

Πτυχιακή εργασία του φοιτητή Κοσμά Ηλία



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Πτυχιακή Εργασία

Κατασκευή Ηλεκτρονικής Πλατφόρμας Προώθησης και Λειτουργίας Τουριστικού Καταλύματος

Του φοιτητή

Κοσμά Ηλία

Αρ. Μητρώου: 113737

Επιβλέπων καθηγητής

Αντωνίου Ευστάθιος

Θεσσαλονίκη 2020

Πτυχιακή εργασία του φοιτητή Κοσμά Ηλία

Τίτλος Π.Ε. Κατασκευή Ηλεκτρονικής Πλατφόρμας Προώθησης και Λειτουργίας Τουριστικού
Καταλύματος

Κωδικός Π.Ε. 19043

Όνοματεπώνυμο φοιτητή/τών Ηλίας Κοσμάς

Όνοματεπώνυμο εισηγητή Ευστάθιος Αντωνίου

Ημερομηνία ανάληψης Π.Ε. 05-11-2019

Ημερομηνία περάτωσης Π.Ε. ...

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ηλία Κοσμά που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

ΠΕΡΙΛΗΨΗ

Η ραγδαία τεχνολογική ανάπτυξη έχει καταστήσει τη χρήση του Διαδικτύου σε αναπόσπαστο κομμάτι της καθημερινότητας εκατομμυρίων ανθρώπων σε όλο τον κόσμο. Η δυνατότητα εύκολης πρόσβασης στο διαδίκτυο ακόμα και από φορητές συσκευές έχει διευρύνει εκθετικά τη δυνατότητα προώθησης του τουριστικού προϊόντος σε όλους τους εν δυνάμει τουρίστες ανά τον κόσμο και από την άλλη μεγαλώνει τη ζήτηση καθώς οι ενδιαφερόμενοι έχουν καλύτερη πρόσβαση σε καλύτερο περιεχόμενο.

Ο κλάδος του τουρισμού έχει γνωρίσει, ειδικά την τελευταία δεκαετία, μεγάλη ανάπτυξη καθώς η χρήση διαδικτυακών εφαρμογών έχει δώσει μεγάλη ώθηση στην προώθηση και διαχείριση του τουριστικού προϊόντος και αφετέρου έχουν κάνει την πρόσβαση, την επιλογή και την κράτηση τουριστικού καταλύματος υπόθεση λίγων λεπτών.

Η πτυχιακή εργασία παρουσιάζει μία διαδικτυακή πλατφόρμα προώθησης και διαχείρισης ενός τουριστικού καταλύματος που επικεντρώνεται στα εργαλεία που μπορεί να παρέχει σε ένα διαχειριστή καταλύματος και στον τρόπο παρουσίασης του προϊόντος προς τον υποψήφιο πελάτη.

Development of a web-based platform for promoting and operating a tourist accommodation

Ilias Kosmas

ABSTRACT

Rapid technological development has made the use of the Internet an integral part of the daily lives of millions of people around the world. The ability to easily access the Internet even from portable devices has exponentially expanded the ability to promote the tourism product to all potential tourists around the world and on the other hand is increasing demand as those interested have better access and better content.

The tourism industry has experienced great growth, especially in the last decade, as the use of web applications has given a great boost to the promotion and management of the tourism product and on the other hand have made access, selection and booking a tourist accommodation in a matter of minutes.

This undergraduate thesis is an online platform for promoting and managing a tourist accommodation that focuses on the tools it can provide to an accommodation manager and how to present the product to the prospective customer.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	3
Development of a web-based platform for promoting and operating a tourist accommodation.....	4
ABSTRACT	4
ΠΕΡΙΕΧΟΜΕΝΑ	5
Κατάλογος Εικόνων.....	7
1. Σκοπός της πτυχιακής εργασίας – Το πρόβλημα που επιλύει.....	9
2. Μελέτη – Αξιολόγηση υφιστάμενων λύσεων (υπηρεσιών) που έχουν αναπτυχθεί.....	11
2.1 Booking.com.....	11
2.2 Airbnb.....	12
2.3 Vrbo (πρώην HomeAway).....	12
3. Περιγραφή του τεχνολογικού περιβάλλοντος που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής.....	13
3.1 Flask Framework και Python.....	13
3.2 Βάσεις Δεδομένων και Διαχείριση.....	14
3.3 Μηχανή Προτύπων Jinja.....	16
3.4 Περιβάλλοντα Ανάπτυξης.....	17
3.5 Επεκτάσεις και Βιβλιοθήκες.....	18
3.6 Front – End.....	19
3.7 Άλλες υπηρεσίες που χρησιμοποιήθηκαν.....	19
3.7.1 MailGun.....	20
3.7.2 Stripe.....	20
4. Οδηγός χρήσης λογισμικού για τον Διαχειριστή της πλατφόρμας.....	21
4.1 Για τον Πίνακα Ελέγχου.....	21
4.2 Είσοδος στην εφαρμογή.....	21
4.3 Περιβάλλον εφαρμογής Πίνακα Ελέγχου.....	22
4.3.1 Το αριστερό μενού του Πίνακα Ελέγχου.....	23
4.3.2 Το κεντρικό παράθυρο του Πίνακα Ελέγχου.....	25

4.3.3 Η επάνω και η κάτω μπάρα πλοήγηση του Πίνακα Ελέγχου.....	25
4.4 Οι επιλογές του διαχειριστή	25
4.4.1 Ημερολόγιο	25
4.4.2 Κρατήσεις	28
4.4.3 Αλλαγή Φωτογραφίας.....	29
4.4.4 Αλλαγή Τιμής.....	31
4.4.5 Προσθήκη Δωματίου	31
4.4.6 Διαγραφή Δωματίου.....	32
5. Περιγραφή της εφαρμογής σε επίπεδο τελικού χρήστη	32
5.1 Περιήγηση στο περιβάλλον εφαρμογής της πλατφόρμας.....	33
5.2 Κάνοντας κράτηση μέσω της πλατφόρμας	36
6. Οδηγός λογισμικού για προγραμματιστή.....	41
6.1 Εκτέλεση της εφαρμογής	41
6.1 Δομή των αρχείων της πλατφόρμας.....	42
6.2 Δομή της βάσης δεδομένων	43
6.3 Οι φόρμες που χρησιμοποιεί η πλατφόρμα	44
6.4 Οι μέθοδοι που χρησιμοποιεί η πλατφόρμα.....	45
6.5 Σύντομη παρουσίαση της λογικής της πλατφόρμας	46
6.5.1 Βασικά χαρακτηριστικά	46
6.5.2 Έλεγχος διαθεσιμότητας.....	47
6.5.3 Πραγματοποίηση Κράτησης	48
6.5.4 Η λογική των εργαλείων του Διαχειριστή	49
Συμπεράσματα και μελλοντικές επεκτάσεις.....	55
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	57

Κατάλογος Εικόνων

Figure 3.1: Το Λογότυπο του πλαισίου Flask	14
Figure 3.2: Το Λογότυπο της γλώσσα προγραμματισμού Python	14
Figure 3.3: Το Λογότυπο της βάσης δεδομένων MySQL	15
Figure 3.4: Το Λογότυπο της SQLAlchemy	16
Figure 3.5: Το Λογότυπο της Jinja	17
Figure 3.6: Το Λογότυπο του Docker	18
Figure 3.7: Λογότυπα των υπηρεσιών mailgun και stripe	20
Figure 8: Η σελίδα εισόδου χρηστών στην πλατφόρμα	22
Figure 9: Η επιλογή "Πίνακας Ελέγχου" μετά την είσοδο του διαχειριστή	22
Figure 10: Άποψη της καρτέλας "Κρατήσεις" στον Πίνακα Ελέγχου	23
Figure 11: Το περιεχόμενο της καρτέλας "Ημερολόγιο"	26
Figure 12: Επιλεγμένες ημερομηνίες από τον διαχειριστή	27
Figure 13: Μήνυμα συστήματος	28
Figure 14: Αλλαγή Φωτογραφίας	30
Figure 15: Προσθήκη Δωματίου	32
Figure 16: Η αρχική σελίδα της πλατφόρμας και η φόρμα για τον έλεγχο διαθεσιμότητας	34
Figure 17: Η σελίδα "Δωμάτια"	34
Figure 18: Η σελίδα "Εγγραφή"	35
Figure 19: Η σελίδα "Είσοδος"	35
Figure 20: Η σελίδα "Λογαριασμός" μετά την είσοδο του χρήστη	36
Figure 21: Μήνυμα αποτυχίας "404"	36
Figure 22: Έλεγχος διαθεσιμότητας δωματίου	37
Figure 23: Μήνυμα ενημέρωσης για μη διαθέσιμες ημερομηνίες	37
Figure 24: Προτεινόμενες ημερομηνίες	37
Figure 25: Η σελίδα συμπλήρωσης στοιχείων για την κράτηση με συνδεδεμένο χρήστη	38
Figure 26: Η σελίδα συμπλήρωσης στοιχείων για την κράτηση με μη συνδεδεμένο χρήστη	38
Figure 27: Μήνυμα αποτυχίας "Ο αριθμός της κάρτας σας δεν είναι έγκυρος"	39
Figure 28: Μήνυμα αποτυχίας "Λανθασμένο email"	39
Figure 29: Μήνυμα επιτυχημένης κράτησης	39
Figure 30: Εκτέλεση της εντολής docker-compose up --build σε ΛΣ Ubuntu στον κατάλογο που βρίσκεται η εφαρμογή	41
Figure 31: Εκτέλεση της εντολής docker-compose down	42
Figure 32: Η κλάση Rooms	44
Figure 33: Εισαγωγή από το αρχείο functions.py και users.py	46

Figure 34: Αρχικοποίηση του flask-login και ρύθμιση παραμέτρων	46
Figure 35: Η προβολή (view) της αρχικής σελίδας, επιστρέφει όλες τις εγγραφές από τους πίνακες room και reviews και στη συνέχεια επιστρέφει τη σελίδα index.html	47
Figure 36: Λειτουργία της check_availability (1).....	47
Figure 37: Λειτουργία της check_availability (2).....	48
Figure 38: Η αρχή της get_info.....	48
Figure 39: Η συνθήκη if που τροποποιεί τον πίνακα Vacancies.....	49
Figure 40: Επιστροφή αρχείου json με τις κρατήσεις του πίνακα guest_reservation.....	50
Figure 41: Επιστροφή αρχείου json με τις κρατήσεις του πίνακα reservations.....	51
Figure 42: Η λογική πίσω από τη λειτουργία για το άνοιγμα ημερομηνιών	52
Figure 43: Το τμήμα του κώδικα που είναι υπεύθυνο για την αλλαγή φωτογραφίας	53

1. Σκοπός της πτυχιακής εργασίας – Το πρόβλημα που επιλύει

Η πτυχιακή εργασία ανήκει στην κατηγορία των εφαρμογών ηλεκτρονικού εμπορίου (e-commerce ή eCommerce ή e-comm) στον κλάδο του τουρισμού, που αναπτύχθηκαν με την εξέλιξη της τεχνολογίας και του μεγάλου ανταγωνισμού που προέκυψε στον κλάδο. Ως ηλεκτρονικό εμπόριο (Η.Ε.) ορίζεται το εμπόριο παροχής αγαθών και υπηρεσιών που πραγματοποιείται εξ αποστάσεως με ηλεκτρονικά μέσα, βασιζόμενο δηλαδή στην ηλεκτρονική μετάδοση δεδομένων, χωρίς να καθίσταται αναγκαία η φυσική παρουσία των συμβαλλομένων μερών, πωλητή-αγοραστή. Ολοένα και περισσότερες τουριστικές επιχειρήσεις αναπτύσσουν τις δικές τους διαδικτυακές εφαρμογές και χρησιμοποιούν τις νέες τεχνολογίες. Το Ηλεκτρονικό Εμπόριο στον κλάδο του τουρισμού παρέχει ένα σύνολο δυνατοτήτων που παρέχουν το διαδίκτυο και οι νέες τεχνολογίες σε άτομα που επιθυμούν να ταξιδέψουν. Οι εν δυνάμει πελάτες μπορούν να ενημερωθούν, να αναζητήσουν, να συγκρίνουν και να επιλέξουν οποιαδήποτε τουριστική υπηρεσία επιθυμούν όπως ξενοδοχειακό κατάλυμα, ενοικίαση αυτοκινήτου, ακτοπλοϊκά και αεροπορικά εισιτήρια.

Οι κύριες λειτουργίες της εν λόγω διαδικτυακής πλατφόρμας είναι από τη μία να παρέχει βασικά εργαλεία διαχείρισης στον διαχειριστή του καταλύματος ώστε να είναι σε θέση να λειτουργεί το κατάλυμα με τον τρόπο που επιθυμεί και αφετέρου να προσφέρει ένα καλαίσθητο περιβάλλον για τον υποψήφιο πελάτη όπου έχει τη δυνατότητα κράτησης στα δωμάτια που προσφέρονται από την πλατφόρμα.

Όπως αναφέρθηκε παραπάνω μία από τις κύριες λειτουργίες της πλατφόρμας είναι η παροχή εργαλείων στο διαχειριστή για την λειτουργία του καταλύματός του. Αυτό σημαίνει ότι η εφαρμογή είναι σε θέση να προσφέρει λειτουργίες όπως: η παρακολούθηση των κρατήσεων που γίνονται από εγγεγραμμένους και μη χρήστες, ο ορισμός ανοικτών και κλειστών ημερομηνιών κατά τη βούληση του διαχειριστή, η αλλαγή στις τιμές των δωματίων, η προσθήκη ή η αφαίρεση ενός δωματίου, η αλλαγή φωτογραφιών στα δωμάτια.

Παράλληλα, από τη μεριά του υποψήφιου πελάτη η εφαρμογή εκτός ενός φιλικού και καλαίσθητου περιβάλλοντος θα προσφέρει δυνατότητες όπως: ο δυναμικός έλεγχος διαθεσιμότητας ημερομηνιών για το συγκεκριμένο δωμάτιο, η πρόταση εναλλακτικών ημερομηνιών αν το εύρος των επιλεγμένων ημερομηνιών

δεν είναι διαθέσιμο, ένα σύστημα εγγραφής νέων πελατών - χρηστών που θα έχουν τη δυνατότητα να παρακολουθούν τις λεπτομέρειες των κρατήσεων τους, να κάνουν μία κριτική, να εγγραφούν στο Newsletter. Επίσης, θα παρέχεται ένα αξιόπιστο σύστημα ηλεκτρονικής πληρωμής για την υλοποίηση της κράτησης. Τέλος, οι χρήστες θα έχουν την επιλογή αλλαγής γλώσσας της πλατφόρμας από Αγγλική σε Ελληνική και το αντίστροφο.

Κλείνοντας, η πλατφόρμα προσπαθεί να πετύχει κάποια βασικά στοιχεία που κάθε εφαρμογή τέτοιου τύπου πρέπει να έχει, όπως το να είναι φιλικό προς τον χρήστη(user-friendly), φιλικό προς κινητές συσκευές(mobile-friendly), υψηλής ευκρίνειας φωτογραφικό υλικό, κριτικές από τους πελάτες ώστε να καλύπτει όσο το δυνατόν μεγαλύτερο εύρος χρηστών.

2.Μελέτη – Αξιολόγηση υφιστάμενων λύσεων (υπηρεσιών) που έχουν αναπτυχθεί

Σε αυτό το κεφάλαιο παρουσιάζονται μερικές από τις πιο γνωστές διαδικτυακές πλατφόρμες που αξιοποιήθηκαν και οι οποίες αφορούν το χώρο του Ηλεκτρονικού Εμπορίου στον κλάδο του τουρισμού. Συστήματα όπως τα παρακάτω παίζουν κυρίαρχο ρόλο στον τομέα του e-commerce, καθώς με την είσοδό τους στην αγορά του τουρισμού άλλαξαν στην ουσία τον τρόπο με τον οποίο πραγματοποιούνται οι διακοπές σήμερα.

2.1 Booking.com

Η Booking.com, ιδρύθηκε το 1996 στο Άμστερνταμ και εξελίχθηκε από μια μικρή ολλανδική start-up σε μία από τις κορυφαίες διαδικτυακές εταιρείες ταξιδιών στον κόσμο. Ως μία από τις μεγαλύτερες πλατφόρμες ηλεκτρονικού εμπορίου στον τομέα των ταξιδιών, τόσο για γνωστές αλυσίδες, όσο και για επιχειρήσεις κάθε μεγέθους, η Booking.com δίνει τη δυνατότητα σε καταλύματα από όλο τον κόσμο να συνδεθούν με ένα παγκόσμιο κοινό και να αναπτύξουν τις επιχειρήσεις τους.

Η πλατφόρμα της Booking.com προσφέρει στον χρήστη που θα την επισκεφτεί μία πληθώρα από επιλογές όπως η αναζήτηση για διαμονή, για πτήσεις, για ενοικίαση αυτοκινήτου κ.α. Συγκεκριμένα, αν αναζητήσουμε κάποιο κατάλυμα συμπληρώνοντας τον τόπο προορισμού, τις ημερομηνίες άφιξης και αναχώρησης θα δούμε ότι μας εμφανίζει όλα τα καταλύματα που είναι διαθέσιμα στον προορισμό και στις ημερομηνίες που έχουμε επιλέξει. Από εκεί και πέρα έχουμε τη δυνατότητα να επιλέξουμε ξεχωριστά το κάθε κατάλυμα, να δούμε τις παροχές που προσφέρει και τους κανονισμούς που έχει και να κάνουμε την κράτησή μας. Επίσης, η πλατφόρμα δίνει τη δυνατότητα δημιουργίας λογαριασμού, με τη βοήθεια του οποίου μπορούμε να διαχειριζόμαστε τις κρατήσεις μας.

Από πλευράς διαχειριστή καταλύματος, η Booking.com προσφέρει τη δυνατότητα διαχείρισης του καταλύματος μέσω της εφαρμογής Extranet. Πρόκειται για μία εφαρμογή που έχουν πρόσβαση μόνο οι διαχειριστές καταλυμάτων και προσφέρει αμέτρητες δυνατότητες για τη διαχείριση του τουριστικού καταλύματος όπως: η αλλαγή διαθεσιμότητας του καταλύματος μέσω ημερολογίου καθώς και η αλλαγή τιμής ανά βραδιά, η προσθήκη ειδικών προσφορών για συγκεκριμένες ημερομηνίες, η προσθήκη ή αλλαγή των παροχών του καταλύματος, η προσθήκη κανονισμών καταλύματος, η παρακολούθηση κρατήσεων, η παρακολούθηση κριτικών του καταλύματος, η αναλυτική καρτέλα για τα οικονομικά στοιχεία κ.ά.

Η Booking.com δίνει όλα τα απαραίτητα εργαλεία στον διαχειριστή για να είναι σε θέση να διαχειρίζεται έτσι όπως αυτός θέλει το κατάλυμά του και από την άλλη προσφέρει στον εν δυνάμει πελάτη αμέτρητες επιλογές και διευκολύνσεις για να βρει αυτό που ακριβώς ψάχνει.

2.2 Airbnb

Μία άλλη δημοφιλής πλατφόρμα ενοικίασης τουριστικού καταλύματος είναι αυτή της Airbnb, με τη διαφορά ότι η συγκεκριμένη πλατφόρμα εκτός της κράτησης ολόκληρου καταλύματος δίνει τη δυνατότητα στον υποψήφιο επισκέπτη – πελάτη κράτησης μέρους του καταλύματος και όχι ολόκληρου του χώρου.

Όπως ακριβώς και με την πλατφόρμα της Booking.com, η πλατφόρμα της Airbnb με το που επιλέξουμε τον τόπο προορισμού, μία ημερομηνία άφιξης και μία ημερομηνία αναχώρησης εμφανίζει τα διαθέσιμα καταλύματα της περιοχής στις συγκεκριμένες ημερομηνίες που είχαμε επιλέξει. Στη συνέχεια, ο χρήστης μπορεί να αναζητήσει και να βρει αυτό που εκείνος επιθυμεί.

Όπως συμβαίνει και με την πλατφόρμα της Booking.com, αντίστοιχες δυνατότητες εύρυθμης λειτουργίας καταλύματος προσφέρει και η πλατφόρμα της Airbnb στο διαχειριστή, καθώς και όλα εκείνα τα εργαλεία που χρειάζεται για να λειτουργεί το κατάλυμά του όπως ακριβώς εκείνος επιθυμεί.

2.3 Vrbo (πρώην HomeAway)

Παρόμοια πλατφόρμα με της Airbnb και Booking.com, είναι αυτή της Vrbo όπου ο επισκέπτης έχει τη δυνατότητα αναζήτησης, επιλογής και κράτησης καταλύματος βάζοντας αρχικά τον τόπο προορισμού, την ημερομηνία άφιξης και την ημερομηνία αναχώρησης.

Αντίστοιχα, ο διαχειριστής έχει τη δυνατότητα να τροποποιήσει το κατάλυμα του και το πώς λειτουργεί με τα εργαλεία που του προσφέρονται από την πλατφόρμα.

Οι παραπάνω, είναι μερικές από τις υπάρχουσες λύσεις που έχουν αναπτυχθεί και από τις οποίες η πλατφόρμα που αναλύεται στην παρούσα πτυχιακή εργασία αντλεί έμπνευση, κυρίως αναφορικά με τα εργαλεία που παρέχονται στον

διαχειριστή καθώς και με κάποιες βασικές λειτουργίες που έχει στη διάθεσή του ο υποψήφιος επισκέπτης – πελάτης.

3. Περιγραφή του τεχνολογικού περιβάλλοντος που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής

Για την ανάπτυξη της διαδικτυακής πλατφόρμας χρησιμοποιήθηκαν μία πληθώρα από τεχνολογίες όπως το Flask Framework, η Jinja (template engine), η SQLAlchemy, το Docker που περιγράφονται αναλυτικά παρακάτω.

3.1 Flask Framework και Python

Το Flask είναι ένα μικροπλαίσιο προγραμματισμού (micro web framework) διαδικτυακών εφαρμογών στην γλώσσα προγραμματισμού Python. Αναφέρεται ως μικροπλαίσιο (microframework), όχι επειδή όλη η διαδικτυακή εφαρμογή μπορεί να χωρέσει σε ένα και μόνο αρχείο Python, ούτε επειδή του λείπει η λειτουργικότητα που έχουν άλλα παρόμοια πλαίσια προγραμματισμού. Αναφέρεται ως τέτοιο διότι έχει ως στόχο να κρατήσει τον πυρήνα του απλό, αλλά παράλληλα εύκολα επεκτάσιμο.

Το Flask δεν έρχεται με προεπιλεγμένες βάσεις δεδομένων, με φόρμες επαλήθευσης ή οτιδήποτε άλλο που να μην έχει ήδη υλοποιηθεί και το πλαίσιο να μη μπορεί να διαχειριστεί. Το Flask υποστηρίζει επεκτάσεις όπου προσθέτουν τέτοιες λειτουργίες όπως για παράδειγμα το FlaskWTF, το Flask-Login, το Flask-Bcrypt κ.α. Όλα εξαρτώνται από τον προγραμματιστή που θα αξιοποιήσει ότι χρειάζεται ακριβώς και τίποτα παραπάνω. Αυτοί είναι και οι λόγοι που επιλέχθηκε το συγκεκριμένο πλαίσιο προγραμματισμού.



Figure 3.1: Το Λογότυπο του πλαισίου Flask

Πέρα από τα παραπάνω, σημαντικό ρόλο στην επιλογή του πλαισίου έπαιξε και η γλώσσα Python. Πρόκειται για μία γλώσσα υψηλού επιπέδου που περιέχει όλες τις λειτουργίες που ένας προγραμματιστής μπορεί να χρειαστεί. Η Python διαθέτει όλες τις απαραίτητες βιβλιοθήκες για ανάπτυξη διαδικτυακών εφαρμογών, είναι εύκολη στη χρήση και διαθέτει πολλά εργαλεία που συμβάλλουν ώστε να απλοποιείται η ανάπτυξη της πλατφόρμας και να διευκολύνεται ο προγραμματιστής.



Figure 3.2: Το Λογότυπο της γλώσσα προγραμματισμού Python

3.2 Βάσεις Δεδομένων και Διαχείριση

Το πλαίσιο Flask που χρησιμοποιήθηκε για την ανάπτυξη της πλατφόρμας υποστηρίζει τη χρήση όλων των γνωστών βάσεων δεδομένων όπως για παράδειγμα η MySQL, Postgres, SQLite κ.α. Για την ανάπτυξη της εφαρμογής έχει επιλεγεί η βάση δεδομένων MySQL. Πρόκειται για ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Είναι μία από τις πιο δημοφιλείς βάσεις δεδομένων για διαδικτυακές εφαρμογές και χρησιμοποιείται από μία σειρά διαδεδομένες διαδικτυακές πλατφόρμες όπως το Google, το Facebook, το YouTube.

Κάποια από τα θετικά χαρακτηριστικά της MySQL που μπορούμε να αναφέρουμε είναι η επεκτασιμότητα και η ευελιξία που προσφέρει καθώς υποστηρίζεται από όλα τα συστήματα Linux, UNIX και Windows. Επίσης το γεγονός ότι είναι λογισμικό ανοικτού κώδικα επιτρέπει την προσαρμογή της στις απαιτήσεις του εκάστοτε προγραμματιστή. Άλλα χαρακτηριστικά που μπορούμε να αναφέρουμε είναι η καλές επιδόσεις, η σταθερή διαθεσιμότητα, η προστασία των δεδομένων. Τέλος, ένας από τους λόγους που η MySQL είναι μία από τις πιο διαδεδομένες βάσεις δεδομένων παγκοσμίως, είναι το γεγονός ότι παρέχει υποστήριξη σε κάθε εφαρμογή που μπορεί να αναπτυχθεί χωρίς να μπαίνει εμπόδιο η γλώσσα προγραμματισμού που έχει επιλεγεί. Για όλους τους παραπάνω λόγους επιλέχθηκε να χρησιμοποιηθεί η MySQL στο περιβάλλον ανάπτυξης για την αποθήκευση και ανάκτηση των δεδομένων.



Figure 3.3: Το Λογότυπο της βάσης δεδομένων MySQL

Το Flask όπως έχει αναφερθεί είναι ένα μικροπλαίσιο το οποίο έχει μεγάλες δυνατότητες επέκτασης. Έτσι λοιπόν όπως και άλλα διαδικτυακά πλαίσια, μπορεί να διαχειριστεί διαφορετικά τη σχεσιακή βάση δεδομένων που χρησιμοποιείται. Χρησιμοποιώντας ένα μηχανισμό που ονομάζεται αντικείμενο-σχεσιακή αντιστοίχιση ή χαρτογράφηση (Object-Relational Mapping ή ORM), δίνει τη δυνατότητα στον προγραμματιστή να μπορεί έμμεσα να αλληλεπιδρά με τη βάση δεδομένων μέσω κλάσεων – μοντέλων τα οποία είναι γραμμένα σε κώδικα Python παρέχοντας τις κατάλληλες συναρτήσεις για την επικοινωνία με τη βάση δεδομένων. Με τη χρήση ενός τέτοιου μηχανισμού ο προγραμματιστής αποφεύγει την απευθείας συναλλαγή με τη βάση δεδομένων γράφοντας κώδικα SQL και επικεντρώνεται σε

κώδικα σε Python που επιστρέφει το ίδιο αποτέλεσμα. Συγκεκριμένα στη διαδικτυακή πλατφόρμα που αναπτύχθηκε έχει χρησιμοποιηθεί η SQLAlchemy μία εργαλειοθήκη της Python SQL και ORM, καθώς είναι ευρέως διαδεδομένη μέσα στην κοινότητα, εύκολη στη χρήση αλλά και στην κατανόηση. Πρόκειται για



Figure 3.4: Το Λογότυπο της SQLAlchemy

Τέλος, χρησιμοποιήθηκε το εργαλείο phpMyAdmin το οποίο δίνει τη δυνατότητα διαχείρισης μίας MySQL βάσης μέσω διαδικτυακής εφαρμογής. Προσφέρει στο χρήστη ένα γραφικό περιβάλλον μέσω του οποίου μπορεί να δημιουργήσει νέες βάσεις δεδομένων ή πίνακες, να τροποποιήσει υπάρχουσες βάσεις και να εισάγει δεδομένα. Υποστηρίζει την εισαγωγή ή εξαγωγή δεδομένων σε διάφορους τύπους αρχείων καθώς επίσης δίνει τη δυνατότητα εκτέλεσης οποιουδήποτε SQL statement.

3.3 Μηχανή Προτύπων Jinja

Η Jinja είναι μία μηχανή προτύπων (template engine) για διαδικτυακές εφαρμογές σε γλώσσα προγραμματισμού Python. Αν και μοιάζει με τη μηχανή προτύπων του Django, διαφέρει στο γεγονός ότι μπορεί να χρησιμοποιεί εκφράσεις πιο κοντά στη σύνταξη της Python. Είναι μία μοντέρνα και φιλική προς τον σχεδιαστή γλώσσα προτύπου καθώς επίσης γρήγορη, διαδεδομένη και ασφαλής με επιλογές εκτέλεσης κώδικα σε απομονωμένο περιβάλλον (sandboxed environment). Κάποια από τα κύρια χαρακτηριστικά της Jinja είναι:

- Εκτέλεση κώδικα σε απομονωμένο περιβάλλον
- Κληρονομικότητα μεταξύ προτύπων
- Εύκολο debugging με αρίθμηση γραμμών και εστίαση στη σωστή γραμμή μέσα στο πρότυπο
- Εύκολη και διαχειρίσιμη σύνταξη

Κλείνοντας, να αναφέρουμε ότι η Jinja αποτελεί την προεπιλεγμένη μηχανή προτύπων του διαδικτυακού πλαισίου Flask.



Figure 3.5: Το Λογότυπο της Jinja

3.4 Περιβάλλοντα Ανάπτυξης

Η ανάπτυξη της διαδικτυακής πλατφόρμας υλοποιήθηκε στο Λειτουργικό Σύστημα (ΛΣ) Ubuntu 18.04 LTS (Bionic Beaver). Το συγκεκριμένο ΛΣ έρχεται με προεγκαταστημένα εργαλεία και βιβλιοθήκες που χρειάζεται οποιοσδήποτε για να ξεκινήσει κατευθείαν να αναπτύσσει την εφαρμογή που επιθυμεί. Για παράδειγμα, έρχεται με προεγκατεστημένη τη γλώσσα προγραμματισμού Python. Είναι επίσης ελαφρύ και δεν επιβαρύνει το σύστημα στο οποίο εγκαθίσταται.

Για την ανάπτυξη διαδικτυακών εφαρμογών προτείνεται η χρήση περιβαλλόντων που είναι απομονωμένα μεταξύ τους. Συνήθως, αυτό γίνεται με τη χρήση εικονικών περιβαλλόντων όπως το `ripenv` ή το `virtualenv`. Για τη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε το `Docker`. Το `Docker`, αντίθετα με πιο παραδοσιακές λύσεις, χρησιμοποιεί `containers` οι οποίοι είναι μία αφαίρεση στο επίπεδο εφαρμογής και περικλείουν τα πακέτα κώδικα και όλες τις εξαρτήσεις (`dependencies`) μαζί. Πολλαπλοί `containers` μπορούν να εκτελεστούν πάνω στο ίδιο μηχάνημα χρησιμοποιώντας το `kernel` του λειτουργικού συστήματος με τον καθένα να τρέχει σε ένα απομονωμένο περιβάλλον. Παράλληλα, γίνεται χρήση των `docker images` που ουσιαστικά είναι αρχεία τα οποία εκτελούν κώδικα μέσα σε κάποιον `container`. Συγκεκριμένα, στην εφαρμογή που έχει αναπτυχθεί, με τη χρήση μιας συγκεκριμένης εντολής εκτελείται κατευθείαν η εφαρμογή μας, ενημερώνονται όλες οι εξαρτήσεις στη νεότερη έκδοση, εκτελείται ο `server`, η βάση δεδομένων, και το `phpMyAdmin`.

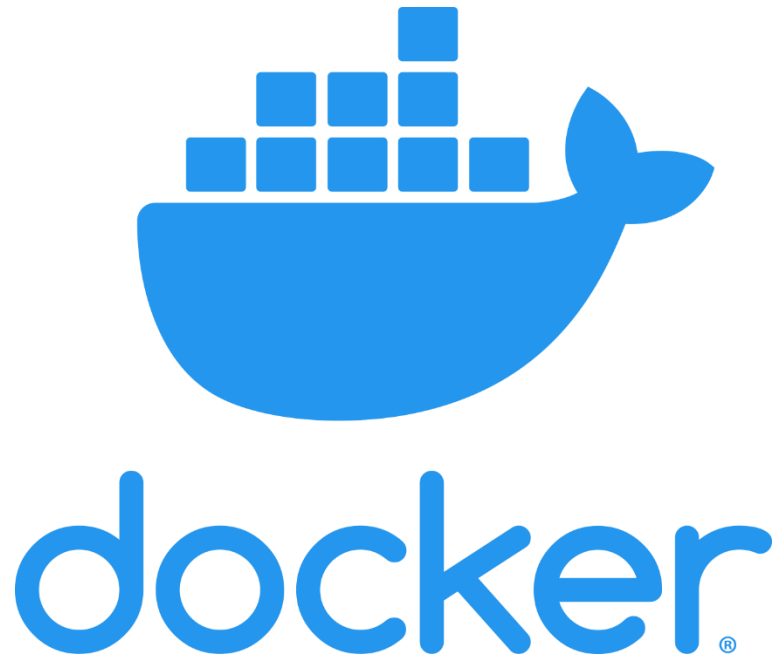


Figure 3.6: Το Λογότυπο του Docker

Ο επεξεργαστής κειμένου που χρησιμοποιήθηκε είναι ο Visual Studio Code (<https://code.visualstudio.com/>). Επιπλέον, πρέπει να αναφερθεί ότι παρόλο που η εφαρμογή έχει αναπτυχθεί σε τοπικό επίπεδο, η μεταφορά της σε διαδικτυακό διακομιστή δεν θα ήταν δύσκολη, καθώς η παραμετροποιήσεις που θα έπρεπε να κάνουμε δεν θα ήταν πολλές.

3.5 Επεκτάσεις και Βιβλιοθήκες

Κατά την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν επεκτάσεις για το πλαίσιο Flask και κάποιες βιβλιοθήκες στη γλώσσα προγραμματισμού Python, οι οποίες είναι απαραίτητες για να υλοποιηθούν συγκεκριμένες λειτουργίες που ήταν αναγκαίες για την εφαρμογή. Σε αυτή την ενότητα αναφέρονται και αναλύονται εν συντομία επεκτάσεις που χρησιμοποιήθηκαν.

Flask-WTF: Πρόκειται για υλοποίηση του πλαισίου Flask με τη βιβλιοθήκη WTForms της Python. Η βιβλιοθήκη υποστηρίζει φόρμες υποβολής, προστασία από επιθέσεις CSRF, Internationalization(I18N) κ.α. Αυτή η επέκταση κάνει εύκολη τη χρήση των WTForms με το πλαίσιο Flask.

Flask-Login: Αυτή η επέκταση δίνει τη δυνατότητα διαχείρισης συνεδρίας χρήστη (user session management). Παρέχει δυνατότητες εισόδου(login) και εξόδου(logout) καθώς και τη δυνατότητα απομνημόνευσης συνεδρίας χρήστη

(remember me) για παρατεταμένο χρονικό διάστημα. Επίσης, παρέχει τη δυνατότητα να απαγορεύεται η πρόσβαση σε σελίδες που είναι αποκλειστικά για εγγεγραμμένους χρήστες.

Flask-Bcrypt.: Το Bcrypt είναι μια επέκταση για το πλαίσιο Flask που παρέχει δυνατότητες κατακερματισμού(hashing). Το Bcrypt προορίζεται για την προστασία ευαίσθητων πληροφοριών όπως είναι για παράδειγμα οι κωδικοί πρόσβασης ενός χρήστη.

Flask-Babel: Το Flask-Babel είναι μία επέκταση για το Flask το οποίο προσθέτει υποστήριξη i18n και l10n σε οποιαδήποτε εφαρμογή που υλοποιείται με το συγκεκριμένο πλαίσιο. Έχει ενσωματωμένη υποστήριξη για μορφοποίηση ημερομηνιών καθώς και εργαλείο λήψης κειμένου προς μετάφραση.

Flask-SQLAlchemy: Το Flask-SQLAlchemy είναι μία επέκταση για το Flask που δίνει δυνατότητα υποστήριξης για την SQLAlchemy πάνω στην εφαρμογή. Σκοπός του είναι να απλοποιήσει τη χρήση της SQLAlchemy με το Flask παρέχοντας εργαλεία που βοηθούν στην εκτέλεση των εργασιών.

Όπως χρειάστηκαν κάποιες επεκτάσεις του Flask, αντίστοιχα χρειάστηκαν και κάποιες βιβλιοθήκες της Python όπως η Pillow(PIL), που ήταν απαραίτητη για την παραμετροποίηση διαστάσεων φωτογραφιών που ανεβάζει στην εφαρμογή ο διαχειριστής ή η βιβλιοθήκη για τη χρήση του Stripe που αναφέρεται παρακάτω.

3.6 Front – End

Για το Front – End κομμάτι της εφαρμογής, πέρα από απλή HTML, χρησιμοποιήθηκε το πλαίσιο CSS Bootstrap για τη σχεδίαση όμορφων σελίδων ανεξαρτήτως των διαστάσεων της οθόνης στην οποία εμφανίζονται (responsive and mobile-friendly websites). Επίσης, σε πολλές περιπτώσεις έχει γραφτεί επιπλέον CSS όπου αυτό κρίθηκε αναγκαίο, για την καλύτερη εμφάνιση σε πολλαπλούς φυλλομετρητές (Google Chrome, Mozilla Firefox κτλ.) και σε διαφορετικές συσκευές (PCs, smartphones, tablets κτλ.). Τέλος, για την καλύτερη απόκριση σε φορητές συσκευές έχει επιλεγεί ένα template που επιτρέπει την εύκολη υλοποίηση του μενού του διαχειριστή (dashboard).

3.7 Άλλες υπηρεσίες που χρησιμοποιήθηκαν

Σε αυτή την ενότητα γίνεται αναφορά στις υπηρεσίες MailGun και Stripe που χρησιμοποιήθηκαν για την λειτουργία του Newsletter και για την ολοκλήρωση της κράτησης με ηλεκτρονική πληρωμή.

3.7.1 MailGun

Το mailgun είναι υπηρεσία που παρέχει APIs, τα οποία δίνουν τη δυνατότητα αποστολής, λήψης και παρακολούθησης email. Μέσω των υπηρεσιών του mailgun έχουμε τη δυνατότητα να φτιάξουμε μία λίστα με διευθύνσεις ηλεκτρονικού ταχυδρομείου και ύστερα να προωθήσουμε το υλικό που επιθυμούμε.

3.7.2 Stripe

Κλείνοντας το κεφάλαιο, πρέπει να αναφέρουμε την υπηρεσία που χρησιμοποιεί η εφαρμογή για την ολοκλήρωση της κράτησης, που δεν είναι άλλη από τη διαδικασία ηλεκτρονικής πληρωμής. Η Stripe ουσιαστικά δίνει τη δυνατότητα σε ιδιώτες και επιχειρήσεις να πραγματοποιούν και να λαμβάνουν πληρωμές μέσω του διαδικτύου.

Για να χρησιμοποιήσει κάποιος την υπηρεσία και το API της Stripe πρέπει αρχικά να δημιουργήσει έναν δωρεάν λογαριασμό όπου θα λάβει ένα δημόσιο κλειδί (Public Key) και ένα ιδιωτικό κλειδί (Private Key). Στη συνέχεια, μέσω της τεκμηρίωσης ο προγραμματιστής μπορεί να επιλέξει τι ακριβώς θα υλοποιήσει. Στο κεφάλαιο «Οδηγός χρήσης λογισμικού για προγραμματιστή» θα αναφερθούμε στην υλοποίηση που χρησιμοποιεί η πλατφόρμα.



Figure 3.7: Λογότυπα των υπηρεσιών mailgun και stripe

4. Οδηγός χρήσης λογισμικού για τον Διαχειριστή της πλατφόρμας

Η διαδικτυακή πλατφόρμα που αναπτύχθηκε προσφέρει εργαλεία για τη διαχείριση του τουριστικού καταλύματος. Μέσω της διαχείρισης του καταλύματος ο διαχειριστής (admin) μπορεί να βελτιώσει τον τρόπο με τον οποίο θα λειτουργεί το κατάλυμά του και επιπρόσθετα να λαμβάνει χρήσιμες πληροφορίες για τις τρέχουσες κρατήσεις.

4.1 Για τον Πίνακα Ελέγχου

Η εφαρμογή προσφέρει ένα εύχρηστο πίνακα ελέγχου μέσω του οποίου ο διαχειριστής μπορεί να παρακολουθεί τις κρατήσεις που έχει το κατάλυμα σε πραγματικό χρόνο.

Ο διαχειριστής μπορεί να παρακολουθεί τις κρατήσεις που έχουν γίνει από χρήστες εγγεγραμμένους στην πλατφόρμα ή και χρήστες που έκαναν κράτηση χωρίς εγγραφή, είτε σε μορφή πίνακα, είτε πάνω σε ημερολόγιο. Έχει τη δυνατότητα επίσης να αλλάξει τη φωτογραφία του κάθε δωματίου που προσφέρεται και την προεπισκόπηση της φωτογραφίας πριν την ανάρτησή (upload) της. Επιπλέον, ο διαχειριστής μπορεί να αλλάξει την τρέχουσα τιμή για κάθε δωμάτιο. Τέλος, δίνεται η δυνατότητα να προστεθεί καινούριο δωμάτιο ή να γίνει αφαίρεση ενός υπάρχοντος δωματίου.

Σκοπός είναι ο πίνακας ελέγχου να είναι όσο το δυνατόν πιο εύχρηστος και φιλικός προς τον χρήστη, καθώς και να ανταποκρίνεται σε πολλαπλές συσκευές με διαφορετικές διαστάσεις οθόνης.

4.2 Είσοδος στην εφαρμογή

Ο διαχειριστής (admin) μπορεί να κάνει είσοδο στην εφαρμογή χρησιμοποιώντας τον παρακάτω σύνδεσμο (υποθέτουμε ότι είμαστε σε τοπικό επίπεδο), <http://localhost:5000/user-login/> και χρησιμοποιώντας τα διαπιστευτήρια (credentials) που ακολουθούν. Το e-mail για την είσοδο του διαχειριστή στον πίνακα ελέγχου είναι «admin@mail.com» και ο κωδικός πρόσβασης είναι «123».

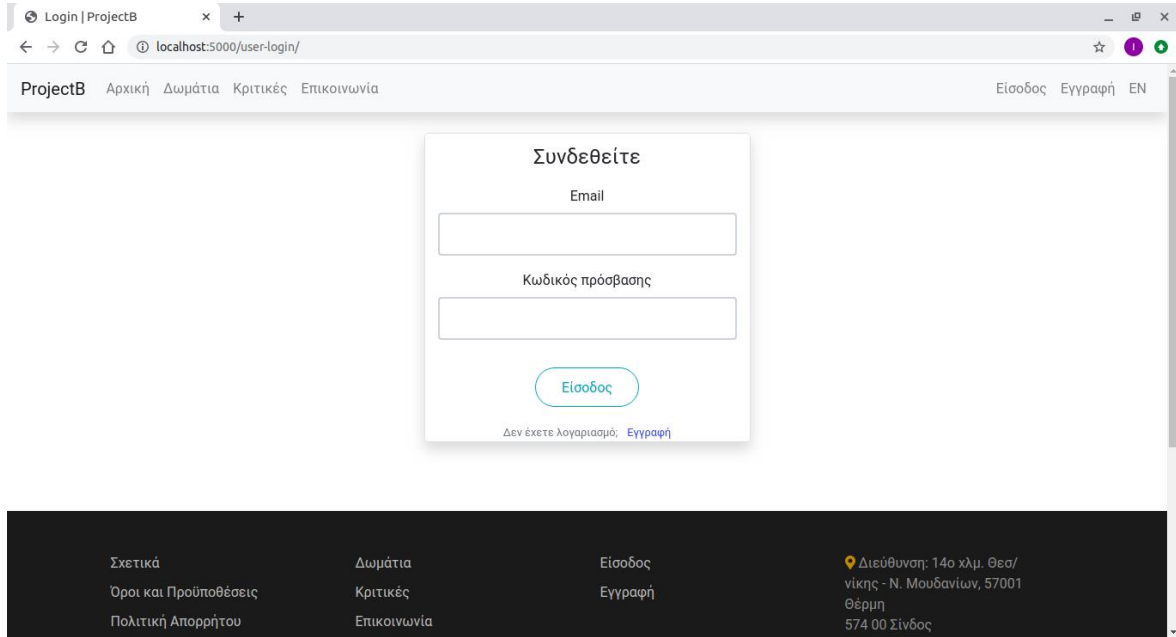


Figure 4.1: Η σελίδα εισόδου χρηστών στην πλατφόρμα

Ο διαχειριστής, μετά την είσοδο στην εφαρμογή θα ανακατευθυνθεί στην αρχική σελίδα της πλατφόρμας όπου θα έχει εμφανιστεί η επιλογή «Πίνακας Ελέγχου» στο κεντρικό μενού.

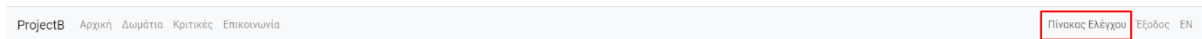


Figure 4.2: Η επιλογή "Πίνακας Ελέγχου" μετά την είσοδο του διαχειριστή

Με την επιλογή «Πίνακας Ελέγχου» ο χρήστης ανακατευθύνεται στην κεντρική σελίδα του μενού διαχείρισης.

4.3 Περιβάλλον εφαρμογής Πίνακα Ελέγχου

Με την είσοδο στον πίνακα ελέγχου ο διαχειριστής έχει πρόσβαση στα εργαλεία και τις δυνατότητες για τη διαχείριση του καταλύματος.

Η αρχική σελίδα που ανακατευθύνεται ο διαχειριστής με την επιλογή «Πίνακας Ελέγχου» είναι αυτή που περιέχει το ημερολόγιο κρατήσεων. Η δομή του πίνακα ελέγχου είναι απλή και παραμένει ίδια σε κάθε επιλογή του διαχειριστή. Ουσιαστικά αποτελείται από 4 κύρια συστατικά: το αριστερό μενού, το κύριο παράθυρο, την επάνω μπάρα πλοήγησης και την κάτω μπάρα πλοήγησης. Η εικόνα που ακολουθεί παρουσιάζει την καρτέλα «Κρατήσεις».

PROJECTS Dashboard Log out

Κρατήσεις από κατοχυρωμένους χρήστες:

NO	ΌΝΟΜΑ	ΕΠΙΘΕΤΟ	E-MAIL	ΔΙΕΥΘΥΝΣΗ	ΑΡΙΘΜΟΣ ΤΗΛΕΦΩΝΟΥ	CHECK-IN	CHECK-OUT	ΔΟΜΑΤΙΟ	ΠΟΣΟ ΤΥΠΩΜΑΤΩΣ
1	User1	User1	user1@mail.com	address1	6969696969	1/4/2020	4/4/2020	Deluxe	250.0
2	User2	User2	user2@mail.com	address2	6969696969	10/4/2020	13/4/2020	Double	250.0

Δεν έχετε κρατήσεις από μη κατοχυρωμένους χρήστες ακόμα.

Home Rooms Reviews © 2020 This is project 3

Figure 4.3: Άποψη της καρτέλας "Κρατήσεις" στον Πίνακα Ελέγχου

Πιο αναλυτικά οι επιλογές:

1. Αριστερό μενού πλοήγησης: στην ουσία όλες οι επιλογές του διαχειριστή βρίσκονται σε αυτό το μενού. Ο διαχειριστής μπορεί να επιλέξει ένα από τα έξι εργαλεία που διατίθενται για την ώρα.
2. Κύριο Παράθυρο: σε αυτό το παράθυρο εμφανίζονται όλες οι πληροφορίες και οι επιλογές που έχει στη διάθεση του ο διαχειριστής. Στην προκειμένη περίπτωση, εμφανίζονται όλες οι εγγραφές σε κρατήσεις από εγγεγραμμένους ή μη χρήστες.
3. Επάνω Μπάρα Πλοήγησης: η επάνω μπάρα πλοήγησης όπως φαίνεται στη φωτογραφία. Προς το παρόν έχει μόνο την επιλογή της αποσύνδεσης του διαχειριστή από την πλατφόρμα.
4. Κάτω Μπάρα Πλοήγησης: η κάτω μπάρα πλοήγησης όπως φαίνεται στη φωτογραφία. Περιλαμβάνει συνδέσμους όπως η Αρχική (Home) σελίδα της εφαρμογής.

4.3.1 Το αριστερό μενού του Πίνακα Ελέγχου

Το αριστερό μενού του πίνακα ελέγχου περιλαμβάνει στην ουσία όλες τις λειτουργίες που έχει στη διάθεσή του ο διαχειριστής. Το μενού αποτελείται για την ώρα από 6 διακριτές υποκατηγορίες όπως φαίνεται στην εικόνα που ακολουθεί:

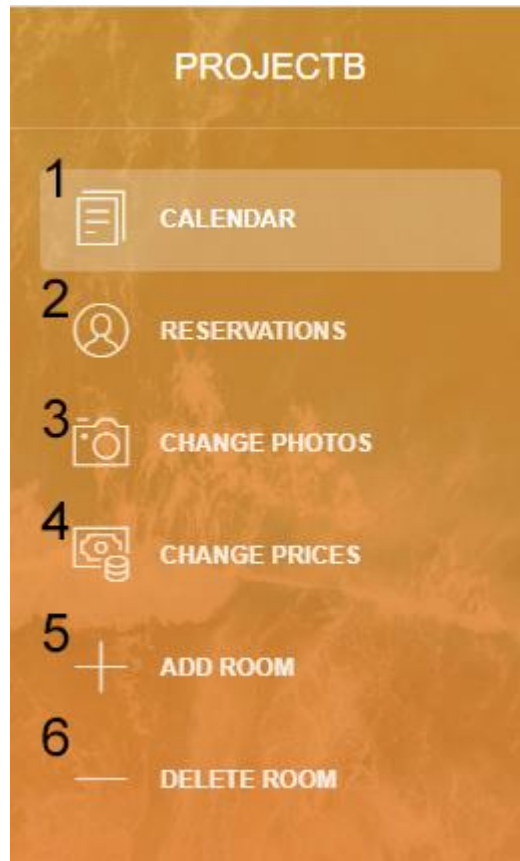


Figure 4.4: Οι επιλογές του Διαχειριστή

Αναλυτικότερα, οι επιλογές είναι οι εξής:

1. Ημερολόγιο: Εμφάνιση του ημερολογίου με τις υπάρχουσες κρατήσεις και τη δυνατότητα ανοίγματος ή κλεισίματος ημερομηνιών
2. Κρατήσεις: Αναλυτική εμφάνιση των κρατήσεων που έχουν πραγματοποιηθεί από εγγεγραμμένους ή μη χρήστες σε πίνακες.
3. Αλλαγή Φωτογραφίας: Δυνατότητα αλλαγής της φωτογραφίας ενός δωματίου.
4. Αλλαγή Τιμής: Δυνατότητα αλλαγής της τιμής ενός δωματίου.
5. Προσθήκη Δωματίου: Δυνατότητα προσθήκης καινούργιου δωματίου.
6. Αφαίρεση Δωματίου: Δυνατότητα αφαίρεσης ενός δωματίου.

Στην ενότητα 4.4 ακολουθεί αναλυτική επεξήγηση των παραπάνω επιλογών που έχει στη διάθεσή του ο διαχειριστής.

4.3.2 Το κεντρικό παράθυρο του Πίνακα Ελέγχου

Στο κεντρικό παράθυρο του πίνακα ελέγχου παρουσιάζονται οι πληροφορίες που αντιστοιχούν στην επιλεγμένη καρτέλα. Ανάλογα με την επιλογή του διαχειριστή εμφανίζονται πληροφορίες ή φόρμες υποβολής για την αντίστοιχη ενέργεια. Για παράδειγμα, εάν επιλεγεί η καρτέλα «Αφαίρεση Δωματίου» στο κεντρικό παράθυρο θα εμφανιστεί μία λίστα (drop-down list) που περιλαμβάνει όλα τα δωμάτια της βάσης δεδομένων και το κουμπί «Διαγραφή». Επιλέγοντας το δωμάτιο που επιθυμεί να αφαιρέσει ο διαχειριστής και κάνοντας κλικ στο κουμπί «Διαγραφή» ολοκληρώνει αυτή την ενέργεια.

4.3.3 Η επάνω και η κάτω μπάρα πλοήγηση του Πίνακα Ελέγχου

Οι επάνω και κάτω μπάρες πλοήγησης περιλαμβάνουν επιλογές όπως η αποσύνδεση του διαχειριστή από την πλατφόρμα πατώντας στον σύνδεσμο με την ονομασία «Αποσύνδεση». Επίσης περιλαμβάνει συνδέσμους που τον ανακατευθύνουν στην αρχική σελίδα της πλατφόρμας, στη σελίδα των δωματίων και στη σελίδα με τις κριτικές των χρηστών.

4.4 Οι επιλογές του διαχειριστή

Όπως έχει αναφερθεί ήδη ο διαχειριστής έχει 6 βασικά εργαλεία στη διάθεσή του για τη διαχείριση του καταλύματός του. Σε αυτή την ενότητα γίνεται ανάλυση σε αυτές τις επιλογές.

4.4.1 Ημερολόγιο

Είναι η αρχική σελίδα που βλέπει ο χρήστης όταν εισέρχεται στον πίνακα ελέγχου. Η καρτέλα «Ημερολόγιο» περιλαμβάνει 3 κύρια χαρακτηριστικά που είναι το ημερολόγιο, η φόρμα για κλείσιμο ημερομηνιών και η φόρμα ανοίγματος των ημερομηνιών.

Πτυχιακή εργασία του φοιτητή Κοσμά Ηλία

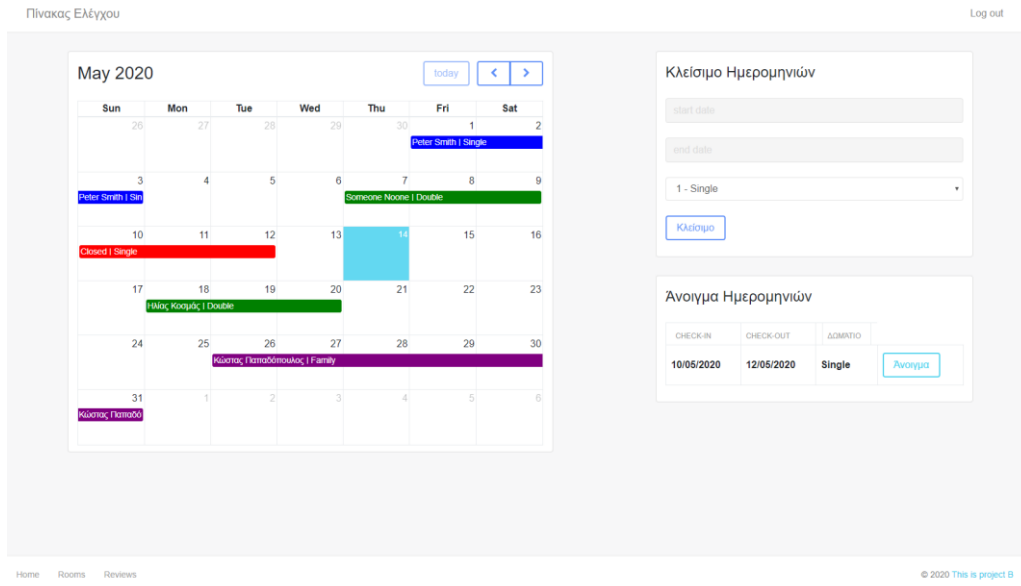


Figure 4.5: Το περιεχόμενο της καρτέλας "Ημερολόγιο"

Ξεκινώντας από τη μορφή του Ημερολογίου διακρίνουμε τα εξής:

- Τον τρέχοντα μήνα: ο τρέχων μήνας αναγράφεται στο επάνω αριστερά μέρος του Ημερολογίου.
- Τα κουμπιά αλλαγής μήνα: στο επάνω δεξιά μέρος του ημερολογίου διακρίνουμε τα κουμπιά με το αριστερό και δεξί βελάκι. Κάνοντας χρήση ο διαχειριστής μπορεί να πλοηγηθεί σε προηγούμενο ή σε επόμενο μήνα.
- Γραμμή Κράτησης: η γραμμή κράτησης αναπαριστά την κράτηση επάνω στο ημερολόγιο καταλαμβάνοντας τις ημέρες που έχει γίνει η κράτηση. Στη γραμμή κράτησης αναγράφεται το όνομα και το επίθετο του πελάτη καθώς και ο τύπος δωματίου που έχει γίνει η κράτηση. Η γραμμή κράτησης έχει διαφορετικό χρώμα για τα βασικά δωμάτια ανάλογα με τον τύπο δωματίου που αντιπροσωπεύει:

- | | |
|---|--------------------|
| a. Δωμάτιο Single : | μπλε χρώμα |
| b. Δωμάτιο Double: | πράσινο χρώμα |
| c. Δωμάτιο Deluxe: | μπλε ανοικτό χρώμα |
| d. Δωμάτιο Family: | μωβ χρώμα |
| e. Δωμάτια που προστίθενται: | πορτοκαλί χρώμα |
| f. Δωμάτια κλειστά από τον Διαχειριστή: | κόκκινο χρώμα |

Δεξιά του Ημερολογίου διακρίνουμε το παράθυρο με όνομα «Κλείσιμο Ημερομηνιών». Το συγκεκριμένο παράθυρο περιλαμβάνει τα πεδία «start date», «end date», τη «Λίστα Δωματίων» και το κουμπί «Κλείσιμο».

Επίσης, παρατηρούμε το παράθυρο με όνομα «Άνοιγμα Ημερομηνιών» που περιλαμβάνει ένα πίνακα με τα πεδία «Check-in», «Check-out», «Δωμάτιο» και το κουμπί «Άνοιγμα». Ακριβώς από κάτω εμφανίζονται το εύρος των ημερομηνιών που έχει επιλέξει ο διαχειριστής να είναι κλειστό.

Για να κλείσουμε ένα δωμάτιο ως διαχειριστές, πρέπει να κάνουμε τις εξής ενέργειες. Αρχικά, για να επιλέξουμε το εύρος των ημερομηνιών που επιθυμούμε να κλείσουμε φέρνουμε το ποντίκι στην πρώτη μέρα του εύρους που θέλουμε και σέρνουμε το ποντίκι στην τελευταία μέρα που επιθυμούμε. Το αποτέλεσμα της κίνησης αυτής φαίνεται στην παρακάτω εικόνα:

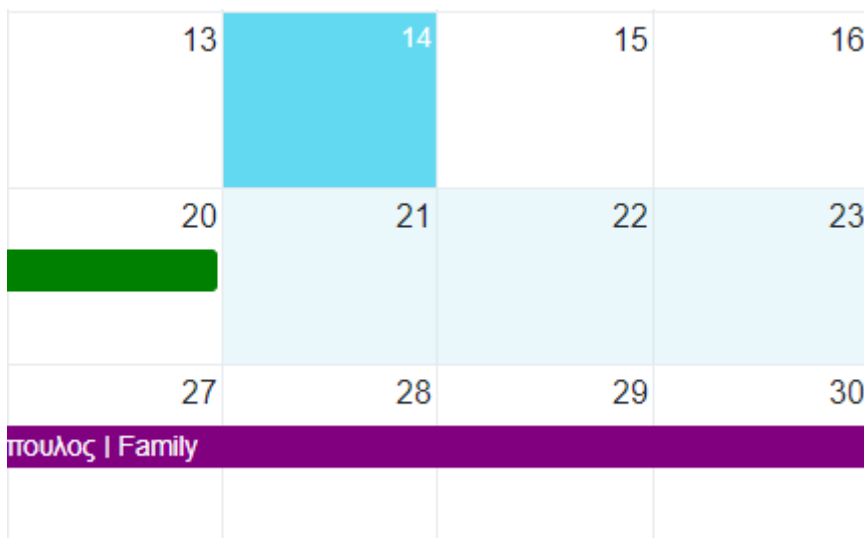


Figure 4.6: Επιλεγμένες ημερομηνίες από τον διαχειριστή

Στο παραπάνω παράδειγμα έχουμε επιλέξει το εύρος ημερομηνιών από τις 21/5/2020 έως 23/5/2020, το οποίο διακρίνεται με το αχνό μπλε χρώμα. Μπορούμε να βεβαιώσουμε ότι έχουμε επιλέξει τις ημερομηνίες που επιθυμούμε, κοιτώντας τα πεδία «start date» και «end date» στο παράθυρο «Κλείσιμο Ημερομηνιών» που ανανεώνεται αυτόματα με κάθε επιλογή εύρους από τον διαχειριστή. Για να ολοκληρωθεί το κλείσιμο των συγκεκριμένων ημερομηνιών και εφόσον έχει επιλεγεί το δωμάτιο από το πεδίο που θέλουμε να κλείσουμε τις ημερομηνίες, ο διαχειριστής κάνει χρήση του κουμπιού «Κλείσιμο». Αν όλα έχουν γίνει σωστά το σύστημα θα μας αποστείλει κατάλληλο μήνυμα επιτυχίας και θα προστεθεί μία Γραμμή Κράτησης που θα καταλαμβάνει το εύρος ημερομηνιών 21/5/2020 έως και 23/5/2020 με την ένδειξη «Closed» και τον τύπο του δωματίου όπως φαίνεται στην εικόνα που ακολουθεί:

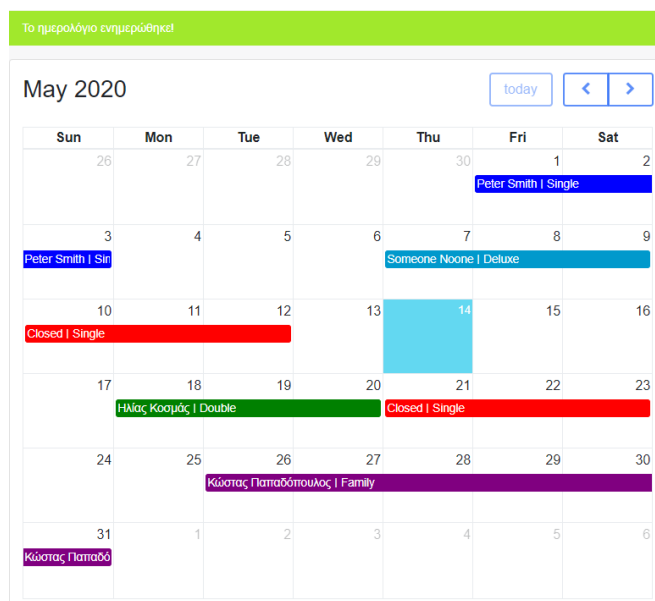


Figure 4.7: Μήνυμα συστήματος

Τέλος, για το κλείσιμο των ημερομηνιών πρέπει να γνωρίζουμε ότι ο διαχειριστής δεν έχει τη δυνατότητα επιλογής ημερομηνιών που έχει ήδη γίνει κράτηση.

Για το άνοιγμα των ημερομηνιών στο διαχειριστή διατίθεται το παράθυρο «Άνοιγμα Ημερομηνιών» το οποίο περιλαμβάνει όλα τα εύρη ημερομηνιών που έχει κλείσει σε έναν πίνακα. Για να ανοίξει το εύρος ημερομηνιών που επιθυμεί ο διαχειριστής μπορεί να πατήσει το κουμπί «Άνοιγμα» στην αντίστοιχη κατοχύρωση. Το σύστημα θα αποστείλει ένα μήνυμα επιτυχίας ότι οι ημερομηνίες είναι ανοιχτές και η Γραμμή Κράτησης θα εξαφανιστεί από το ημερολόγιο.

4.4.2 Κρατήσεις

Η καρτέλα «Κρατήσεις» είναι η δεύτερη κατά σειρά επιλογή που έχει ο διαχειριστής στη διάθεσή του. Όταν επιλεγεί στο κεντρικό παράθυρο εμφανίζονται σε πίνακα όλες οι πληροφορίες των πελατών που έχουν κάνει κράτηση στο κατάλυμα μέσω της πλατφόρμας.

Πραγματοποιούνται δύο είδη κρατήσεων μέσω της πλατφόρμας. Κράτηση για εγγεγραμμένους χρήστες και κράτηση για μη εγγεγραμμένους χρήστες. Το σύστημα επιστρέφει δύο πίνακες αντίστοιχα. Όταν δεν υπάρχει καμία κράτηση από εγγεγραμμένο χρήστη το σύστημα εμφανίζει την ένδειξη «Δεν έχετε κρατήσεις από εγγεγραμμένους χρήστες ακόμα.» και αντίστοιχα την ένδειξη «Δεν έχετε κρατήσεις από μη εγγεγραμμένους χρήστες ακόμα.» για τους μη εγγεγραμμένους χρήστες.

Όταν υπάρχει κράτηση εμφανίζεται πίνακας που περιλαμβάνει 10 στήλες οι οποίες είναι οι εξής:

1. ΝΟ: ο α/α των κρατήσεων στον πίνακα
2. ΟΝΟΜΑ: το όνομα του πελάτη
3. ΕΠΙΘΕΤΟ: το επίθετο του πελάτη
4. E-MAIL: η διεύθυνση ηλεκτρονικού ταχυδρομείου του πελάτη
5. ΔΙΕΥΘΥΝΣΗ: η διεύθυνση που έχει δηλώσει ο πελάτης
6. ΑΡΙΘΜΟΣ ΤΗΛΕΦΩΝΟΥ: ο αριθμός τηλεφώνου που έχει δηλώσει ο πελάτης
7. CHECK-IN: η ημερομηνία άφιξης του πελάτη
8. CHECK-OUT: η ημερομηνία αναχώρησης του πελάτη
9. ΔΩΜΑΤΙΟ: ο τύπος δωματίου για το οποίο έχει γίνει η κράτηση
10. ΠΟΣΟ ΠΛΗΡΩΜΗΣ: το συνολικό ποσό πληρωμής σε € που καταβλήθηκε για την κράτηση

4.4.3 Αλλαγή Φωτογραφίας

Η τρίτη κατά σειρά επιλογή στο αριστερό μενού του πίνακα ελέγχου είναι η καρτέλα «Αλλαγή Φωτογραφίας». Όπως υποδηλώνει και το όνομα, αυτή η καρτέλα δίνει τη δυνατότητα στον διαχειριστή να αλλάξει τη φωτογραφία ενός δωματίου.

Όταν επιλεγεί η καρτέλα «Αλλαγή Φωτογραφίας» εμφανίζεται στο κύριο παράθυρο του πίνακα ελέγχου μία φόρμα με τα εξής στοιχεία: μία «Λίστα Δωματίων» ένα κουμπί «Επιλογή αρχείου» και ένα κουμπί «Ενημέρωση Φωτογραφίας». Επίσης παρατηρούμε τον τίτλο «Προεπισκόπηση Φωτογραφίας».

Για να αλλάξουμε φωτογραφία σε κάποιο από τα δωμάτια του καταλύματός μας ακολουθούμε την εξής διαδικασία:

- Από τη «Λίστα Δωματίων» επιλέγουμε το δωμάτιο στο οποίο επιθυμούμε να αλλάξουμε τη φωτογραφία κάνοντας κλικ επάνω στη λίστα για να μας εμφανίσει τα δωμάτια που έχει το κατάλυμά μας.
- Κάνουμε κλικ στο κουμπί «Επιλογή αρχείου»
- Μας εμφανίζεται ένα νέο παράθυρο επιλογής αρχείου
- Βρίσκουμε τη φωτογραφία που επιθυμούμε να ανεβάσουμε και την επιλέγουμε
- Με το που επιλέξουμε τη φωτογραφία που θέλουμε να ανεβάσουμε το όνομα της επιλεγμένης φωτογραφίας εμφανίζεται δίπλα στο κουμπί «Επιλογή αρχείου» και κάτω από τον τίτλο «Προεπισκόπηση Φωτογραφίας» εμφανίζεται η φωτογραφία που έχουμε επιλέξει
- Για να ολοκληρώσουμε την αλλαγή επιλέγουμε το κουμπί «Ενημέρωση Φωτογραφίας»

Στη φωτογραφία που ακολουθεί ο χρήστης έχει επιλέξει να αλλάξει τη φωτογραφία στο Δωμάτιο Single, με το όνομα της φωτογραφίας να φαίνεται στα δεξιά του κουμπιού «Επιλογή αρχείου». Κάτω από τον τίτλο «Προεπισκόπηση Φωτογραφίας» βλέπουμε τη φωτογραφία που πρόκειται να αντικαταστήσει την υπάρχουσα φωτογραφία του Δωματίου Single. Με την επιλογή «Ενημέρωση Φωτογραφίας» το σύστημα θα μας αποστείλει μήνυμα επιτυχίας ότι η φωτογραφία έχει αλλάξει με επιτυχία.

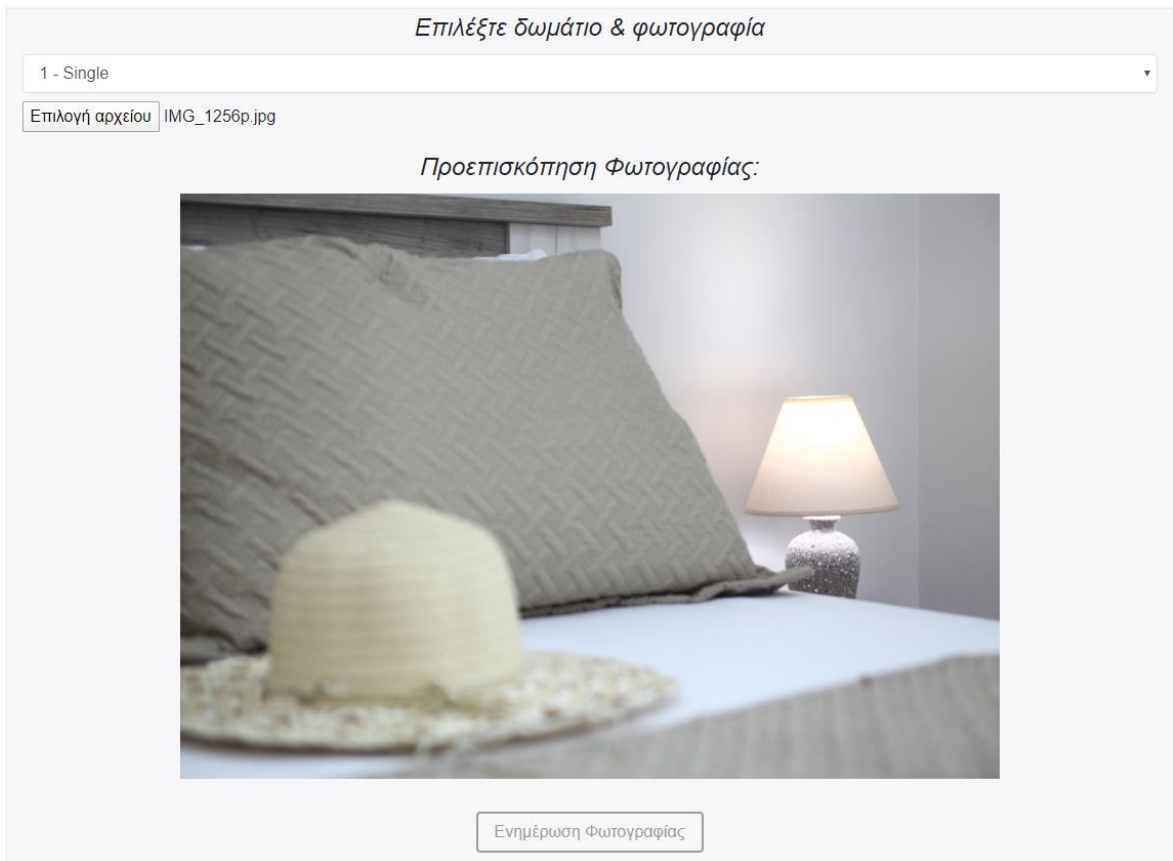


Figure 1: Αλλαγή Φωτογραφίας

Κλείνοντας την ενότητα για την αλλαγή φωτογραφίας πρέπει να αναφέρουμε τα εξής:

- Οι τύποι αρχείων φωτογραφίας που επιτρέπονται από την πλατφόρμα αυτή τη στιγμή είναι οι «jpg» και «png»
- Εάν επιλέξουμε άλλο τύπου αρχείου το σύστημα θα μας ειδοποιήσει με κατάλληλο μήνυμα
- Οι διαστάσεις που προτείνονται είναι 800 x 571 pixels. Στην προεπισκόπηση φωτογραφίας η επιλεγμένη φωτογραφία μορφοποιείται σε διαστάσεις 800 x 571 pixels. Εάν η φωτογραφία που έχει επιλεγθεί δεν είναι σε αυτές τις διαστάσεις το σύστημα θα την μορφοποιήσει σε διαστάσεις 800 x 571 pixels.

4.4.4 Αλλαγή Τιμής

Η τέταρτη επιλογή στα εργαλεία του διαχειριστή είναι η καρτέλα «Αλλαγή Τιμής». Στην καρτέλα αυτή ο διαχειριστής έχει τη δυνατότητα να αλλάξει την τιμή ανά βραδιά σε οποιοδήποτε από τα δωμάτια του καταλύματος.

Με την επιλογή της καρτέλας «Αλλαγή Τιμής» στο κύριο παράθυρο εμφανίζονται τα εξής στοιχεία: οι τρέχουσες τιμές όλων των δωματίων ανά βραδιά, μία λίστα δωματίων, το πεδίο «Νέα Τιμή» και το κουμπί «Ενημέρωση Τιμής»

Ο διαχειριστής για να αλλάξει τιμή σε ένα από τα δωμάτια πρέπει να επιλέξει το δωμάτιο που επιθυμεί και να συμπληρώσει στο πεδίο «Νέα Τιμή» το νέο ποσό που επιθυμεί να χρεώνει ανά βραδιά. Ύστερα επιλέγοντας το κουμπί «Ενημέρωση Τιμής» ολοκληρώνει την ενημέρωση. Το σύστημα θα αποστείλει κατάλληλο μήνυμα επιτυχίας. Τέλος, να σημειωθεί ότι η πλατφόρμα δέχεται μόνο ακέραιους αριθμούς για την ενημέρωση τιμής.

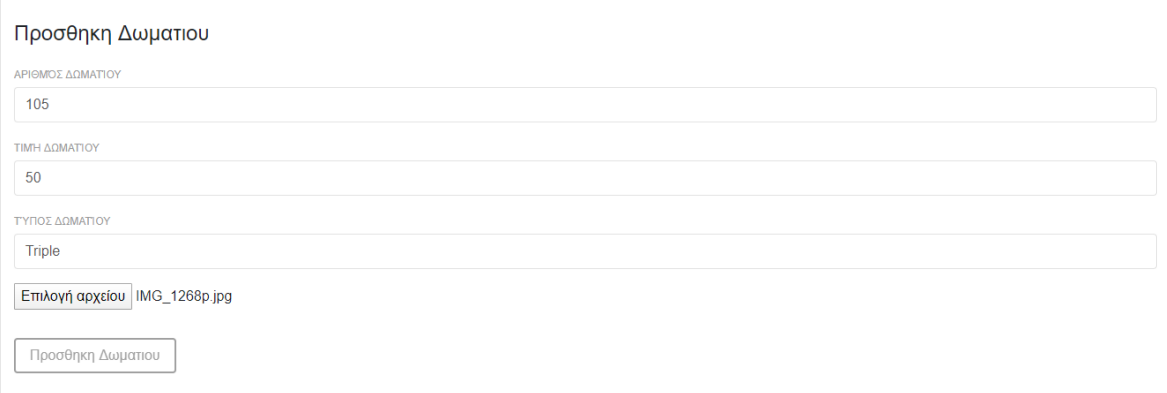
4.4.5 Προσθήκη Δωματίου

Η πέμπτη επιλογή που έχει ο διαχειριστής είναι αυτή της «Προσθήκης Δωματίου» όπου δίνεται η δυνατότητα να προσθέσουμε ένα νέο δωμάτιο στο κατάλυμά μας.

Όταν επιλέξουμε την καρτέλα «Προσθήκη Δωματίου» στο κύριο παράθυρο εμφανίζεται μία φόρμα με τα εξής πεδία:

1. Αριθμός Δωματίου: ο αριθμός του δωματίου που θέλουμε (πρέπει να είναι ακέραιος αριθμός)
2. Τιμή Δωματίου: η τιμή ανά βραδιά που θέλουμε να χρεώνουμε (πρέπει να είναι ακέραια τιμή)
3. Τύπος Δωματίου: ο τύπος του δωματίου (δεν μπορεί να ξεπερνά τους 20 χαρακτήρες)
4. Επιλογή Φωτογραφίας: το αρχείο φωτογραφίας που θέλουμε να έχει το καινούργιο δωμάτιο (επιτρέπονται αρχεία «jpg» και «png»)
5. Προσθήκη Δωματίου: το κουμπί «Προσθήκη Δωματίου» ολοκληρώνει την ενέργεια του διαχειριστή

Στην παρακάτω φωτογραφία ο διαχειριστής προσθέτει καινούριο δωμάτιο με αριθμό δωματίου 105, τιμή ανά βραδιά 50€, τύπο δωματίου «Triple» και έχει επιλεγθεί αρχείο φωτογραφίας με όνομα «IMG_1268p».



The screenshot shows a web form titled "Προσθήκη Δωματίου" (Add Room). It contains several input fields and a button:

- A label "ΑΡΙΘΜΟΣ ΔΩΜΑΤΙΟΥ" (Room Number) above a text input field containing "105".
- A label "ΤΙΜΗ ΔΩΜΑΤΙΟΥ" (Room Price) above a text input field containing "50".
- A label "ΤΥΠΟΣ ΔΩΜΑΤΙΟΥ" (Room Type) above a text input field containing "Triple".
- A file selection area with a button labeled "Επιλογή αρχείου" (Select file) and the text "IMG_1268p.jpg" next to it.
- A button at the bottom labeled "Προσθήκη Δωματίου" (Add Room).

Figure 2: Προσθήκη Δωματίου

4.4.6 Διαγραφή Δωματίου

Η τελευταία καρτέλα στα εργαλεία του διαχειριστή είναι η «Αφαίρεση Δωματίου». Όταν επιλεγεί, εμφανίζεται μία φόρμα η οποία έχει τη λίστα με τα δωμάτια του καταλύματος και ένα κουμπί «Αφαίρεση».

Για την αφαίρεση ενός δωματίου από το κατάλυμα ο διαχειριστής επιλέγει από τη λίστα δεδομένων το δωμάτιο που θέλει να αφαιρέσει και ύστερα κάνει κλικ στο κουμπί «Αφαίρεση». Το σύστημα θα εμφανίσει μήνυμα αν είμαστε σίγουροι ότι θέλουμε να διαγράψουμε το δωμάτιο. Κάνοντας κλικ στο κουμπί «Κλείσιμο» ακυρώνουμε την ενέργεια και δε διαγράφουμε το δωμάτιο. Με κλικ στο κουμπί «Αφαίρεση» ολοκληρώνουμε τη διαγραφή του δωματίου και το σύστημα αποστέλει μήνυμα επιτυχίας.

5. Περιγραφή της εφαρμογής σε επίπεδο τελικού χρήστη

Η διαδικτυακή πλατφόρμα που αναπτύχθηκε δίνει ιδιαίτερη έμφαση στην εμπειρία πλοήγησης, καθώς έχει ως στόχο τη μετατροπή του απλού χρήστη σε πελάτη. Οι χρήστες που θα χρησιμοποιήσουν την πλατφόρμα ως εν δυνάμει πελάτες μπορούν να πλοηγηθούν εκτός της κεντρικής σελίδας (Home) που περιλαμβάνει όλα τα κύρια χαρακτηριστικά που ένας χρήστης ψάχνει για να προχωρήσει σε κράτηση ενός τουριστικού καταλύματος, στις σελίδες Δωμάτια (Rooms), Κριτικές (Reviews), Επικοινωνία (Contact), Είσοδος (Login), Εγγραφή (Register). Επιπλέον, υπάρχουν οι σελίδες για τη συμπλήρωση των στοιχείων του χρήστη, σε περίπτωση που θέλει να κάνει κράτηση και η σελίδα «Λογαριασμός» για χρήστες που έχουν κάνει εγγραφή στην πλατφόρμα. Στη διάθεσή τους επίσης, υπάρχει η κεντρική μπάρα πλοήγησης που περιλαμβάνει όλους τους συνδέσμους για τις σελίδες που αναφέρθηκαν παραπάνω, καθώς και κουμπί για την αλλαγή γλώσσας της πλατφόρμας. Τέλος, να αναφέρουμε ότι το front-end της εφαρμογής είναι σχεδιασμένο ώστε να προσαρμόζεται σε οθόνες διαφόρων διαστάσεων όπως σταθερών ή φορητών υπολογιστών, tablets ή κινητά τηλέφωνα. Ακολουθεί μία ανάλυση των επιμέρους στοιχείων του περιβάλλοντος της εφαρμογής..

5.1 Περιήγηση στο περιβάλλον εφαρμογής της πλατφόρμας

Όπως έχουμε ήδη αναφέρει, ο χρήστης μπορεί να πλοηγηθεί σε διάφορες σελίδες μέσα στην πλατφόρμα ανάλογα με αυτό που ψάχνει κάθε φορά. Στην ενότητα αυτή περιγράφουμε το περιεχόμενο που θα συναντήσει.

Ξεκινώντας, ο χρήστης βρίσκεται στην αρχική σελίδα της πλατφόρμας που περιέχει ένα μήνυμα καλωσορίσματος και εν συνεχεία έναν πίνακα για έλεγχο της διαθεσιμότητας δωματίου. Ο χρήστης μπορεί συμπληρώνοντας την ημερομηνία άφιξης, την ημερομηνία αναχώρησης και τον τύπο δωματίου που επιθυμεί, να ελέγξει αν υπάρχει διαθεσιμότητα.

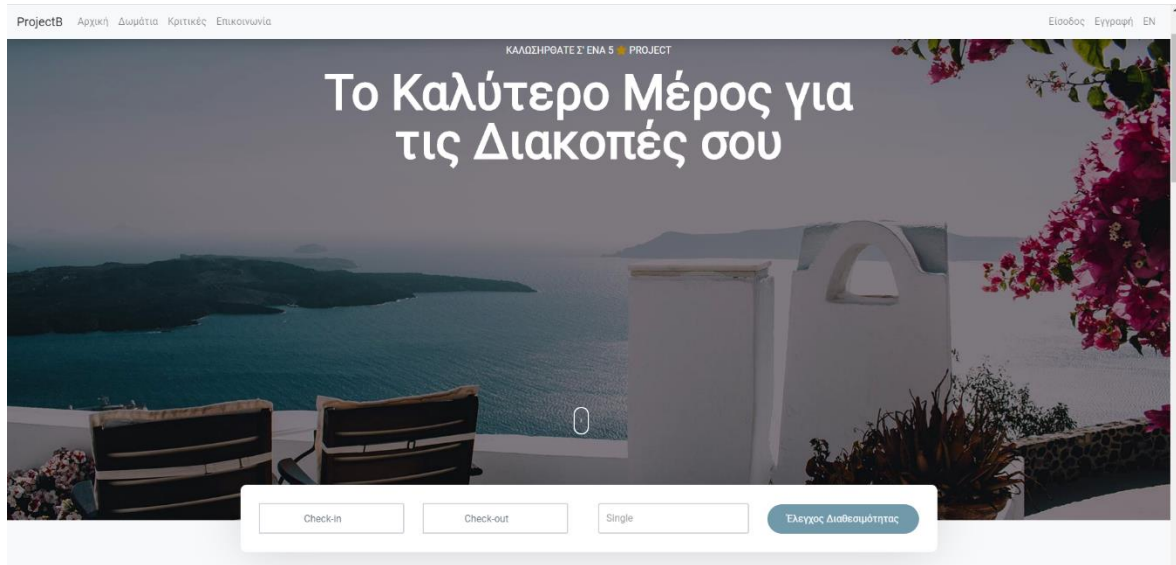


Figure 3: Η αρχική σελίδα της πλατφόρμας και η φόρμα για τον έλεγχο διαθεσιμότητας

Συνεχίζοντας, ο χρήστης μπορεί να δει την ενότητα που παρουσιάζει κάποια από τα δωμάτια που διαθέτει το κατάλυμα, καθώς και κριτικές που έχουν υποβληθεί από τους χρήστες. Επίσης, στη σελίδα «Επικοινωνία» ο χρήστης μπορεί να βρει βασικές πληροφορίες επικοινωνίας για το κατάλυμα.

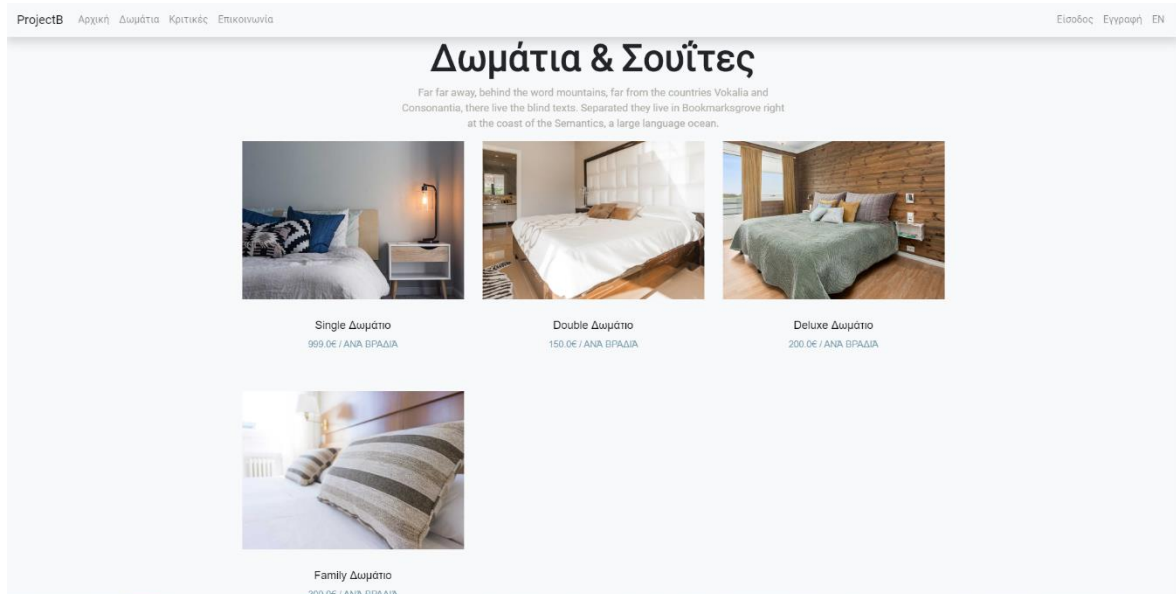


Figure 4: Η σελίδα "Δωμάτια"

Η πλατφόρμα επίσης προσφέρει τις σελίδες «Είσοδος», «Εγγραφή» και «Λογαριασμός». Στη σελίδα «Εγγραφή», ο χρήστης μπορεί να δημιουργήσει ένα καινούργιο λογαριασμό στην πλατφόρμα συμπληρώνοντας τα προσωπικά του στοιχεία. Δημιουργώντας λογαριασμό

ο χρήστης έχει τη δυνατότητα να εισέλθει στη σελίδα «Λογαριασμός», από όπου μπορεί να παρακολουθεί τις κρατήσεις του, εάν υπάρχουν, να υποβάλλει κριτική – σχόλιο για το κατάλυμα ή να εγγραφεί στην υπηρεσία Newsletter της πλατφόρμας. Για να εισέλθει στη σελίδα «Λογαριασμός», ο χρήστης θα πρέπει να συμπληρώσει το e-mail και τον κωδικό πρόσβασης στην σελίδα «Είσοδος». Ακολουθούν στιγμιότυπα από τις σελίδες «Είσοδος», «Εγγραφή», «Λογαριασμός».

ProjectB Αρχική Διαμότια Κριτικές Επικοινωνία Είσοδος Εγγραφή EN

Εγγραφείτε Σήμερα

Όνομα

Επίθετο

Email

Κωδικός πρόσβασης

Επιβεβαίωση κωδικού

Αριθμός Τηλεφώνου

Διεύθυνση

Εγγραφή

Έχετε ήδη λογαριασμό; Συνδεθείτε

Figure 5: Η σελίδα "Εγγραφή"

ProjectB Αρχική Διαμότια Κριτικές Επικοινωνία Είσοδος Εγγραφή EN

Συνδεθείτε

Email

Κωδικός πρόσβασης

Είσοδος

Δεν έχετε λογαριασμό; Εγγραφή

Figure 6: Η σελίδα "Είσοδος"

ProjectB Αρχική Δωμάτια Κριτικές Επικοινωνία Λογαριασμός Έξοδος EN

Οι κρατήσεις σας:

No	Όνομα	Επίθετο	Email	Διεύθυνση	Αριθμός Τηλεφώνου	Check-in	Check-out	Δωμάτιο
0	Κώστας	Παπαδοπουλος	kp@mail.com	Καρου 20	6969696969	20/5/2020	23/5/2020	Double

Σας άρεσε η διαμονή σας;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque et faucibus lectus. Praesent vel porttitor massa. volutpat semper felis. Vestibulum sit amet dicitum purus, aliquet efficitur est.

Κάντε μια Κριτική

Τίτλος


Κείμενο

[Σχολιάστε](#)

Figure 7: Η σελίδα "Λογαριασμός" μετά την είσοδο του χρήστη

Τέλος, εάν για οποιονδήποτε λόγο το σύστημα δε βρει κάποια σελίδα ή ο χρήστης προσπαθεί να βρει μια σελίδα που δεν υπάρχει στην πλατφόρμα θα ανακατευθυνθεί σε σελίδα σφάλματος που θα τον ενημερώνει σχετικά.

ProjectB Αρχική Δωμάτια Κριτικές Επικοινωνία Είσοδος Εγγραφή EN



Συγγνώμη, αλλά δεν μπορούσαμε να βρούμε τη σελίδα που ψάχνατε.

[Πίσω στην αρχική σελίδα.](#)

Figure 8: Μήνυμα αποτυχίας "404"

5.2 Κάνοντας κράτηση μέσω της πλατφόρμας

Ο χρήστης έχει τη δυνατότητα να κάνει κράτηση δωματίου μέσω της πλατφόρμας, είτε ως εγγεγραμμένος, είτε ως μη εγγεγραμμένος χρήστης. Συγκεκριμένα, ο χρήστης από την αρχική οθόνη στη φόρμα ελέγχου διαθεσιμότητας δωματίων συμπληρώνει τα πεδία check-in, check-out και διαλέγει τύπο δωματίου. Παρακάτω ακολουθεί εικόνα με τον χρήστη να επιλέγει για ημερομηνία άφιξης τις 28/5/2020, ημερομηνία αναχώρησης 31/5/2020 για το δωμάτιο τύπου Double.

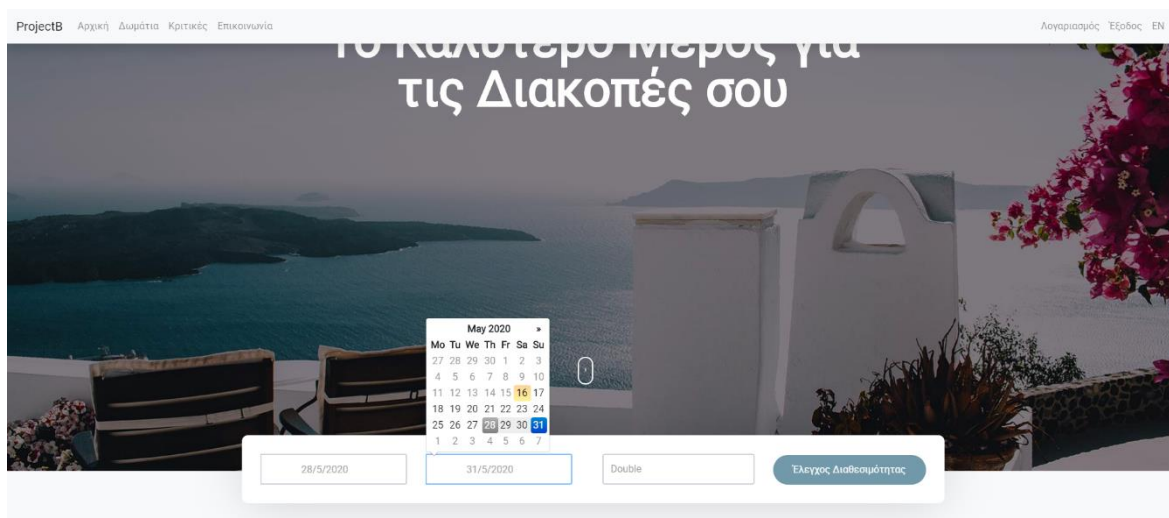


Figure 9: Έλεγχος διαθεσιμότητας δωματίου

Σε περίπτωση που το δωμάτιο δεν είναι διαθέσιμο τις ημερομηνίες που έχουμε επιλέξει, το σύστημα θα μας αποστείλει σχετικό μήνυμα και κάτω από τη φόρμα ελέγχου διαθεσιμότητας θα μας προτείνει διαφορετικές ημερομηνίες που είναι διαθέσιμες για το δωμάτιο.

Οι ημερομηνίες που έχετε επιλέξει δεν είναι διαθέσιμες.

Figure 10: Μήνυμα ενημέρωσης για μη διαθέσιμες ημερομηνίες

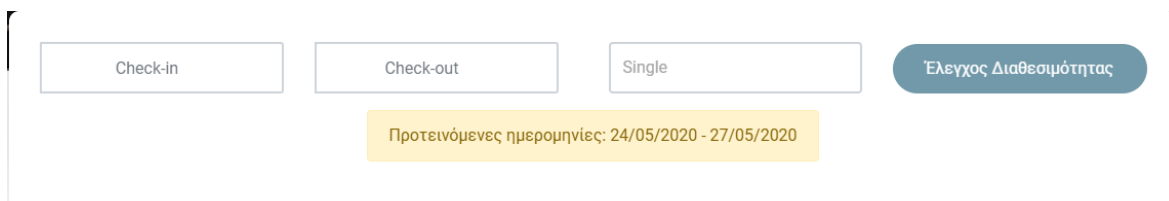


Figure 11: Προτεινόμενες ημερομηνίες

Εάν το δωμάτιο Double είναι διαθέσιμο στις συγκεκριμένες ημερομηνίες το σύστημα θα μας ανακατευθύνει σε σελίδα συμπλήρωσης των στοιχείων μας που είναι απαραίτητα για την κράτηση. Επιπλέον, εάν έχουμε δημιουργήσει λογαριασμό στην πλατφόρμα και είμαστε συνδεδεμένοι, το σύστημα αυτόματα θα έχει συμπληρώσει τη φόρμα των στοιχείων μας. Εάν δεν έχουμε δημιουργήσει λογαριασμό ή δεν είμαστε συνδεδεμένοι θα πρέπει να συμπληρώσουμε τα στοιχεία μας.

Οι ημερομηνίες που έχετε επιλέξει είναι διαθέσιμες. Παρακαλώ συμπληρώστε τα στοιχεία σας.

Κράτηση ως Χρήστης

Όνομα
Κώστας

Επίθετο
Παπαδοπουλος

Email
kr@mail.com

Αριθμός Τηλεφώνου
6969696969

Διεύθυνση
Καρου 20

Δωμάτιο

Figure 12: Η σελίδα συμπλήρωσης στοιχείων για την κράτηση με συνδεδεμένο χρήστη

Οι ημερομηνίες που έχετε επιλέξει είναι διαθέσιμες. Παρακαλώ συμπληρώστε τα στοιχεία σας.

Κράτηση ως Επισκέπτης

Όνομα

Επίθετο

Email

Αριθμός Τηλεφώνου

Διεύθυνση

Δωμάτιο

Εγγραφείτε Σήμερα

Συνδεθείτε για κράτηση με τα αποθηκευμένα στοιχεία σας ή εγγραφείτε για να διαχειρίζεστε τις κρατήσεις σας από παντού!

Figure 13: Η σελίδα συμπλήρωσης στοιχείων για την κράτηση με μη συνδεδεμένο χρήστη

Συνολικά, ο χρήστης θα πρέπει να συμπληρώσει 6 πεδία για να μπορέσει να ολοκληρώσει την κράτηση. Τα πεδία είναι τα εξής: Όνομα, Επίθετο, Email, Αριθμός Τηλεφώνου, Διεύθυνση και τα στοιχεία πιστωτικής ή χρεωστικής κάρτας. Εάν ο χρήστης δεν συμπληρώσει κάποιο από αυτά τα στοιχεία ή τα συμπληρώσει λανθασμένα, το σύστημα θα αποστείλει ανάλογο μήνυμα λάθους.

Πιστωτική ή χρεωστική κάρτα

1234 5678 9123 4567| MM / EE CVC

Ο αριθμός της κάρτας σας δεν είναι έγκυρος.

Κάντε Κράτηση

Figure 14: Μήνυμα αποτυχίας "Ο αριθμός της κάρτας σας δεν είναι έγκυρος"

Email

iliaskosmas.com

Λανθασμένο email.

Figure 15: Μήνυμα αποτυχίας "Λανθασμένο email"

Εάν ο χρήστης έχει συμπληρώσει όλα τα στοιχεία σωστά με το πάτημα του κουμπιού «Κράτηση Τώρα» ολοκληρώνει την κράτηση στο συγκεκριμένο δωμάτιο. Το σύστημα εμφανίζει μήνυμα επιτυχίας.



Figure 16: Μήνυμα επιτυχημένης κράτησης

6. Οδηγός λογισμικού για προγραμματιστή

Σε αυτό το κεφάλαιο περιγράφουμε το μοντέλο ανάπτυξης της πλατφόρμας από τη μεριά του προγραμματιστή. Το σχεδιαστικό πρότυπο που ακολουθεί η εφαρμογή είναι το MTV (Model – Template - View) που στην ουσία ορίζει τα μοντέλα ως κλάσεις που αλληλοεπιδρούν μέσω της SQLAlchemy με τη βάση δεδομένων, τις προβολές (views) που είναι κλάσεις ή συναρτήσεις που προσδιορίζουν την λογική και τα templates που εμφανίζουν το περιεχόμενο στον χρήστη. Για την καλύτερη κατανόηση της δομής της εφαρμογής το κεφάλαιο παρέχει αρκετά αποσπάσματα του κώδικα που αναπτύχθηκε.

6.1 Εκτέλεση της εφαρμογής

Όπως αναφέραμε στο Κεφάλαιο 2 η εφαρμογή αναπτύχθηκε στο λειτουργικό σύστημα Ubuntu 18.04 LTS (Bionic Beaver). Για να τρέξουμε την εφαρμογή χρειαζόμαστε το Docker (<https://www.docker.com/>). Αφού εγκαταστήσουμε το Docker, ανοίγουμε ένα τερματικό παράθυρο (terminal) ανάλογα με το λειτουργικό που χρησιμοποιούμε και πηγαίνουμε στον κατάλογο φάκελο που βρίσκεται η εφαρμογή μας. Εκτελούμε την εντολή `docker-compose up --build`.

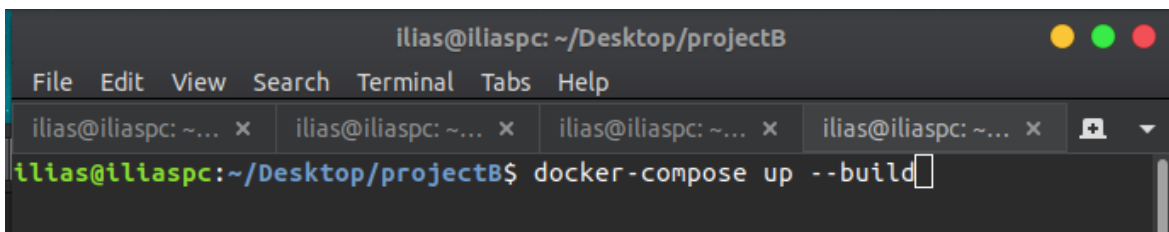


Figure 17: Εκτέλεση της εντολής `docker-compose up --build` σε ΛΣ Ubuntu στον κατάλογο που βρίσκεται η εφαρμογή

Τρέχοντας την συγκεκριμένη εντολή, το Docker θα κατεβάσει και θα εγκαταστήσει όλες τις απαραίτητες εξαρτήσεις (dependencies) που χρειάζεται η εφαρμογή για να λειτουργήσει. Μπορούμε να δούμε τι ακριβώς θα εγκατασταθεί ανατρέχοντας στο αρχείο `requirements.txt` που βρίσκεται στη διαδρομή `/projectB/app/`. Επίσης, το Docker θα δημιουργήσει τον τοπικό server, το `phpMyAdmin`, και θα αρχικοποιήσει τη βάση δεδομένων διαβάζοντας το αρχείο `init.sql` που βρίσκεται στη διαδρομή `/projectB/db/`. Το Docker χρησιμοποιεί το αρχείο `docker-compose.yml` που βρίσκεται στον κατάλογο `projectB`.

Εάν θέλουμε να σταματήσουμε να τρέχουμε ή αλλιώς να σταματήσουμε τον server χρησιμοποιούμε τη συντόμευση `Ctrl + C` στο τερματικό. Όταν θέλουμε να κλείσουμε την πλατφόρμα χρησιμοποιούμε την εντολή `docker-compose down`.

```
^CGracefully stopping... (press Ctrl+C again to force)
Stopping projectb_app_1 ... done
Stopping projectb_phpmyadmin_1 ... done
Stopping projectb_db_1 ... done
ilias@iliaspc:~/Desktop/projectB$ docker-compose down
Removing projectb_app_1 ... done
Removing projectb_phpmyadmin_1 ... done
Removing projectb_db_1 ... done
Removing network projectb_accomodation
ilias@iliaspc:~/Desktop/projectB$
```

Figure 18: Εκτέλεση της εντολής docker-compose down

6.1 Δομή των αρχείων της πλατφόρμας

Όλα τα αρχεία της εφαρμογής βρίσκονται στον φάκελο projectB. Ανοίγοντάς τον έχουμε τους φακέλους:

- app
- db

και τα αρχεία:

- docker-compose.yml: ρυθμίζει τα docker-images που θα δημιουργήσει το docker
- babel.cfg και messages.pot: χρησιμοποιούνται από την επέκταση flask-babel για την αλλαγή γλώσσας
- README.md και .gitignore: χρησιμοποιούνται για την πλατφόρμα github δεν είναι απαραίτητα για την εφαρμογή

Ανοίγοντας τον φάκελο app έχουμε τους εξής φακέλους:

- Models: περιέχει όλες τις κλάσεις – μοντέλα τις βάσεις δεδομένων
- Static: περιέχει όλα τα στατικά στοιχεία της εφαρμογής όπως αρχεία css και javascript καθώς και εικόνες
- Templates: περιέχει όλα τα αρχεία html που χρησιμοποιεί η εφαρμογή
- Translations: περιέχει τα αρχεία μετάφρασης

Επίσης τα αρχεία:

- app.py: περιέχει τα views της εφαρμογής – στην ουσία όλη τη λογική της πλατφόρμας
- database.py: για χρήση της SQLAlchemy
- dockerfile: αρχείο ρυθμίσεων για το Docker
- forms.py: περιέχει όλες τις φόρμες που χρησιμοποιεί η εφαρμογή

- functions.py: περιέχει συναρτήσεις που χρησιμοποιούνται πολλές φορές από την εφαρμογή
- requirements.txt: περιέχει όλες τις βιβλιοθήκες και επεκτάσεις που χρησιμοποιεί η εφαρμογή

Τέλος, ο φάκελος db περιέχει το αρχείο init.sql για την αρχικοποίηση του πίνακα.

6.2 Δομή της βάσης δεδομένων (Model)

Η βάση δεδομένων είναι γραμμένη σε MySQL. Η δημιουργία της έγινε με το εργαλείο phpMyAdmin. Μπορούμε να έχουμε πρόσβαση πηγαίνοντας στον παρακάτω σύνδεσμο <http://localhost:8082/>. Για να συνδεθούμε στη βάση βάζουμε τα εξής στοιχεία:

- στο πεδίο «Διακομιστής»: db
- στο πεδίο «Όνομα χρήστη»: root
- στο πεδίο «Κωδικός πρόσβασης»: root

Το όνομα της βάσης είναι «hoteldb» και περιέχει τους εξής πίνακες:

1. guest_reservations: περιέχει τις εγγραφές για τις κρατήσεις από μη εγγεγραμμένους χρήστες στην πλατφόρμα
2. hotel
3. reservations: περιέχει όλες τις εγγραφές για τις κρατήσεις από εγγεγραμμένους χρήστες στην πλατφόρμα
4. reviews: περιέχει όλες τις κριτικές – σχόλια χρηστών
5. rooms: περιέχει όλα τα δωμάτια
6. users: περιέχει όλους τους χρήστες και τον διαχειριστή
7. vacancies: περιέχει τα διαστήματα ημερομηνιών διαθεσιμότητας για όλα τα δωμάτια

Το αρχείο init.sql αρχικοποιεί τη βάση δημιουργώντας τους παραπάνω πίνακες και εισάγει στον πίνακα Users: user1@mail.com, user2@mail.com και admin@mail.com, στον πίνακα Rooms τα δωμάτια: Single, Double, Deluxe, Family και τις αντίστοιχες καταχωρήσεις στον πίνακα Vacancies.

Σε αυτή την ενότητα πρέπει να αναφέρουμε τις αντίστοιχες κλάσεις - μοντέλα που έπρεπε να δημιουργήσουμε για την επικοινωνία με τη βάση δεδομένων. Όλα τα μοντέλα βρίσκονται στην διαδρομή /projectB/app/models και έχουν το αντίστοιχο όνομα με τον πίνακα στη βάση δεδομένων. Έτσι για παράδειγμα έχει δημιουργηθεί το αρχείο rooms.py που αντιστοιχίζεται με τον πίνακα rooms της βάσης.

```

8
9 class Rooms(Base):
10
11     __table_args__ = {'extend_existing': True}
12     __tablename__ = 'rooms'
13     id = db.Column(db.Integer, primary_key=True, unique=True)
14     number = db.Column(db.Integer, index=True, unique=True)
15     price = db.Column(db.Float)
16     type = db.Column(db.String(255))
17     image_file = db.Column(db.String(20), nullable=False)
18     reservations = db.relationship('Reservations', backref='room', lazy=True)
19     guest_reservations = db.relationship('GuestReservations', backref='room', lazy=True)
20     vacancies = db.relationship('Vacancies', backref='room', lazy=True)
21
22     def __init__(self, number, price, type, image_file):
23         self.number = number
24         self.price = price
25         self.type = type
26         self.image_file = image_file
27
28
29 class RoomsSchema(Schema):
30     class Meta:
31         fields = ('number', 'price', 'type', 'image_file')
32

```

Figure 19: Η κλάση Rooms

Αν μελλοντικά χρειαστεί να δημιουργήσουμε επιπλέον πίνακες στη βάση πρέπει να δημιουργήσουμε και το αντίστοιχο μοντέλο για να μπορούμε να αλληλεπιδράσουμε με τη βάση δεδομένων.

6.3 Οι φόρμες που χρησιμοποιεί η πλατφόρμα

Η πλατφόρμα χρησιμοποιεί σε διάφορα σημεία φόρμες συμπλήρωσης στοιχείων όπως για παράδειγμα όταν ένας χρήστης θέλει να κάνει μία κράτηση ή όταν ένας χρήστης θέλει να δημιουργήσει έναν καινούριο λογαριασμό στην εφαρμογή. Επίσης, σε όλες τις φόρμες χρησιμοποιούνται επικυρωτές (validators). Επειδή το να ορίσουμε εμείς τους τύπους πεδίων (field types) και τους επικυρωτές θα ήταν χρονοβόρο και επειδή έχουμε ήδη ορίσει τα μοντέλα μας, χρησιμοποιούμε τα WTForms-Alchemy και το flask-wtforms που μας επιτρέπει να δημιουργήσουμε κλάσεις Form με βάση τα μοντέλα που έχουμε δημιουργήσει. Οι φόρμες που έχουμε δημιουργήσει για την εφαρμογή βρίσκονται στο αρχείο forms.py στον κατάλογο /projectB/app και είναι οι παρακάτω:

Όνομα Κλάσης Φόρμας	Χρήση Φόρμας
LoginForm	η φόρμα που χρησιμοποιείται για τη σελίδα «Είσοδος»
RegiserForm	η φόρμα που χρησιμοποιείται για τη σελίδα «Εγγραφή»

GuestForm	η φόρμα που χρησιμοποιείται για την κράτηση
ReviewForm	η φόρμα που χρησιμοποιείται για τη δημιουργία κριτικής
UpdateRoomPhotos	η φόρμα που χρησιμοποιείται για την αλλαγή φωτογραφίας δωματίου από τον διαχειριστή
UpdateRoomPrices	η φόρμα που χρησιμοποιείται για την αλλαγή τιμής δωματίου από τον διαχειριστή
AddRoom	η φόρμα που χρησιμοποιείται για την προσθήκη νέου δωματίου από τον διαχειριστή

6.4 Πρότυπα (Templates)

Το σχεδιαστικό πρότυπο που ακολουθεί η εφαρμογή είναι το MTV. Για την παρουσίαση των δεδομένων στον χρήστη δημιουργήθηκαν πρότυπα (templates), όπως το index.html που παρουσιάζει την αρχική σελίδα της εφαρμογής (Home page). Κάποια από τα πρότυπα που έχουν δημιουργηθεί είναι:

rooms	Παρουσιάζει τα διαθέσιμα δωμάτια στον χρήστη
reviews	Παρουσιάζει τις κριτικές που έχουν δημιουργηθεί
admin_calendar	Παρουσιάζει το ημερολόγιο στον διαχειριστή
change_photos	Παρουσιάζει το μενού για αλλαγή φωτογραφίας στον διαχειριστή
change_prices	Παρουσιάζει το μενού για αλλαγή τιμής ενός δωματίου στον διαχειριστή
get_info	Παρουσιάζει το μενού κράτησης προς τον χρήστη
register	Παρουσιάζει το φόρμα εγγραφής χρήστη

Όλα τα πρότυπα βρίσκονται στον φάκελο templates, στη διαδρομή projectB/templates.

6.5 Οι μέθοδοι και οι προβολές που χρησιμοποιεί η πλατφόρμα (Views)

Για την καλύτερη λειτουργία της εφαρμογής έχουν δημιουργηθεί κάποιες μέθοδοι που βρίσκονται στη διαδρομή /projectB/app και συγκεκριμένα στο αρχείο functions.py. Παρακάτω αναφέρεται το όνομα της μεθόδου και η λειτουργία της:

get_vacancy_by_id: επιστρέφει ένα id εάν υπάρχει διαθέσιμο εύρος ημερομηνιών

recommend_dates: επιστρέφει μία πλειάδα (tuple) με την προτεινόμενη ημερομηνία άφιξης και την προτεινόμενη ημερομηνία αναχώρησης

count_amount_per_night: επιστρέφει την συνολική τιμή χρέωσης ενός δωματίου

convertTuple: η μέθοδος μετατρέπει μία πλειάδα (tuple) σε String

subscribe_user_newsletter: χρησιμοποιείται από την υπηρεσία mailgun

check_if_current_user_has_reservation: επιστρέφει True εάν ο συνδεδεμένος χρήστης έχει κράτηση

change_admin_calendar_color: επιστρέφει το χρώμα που θα πάρει η γραμμή κράτησης ανάλογα με το ποιο δωμάτιο έχει επιλεγεί

Το σχεδιαστικό μοντέλο MTV δε θα μπορούσε να λειτουργήσει χωρίς τις Προβολές (Views) που φέρουν τη λογική της εφαρμογής. Στην εφαρμογή βρίσκονται στο αρχείο app.py. Υπάρχουν Προβολές που υλοποιούν μια απλή λειτουργία όπως να προβάλουν κάποιο Πρότυπο, για παράδειγμα η Προβολή home. Από την άλλη υπάρχουν Προβολές, όπως η get_info η οποία είναι υπεύθυνη για την πραγματοποίηση μιας κράτησης. Παρακάτω αναλύονται κάποιες από τις βασικές Προβολές της εφαρμογής.

6.5 Σύντομη παρουσίαση της λογικής της πλατφόρμας

Αναφέραμε παραπάνω ότι όλη η λογική της πλατφόρμας βρίσκεται στο αρχείο app.py. Σε αυτή την ενότητα θα αναφερθούμε σε κομμάτια κώδικα που υλοποιούν κάποιες από τις βασικές λειτουργίες της εφαρμογής.

6.5.1 Βασικά χαρακτηριστικά

Κοιτώντας το αρχείο app.py παρατηρούμε στην αρχή ότι γίνεται εισαγωγή κλάσεων και μεθόδων καθώς και αρχικοποιήσεις που χρειάζονται διάφορες από τις επεκτάσεις που χρησιμοποιούμε στην εφαρμογή όπως η SQLAlchemy, το flask-login, το flask-bcrypt κ.α.

```
from functions import get_vacancy_id, recommend_dates, count_amount_per_night, convertTuple, subscribe_user_newsletter, check_if_current_user_has_reservation, change_admin_calendar_color
from models.users import Users
```

Figure 20: Εισαγωγή από το αρχείο functions.py και users.py

```
43 login_manager = LoginManager(app)
44 login_manager.login_view = "user_login"
45 login_manager.login_message = lazy_gettext("Please login to access this page.")
46 login_manager.login_message_category = "info"
47 login_manager.init_app(app)
48
```

Figure 21: Αρχικοποίηση του flask-login και ρύθμιση παραμέτρων

Στη συνέχεια ακολουθούν τα views όπου δίνουν τη λογική στην εφαρμογή μας.

```

93 @app.route("/")
94 @app.route("/home/")
95 def home():
96     rooms = Rooms.query.all()
97     reviews = Reviews.query.all()
98     return render_template('index.html', title="Home", rooms=rooms, reviews=reviews)
99

```

Figure 22: Η προβολή (view) της αρχικής σελίδας, επιστρέφει όλες τις εγγραφές από τους πίνακες room και reviews και στη συνέχεια επιστρέφει τη σελίδα index.html

6.5.2 Έλεγχος διαθεσιμότητας

Για να ελέγξει αν υπάρχει διαθεσιμότητα η εφαρμογή χρησιμοποιεί την προβολή check_availability. Για αρχή, αποθηκεύει σε μεταβλητές την ημερομηνία άφιξης (start_date), την ημερομηνία αναχώρησης (end_date) και τον τύπο του δωματίου. Στη συνέχεια, ελέγχει αν ο χρήστης έχει όντως συμπληρώσει αυτά τα πεδία. Εάν κάποιο από τα πεδία είναι κενό, ο χρήστης ανακατευθύνεται πίσω στην αρχική σελίδα όπου εμφανίζεται σχετικό μήνυμα.

```

116 @app.route("/check_availability/", methods=["POST", "GET"])
117 def check_availability():
118     start_date = request.form["start_date"] if request.form else request.args.get("start_date")
119     end_date = request.form["end_date"] if request.form else request.args.get("end_date")
120     selected_room = request.form["room"] if request.form else request.args.get("selected_room")
121     if start_date == "" or end_date == "":
122         flash(lazy_gettext("Dates cannot be null"), "info")
123         return redirect(url_for("home"))
124     else:

```

Figure 23: Λειτουργία της check_availability (1)

Εάν ο χρήστης έχει συμπληρώσει τα πεδία, τότε οι ημερομηνίες από String μετατρέπονται σε datetime για να μπορέσουμε να κάνουμε πράξεις μεταξύ τους. Υπολογίζεται η διαφορά μεταξύ ημερομηνίας αναχώρησης και άφιξης και περνάμε σε μεταβλητή τις διαθέσιμες ημερομηνίες από τον πίνακα Vacancies. Η μέθοδος get_vacancy_id αναλαμβάνει να βρει εάν υπάρχει καταχώρηση στον πίνακα Vacancies και μας επιστρέφει το id της καταχώρησης εάν υπάρχει. Στη συνέχεια με ένα if ελέγχουμε αν η μέθοδος μας επέστρεψε κάποιο id ή όχι. Αν μας επέστρεψε ένα id ανακατευθύνουμε τον χρήστη στην επόμενη σελίδα για να συμπληρώσει τα στοιχεία του. Εάν δεν μας επέστρεψε κάποιο id, σημαίνει ότι δεν υπάρχει κάποια διαθέσιμη ημερομηνία, οπότε ανακατευθύνουμε τον χρήστη πίσω στην αρχική σελίδα και με τη χρήση της μεθόδου recommend_dates επιστρέφουμε τις προτεινόμενες ημερομηνίες.

```

125 # Converts input dates from string to datetime
126 start = datetime.strptime(start_date, '%d/%m/%Y')
127 end = datetime.strptime(end_date, '%d/%m/%Y')
128 days_difference = (end - start).days
129 vacancies = Vacancies.query.all()
130 vacancy_id = get_vacancy_id(vacancies, start, end, selected_room)
131
132 if vacancy_id is not None:
133     flash(lazy_gettext("The dates you selected are available. Please fill in your information."), "success")
134     return redirect(url_for("get_info", start_date=start_date, end_date=end_date, selected_room=selected_room))
135 else:
136     flash(lazy_gettext("The dates you selected are not available."), "info")
137     recommended_from_date = convertTuple(recommend_dates(vacancies, days_difference, selected_room)[0])
138     recommended_to_date = convertTuple(recommend_dates(vacancies, days_difference, selected_room)[1])
139     return redirect(url_for("home", recommended_from_date=recommended_from_date, recommended_to_date=recommended_to_date))
140

```

Figure 24: Λειτουργία της check_availability (2)

6.5.3 Πραγματοποίηση Κράτησης

Η προβολή που διαχειρίζεται το πώς θα γίνει η κράτηση είναι η get_info. Το πρώτο πράγμα που κάνει η συγκεκριμένη προβολή, είναι να αποθηκεύσει σε μεταβλητές τις ημερομηνίες άφιξης, αναχώρησης και το επιλεγμένο δωμάτιο που δέχεται από την προβολή check_availability. Στη συνέχεια, με τη μέθοδο count_amount_per_night υπολογίζει το συνολικό ποσό που θα χρεώσει και αρχικοποιεί τη φόρμα guestform ανάλογα με το αν υπάρχει ή όχι συνδεδεμένος χρήστης.

```

151 @app.route("/get_info/", methods=["POST", "GET"])
152 def get_info():
153     start_date = request.form["start_date"] if request.form else request.args.get("start_date")
154     end_date = request.form["end_date"] if request.form else request.args.get("end_date")
155     selected_room = request.form["room"] if request.form else request.args.get("selected_room")
156     if start_date is None or end_date is None:
157         flash(lazy_gettext("Dates cannot be null"), "danger")
158         return redirect(url_for("home"))
159     else:
160         amount = count_amount_per_night(start_date, end_date, selected_room)
161         guestform = GuestForm(start_date=start_date, end_date=end_date, room=selected_room, pe
162         loginform = LoginForm()
163         if current_user.is_authenticated:
164             guestform = GuestForm(first_name=current_user.first_name, last_name=current_user.l

```

Figure 25: Η αρχή της get_info

Έπειτα, ελέγχει κατά πόσο στάλθηκε το αίτημα POST από τη φόρμα. Εάν εντοπίσει αίτημα POST και έχει συμπληρωθεί σωστά η φόρμα χρησιμοποιείται η μέθοδος get_vacancy_id που επιστρέφει το διαθέσιμο id εάν υπάρχει. Στη συνέχεια, ελέγχονται 4 διαφορετικές περιπτώσεις που μπορεί να έχουμε και οι οποίες είναι οι εξής:

1. Ο χρήστης να έχει επιλέξει όλο το διαθέσιμο εύρος ημερομηνιών που υπάρχει, οπότε αφαιρείται η κατοχύρωση από τον πίνακα Vacancies.
2. Ο χρήστης να έχει εμπλέξει ίδια ημερομηνία άφιξης και διαφορετική ημερομηνία αναχώρησης, οπότε η ημερομηνία άφιξης στον πίνακα

Vacancies γίνεται η ημερομηνία αναχώρησης που έχει επιλέξει ο χρήστης μεγαλύτερη κατά μία ημέρα.

3. Ο χρήστης να έχει επιλέξει ίδια ημερομηνία αναχώρησης και διαφορετική ημερομηνία άφιξης, οπότε η ημερομηνία αναχώρησης στον πίνακα Vacancies γίνεται η ημερομηνία άφιξης μειωμένη κατά μία ημέρα.
4. Ο χρήστης να έχει επιλέξει διαφορετική ημερομηνία άφιξης και διαφορετική ημερομηνία αναχώρησης, οπότε δημιουργείται καινούρια καταχώρηση στον πίνακα Vacancies με ημερομηνία άφιξης την ημερομηνία αναχώρησης μεγαλύτερη κατά μία ημέρα και ημερομηνία αναχώρησης την ημερομηνία αναχώρησης του εκάστοτε vacancy. Επιπλέον, η ημερομηνία αναχώρησης του εκάστοτε vacancy γίνεται η ημερομηνία άφιξης που έχει επιλέξει ο χρήστης αυξημένη κατά μία ημέρα.

```

172 if vacancy_id is not None:
173     vacancy = Vacancies.query.filter_by(id=vacancy_id).one()
174
175     diff_to_date = (datetime.strptime(vacancy.to_date, '%d/%m/%Y') - end).days
176     diff_from_date = (start - datetime.strptime(vacancy.from_date, '%d/%m/%Y')).days
177
178     if datetime.strptime(vacancy.from_date, '%d/%m/%Y') == start and datetime.strptime(vacancy.to_date, '%d/%m/%Y') == end:
179         local_object = db.session.merge(vacancy)
180         db.session.delete(local_object)
181     elif datetime.strptime(vacancy.from_date, '%d/%m/%Y') == start and diff_to_date > 0:
182         vacancy.from_date = (end + timedelta(days=1)).strftime('%d/%m/%Y')
183         local_object = db.session.merge(vacancy)
184     elif datetime.strptime(vacancy.to_date, '%d/%m/%Y') == end and diff_from_date > 0:
185         vacancy.to_date = (start - timedelta(days=1)).strftime('%d/%m/%Y')
186         local_object = db.session.merge(vacancy)
187     elif diff_from_date > 0 and diff_to_date > 0:
188         temp = vacancy.to_date
189         vacancy.to_date = (start - timedelta(days=1)).strftime('%d/%m/%Y')
190         local_object = db.session.merge(vacancy)
191         new_vac = Vacancies(room_id=selected_room[:1], from_date=(end + timedelta(days=1)).strftime('%d/%m/%Y'), to_date=temp)
192         db.session.add(new_vac)

```

Figure 26: Η συνθήκη if που τροποποιεί τον πίνακα Vacancies

Στη συνέχεια, ανάλογα με το εάν ο χρήστης είναι συνδεδεμένος ή όχι, στην πλατφόρμα δημιουργείται ένα αντικείμενο reservation ή guest_reservation που περιλαμβάνει όλα τα στοιχεία της κράτησης. Επιπλέον, δημιουργούνται τα αντικείμενα customer και charge τα οποία χρησιμοποιούνται από το API της υπηρεσίας Stripe για να ολοκληρωθεί η ηλεκτρονική πληρωμή. Τέλος, όλες οι αλλαγές γίνονται commit στη βάση δεδομένων και ανακατευθύνουμε τον χρήστη σε μήνυμα επιτυχίας της κράτησης.

6.5.4 Η λογική των εργαλείων του Διαχειριστή

Σε αυτή την ενότητα θα αναφερθούμε στον τρόπο λειτουργίας των εργαλείων που έχει στη διάθεσή του ο Διαχειριστής.

Με την καρτέλα «Ημερολόγιο», η εφαρμογή εμφανίζει το ημερολόγιο με τις κρατήσεις και τα παράθυρα για το άνοιγμα και το κλείσιμο ημερομηνιών, χρησιμοποιούνται δύο προβολές η `admin_calendar` και η `cancel_close_date`, με την πρώτη να αναλαμβάνει τον κύριο όγκο εργασίας. Επίσης, χρησιμοποιείται το πρότυπο (template) με την ονομασία `dashboard.html` και το API του FullCalendar (<https://fullcalendar.io/>).

Αρχικά, για την προβολή των γραμμών κράτησης επάνω στο ημερολόγιο, συλλέγουμε όλες τις κρατήσεις από τους πίνακες `reservations` και `guest_reservations` της βάσης δεδομένων και τα εισάγουμε σε πίνακες με συγκεκριμένη μορφή όπως αναφέρεται στον παρακάτω σύνδεσμο από το `documentation` του api του FullCalendar (<https://fullcalendar.io/docs/events-json-feed>). Στη συνέχεια, τα προωθούμε στο template ως αρχεία json όπου τα διαχειρίζεται το `fullcalendar`. Στην παρακάτω εικόνα εμφανίζεται το τμήμα του κώδικα που επιστρέφει το αρχείο json με τις κρατήσεις από τον πίνακα `guest_reservations`.

```
343     x = []
344     for guest_reservation in guest_reservations:
345         start = datetime.strptime(guest_reservation.from_date, '%d/%m/%Y').strftime('%Y-%m-%d')
346         end = (datetime.strptime(guest_reservation.to_date, '%d/%m/%Y') + timedelta(days=1)).strftime('%Y-%m-%d')
347         room = db.session.query(Rooms).filter_by(id=guest_reservation.room_id).first().type
348         x.append({
349             "title": guest_reservation.first_name + " " + guest_reservation.last_name + " | " + room,
350             "start": start,
351             "end" : end,
352             "color": change_admin_calendar_color(guest_reservation.room_id),
353         })
354
355     y = json.dumps(x)
356     resp = json.loads(y)
```

Figure 27: Επιστροφή αρχείου json με τις κρατήσεις του πίνακα `guest_reservation`

Με ανάλογο τρόπο δουλεύει και το κομμάτι κώδικα για τις κρατήσεις του πίνακα `reservations`, με τη διαφορά ότι διαχειρίζεται και τις γραμμές κράτησης με τις ημερομηνίες που κλείνει ο διαχειριστής (εμφανίζονται με κόκκινο χρώμα στο ημερολόγιο).

```

359     res = []
360     for reservation in reservations:
361         start = datetime.strptime(reservation.from_date, '%d/%m/%Y').strftime('%Y-%m-%d')
362         end = (datetime.strptime(reservation.to_date, '%d/%m/%Y') + timedelta(days=1)).strftime('%Y-%m-%d')
363         first_name = db.session.query(Users).filter_by(id=reservation.user_id).first().first_name
364         last_name = db.session.query(Users).filter_by(id=reservation.user_id).first().last_name
365         admin_id = db.session.query(Users).filter_by(id=reservation.user_id).first().id
366         room = db.session.query(Rooms).filter_by(id=reservation.room_id).first().type
367         if current_user.id == admin_id:
368             res.append({
369                 "title": "Closed" + " | " + room,
370                 "start": start,
371                 "end": end,
372                 "color": "red"
373             })
374         else:
375             res.append({
376                 "title": first_name + " " + last_name + " | " + room,
377                 "start": start,
378                 "end": end,
379                 "color": change_admin_calendar_color(reservation.room_id)
380             })
381
382     z = json.dumps(res)
383     respp = json.loads(z)

```

Figure 28: Επιστροφή αρχείου json με τις κρατήσεις του πίνακα reservations

Στη συνέχεια της προβολής admin_calendar ακολουθεί το κομμάτι κώδικα που είναι υπεύθυνο για το κλείσιμο ημερομηνιών σε συγκεκριμένο δωμάτιο. Διαχειριζόμαστε αυτή τη λειτουργία σαν να κάνουμε μία καινούργια κράτηση. Η λογική για το κλείσιμο των ημερομηνιών, καθώς και ο κώδικας, είναι ίδιος με αυτή που χρησιμοποιήθηκε στην προβολή get_info. Έχουμε τις ίδιες περιπτώσεις για τις ημερομηνίες που μπορεί να επιλέξει ο χρήστης – διαχειριστής άρα διαχειριζόμαστε με τον ίδιο τρόπο το πώς θα διαμορφωθούν οι καταχωρήσεις στον πίνακα Vacancies. Τέλος δημιουργούμε το αντικείμενο reservation το οποίο εκτός του επιλεγμένου δωματίου, της ημερομηνία άφιξης και την ημερομηνία αναχώρησης έχει ως id χρήστη το id του διαχειριστή για να το αναγνωρίζουμε και να το διαχειριζόμαστε αναλόγως.

Για το άνοιγμα των ημερομηνιών χρησιμοποιείται η προβολή cancel_close_date, η οποία αναλαμβάνει να ακυρώσει στην ουσία μία κλειστή ημερομηνία από τον διαχειριστή. Οι περιπτώσεις για τις καταχωρήσεις του πίνακα Vacancies παραμένουν οι ίδιες, όπως τις αναφέραμε στην περιγραφή της προβολής get_info, με τη διαφορά της διαχείρισης και των συνθηκών που ελέγχουμε για να προσαρμόσουμε τις καταχωρήσεις του πίνακα Vacancies με τον σωστό τρόπο. Επίσης, στο τέλος διαγράφουμε την καταχώρηση της κράτησης για την κλειστή ημερομηνία. Ακολουθεί το τμήμα του κώδικα που διαχειρίζεται τις περιπτώσεις για τις διαθέσιμες ημερομηνίες.

```

596 reservation = Reservations.query.get_or_404(res_id)
597 start_date = datetime.strptime(reservation.from_date, '%d/%m/%Y')
598 end_date = datetime.strptime(reservation.to_date, '%d/%m/%Y')
599 vacancies = Vacancies.query.filter_by(room_id=reservation.room_id)
600 ids = {"from_date": None, "to_date": None}
601
602 for vacancy in vacancies:
603     if start_date == (datetime.strptime(vacancy.to_date, '%d/%m/%Y') + timedelta(days=1)):
604         ids["to_date"] = vacancy.id
605     elif end_date == (datetime.strptime(vacancy.from_date, '%d/%m/%Y') - timedelta(days=1)):
606         ids["from_date"] = vacancy.id
607
608 if ids["from_date"] == None and ids["to_date"] == None:
609     new_vac = Vacancies(room_id=reservation.room_id, from_date=start_date, to_date=end_date)
610     db.session.add(new_vac)
611 elif ids["from_date"] != None and ids["to_date"] == None:
612     vacancy = Vacancies.query.filter_by(id=ids["from_date"]).one()
613     vacancy.from_date = start_date
614     local_object = db.session.merge(vacancy)
615 elif ids["to_date"] != None and ids["from_date"] == None:
616     vacancy = Vacancies.query.filter_by(id=ids["to_date"]).one()
617     vacancy.to_date = end_date
618     local_object = db.session.merge(vacancy)
619 else:
620     vacancy = Vacancies.query.filter_by(id=ids["from_date"]).one()
621     temp_to_date = vacancy.to_date
622     local_object = db.session.merge(vacancy)
623     db.session.delete(local_object)
624     vacancy = Vacancies.query.filter_by(id=ids["to_date"]).one()
625     temp_from_date = vacancy.from_date
626     local_object1 = db.session.merge(vacancy)
627     db.session.delete(local_object1)
628     new_vac = Vacancies(room_id=reservation.room_id, from_date=temp_from_date, to_date=temp_to_date)
629     db.session.add(new_vac)
630
631 local_object = db.session.merge(reservation)
632 db.session.delete(local_object)
633 db.session.commit()
634 flash(lazy_gettext("Dates opened!"), "success")
635 return redirect(url_for("dashboard"))

```

Figure 29: Η λογική πίσω από τη λειτουργία για το άνοιγμα ημερομηνιών

Μία άλλη λειτουργία που έχει στη διάθεση του ο Διαχειριστή είναι αυτή της αλλαγής φωτογραφίας. Η προβολή που διαχειρίζεται αυτή τη λειτουργία είναι η `change_photos`. Το πρώτο βήμα της προβολής, αφού αποθηκεύσουμε σε μεταβλητή το `id` του επιλεγμένου δωματίου, είναι να διαγράψουμε το αρχείο φωτογραφίας που έχει το συγκεκριμένο δωμάτιο. Στη συνέχεια, διαμορφώνουμε ένα τυχαίο καινούριο όνομα για το αρχείο φωτογραφίας που πρόκειται να ανεβάσουμε. Αυτό το κάνουμε σε περίπτωση που υπάρχει ήδη αρχείο φωτογραφίας με το ίδιο όνομα. Στη συνέχεια αλλάζουμε τις διαστάσεις τις φωτογραφίας. Τέλος, ανανεώνουμε τη στήλη `image_file` της καταχώρησης του δωματίου με το όνομα του αρχείου φωτογραφίας στη βάση δεδομένων και αναρτούμε τη φωτογραφία.

```

437     form = UpdateRoomPhotos()
438     rooms = Rooms.query.all()
439     if form.validate_on_submit():
440         if form.picture.data:
441             # print(request.form, flush=True)
442             # getting selected room by id
443             selected_room = request.form.get('selected_room')
444             room = Rooms.query.filter_by(id=selected_room).first()
445             # delete previous img from file system
446             os.remove(os.path.join('static/images', room.image_file))
447             # random_hex: random name for new img file
448             random_hex = secrets.token_hex(8)
449             # getting img ext
450             _, f_ext = os.path.splitext(form.picture.data.filename)
451             # creating new picture file name
452             picture_fn = random_hex + f_ext
453             # creating picture path
454             picture_path = os.path.join('static/images', picture_fn)
455             # resizing img
456             output_size = (800, 571)
457             i = Image.open(form.picture.data)
458             i = i.resize(output_size, Image.ANTIALIAS)
459             # saving picture to file system
460             i.save(picture_path)
461
462             # putting picture_file to room image_file at db
463             room.image_file = picture_fn
464             db.session.merge(room)
465             db.session.commit()
466             flash(lazy_gettext("The photo has been changed!"), "success")
467             return redirect(url_for("change_photos"))

```

Figure 30: Το τμήμα του κώδικα που είναι υπεύθυνο για την αλλαγή φωτογραφίας

Επιπλέον της αλλαγής φωτογραφίας, ο διαχειριστής έχει και το εργαλείο της αλλαγής τιμής δωματίου, όπου υπεύθυνη για τη λειτουργία αυτή είναι η προβολή `change_prices`. Η προβολή χρησιμοποιεί την φόρμα `UpdateRoomPrices` και επιλέγοντας το δωμάτιο που θέλουμε να κάνουμε την αλλαγή. Στη συνέχεια, για το επιλεγμένο δωμάτιο αλλάζουμε τα δεδομένα της στήλης `price` του πίνακα `rooms` της βάσης δεδομένων.

Συνεχίζοντας με τα εργαλεία που έχει στη διάθεση του ο διαχειριστής, έχουμε την προσθήκη καινούριου δωματίου όπου υπεύθυνη για τη λειτουργία είναι η προβολή `add_room`. Η συγκεκριμένη προβολή, χρησιμοποιεί τη φόρμα `AddRoom` για να προσθέσει ένα νέο δωμάτιο στη βάση. Επίσης, γίνεται χρήση μέρους του κώδικα που χρησιμοποιούμε στην προβολή `change_photo` για τη ανάρτηση της φωτογραφίας του νέου δωματίου. Εκτός του καινούριου δωματίου που προστίθεται στη βάση, προσθέτουμε και μία καινούρια εγγραφή στον πίνακα `Vacancies` όπου είναι το αρχικό εύρος ημερομηνιών που είναι διαθέσιμες για το συγκεκριμένο δωμάτιο.

Τέλος, για τη λειτουργία της καρτέλας «Διαγραφή Δωματίου» είναι υπεύθυνη η προβολή `remove_room`. Αφού πάρουμε σε μεταβλητή την `id` του δωματίου που έχει περάσει ο διαχειριστής, ελέγχουμε αν το συγκεκριμένο δωμάτιο έχει κρατήσεις στους πίνακες `reservations` ή `guest_reservations`. Αν υπάρχει κράτηση, εμφανίζουμε ανάλογο μήνυμα, αλλιώς προχωράμε διαγράφοντας πρώτα τη φωτογραφία του δωματίου, την καταχώρηση του δωματίου από τον πίνακα `room` και τις καταχωρήσεις που έχει το συγκεκριμένο δωμάτιο στον πίνακα `vacancies`.

Συμπεράσματα και μελλοντικές επεκτάσεις

Οι διαδικτυακές πλατφόρμες ηλεκτρονικού εμπορίου στον τουρισμό έχουν αλλάξει συθέμελα το πώς λειτουργεί σήμερα ο κλάδος καθώς άνοιξαν νέες ευκαιρίες στις επιχειρήσεις. Η δυναμική που αποκτά ένα τουριστικό κατάλυμα μέσω μιας διαδικτυακής εφαρμογής είναι πολλαπλάσια των παραδοσιακών μεθόδων. Αυτό το συμπέρασμα καθιστά τέτοιες εφαρμογές ζωτικής σημασίας για μία επιχείρηση, όπως ενός τουριστικού καταλύματος. Οι συγκεκριμένες κατηγορίες επιχειρήσεων πρέπει να εκμεταλλεύονται όλες τις νέες τεχνολογίες που είναι διαθέσιμες ώστε να έχουν το καλύτερο δυνατό αποτέλεσμα όσον αναφορά την αύξηση της ποιότητας των ηλεκτρονικών υπηρεσιών που παρέχουν προς τους χρήστες.

Από την άλλη μεριά, οι προγραμματιστές τέτοιων εφαρμογών πρέπει να παρακολουθούν όλες τις τελευταίες τεχνολογικές εξελίξεις, ώστε να είναι σε θέση να παρέχουν ένα ασφαλές και αποδοτικό περιβάλλον για τον εκάστοτε διαχειριστή μιας τουριστικής πλατφόρμας.

Ως προς την ανάπτυξη μιας διαδικτυακής πλατφόρμας για τουριστικό κατάλυμα, πρέπει να γίνεται καλός σχεδιασμός ως προς το τι δεδομένα θα επεξεργάζεται η εφαρμογή, άρα το πώς θα σχεδιαστεί η βάση δεδομένων, το τι λειτουργίες θέλουμε να έχει η πλατφόρμα και τι τεχνολογίες θα χρησιμοποιήσουμε για να φτάσουμε στο επιθυμητό αποτέλεσμα.

Επεκτάσεις που μπορούν να γίνουν στην παρούσα εφαρμογή θα ήταν όσον αναφορά τα εργαλεία που έχει στη διάθεση του ο διαχειριστής όπως προσθήκη καρτέλας παρακολούθησης των οικονομικών στοιχείων καθώς και καρτέλα παρακολούθησης των κριτικών των χρηστών και διαχείρισής τους. Επίσης, θα μπορούσε να προστεθεί υπηρεσία ανταλλαγής μηνυμάτων μεταξύ διαχειριστή και πελάτη που έχει κάνει κράτηση. Επιπλέον, θα μπορούσαμε να προσθέσουμε δυνατότητα προσθήκης προσφοράς για κάποιο από τα δωμάτια.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «E-commerce». [Online]. Available: <https://en.wikipedia.org/wiki/E-commerce>.
- [2] «Travel technology». [Online]. Available: https://en.wikipedia.org/wiki/Travel_technology.
- [3] «Σχετικά με την Booking.com™». [Online]. Available: <https://www.booking.com/content/about>.
- [4] «Τι είναι η Airbnb και πώς λειτουργεί;». [Online]. Available: <https://www.airbnb.gr/help/article/2503>.
- [5] «Vbro.com». [Online]. Available: <https://www.vrbo.com/el-gr/info/about-the-family/>.
- [6] «Flask Documentation». [Online]. Available: <https://flask.palletsprojects.com/en/1.1.x/>.
- [7] «Flask (web framework)». [Online]. Available: [https://en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework)).
- [8] «Jinja Documentaion». [Online]. Available: <https://jinja.palletsprojects.com/en/2.11.x/>
- [9] «Jinja (template engine)». [Online]. Available: [https://en.wikipedia.org/wiki/Jinja_\(template_engine\)](https://en.wikipedia.org/wiki/Jinja_(template_engine)).
- [10] «Python 3 Documentation». [Online]. Available: <https://docs.python.org/3/>
- [11] «SQLAlchemy Documentation». [Online]. Available: <https://docs.sqlalchemy.org/en/13/>.
- [12] «Docker Documentation». [Online]. Available: <https://docs.docker.com/>.
- [13] «Flask-Login Documentation», [Online]. Available: <https://flask-login.readthedocs.io/en/latest/>.
- [14] «Flask-Babel Documentation». [Online]. Available: <https://pythonhosted.org/Flask-Babel/>.
- [15] «Flask-Bcrypt Documentation». [Online]. Available: <https://flask-bcrypt.readthedocs.io/en/latest/>.
- [16] «WTForms-Alchemy Documentation». [Online]. Available: <https://wtforms-alchemy.readthedocs.io/en/latest/index.html>.
- [17] «Flask-WTF Documentation». [Online]. Available: <https://flask-wtf.readthedocs.io/en/stable/>.
- [18] «Bootstrap Documentation». [Online]. Available: <https://getbootstrap.com/>.

[19] «FullCalendar Documentation». [Online]. Available: <https://fullcalendar.io/docs>.

[20] «Bootstrap-Datepicker Documentation». [Online]. Available: <https://bootstrap-datepicker.readthedocs.io/en/latest/>.

[21] «Stripe Documentation». [Online]. Available: <https://stripe.com/docs> .

[22] «Mailgun Documentation». [Online]. Available: <https://documentation.mailgun.com/en/latest/index.html>.

[23] «Light Bootstrap Dashboard Documentation». [Online]. Available: <https://demos.creative-tim.com/light-bootstrap-dashboard/documentation/tutorial-components.html>.

[24] «HTML-CSS-Javascript». [Online]. Available: <https://documentation.mailgun.com/en/latest/index.html>.