά επεξεργασίας και διαγραφής ταξιδιωτικού πακέτου

Προχωρώντας στο section των αξιολογήσεων των χρηστών, έχετε τη δυνατότητα διαγραφής αξιολογήσεων, εάν αυτό κριθεί σκόπιμο, πατώντας πάνω στο εικονίδιο διαγραφής που υπάρχει σε κάθε καρτέλα αξιολόγησης.



ΕΙΚΟΝΑ 32: Αξιολογήσεις χρηστών με δυνατότητα διαγραφής

Πέρα από τις παραπάνω λειτουργίες, που μπορείτε να βρείτε στην κεντρική σελίδα της εφαρμογής, μπορείτε να επιλέξετε το κουμπί “Dashboard” που υπάρχει στο drop-down menu κάτω από το προφίλ σας. Αυτό θα σας μεταφέρει σε μία σελίδα, στην οποία έχετε πρόσβαση μόνο εσείς ως διαχειριστής, όπου θα μπορείτε να δείτε συγκεντρωμένες όλες τις διαχειριστικές λειτουργίες, τις οποίες μπορείτε να υλοποιήσετε.



ΕΙΚΟΝΑ 33: Dashboard διαχειριστή

Στο κεντρικό αυτό dashboard, μπορείτε να δείτε τους χρήστες της εφαρμογής, τις κρατήσεις, τα πακέτα, καθώς και τα έσοδα από τις διάφορες πληρωμές των χρηστών.

Διαχείριση χρηστών

Επιλέγοντας από το Sidebar το κουμπί Users, εμφανίζονται όλοι οι χρήστες της εφαρμογής σε έναν πίνακα. Για κάθε έναν από αυτούς μπορείτε να δείτε λεπτομέρειες, επιλέγοντας το κουμπί “View”. Ακόμα μπορείτε να διαγράψετε κάποιον χρήστη, ή να προσθέσετε, επιλέγοντας τα αντίστοιχα κουμπιά.

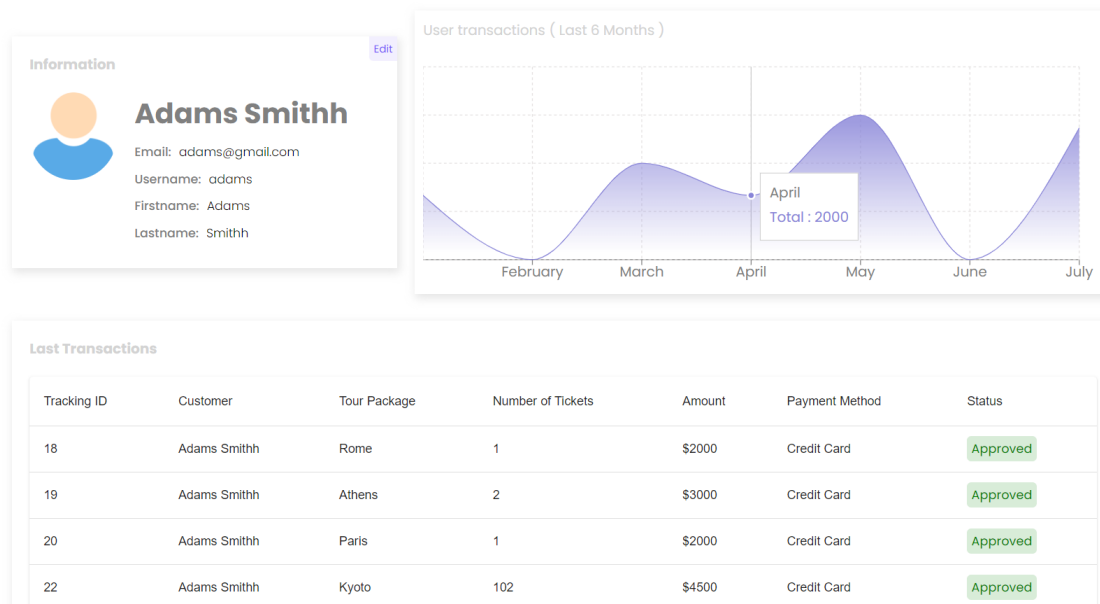
Add New User Add New

<input type="checkbox"/>	ID	First name	Last name	Username	Subscribed	Email	Action
<input type="checkbox"/>	1	Mariam	Fotopoulou	mariam	false	mariamfotopoulou@g...	View Delete
<input type="checkbox"/>	2	Bill	Fotopoulos	bill	false	billfotop@gmail.com	View Delete
<input type="checkbox"/>	3	Alexis	Fotopoulos	alex	false	alex@gmail.com	View Delete
<input type="checkbox"/>	4	Rafail	Fotopoulos	raf	false	raf@gmail.com	View Delete
<input type="checkbox"/>	5	Adams	Smithh	adams	true	adams@gmail.com	View Delete
<input type="checkbox"/>	6	admin	admin	admin	false	admin@gmail.com	View Delete

ΕΙΚΟΝΑ 34: Πίνακας χρηστών

Πατώντας το κουμπί “View” για κάποιον από αυτούς τους χρήστες, μεταβαίνετε στα στοιχεία του, βλέποντας, παράλληλα, τις συναλλαγές που έχει

πραγματοποιήσει τους τελευταίους 6 μήνες, καθώς και ποια ταξιδιωτικά πακέτα έχει κλείσει.



ΕΙΚΟΝΑ 35: Πληροφορίες Χρήστη

Αν στον κεντρικό πίνακα με τους χρήστες πατήσετε το κουμπί “Add New” που υπάρχει στο πάνω δεξιά μέρος, θα μεταβείτε στη σελίδα προσθήκης χρήστη, όπου καλείστε να συμπληρώσετε τα στοιχεία του νέου χρήστη που θέλετε να δημιουργήσετε.

The form is titled "Add new User" and contains fields for Username, Lastname, Password, Firstname, and Email. A "Send" button is located at the bottom right.

ΕΙΚΟΝΑ 36: Προσθήκη νέου χρήστη

Διαχείριση ταξιδιωτικών πακέτων

Επιλέγοντας στο Sidebar το κουμπί “Packages” μεταβαίνετε σε έναν πίνακα με όλα τα ταξιδιωτικά πακέτα, τα οποία μπορείτε να δείτε ή να διαγράψετε. Οι λειτουργίες της επεξεργασίας και της προσθήκης ταξιδιωτικών πακέτων υπάρχει στην κεντρική σελίδα της εφαρμογής, όπως αναλύθηκε παραπάνω.

Πτυχιακή Εργασία – Φωτοπούλου Μαριάμ

<input type="checkbox"/>	ID	Title	Place	Price	Duration	Rating	Action
<input type="checkbox"/>	1	Magic Adventure in Kyo...	Kyoto, Japan	\$1800	8 days	4.7	View Delete
<input type="checkbox"/>	2	Roman Getaway: Explor...	Rome, Italy	\$2,000	6 days	4.6	View Delete
<input type="checkbox"/>	3	Athens Escape: Uncover...	Athens, Greece	\$800	5 days	4.9	View Delete
<input type="checkbox"/>	4	Parisian Charm: Experie...	Paris, France	\$2,200	4 days	4.8	View Delete
<input type="checkbox"/>	5	London Calling: Explore ...	London, UK	\$2,500	4 days	4.7	View Delete
<input type="checkbox"/>	6	Istanbul Delights: Imme...	Instabul, Turkey	\$600	3 days	4.2	View Delete
<input type="checkbox"/>	7	Enchanting Budapest: A...	Budapest, Hung...	\$2,000	4 days	4.8	View Delete

EIKONA 37: Ταξιδιωτικά πακέτα

Διαχείριση ιστορικού συναλλαγών

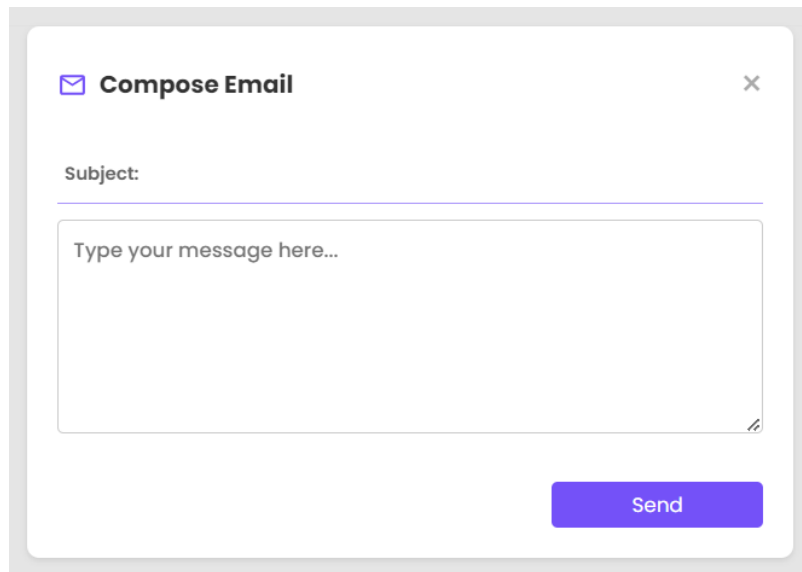
Το κουμπί “Past Bookings” στο Sidebar του dashboard εμφανίζει έναν πίνακα με όλες τις συναλλαγές που έχουν πραγματοποιήσει οι χρήστες. Υπάρχει εκεί η δυνατότητα διαγραφής κάποιας συναλλαγής από το ιστορικό, με την επιλογή του κουμπιού “Delete”.

<input type="checkbox"/>	ID	User	Package	Number of Tic...	Total Price	Action
<input type="checkbox"/>	18	Adams Smithh	Roman Getaway: Explore the Et...	1	2000	Delete
<input type="checkbox"/>	19	Adams Smithh	Athens Escape: Uncover Treas...	2	3000	Delete
<input type="checkbox"/>	20	Adams Smithh	Parisian Charm: Experience the...	1	2000	Delete
<input type="checkbox"/>	22	Adams Smithh	Magic Adventure in Kyoto	102	4500	Delete
<input type="checkbox"/>	25	Adams Smithh	Magic Adventure in Kyoto	111	4100	Delete

EIKONA 38: Ιστορικό συναλλαγών

Επικοινωνία με subscribed users

Εάν θέλετε να δείτε τους subscribed χρήστες της εφαρμογής και να στείλετε email σε κάποιον ή σε όλους, επιλέξτε το κουμπί “Subscriptions” από το Sidebar του dashboard. Μπορείτε να πατήσετε την επιλογή “Send mail to all subscribers” ή την επιλογή “Send mail” σε έναν subscriber, και θα σας εμφανιστεί η ακόλουθη φόρμα για την αποστολή του email στον αντίστοιχο παραλήπτη ή παραλήπτες.

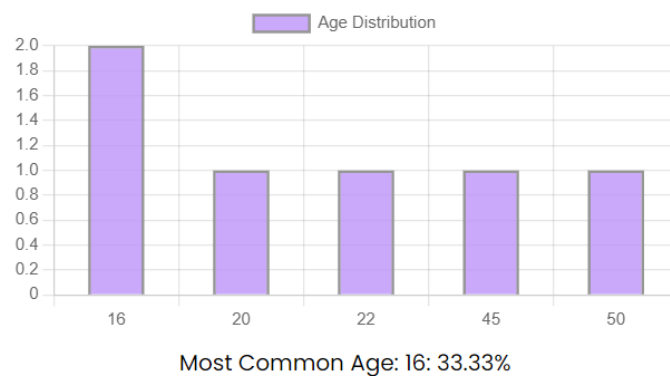


ΕΙΚΟΝΑ 39: Αποστολή email

Προβολή στατιστικών

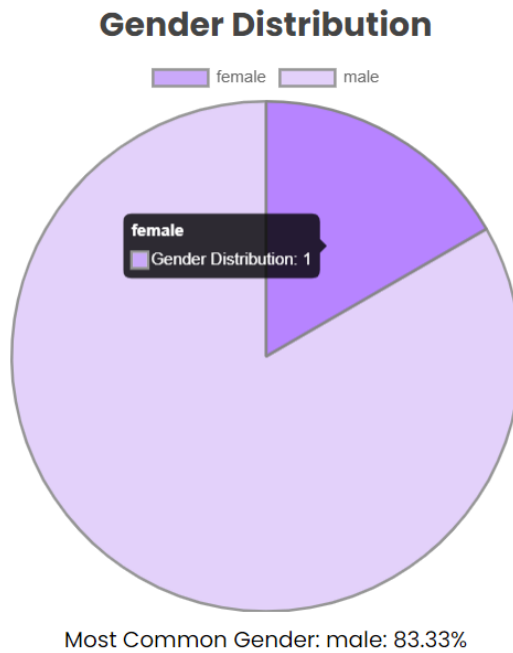
Η επιλογή του κουμπιού “Stats” στο Sidebar θα σας μεταφέρει στη σελίδα με συγκεντρωμένα στατιστικά για τους χρήστες και την εφαρμογή. Υπάρχουν 4 γραφήματα. Το πρώτο από αυτά αφορά την κατανομή της ηλικίας των χρηστών, όπως φαίνεται παρακάτω:

Age Distribution



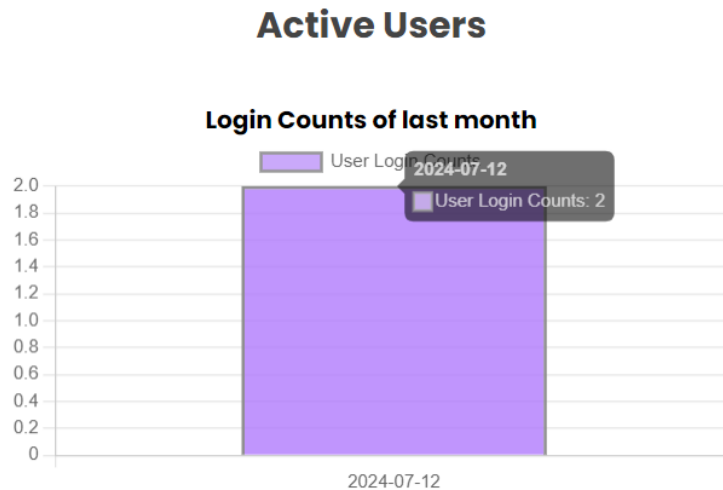
ΕΙΚΟΝΑ 40: Κατανομή ηλικίας των χρηστών

Το δεύτερο γράφημα αφορά το φύλο των χρηστών (male ή female).



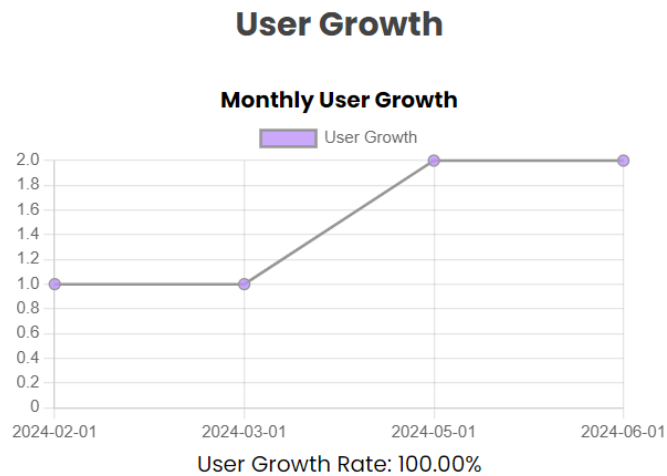
ΕΙΚΟΝΑ 41: Γράφημα φύλου των χρηστών

Το τρίτο γράφημα αφορά τη δραστηριότητα των χρηστών στην εφαρμογή, δείχνοντας τον αριθμό των logins τον τελευταίο μήνα.



ΕΙΚΟΝΑ 42: Γράφημα δραστηριότητας των χρηστών

Το τελευταίο γράφημα, τέλος, δείχνει τον ρυθμό ανάπτυξης της εφαρμογής, συγκρίνοντας τον αριθμό των καινούργιων χρηστών που εγγράφηκαν κάποιον μήνα σε σχέση με τις εγγραφές του προηγούμενου μήνα.



ΕΙΚΟΝΑ 43: Γράφημα ρυθμού ανάπτυξης των χρηστών

Τέλος, για να αποσυνδεθείτε από την εφαρμογή, πατήστε το κουμπί “Logout” από το Sidebar. Με αυτόν τον τρόπο θα αποσυνδεθείτε επιτυχώς, μεταβαίνοντας στην κεντρική σελίδα της εφαρμογής, με τις επιλογές Sign up και Sign in διαθέσιμες ξανά στο Navbar.

Μελλοντικές επεκτάσεις της εφαρμογής

Οι μελλοντικές επεκτάσεις της εφαρμογής TravelEase στοχεύουν στην περαιτέρω βελτίωση των υπηρεσιών και της εμπειρίας των χρηστών.

Μία μελλοντική επέκταση θα είναι η βελτίωση των προτάσεων ταξιδιωτικών πακέτων, λαμβάνοντας υπόψη όχι μόνο τις αλληλεπιδράσεις των χρηστών, αλλά και το ιστορικό των κρατήσεών τους. Αυτό θα επιτρέψει την παροχή ακόμη πιο ακριβών και σχετικών προτάσεων, προσαρμοσμένων στις προτιμήσεις και τις ανάγκες του κάθε χρήστη.

Επιπλέον, η ενσωμάτωση λειτουργιών κοινωνικών δικτύων, όπως η δυνατότητα σύνδεσης με λογαριασμούς σε Facebook, Google ή Twitter, θα διευκολύνει την εγγραφή και τη σύνδεση των χρηστών, ενώ παράλληλα θα ενισχύσει τη διαδραστικότητα της εφαρμογής.

Επιπλέον, η ενσωμάτωση ακτοπλοϊκών και αεροπορικών κρατήσεων στην εφαρμογή θα παρέχει στους χρήστες μια πιο ολοκληρωμένη εμπειρία, επιτρέποντάς τους να κλείνουν τα ταξίδια τους συνολικά απευθείας από την πλατφόρμα. Η δυνατότητα αυτή θα απλοποιήσει τη διαδικασία προγραμματισμού ταξιδιών και θα αυξήσει την ελκυστικότητα της εφαρμογής ως ένα ολοκληρωμένο εργαλείο ταξιδιωτικού σχεδιασμού.

Για την επέκταση της εφαρμογής στο ευρύτερο κοινό κρίνεται απαραίτητη και η υποστήριξη πολλαπλών γλωσσών, ενώ η ανάπτυξη εφαρμογής για κινητές συσκευές (iOS και Android) θα επιτρέψει στους χρήστες να έχουν πρόσβαση στις υπηρεσίες από οπουδήποτε και οποτεδήποτε.

Η ενσωμάτωση, τέλος, ενός πραγματικού και όχι εικονικού περιβάλλοντος πληρωμών είναι αναγκαία, προσφέροντας μία ολοκληρωμένη εμπειρία στους χρήστες με ασφάλεια και ευκολία.

Με αυτές τις μελλοντικές επεκτάσεις, η εφαρμογή μπορεί να αναβαθμίσει σημαντικά τις δυνατότητές της, προσφέροντας την καλύτερη δυνατή εξυπηρέτηση στους πελάτες και προσελκύοντας νέους χρήστες για την παροχή ολοκληρωμένων υπηρεσιών τουρισμού.

Συμπεράσματα

Η εφαρμογή αποτελεί μια ολοκληρωμένη και ευέλικτη πλατφόρμα που παρέχει εξαιρετικές υπηρεσίες στους χρήστες και τους διαχειριστές. Αναπτυγμένη με τη χρήση σύγχρονων τεχνολογιών όπως Spring Boot για το back-end και React για το front-end, εξασφαλίζει υψηλή απόδοση, ευελιξία και φιλικότητα προς τον χρήστη.

Για τους χρήστες, η εφαρμογή προσφέρει την ευκαιρία να ανακαλύψουν και να κάνουν κρατήσεις σε ταξιδιωτικά πακέτα και ξενοδοχεία, Οι προσωποποιημένες προτάσεις ταξιδιωτικών πακέτων, βασισμένες στο item-based collaborative filtering, επιτρέπουν στους χρήστες να εξερευνούν προορισμούς που ταιριάζουν στα ενδιαφέροντά τους. Η δυνατότητα προβολής στατιστικών και αναλύσεων βοηθά τους χρήστες να κάνουν ενημερωμένες επιλογές, ενώ η αποστολή email τους ενημερώνει για τις κρατήσεις τους και τις επιτυχείς συναλλαγές τους.

Οι διαχειριστές έχουν στη διάθεσή τους ένα ολοκληρωμένο περιβάλλον για τις διαχειριστικές λειτουργίες της εφαρμογής. Μπορούν να προσθέτουν, να επεξεργάζονται και να διαγράφουν ταξιδιωτικά πακέτα και χρήστες, καθώς και να παρακολουθούν τις δραστηριότητες των χρηστών και τις κρατήσεις. Οι πίνακες και τα γραφήματα παρέχουν μια σαφή εικόνα των στατιστικών στοιχείων της πλατφόρμας, επιτρέποντας τη λήψη στρατηγικών αποφάσεων για την βελτίωση των παρεχόμενων υπηρεσιών.

Η ευέλικτη αρχιτεκτονική της εφαρμογής επιτρέπει την εύκολη προσαρμογή και επέκταση μελλοντικών λειτουργιών, όπως η ενσωμάτωση ακτοπλοϊκών και αεροπορικών κρατήσεων, η υποστήριξη πολλαπλών γλωσσών, και η ανάπτυξη εφαρμογής για κινητές συσκευές. Αυτές οι επεκτάσεις θα καταστήσουν την πλατφόρμα ακόμη πιο ολοκληρωμένη και ελκυστική για τους χρήστες.

Συνοψίζοντας, η web εφαρμογή TravelEase προσφέρει εξαιρετικές λειτουργίες για τον προγραμματισμό και την κράτηση ταξιδιών, ενώ με την εφαρμογή των μελλοντικών επεκτάσεων, θα αποτελεί ένα ολοκληρωμένο εργαλείο, μπορώντας να αξιοποιηθεί πλήρως στον κλάδο του τουρισμού.

Βιβλιογραφία

- Kontogianni, Aristeia, and Efthimios Alepis. 2020. “Smart Tourism: State of the Art and Literature Review for the Last Six Years.” *Array* 6: 100020.
- ———. 2022. “AI, Blockchain & Cyber Tourism Joining the Smart Tourism Realm.” In *2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 1–6. IEEE.
- ———. 2023. “Social Network Data Enabling Smart Tourism.” In *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 1–6. IEEE.
- Kontogianni, Aristeia, Efthimios Alepis, and Constantinos Patsakis. 2022a. “Promoting Smart Tourism Personalised Services via a Combination of Deep Learning Techniques.” *Expert Systems with Applications* 187: 115964.
- ———. 2022b. “Smart Tourism and Artificial Intelligence: Paving the Way to the Post-Covid-19 Era.” *Advances in Artificial Intelligence-Based Technologies: Selected Papers in Honour of Professor Nikolaos G. Bourbakis—Vol. 1*, 93–109.
- Full-Stack Java Development with Spring Boot 3 & React – By Ramesh Fadatare: <https://www.udemy.com/course/full-stack-java-development-with-spring-boot-react/?couponCode=ACCAGE0923>
- Spring Boot Tutorial – By Amigoscode: <https://www.youtube.com/watch?v=9SGDpanrc8U>
- React Course – Beginner’s Tutorial for React JavaScript Library: <https://www.youtube.com/watch?v=9SGDpanrc8U>
- Spring Boot 3 + Spring Security 6 – JWT Authentication and Authorization: <https://www.youtube.com/watch?v=KxqUblhzfl>
- React icons: <https://react-icons.github.io/react-icons/>
- Material UI – Data Grid: <https://mui.com/x/react-data-grid/getting-started/>
- Material UI – Table: <https://mui.com/material-ui/react-table/>
- Recharts: <https://recharts.org/en-US/>
- Chart.js: <https://www.npmjs.com/package/chart.js?activeTab=readme>
- JSON Web Tokens: <https://jwt.io/introduction>
- Spring Data JPA: <https://www.geeksforgeeks.org/spring-boot-spring-data-jpa/>
- Collaborative filtering – Wikipedia: https://en.wikipedia.org/wiki/Collaborative_filtering
- Collaborative Filtering in Machine Learning: <https://www.geeksforgeeks.org/collaborative-filtering-ml/>
- Item-based collaborative filtering: <https://www.geeksforgeeks.org/item-to-item-based-collaborative-filtering/>

- Item based Collaborative Filtering: <https://www.youtube.com/watch?v=RiTc6PxMchM>
- RapidApi: <https://rapidapi.com/hub>
- Google-maps-react-markers: <https://www.npmjs.com/package/google-maps-react-markers>
- Google-map-react: <https://www.npmjs.com/package/google-map-react>
- Axios: <https://www.npmjs.com/package/axios>