



Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας

WEB ΕΦΑΡΜΟΓΗ ΜΕ ΧΡΗΣΗ PHP

ECE541-Συστήματα Βάσεων Δεδομένων II

Χειμερινό Εξάμηνο 2023-2024
Διδάσκουσα: Τουσίδου Ελένη

Προθεσμία παράδοσης: Δευτέρα, Σεπτέμβριος 16, 2024

Διαμαντή Βασιλεία 03296

Καραϊσκος Νέστορας 03285

Περιεχόμενα

Θέμα Εργασίας	2
Οδηγίες Εγκατάστασης – Χρήση DBMS	2
Διάγραμμα Οντοτήτων-Συσχετίσεων – Προδιαγραφές Βάσης	4
Μορφή Ιστοσελίδας – Οργάνωση Προγράμματος	8
Σύνδεση Κώδικα PHP Με Βάση Δεδομένων	9
Sessions: Δημιουργία Και Χρήση	9
Αρχική Σελίδα - Authentication Page.....	10
User Login – Admin Login.....	10
User Signup	12
Website - User Side.....	13
Main Page – Home	14
Our Rooms	16
Μετάβαση από καρτέλα – Προβολή όλων των δωματίων	17
Χρήση φίλτρων – Προβολή δωματίων με συνθήκες.....	17
Κράτηση Δωματίου – Εισαγωγή στον πίνακα “make_simple_resrv”.....	23
Services	27
Αρχική σελίδα – Προβολή Ενδεικτικών Υπηρεσιών	27
Σελίδα Υποκατηγορίας Υπηρεσιών – Προβολή Υποκατηγορίας	30
Σελίδα υποκατηγορίας - Προβολή υποκλάσης	31
Ραντεβού για υπηρεσία – Εισαγωγή στον πίνακα book_service	32
Special Offers	33
Αρχική Σελίδα – Μπάρα Αναζήτησης Πακέτων.....	34
Επιλογή δωματίων από πακέτο – Διαχείριση τριμερής σχέσης.....	36
Κράτηση Δωματίου με χρήση πακέτου – Εισαγωγή στο “make_special_resrv”	39
Website – Administrator Side	41
Admin – Our Rooms	41
Προβολή Δωματίων	42
Προσθήκη Δωματίου.....	43
Επεξεργασία Δωματίου.....	44
Διαγραφή Δωματίου	45
Admin – Services	46
Προσθήκη Υπηρεσίας.....	47
Επεξεργασία – Διαγραφή Υπηρεσίας.....	48
Admin – Special Offers	48
Προσθήκη Special Offer	48
Προσθήκη Special Offer Σε Δωμάτιο.....	49
Αφαίρεση Συσχέτισης Special Offer Με Δωμάτιο	49
Μελλοντικές Επεκτάσεις.....	50
Πηγές	50
Παραδοτέα Αρχεία	50

Θέμα Εργασίας

Ως εργασία εξαμήνου επιλέχθηκε η δημιουργία μιας web εφαρμογής με χρήση PHP για την προβολή του περιεχομένου μιας ΒΔ σε ιστοσελίδα. Σε αυτό το report αναλύεται η υλοποίηση της ιστοσελίδας ενός ξενοδοχείου με βάση τα εργαλεία της γλώσσας PHP. Περιγράφονται οι διαδικασίες με βάση τις οποίες ο χρήστης μπορεί να αναζητήσει πληροφορίες για τις παροχές, να θέσει τα επιθυμητά φίλτρα του και να αναζητήσει πληκτρολογώντας στη μπάρα αναζήτησης και τέλος να κάνει κράτηση για δωμάτια και υπηρεσίες. Επιπλέον, με τη σύνδεση ως administrator, ο χρήστης μπορεί να προσθέσει, να τροποποίησει και να αφαιρέσει τα περιεχόμενα που εμφανίζονται στους πελάτες, δίνοντας δυναμικό χαρακτήρα στην ιστοσελίδα.

Οδηγίες Εγκατάστασης – Χρήση DBMS

Για μπορούμε να χρησιμοποιήσουμε ένα local web server environment, επιλέξαμε τη χρήση του XAMPP, ένα ευρέως διαδομένο open-source web development environment. Χρησιμοποιεί το MySQL ως DBMS και δέχεται ως server-side scripting language την PHP.

Για την εγκατάσταση περιβάλλοντος πηγαίνουμε στην επίσημη σελίδα της Apache Friends – XAMPP και επιλέγουμε το Download για το λειτουργικό σύστημα της συσκευής μας.

XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.



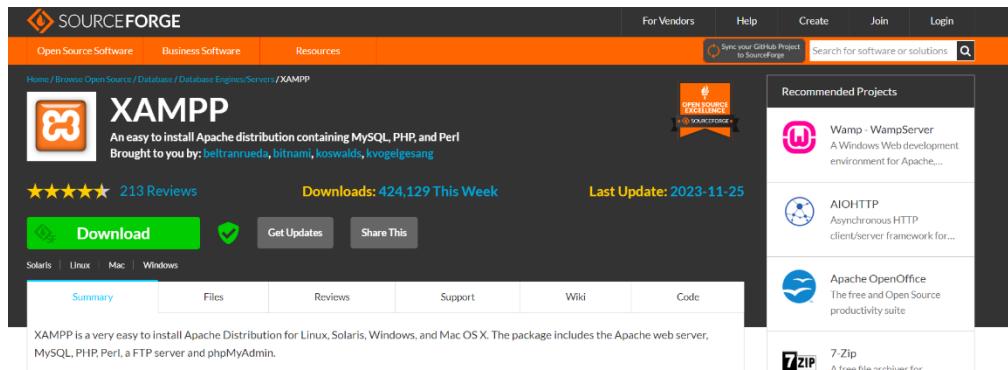
Download
Click here for other versions

 XAMPP for Windows
8.2.12 (PHP 8.2.12)

 XAMPP for Linux
8.2.12 (PHP 8.2.12)

 XAMPP for OS X
8.2.4 (PHP 8.2.4)

Στη συνέχεια, μεταφερόμαστε στην παρακάτω σελίδα, όπου θα πρέπει να αρχίσει αυτόματα το download, διαφορετικά πατάμε το αντίστοιχο κουμπί.

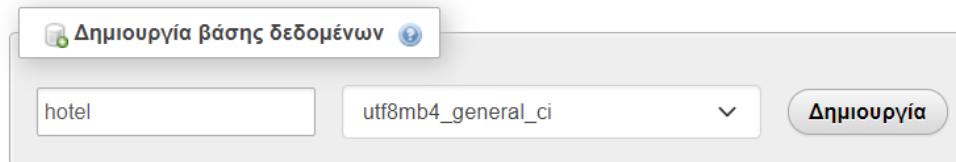


Αφού εγκατασταθεί το .exe αρχείο, το περιβάλλον έχει εγκατασταθεί. Για να το ενεργοποιήσουμε, ανοίγουμε την εφαρμογή και εκκινούμε τις επιλογές “Apache” και “MySQL”.



Μέσα στο installation folder υπάρχει ο φάκελος htdocs (full pathname: “C:\xampp\htdocs”), στο οποίο τοποθετούμε τον φάκελο “MtGateway”. Ανοίγοντας έναν οποιοδήποτε browser, μπορούμε πλέον να γράψουμε “localhost/ MtGateway/” και να μεταφερθούμε στην αρχική σελίδα της ιστοσελίδας μας.

Βάσεις δεδομένων



Τέλος, πρέπει να φορτώσουμε τη βάση δεδομένων μας. Για να το κάνουμε αυτό χρησιμοποιούμε το GUI που μας παρέσει το phpMyAdmin, ένα web-based εργαλείων διαχείρησης MySQL βάσεων δεδομένων. Πατώντας το κουμπί “Admin” στη γραμμή “MySQL” (όπως φαίνεται στην παραπάνω εικόνα), πηγαίνουμε στο αντίστοιχο περιβάλλον και επιλέγουμε το κουμπί για τη δημιουργία νέας βάσης με όνομα “hotel”.

Έχοντας δημιουργήσει μια άδεια βάση επιλέγουμε το κουμπί “Εισαγωγή” και εισάγουμε δεδομένα απευθείας από το αρχείο “hotel_database.sql”, το οποίο υπάρχει μέσα στο zip.

Εισαγωγή στη βάση δεδομένων «hotel»

Αρχείο για εισαγωγή:

Το αρχείο μπορεί να συμπιεστεί (gzip, bzip2) ή να αποσυμπιεστεί.
Το όνομα του συμπιεσμένου αρχείου πρέπει να λήγει σε **[format].[compression]**. Παράδειγμα: **.sql.zip**

Περιηγηθείτε στον υπολογιστή σας. (Μέγιστο μέγεθος: 40MB)

Choose File

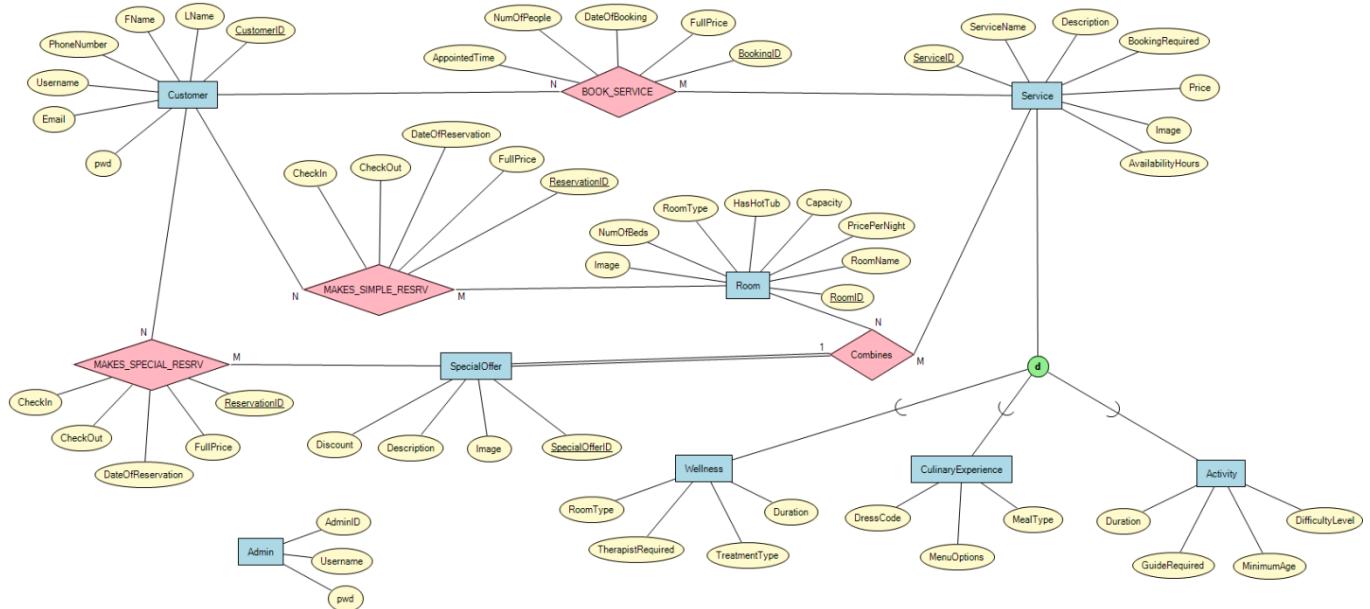
Μπορείτε να σύρετε και να αφήσετε ένα αρχείο σε οποιαδήποτε σελίδα.

Σύνολο χαρακτήρων του αρχείου:

utf-8

Διάγραμμα Οντοτήτων-Συσχετίσεων – Προδιαγραφές Βάσης

Παρατίθεται το Διάγραμμα Οντοτήτων-Συσχετίσεων με βάση το οποίο δημιουργήθηκε η βάση δεδομένων μας. Σχεδιάστηκε με χρήση της εφαρμογής ER2SQL.



Στη συνέχεια, αναλύουμε τις οντότητες, τις σχέσεις, τα αντίστοιχα γνωρίσματα και τους περιορισμούς που έχουμε θέσει.

- ❖ **Customer:** Κάθε χρήστης της ιστοσελίδας πρέπει να κάνει login, δίνοντας username και password για να μεταφερθεί στο main page. Τα στοιχεία του πελάτη έχουν αποθηκευτεί στον πίνακα κατά το sign up και έχουμε θεωρήσει ως προϋπόθεση ότι είναι απαραίτητο ο νέος χρήστης να συμπληρώσει όλα τα πεδία. Εκτός από αυτά που αναφέρθηκαν, τα επιπλέον γνωρίσματα του πίνακα είναι: CustomerID (primary key), Email, PhoneNumber, Fname, Lname.

- ❖ **Room:** Θεωρούμε ότι κάθε δωμάτιο είναι μία μοναδική εγγραφή στο πίνακα και αποτελείται από οκτώ γνωρίσματα: RoomID (primary key), RoomName, PricePerNight, Capacity, HasHotTub, RoomType, NumOfBeds, Image.
- ❖ **Service:** Έχοντας ένα μεγάλο πλήθος και διαφορετικά είδη παροχών, δημιουργήσαμε αυτή την υπερκλάση, η οποία σπάει σε τρεις υποκλάσεις: Activity, Wellness, Culinary_Experience. Τα γνωρίσματα της οντότητα κληρονομούνται από τις υποκατηγορίες της και είναι τα εξής: ServiceID (primary key), ServiceName, Price, BookingRequired, Description AvailabilityHours, Image. Κάθε εγγραφή του θεωρούμε ότι πρέπει να ανήκει σε ακριβώς μία υποκλάση (χρήση του disjoint – δεν μπορεί να υπάρξει overlap). Οι εγγραφές στους διαφορετικούς πίνακες συνδέονται με το κοινό τους primary key ServiceID. Έτσι τα κοινά γνωρίσματα βρίσκουνται όλα συγκεντρωμένα σε αυτόν τον πίνακα.
- ❖ **Activity:** Υποκλάση της οντότητας Service αφορά αθλητικές δραστηριότητες που παρέχει το ξενοδοχείο και έχει τα επιπλέον γνωρίσματα: DifficultyLevel, MinimumAge, GuideRequired, Duration.
- ❖ **Wellness:** Η δεύτερη υποκλάση περιέχει όλες τις υπηρεσίες ευεξίας με τα επιπλέον γνωρίσματα: RoomType TherapistRequired TreatmentType Duration.
- ❖ **Culinary_Experience:** Η τρίτη και τελευταία υποκλάση με τίτλο "γαστρονομική εμπειρία" προβάλει όλες τις δυνατότητες που έχει πελάτης για φαγητό/γλυκό/ποτό. Τα επιπλέον γνωρίσματά του είναι: MealType, MenuOptions, DressCode.
- ❖ **Special_Offer:** Μέσω αυτή της οντότητας, το ξενοδοχείο δημιουργεί πακέτα προσφορών, συνδέοντας μια έκπτωση για την τιμή του δωματίου με την δωρεάν είσοδο σε μία συγκεκριμένη υπηρεσία. Κάθε μία εγγραφή είναι μοναδική και συνδέεται με ακριβώς μία υπηρεσία μέσω ενός foreign key. Ένα Service μπορεί να υπάρχει σε περισσότερα από ένα πακέτα. Τα γνωρίσματα της οντότητας είναι: SpecialOfferID (primary key), Description, Discount, Image, ServiceID (foreign key).
- ❖ **Make_Simple_Resrv:** Ο πελάτης έχει τη δυνατότητα να κάνει κράτηση ένα δωμάτιο απευθείας (χωρίς τη χρήση κάποιου πακέτου προσφορών). Αυτού του είδους κράτηση γίνεται τόσο στο main page (επιλέγοντας τις επιθυμητές ημερομηνίες) όσο και στην ενότητα "Our Rooms". Τα γνωρίσματα πως καθορίζονται από το χρήστη (δηλαδή τα CheckIn και CheckOut) είναι απαραίτητο να συμπληρωθούν από τον χρήστη, διαφορετικά το πρόγραμμα δεν θα του επιτρέψει να ολοκληρώσει την κράτησή του. Έχοντας επιλέξει το επιθυμητό δωμάτιο, θα προκύψουν αυτόματα οι τιμές των γνωρισμάτων FullPrice και FK2_RoomID, που αποτελεί foreign key του πίνακα Customer. Τέλος, το επίσης ξένο κλειδί FK1_CustomerID θα προκύψει αυτόματα λόγω του ότι ο χρήστης είναι συνδεδεμένος με αυτό, έχοντας κάνει login, ενώ τα γνωρίσματα ReservationID (primary key) και DateOfReservation θα πάρουν τις προκαθορισμένες τους τιμές. Η σχέση αυτή είναι N:M μεταξύ των οντοτήτων Customer και Room, επομένως ένας πελάτης μπορεί να κλείσει πολλά δωμάτια (κανοντας διαφορετικές κρατήσεις) και ένα δωμάτιο μπορεί να έχει επιλεχθεί από πολλούς πελάτες. Κατά τη διαδικασία της κράτηση γίνεται έλεγχος ώστε να μην μπορεί να επιλεχθεί δωμάτιο που δεν είναι διαθέσιμο τις επιθυμητές ημερομηνίες.
- ❖ **Book_Service:** Με όμοιο τρόπο λειτουργεί και αυτή η N:M σχέση μεταξύ των οντοτήτων Customer και Service. Θεωρούμε πως κάποιος πελάτης μπορεί να κάνει κράτηση για οποιαδήποτε υπηρεσία ανεξαρτήτως αν έχει κάνει κράτηση ή όχι για κάποιο δωμάτιο. Τα γνωρίσματα που θέτει ο χρήστης (AppointedTime και NumOfPeople) είναι πάλι απαραίτητα, ενώ τα γνωρίσματα DateOfBooking, FullPrice και τα ξένα κλειδιά FK1_CustomerID και FK2_ServiceID τίθονται αυτόματα όπως προηγουμένως. Τέλος, κάθε

κράτηση αφορά αυστηρά την επόμενη μέρα και χαρακτηρίζεται από το primary key BookingID.

- ❖ **Make_Special_Resrv:** Έχοντας τα ίδια γνωρίσματα με τη “Make_Simple_Resrv”, η Ν:Μ σχέση είναι μεταξύ των οντοτήτων Customer και Special_Offer. Για να κλείσει ένα δωμάτιο ο πελάτης χρησιμοποιώντας ένα πακέτο προσφοράς, πηγαίνει στην ενότητα “Special Offers”, επιλέγει το πακέτο που τον ενδιαφέρει και εμφανίζονται τα δωμάτια που είναι συνδεδεμένα με αυτό.

- ❖ **Combination:** Για να αποτυπώσουμε την τριμερή σχέση Combines που φαίνεται στο ER, δημιουργήσαμε τον πίνακα Combination, ο οποίος αποτελείται από δύο πεδία RoomID και SpecialOfferID. Τα δύο ξένα κλειδιά, των ο οποίων ο συνδυασμός είναι μοναδικός και άρα πρωτεύον κλειδί της σχέσης, δίνει τη δυνατότητα να θέτουμε για ποια δωμάτια ισχύει το κάθε πακέτο προσφορών. Έτσι με ServiceID να αποτελεί ξένο κλειδί στην οντότητα “Special_Offer”, δημιουργούνται οι συνδυασμοί τις τριμερής σχέσης, με ένα special offer να αντιστοιχεί σε ένα μοναδικό Service αλλά σε πολλά δωμάτια και ένα δωμάτιο ή ένα Service να αντιστοιχούν σε πολλά Special Offers. Για αυτό το λόγο και η σχέση έχει αποτυπωθεί ως 1:N:M για τις οντότητες Special_Offer, Room και Service αντίστοιχα.
- ❖ **Admin:** Ο πίνακας αυτός περιέχει τα στοιχεία AdminID (primary key), Username και pwd. Έχουμε αποθηκεύσει εξαρχής στη βάση μας μια εγγραφή με τιμές <1, ‘admin’, ‘admin’>, ώστε να μπορέσουμε να κάνουμε απευθείας login ως administrators και να επεξεργαστούμε τα δεδομένα της σελίδας μας.

Παρατίθουμε μερικές σημειώσεις για τα γνωρίσματα της βάσης μας:

- Σε όλους τους παραπάνω πίνακες, τα primary keys έχουν οριστεί με την προεπιλογή “AUTO_INCREMENT”, ώστε να τίθονται αυτόματα με μοναδικές τιμές.
- Όλα τα γνωρίσματα για ημερομηνίες είναι τύπου “DATE”, ενώ συγκεκριμένα αυτά με όνομα “DateOfReservation” ή “DateOfBooking” έχουν οριστεί με προκαθορισμένη τιμή “current_timestamp()”, για να συμπληρώνεται με τη σημερινή ημερομηνία.
- Ο τύπος “TINYINT(1)” χρησιμοποιείται από τη MySQL για να αναπαραστήσει τιμές Boolean, καθώς τέτοιο δεν έχει εγγενή τύπο δεδομένων. Με ‘0’ αναπαρίσταται το false και με ‘1’ το true. Τέτοια γνωρίσματα είναι τα “HasHotTub”, “TherapistRequired”, “GuideRequired” και “BookingRequired”.
- Ακολουθήσαμε την πιο αποδοτική προσέγγιση για την αποθήκευση εικόνων στη βάση μας, καθώς στη MySQL αποθηκεύεται συνήθως μόνο η διαδρομή αρχείου (file path). Για αυτό το λόγο στο φάκελο “MtGateway” υπάρχει ο φάκελος “images” όπου αποθηκεύονται όλες οι εικόνες που προβάλονται στην ιστοσελίδα. Έτσι όλα τα γνωρίσματα “Image” είναι τύπου varchar(255).

Μορφή ιστοσελίδας – Οργάνωση Προγράμματος

Η αρχική σελίδα της ιστοσελίδας μας είναι το authentication page και αποτελείται από τρία κομμάτια: User Login, User Signup και Admin Login με το καθένα να έχει ορισμένα πεδία που πρέπει να συμπληρωθούν και το αντίστοιχο κουμπί υποβολής. Τα δύο login μας μεταφέρουν σε μια νέα σελίδα, main page και admin page αντίστοιχα, ενώ με το signup δημιουργείται ένας νέος λογαριασμός με τον οποίο ο χρήστης μπορεί πλέον να συνδεθεί.

Το main page αποτελείται από 4 βασικές ενότητες: “Home”, “Our Rooms”, “Services”, και “Special Offers”, στις οποίες ο χρήστης μπορεί να περιγιηθεί και να βρει πληροφορίες, να πειραματιστεί με τα φίλτρα και εν τέλη να κάνει κράτηση, για κάποιο δωμάτιο, κάποια υπηρεσία ή των συνδυασμό αυτών μέσου ενός πακέτου. Ο χρήστης παραμένει συνδεδεμένος μέχρι να πατήσει το κουπί “Logout” και να επιστρέψει στην αρχική σελίδα.

Από την άλλη το admin page ο υπεύθυνος του ξενοδοχείου μπορεί να προσθέσει, να επεργαστεί και να διαγράψει δωμάτια, υπηρεσίες καθώς και επίσης να δημιουργήσει πακέτα προσφορών συνδέοντας δωμάτια και υπηρεσίες (με τροποποίηση και διαγραφή επίσης).

Για τη μορφοποίηση της ιστοσελίδας χρησιμοποιήθηκε έτοιμο template (βλ “Πηγές”) και όλα του τα αρχεία/φάκελοι τοποθετήθηκαν στο φάκελο “MtGateway” που περιέχει και όλα τα απαραίτητα αρχεία κώδικα για να τρέχει η ιστοσελίδα μας. Συγκεκριμένα, έχουμε την εξής δομή:

- Όλα τα αρχεία στο αρχικό directory είναι οι βασικές μας σελίδες.
- Στον φάκελο “images” αποθηκεύονται όλες οι εικόνες που προβάλλονται.
- Στον φάκελο “includes” βρίσκονται τα αρχεία “dbh.inc.php” (για τη σύνδεση με τη MySQL βάση δεδομένων μας) και “config_session.inc” (για τη διαχείριση του session management), καθώς επίσης και φάκελοι στους οποίους έχουμε ταξινομήσει τα αρχεία κώδικα, ανάλογα με τη λειτουργία τους.

Η γενική δομή που ακολουθήσαμε χρησιμοποιεί τρία είδη αρχείων:

- ❖ model -> Είναι υπεύθυνο για τη διαχείριρης των queries και την απόκτηση δεδομένων από τη βάση μας.
- ❖ view -> Αποτελείται από συναρτήσεις που αφορούν την προβολή δεδομένων που πήραμε από τη βάση στην ιστοσελίδα.
- ❖ controller -> Αποτελείται από συναρτήσεις που επεξεργάζοντα δεδομένα. Βασική λειτουργία είναι να ελέγχουν το input του χρήστη και να διαχειρίζονται errors που προκύπτουν.

Εκτός από τα παραπάνω έχουμε και μερικά pure php files, τα οποία χρησιμοποιούνται κυρίως στη μετάβαση από τη μία σελίδα στην άλλη και ο χρήστης δεν έχει πρόσβαση σε αυτές.

Σύνδεση Κώδικα PHP Με Βάση Δεδομένων

Για να υπάρχει πρόσβαση στη βάση δεδομένων που έχουμε ανεβάσει, απαιτείται η ύπαρξη ενός PHP script που δημιουργεί τη σύνδεση με τη MySQL database χρησιμοποιώντας ένα PDO (PHP Data Objects). Όπως φαίνεται παρακάτω, θέτουμε ως \$dbname το όνομα “hotel”, που δώσαμε στην βάση μας κατά την εγκατάσταση της εφαρμογής. Αυτό το script βρίσκεται στο αρχείο “dbh.inc.php”.

```
<!-- DATABASE HANDLER -->
<?php

$host = 'localhost';
$dbname = 'hotel';
$dbusername = 'root';
$dbpassword = "";

// CONNECT WITH DATABASE
try {
    // Connection object for our database.
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $dbusername, $dbpassword);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}
```

Sessions: Δημιουργία Και Χρήση

Όπως προαναφέρθηκε, η είσοδος στην ιστοσελίδα μας απαιτεί από το χρήστη να κάνει login. Επειδή το login αναφέρεται στη διαδικασία με την οποία ένας χρήστης εισέρχεται στην εφαρμογή, είναι απαραίτητο να δημιουργθεί μία σύνδεση για το χρήστη, η οποία θα επιτρέπει την πρόσβαση στις περιοχές της εφαρμογής που απαιτούν αυθεντικοποίηση. Έτσι ξεκινάμε ένα PHP session, το οποίο είναι ένας μηχανισμός που επιτρέπει αποθήκευση και διαχείριση δεδομένων για έναν χρήστη κατά τη διάρκεια της επίσκεψής του σε μια ιστοσελίδα.

```
<?php
//create session

ini_set('session.use_only_cookies',1);
ini_set('session.use_strict_mode',1);

session_set_cookie_params([
    'lifetime' => 1800,
    'domain' => 'localhost',
    'path' => '/',
    'secure' => true,
    'httponly' => true
]);
session_start();
```

Στο “config_session.inc.php” δημιουργούμε ένα session, όπως φαίνεται στο στιγμιότυπο, και κάνουμε include το αρχείο όποτε είναι απαραίτητο να θέσουμε νέες ή να χρησιμοποιήσουμε υπάρχοντες Session μεταβλητές. Αυτές οι μεταβλητές διατηρούν την τιμής τους και είναι ορατές σε όλη την εκτέλεση του προγράμματος, όσο είναι ενεργό το συγκεκριμένο session.

Αρχική Σελίδα - Authentication Page

Όπως έχει ήδη αναφερθεί, η αρχική σελίδα αποτελείται από User Login, User Signup και Admin Login. Όλα τα παραπάνω βρίσκουνται στο αρχείο index.php, το οποίο αποτελεί το default entry point για κάθε web application. Κάθε λειτουργία έχει μία φόρμα, που ο χρήστης εισάγει τα αντίστοιχα στοιχεία και πατάει το κουμπί υποβολής της.

The screenshot shows the authentication page for 'Mt Gateway'. At the top left is the logo 'Mt Gateway' with a mountain icon. Below it are three main sections: 'User Login' (with fields for Username and Password, and a blue 'Login' button), 'Admin Login' (with fields for Username and Password, and a red 'Login' button), and 'Signup' (with fields for Username, First Name, Last Name, Password, E-mail, and Phone Number, followed by a green 'Signup' button). A large black rectangular redaction box covers the bottom portion of the page content.

User Login – Admin Login

Αν ένας χρήστης υπάρχει ήδη στη βάση δεδομένων, τότε χρειάζεται να κάνει μόνο login για να μεταφερθεί στο main page, συμπληρώνοντας τα στοιχεία του. Στο αρχείο index.php υπάρχει η εξής php φόρμα:

```
<h3>User Login</h3>
<form action="includes/login/login.inc.php" method="post">
    <div class="form-group">
        <input type="text" name="username" class="form-control" placeholder="Username">
    </div>
    <div class="form-group">
        <input type="password" name="pwd" class="form-control" placeholder="Password">
    </div>
    <input type="hidden" name="user_type" value="user">
    <button class="btn btn-primary" type="submit" value="login">Login</button>
</form>
```

Στο παραπάνω απόσπασμα κώδικα παρατηρούμε ότι έχουμε τρία πεδία input: "username", "pwd" και "user_type". Τα δύο πρώτα αφορούν τα στοιχεία που συπλήρωσε ο χρήστης, ενώ το τελευταίο είναι τύπου "hidden", το οποίο σημαίνει ότι δεν φαίνεται στο user και δηλώνεται εσωτερικά στο πρόγραμμα. Η μεταβλητή αυτή χρησιμοποιείται για να ξεχωρίζουμε το user από το admin login χρησιμοποιώντας το ίδιο php file για τη διαχείρισή τους. Επιπλέον σημαντικό στοιχείο αποτελεί το πεδίο "action", στο οποίο θέτουμε το path του αρχείου που θα διαχειριστεί τα δεδομένα που έδωσε ο χρήστης και θέλουμε να εκτελεστεί όταν πατηθεί το κουμπί τύπου "submit".

Το αρχείο με όνομα “login.inc.php” είναι ένα pure php file αυτής της μορφής:

```
<?php

// Do not allow to access this page if the form is not filled.
> if ($_SERVER["REQUEST_METHOD"] == "POST") { ...
}
else {
    header("Location: ../../index.php");
    die();
}
```

Το if-statement που έχουμε ορίσει δεν επιτρέπει να εκτελεστεί αυτό το αρχείο παρά μόνο μέσω της υποβολής της φόρμας. Αν με κάποιο μη έγκυρο τρόπο πάει να πραγματοποιηθεί εκτέλεση, θα γίνει ανακατεύθυνση πίσω στην αρχική σελίδα.

Αν έχει πατηθεί κανονικά το κουμπί υποβολής φόρμας, παίρνουμε την είσοδο, θέτοντας τα δεδομένα σε τοπικές μεταβλητές: πχ. `$username = $_POST["username"]`; με το `$_POST` να είναι ένα PHP super global array, που χρησιμοποιείται για να πάρει την είσοδο του χρήστη σε μία HTML form που χρησιμοποιεί την “POST” ως μέθοδο, όπως η παραπάνω, ενώ το “username” χρησιμοποιείται καθώς αυτό σίχαμε θέσει ως name στη φόρμα.

```
// ERROR HANDLING
$errors = [];
// 1) Empty data
if (is_input_empty($username, $pwd)) {
    $errors["empty_input"] = "Fill in all fields!";
}

if (!strcmp($usertype, "user")) {
    $results = get_user($pdo, $username);
} else {
    $results = get_admin($pdo, $username);
}

// 2) Non-existent username
if (is_username_wrong($username)) {
    $errors["login_incorrect"] = "Incorrect login info!";
} else if (is_password_wrong($pwd, $results["pwd"], $usertype)) {
    $errors["login_incorrect"] = "Incorrect login info!";
}
```

Στη συνέχεια, ελέγχουμε για την εγκυρότητα των δεδομένων που δώθηκαν και πιο συγκεκριμένα ελέγχουμε: αν υπάρχει άδειο πεδίο, αν υπάρχει χρήστης με αυτό το username στη βάση, και αν ναι, αν είναι έγκυρος ο κωδικός.

Οι συναρτήσεις ελέγχου βρίσκονται μέσα στο αρχείο “login_control.inc.php”, αφού διαχειρίζονται πληροφορίες, ενώ τα `get_user()` και `get_admin` είναι στο “login_model.inc.php”, αφού επικοινωνούν μέσω query με τη βάση:

```
function get_user(object $pdo, string $username) {

    // Search if the given username exists in table "customer" and if it does, return the whole registry.
    $query = "SELECT * FROM customer WHERE username = :username;";

    $stmt = $pdo->prepare($query);

    $stmt->bindParam(":username", $username);
    $stmt->execute();

    $results = $stmt->fetch(PDO::FETCH_ASSOC); // Use the name of the column and
                                                // not the index of the table for reference.
}
```

```
require_once "../config_session.inc.php";

$newSessionId = session_create_id();
$sessionId = $newSessionId . "_" . $results["id"];
session_id($sessionId);

// User Logged in
$_SESSION["user_id"] = $results["CustomerID"];
$_SESSION["username"] = htmlspecialchars($results["Username"]); // Sanitize data
$_SESSION["last_regeneration"] = time();

if (!strcmp($usertype, "user")) {
    header("Location: ../../index.php?login_user=success");
}
else {
    header("Location: ../../index.php?login_admin=success");
}

$pdo = null;
$stmt = null;

die();
```

Σε περίπτωση προβλήματος, επιστρέφουμε στην αρχική σελίδα με το login να έχει αποτύχει. Διαφορετικά, δημιουργούμε ένα νέο session για το χρήστη που έκανε login και αποθηκεύουμε σε session variables τα στοιχεία που χρειαζόμαστε όσο αυτός είναι σε ενεργή σύνδεση στη σελίδα μας, συγκεκριμένα τα user_id και username.

To Admin Login ακολουθεί την ίδια ακριβώς διαδικασία, με μόνη διαφορά τη τιμή του hidden input να τίθεται σε “admin” και όχι “user”, ώστε η αναζήτηση της εγγραφής να γίνεται στον πίνακα “Admin_Table” και όχι “Customer”. Άμα έχουμε επιτυχημένη είσοδο, ο διαχειριστής οδηγείται στην ξεχωριστή πλευρά της σελίδας για εισαγωγές, διαγραφές και τροποποιήσεις.

User Signup

Για να μπορέσει ένας χρήστης να συνδεθεί στην ιστοσελίδα του ξενοδοχείου θα πρέπει πρώτα να έχει δηλώσει τα στοιχεία του μέσω της φόρμας “User Signup”. Στο αρχείο index.php καλούνται δύο συναρτήσεις:

1) signup_inputs():

Δημιουργήσαμε τη φόρμα του signup με διαφορετικό τρόπο από αυτή του login, ενσωματώνοντας HTML μέσα σε PHP κώδικα. Με αυτό τον τρόπο η φόρμα γίνεται δυναμική. Έτσι σε περίπτωση προβλήματος κατά την υποβολή, τα έγκυρα πεδία δεν θα σβηστούν, ώστε ο χρήστης να μην χρειαστεί να τα ξαναγράψει.

Αφού υποβληθεί η φόρμα μεταφερόμαστε στο αρχείο signup.inc.php, όπου ελέγχουμε την είσοδο του χρήστη. Συγκεκριμένα ελέγχουμε:

- Αν άφησε κάποιο πεδίο κενό
- Αν έχει δωθεί έγκυρο mail format
- Αν το username υπάρχει ήδη (χρήση query)
- Αν το mail υπάρχει ήδη (χρήση query)
- Αν το τηλέφωνο έχει έγκυρη δομή

```
// Search table "customer" to check if the given username already exists.
$query = "SELECT username FROM customer WHERE username = :username;"

// Search table "customer" to check if the given email already exists.
$query = "SELECT email FROM customer WHERE email = :email;"
```

```
// ERROR HANDLING
$errors = [];
// 1) Empty data
if (is_input_empty($username, $pwd, $email)) {
    $errors["empty_input"] = "Fill in all fields!";
}

// 2) Invalid Email
if (is_email_invalid($email)) {
    $errors["invalid_email"] = "Invalid email used!";
}

// 3) Username already in use
if (is_username_taken($pdo, $username)) {
    $errors["username_taken"] = "Username already taken!";
}

// 4) Email already in use
if (is_email_registered($pdo, $email)) {
    $errors["email_used"] = "Email already registered!";
}

// 5) Invalid Phone Number
if (is_phone_invalid($phoneNum)) {
    $errors["invalid_phoneNum"] = "Invalid phone number!";
}
```

Αν τα δεδομένα ικανοποιούν όλους τους ελέγχους τότε καλούμε την παρακάτω συνάρτηση για να προσθέσουμε το νέο χρήστη στον πίνακα “Customer” (αρχείο signup_model.inc.php):

```
function set_user(object $pdo, string $username, string $fname, string $lname, string $phoneNum, string $pwd, string $email) {
    // Search table "customer" to check if the given email already exists.
    $query = "INSERT INTO customer (username, pwd, email, fname, lname, phoneNumber) VALUES (:username, :pwd, :email, :fname, :lname, :phoneNum);";

    $stmt = $pdo->prepare($query);

    // Hash the password for security
    $options = [
        'cost' => 12
    ];
    $hashedPwd = password_hash($pwd, PASSWORD_BCRYPT, $options);

    $stmt->bindParam(":username", $username);
    $stmt->bindParam(":pwd", $hashedPwd);
    $stmt->bindParam(":email", $email);
    $stmt->bindParam(":fname", $fname);
    $stmt->bindParam(":lname", $lname);
    $stmt->bindParam(":phoneNum", $phoneNum);
    $stmt->execute();
}
```

Οι παράμετροι που περνάμε στη συνάρτηση είναι οι τιμές που έδωσε μέσω της φόρμας ο χρήστης με εξαίρεση το “pwd”, το οποίο κωδικοποιούμε για λόγους ασφαλείας, όπως φαίνεται παρακάτω:

```
// Hash the password for security
$options = [
    'cost' => 12
];
$hashedPwd = password_hash($pwd, PASSWORD_BCRYPT, $options);
```

Έτσι αν ο χρήστης εισάγει '123' ως κωδικό, το αντίστοιχο πεδίο της βάση θα περιέχει:

'\$2y\$12\$ZrysYx4hKleYXHVF/4xcGe.yOVW0D3wbNABdKef3LKW...'. Στο login χρησιμοποιώντας την αντίστοιχη συνάρτηση ελέγχουμε την εγκυρότητα του κωδικού που μας δώθηκε.

Ο χρήστης επιστρέφει στην σελίδα που βρισκόταν με τη URL να έχει αλλάξει σε:

localhost/MtGateway/index.php?signup=success

Τώρα είναι έτοιμος να κάνει login και να περιηγηθεί στη σελίδα του ξενοδοχείου.

Website - User Side

Έχοντας συνδεθεί πλέον, ο χρήστης μπορεί να περιηγηθεί στην ιστοσελίδα. Κάθε σελίδα έχει μια βασική μορφή με το header και το footer να είναι ίδια σε όλα, με εξαίρεση το ότι κάθε φορά φαίνεται μέσω της υπογράμμισης σε ποια ενότητα βρισκόμαστε.

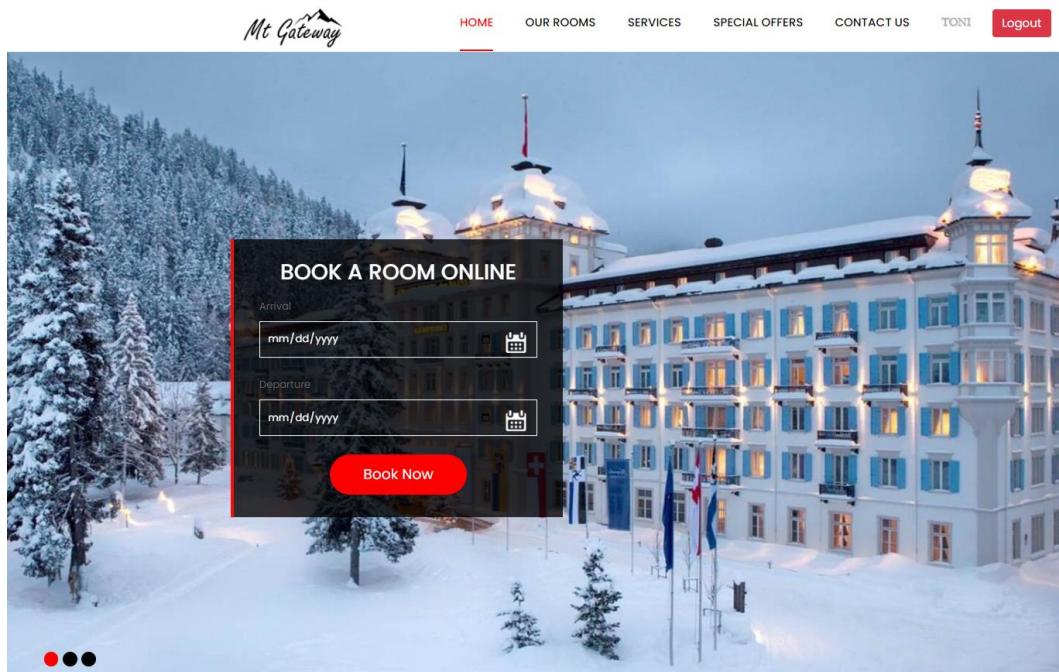
Το header αποτελείται από τις ενότητες με βάση τις οποίες χωρίζεται η σελίδα. Αριστέρα έχει τοποθετηθεί το logo του ξενοδοχείου, ενώ δεξιά φαίνεται το username του χρήστη που έχει συνδεθεί και το κουμπί logout με το οποίο μπορεί να αποχωρίσει από την ιστοσελίδα.

Το footer προβάλει πάλι τη λίστα των ενοτήτων, ενώ δίνει και credits στην πηγή από όπου πήραμε το template.

The screenshot shows the Mt Gateway website's user interface. At the top, there is a header with the logo 'Mt Gateway' on the left and a navigation menu with links for HOME, OUR ROOMS, SERVICES, SPECIAL OFFERS, CONTACT US, and a user account section (TONI and Logout). Below the header is a dark footer section containing three columns: 'Contact US' with address, phone number, and email information; 'Menu Link' with a list of site pages; and 'News letter' with an input field for email and a subscribe button. Social media icons for Facebook, Twitter, LinkedIn, and YouTube are also present. At the very bottom, there is small text indicating copyright and distribution details.

Main Page – Home

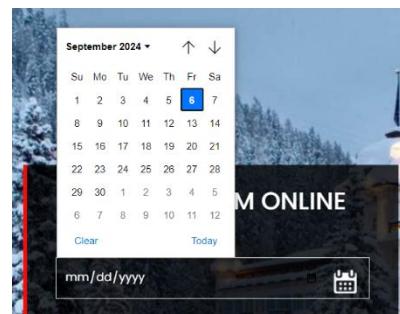
Η παρακάτω είναι η πρώτη εικόνα που αντικρίζει ο χρήστης μετά τη σύνδεσή σου. Το περιχόμενο της σελίδας αποτελείται από ένα Carousel - Slider Images, πάνω στο οποίο υπάρχει ένα παράθυρο για να επιλέχθει η ημερομηνία άφιξης και αναχώρισης στο ξενοδοχείο. Με αυτόν τον τρόπο, ο χρήστης μπορεί άμεσα να αναζητήσει τα διαθέσιμα δωμάτια για τις ημερομηνίες που τον ενδιαφέρουν.



Στην πραγματικότητα, αυτό το παράθυρο αποτελεί μια ακόμη φόρμα με δύο πεδία εισόδου "arrival" και "departure" και με το κουμπί υποβολής να ανακατευθύνει στο αρχείο "book_now.inc.php". Αυτό το αρχείο, όπως και κάθε άλλο που διαχειρίζεται την υποβολή μας φόρμας, δεν επιτρέπεται να εκτελεστεί αν δεν έχει γίνει η ανακατεύθυνση με έγκυρο τρόπο.

```
<div class="book_room">
  <h1>Book a Room Online</h1>
  <form class="book_now" action="includes/book_now/book_now.inc.php" method="post">
    <div class="row">
      <div class="col-md-12">
        <label for="arrival">Arrival</label>
        
        <input id="arrival" class="online_book" type="date" name="arrival" required>
      </div>
      <div class="col-md-12">
        <label for="departure">Departure</label>
        
        <input id="departure" class="online_book" type="date" name="departure" required>
      </div>
      <div class="col-md-12">
        <button class="book_btn" type="submit">Book Now</button>
      </div>
    </div>
  </form>
</div>
```

Ελέγχουμε ότι τα inputs είναι έγκυρα, δηλαδή μη κενά, γραμμένα με date format, με το arrival να δείχνει σε μελλοντική ημερομηνία και το departure σε κάποια επόμενη. Αν και έχει δωθεί ο τύπος "date" στα πεδία της φόρμας, οι έλεγχοι πρέπει να γίνονται καθώς η html και css δεν είναι υπεύθυνες για τη διαχείριση πληροφοριών και μπορούν εύκολα να παρακαμφθούν. Αν δεν προκύψει πρόβλημα τότε καλείται η συνάρτηση `get_available_rooms()`, με ορίσματα τις δοσμένες ημερομηνίες, ώστε μέσω ενός query αναζήτησης να λάβουμε όλα τα διαθέσιμα προς κράτηση δωμάτια για το για το χρονικό διάστημα που ζητήθηκε. Η συνάρτηση αυτή θα αναλυθεί σε επόμενη ενότητα.



Τέλος, θέτουμε ένα session array “available_rooms” στο οποίο αποθηκεύουμε τα αποτελέσματα του query. Μεταφερόμαστε στο αρχείο “room.php”, θέτοντας και δύο query parameters με τις τιμές των επιλεγμένων arrival και departure. Με αυτόν τον τρόπο μπορούμε να μεταφέρουμε δεδομένα από τη μία σελίδα στην άλλη μέσω της διεύθυνσης URL.

```
if ($errors) {
    $_SESSION["errors_book_now"] = $errors;
    header("Location: ../../main_page.php");
    die();
}

$availableRooms = get_available_rooms($pdo, $arrival, $departure, "", "", "", "", "", "");
// Store the available rooms in the session
$_SESSION['available_rooms'] = $availableRooms;

// Redirect to a new page
header("Location: ../../room.php?arrival=" . urlencode($arrival) . "&departure=" . urlencode($departure));
exit(); // Exit after redirect
```

Our Rooms

Η ενότητα “Our Rooms” έχει την παρακάτω μορφή, με τα βασικά στοιχεία να είναι το filters section και τα δωμάτια με τις πληροφορίες τους.

The screenshot shows the 'Mt Gateway' website's 'OUR ROOMS' section. At the top, there is a navigation bar with links for HOME, OUR ROOMS (which is highlighted in red), SERVICES, SPECIAL OFFERS, CONTACT US, and a user account section with 'TONI' and 'Logout'. Below the navigation is a large black banner with the text 'ALL AVAILABLE ROOMS FOR YOUR SELECTED DATES' in white. Underneath the banner is a form for applying filters, including fields for Arrival Date, Departure Date, Room Type, Number of Beds, Capacity, and Order by Price, along with 'Apply Filters' and 'Clear Filters' buttons. The main content area displays three room categories with their respective images, names, descriptions, and booking buttons:

Room Category	Image	Name	Room Type	Number Of Beds	Capacity	Hot Tub Included	Price Per Night	Action
Royal Life		Royal Life	Queen	1	2	No	\$30	Book Now
Rock&Roll		Rock&Roll	Double	2	2	Yes	\$80	Book Now
Peace and Quiet		Peace and Quiet	Single	1	1	No	\$35	Book Now

Υπάρχουν δύο βασικοί τρόποι να βρεθεί ο χρήστης σε αυτή την σελίδα (“room.php”):

- 1) Μέσω της επιλογής της ενότητας “Our Rooms” από το header ή το footer, στο οποίο εμφανίζονται όλα τα υπάρχοντα δωμάτια.
- 2) Μέσω της εφαρμογής φίλτρων, που εμφανίζονται τα δωμάτια που ικανοποιούν ορισμένες συνθήκες (αυτές που όρισε ο χρήστης)

Μετάβαση από καρτέλα – Προβολή όλων των δωματίων

Όταν επιλεγεί η ενότητα “Our Rooms”, όπως έχει οριστεί και στο κώδικα, θα πρέπει αρχικά να μεταφερθούμε στο αρχείο “get_all_rooms.php” και μετά με ανακατεύθυνση θα φτάσουμε στο “room.php”.

Αυτό γίνεται ώστε να κληθεί η συνάρτηση get_all_rooms(), η οποία ανακτά όλες τις εγγραφές από τον πίνακα “room”. Η διαδικασία αυτή είναι απαραίτητη διότι η σελίδα “room.php” θα εμφανίσει στην οθόνη τις πληροφορίες των δωματίων που είναι αποθηκευμένες στο \$_SESSION['available_rooms']. Επομένως, το session array θα πρέπει να έχει αρχικοποιηθεί και αφού δεν υπάρχει για την ώρα κάποιος περιορισμός/φίλτρο, ανακτάμε όλοκληρο τον πίνακα.

```
<li class="nav-item">
  <a class="nav-link" href="get_all_rooms.php">Our rooms</a>
</li>

function get_all_rooms($pdo) {
    $query = "SELECT * FROM room";
    $stmt = $pdo->prepare($query);
    $stmt->execute();

    $availableRooms = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $availableRooms;
}
```

Χρήση φίλτρων – Προβολή δωματίων με συνθήκες

Όλο το filter section είναι δημιουργημένο ως ένα μεγάλο form, το οποίο διατηρεί τις τιμές των πεδίων αν είχαν συμπληρωθεί προηγουμένως. Έτσι, ο χρήστης μπορεί να εφαρμόζει φίλτρα σειριακά χωρίς να χάνονται οι προηγούμενες επιλογές του. Η σελίδα έχει πρόσβαση στα παλιότερα δεδομένα, διότι τα μεταφέρουμε μέσω της URL ως query parameters. Αν λοιπόν δίνεται τιμή για το αντίστοιχο label μέσω της super global variable \$_GET(), κρατώ το προηγούμενο input, διαφορετικά το πεδίο παραμένει κενό.

Οι δυνατότητες που δίνονται στα φίλτρα είναι οι εξής:

1) Ένα input τύπου date για το arrival και ένα αντίστοιχο για το πεδίο departure.

```
<div class="col-md-3">
  <label for="arrival">Arrival Date:</label>
  <input type="date" name="arrival" id="arrival" class="form-control"
         value=<?php echo isset($_GET['arrival']) ? htmlspecialchars($_GET['arrival']) : null; ?>">
</div>
```

2) Τρία φίλτρα τύπου dropdown menu για το “Room Type”, το “Number of Beds” και το “Capacity”.

```
<div class="col-md-3">
  <label for="num_beds">Number of Beds:</label>
  <select name="num_beds" id="num_beds" class="form-control">
    <option value="">Any</option>
    <option value="1" ><?php echo isset($_GET['num_beds']) && $_GET['num_beds'] == '1' ? 'selected' : null; ?>>1</option>
    <option value="2" ><?php echo isset($_GET['num_beds']) && $_GET['num_beds'] == '2' ? 'selected' : null; ?>>2</option>
    <option value="3" ><?php echo isset($_GET['num_beds']) && $_GET['num_beds'] == '3' ? 'selected' : null; ?>>3</option>
    <option value="4" ><?php echo isset($_GET['num_beds']) && $_GET['num_beds'] == '4' ? 'selected' : null; ?>>4</option>
  </select>
</div>
```

- 3) Ένα ακόμη dropdown menu που αφορά τη ταξινόμηση των δωματίων που θα εμφανιστούν με βάση τη τιμή τους. Η επιλογή “Any” δηλώνει πως δεν υπάρχει κάποια σειρά και τα δωμάτια εμφανίζονται με τη σειρά που ανακτήθηκαν από τη βάση.

```
<div class="col-md-3">
    <label for="sort_order">Order by Price:</label>
    <select name="sort_order" id="sort_order" class="form-control">
        <option value="">Any</option>
        <option value="asc" <?php echo isset($_GET['sort_order']) && $_GET['sort_order'] == 'asc' ? 'selected' : null; ?>>Low to High</option>
        <option value="desc" <?php echo isset($_GET['sort_order']) && $_GET['sort_order'] == 'desc' ? 'selected' : null; ?>>High to Low</option>
    </select>
</div>
```

- 4) Μια μπάρα αναζήτησης, στην οποία ο χρήστης μπορεί να πληκτρολογήσει και να ψάξει δωμάτια με βάση το όνομά τους. Αν κάποιο όνομα περιέχει το input του χρήστη αυτό θα εμφανιστεί στα διαθέσιμα δωμάτια, διαφορετικά όχι.

```
<div class="col-md-3">
    <label for="search" class="form-label"></label>
    <input type="text" name="search" id="search" class="form-control" placeholder="Search by room name...">
    <input type="text" name="search" id="search" class="form-control" value="<?php echo isset($_GET['search']) ? htmlspecialchars($_GET['search']) : null; ?>">
</div>
```

Τέλος, η φόρμα περιέχει δύο κουμπιά: ένα “Apply Filters” για την υποβολή της και ένα “Clear Filters”, το οποίο διαγράφει οποιαδήποτε επιλογή υπήρχε και μεταφερόμαστε στην αντίστοιχη σελίδα με κάθε input να είναι άδειο.

```
<button type="submit" class="btn btn-primary mt-3">Apply Filters</button>
<button type="reset" class="btn btn-secondary mt-3" onclick="window.location.href='filter_rooms.php';">Clear Filters</button>
```

Πατώντας οποιοδήποτε από τα δύο κουμπιά μεταφερόμαστε στη σελίδα “filter_rooms.php”, στην οποία ελέγχουμε, μέσω του \$_GET(), ποια πεδία συμπληρώθηκαν και κρατάμε στην αντίστοιχη μεταβλητή την τιμή του. Η τιμή αυτή αποθηκεύεται αφού περάσει από τη htmlspecialchars(), μία build-in συνάρτηση της PHP, που χρησιμοποιείται για την ασφαλή απόδοση χαρακτήρων HTML, προκειμένου να αποτραπεί η εκτέλεση κακόβουλου κώδικα μέσω εισαγωγής HTML.

Όπως φαίνεται στο στιγμιότυπο, καλείται η συνάρτηση get_available_rooms με παραμέτρους όλες τις μεταβλητές που αντιστοιχούν σε φίλτρα. Επιστρέφεται ένα σύνολο εγγραφών από τον πίνακα “room” που συμφωνεί με τους καθορισμένους περιορισμούς και αποθηκεύεται σε ένα session variable για να υπάρχει πρόσβαση σε αυτό μετά τη μετακίνηση σε άλλη σελίδα. Τέλος, γίνεται ανακατεύθυνση στην προηγούμενη σελίδα “rooms.php”, θέτοντας αυτή τη φορά όλα τα query parameters με τις αντίστοιχες τιμές των μεταβλητών. Έτσι ο χρήστης, μετά την εφαρμογή των φίλτρων, θα βρεθεί στην ίδια σελίδα με τη διαφορά ότι τα φίλτρα που είχε επιλέξει θα εξακολουθούν να υπάρχουν και τα εμφανιζόμενα δωμάτια θα ικανοποιούν τους περιορισμούς του. Η διαδικασία αυτή μπορεί να επαναληφθεί προσθέτοντας ή τροποποιώντας τις επιλογές στα φίλτρα.

```

<?php

session_start(); // Ensure session is started

// Retrieve form data
$arrival = isset($_GET['arrival']) ? htmlspecialchars($_GET['arrival']) : null;
$departure = isset($_GET['departure']) ? htmlspecialchars($_GET['departure']) : null;
$room_type = isset($_GET['room_type']) ? htmlspecialchars($_GET['room_type']) : null;
$num_beds = isset($_GET['num_beds']) ? htmlspecialchars($_GET['num_beds']) : null;
$capacity = isset($_GET['capacity']) ? htmlspecialchars($_GET['capacity']) : null;
$sort_order = isset($_GET['sort_order']) ? htmlspecialchars($_GET['sort_order']) : null;
$search = isset($_GET['search']) ? htmlspecialchars($_GET['search']) : null;

try {
    require_once "includes/dbh.inc.php";      // Connect to the database.
    require_once "includes/book_now/book_now_model.inc.php";

    // Call the function to get available rooms
    $availableRooms = get_available_rooms($pdo, $arrival, $departure, $room_type, $num_beds, $capacity, $sort_order, $search);

    // Store the available rooms in the session
    $_SESSION['available_rooms'] = $availableRooms;

    // Redirect to a new page with filter parameters
    header("Location: room.php?arrival=" . urlencode($arrival) . "&departure=" . urlencode($departure) . "&room_type=" .
        urlencode($room_type) . "&num_beds=" . urlencode($num_beds) . "&capacity=" .
        urlencode($capacity) . "&sort_order=" . urlencode($sort_order) . "&search=" .
        urlencode($search));
    exit(); // Make sure to exit after redirect
} catch (PDOException $e) {
    die("Query failed: " . $e->getMessage());
}
?>

```

Παραθέτουμε ένα παράδειγμα εφαρμογής: Στην πρώτη εικόνα φαίνεται η αρχική προβολή όλων των δωματίων, ενώ στη δεύτερη έχει δωθεί στο φίλτρο “Capacity” η τιμή ‘2’, με αποτέλεσμα το τρίτο δωμάτιο που φαινόταν προηγουμένως να έχει απορριφθεί πλέον καθώς σε αυτό το πεδίο έχει την τιμή ‘1’. Στο επόμενο στιγμιότυπο παρατηρούμε ότι το προηγούμενο φίλτρο “Capacity” έχει διατηρηθεί, ενώ τώρα τα δωμάτια ικανοποιούν το συνδυασμό των δύο περιορισμών. Για να γίνει πιο συγκεκριμένος ο χρήστης πληκτρολόγισε τη λέξη “love” στο search bar, με αποτέλεσμα από τα δωμάτια που υπήρχαν να μείνουν μόνο αυτά που περιέχουν τη λέξη-κλειδί στον τίτλο τους. Τέλος, πατώντας το κουμπί “Clear Filters” επιστρέφουμε στην αρχική κατάσταση της σελίδας.

Arrival Date: mm/dd/yyyy Departure Date: mm/dd/yyyy Room type: Any

Number of Beds: Any Capacity: Any Order by Price: Any

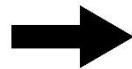
Arrival Date: mm/dd/yyyy Departure Date: mm/dd/yyyy Room type: Any

Number of Beds: 2 Capacity: Any Order by Price: Any

Royal Life
Room Type: Queen
Number Of Beds: 1
Capacity: 2
Hot Tub Included
Price Per Night: \$30
[Book Now](#)

Rock&Roll
Room Type: Double
Number Of Beds: 2
Capacity: 2
Hot Tub Included
Price Per Night: \$80
[Book Now](#)

Peace and Quiet
Room Type: Single
Number Of Beds: 1
Capacity: 1
Hot Tub Included
Price Per Night: \$35
[Book Now](#)



Royal Life
Room Type: Queen
Number Of Beds: 1
Capacity: 2
Hot Tub Included
Price Per Night: \$30
[Book Now](#)

Rock&Roll
Room Type: Double
Number Of Beds: 2
Capacity: 2
Hot Tub Included
Price Per Night: \$80
[Book Now](#)

Love Suite
Room Type: Double
Number Of Beds: 1
Capacity: 2
Price Per Night: \$45
[Book Now](#)

Arrival Date: mm/dd/yyyy Departure Date: mm/dd/yyyy Room type: Double

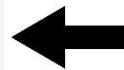
Number of Beds: 2 Capacity: Any Order by Price: Any

Arrival Date: mm/dd/yyyy Departure Date: mm/dd/yyyy Room type: Double

Number of Beds: 2 Capacity: Any Order by Price: Any

Love Suite
Room Type: Double
Number Of Beds: 1
Capacity: 2
Price Per Night: \$45
[Book Now](#)

Love For two
Room Type: Double
Number Of Beds: 1
Capacity: 2
Hot Tub Included
Price Per Night: \$50
[Book Now](#)



Rock&Roll
Room Type: Double
Number Of Beds: 2
Capacity: 2
Hot Tub Included
Price Per Night: \$80
[Book Now](#)

Love Suite
Room Type: Double
Number Of Beds: 1
Capacity: 2
Price Per Night: \$45
[Book Now](#)

Love For two
Room Type: Double
Number Of Beds: 1
Capacity: 2
Hot Tub Included
Price Per Night: \$50
[Book Now](#)

Arrival Date: mm/dd/yyyy Departure Date: mm/dd/yyyy Room type: Any

Number of Beds: Any Capacity: Any Order by Price: Any

Royal Life
Room Type: Queen
Number Of Beds: 1
Capacity: 2
Hot Tub Included
Price Per Night: \$30
[Book Now](#)

Rock&Roll
Room Type: Double
Number Of Beds: 2
Capacity: 2
Hot Tub Included
Price Per Night: \$80
[Book Now](#)

Peace and Quiet
Room Type: Single
Number Of Beds: 1
Capacity: 1
Hot Tub Included
Price Per Night: \$35
[Book Now](#)

Ο δεύτερος τρόπος που μπορούμε να φτάσουμε στη σελίδα “room.php” είναι το κουμπί “Book Now” στο main page. Στην πραγματικότητα, αυτό δουλεύει στην λογική των φίλτρων, αλλά μόνο για τα πεδία arrival και departure, αφού καλεί την ίδια συνάρτηση, δίνοντας το κενό (“”) ως τιμή για τα υπόλοιπα πεδία και θέτει τη `$_SESSION['available_rooms']` με βάση το αποτέλεσμα που επέστρεψε.

Στη συνέχεια αναλύουμε την υλοποίηση της συνάρτησης `get_available_rooms` με args τα: `$pdo, $arrival, $departure, $room_type, $num_beds, $capacity, $sort_order, $search`.

Αν έχουν δωθεί οι ημερομηνίες διαμονής, τότε είναι απαραίτητο να ελέγξουμε αν υπάρχουν κρατήσεις δωματίων για κάποια περίοδο που έχει επικάλυψη με την επιθυμητή, ώστε να τα αποκλείσουμε από τα διαθέσιμα.

Αρχικά, χρησιμοποιούμε το παρακάτω query για να ανακτήσουμε όλα τα δωμάτια που είναι κλεισμένα εκείνο το διάστημα. Για να καλύψουμε κάθε σενάριο overlap των ημερομηνιών, θέσαμε τη συνθήκη `!(B <= C || A >= D)` με `[A, B]` και `[C, D]` διαστήματα. Επιπλέον, όπως έχει αναφερθεί υπάρχουν δύο πίνακες στους οποίους αποθηκεύονται κρατήσεις δωματίων: ο “make_simple_resv” και “make_special_resv”. Άρα αναζητάμε στον κάθε πίνακα τα δωμάτια που είναι κλεισμένα σε εκείνο το διάστημα και κρατάμε την ένωση των δύο συνόλων, χρησιμοποιώντας την εντολή “UNION” της SQL. Επιλέγουμε να ανακτήσουμε μόνο τα roomIDs αφού εν τέλει αυτές οι εγγραφές θα αποκλειστούν.

```
if ($arrival != null && $departure != null) {
    // Step 1: Find all room IDs that are already booked during the given period
    $query = "
        SELECT FK2_RoomID
        FROM makes_simple_resrv
        WHERE NOT (:departure <= checkIn OR :arrival >= checkout)
        UNION
        SELECT FK2_RoomID
        FROM makes_special_resrv
        WHERE NOT (:departure <= checkIn OR :arrival >= checkout)
    ";

    $stmt = $pdo->prepare($query);
    $stmt->bindParam(':arrival', $arrival);
    $stmt->bindParam(':departure', $departure);
    $stmt->execute();

    $bookedRooms = $stmt->fetchAll(PDO::FETCH_COLUMN, 0); // Fetch only the room IDs
} else {
    $bookedRooms = [];
}
```

Στη συνέχεια, χτίζουμε σταδιακά το SQL query με βάση τις τιμές των φίλτρων που έχουμε. Ξεκινάμε με την εντολή “`SELECT * FROM room`”, αφού θέλουμε πρόσβαση σε όλα τα πεδία της εγγραφής του δωματίου. Αν ο πίνακας `$bookedRooms` δεν είναι άδειος, τότε θέλουμε να αποκλείσουμε όλες τις πλειάδες που περιέχει, καθώς δεν είναι διαθέσιμες. Χρησιμοποιούμε τον τελεστή “`NOT IN`” για να αποκλείσουμε οποιαδήποτε εγγραφή του πίνακα “room” ανήκει σε αυτό το σύνολο.

Επισημαίνουμε πως για κάθε νέα συνθήκη, προσθέτουμε το SQL statement σε έναν πίνακα `$conditions`, ενώ στον πίνακα `$params` αποθηκεύουμε τα αντίστοιχα placeholders, στα οποία αναθέτουμε και την τιμή της αντίστοιχης μεταβλητής.

```

// Step 2: Build the base query
$query = "SELECT * FROM room";
$conditions = [];

if (!empty($bookedRooms)) {
    // Use named placeholders for the NOT IN clause
    $placeholders = [];
    foreach ($bookedRooms as $index => $roomID) {
        $placeholders[] = ":roomID_{$index}";
        $params[":roomID_{$index}"] = $roomID;
    }
    $conditions[] = "RoomID NOT IN (" . implode(', ', $placeholders) . ")";
}

```

Στη συνέχεια, ελέγχουμε αν έχει συμπληρωθεί κάποιο από τα “Room Type”, “Number of Beds” ή/και “Capacity”. Αν ναι, πρέπει να προσθέσουμε τη συνθήκη ισότητας για το αντίστοιχο γνώρισμα.

Σειρά έχει η μπάρα αναζήτησης. Αν ο χρήστης έχει πληκτρολογήσει κάτι, τότε θέλουμε να εμφανίσουμε δωμάτια, για τα οποία το όνομά τους περιέχουν, σαν μέρος της συμβολοσειράς τους, το input του χρήστη. Για αυτό το λόγο χρησιμοποιείται ο τελεστής “LIKE” στο γνώρισμα “RoomName”.

```

// Apply filters if provided
if (!empty($room_type)) {
    $conditions[] = "RoomType = :room_type";
    $params[':room_type'] = $room_type;
}
if (!empty($num_beds)) {
    $conditions[] = "NumOfBeds = :num_beds";
    $params[':num_beds'] = $num_beds;
}
if (!empty($capacity)) {
    $conditions[] = "Capacity = :capacity";
    $params[':capacity'] = $capacity;
}

```

```

// Add search filter if provided
if (!empty($search)) {
    $conditions[] = "(RoomName LIKE :search)";
    $params[':search'] = '%' . $search . '%'; // Add wildcards for partial matching
}

```

Όλες οι παραπάνω προϋποθέσεις, είναι συνθήκες για το τμήμα “WHERE” του query και θέλουμε να ισχύουν ταυτόχρονα επομένως θα χρησιμοποιήσουμε τον τελεστή “AND” και θα συμπληρώσουμε το αρχικό SQL ερώτημα με βάση τους δύο πίνακες που συμπληρώσαμε.

```

// Combine conditions into the query
if (!empty($conditions)) {
    $query .= " WHERE " . implode(" AND ", $conditions);
}

```

Τέλος, αν ο χρήστης έχει επιλέξει τρόπο ταξινόμησης, προσθέτουμε στο ερώτημα και το τμήμα “ORDER BY”, εκτελούμε το query και τελικά επιστρέφουμε το αποτέλεσμα.

```

// Add sorting if the sort_order is provided
if (!empty($sort_order)) {
    $query .= " ORDER BY PricePerNight " . ($sort_order === 'asc' ? 'ASC' : 'DESC');
}

// Prepare and execute the query
$stmt = $pdo->prepare($query);

// Bind all parameters directly in the execute() method
$stmt->execute($params);

$availableRooms = $stmt->fetchAll(PDO::FETCH_ASSOC);
return $availableRooms;

```

Με αυτόν τον τρόπο, μπορούμε να χρησιμοποιούμε αυτή τη συνάρτηση ανεξάρτητα από το πόσα και ποια φίλτρα έχουν επιλεγεί, και να βρίσκουμε όλα τα διαθέσιμα δωμάτια που μπορεί να ενδιαφέρουν τον χρήστη.

Σενεπώς, έχοντας συμπληρώσει σε κάθε περίπτωση τη \$_SESSION['available_rooms'], μέσω της συνάρτησης output_available_rooms() εμφανίζονται στην ιστοσελίδα οι πληροφορίες για τα δωμάτια που ανακτήθηκαν.

Κράτηση Δωματίου – Εισαγωγή στον πίνακα “make_simple_resrv”

Για να κάνει κράτηση ενός δωματίου, ο χρήστης πατάει το αντίστοιχο κουμπί “Book Now”. Όπως φαίνεται στη συνάρτηση output_available_rooms() (αρχείο book_now_view.inc.php), κάθε δωμάτιο έχει το δικό του κουμπί κράτησης, που μας μεταφέρει στην αντίστοιχη URL, η οποία συμπεριλαμβάνει ως δεδομένα, το RoomID του δωματίου και τις επιλιγμένες ημερομηνίες:

```

// URL for booking the specific room
$bookingurl = 'room_details.php?room_id=' . urlencode($room['RoomID'])
. '&arrival=' . urlencode($arrival) . '&departure=' . urlencode($departure);

```

Στο αρχείο “room_details.php”, ανακτούμε αυτά τα δεδομένα, ενώ αν κάποιο πεδίο είναι άδειο επιστρέφουμε πίσω στην προηγούμενη σελίδα, καθώς δεν είναι έγκυρη η κράτηση αν δεν είναι επιλεγμένες οι ημερομηνίες ή το δωμάτιο.

```

$arrival = isset($_GET['arrival']) ? htmlspecialchars($_GET['arrival']) : null;
$departure = isset($_GET['departure']) ? htmlspecialchars($_GET['departure']) : null;
$room_id = isset($_GET['room_id']) ? htmlspecialchars($_GET['room_id']) : null;
$user_id = $_SESSION["user_id"];
$username = isset($_SESSION['user_username']) ? $_SESSION['user_username'] : null;

if ($arrival == null || $departure == null || $room_id == null) {
    header("Location: room.php");
    exit(); // Make sure to exit after redirect
}

$curren_room = get_room($pdo, $room_id);
?>

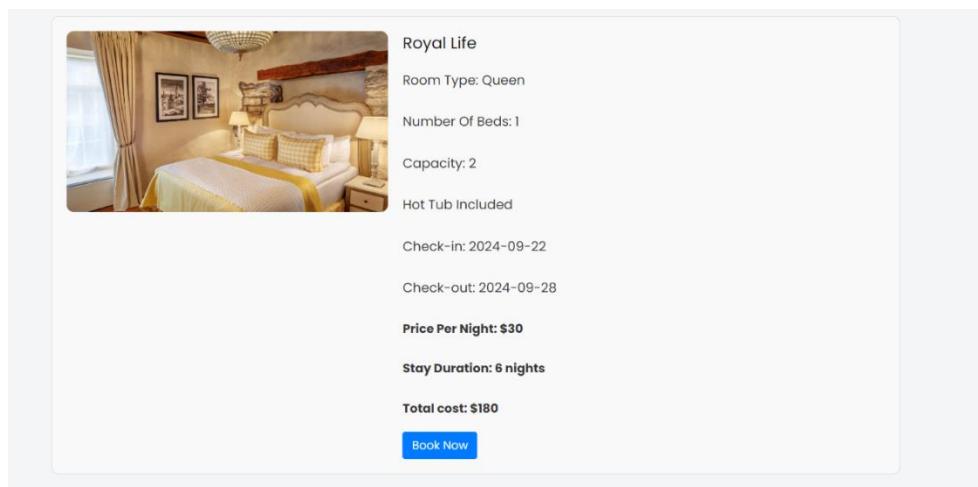
```

Διαφορετικά, καλείται η συνάρτηση `get_room()` (αρχείο `book_now_model.inc.php`), για να ανακτήσουμε από τη βάση όλα τα γνωρίσματα του επιλεγμένου δωματίου.

```
function get_room($pdo, $room_id) {
    $query = "SELECT * FROM room WHERE RoomID = :room_id";
    $stmt = $pdo->prepare($query);
    $stmt->bindParam(':room_id', $room_id, PDO::PARAM_INT);
    $stmt->execute();

    $curr_room = $stmt->fetch(PDO::FETCH_ASSOC);
    return $curr_room;
}
```

Μέσα στη σελίδα καλείται η συνάρτηση `output_room_info()` (αρχείο `book_now_view.inc.php`), η οποία προβάλει τα γνωρίσματα της εγγραφής και τις λεπτομέρεις της κράτησης.



Πατώντας το κουμπί “Book Now” ο χρήστης επιβεβαιώνει την κράτηση του και προστίθεται μία νέα εγγραφή στον πίνακα “make_simple_resrv”. Στην πραγματικότητα, το κουμπί υποβάλλει μία φόρμα της οποία τα πεδία είναι προεπιλεγμένα (hidden). Συγκεκριμένα, ο χρήστης είχε επιλεγμένα τα γνωρίσματα “RoomID”, “CheckIn” και “CheckOut” από τις προηγούμενες σελίδες, ενώ το “FK1_CustomerID” το λαμβάνω από τη σύνδεση του χρήστη (`$_SESSION["user_id"]`). Τέλος το “FullPrice” υπολογίζεται με έναν απλό πολλαπλασιασμό του πλήθους των ημερών διαμονή και του γνωρίσματος του δωματίου “PricePerNight”.

```

// Convert arrival and departure to DateTime objects
$arrivalDate = new DateTime($arrival);
$departureDate = new DateTime($departure);

// Calculate the difference in days between check-in and check-out
$stayDuration = $departureDate->diff($arrivalDate)->days;

echo '<h3>' . "Check-in: " . htmlspecialchars($arrival) . '</h3>';
echo '<h3>' . "Check-out: " . htmlspecialchars($departure) . '</h3>';
echo '<h3 class="price-per-night">Price Per Night: $' . htmlspecialchars($curr_room['PricePerNight']) . '</h3>';
echo '<h3 class="price-per-night">Stay Duration: ' . $stayDuration . ' nights</h3>';
echo '<h3 class="price-per-night">Total cost: $' . ($stayDuration * $curr_room['PricePerNight']) . '</h3>';

echo '<form action="includes/book_now/make_simple_resrv.inc.php" method="POST">';
echo '<input type="hidden" name="room_id" value="' . htmlspecialchars($curr_room['RoomID']) . '">';
echo '<input type="hidden" name="user_id" value="' . htmlspecialchars($user_id) . '">';
echo '<input type="hidden" name="arrival" value="' . htmlspecialchars($arrival) . '">';
echo '<input type="hidden" name="departure" value="' . htmlspecialchars($departure) . '">';
echo '<input type="hidden" name="totalCost" value="' . htmlspecialchars($stayDuration * $curr_room['PricePerNight']) . '">';
echo '<button type="submit" class="btn btn-primary">Book Now</button>';
echo '</form>';

```

Με την υποβολή της φόρμας, μεταφερόμαστε στο αρχείο “make_simple_resrv.inc.php”, όπου θα πραγματοποιηθεί η προσθήκη εγγραφής στον αντίστοιχο πίνακα με γνωρίσματα τα πεδία της φόρμας. Αν η προσθήκη είναι επιτυχής μεταφερόμαστε στη σελίδα “reservation_success.php”, διαφορετικά, αν το query απέτυχε, στην “reservation_error.php”.

```

// Submit Reservation
// Check if the form was submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the data from the form
    $user_id = $_POST['user_id'];
    $room_id = $_POST['room_id'];
    $arrival = $_POST['arrival'];
    $departure = $_POST['departure'];
    $totalCost = $_POST['totalCost'];

    $res = make_simple_resrv($pdo, $user_id, $room_id, $arrival, $departure, $totalCost);

    if ($res == 1) {
        // Redirect to a success page
        header('Location: reservation_success.php');
        exit();
    } else {
        // Redirect to an error page
        header('Location: reservation_error.php');
        exit();
    }
}

```

Reservation Successful!

Your reservation has been completed successfully. Thank you for booking with us.

[Return to Homepage](#)

Reservation Error

There was an issue with your reservation. Please try again later.

[Return to Homepage](#)

Η συνάρτηση που κάνει την προσθήκη βρίσκεται στο αρχείο “book_now_model.inc.php” και παίρνει ως ορίσματα όλα τα πεδία του αντίστοιχου πίνακα, εκτός των “ReservationID” και “DateOfReservation” που τίθονται αυτόματα.

```
function make_simple_resrv($pdo, $user_id, $room_id, $arrival, $departure, $totalCost) {
    // Insert the reservation into the makes_simple_resrv table
    $query = "INSERT INTO makes_simple_resrv (FullPrice, CheckIn, Checkout, FK1_CustomerID, FK2_RoomID)
              VALUES (:fullPrice, :checkIn, :checkOut, :fK1_CustomerID, :fK2_RoomID);";

    $stmt = $pdo->prepare($query);
    $stmt->bindParam(':fullPrice', $totalCost);
    $stmt->bindParam(':checkIn', $arrival);
    $stmt->bindParam(':checkOut', $departure);
    $stmt->bindParam(':fK1_CustomerID', $user_id);
    $stmt->bindParam(':fK2_RoomID', $room_id);

    // Execute the query
    if ($stmt->execute()) {
        return 1;
    } else {
        return -1;
    }
}
```

Services

Η τρίτη ενότητα της ιστοσελίδας μας αφορά τις υπηρεσίες που παρέχει το ξενοδοχείο. Αυτές διακρίνονται σε τρεις κατηγορίες: Activities, Wellness, Culinary Experiences.

Αρχική σελίδα – Προβολή Ενδεικτικών Υπηρεσιών

Πατώντας την επιλογή “Services” (από το header ή το footer), γίνεται ανακατεύθυνση στο pure php file “get_services.php”.

Κάθε υποκατηγορία έχει ένα πλήθος υπηρεσιών. Στην αρχική σελίδα, θέλουμε να αναδείξουμε τις τρεις κατηγορίες, προβάλλοντας τρεις εγγραφές από το καθένα. Έτσι αποθηκεύουμε κάθε τριάδα σε ένα διαφορετικό session variable και μεταφερόμαστε στη σελίδα “services.php”.

Η συναρτήσεις “get_some_activities()”, “get_some_wellness()”, “get_some_culinary ()” είναι παρόμοιες με τη μόνη διαφορά να είναι ο πίνακας στον οποίο γίνεται η αναζήτηση.

```
<li class="nav-item">
    <a class="nav-link" href="get_services.php">Services</a>
</li>
```

```
<?php

session_start(); // Ensure session is started

try {
    require_once "includes/dbh.inc.php"; // Connect to the database.
    require_once "includes/services/services_model.inc.php";

    $activities = get_some_activities($pdo);
    $_SESSION['activities'] = $activities;

    $wellness = get_some_wellness($pdo);
    $_SESSION['wellness'] = $wellness;

    $culinary = get_some_culinary($pdo);
    $_SESSION['culinary'] = $culinary;

    // Redirect to a new page with filter parameters
    header("Location: services.php");
    exit(); // Make sure to exit after redirect
} catch (PDOException $e) {
    die("Query failed: " . $e->getMessage());
}

?>
```

```
function get_some_activities($pdo) {
    // Step 1: Retrieve the first three ServiceID values from the activities table
    $sql1 = "
        SELECT ServiceID
        FROM activity
        ORDER BY ServiceID
        LIMIT 3
    ";

    $stmt1 = $pdo->prepare($sql1);
    $stmt1->execute();

    // Fetch ServiceIDs
    $serviceIds = $stmt1->fetchAll(PDO::FETCH_COLUMN, 0);

    if (empty($serviceIds)) {
        echo "No activities found.";
        exit();
    }

    // Step 2: Fetch ServiceName, Description, and Image from the service table using the ServiceIDs
    $placeholders = implode(',', array_fill(0, count($serviceIds), '?'));
    $sql2 = "
        SELECT ServiceName, Description, Image
        FROM service
        WHERE ServiceID IN ($placeholders)
    ";

    $stmt2 = $pdo->prepare($sql2);
    $stmt2->execute($serviceIds);

    // Fetch results
    $services = $stmt2->fetchAll(PDO::FETCH_ASSOC);

    return $services;
}
```

Η διαδικασία ανάκτησης των πλειάδων ανάγεται σε δύο στάδια: ένα για την απόκτηση του συνόλου που αποτελείται από τις τιμές του πεδίου “ServiceID” για τις πρώτες τρεις εγγραφές της αντίστοιχης κατηγορίας και ένα για την ανάκτηση των πεδίων “ServiceName”, “Description” και “Image” από τον πίνακα “Service”, την υπερκλάση.

Το πρώτο SQL ερώτημα χρησιμοποιεί “SELECT” για το πρωτεύον κλειδί της σχέσης, στον πίνακα της υποκλάσης (συγκεκριμένα στην εικόνα στο activity). Επιλέγουμε να ταξινομήσουμε με “ORDER BY” ως προς το κλειδί για να πάρουμε τις παλαιότες προσθήκες στη βάση, ενώ περιορίζουμε τον αριθμό των εγγραφών που θα λάβουμε σε τρία με τον τελεστή “LIMIT”.

Έπειτα, αναζητούμε στον πίνακα “Service” τα πεδία που αναφέρθηκαν με τη συνθήκη ότι οι εγγραφές θα πρέπει να ανήκουν στο σύνολο που ανακτήσαμε νωρίτερα, με το ServiceID δηλαδή να ανήκει στο σύνολο των επιθυμητών κλειδιών, το οποίο επιτυγχάνεται με τον τελεστή “IN”.

Έχοντας πλέον τα τρία σύνολα πλειάδων, μεταφερόμαστε στην αρχική σελίδα της ενότητας “Services”, όπου παρουσιάζονται οι τρεις κατηγορίες με αντιπροσωπευτικά παραδείγματα. Σε κάθε τμήμα, υπάρχει ένα κουμπί “See more”, ώστε ο χρήστης να δει όλο το πλήθος της αντίστοιχης κατηγορίας.

SERVICES

Activities



Indoor climbing

Experience the thrill of climbing right within our hotel! Our state-of-the-art indoor climbing facility offers an exhilarating experience for climbers of all skill levels. Whether you're a seasoned climber or a beginner, our climbing walls provide a range of routes and challenges.



Ice Skating

Glide across a frozen lake nestled beside a serene forest. Enjoy a magical ice skating experience surrounded by natural beauty. Perfect for families, couples, and anyone seeking winter fun in a breathtaking setting. Create unforgettable memories on the ice.



Ski/snowboarding

Hit the slopes for an exhilarating ski or snowboarding experience. Perfect for all skill levels, enjoy breathtaking mountain views and expertly groomed trails. Whether you're seeking thrills or a leisurely ride, our winter wonderland awaits.

[See more](#)

Wellness



Spa

Escape to our serene spa, where expert therapists offer personalized treatments, from soothing massages to revitalizing facials. Enjoy our tranquil atmosphere, luxury amenities, and a full range of wellness services designed to refresh your body and mind. Relax, rejuvenate, and unwind.



Facial Treatment

Refresh and rejuvenate your skin with our luxurious face treatments. Tailored to your skin's needs, our expert therapists use premium products to cleanse, hydrate, and revitalize your complexion. Leave with glowing, youthful, and radiant skin.



Sauna

Unwind in our soothing sauna, designed for ultimate relaxation. Enjoy the calming heat that helps relieve stress, relax muscles, and detoxify your body. Perfect for a rejuvenating experience after a day of activities or to simply unwind and refresh.

[See more](#)

Culinary Experiences



Private Dinner

Indulge in a bespoke dining experience with our private chef. Enjoy gourmet meals crafted to your taste, in the comfort of your suite or any location you choose. Perfect for intimate dinners, celebrations, or savoring exquisite cuisine without leaving your space.



Bar

Unwind at our stylish hotel bar, offering a curated selection of cocktails, fine wines, and craft beers. Enjoy a sophisticated ambiance, comfortable seating, and attentive service, perfect for socializing or relaxing after a day of activities.



Restaurant

Enjoy a varied dining experience with our diverse menu. Choose from à la carte dishes, hearty mains, and delectable desserts, all crafted from fresh, high-quality ingredients. Whether you're in the mood for a fine dining experience or casual fare, we have something for every taste.

[See more](#)

Σελίδα Υποκατηγορίας Υπηρεσιών – Προβολή Υποκατηγορίας

Το κουμπί “See more” οδηγεί στο αρχείο “services.in.php”, περνώντας μέσω της URL την κατηγορία που επέλεξε ο χρήστης.

```
<a href="includes/services/services.inc.php?services=Activities" class="btn btn-primary"
    style="padding: 10px 20px; background-color: #007bff; color: white; text-decoration: none;">
    See more
</a>
```

Σε αυτό αρχείο php θέλουμε να θέσουμε δύο session variables: ένα που δηλώνει τον επιλεγμένο τύπο της κατηγορίας και ένα σύνολο υπηρεσιών που ανακτούμε από τη βάση και θα προβληθεί στην ιστοσελίδα.

Το πρώτο τίθεται από τη παράμετρο της URL και με βάση αυτό καλείται η αντίστοιχη συνάρτηση για να γεμίσει με πλειάδες το σύνολο των προβαλόμενων υπηρεσιών.

Ομοίως με πριν, οι συναρτήσεις είναι ίδιες με εξαίρεση τον πίνακα της υποκατηγορίας που γίνεται η αναζήτηση.

```
try {
    require_once "../dbh.inc.php"; // Connect to the database.
    require_once "services_model.inc.php";

    $type = htmlspecialchars($_GET['services']);

    if ($type == "Wellness") {
        $services = get_all_wellness($pdo);
    }
    else if ($type == "Activities") {
        $services = get_all_activities($pdo);
    }
    else if ($type == "Culinary") {
        $services = get_all_culinary($pdo);
    }
    else {
        // Redirect to a main page -> ERROR
        header("Location: ../../main_page.php");
        exit();
    }

    // Store the data to a session variable
    $_SESSION['services'] = $services;
    $_SESSION['type'] = $type;

    // Redirect to a new page with filter parameters
    header("Location: ../../services_details.php");
    exit();
}

} catch (PDOException $e) {
    die("Query failed: " . $e->getMessage());
}
```

Αφού η εγγραφή ανήκει σε μία υποκλάση, για να πάρουμε όλα τα γνωρίσματά της πρέπει να πάρουμε όλα τα πεδία τόσο από τον πίνακα της υποκλάσης όσο και της υπερκλάσης. Παίρνοντας σαν βάση την υποκλάση, συνδυάζουμε όλες της εγγραφές της με τις αντίστοιχες του πίνακα “service” με βάση το πρωτεύον κλεδί και αυτό επιτυγχάνεται με τον τελεστή JOIN.

```
function get_all_activities($pdo) {
    // Step 1: Retrieve all records from the activity table and
    // their corresponding data from the service table
    $sql = "
        SELECT activity.*, service.*
        FROM activity
        JOIN service ON activity.ServiceID = service.ServiceID
    ";

    $stmt = $pdo->prepare($sql);
    $stmt->execute();

    // Fetch all combined activity and service records
    $activities = $stmt->fetchAll(PDO::FETCH_ASSOC);

    return $activities;
}
```

Σελίδα υποκατηγορίας - Προβολή υποκλάσης

Αφού ορίστηκαν οι session μεταβλητές για την κατηγορία υπηρεσιών και το σύνολο αυτών που θα προβληθούν, μεταφερόμαστε στη σελίδα "services_details.php", όπου παρουσιάζονται όλες οι λεπτομέρεις για όλες τις παροχές. Ανάλογα με τον επιλεγμένο τύπο, καλείται η αντίστοχη συνάρτηση στο αρχείο "services_view.inc.php" (πχ output_all_activities()).



Indoor climbing

Description: Experience the thrill of climbing right within our hotel! Our state-of-the-art indoor climbing facility offers an exhilarating experience for climbers of all skill levels. Whether you're a seasoned climber or a beginner our climbing walls provide a range of routes and challenges.

Difficulty Level: All levels available

Minimum Age: 5

Guide Required: 0

Duration: No limit

Price: 10

Availability Hours: 09:00 – 22:00

No booking required



Ski/Snowboarding

Description: Hit the slopes for an exhilarating ski or snowboarding experience. Perfect for all skill levels, enjoy breathtaking mountain views and expertly groomed trails. Whether you're seeking thrills or a leisurely ride, our winter wonderland awaits.

Difficulty Level: Moderate, difficult

Minimum Age: 5

Guide Required: 0

Duration: No limit

Price: 20

Availability Hours: 09:00 – 16:00

Number of people:

Select time of booking:

Book Now

Αν το πεδίο "BookingRequired" έχει τεθεί ως true, εμφανίζεται μία φόρμα κράτησης. Αποτελείται από δύο hidden πεδία: το service_id και το type, τα οποία τίθονται αυτόματα από το γνώρισμα της αντίστοιχης πλειάδας και από το session variable. Επιπλέον, υπάρχουν άλλα δύο πεδία τα οποία συμπληρώνονται από το χρήστη: το num_people, που δίνεται ως ένα dropdown menu με επιλογές από το 1 έως το 10 και το appointed_time που είναι πεδίο τύπου date.

```
if ($activity['BookingRequired'] == 1) {  
    // Start form  
    echo "<form action='book_service.php' method='POST'>";  
    echo "<input type='hidden' name='service_id' value='".$htmlspecialchars($activity['ServiceID'])."' />";  
    echo "<input type='hidden' name='type' value='".$htmlspecialchars($_SESSION['type'])."' />";  
  
    echo "<label for='num_people'>Number of people: </label>";  
    echo "<select name='num_people' id='num_people'>";  
    echo "<option value='1'>1</option>";  
    echo "<option value='2'>2</option>";  
    echo "<option value='3'>3</option>";  
    echo "<option value='4'>4</option>";  
    echo "<option value='5'>5</option>";  
    echo "<option value='6'>6</option>";  
    echo "<option value='7'>7</option>";  
    echo "<option value='8'>8</option>";  
    echo "<option value='9'>9</option>";  
    echo "<option value='10'>10</option>";  
    echo "</select>";  
  
    echo "<br>";  
    echo "Select time of booking: ";  
    echo "<input type='time' name='appointed_time' step='1800' />";  
  
    echo "<div style='text-align: left; margin-top: 10px;'>";  
    echo "<button type='submit' class='btn btn-primary' style='padding: 10px 20px; background-color: #007bff; color: white; text-decoration: none;'>";  
    echo "Book Now";  
    echo "</button>";  
    echo "</div>";  
  
    echo "</form>"; // End form
```

Ραντεβού για υπηρεσία – Εισαγωγή στον πίνακα book_service

Πατώντας το κουμπί “Book Now”, ο χρήστης υποβάλει τη φόρμα και μεταφέρεται στη σελίδα “book_service.php”. Πριν το HTML section, εκτελούμε το παρακάτω κομμάτι κώδικα php. Για να υπάρχει πρόσβαση σε αυτή τη σελίδα θα πρέπει να έχει υποβληθεί με έγκυρο τρόπο η αντίστοιχη φόρμα, ώστε να υπάρχει πρόσβαση στα επιθυμητά δεδομένα. Αποθηκεύουμε τις τιμές από τα πεδία της φόρμας σε session variables και καλούμε τη συνάρτηση get_service(), για να πάρουμε ολόκληρη την επιθυμητή εγγραφή, την οποία αποθηκεύουμε επίσης ως session variable.

```
// Ensure the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Retrieve the form data
    $service_id = isset($_POST['service_id']) ? $_POST['service_id'] : null;
    $type = isset($_POST['type']) ? $_POST['type'] : null;
    $num_people = isset($_POST['num_people']) ? intval($_POST['num_people']) : 0;
    $appointed_time = isset($_POST['appointed_time']) ? $_POST['appointed_time'] : null;

    if ($service_id == null || $type == null || $num_people == 0 || $appointed_time == null) {
        // ERROR
        header('Location: ../../services_details.php');
        exit();
    }

    $_SESSION["service_id"] = $service_id;
    $_SESSION["type"] = $type;
    $_SESSION["num_people"] = $num_people;
    $_SESSION["appointed_time"] = $appointed_time;
    $username = isset($_SESSION['user_username']) ? $_SESSION['user_username'] : null;

    $curr_service = get_service($pdo, $service_id, $type);
    $_SESSION["curr_service"] = $curr_service;
} else {
    // ERROR
    header('Location: ../../services_details.php');
    exit();
}
```

Ανάλογα με τη τιμή του \$type αναζητάμε στον αντίστοιχο πίνακα της υποκλάσης. Θέλουμε να ανακτήσουμε όλα τα γνωρίσματα της εγγραφής επομένως και από τους δύο πίνακες, υπερκλάσης και υποκλάσης. Ακολουθούμε την ίδια λογική με προηγουμένως, με τον επιπλέον περιορισμό ότι ψάχνουμε για συγκεκριμένη τιμή στο πρωτεύον κλειδί και επομένως θα ανακτήσουμε ακριβώς μία εγγραφή.

Όπως είχε γίνει και για την κράτηση δωματίου, έτσι και εδώ προβάλλονται οι λεπτομέρειες τις κράτησης, μέσω της συνάρτησης output_booking_service() (αρχείο “services_view.inc.php”):

```
function get_service($pdo, $service_id, $type) {
    if ($type == "Activities") {
        $sql = "
            SELECT activity.*, service.*
            FROM activity
            JOIN service ON activity.ServiceID = service.ServiceID
            WHERE activity.ServiceID = :service_id
        ";
    } else if ($type == "Wellness") {
        $sql = "
            SELECT wellness.*, service.*
            FROM wellness
            JOIN service ON wellness.ServiceID = service.ServiceID
            WHERE wellness.ServiceID = :service_id
        ";
    } else if ($type == "Culinary") {
        $sql = "
            SELECT culinary_experience.*, service.*
            FROM culinary_experience
            JOIN service ON culinary_experience.ServiceID = service.ServiceID
            WHERE culinary_experience.ServiceID = :service_id
        ";
    }

    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':service_id', $service_id, PDO::PARAM_INT);
    $stmt->execute();

    $service = $stmt->fetch(PDO::FETCH_ASSOC);

    return $service;
}
```



Ski/Snowboarding

Appointed Time: 10:00

Price Per Person: \$20

Number of People: 2

Total cost: \$40

[Book Now](#)

Η το κουμπί υποβάλει την τελική φόρμα για την κράτηση της υπηρεσίας με μόνο ένα hidden input το full_price, το οποίο υπολογίζεται με όπως φαίνεται στην εικόνα.

```
echo '<form action="includes/services/book_service.inc.php" method="POST">';
echo '<input type="hidden" name="full_price" value="" . ($num_people * $curr_service['Price']) . ">';
echo '<button type="submit" class="btn btn-primary">Book Now</button>';
echo '</form>';
```

Τα υπόλοιπα στοιχεία που απαιτούνται είναι ήδη αποθηκευμένα σε session variables επομένως έχουμε πρόσβαση. Έτσι, μεταφερόμαστε στο αρχείο “book_service.inc.php”, όπου με όμοιο τρόπο, διαχειρίζομαστε την κράτηση και μεταφερόμαστε στην αντίστοιχη σελίδα επιτυχίας ή αποτυχίας.

```
// Submit Reservation
// Check if the form was submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the data from the form
    $full_price = $_POST['full_price'];

    $service_id = $_SESSION["service_id"];
    $num_people = $_SESSION["num_people"];
    $appointed_time = $_SESSION["appointed_time"];

    $user_id = $_SESSION["user_id"];

    $res = book_service($pdo, $user_id, $service_id, $num_people, $appointed_time, $full_price);

    if ($res == 1) {
        // Redirect to a success page
        header('Location: ../book_now/reservation_success.php');
        exit();
    } else {
        // Redirect to an error page
        header('Location: ../book_now/reservation_error.php');
        exit();
    }
}
```

Παρακάτω δίνεται η συνάρτηση book_service(), η οποία με χρήση του “INSERT” προσθέτει μία νέα εγγραφή στον πίνακα “book_service”, δίνοντας τις τιμές των γνωρισμάτων, με εξαίρεση των “BookingID” και “DateOfBooking” που συμπληρώνονται αυτόματα.

```
function book_service($pdo, $user_id, $service_id, $num_people, $appointed_time, $full_price) {

    $query = "INSERT INTO book_service (AppointedTime, NumOfPeople, FullPrice, FK1_CustomerID, FK2_ServiceID)
              VALUES (:appointed_time, :num_people, :full_price, :fk1_CustomerID, :fk2_ServiceID);";

    $stmt = $pdo->prepare($query);
    $stmt->bindParam(':appointed_time', $appointed_time);
    $stmt->bindParam(':num_people', $num_people);
    $stmt->bindParam(':full_price', $full_price);
    $stmt->bindParam(':fk1_CustomerID', $user_id);
    $stmt->bindParam(':fk2_ServiceID', $service_id);

    // Execute the query
    if ($stmt->execute()) {
        return 1;
    } else {
        return -1;
    }
}
```

Special Offers

Η τελευταία ενότητα αφορά τα πακέτα προσφορών. Το ξενοδοχείο προσφέρει μια έκπτωση στο επιλεγμένο δωμάτιο και ελέυθερη είσοδο σε κάποια υπηρεσία. Κάθε πακέτο είναι συγκεκριμένο και ισχύει για κάποια δωμάτια, με κάποια δωμάτια να ανήκουν σε περισσότερα από ένα πακέτα ή και σε κανένα. Ο χρήστης μπορεί να κάνει κράτηση για ένα ή περισσότερα πακέτα.

Αρχική Σελίδα – Μπάρα Αναζήτησης Πακέτων

Επιλέγοντας την ενότητα “Special Offers” εμφανίζεται η παρακάτω σελίδα, που αποτελείται από μία μπάρα αναζήτησης με τα κουμπιά “Apply Filters” και “Clear Filters” και ένα πλήθος από πακέτα του ξενοδοχείου για τα οποία βλέπουμε μια εικόνα, την προβλεπόμενη έκπτωση και μια περιγραφή του πακέτου.

SPECIAL OFFERS

Search...

[Apply Filters](#) [Clear Filters](#)



Discount: 25%

Enjoy a romantic getaway with our special couples offer. Stay in a beautifully appointed room and indulge in unlimited spa sessions, designed to relax and rejuvenate. Perfect for couples seeking a luxurious escape.

[See More](#)



Discount: 25%

Enjoy a cozy stay for four and take advantage of free ski passes for the entire family. Book now and embrace the ultimate winter adventure!

[See More](#)



Discount: 25%

Enjoy a free pass to our buffet with your stay! Savor a variety of delicious dishes, from breakfast to dinner, at no extra cost. Perfect for food lovers seeking a culinary adventure during their visit. Book now to treat yourself to endless dining delights!

[See More](#)

Όπως και προηγούμενως, όταν επιλέγεται η ενότητα από το header ή το footer, πηγαίνουμε πρώτα στο αρχείο “get_all_offers.php”, προτού πάμε στη σελίδα “special_offers.php”, για να εκτελέσουμε το query με το οποίο θα ανακτήσουμε όλες τις εγγραφές του πίνακα special_offers, το οποίο αποθηκεύεται ως \$_SESSION['special_offers'].

```
<li class="nav-item">
| <a class="nav-link" href="get_all_offers.php">Special Offers</a>
</li>

function get_all_offers($pdo) {
    $query = "SELECT * FROM special_offer";
    $stmt = $pdo->prepare($query);
    $stmt->execute();

    $offers = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $offers;
}
```

Η μπάρα αναζήτησης αποτελεί το φίλτρο με βάση το οποίο ο χρήστης μπορεί να περιορίσει τα πακέτα που του εμφανίζονται. Όπως και στην ενότητα “Our Rooms” δημιουργούμε μια φόρμα μέσω της οποία μεταφέρουμε τα δεδομένα στην επόμενη σελίδα, ενώ αν έχει εφαρμοστεί ήδη το φίλτρο, η μπάρα θα κρατήσει τα δεδομένα, παίρνοντας τη τιμή από την παράμετρο στη URL.

```
<form method="GET" action="includes/offers/filter_offers.inc.php">
    <div class="row">
        <div class="col-md-3">
            <label for="search" class="form-label"></label>
            <input type="text" name="search" id="search" class="form-control" placeholder="Search..." value="<?php echo isset($_GET['search']) ? htmlspecialchars($_GET['search']) : null; ?>">
        </div>
    </div>
    <button type="submit" class="btn btn-primary mt-3">Apply Filters</button>
    <button type="reset" class="btn btn-secondary mt-3" onclick="window.location.href='includes/offers/filter_offers.inc.php';">Clear Filters</button>
</form>
```

Υποβάλλοντας τη φόρμα, μεταφερόμαστε στο αρχείο “filter_offers.inc.php”, όπου περνάμε σαν όρισμα την είσοδο του χρήστη στη συνάρτηση `get_search_offers()` και ανακτούμε τις εγγραφές που ικανοποιούν αυτό το φίλτρο, αποθηκεύοντας τα στο αντίστοιχο session variable. Τέλος, επιστρέφουμε στην προηγούμενη σελίδα, με τη τιμή του search ως παράμετρο στη URL.

```
<?php
session_start(); // Ensure session is started

// Retrieve form data
$search = isset($_GET['search']) ? htmlspecialchars($_GET['search']) : null;

try {
    require_once "../dbh.inc.php"; // Connect to the database.
    require_once "offers_model.inc.php";

    // Call the function to get available rooms
    $search_offers = get_search_offers($pdo, $search);

    if (!empty($search_offers)) {
        // Store the available rooms in the session
        $_SESSION['special_offers'] = $search_offers;
    }

    // Redirect to a new page with filter parameters
    header("Location: ../../special_offers.php?search=" . urlencode($search));
    exit(); // Make sure to exit after redirect
} catch (PDOException $e) {
    die("Query failed: " . $e->getMessage());
}
?>
```

Το query που χρησιμοποιούμε χρησιμοποιεί τον τελεστή “LIKE”, αναζητώντας στον γνώρισμα Description, την ορισμένη από το χρήστη συμβολοσειρά.

```
function get_search_offers($pdo, $search) {
    // Prepare the SQL query with a placeholder for the search term
    $sql = "SELECT * FROM special_offer WHERE Description LIKE :search";

    // Prepare the statement
    $stmt = $pdo->prepare($sql);

    // Bind the search term with wildcards for partial matching
    $stmt->bindValue(':search', '%' . $search . '%');

    // Execute the query
    $stmt->execute();

    // Fetch all the results
    $offers = $stmt->fetchAll(PDO::FETCH_ASSOC);

    // Return the results
    return $offers;
}
```

Παρατηρούμε πως αν ο χρήστης πληκτρολογήσει τη λέξη "ski", θα εμφανιστούν μόνο δύο πακέτα, που περιέχουν τη λέξη αυτή στην περιγραφή τους.

SPECIAL OFFERS

ski

Apply Filters Clear Filters



Discount: 25%

Enjoy a cozy stay for four and take advantage of free ski passes for the entire family. Book now and embrace the ultimate winter adventure!

[See More](#)



Discount: 30%

Get a generous discount and free ski passes for two! Perfect for couples or friends looking to hit the slopes together without the extra cost. Embrace the thrill of skiing and make the most of your winter getaway.

[See More](#)

Επιλογή δωματίων από πακέτο – Διαχείριση τριμερής σχέσης

Κάτω από την περιγραφή κάθε πακέτου υπάρχει ένα κουμπί "See More", μέσω του οποίου ο χρήστης μπορεί να επιλέξει το πακέτο και να δει τα διαθέσιμα δωμάτια για τα οποία ισχύει.

Όπως και προηγουμένως, το κουμπί υποβάλλει μία φόρμα που μας μεταφέρει στο αρχείο "offers.inc.php". Με το "offer_id" να λαμβάνεται ως hidden input, ανακτούμε από τον πίνακα "special_offer" την εγγραφή με αυτό το πρωτεύον κλειδί.

```
<?php  
  
session_start(); // Ensure session is started  
  
try {  
    require_once "../dbh.inc.php"; // Connect to the database.  
    require_once "offers_model.inc.php";  
  
    $offer_id = htmlspecialchars($_GET['offer_id']);  
  
    $special_offer = get_special_offer($pdo, $offer_id);  
    $offer_rooms = get_offer_rooms($pdo, $offer_id);  
    $service_name = get_service_name($pdo, $special_offer['ServiceID']);  
  
    $_SESSION['special_offer'] = $special_offer;  
    $_SESSION['offer_rooms'] = $offer_rooms;  
    $_SESSION['service_name'] = $service_name;  
  
    // Redirect to a new page with filter parameters  
    header("Location: ../../offer_page.php");  
    exit(); // Make sure to exit after redirect  
  
} catch (PDOException $e) {  
    die("Query failed: " . $e->getMessage());  
}
```

```
function get_special_offer($pdo, $offer_id) {  
    $query = "SELECT * FROM special_offer WHERE SpecialOfferID = :offer_id";  
    $stmt = $pdo->prepare($query);  
    $stmt->bindParam(':offer_id', $offer_id, PDO::PARAM_INT);  
    $stmt->execute();  
  
    $special_offer = $stmt->fetch(PDO::FETCH_ASSOC);  
    return $special_offer;  
}
```

Στη συνέχεια, αναζητώ όλα τα RoomID's στον πίνακα "combination" που συνδέονται με το κλειδί του επιλεγμένου πακέτου. Έχοντας δημιουργήσει αυτό το σύνολο, ανακτώ τις εγγραφές από τον στο πίνακα "room", των οποίων τα κλειδά υπάρχουν στο σύνολο που δημιουργήθηκε, με τη χρήση του τελεστή "IN".

Τέλος, καλείται η συνάρτηση `get_service_name()`, ώστε να έχουμε σε πρόσβαση το όνομα της υπηρεσίας που καλύπτει το επιλεγμένο πακέτο, αφού για την ώρα είχαμε μόνο το πρωτεύον κλειδί της.

```
function get_offer_rooms($pdo, $offer_id) {
    // Step 1: Get RoomIDs from the combination table where SpecialOfferID matches $offer_id
    $sql1 = "
        SELECT RoomID
        FROM combination
        WHERE SpecialOfferID = ?
    ";
    $stmt1 = $pdo->prepare($sql1);
    $stmt1->execute([$offer_id]);

    // Fetch RoomIDs
    $roomIds = $stmt1->fetchAll(PDO::FETCH_COLUMN, 0);

    if (empty($roomIds)) {
        echo "No rooms found for the given offer.";
        return [];
    }

    // Step 2: Get details of rooms from the room table using the fetched RoomIDs
    $placeholders = implode(',', array_fill(0, count($roomIds), '?'));
    $sql2 = "
        SELECT *
        FROM room
        WHERE RoomID IN ($placeholders)
    ";
    $stmt2 = $pdo->prepare($sql2);
    $stmt2->execute($roomIds);

    // Fetch and return room details
    $rooms = $stmt2->fetchAll(PDO::FETCH_ASSOC);

    return $rooms;
}
```

```
function get_service_name($pdo, $service_id) {
    $query = "SELECT ServiceName FROM service WHERE ServiceID = :service_id";
    $stmt = $pdo->prepare($query);
    $stmt->bindParam(':service_id', $service_id, PDO::PARAM_INT);
    $stmt->execute();

    $result = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($result) {
        return $result['ServiceName'];
    } else {
        return null; // or handle the case where the service is not found
    }
}
```

Τελικά, καταλήγουμε στην παρακάτω σελίδα. Με το πρώτο τμήμα της να είναι τα φίλτρα των δωματίων, η περιγραφή του special_offer και στη συνέχεια οι λεπτομέρειες κάθε δωματίου, που είναι δηλωμένα σε αυτή την προσφορά.

Arrival Date: mm/dd/yyyy
Departure Date: mm/dd/yyyy
Room Type: Any
Number of Beds: Any
Capacity: Any
Order by Price: Any

SPECIAL OFFERS

Enjoy a romantic getaway with our special couples offer. Stay in a beautifully appointed room and indulge in unlimited spa sessions, designed to relax and rejuvenate. Perfect for couples seeking a luxurious escape.

 Love Suite Room Type: Double Number Of Beds: 1 Capacity: 2 Book Now	 Love For two Room Type: Double Number Of Beds: 1 Capacity: 2 Hot Tub Included Book Now
--	---

Η σελίδα αυτή λειτουργεί με παρόμοιο τρόπο με την αντίστοιχη στην ενότητα “Our Rooms”, με μία διαφοροποίηση στη συνάρτηση `get_offer_available_rooms()` (αρχείο `offers_model.inc.php`).

Αρχικά, αναζητάμε τα RoomID's των δωματίων που συνδέονται με την επιλεγμένη προσφορά όπως και πριν. Αν ο πίνακας αυτός είναι άδειος τότε και το αποτέλεσμα δεν επιστρέφουμε καμία εγγραφή.

Επομένως από τον πίνακα “room” κρατάμε όλοκληρες τις εγγραφές, μόνο αν τα αντίστοιχα κλειδιά υπάρχουν στο σύνολο που ανακτήσαμε προηγουμένως. Τα queries των φίλτρων λειτουργούν με ακριβώς τον ίδιο τρόπο.

```
// Step 1: Find all RoomIDs associated with the given SpecialOfferID
$query = "
  SELECT RoomID
  FROM combination
  WHERE SpecialOfferID = :offer_id
";

$stmt = $pdo->prepare($query);
$stmt->bindParam(':offer_id', $offer_id);
$stmt->execute();

$offerRooms = $stmt->fetchAll(PDO::FETCH_COLUMN, 0); // Fetch only the room IDs

if (empty($offerRooms)) {
    return $offerRooms;
}

// Step 2: Build the base query
$query = "SELECT * FROM room";
$conditions = [];

// Filter to include only rooms associated with the special offer
$placeholders = [];
foreach ($offerRooms as $index => $roomID) {
    $placeholder = ":offerRoom_$index";
    $placeholders[] = $placeholder;
    $params[$placeholder] = $roomID;
}
$conditions[] = "RoomID IN (" . implode(', ', $placeholders) . ")";
```

Κράτηση Δωματίου με χρήση πακέτου – Εισαγωγή στο “make_special_resrv”

Πατώντας το κουμπί “Book Now”, υποβάλλεται η παρακάτω φόρμα:

```
echo '<form action="book_offer.php" method="POST">';
echo '<input type="hidden" name="room_id" value="' . htmlspecialchars($room['RoomID']) . '">';
echo '<input type="hidden" name="arrival" value="' . htmlspecialchars($arrival) . '">';
echo '<input type="hidden" name="departure" value="' . htmlspecialchars($departure) . '">';
echo '<button type="submit" class="btn btn-primary">Book Now</button>';
echo '</form>';
```

Επομένως, έχοντας τις επιλεγμένες ημερομηνίες διαμονής και το κλειδί του επιλεγμένου δωματίου, μπορούμε να περάσουμε στην επόμενη και τελική σελίδα που προβάλλει τις λεπτομέρεις της κράτησης.

```
// Submit Reservation
// Check if the form was submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $arrival = $_POST['arrival'];
    $departure = $_POST['departure'];
    $room_id = $_POST['room_id'];

    if (empty($arrival) || empty($departure) || empty($room_id)) {
        echo "One or more fields are empty.";
        header('Location: offer_page.php');
        exit();
    }

    if (!isset($_SESSION['special_offer']) || !isset($_SESSION['service_name'])) {
        // ERROR!!!
        echo "Something went wrong";
        header('Location: ../../offer_page.php');
        exit();
    }

    $special_offer = $_SESSION['special_offer'];
    $service_name = $_SESSION['service_name'];
    $user_id = $_SESSION['user_id'];
    $username = isset($_SESSION['user_username']) ? $_SESSION['user_username'] : null;

    $curr_room = get_room($pdo, $room_id);
}
```

Η σελίδα έχει αυτή τη μορφή:



Love For two

Room Type: Double

Number Of Beds: 1

Capacity: 2

Hot Tub Included

Check-in: 2024-09-15

Check-out: 2024-09-21

Included service: Spa

Price Per Night: \$50

Stay Duration: 6 nights

Offer Discount: 25

Total cost: \$225

Book Now

Με το κουμπί να υποβάλλει την παρακάτω φόρμα κράτησης:

```
echo '<form action="includes/offers/make_special_resrv.inc.php" method="POST">';
echo '<input type="hidden" name="room_id" value="' . htmlspecialchars($curr_room['RoomID']) . '">';
echo '<input type="hidden" name="user_id" value="' . htmlspecialchars($user_id) . '">';
echo '<input type="hidden" name="offer_id" value="' . htmlspecialchars($special_offer['SpecialOfferID']) . '">';
echo '<input type="hidden" name="arrival" value="' . htmlspecialchars($arrival) . '">';
echo '<input type="hidden" name="departure" value="' . htmlspecialchars($departure) . '">';
echo '<input type="hidden" name="totalCost" value="' . $total_cost . '">';
echo '<button type="submit" class="btn btn-primary">Book Now</button>';
echo '</form>';
```

Όλα τα πεδία είναι hidden καθώς οι τιμές τους έχουν τεθεί από προηγούμενες σελίδες, εκτός από το totalCost το οποίο υπολογίστηκε παραπάνω, λαμβάνοντας υπόψη τις ημέρες διαμονής, το κόστος του δωματίου και την έκπτωση:

```
$total_cost = $stayDuration * $curr_room['PricePerNight'] -  
| | | | | $stayDuration * $curr_room['PricePerNight'] * htmlspecialchars($special_offer['Discount'])/100;
```

Στο αρχείο “make_special_resrv.inc.php”, με προϋπόθεση ότι τα απαραίτητα πεδία είναι συμπληρωμένα, καλείται η make_special_resrv() (αρχείο offers_model.inc.php).

Κάνοντας “**INSERT**”, εισάγουμε μία νέα εγγραφή στο πίνακα “**make_special_resrv**”, δίνοντας τις αντίστοιχες τιμές για όλα τα γνωρίσματα εκτός των “**ReservationID**” και “**DateOfReservation**” που τίθονται αυτόματα.

Με την ολοκλήρωση της κράτησης μεταφερόμαστε στην αντίστοιχη σελίδα επιτυχίας ή αποτυχίας.

```
function make_special_resrv($pdo, $user_id, $room_id, $offer_id, $arrival, $departure, $totalCost) {
    // Insert the reservation into the makes_special_resrv table
    $query = "
        INSERT INTO makes_special_resrv
        (FullPrice, CheckIn, Checkout, FK1_CustomerID, FK2_RoomID, FK3_SpecialOfferID)
        VALUES
        (:fullPrice, :checkIn, :checkOut, :fk1_CustomerID, :fk2_RoomID, :fk3_SpecialOfferID);
    ";

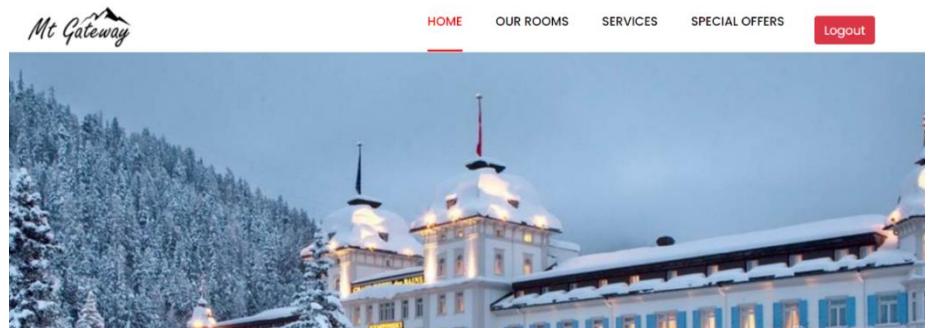
    $stmt = $pdo->prepare($query);

    // Bind parameters to the SQL query
    $stmt->bindParam(':fullPrice', $totalCost);
    $stmt->bindParam(':checkIn', $arrival);
    $stmt->bindParam(':checkOut', $departure);
    $stmt->bindParam(':fk1_CustomerID', $user_id);
    $stmt->bindParam(':fk2_RoomID', $room_id);
    $stmt->bindParam(':fk3_SpecialOfferID', $offer_id);

    // Execute the query
    if ($stmt->execute()) {
        return 1; // Success
    } else {
        return -1; // Failure
    }
}
```

Website – Administrator Side

Μόλις εισάγουμε το Username/Password **admin/admin** στη φόρμα login του admin, μεταφερόμαστε στην πλευρά της ιστοσελίδας διαχειριστή. Αυτή η πλευρά είναι τελείως διαχωρισμένη από την πλευρά του χρήστη. Η **Main Page/Home** του admin δεν έχει κάποια ειδική λειτουργικότητα, απλά είναι η αρχική σελίδα μετά από το index. Μέσω της μπάρας στο πάνω μέρος της ιστοσελίδας, μεταφερόμαστε στις λειτουργικές σελίδες του διαχειριστή.



Admin – Our Rooms

Η πρώτη σημαντική σελίδα του διαχειριστή. Το περιεχόμενό της αποτελείται από 3 λειτουργίες του χρήστη και 1 λειτουργία της σελίδας. Ο χρήστης έχει την δυνατότητα να κάνει εισαγωγή στη βάση(πίνακα room) νέου δωματίου(Add New Room), να αλλάξει τα χαρακτηριστικά και την περιγραφή ενός υπάρχοντος δωματίου(Edit) και να διαγράψει ένα δωμάτιο από την ιστοσελίδα, αφαιρώντας αυτό και τις διασυνδέσεις του(π.χ. booking) από την βάση δεδομένων (Delete). Η λειτουργία από πλευρά εφαρμογής είναι η δυναμική εμφάνιση στον διαχειριστή όλων των δωματίων που περιλαμβάνονται στην βάση δεδομένων.

A screenshot of the "Our Rooms" section of the website. The top navigation bar includes "HOME", "OUR ROOM" (which is highlighted in red), "SERVICES", "SPECIAL OFFERS", and "Logout". Below the navigation is a dark banner with the word "ROOMS" in white. The main content area displays three room categories as cards: "Royal Life" (Queen bed, 1 bed, capacity 2, hot tub included, price 30€), "Rock&Roll" (Double bed, 2 beds, capacity 2, hot tub included, price 80€), and "Peace and Quiet" (Single bed, 1 bed, capacity 1, hot tub included, price 35€). Each card has "Edit" and "Delete" buttons at the bottom. At the bottom of the section is a blue "Add New Room" button.

Παράδειγμα σελίδας δωματίων με 3 δωμάτια

Προβολή Δωματίων

Η μέθοδος που χρησιμοποιείται για την εμφάνιση του κάθε δωματίου και των χαρακτηριστικών του στον administrator, είναι αντίστοιχη σε μορφή με αυτή που χρησιμοποιείται για την εμφάνιση του στον επισκέπτη της σελίδας και μελλοντικό πελάτη. Μόνη διαφορά είναι ότι το κουμπί για να κλείσεις το δωμάτιο έχει αντικατασταθεί από το Edit και το Delete.

```
function display_admin_rooms($pdo): void {
    require_once "includes/dbh.inc.php";
    $rooms = get_admin_rooms($pdo);
    if(empty($rooms)) {
        echo 'No available rooms.';
    }
    else {
        // echo '<div class = "our_room">';
        echo '<div class="row">'; // Open row with rooms
        foreach ($rooms as $room) {
            echo '<div class="col-md-4 col-sm-6">';
            echo '<div id="serv_hover" class="room">';

            // Display the room image wrapped in an anchor tag
            echo '<div class="room_img">';
            if (!empty($room['Image'])) {
                // use the relative path stored in the database
                $imageSrc = htmlspecialchars($room['Image']);
                echo "<figure><img src='".$imageSrc . "' alt='Room Image' /></figure>";
            } else {
                echo '<p>No image available.</p>';
            }
            echo '</div>'; // Close .room
            echo '</div>'; // Close .col-md-4
        }
        echo '</div>'; // Close .row
    }
}
```

Συνάρτηση προβολής δωματίων

Προσθήκη Δωματίου

Όταν ο admin πατήσει το κουμπί για εισαγωγή δωματίου, μία modal φόρμα θα εμφανιστεί επικαλύπτοντας την σελίδα των δωματίων. Μέσα σε αυτή δίνονται κενά για να συμπληρωθούν τα περιεχόμενα/χαρακτηριστικά του δωματίου. Κανένα από τα κενά, καθώς και η φωτογραφία, δεν μπορούν να μην εισαχθούν για τη επιτυχή δημιουργία του δωματίου.

Φόρμα δημιουργίας δωματίου

Μόλις γίνει επιτυχημένο submission των χαρακτηριστικών του νέου δωματίου, τα δεδομένα που εισήγαγε ο χρήστης μεταφέρονται μέσω **POST** στο αρχείο **add_room.inc.php**. Εκεί πραγματοποιείται έλεγχος για την εισαγωγή συμβατής φωτογραφίας από τον χρήστη και στην περίπτωση αποτυχίας, το δωμάτιο δεν δημιουργείται. Σε περίπτωση επιτυχίας(σωστού αρχείου), τότε περνάμε στη σύνδεση με τη βάση δεδομένων και στο αναγκαίο query για την **εισαγωγή** του δωματίου ως νέα εγγραφή στον πίνακα **room**. Αυτό είναι το **insert_room_to_table**.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $roomname = $_POST["room_name"];  
    $pricepernight = $_POST["price_per_night"];  
    $numofbeds = $_POST["num_of_beds"];  
    $roomtype = $_POST["room_type"];  
    $hashottub = isset($_POST["has_hot_tub"]) ? 1 : 0;  
    $capacity = $_POST["capacity"];  
    $image = $_FILES["image"]["tmp_name"];  
}
```

Λήψη συμπληρωμένων δεδομένων από την φόρμα

```
function insert_room_to_table($pdo, $RoomName, $PricePerNight, $NumOfBeds, $RoomType, $HasHotTub, $Capacity, $imgContent): void {  
    // Insert the room into the database  
    $sql = "INSERT INTO room (RoomName, PricePerNight, NumOfBeds, RoomType, HasHotTub, Capacity, Image)  
           VALUES (:RoomName, :PricePerNight, :NumOfBeds, :RoomType, :HasHotTub, :Capacity, :Image)";  
    $stmt = $pdo->prepare($sql);  
    $stmt -> bindParam(":RoomName", $RoomName);  
    $stmt -> bindParam(":PricePerNight", $PricePerNight);  
    $stmt -> bindParam(":NumOfBeds", $NumOfBeds);  
    $stmt -> bindParam(":RoomType", $RoomType);  
    $stmt -> bindParam(":HasHotTub", $HasHotTub);  
    $stmt -> bindParam(":Capacity", $Capacity);  
    $stmt->bindParam(":Image", $imgContent); // Ensure it's treated as a BLOB  
    $stmt -> execute();  
}
```

Query εισαγωγής δωματίου στο "room"

Επεξεργασία Δωματίου

Όταν σε ένα δωμάτιο πατάμε το κουμπί **Edit**, εμφανίζεται μία modal φόρμα που να έχει ακριβώς την ίδια μορφή με αυτή της προσθήκης. Βέβαια, αυτή τη φορά τα στοιχεία είναι ήδη συμπληρωμένα, εκτός από το αρχείο της φωτογραφίας. Αυτό επιτυγχάνεται με ένα Query εντός του view αρχείου που πριν εκτυπώσει τις πληροφορίες ενός δωματίου, παίρνει και αποθηκεύει σε μεταβλητές όλες τις πληροφορίες για αυτό από τον πίνακα της βάσης.

Ο διαχειριστής έχει την δυνατότητα να αλλάξει οποιδήποτε γνώρισμα του δωματίου. Όταν πατήσει **Save**, η φόρμα κάνει **POST** τα δεδομένα που εισήγαγε ο χρήστης, καθώς και το **ID** δωματίου στο οποίο δούλεψε, ως κρυφό γνώρισμα στο αρχείο **edit_room.inc.php**.

Δεχόμαστε τα δεδομένα. Αν ο χρήστης έχει ανεβάσει καινούργια φωτογραφία τότε το **imagepath** της προς τον φάκελο **images** εισάγεται, όπως γίνεται και στην δημιουργία του δωματίου. Αλλιώς, λαμβάνεται από την βάση με το αντίστοιχο query το

```
// Handle image upload if a new image is uploaded
if (!empty($_FILES['image']['tmp_name'])) {
    $imagePath = 'images/' . $_FILES['image']['name'];
} else {
    $sql = 'SELECT Image FROM room WHERE RoomID = :RoomID';
    $stmt = $pdo->prepare(query: $sql);
    $stmt->bindParam(param: ':RoomID', var: &$roomid, type: PDO::PARAM_INT);
    $stmt->execute();

    $imagePath = $stmt->fetchColumn();
}
```

Διαχείριση φωτογραφίας

imagepath που οδηγεί στην υπάρχουσα φωτογραφία του δωματίου. Τα δεδομένα εισάγονται στην συνάρτηση **edit_room**. Αυτή ενημερώνει με τις νέες τιμές τα γνωρίσματα του δωματίου.

```
function edit_room($pdo, $RoomID, $RoomName, $PricePerNight, $NumOfBeds, $RoomType, $HasHotTub, $Capacity, $ImgContent): void {
    $sql = "UPDATE room
        SET RoomName = :RoomName, PricePerNight = :PricePerNight, NumOfBeds = :NumOfBeds,
            RoomType = :RoomType, HasHotTub = :HasHotTub, Capacity = :Capacity, Image = :ImgContent
        WHERE RoomID = $RoomID";

    $stmt = $pdo->prepare($sql);

    $stmt -> bindParam(":RoomName", $RoomName);
    $stmt -> bindParam(":PricePerNight", $PricePerNight);
    $stmt -> bindParam(":NumOfBeds", $NumOfBeds);
    $stmt -> bindParam(":RoomType", $RoomType);
    $stmt -> bindParam(":HasHotTub", $HasHotTub);
    $stmt -> bindParam(":Capacity", $Capacity);
    $stmt->bindParam(":ImgContent", $ImgContent);

    $stmt -> execute();
}
```

Edit room query

Διαγραφή Δωματίου

Για την διαγραφή ενός δωματίου είναι αρκετό να προσθέσουμε ένα κουμπί μέσα στην συνάρτηση προβολής των δωματίων. Το συνδέουμε μέσω του id του δωματίου. Όταν το κουμπί πατηθεί από τον διαχειριστή, θα του εμφανιστεί confirmation.

Μόλις πατηθεί το **OK**, η εφαρμογή θα αποκριθεί στο αρχείο **delete_room.inc.php**.

Αυτό με τη σειρά του όπως έγινε στην προσθήκη και στην αλλαγή στοιχείων δωματίου θα καλέσει Query για να διαγράψει το δωμάτιο από τον πίνακα **room** μαζί με τις εγγραφές σε άλλους πίνακες τις βάσεις οι οποίες περιλαμβάνουν το **RoomID** του ως **Foreign Key** (π.χ. **makes_simple_resrv**).

localhost says

Are you sure you want to delete this room?

OK

Cancel

Confirmation

```
function delete_room($pdo, $RoomID): void {  
    $sql = "DELETE FROM room WHERE RoomID = $RoomID";  
  
    $stmt = $pdo->prepare($sql);  
    $stmt -> execute();  
}
```

Delete room query

Admin – Services



HOME OUR ROOMS SERVICES SPECIAL OFFERS Logout

SERVICES

Τα Services στη σελίδα του admin είναι λειτουργικά πολύ παρόμοια με τα δωμάτια, απλά σε μεγαλύτερη κλίμακα. Πάλι ο διαχειριστής έχει την ικανότητα να κάνει **Add**, **Edit** και **Delete** τα 3 ξεχωριστά είδη από Service: **Activities**, **Wellness** και **Culinary Experiences**.

Culinary Experiences



Private Dinner

Description: Indulge in a bespoke dining experience with our private chef. Enjoy gourmet meals crafted to your taste, in the comfort of your suite or any location you choose. Perfect for intimate dinners, celebrations, or savoring exquisite cuisine without leaving your space.

Meal Type: Any

Menu Options: Any

Dress Code: Any

Booking Required: Yes

Price: 40€

Availability Hours: 19:00 – 23:00

[Edit](#)

[Delete](#)



Bar

Description: Unwind at our stylish hotel bar, offering a curated selection of cocktails, fine wines, and craft beers. Enjoy a sophisticated ambiance, comfortable seating, and attentive service, perfect for socializing or relaxing after a day of activities.

Meal Type: Cocktails&Appetizers

Menu Options: Small Plates

Dress Code: Smart Casual

Booking Required: No

Price: 15€

Availability Hours: 19:00 – 04:00

[Edit](#)

[Delete](#)



Restaurant

Description: Enjoy a varied dining experience with our diverse menu. Choose from à la carte dishes, hearty mains, and delectable desserts, all crafted from fresh, high-quality ingredients. Whether you're in the mood for a fine dining experience or casual fare, we have something for every taste.

Meal Type: Dinner

Menu Options: Vegan, Gluten-Free

Dress Code: Formal

Booking Required: Yes

Price: 15€

Availability Hours: 13:00 – 22:00

[Edit](#)

[Delete](#)

[Add Culinary Experience](#)

Τμήμα σελίδας κατηγορίας από Services

Προσθήκη Υπηρεσίας

Η δημιουργία καινούργιας Υπηρεσίας από πλευρά χρήστη είναι η ίδια με ενός δωματίου. Η φόρμα για κάθε είδος υπηρεσίας περιλαμβάνει τα κοινά χαρακτηριστικά από κάθε Service που ανήκουν στον πίνακα **service** και τα ειδικά χαρακτηριστικά της κάθε κατηγορίας (π.χ. για τις υπηρεσίες μαγειρικής, συμπεριλαμβάνονται τα γνωρίσματα του πίνακα **culinary_experience**).

Με το πάτημα του Save γίνεται η ίδια διαδικασία όπως στην προσθήκη δωματίου, με την διαφορά ότι αντί για 1 query, έχουμε 2 συνολικά, ένα που προσθέτει τα γνωρίσματα στον πίνακα **services** και ένα που προσθέτει στον ανάλογο πίνακα του είδους της υπηρεσίας. Στον ειδικό πίνακα προστίθεται χειροκίνητα και το ID του Service για να συνδεθεί με την εγγραφή στο **service**.

```
// Add service to the generic service table
add_service_to_table($pdo, $price, $bookingRequired, $description, $serviceName, $availabilityHours, $imagepath);

// Get the last inserted service ID
$serviceID = $pdo->lastInsertId();

// Add service to the culinary_experience table
add_service_to_culinary($pdo, $serviceID, $mealType, $menuOptions, $dressCode, $bookingRequired);
```

Κλήση των Queries

The screenshot shows a modal dialog box titled "Add Culinary Experience". It contains several input fields: "Service Name", "Price", "Availability Hours", "Description", "Image" (with a "Choose File" button and "No file chosen" placeholder), "Meal Type", "Menu Options", "Dress Code", and a checkbox for "Booking Required". At the bottom right are "Close" and "Save" buttons.

Προσθήκη υπηρεσίας

```
function add_service_to_table($pdo, $price, $bookingRequired, $description, $serviceName, $availabilityHours, $imagepath): void {
    $sql = "INSERT INTO service (Price, BookingRequired, Description, ServiceName, AvailabilityHours, Image)
           VALUES (:Price, :BookingRequired, :Description, :ServiceName, :AvailabilityHours, :Image)";
    $stmt = $pdo ->prepare($sql);
    $stmt ->bindParam(':Price', $price);
    $stmt ->bindParam(':BookingRequired', $bookingRequired);
    $stmt ->bindParam(':Description', $description);
    $stmt ->bindParam(':ServiceName', $serviceName);
    $stmt ->bindParam(':AvailabilityHours', $availabilityHours);
    $stmt ->bindParam(':Image', $imagepath);
    $stmt ->execute();
}
```

Προσθήκη εγγραφής στον πίνακα services

```
function add_service_to_culinary($pdo, $serviceID, $mealType, $menuOptions, $dressCode): void {
    $sql = "INSERT INTO culinary_experience (ServiceID, MealType, MenuOptions, DressCode)
           VALUES (:ServiceID, :MealType, :MenuOptions, :DressCode)";
    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':ServiceID', $serviceID);
    $stmt->bindParam(':MealType', $mealType);
    $stmt->bindParam(':MenuOptions', $menuOptions);
    $stmt->bindParam(':DressCode', $dressCode);
    $stmt->execute();
}
```

Προσθήκη εγγραφής στον ειδικό της πίνακα

Επεξεργασία – Διαγραφή Υπηρεσίας

Ακριβώς οι ίδιες διαδικασίες με αυτές του δωματίου, απλά και με τους 2 πίνακες της κάθε υπηρεσίας, αντί για τον 1 πίνακα του δωματίου.

Admin – Special Offers

Η τελευταία σελίδα του admin είναι η **Special Offers**. Οι λειτουργίες **Edit** και **Delete** είναι ίδιες με τις προηγούμενες 2 σελίδες που αναφέρθηκαν, αυτή τη φορά πάνω στον πίνακα **special_offer** (η διαγραφή γίνεται πάλι σε όποιον πίνακα βρίσκουμε το ID του special offer που διαγράφεται).

The screenshot shows the 'SPECIAL OFFERS' section of the Mt Gateway website. It features three offers:

- Offer 1:** Enjoy a romantic getaway with our special couples offer. Stay in a beautifully appointed room and indulge in unlimited spa sessions, designed to relax and rejuvenate. Perfect for couples seeking a luxurious escape.
Discount: 25%
Associated Service: Spa
Buttons: Add to Room (green), Remove Connection to Room (green), Edit (yellow), Delete (red)
- Offer 2:** Enjoy a cozy stay for four and take advantage of free ski passes for the entire family. Book now and embrace the ultimate winter adventure!
Discount: 25%
Associated Service: Ski/Snowboarding
Buttons: Add to Room (green), Remove Connection to Room (green), Edit (yellow), Delete (red)
- Offer 3:** Enjoy a free pass to our buffet with your stay! Savor a variety of delicious dishes, from breakfast to dinner, at no extra cost. Perfect for food lovers seeking a culinary adventure during their visit. Book now to treat yourself to endless dining delights!
Discount: 25%
Associated Service: Buffet Lounge
Buttons: Add to Room (green), Remove Connection to Room (green), Edit (yellow), Delete (red)

Add Special Offer button is located at the bottom left.

Παράδειγμα της σελίδας με 3 εγγραφές

Για κάθε εγγραφή ειδικής προσφοράς απεικονίζεται η περιγραφή της, το ποσό της έκπτωσης που προσφέρεται στο πακέτο συνδυασμού δωματίου και επιλεγμένης υπηρεσίας και τέλος το όνομα της υπηρεσίας στην οποία είναι συνδεδεμένη η προσφορά.

Προσθήκη Special Offer

Η προσθήκη νέας προσφοράς εκτός από τα κοινά χαρακτηριστικά που βρίσκονται και στις άλλες σελίδες του διαχειριστή, περιλαμβάνει και την επιλογή της υπηρεσίας που θα συνδεθεί μαζί της για να

```
$services = get_services_with_types($pdo); // Function to fetch services and types
foreach ($services as $service) {
    if($service['ServiceID'] == $offer['ServiceID']) {
        $serviceName = $service['ServiceName'];
        break;
    }
}

echo '<div class="bed_room">';
echo '<h3>' . htmlspecialchars(string: $offer['Description']) . '</h3>';
echo '<p>Discount: ' . htmlspecialchars(string: $offer['Discount']) . '%<br>';
echo '<p>Associated Service: ' . htmlspecialchars(string: $serviceName) . '</p>';
```

Τρόπος εύρεσης του ονόματος της υπηρεσίας μέσω του ServiceID της

```
function get_services_with_types($pdo): mixed {
    $sql = "SELECT s.ServiceID, s.ServiceName, IF(a.ServiceID IS NOT NULL, 'Activity',
    IF(w.ServiceID IS NOT NULL, 'Wellness', 'Culinary')) AS ServiceType
    FROM service s
    LEFT JOIN activity a ON s.ServiceID = a.ServiceID
    LEFT JOIN wellness w ON s.ServiceID = w.ServiceID
    LEFT JOIN culinary_experience c ON s.ServiceID = c.ServiceID";

    $stmt = $pdo->prepare($sql);
    $stmt->execute();
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}
```

Εύρεση και κατηγοριοποίηση όλων των Services

ενωθούν με ένα δωμάτιο πιο μετά. Αυτό επιτυγχάνεται με την εύρεση και την κατηγοριοποίηση όλων των αποθηκευμένων services στην βάση δεδομένων του ξενοδοχείου και την εμφάνισή τους σε μία **dropdown** λίστα, μαζί με την υποκατηγορία τους.

Προσθήκη Special Offer Σε Δωμάτιο

Πατώντας το κουμπί **Add to Room** πάνω σε μία εγγραφή, εμφανίζεται μια φόρμα επιλογής δωματίου. Ο διαχειριστής μπορεί να επιλέξει ένα από τα δωμάτια που είναι αποθηκευμένα στην βάση δεδομένων και να πατήσει το **Add Offer to Room**.

Στην περίπτωση που το δωμάτιο δεν είναι συνδεδεμένο με την προσφορά, θα δημιουργηθεί καινούργια εγγραφή στον πίνακα **combination** που θα περιλαμβάνει το **RoomID** και το **SpecialOfferID**. Αυτός είναι ο πίνακας που χρησιμοποιείται για την εμφάνιση των προσφορών από την πλευρά του επισκέπτη/πελάτη στην αντίστοιχη σελίδα του.

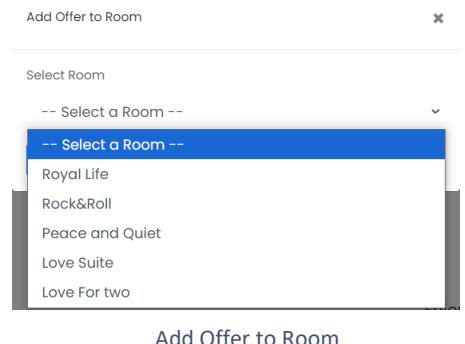
Στην περίπτωση που ο συνδυασμός δωματίου και προσφοράς υπάρχει, δεν εισάγεται ξανά εγγραφή στον πίνακα combination.

[admin_special_offers.php?combinationAlreadyExists=True](#)

Combination Already Exists message



Services Drop Down Menu



Add Offer to Room

Αφαίρεση Συσχέτισης Special Offer Με Δωμάτιο

Η τελευταία λειτουργία του admin. Εμφανίζονται μόνο τα δωμάτια στα οποία το επιλεγμένο Special Offer. Ο διαχειριστής επιλέγει δωμάτιο και όταν πατάει **Confirm** διαγράφεται η εγγραφή στον πίνακα **combination** που συνδέει το δωμάτιο με την προσφορά.

```
1 reference
function add_new_combination($pdo, $roomID, $offerID): void {
    $sql = "INSERT INTO combination (RoomID, SpecialOfferID) VALUES (:RoomID, :OfferID)";
    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':RoomID', $roomID);
    $stmt->bindParam(':OfferID', $offerID);
    $stmt->execute();
}

1 reference
function check_if_combination_exists($pdo, $roomID, $offerID): bool{
    $sql = 'SELECT * FROM combination WHERE (RoomID = :RoomID and SpecialOfferID = :OfferID)';
    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':RoomID', $roomID);
    $stmt->bindParam(':OfferID', $offerID);
    $stmt->execute();
    if(!empty($stmt->fetchAll(PDO::FETCH_ASSOC))){
        return true;
    } else return false;
}

1 reference
function delete_combination($pdo, $offerID, $roomID): void {
    $sql = 'DELETE FROM combination WHERE (RoomID = :RoomID and SpecialOfferID = :OfferID)';
    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':RoomID', $roomID);
    $stmt->bindParam(':OfferID', $offerID);
    $stmt->execute();
}
```

Combination queries

Μελλοντικές Επεκτάσεις

Σαν επιπλέον λειτουργικότητα που θα μπορούσε να έχει στο μέλλον η ιστοσελίδα μας είναι τα παρακάτω στοιχεία:

- Μια ενότητα “My reservations”, στην οποία ο χρήστης θα μπορούσε να δει οποιαδήποτε κράτηση έχει κάνει (δωματίου, υπηρεσίας ή πακέτου). Θα μπορούσε υπό συνθήκες να του δίνεται η δυνατότητα να επεξεργαστεί ή διαγράψει μια κράτηση, ενώ επιπρόσθετα θα μπορούσε να δει το ιστορικό των παλαιότερων κρατήσεων του.
- Ύπαρξη πλειότιμου γνωρίσματος “tags” στον πίνακα “special_offers”. Με τη χρήση αυτού του πεδίου θα μπορούσαμε να προσθέτουμε λέξεις κλειδιά για καλύτερη αναζήτηση του χρήστη.
- Διαφήμιση πακέτων προσφορών πριν την τελική κράτηση ενός δωματίου. Αφού ο χρήστης έχει επιλέξει δωμάτιο, θα μπορούσαμε να του δείχνουμε τις διαθέσιμες προσφορές που περιλαμβάνουν το επιθυμητό δωμάτιο.

Πηγές

Εκμάθηση της PHP: PHP Course for Beginners - Dani Krossing

<https://youtube.com/playlist?list=PL0eyrZgxdwhwwQQZA79OzYwl5ewA7HQih&si=mB3SjFCfkuvkAHc->

Θεωρία SQL: Διαφάνειες Διαλέξεων των μαθημάτων Βάσεις I & II

Βοηθήματα: <https://www.w3schools.com/>

Η εργασία πραγματοποίηθηκε με την εφαρμογή XAMPP - Πηγή: www.apachefriends.org

Χρήση template από: <https://html.design/>

Παραδοτέα Αρχεία

- Φάκελος “MtGateway”: Περιέχει τον κώδικα της ιστοσελίδας
- Hotel_ER.erx: Το ER της βάσης μας δημιουργήθηκε με την εφαρμογή ER2SQL
- Database.sql: Η βάση δεδομένων που πρέπει να φορτωθεί στο xampp
- Report_03296_03285.pdf