

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



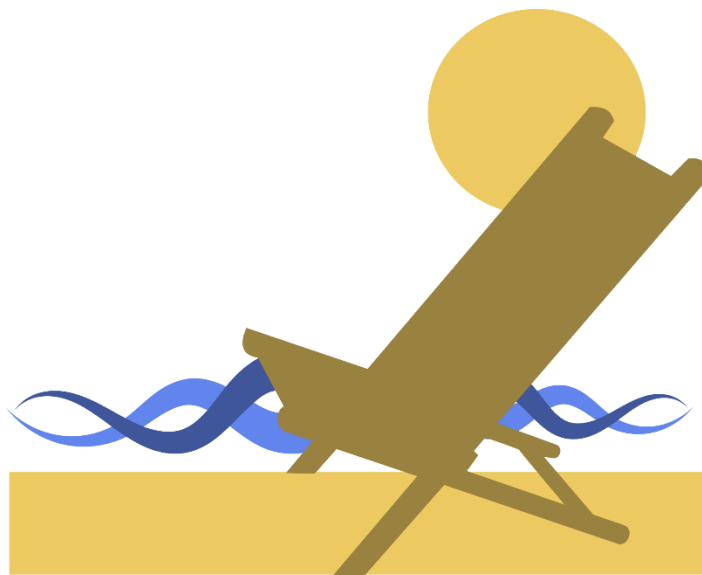
Πανεπιστήμιο
Ιωαννίνων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

«Πλατφόρμα Ηλεκτρονικής Κράτησης»

«Online booking platform»

«Vres Xaplostra»



Βλάχος Σπυρίδων

ΑΜ: 1159 εξάμηνο : 14^ο

Email: spiros.vlachos@icloud.com

Εποπτεύων καθηγητής

Γλαβάς Ευριπίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



Πανεπιστήμιο
Ιωαννίνων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

«Πλατφόρμα Ηλεκτρονικής Κράτησης»

«Online booking platform»

«Vres Xaplostra»

Βλάχος Σπυρίδων

ΑΜ: 1159 εξάμηνο : 14^ο

Email: spiros.vlachos@icloud.com

Εποπτεύων καθηγητής

Γλαβάς Ευριπίδης

Εγκρίθηκε από τριμελή Εξεταστική Επιτροπή

Άρτα, 22/07/2021

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Εποπτεύον Καθηγητής

Γλαβάς Ευριπίδης

2. Μέλος Επιτροπής

Τζάλλας Αλέξανδρος

Επίκουρος Καθηγητής

3. Μέλος Επιτροπής

Γιαννακέας Νικόλαος

Επίκουρος Καθηγητής

Ο Πρόεδρος του Τμήματος

Γλαβάς Ευριπίδης

© Βλάχος Σπυρίδων, 2021.

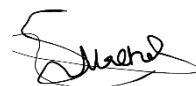
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα Πτυχιακή Εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής, ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη Βιβλιογραφία.

Βλάχος Σπυρίδων

Υπογραφή



ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ευχαριστίες και την ευγνωμοσύνη μου στον καθηγητή μου κ. Ευριπίδη Γλαβά για τη βοήθειά του, το ενδιαφέρον του αλλά και το χρόνο που διέθεσε για τη διεκπεραίωση της Πτυχιακής μου Εργασίας.

ΠΕΡΙΛΗΨΗ

Στην παρούσα Πτυχιακή Εργασία θα μελετήσουμε τον τρόπο κατασκευής μιας πλατφόρμας κρατήσεων που ανέπτυξα σε γλώσσα προγραμματισμού PHP και HTML η οποία φιλοξενείται σε έναν Apache Webserver.

Το θέμα της Πτυχιακής είναι η κατασκευή μιας διαδικτυακής πλατφόρμας με σκοπό το κλείσιμο ξαπλωστών σε Ελληνικές παραλίες.

Αρχικά θα ορίσουμε τις ορολογίες, θα μάθουμε την ιστορία των τεχνολογιών που θα χρησιμοποιήσουμε και θα γνωρίσουμε την αρχιτεκτονική και τον τρόπο ενσωμάτωσης των.

Στην πορεία θα εμβαθύνουμε στον τρόπο προγραμματισμού και στο πρακτικό κομμάτι της Πτυχιακής.

Τέλος, θα παρουσιαστεί η λειτουργία της πλατφόρμας και η χρησιμότητα της.

Λέξεις κλειδιά: vresxaplostra, booking platform, PHP, HTML, MySQL, SQL, JavaScript, CSS, domain, hosting, programming

ABSTRACT

In this thesis we will study how to build a booking platform that I developed in PHP and HTML and it is hosted on an Apache Webserver.

The topic of the dissertation is the construction of an online platform in order to book sunbeds on Greek beaches.

First, we will define the terminology, we will learn the history of the technologies we will use, and we will get to know the architecture and the way of their integration.

Continuing, we will delve into the way of programming and the in general at the practical part of the dissertation.

Finally, we will present the operation of the platform and its usefulness.

Keywords: vresxaplostra, booking platform, PHP, HTML, MySQL, SQL, JavaScript, CSS, domain, hosting, programming

Περιεχόμενα

Κεφάλαιο 1^ο Εισαγωγή στις Γλώσσες Προγραμματισμού	13
1.1 Πίνακας Ορολογιών.....	13
1.2 Εισαγωγή στην HTML.....	14
1.3 Εισαγωγή στην PHP.....	14
1.4 Εισαγωγή στην MySQL.....	16
1.5 Εισαγωγή στο Bootstrap	17
Κεφάλαιο 2^ο Εισαγωγή στις έννοιες των Ιστοσελίδων	18
2.1 Τί είναι domain	18
2.2 Πώς να αποκτήσεις ένα domain name	18
2.3 Τί είναι το hosting.....	19
2.4 Πώς να αποκτήσεις έναν hosting server.....	19
2.5 Πώς να συνδέσεις το domain με τον hosting server	20
Κεφάλαιο 3^ο Επεξήγηση κώδικα εργασίας.....	21
3.1 Επεξήγηση βασικών αρχών.....	21
3.2 Δημιουργία βάσης δεδομένων	24
3.3 Εγγραφή νέου χρήστη.....	26
3.4 Σύνδεση χρήστη	30
3.5 Καταχώρηση κράτησης.....	31
3.6 Δημιουργία κράτησης στη βάση	35
3.7 Προβολή κρατήσεων συνεργάτη	37
3.8 Διαχείριση κρατήσεων συνεργάτη.....	39
3.9 Διαχείριση τιμών συνεργάτη	41
3.10 Εύρεση συνεργάτη από διαχειριστή.....	43
3.11 Εμφάνιση κρατήσεων πλατφόρμας διαχειριστή	51
3.12 Εξαγωγή στατιστικών πλατφόρμας διαχειριστή.....	52
3.13 Προσθήκη νέα παραλίας από διαχειριστή	55
3.14 Προσθήκη τμήματος παραλίας από διαχειριστή.....	57
3.15 Διαγραφή τμήματος παραλίας από διαχειριστή	60
3.16 Ενσωμάτωση πλατφόρμας PayPal	62
Κεφάλαιο 4^ο Παρουσίαση λειτουργίας εργασίας.....	66
4.1 Παρουσίαση πλατφόρμας σαν Συνεργάτης.....	66
4.2 Παρουσίαση πλατφόρμας σαν Administrator	68
4.3 Παρουσίαση πλατφόρμας σαν Χρήστης	72

ΣΥΜΠΕΡΑΣΜΑΤΑ	75
ΕΡΕΥΝΑ	76
ΒΙΒΛΙΟΓΡΑΦΙΑ	79

Κεφάλαιο 1^ο Εισαγωγή στις Γλώσσες Προγραμματισμού

1.1 Πίνακας Ορολογιών

CSS	Cascading Style Sheet
PHP	Nonsensical acronym for PHP programming language and interpreter
HTML	Hypertext Markup Language
MySQL	Relational database management system
SQL	Structured query language
XAMPP	Cross platform, Apache HTTP server, MySQL, PHP
JS	Javascript
API	Application Programming Interface
Bootstrap	Front-end development framework

1.2 Εισαγωγή στην HTML

Η HTML είναι μια γλώσσα επεξεργασίας υπερκειμένου που τρέχει σε browsers και είναι ικανή να δημιουργεί σελίδες με κείμενο, χρώματα, φόρμες, αντικείμενα κτλ. Δεν θεωρείται γλώσσα προγραμματισμού, αλλά μια γλώσσα που προσδιορίζει την έννοια, το σκοπό και τη δομή του κειμένου μέσα σε ένα έγγραφο.

Όλα ξεκίνησαν με την ανακάλυψη του World Wide Web το 1990 όπου το hypertext ή αλλιώς στα Ελληνικά υπερκείμενο άρχισε να χρησιμοποιείται και μαζί του να αυξάνεται και το ποσοστό των χρηστών του ίντερνετ. Εκεί λοιπόν πήρε κόκκαλα και οστά η HTML με αρχικούς ευρετές του τους Jim Clark και Marc Andreessen.

Από τότε η HTML έχει αλλάξει κατά πολύ και μαζί της έχουν αλλάξει και οι browsers. Στην αρχή της ζωής της, γινόταν η προσπάθεια τυποποίησης κάποιων κανόνων και προτύπων της γλώσσας, έτσι ώστε να τρέχει και να συμπεριφέρεται πανομοιότητα σε όλους τους browsers. Έτσι, το 1993 έγινε και η πρώτη σύνδεση δύο απομακρυσμένων υπολογιστών όπου κατάφεραν να κατεβάσουν μεταξύ τους μια σελίδα στον browser.

Σήμερα η HTML έχει φτάσει στην έκδοση 5 και έχει δεχθεί πολλές αλλαγές και βελτιώσεις. Ο σκοπός των αλλαγών ανάμεσα στις εκδόσεις είναι η ενημέρωση της γλώσσας με τους νέους τρόπους προγραμματισμού, η τυποποίηση βέλτιστων πρακτικών και η ενσωμάτωση των νέων δυνατοτήτων που προσφέρουν οι browsers. [1]

1.3 Εισαγωγή στην PHP

Η PHP είναι μια από τις πιο δημοφιλής server-side scripting γλώσσα προγραμματισμού που αρχικά σχεδιάστηκε για την ανάπτυξη ιστοσελίδων και για την ακρίβεια δυναμικών ιστοσελίδων.

Για το σκοπό αυτό, ο κώδικας της PHP ενσωματώνεται στον κώδικα της HTML και «μεταφράζεται» από έναν διακομιστή ιστού (Apache server) όπου έχει υποστήριξη για PHP. Στη σημερινή έκδοση της γλώσσας υπάρχουν πολλές δυνατότητες όπως για παράδειγμα η δημιουργία δυναμικού περιεχομένου, η ενσωμάτωση λειτουργιών υποστήριξης βάσεων

δεδομένων, η αποστολή φορμών δεδομένων αλλά και η δημιουργία και διαχείριση των cookies.

Επειδή η PHP εκτελείται από την πλευρά του διακομιστή, ο υπολογιστής-πελάτης δεν μπορεί να αλλάξει ή να μεταβάλλει το περιεχόμενο του πηγαίου κώδικα παρά μόνο να τον τρέξει στον browser του. Με αυτό τον τρόπο μια ιστοσελίδα PHP προεπεξεργάζεται από τον interpreter το περιεχόμενο της και έπειτα στέλνεται στον browser. Οι πιο γνωστοί browsers αυτή την χρονική στιγμή είναι ο Google Chrome, ο Mozilla Firefox, ο Microsoft Edge και ο Apple Safari.

Παρόμοια γλώσσα προγραμματισμού με αυτή της PHP είναι η ASP της Microsoft και η JSP της Java. Γενικά, η PHP έχει ως βάση τις βασικές αρχές της γλώσσας C αλλά και μερικά δικά της μοναδικά χαρακτηριστικά. Ένα από αυτά τα χαρακτηριστικά είναι το scripting, το οποίο καθιστά την PHP από τις πιο δημοφιλείς γλώσσες και πολύ αγαπητή προς τους web developers λόγω της ευκολίας της να παράγουν τις ιστοσελίδες.

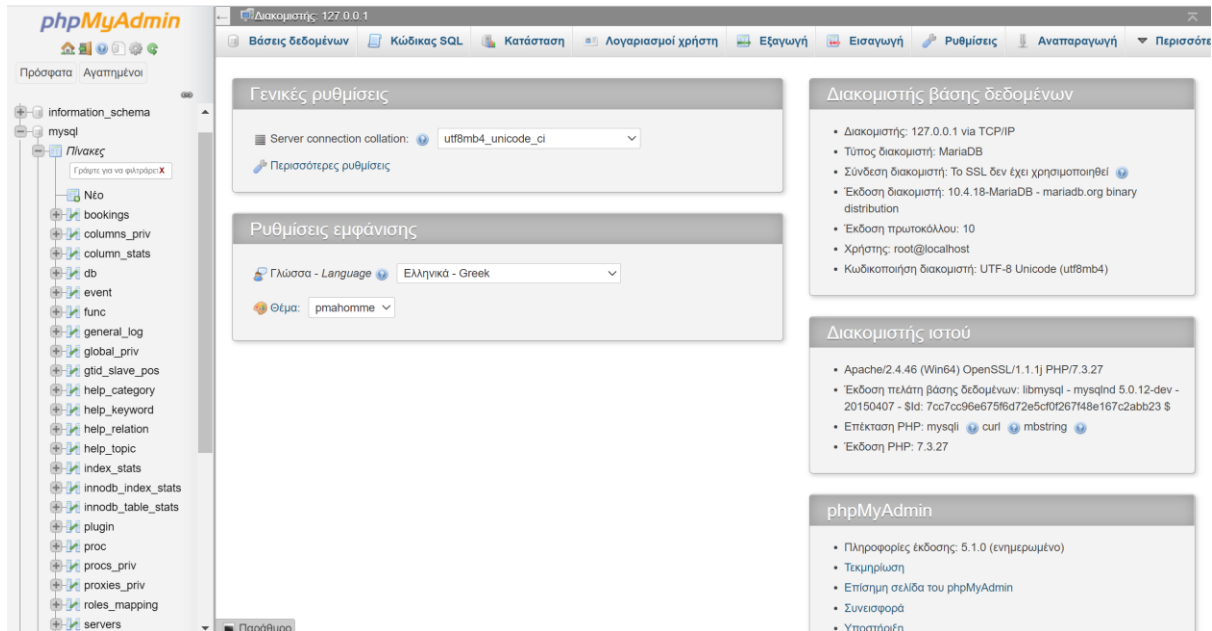
Επιπλέον είναι εύκολο να εντοπιστούν bug (σφάλματα) κατά την εκτέλεση του κώδικα εξαιτίας του ενσωματωμένου συστήματος debugging που έχουν οι browsers. Παρόλα αυτά, ένα από τα μειονεκτήματα της είναι η φτώχη της δυνατότητα να αναγνωρίσει λογικά λάθη που έχει κάνει ο προγραμματιστής καθώς δεν υπάρχει μεταγλωττιστής που να σε προειδοποιεί.

Κάποιες επιπλέον δυνατότητες της PHP είναι η παραγωγή και εξαγωγή αρχείων, όπως για παράδειγμα xls ή και pdf, η εξαγωγή εικόνων, το ανέβασμα αρχείων σε έναν server η δημιουργία δυναμικών πινάκων από βάση δεδομένων και ο έλεγχος διαπιστευτηρίων χρήστη με βάση τη συνεδρία του browser.

Τέλος, η PHP είναι μια γλώσσα εύκολη στην εκμάθηση καθώς ενσωματώνει εντολές φυσικής γλώσσας στα Αγγλικά και βασικών αρχών προγραμματισμού. Επίσης κάποιος μπορεί να βρει βοήθεια στις απορίες του με την χρήση του ίντερνετ καθώς υπάρχουν πολλά blogs και εκπαιδευτικό περιεχόμενο δωρεάν. [2]

1.4 Εισαγωγή στην MySQL

Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιείται για να αλληλοεπιδρά με την SQL που είναι η γλώσσα πρόσβασης σε ένα περιβάλλον βάσης δεδομένων. Είναι ένα γρήγορο και εύκολο περιβάλλον που επιτρέπει την προσπέλαση, αναζήτηση, προσθήκη και ταξινόμηση των δεδομένων σε μια βάση.



Εικόνα 1 : Πλατφόρμα διαχείρισης MySQL

Με χρήση της MySQL μπορούν ταυτόχρονα πολλαπλοί χρήστες να έχουν πρόσβαση σε μια βάση δεδομένων με έλεγχο των διαπιστευτηρίων τους, ώστε να διασφαλιστεί η προστασία στα δεδομένα πρόσβασης.

Η MySQL έχει έναν server, ο οποίος είναι μέρος του πακέτου εγκατάστασης της και τυπικά αρχικοποιείται και εκκινεί κατά την έναρξη του υπολογιστή. Είναι έτοιμη να δεχθεί εντολές και να διαχειρίζεται αιτήσεις μεταξύ ενός συστήματος π.χ. browser ή εφαρμογής με τη βάση δεδομένων όπου συνήθως γίνεται με χρήση της PHP.

Η όλη επικοινωνία γίνεται μέσω εντολών SQL (Structured Query Language) όπου γνωρίζουν τα συστήματα διαχείρισης βάσεων (RDBMS). Ουσιαστικά η PHP δεν μπορεί να αναγνωρίσει τις εντολές αυτές, αλλά είναι υπεύθυνη για τη δημιουργία της σύνδεσης με την MySQL ώστε να μπορέσουν να ανταλλάξουν μηνύματα. Για το λόγο αυτό συνηθίζεται να φτιάχνεται ένα

αρχείο τύπου config.php όπου περιέχει τα στοιχεία σύνδεσης με τη βάση δεδομένων για ευκολία κατά την κλήση της.[3]

1.5 Εισαγωγή στο Bootstrap

Το Bootstrap είναι κώδικας που βοηθάει την εύκολη ανάπτυξη του γραφικού περιβάλλοντος μιας σελίδας. Ουσιαστικά είναι μια τεράστια συλλογή από εύχρηστα και επαναχρησιμοποιούμενα κομμάτια κώδικα γραμμένα σε HTML, CSS και JavaScript.

Επιπλέον, είναι ένα πλαίσιο ανάπτυξης γραφικών που επιτρέπει στους προγραμματιστές να δημιουργούν fully responsive ιστοσελίδες εύκολα και γρήγορα. Ουσιαστικά, το Bootstrap εξοικονομεί χρόνο από την σύνταξη πολλαπλού κώδικα σε CSS, δίνοντας έτσι περισσότερο χρόνο στο σχεδιασμό της ιστοσελίδας.

Μερικά από τα πλεονεκτήματα είναι:

- ❖ Είναι δωρεάν
- ❖ Έχει γρήγορη απόκριση
- ❖ Προσαρμόζει τις εικόνες ώστε να ανταποκρίνονται
- ❖ Έχει μια μεγάλη γκάμα από έτοιμα components (menu, dropdown lists)
- ❖ Έχει πολύ καλή βιβλιογραφία
- ❖ Είναι εύκολα προσαρμόσιμος [4]



Εικόνα 2 : Παράδειγμα χρήσης Bootstrap

Κεφάλαιο 2: Εισαγωγή στις έννοιες των Ιστοσελίδων

2.1 Τί είναι domain

Το domain είναι ουσιαστικά ένα όνομα μιας ιστοσελίδας που ισοδυναμεί με τη φυσική της IP διεύθυνση. Αντί λοιπόν να απομνημονεύουμε ολόκληρες διευθύνσεις, όπως για παράδειγμα 172.217.168.67 (www.google.gr), μας είναι πιο εύκολο να χρησιμοποιούμε τα domain name τους για τις βρούμε.

Το domain name αποτελείται από δυο πράγματα, το όνομα της ιστοσελίδας και την κατάληξη της. Για παράδειγμα, το domain name του facebook.com περιέχει το όνομα της ιστοσελίδας (Facebook) και την κατάληξη του domain (.com). Όταν κάποιος αγοράσει ένα domain name έχει τη δυνατότητα να επιλέξει το που θα «κατευθύνεται» η ιστοσελίδα αυτή.

Την αγορά των domain names επιβλέπει ένας Διεθνής Οργανισμός που λέγεται ICANN (Internet Corporation for Assigned Names and Numbers). Το ICANN καθορίζει ποιες καταλήξεις (π.χ.: .com, .gr, .edu) είναι διαθέσιμες και συντηρεί μια κεντρική βάση δεδομένων με το που καταλήγουν οι διευθύνσεις αυτές. [5]

Η κατάληξη .gr που χρησιμοποιείται κυρίως από Ελληνικές ιστοσελίδες, διαχειρίζεται από το Ελληνικό Ινστιτούτο Πληροφορικής (Ι.Π.). Το Ι.Π. ιδρύθηκε το 1983 και έχει βάση του το Ηράκλειο Κρήτης είναι ένα από τα οχτώ ερευνητικά Ινστιτούτα του Ιδρύματος Τεχνολογίας και Έρευνας (ΙΤΕ), όπου διεξάγει βασική και εφαρμοσμένη έρευνα, και διαδραματίζει πρωταγωνιστικό ρόλο στην ανάπτυξη της Κοινωνίας της Πληροφορίας σε εθνικό και διεθνές επίπεδο. Ένα από τα πολλά τμήματα του, το Μητρώο ονομάτων χώρου διαδικτύου είναι αυτό που είναι υπεύθυνο για τις καταλήξεις .gr και .ελ. [6]

2.2 Πως να αποκτήσεις ένα domain name

Υπάρχουν πολλοί προμηθευτές αγοράς domain name στην αγορά όπου με διαφορετικές τιμές και παροχές ο καθένας. Εγώ επέλεξα να αγοράσω ένα domain name από το [hostinger.com](https://www.hostinger.com) στην περίπτωση μας.

Το πρώτο πράγμα που έχει να κάνει κάποιος είναι να βρει το όνομα και την κατάληξη που του αρέσει και να ελέγξει αν είναι διαθέσιμη για αγορά. Πηγαίνοντας στο <https://www.hostinger.com/domain-checker> έλεγξα ότι το domain name: “vresxaplostra.site” είναι διαθέσιμο και προχώρησα προς αγορά.

Αφού ολοκληρώσουμε και κατοχυρώσουμε το domain name μας πρέπει να το κατευθύνουμε προς τον server μας ή αλλιώς τον εξυπηρετητή που θα είναι υπεύθυνος για να κάνει host την ιστοσελίδα μας.

2.3 Τί είναι το hosting

Το hosting είναι η λειτουργία ενός φυσικού server ο οποίος είναι συνδεδεμένος με το διαδίκτυο και φιλοξενεί μια ιστοσελίδα. Ουσιαστικά είναι ένας υπολογιστής με έναν σκληρό δίσκο όπου έχει αποθηκευμένα τα αρχεία της ιστοσελίδας μας καθώς και τη βάση δεδομένων που θα χρησιμοποιήσουμε. Λέγεται λοιπόν εξυπηρετητής γιατί εξυπηρετεί αιτήσεις χρηστών που θέλουν να επισκεφτούν το site μας.

Όπως ακριβώς συμβαίνει και με την αγορά domain name, έτσι λειτουργεί και το hosting με την έννοια ότι πρέπει να αγοράσεις έναν server από έναν προμηθευτή υπηρεσιών ώστε να φιλοξενήσεις την ιστοσελίδα σου. Υπάρχουν πολλοί προμηθευτές που κάνουν την ίδια δουλειά αλλά διαφοροποιούνται στις υπηρεσίες που σου προσφέρουν, όπως για παράδειγμα ταχύτητα σύνδεσης, bandwidth δικτύου, ταχύτητα επεξεργαστή server και διαθεσιμότητα (uptime).

2.4 Πώς να αποκτήσεις έναν hosting server

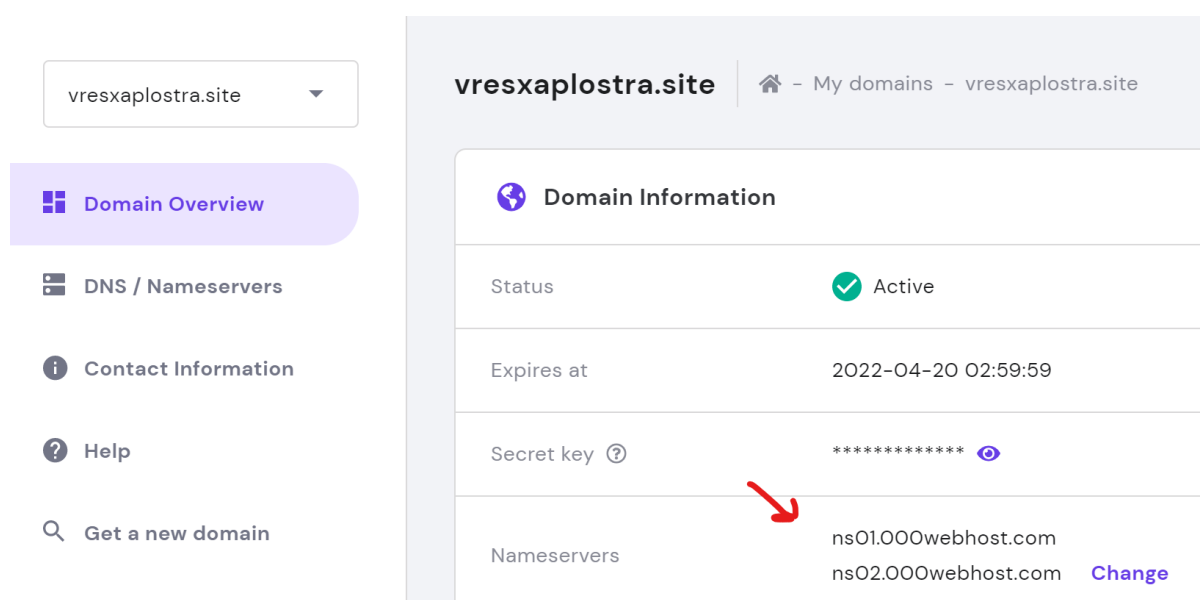
Όπως είπαμε και προηγουμένως, υπάρχουν πολλοί hosting providers. Στα πλαίσια της εργασίας αυτής και έπειτα από αναζήτηση που έκανα, βρήκα το <https://www.000webhost.com/> που σου προσφέρει δωρεάν hosting μίας μόνο ιστοσελίδας για απλή χρήση. Έτσι λοιπόν, αφού κάνουμε εγγραφή στο παραπάνω site, πατάμε μεταφόρτωση των αρχείων μας στον server ώστε να υπάρχουν στο δημόσιο (public_html) φάκελο του εξυπηρετητή.

Εκτός από τη μεταφόρτωση των αρχείων μας, θα χρειαστεί να δημιουργήσουμε και τη βάση δεδομένων μας. Επιλέγοντας λοιπόν database manager και στη συνέχεια + new database δημιουργούμε την MySQL βάση, δίνοντας της ένα όνομα, username και τέλος έναν ασφαλή κωδικό πρόσβασης.

2.5 Πώς να συνδέσεις το domain με τον hosting server

Αφού έχουμε ολοκληρώσει και την απόκτηση του domain name μας αλλά και την απόκτηση του hosting server μας, ήρθε η ώρα να τα συνδέσουμε. Ουσιαστικά με τον όρο «συνδέσουμε» εννοούμε την κατεύθυνση του domain name μας προς στην IP διεύθυνση του server μας.

Η διαδικασία γίνεται ως εξής. Αρχικά θα πρέπει να μεταβούμε στο domain προμηθευτή μας, στη δική μας περίπτωση <https://hpanel.hostinger.com/> και από εκεί να βρούμε τους nameservers μας.



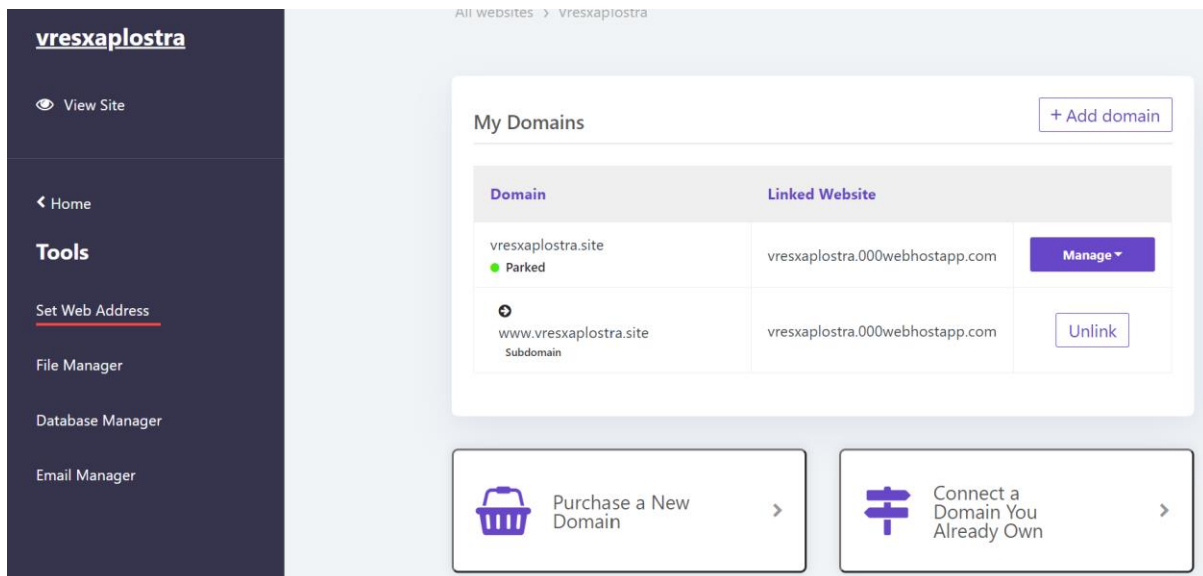
Εικόνα 3 : Πλατφόρμα hosting server

Υπάρχει πάντα η πρωτεύουσα διεύθυνση (ns01.000webhost.com) και η δευτερεύουσα (ns02.000webhost.com) για backup λειτουργία.

Αφού τα βρούμε και τα σημειώσουμε, θα πρέπει να μεταβούμε στο προμηθευτή του domain hosting μας και να του δηλώσουμε που θέλουμε να ανακατευθύνεται η ιστοσελίδα μας.

Στη δική μας περίπτωση θα μεταβούμε στο <https://www.000webhost.com/members/website/list> και θα επιλέξουμε από τα tools την επιλογή “set web address”. Στη συνέχεια επιλέγουμε “add a domain” και έπειτα “point a domain”.

Εκεί θα δηλώσουμε το domain name μας και θα συμπληρώσουμε τις δύο nameserver διευθύνσεις προς ανακατεύθυνση.



Εικόνα 4 : Πλατφόρμα domain name

Όταν ολοκληρωθεί η παραπάνω διαδικασία η εικόνα θα είναι όπως αυτή επάνω και θα μπορούμε να πούμε ότι έχουμε αγοράσει και φιλοξενούμε μια ιστοσελίδα με επιτυχία. Λάβετε υπόψιν βέβαια πως η αλλαγή αυτή μπορεί να πάρει μέχρι και 24 ώρες για να ολοκληρωθεί, χρόνο τον οποίο χρειάζονται οι πάροχοι για να ενημερώσουν τους dns servers τους.

Κεφάλαιο 3^ο Επεξήγηση κώδικα εργασίας

Σε αυτό το κεφάλαιο της παρούσας πτυχιακής εργασίας θα μελετήσουμε την αρχιτεκτονική του κώδικα που χρησιμοποιήθηκε στο πρακτικό κομμάτι της εργασίας.

3.1 Επεξήγηση βασικών αρχών

Αρχικά, πάμε να δούμε βασικά tags της HTML που χρησιμοποιήσαμε:

<!DOCTYPE html>	Ορίζει τον τύπο εγγράφου
<a>	Ορίζει έναν υπερσύνδεσμο
 	Κενή γραμμή

<button>	Κουμπί
<div>	Ορίζει μια ενότητα σε ένα έγγραφο
<footer>	Ορίζει ένα υποσέλιδο για ένα έγγραφο
<form>	Ορίζει μια φόρμα για είσοδο δεδομένων
<h1> μέχρι και <h6>	Ορίζει το μέγεθος των επικεφαλίδων
<head>	Περιέχει μεταδεδομένα / πληροφορίες για το έγγραφο
<header>	Ορίζει μια κεφαλίδα για ένα έγγραφο
	Ορίζει μια εικόνα
<input>	Ορίζει ένα στοιχείο ελέγχου εισόδου
<label>	Ορίζει μια ετικέτα για ένα στοιχείο <input>
	Ορίζει ένα στοιχείο λίστας
	Ορίζει μια λίστα με σειρά
<table>	Ορίζει έναν πίνακα
<script>	Ορίζει έναν κώδικα που τρέχει από την πλευρά του χρήστη
<section>	Ορίζει μια ενότητα σε ένα έγγραφο
<select>	Ορίζει μια αναπτυσσόμενη λίστα
<style>	Ορίζει πληροφορίες για το στυλ εγγράφου
<td>	Ορίζει ένα κελί σε έναν πίνακα
<tr>	Ορίζει μια σειρά σε έναν πίνακα

<title>	Ορίζει έναν τίτλο για το έγγραφο	[7]
---------	----------------------------------	-----

Εκτός των HTML tags όμως υπάρχουν και οι κανόνες-πρακτικές που χρησιμοποιούνται από την PHP.

Πάντα, ένα PHP κομμάτι κώδικα ξεκινάει με **<?php** και τελειώνει με **?>** .

Σαν πρακτική, συνηθίζεται να δημιουργούμε ένα έγγραφο τύπου php με το όνομα "config.php" όπου εκεί κρατάμε τις πληροφορίες και τους κωδικούς για τη σύνδεση με τη βάση δεδομένων μας.

Ένα τέτοιο αρχείο περιέχει τα εξής:

```
<?php

define('DB_SERVER', 'localhost');

define('DB_USERNAME', 'root');

define('DB_PASSWORD', '');

define('DB_NAME', 'mysql');

$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

if($link === false){

    die("ERROR: Could not connect. " . mysqli_connect_error());

}

?>
```

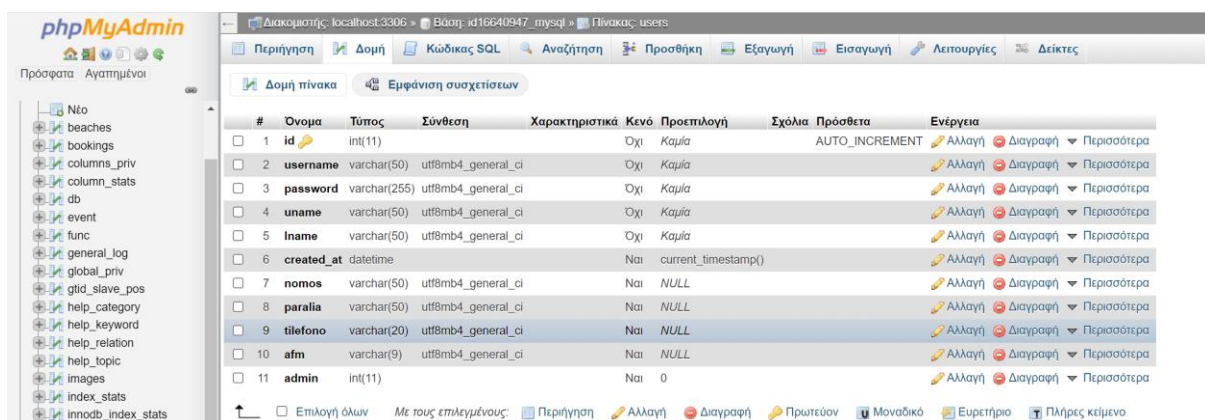
Αφού δημιουργήσουμε το αρχείο μας και το αποθηκεύσουμε στον ίδιο φάκελο με τα υπόλοιπα αρχεία της εργασίας μας, πρέπει να το συνδέσουμε και με τον κώδικα μας.

Για να το κάνουμε αυτό, στο php κομμάτι του εγγράφου μας θα πρέπει να το ορίσουμε ως εξής: `require_once "config.php";`. Έτσι, όταν ο κώδικας μας θα θελήσει να κάνει επικοινωνία με τη βάση θα χρησιμοποιήσει τις πληροφορίες και τα διαπιστευτήρια που περιέχονται μέσα στο έγγραφο αυτό.

3.2 Δημιουργία βάσης δεδομένων

Μπορούμε να χρησιμοποιήσουμε διαφορετικά προγράμματα για βάση δεδομένων, η πιο γνωστή παρόλα αυτά είναι η MySQL. Την MySQL (PHPMyAdmin) μπορούμε να την αποκτήσουμε είτε μέσω του πακέτου XAMPP είτε να την διαμορφώσουμε στον server ο οποίος θα μας κάνει host την ιστοσελίδα.

Παρακάτω θα δείτε το πως έχουμε διαμορφώσει την βάση μας καθώς και τους πίνακες που έχουμε χρησιμοποιήσει για τις ανάγκες της πλατφόρμας μας.



#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
1	id	int(11)			Όχι	Καμία		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
2	username	varchar(50)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
3	password	varchar(255)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
4	uname	varchar(50)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
5	lname	varchar(50)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
6	created_at	datetime			Ναι	current_timestamp()			Αλλαγή Διαγραφή Περισσότερα
7	nomos	varchar(50)	utf8mb4_general_ci		Ναι	NULL			Αλλαγή Διαγραφή Περισσότερα
8	paralia	varchar(50)	utf8mb4_general_ci		Ναι	NULL			Αλλαγή Διαγραφή Περισσότερα
9	tilefono	varchar(20)	utf8mb4_general_ci		Ναι	NULL			Αλλαγή Διαγραφή Περισσότερα
10	afm	varchar(9)	utf8mb4_general_ci		Ναι	NULL			Αλλαγή Διαγραφή Περισσότερα
11	admin	int(11)			Ναι	0			Αλλαγή Διαγραφή Περισσότερα

Όπως είδατε παραπάνω, φαίνεται ο πίνακας users με όλα τα πεδία του καθώς και το πρωτεύον κλειδί (id) που είναι και auto increment, δηλαδή παίρνει μόνο του και σειριακά τιμές.

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
1	bookingid	int(10)			Όχι	Καμία		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
2	created_at	datetime			Όχι	current_timestamp()			Αλλαγή Διαγραφή Περισσότερα
3	postoid	int(10)			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
4	beachid	int(10)			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
5	sunbeds	int(10)			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
6	umbrellas	int(10)			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
7	date	varchar(100)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
8	time	varchar(10)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
9	name	varchar(100)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
10	surname	varchar(100)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
11	email	varchar(100)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
12	phone	varchar(20)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
13	cost	varchar(10)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
14	paid	varchar(10)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
15	accepted	varchar(10)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
16	comments	text	utf8mb4_general_ci		Ναι				Αλλαγή Διαγραφή Περισσότερα

Παραπάνω θα δείτε τον πίνακα bookings και τα πεδία του. Εδώ θα δείτε, ότι σαν πρωτεύον κλειδί έχει οριστεί το bookingid το οποίο και αυτό είναι πάλι auto increment.

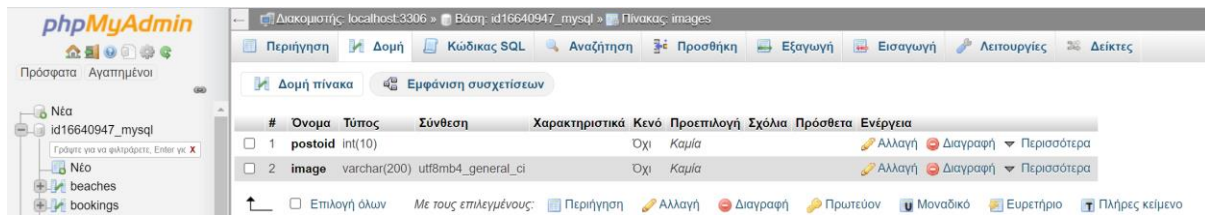
#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
1	beachid	int(4)			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
2	beachname	varchar(100)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
3	nomos	varchar(100)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

Στην εικόνα επάνω θα δείτε τον πίνακα beaches όπου και αυτός έχει κλειδί το beachid και είναι και αυτό auto increment.

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
1	postoid	int(10)			Όχι	Καμία		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
2	postoname	varchar(100)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
3	userid	int(10)			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
4	beachid	int(4)			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
5	latitude	varchar(10)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
6	longitude	varchar(10)	utf8mb4_general_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
7	xaplostra	float			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
8	ombrella	float			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
9	last_edit	datetime			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

Παραπάνω στην εικόνα βλέπετε αντίστοιχα τον πίνακα posto όπου κι αυτός σαν πρωτεύον κλειδί έχει το postoid και είναι κι αυτό auto increment.

Τελευταίος μας πίνακας είναι ο images στον οποίο θα αποθηκεύουμε τις εικόνες από τα τμήματα παραλίας (πόστο).



Όπως θα καταλάβατε αυτός ο πίνακας δεν έχει πρωτεύον κλειδί, αυτό συμβαίνει γιατί είναι ένας βοηθητικός πίνακας που αποθηκεύεται η εικόνα κάθε πόστου.

3.3 Εγγραφή νέου χρήστη

Για να εγγράψουμε ένα νέο χρήστη στη βάση δεδομένων μας, χρειαζόμαστε από αυτόν να συμπληρώσει κάποια απαραίτητα στοιχεία, όπως για παράδειγμα στην περίπτωση μας είναι τα

εξής: Όνομα, Επώνυμο, Email και κωδικό πρόσβασης. Τα στοιχεία αυτά τα ζητάμε χρησιμοποιώντας την παρακάτω φόρμα που βρίσκεται στην σελίδα signup.php

```
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
  <div class="col-md-6">
    <h3 class="section-title">Ελάτε να γίνουμε συνεργάτες!</h3>
    <p>Συμπληρώστε τα απαραίτητα στοιχεία και εμείς θα έρθουμε σε επικοινωνία μαζί σας.</p>
  </div>
  <div class="col-md-6">
    <div class="row">
      <div class="col-md-12">
        <div class="form-group <?php echo (!empty($uname_err)) ? 'has-error' : ''; ?>">
          <label>Όνομα</label>
          <input type="text" name="uname" class="form-control" value="<?php echo $uname; ?>" >
          <span class="help-block"><?php echo $uname_err; ?></span>
        </div>
      </div>
      <div class="col-md-12">
        <div class="form-group <?php echo (!empty($lname_err)) ? 'has-error' : ''; ?>">
          <label>Επώνυμο</label>
          <input type="text" name="lname" class="form-control" value="<?php echo $lname; ?>" >
          <span class="help-block"><?php echo $lname_err; ?></span>
        </div>
      </div>
      <div class="col-md-12">
        <div class="form-group <?php echo (!empty($username_err)) ? 'has-error' : ''; ?>">
          <label>Email</label>
          <input type="text" name="username" class="form-control" value="<?php echo $username; ?>" >
          <span class="help-block"><?php echo $username_err; ?></span>
        </div>
      </div>
      <div class="col-md-12">
        <div class="form-group <?php echo (!empty($password_err)) ? 'has-error' : ''; ?>">
          <label>Κωδικός</label>
          <input type="password" name="password" class="form-control" value="<?php echo $password; ?>" >
          <span class="help-block"><?php echo $password_err; ?></span>
        </div>
      </div>
      <div class="col-md-12">
        <div class="form-group <?php echo (!empty($confirm_password_err)) ? 'has-error' : ''; ?>">
          <label>Επλήρωση κωδικού</label>
          <input type="password" name="confirm_password" class="form-control" value="<?php echo $confirm_password; ?>" >
          <span class="help-block"><?php echo $confirm_password_err; ?></span>
        </div>
      </div>
      <div class="col-md-12">
        <div class="form-group">
          <input type="submit" value="Εγγραφή" class="btn btn-primary">
        </div>
      </div>
    </div>
  </div>
</form>
```

Τα δεδομένα αυτά εκχωρούνται σε μεταβλητές της php όπου η κάθε μια έχει το όνομα της το οποίο και παίρνει από το μαυρισμένο κομμάτι του κώδικα: `<input type="text" name="uname" class="form-control" value="<?php echo $uname; ?>" >`. Αφού λοιπόν ο χρήστης τα υποβάλλει, τα δεδομένα αυτά περνάνε από έναν έλεγχο που ορίζουμε παρακάτω. Ένας έλεγχος

που κάνουμε είναι για παράδειγμα το email που μας έδωσε ο χρήστης να μην υπάρχει ήδη στη βάση μας. Αυτός ο έλεγχος πραγματοποιείται με τον παρακάτω κώδικα:

```
if(empty(trim($_POST["username"]))){
    $username_err = "Παρακαλώ συμπληρώστε Email!";
}

else{
    // Prepare a select statement
    $sql = "SELECT username FROM users WHERE username = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "s", $param_username);

        // Set parameters
        $param_username = trim($_POST["username"]);

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            /* store result */
            mysqli_stmt_store_result($stmt);

            if(mysqli_stmt_num_rows($stmt) == 1){
                $username_err = "Αυτό το Email χρησιμοποιείται ήδη!";
            } else{
                $username = trim($_POST["username"]);
            }
        } else{
            echo "Ωπα! Κάτι πήγε στραβά. Παρακαλώ προσπαθήστε ξανά αργότερα.";
        }
    }

    // Close statement
    mysqli_stmt_close($stmt);
}
```

Εκτός του ελέγχου για το email, ένας επιπλέον έλεγχος που πραγματοποιούμε είναι για τον κωδικό. Όπως θα δείτε και στον παραπάνω κώδικα, χρησιμοποιούμε δύο πεδία, έναν για τον κωδικό και άλλο ένα για την επαλήθευση του κωδικού ώστε να σιγουρευτούμε ότι ο χρήστης έχει πληκτρολογήσει το ίδιο και στα δύο αυτά πεδία. Ο έλεγχος γίνεται χρησιμοποιώντας τον παρακάτω κώδικα:


```
// Validate password
if(empty(trim($_POST["password"]))){
    $password_err = "Παρακαλώ συμπληρώστε Κωδικό!";
} elseif(strlen(trim($_POST["password"])) < 6){
    $password_err = "Password must have atleast 6 characters.";
} else{
    $password = trim($_POST["password"]);
}

// Validate confirm password
if(empty(trim($_POST["confirm_password"]))){
    $confirm_password_err = "Παρακαλώ συμπληρώστε ξανά τον Κωδικό!";
} else{
    $confirm_password = trim($_POST["confirm_password"]);
    if(empty($password_err) && ($password != $confirm_password)){
        $confirm_password_err = "Οι κωδικοί δεν ταιριάζουν!";
    }
}
}
```

Εφόσον όλα τα παραπάνω στάδια ξεπεραστούν, τότε τα εισαχθέντα δεδομένα αποστέλλονται στη βάση δεδομένων μας προς μόνιμη αποθήκευση. Αυτό γίνεται με τον παρακάτω κώδικα:

```
// Check input errors before inserting in database
if(empty($username_err) && empty($password_err) && empty($confirm_password_err)&& empty($uname_err)&&empty($lname_err)&&empty($email_err)){

    // Prepare an insert statement
    $sql = "INSERT INTO users (uname, lname, username, password) VALUES (?, ?, ?, ?)";
    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "ssss", $param_uname, $param_lname, $param_username, $param_password);

        // Set parameters
        $param_uname = $uname;
        $param_lname = $lname;
        $param_username = $username;
        $param_password = password_hash($password, PASSWORD_DEFAULT); // Creates a password hash

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            // Redirect to login page
            header("location: login.php");
        } else{
            echo "Όπα! Κάτι πήγε στραβά. Παρακαλώ προσπαθήστε ξανά αργότερα.";
        }
    }

    // Close statement
    mysqli_stmt_close($stmt);
}

// Close connection
mysqli_close($link);
```

Εδώ θέλω να λάβετε υπόψη σας, ότι ο κωδικός που αποστέλλεται στην βάση είναι της μορφής hashed δηλαδή δεν είναι απλά το πεδίο του κωδικού που λάβαμε αλλά είναι σε κρυπτογραφημένη μορφή για περισσότερη ασφάλεια.

3.4 Σύνδεση χρήστη

Για να ξεκινήσουμε μια συνεδρία μεταξύ του server και του χρήστη μας π.χ. για να γίνει σύνδεση στο λογαριασμό του, χρησιμοποιούμε τον κώδικα `session_start()`;, ενώ όταν θα θελήσουμε να σταματήσουμε την συνεδρία π.χ. γιατί ο πελάτης θέλει να κάνει αποσύνδεση από τον λογαριασμό του, χρησιμοποιούμε τον κώδικα `session_destroy()`; .

Κατά τη διαδικασία σύνδεσης χρησιμοποιούμε τον παρακάτω κώδικα για να καταγράψουμε κάποια στοιχεία στη συνεδρία του χρήστη που θα χρειαστούμε αργότερα.

```
$_SESSION["loggedin"] = true; (ορίζουμε την μεταβλητή loggedin της συνεδρίας με την τιμή true.)
```

```
$_SESSION["username"] = $username; (ορίζουμε την μεταβλητή username της συνεδρίας με την τιμή username.)
```

```
$_SESSION["uname"] = $uname; (ορίζουμε την μεταβλητή uname της συνεδρίας με την τιμή uname.)
```

```
$_SESSION["lname"] = $lname; (ορίζουμε την μεταβλητή lname της συνεδρίας με την τιμή lname.)
```

```
$_SESSION["admin"] = $admin; (ορίζουμε την μεταβλητή admin της συνεδρίας με την τιμή admin.)
```

```
$_SESSION["administrator"] = $administrator; (ορίζουμε την μεταβλητή administrator της συνεδρίας με την τιμή administrator.)
```

Όταν έχει πραγματοποιηθεί η σύνδεση του χρήστη και θέλουμε για παράδειγμα να τον ανακατευθύνουμε με μια άλλη σελίδα, τότε χρησιμοποιούμε την εντολή `header("location:`

adminportal.php"); και στη θέση του location γράφουμε τη σελίδα που θέλουμε να μεταβεί ο χρήστης.

3.5 Καταχώρηση κράτησης

Για να πραγματοποιηθεί μια νέα κράτηση στη βάση μας χρησιμοποιούμε ερωτήσεις προς τον χρήστη ώστε να πάρουμε από αυτόν δεδομένα μέσω φορμών. Στην αρχική σελίδα του ζητείται να επιλέξει τον Νομό στον οποίο θέλει να βρει παραλία.

```
<div class="col-sm-12 mt">
    <form action="search.php" method="GET">
        <section>
            <label for="class">Νομός:</label>
            <br>
            <?php
echo "<select id='nomos' name='nomos' class='cs-select cs-skin-border' required";
echo '<option value="">Επιλογή Νομού</option>';

while ($row = mysqli_fetch_array($result)) {
    unset($nomos);
    $nomos= $row['nomos'];
    echo '<option value="'. $nomos. '">'. $nomos. '</option>';
}
echo "</select>";
?>

        </section>

        <br><br>
        <div class="col-xs-12">
            <input type="submit" class="btn btn-primary btn-block" value="Αναζήτηση">
        </div>
    </form>
</div>
```

Όπως θα δείτε στον παραπάνω κώδικα, χρησιμοποιούμε μια φόρμα η οποία έχει ως ενέργεια να μας ανακατευθύνει στην σελίδα search.php (με χρήση GET) όταν πατήσουμε το κουμπί submit, περνώντας σαν πληροφορία την τιμή του Νομού που θα μας δώσει ο χρήστης. Η τιμή του Νομού γεμίζει μια μέθοδο select όπου κάθε option της είναι και ένα αποτέλεσμα του κώδικα sql: "SELECT distinct nomos from beaches"; .

Στη σελίδα search.php λοιπόν χρησιμοποιώντας τον κώδικα \$nomos = \$_GET['nomos']; παίρνουμε την τιμή του GET και την εκχωρούμε στην μεταβλητή \$nomos. Με αυτό τον τρόπο

μπορούμε στη συνέχεια να τη χρησιμοποιήσουμε για να βρούμε τις παραλίες που αντιστοιχούν στον επιλεγθέντα Νομό.

Αυτό γίνεται με τον κώδικα `$sql = "SELECT beachname,beachid from beaches where nomos ='$nomos'; "`; όπου επιστρέφουμε πάλι με τον ίδιο τρόπο μέσω ενός select τις παραλίες που υπάρχουν στη βάση μας και αντιστοιχούν στη μεταβλητή `$nomos`.

```
<div class="col-sm-12 mt-2">
    <form action="searchmid.php" method="GET">
    <section>
        <label for="class">Παραλία:</label>
        <br>
        <?php
echo "<select id='beachid' name='beachid' class='cs-select cs-skin-border' required";
echo '<option value="">Επιλογή Παραλίας</option>';

while ($row = mysqli_fetch_array($result)) {
    unset($beachid, $beachname);
    $beachid= $row['beachid'];
    $beachname.= $row['beachname'];
    echo '<option value="'. $beachid. '">'. $beachname. '</option>';
}
echo "</select>";
?>

    </section>
    <input type="hidden" name="beachname" value="<?php echo $beachname ?>">
    <br><br>
    <div class="col-xs-12">
    <input type="submit" class="btn btn-primary btn-block" value="Αναζήτηση">
    </div>
    </form>
```

Αυτή τη φορά όταν συμπληρωθεί η φόρμα, όπως μπορούμε να δούμε και από τον παραπάνω κώδικα θα μεταφερθούμε στη σελίδα `searchmid.php` (πάλι με χρήση GET) όπου εκεί σύμφωνα με την επιλογή της παραλίας θα πρέπει να επιλέξουμε τμήμα παραλίας ή αλλιώς πόστο. Κάθε τμήμα παραλίας ανήκει και σε έναν συνεργάτη.

Σε αυτή την σελίδα μέσω `php` θα δημιουργηθούν δυναμικά εικόνες όπου θα είναι της μορφής `form` και θα φέρουν πληροφορίες για τις τιμές, την τοποθεσία και την εικόνα του τμήματος της παραλίας του συνεργάτη.

```

<?php
while ($row = mysqli_fetch_array($result)) {
    echo '<form action="searchfinal.php" method="GET">';
    echo '<input type="hidden" name="beachid" value="'. $paraliaid. '">';
    echo '<input type="hidden" name="postoid" value="'. $row['postoid']. '">';
    $var1 = $row['postoid'];
    $sql1= "SELECT image FROM images WHERE postoid ='$var1' ";
    $result1 = mysqli_query($link, $sql1);
    echo '<div class="col-md-4 col-sm-6 fh5co-tours animate-box" data-animate-effect="fadeIn">';
    while ($row1 = mysqli_fetch_array($result1)) {
        echo "<div id='img_div'>";
        echo "<img src='images/'. $row1['image']. "' class='img-responsive'>";
        echo '<div class="desc">';
    }
    echo '<span></span>';
    echo '<h3>';
    echo $row['postoname'];
    echo '</h3>';
    echo '<span>Τιμή Ξαπλώστρας: ';
    echo $row['xaplostra'];
    echo '€</span>';
    echo '<span>Τιμή Ομπρέλας: ';
    echo $row['ombrella'];
    echo '€</span>';
    echo '<span>Τοποθεσία: <a href="http://maps.google.com/maps?q=';
    echo $row['latitude'];
    echo ',';
    echo $row['longitude'];
    echo '"target="_blank"> Google maps </a> </span>';
    echo '<input type="submit" class="btn btn-primary btn-outline" value="Επιλογή"></input>';
    echo '</div>';
    echo '</div>';
    echo '</div>';
    echo '</form>';
}
?>

```

Εδώ, αφού ο χρήστης επιλέξει το κουμπί «Επιλογή» του αντίστοιχου τμήματος παραλίας, θα μεταφερθεί στη σελίδα searchfinal.php (πάλι με χρήση GET) όπου θα περαστούν και οι τιμές beachid και beachname.

Στην τελευταία σελίδα της κράτησης, ο χρήστης θα πρέπει να συμπληρώσει τα στοιχεία που του ζητούνται στην παρακάτω φόρμα.

```

<form action="make_booking.php" method="POST">
  <div class="row">
    <div class="col-xxs-12 col-xs-6 mt">
      <div class="input-field">
        <label for="date-start">'Όνομα:</label>
        <input type="text" class="form-control" name="name" required/>
      </div>
    </div>
    <div class="col-xxs-12 col-xs-6 mt">
      <div class="input-field">
        <label for="date-start">Επώνυμο:</label>
        <input type="text" class="form-control" name="surname" required/>
      </div>
    </div>
    <div class="col-xxs-12 col-xs-6 mt">
      <div class="input-field">
        <label for="date-start">Email:</label>
        <input type="email" class="form-control" name="email" required/>
      </div>
    </div>
    <div class="col-xxs-12 col-xs-6 mt">
      <div class="input-field">
        <label for="date-start">Κινητό(με προθ. χώρας):</label>
        <input id="phone" type="tel" class="form-control" name="phone" placeholder="+3069XXXXXXX" required/>
      </div>
    </div>
    <div class="col-xxs-12 col-xs-6 mt">
      <div class="input-field">
        <label for="date-start">Ημερομηνία:</label>
        <input type="date" class="form-control" id="date" name="date" placeholder="mm/dd/yyyy" min="<?php echo date("Y-m-d"); ?>" required/>
      </div>
    </div>
    <div class="col-xxs-12 col-xs-6 mt">
      <div class="md-form">
        <label for="time-start">Ώρα Άφίξης:</label>
        <input type="time" class="form-control" name="time" required/>
      </div>
    </div>
    <div class="col-xxs-12 col-xs-6 mt">
      <section>
        <label for="class">Ξαπλώστρες:</label>
        <select class="cs-select cs-skin-border" name="sunbeds" required>
          <option value="0">0</option>
          <option value="1">1</option>
          <option value="2">2</option>
          <option value="3">3</option>
          <option value="4">4</option>
          <option value="5">5</option>
          <option value="6">6</option>
          <option value="7">7</option>
          <option value="8">8</option>
          <option value="9">9</option>
          <option value="10">10</option>
        </select>
      </section>
    </div>
  </div>
  <div class="col-xxs-12 col-xs-6 mt">
    <section>
      <label for="class">Ομπρέλες:</label>
      <select class="cs-select cs-skin-border" name="umbrellas" required>
        <option value="0">0</option>
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
      </select>
    </section>
  </div>
  <input type="hidden" name="postoid" value="<?php echo $postoid ?>">
  <input type="hidden" name="beachid" value="<?php echo $beachid ?>">
  <div class="col-xs-12">
    <input type="submit" class="btn btn-primary btn-block" value="Κάνε Κράτηση">
  </div>
</form>

```

```

  </select>
</section>
</div>
<div class="col-xxs-12 col-xs-6 mt">
  <section>
    <label for="class">Ομπρέλες:</label>
    <select class="cs-select cs-skin-border" name="umbrellas" required>
      <option value="0">0</option>
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
      <option value="5">5</option>
      <option value="6">6</option>
      <option value="7">7</option>
      <option value="8">8</option>
      <option value="9">9</option>
      <option value="10">10</option>
    </select>
  </section>
</div>
<input type="hidden" name="postoid" value="<?php echo $postoid ?>">
<input type="hidden" name="beachid" value="<?php echo $beachid ?>">
<div class="col-xs-12">
  <input type="submit" class="btn btn-primary btn-block" value="Κάνε Κράτηση">
</div>
</form>

```

Τα στοιχεία αυτά είναι: Όνομα, Επώνυμο, Email, Κινητό με το πρόθεμα της χώρας του (π.χ. +30), αριθμό ξαπλωστών και αριθμό ομπρελών που επιθυμεί. Τα στοιχεία αυτά, αποστέλλονται μέσω POST σε μια php κρυφή για το χρήστη σελίδα, δουλειά της οποίας είναι να τα καταχωρεί μόνιμα μέσα στη βάση δεδομένων μας καθώς και να ενημερώνει το συνεργάτη με γραπτό μήνυμα sms ότι έχει δημιουργηθεί νέα κράτηση για το πόστο του και πρέπει να την ελέγξει.

3.6 Δημιουργία κράτησης στη βάση

Αφότου λάβουμε τα παραπάνω στοιχεία, η φόρμα μας ανακατευθύνει στη κρυφή σελίδα make_booking.php. Με τον όρο κρυφή εννοούμε μια σελίδα η οποία δεν αναπαράγει κάποιο γραφικό περιβάλλον στο χρήστη, αλλά η χρησιμότητα της είναι να κάνει πράγματα για την πλατφόρμα.

```
$cost = $bid = $xaplostra = $ombrella = $userid = $tilefono = "";
$postoid = $_POST['postoid'];
$beachid = $_POST['beachid'];
$date = $_POST['date'];
$time = $_POST['time'];
$sunbeds = $_POST['sunbeds'];
$umbrellas = $_POST['umbrellas'];
$name = $_POST['name'];
$surname = $_POST['surname'];
$email = $_POST['email'];
$phone = $_POST['phone'];
$paid = "NO";
$accepted = "NO";
$sql0 = "SELECT xaplostra,ombrella,userid FROM posto WHERE postoid = '$postoid' ";
$result = mysqli_query($link, $sql0);
while($row = $result->fetch_assoc()) {
    $xaplostra = $row['xaplostra'];
    $ombrella = $row['ombrella'];
    $userid = $row['userid'];
}
$cost = $xaplostra * $sunbeds + $ombrella * $umbrellas ;

$sql1 = "INSERT INTO bookings (postoid,beachid,sunbeds,umbrellas,date,time,name,surname,email,phone,cost,paid,accepted) VALUES ('$postoid','$beachid','$sunbeds','$umbrellas','$date','$time','$name','$surname','$email','$phone','$cost','$paid','$accepted')";
mysqli_query($link,$sql1);
```

Αρχικά, όπως θα δείτε στον παραπάνω κώδικα, εκχωρούμε τα δεδομένα των τιμών της φόρμας (POST) σε τοπικές μεταβλητές καθώς και γίνεται αρχικοποίηση μερικών μεταβλητών όπως για παράδειγμα της \$cost με το κενό και της \$paid με την τιμή “NO”. Στη συνέχεια, με την εντολή \$sql0 παίρνουμε από τη βάση μας (posto) τις τιμές της ξαπλώστρας και της ομπρέλας που έχει ορίσει ο συνεργάτης, ώστε να υπολογίσουμε το συνολικό κόστος της κράτησης. Επίσης ζητάμε και από την sql να μας δώσει το userid του συνεργάτη γιατί θα το χρειαστούμε

στην πορεία. Αφού γίνει και ο υπολογισμός του κόστους, τότε τα δεδομένα εισάγονται μέσω της \$sql1 στη βάση για μόνιμη αποθήκευση.

```
$sql2 = "SELECT bookingid FROM bookings WHERE email = '$email'";  
$result2 = mysqli_query($link, $sql2);  
while($row = $result2->fetch_assoc()) {  
    $bid = $row['bookingid'];  
}  
$sql3 = "SELECT tilefono FROM users WHERE id = '$userid'";  
$result3 = mysqli_query($link, $sql3);  
while($row = $result3->fetch_assoc()) {  
    $recipients = [ $row['tilefono'] ];  
}
```

Αφού γίνει η καταχώρηση, η παραπάνω \$sql2 θα μας επιστρέψει στη μεταβλητή \$bid τον αριθμό της κράτησης από τον πίνακα bookings της βάσης μας. Επίσης η \$sql3 θα μας επιστρέψει στη μεταβλητή \$recipients, το τηλέφωνο του συνεργάτη ώστε αν του αποστείλουμε μήνυμα.

Η αποστολή του μηνύματος γίνεται χρησιμοποιώντας την πλατφόρμα <https://gatewayapi.com/app> η οποία προσφέρει αποστολή μηνυμάτων μέσω rhr api με κόστος περίπου 0,03 ευρώ ανά sms.

```
$url = "https://gatewayapi.com/rest/mtsms";  
$api_token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";  
  
$json = [  
    'sender' => 'XAPLOSTRA',  
    'message' => "New booking has been added. Please login to respond ASAP vresxaplostra.site/edit-booking.php?bookingid=$bid .",  
    'recipients' => [],  
];  
foreach ($recipients as $msisdn) {  
    $json['recipients'][] = ['msisdn' => $msisdn];  
}  
  
//Make and execute the http request  
//Using the built-in 'curl' library  
$ch = curl_init();  
curl_setopt($ch,CURLOPT_URL, $url);  
curl_setopt($ch,CURLOPT_HTTPHEADER, array("Content-Type: application/json"));  
curl_setopt($ch,CURLOPT_USERPWD, $api_token . ":");  
curl_setopt($ch,CURLOPT_POSTFIELDS, json_encode($json));  
curl_setopt($ch,CURLOPT_RETURNTRANSFER, true);  
$result = curl_exec($ch);  
curl_close($ch);  
  
//$json = json_decode($result);  
header("location: thankyou.php?bookingid=$bid");  
?>
```


Παραπάνω βλέπετε την αρχικοποίηση των μεταβλητών \$url, \$api_token, τα οποία χρησιμοποιούνται για επικοινωνία και ταυτοποίηση με την πλατφόρμα των sms καθώς και την μεταβλητή \$json. Η μεταβλητή αυτή λειτουργεί σαν ένα μικρό αρχείο json έχοντας στο πεδίο sender το όνομα του αποστολέα του μηνύματος, στο πεδίο message το κείμενο του μηνύματος και στο πεδίο recipients τον αριθμό ή τους αριθμούς των παραληπτών του.

Όπως θα παρατηρήσετε, στο πεδίο message υπάρχει ένας σύνδεσμος που ανακατευθύνει τον παραλήπτη σε μια συγκεκριμένη σελίδα την edit-booking.php περνώντας της την τιμή του αριθμού κράτησης μέσω get. Τέλος, με χρήση της ενσωματωμένης βιβλιοθήκης της php curl, κωδικοποιείται το μήνυμα και αποστέλλεται το https request στους servers της πλατφόρμας των μηνυμάτων. Σε όλο αυτό, το μόνο που θα καταλάβει ο χρήστης από όταν συμπληρώσει τα στοιχεία της κράτησης του είναι μια τελική ανακατεύθυνση στη σελίδα `thankyou.php?bookingid=$bid` όπου εκεί περνώντας με get το bookingid θα ενημερωθεί ότι η κράτηση του έχει καταχωρηθεί και θα του παρουσιαστεί και ο αριθμός της.

3.7 Προβολή κρατήσεων συνεργάτη

Ο συνεργάτης αφού έχει συνδεθεί και πιστοποιηθεί χρησιμοποιώντας τα διαπιστευτήρια του, μπορεί να επιλέξει να μεταβεί στη πλατφόρμα συνεργατών (adminportal.php) και από εκεί μπορεί να κάνει κλικ στο σύνδεσμο my-bookings.php. Σε αυτή την σελίδα χρησιμοποιώντας τον παρακάτω κώδικα, παρουσιάζουμε στο συνεργάτη έναν πίνακα με όλες τις κρατήσεις του.

```

<?php
require_once "config.php";
session_start();
$user_id = $admin=$current_x=$current_o =$name ="";
$user_id= $_SESSION["id"];
if(!(isset($_SESSION["admin"]) && $_SESSION["admin"] === 1) )
{
    header("location: index.php");
    exit;
}
$sql0 = "SELECT paralia FROM users WHERE id = '$user_id' ";
$result = mysqli_query($link, $sql0);
while($row = $result->fetch_assoc()) {
    $name = $row['paralia'];
}
?>
<!DOCTYPE html>

```

Για να πραγματοποιήσει λοιπόν κάποιος πρόσβαση στη σελίδα αυτή εξασφαλίζοντας τη μέγιστη ασφάλεια, κάνουμε έναν έλεγχο σε επίπεδο session για το εάν ο συνδεδεμένος χρήστης είναι και πιστοποιημένος για να διαχειριστεί τις κρατήσεις του. Εάν δεν είναι admin ο συνδεδεμένος χρήστης, τότε οδηγείται απευθείας στην index.php απαγορεύοντας του την πρόσβαση στη διαχείριση των κρατήσεων μέσω της σελίδας αυτής.

Αφότου γίνει ο παραπάνω έλεγχος, καταχωρούμε το session id του στην τοπική μεταβλητή user_id όπου θα χρησιμοποιήσουμε παρακάτω για να προβάλλουμε μόνο τις κρατήσεις του συγκεκριμένου συνεργάτη.

```

$sql = " SELECT bookingid, sunbeds, umbrellas, date, time, name, surname, email, phone, cost, paid, accepted, comments from bookings where postoid = '$name' ORDER BY
bookingid DESC";
$result = $link->query($sql);

$result = mysqli_query($link, $sql); // First parameter is just return of "mysqli_connect()" function
if ($result->num_rows > 0) {
    echo "<tr>";
    echo "<th>BookingID</th>";
    echo "<th>Σανλώςτες</th>";
    echo "<th>Ομπρέλες</th>";
    echo "<th>Ημερομηνία</th>";
    echo "<th>Ώρα</th>";
    echo "<th>Όνομα</th>";
    echo "<th>Επώνυμο</th>";
    echo "<th>Email</th>";
    echo "<th>Τηλέφωνο</th>";
    echo "<th>Συνολικό Ποσό</th>";
    echo "<th>Πληρώθηκε</th>";
    echo "<th>Αποδοχή</th>";
    echo "<th>Σχόλια</th>";
    echo "<th>Επεξεργασία</th>";
    echo "</tr>";
    while ($row = mysqli_fetch_assoc($result)) { // Important line !!! Check summary get row on array ..
        foreach ($row as $field => $value) { // I you want you can right this line like this: foreach($row as $value) {
            echo "<td>";
            echo " $value "; // I just did not use "htmlspecialchars()" function.
            echo "</td>";
        }
        if ($row['paid'] == 'NO'){
            echo "<form action='edit-booking.php' method='GET'>";
            echo "<input type='hidden' name='bookingid' value='". $row['bookingid'] .'>";
            echo "<td>";
            echo "<input type='submit' value='Επεξεργασία' ";
            echo "</td>";
            echo "</form>";
            echo "</tr>";
        }
        elseif ($row['paid'] == 'YES'){
            echo "<td>";
            echo "<input type='submit' value='Επεξεργασία' readonly' ";
            echo "</td>";
            echo "</tr>";
        }
    }
}

```

Χρησιμοποιώντας την \$sql0 παίρνουμε το αναγνωριστικό του τμήματος της παραλίας του συνεργάτη και το δίνουμε στην \$sql1 ώστε να μας φέρει τις εγγραφές στη βάση bookings που υπάρχουν για το συγκεκριμένο τμήμα παραλίας.

Τα δεδομένα που θα επιστραφούν, παρουσιάζονται στο συνεργάτη με τη μορφή πίνακα. Ενδεικτικά προβάλλουμε τον αριθμό της κράτησης, τις ξαπλώστρες, τις ομπρέλες, το Όνομα, το Επώνυμο, το email και το κινητό του ατόμου που έκανε την κράτηση, καθώς και το συνολικό ποσό κράτησης, εάν πληρώθηκε, εάν έχει γίνει αποδοχή από τον συνεργάτη, τυχόν σχόλια και τέλος ένα κουμπί τύπου submit όπου μας ανακατευθύνει στη σελίδα edit-booking.php με όρισμα (GET) τον συγκεκριμένο αριθμό κράτησης.

3.8 Διαχείριση κρατήσεων συνεργάτη

Εάν ο χρήστης πατήσει στην επεξεργασία μιας κράτησης, τότε μεταφέρεται στη σελίδα edit-booking.php όπου εκεί θα δει αναλυτικότερα όλες τις πληροφορίες της κράτησης. Επίσης, εδώ

του δίνεται η δυνατότητα να αποδεχθεί ή να απορρίψει την κράτηση αυτή, καθώς και να προσθέσει σχόλια προς το χρήστη που την έκανε.

Όλα τα παραπάνω γίνονται μέσω της παρακάτω φόρμας:

```
<form action="edit_booking.php" method="POST">
  <input type="hidden" name="booking" value="<?php echo $bid ?>" >
  <input type="hidden" name="telefono" value="<?php echo $phone ?>" >
  <div class="row">
    <div class="col-md-6">
      <label>Κατάσταση Αποδοχής</label>
      <select name="acc" class="form-control" required="true">
        <option value=""></option>
        <option value="YES">ΝΑΙ</option>
        <option value="NO">ΟΧΙ</option>
      </select>
    </div>
    <div class="col-md-12">
      <div class="form-group">
        <textarea name="comm" class="form-control" cols="30" rows="7" placeholder="Σχόλια *Μόνο στα Αγγλικά ή με
        Λατινικούς χαρακτήρες*" ></textarea>
      </div>
      <div class="col-md-12">
        <div class="form-group" align="center">
          <input type="submit" value="ΥΠΟΒΟΛΗ" class="btn btn-primary">
        </div>
      </div>
    </div>
  </form>
```

Η φόρμα αυτή έχει ένα select το οποίο δέχεται σαν είσοδο τις τιμές “YES και “NO” καθώς και ένα πεδίο τύπου textarea για τα σχόλια. Όταν πατηθεί το κουμπί “ΥΠΟΒΟΛΗ”, τότε όπως θα δείτε ο συνεργάτης μεταφέρεται στη σελίδα edit_booking.php αποστέλλοντας τα δεδομένα μέσω της μεθόδου POST.

Η σελίδα αυτή, πάλι, είναι μια κρυφή σελίδα όπου έχει σκοπό να καταχωρεί τα δεδομένα της φόρμας στη βάση καθώς και να ενημερώνει με γραπτό μήνυμα sms το χρήστη για να πραγματοποιήσει την πληρωμή του.

```

$recipients = [$_POST ['telefono'] ];

$update = "UPDATE bookings set accepted='$acc', comments='$comm'where bookingid='$booking' ";
mysqli_query($link,$update);

if ($acc== 'NO'){
    $json = [
        'sender' => 'XAPLOSTRA',
        'message' => "Your booking #$booking has been rejected. Comments: $comm . Thank you for using VresXaplostra.",
        'recipients' => [],
    ];
    foreach ($recipients as $msisdn) {
        $json['recipients'][] = ['msisdn' => $msisdn];
    }

    //Make and execute the http request
    //Using the built-in 'curl' library
    $ch = curl_init();
    curl_setopt($ch,CURLOPT_URL, $url);
    curl_setopt($ch,CURLOPT_HTTPHEADER, array("Content-Type: application/json"));
    curl_setopt($ch,CURLOPT_USERPWD, $api_token.");");
    curl_setopt($ch,CURLOPT_POSTFIELDS, json_encode($json));
    curl_setopt($ch,CURLOPT_RETURNTRANSFER, true);
    $result = curl_exec($ch);
    curl_close($ch);
    print($result);
    $json = json_decode($result);
}

elseif ($acc== 'YES') {
    $json = [
        'sender' => 'XAPLOSTRA',
        'message' => "Your booking #$booking has been confirmed! Pay here: /Ptixiaki/checkout.php?bookingid=$booking.",
        'recipients' => [],
    ];
    foreach ($recipients as $msisdn) {
        $json['recipients'][] = ['msisdn' => $msisdn];
    }
}

```

Έτσι, χρησιμοποιώντας την sql εντολή της \$update, αλλάζουν τα πεδία accepted και comments της κράτησης στον πίνακα bookings. Επίσης, αφού αλλάζουν, αποστέλλεται ένα sms στο χρήστη που έκανε την κράτηση ώστε με ένα σύνδεσμο να πραγματοποιήσει την πληρωμή του εφόσον η κράτηση έχει γίνει αποδοχή ή αντίστοιχα ένα μήνυμα απόρριψης της κράτησης του με τα σχόλια του συνεργάτη.

3.9 Διαχείριση τιμών συνεργάτη

Ο συνεργάτης έχει τη δυνατότητα μέσα από τη σελίδα adminportal.php να επιλέξει να μεταβεί στη σελίδα διαχείρισης τιμών για το τμήμα της παραλίας του. Η όλη διαδικασία πραγματοποιείται μέσω της σελίδας edit-price.php.

Στη σελίδα αυτή, προβάλλονται με την χρήση του παρακάτω κώδικα οι υπάρχουσες τιμές για τις ξαπλώστρες και τις ομπρέλες του, καθώς και η τελευταία ημερομηνία που έκανε την αλλαγή τους.

```

$user_id= $_SESSION["id"];
if(!isset($_SESSION["admin"]) && $_SESSION["admin"] === 1) )
{
    header("location: index.php");
    exit;
}
$sql = " SELECT xaplostra,ombrella,last_edit from posto where userid ='$user_id'";
$result = $link->query($sql);

if ($result->num_rows > 0) {
// output data of each row
while($row = $result->fetch_assoc()) {

    $current_x = $row["xaplostra"] ;
    $current_o = $row["ombrella"] ;
    $lastedit = $row["last_edit"] ;
}
}
?>

```

Αφού γίνει ο έλεγχος ξανά για την πιστοποίηση του συνεργάτη, μέσω της \$sql εκχωρούμε στις αντίστοιχες μεταβλητές τις τιμές που μας επιστρέφονται από τον πίνακα posto για τον συγκεκριμένο συνεργάτη.

```

<div class="row animate-box" align="center">
<h3> Υφιστάμενες Τιμές</h3>
<h3> Ξαπλώστρα: <?php echo $current_x ?>€ Ομπρέλα: <?php echo $current_o ?>€ </h3>
<h4> Τελευτ. επεξεργασία: <?php echo $lastedit ?> </h4>

<form action="change_price.php" method="POST">
<div class="row">
<div class="col-md-6">
<div class="form-group">
<input type="number" name="xaplostra" class="form-control" placeholder="Νέα Τιμή Ξαπλώστρας" required>
</div>
</div>
<div class="col-md-6">
<div class="form-group">
<input type="number" name="ombrella" class="form-control" placeholder="Νέα Τιμή Ομπρέλας" required>
</div>
</div>
</div>
<div class="col-md-12">
<div class="form-group" align="center">
<input type="submit" value="ΥΠΟΒΟΛΗ" class="btn btn-primary">
</div>
</div>
</form>

```

Στο παραπάνω κομμάτι κώδικα, όπως θα δείτε γίνεται η προβολή των τιμών μέσω echo της php στα h3 και h4 μεγέθη προβολής κειμένου της html. Επίσης, παρακάτω θα δείτε μια φόρμα όπου δέχεται σαν είσοδο δύο αριθμητικές τιμές για την τιμή της ξαπλώστρας και της ομπρέλας αντίστοιχα.

Οι τιμές αυτές θα αποσταλούν με χρήστη POST στη σελίδα change_price.php όπου μας ανακατευθύνει το κουμπί “ΥΠΟΒΟΛΗ”.

```
<?php
require_once "config.php";
session_start();
if(!isset($_SESSION["admin"]) && $_SESSION["admin"] === 1 )
{
    header("location: index.php");
    exit;
}

$name = $_SESSION["id"];
$xaplostra= $_POST['xaplostra'];
$ombrella= $_POST['ombrella'];

$update1 = "UPDATE posto set xaplostra='$xaplostra', ombrella= '$ombrella', last_edit=sysdate() where userid='$name' ";
mysqli_query($link,$update1);

header("location: edit-price.php");
?>
```

Από ότι θα διακρίνετε στον παραπάνω κώδικα, κάνουμε εκχώρηση των δεδομένων των POST μεταβλητών στις αντίστοιχες τοπικές μεταβλητές μας. Έπειτα με την εντολή \$update1 πραγματοποιούμε αλλαγή των τιμών του πίνακα posto με τις τιμές των δεδομένων των μεταβλητών μας. Τέλος, αφότου γίνει η παραπάνω κρυφή διαδικασία, ο συνεργάτης μεταφέρεται στην ίδια πάλι σελίδα (γίνεται refresh δηλαδή) όπου πλέον του παρουσιάζονται ανανεωμένα τα δεδομένα που μόλις υπέβαλε μέσω της φόρμας.

3.10 Εύρεση συνεργάτη από διαχειριστή

Ένας διαχειριστής πλατφόρμας έχει επιπλέον δυνατότητες μέσω της σελίδας adminportal.php. Μία από τις δυνατότητες που έχει είναι να κάνει κλικ στη σελίδα find-user.php όπου από εκεί μέσω του παρακάτω κώδικα να κάνει εύρεση ενός συνεργάτη δίνοντας ως στοιχείο αναζήτησης το ΑΦΜ του, το κινητό του ή το email του.

```

require_once "config.php";
session_start();
if(!(isset($_SESSION["admin"]) && $_SESSION["admin"] === 2) )
{
    header("location: index.php");
    exit;
}
$userid = $admin = "";
$user_afm = $tilefono = $username = $userid = "";
?>

```

Όπως θα διακρίνατε από τον παραπάνω κώδικα, πλέον ο έλεγχος πρόσβασης στη σελίδα γίνεται με τη συνθήκη αν η μεταβλητή session της συνεδρίας έχει την τιμή «2». Αυτό πρακτικά συμβολίζει ότι την τιμή 2 την έχουν μόνο οι συνδεδεμένοι διαχειριστές της πλατφόρμας και όχι οι απλοί συνεργάτες (έχουν την τιμή 1). Με αυτό τον έλεγχο διασφαλίζεται ότι μόνο οι διαχειριστές θα μπορούν να έχουν πρόσβαση και να μεταβάλλουν ευαίσθητα δεδομένα για τη λειτουργία της πλατφόρμας.

```

<form method="GET" action="">
<div class="form-group">
<label>ΑΦΜ Συνεργάτη</label>
<input type="number" name="user_afm" class="form-control" required="true" value="<?php echo $user_afmm; ?>">
</div>
<div class="form-group">
<input type="submit" class="btn btn-primary" name="submit1" value="Αναζήτηση">
</div>
</form>
<form method="GET" action="">
<div class="form-group">
<label>Κινητό Πελάτη</label>
<input type="number" name="tilefono" class="form-control" value="<?php echo $tilefono; ?>">
</div>
<div class="form-group">
<input type="submit" class="btn btn-primary" name="submit2" value="Αναζήτηση">
</div>
</form>
<form method="GET" action="">
<div class="form-group">
<label>Email Πελάτη</label>
<input type="email" name="username" class="form-control" value="<?php echo $username; ?>">
</div>
<div class="form-group">
<input type="submit" class="btn btn-primary" name="submit3" value="Αναζήτηση">
</div>
</div>
</form>

```


Ο παραπάνω κώδικας περιέχει τρεις φόρμες ώστε να μπορεί να γίνει αναζήτηση με χρήση ΑΦΜ, email ή κινητού. Αναλόγως ποια φόρμα θα υποβληθεί, στη σελίδα θα παρουσιαστεί το αντίστοιχο αποτέλεσμα των παρακάτω εντολών.

Αν γίνει αναζήτηση με χρήση ΑΦΜ:

```
<?php
if(isset($_GET['submit1'])){
$user_afm=$_GET['user_afm'];
$sql1 = "SELECT uname, lname , afm, tilefono, username, id FROM users WHERE afm='$user_afm' ";
if($result = mysqli_query($link, $sql1)){
    if(mysqli_num_rows($result) > 0){
echo "<div align='center' class='form-group'>";
    echo "<table>";
    echo "<br/>";
    echo "<div class='section_title'>Ο πελάτης βρέθηκε!</div>";
    echo "<tr>";
        echo "<th>Όνομα </th>";
        echo "<th>Επώνυμο </th>";
        echo "<th>ΑΦΜ </th>";
        echo "<th>Τηλέφωνο </th>";
        echo "<th>Email </th>";
        echo "<th>ID </th>";
    echo "</tr>";
    while($row = mysqli_fetch_array($result)){
        echo "<tr>";
            echo "<td>" . $row['uname'] . "</td>";
            echo "<td>" . $row['lname'] . "</td>";
            echo "<td>" . $row['afm'] . "</td>";
            echo "<td>" . $row['tilefono'] . "</td>";
            echo "<td>" . $row['username'] . "</td>";
            echo "<td>" . $row['id'] . "</td>";
            echo "<form action='edit-user.php' method='GET'>";
            echo "<input type='hidden' name='user_id' value='". $row['id'] . "'>";
            echo "<td>";
            echo "<input type='submit' value='Επεξεργασία'>";
            echo "</form>";
            echo "</td>";
        echo "</tr>";
    }
    echo "</table>";
    echo "</div>";
    // Free result set
    mysqli_free_result($result);
} else{
    echo "<div align='center' class='section_title'>Δεν βρέθηκε ο πελάτης.</div><br/>";
}
```

Αν γίνει αναζήτηση με χρήση Κινητού:

```
if(isset($_GET['submit2'])){
    $tilefono=$_GET['tilefono'];
    $sql2 = "SELECT uname, lname , afm, tilefono, username, id FROM users WHERE tilefono='$tilefono' ";

    if($result = mysqli_query($link, $sql2)){
        if(mysqli_num_rows($result) > 0){
            echo "<div align='center' class='form-group'>";
            echo "<table>";
            echo "<br/>";
            echo "<div class='section_title'>Ο πελάτης βρέθηκε!</div>";
            echo "<tr>";
            echo "<tr>";
            echo "<th>Όνομα </th>";
            echo "<th>Επώνυμο </th>";
            echo "<th>ΑΦΜ </th>";
            echo "<th>Τηλέφωνο </th>";
            echo "<th>Email </th>";
            echo "<th>ID </th>";
            echo "</tr>";
            while($row = mysqli_fetch_array($result)){
                echo "<tr>";
                echo "<td>" . $row['uname'] . "</td>";
                echo "<td>" . $row['lname'] . "</td>";
                echo "<td>" . $row['afm'] . "</td>";
                echo "<td>" . $row['tilefono'] . "</td>";
                echo "<td>" . $row['username'] . "</td>";
                echo "<td>" . $row['id'] . "</td>";
                echo "<form action='edit-user.php' method='GET'>";
                echo "<input type='hidden' name='user_id' value='". $row['id'] ."'>";
                echo "<td>";
                echo "<input type='submit' value='Επεξεργασία'>";
                echo "</form>";
                echo "</td>";
                echo "</tr>";
            }
            echo "</table>";
            echo "</div>";
            // Free result set
            mysqli_free_result($result);
        } else{
            echo "<div align='center' class='section_title'>Δεν βρέθηκε ο πελάτης.</div><br/>";
        }
    }
}
```

Αν γίνει αναζήτηση με χρήση Email:

```
if(isset($_GET['submit3'])){
$username=$_GET['username'];
$sql2 = "SELECT uname, lname , afm, tilefono, username, id FROM users WHERE username='$username' ";

if($result = mysqli_query($link, $sql2)){
    if(mysqli_num_rows($result) > 0){
        echo "<div align='center' class='form-group'>";
        echo "<table>";
        echo "<br>";
        echo "<div class='section_title'>Ο πελάτης βρέθηκε!</div>";
        echo "<tr>";
        echo "<th>Όνομα </th>";
        echo "<th>Επώνυμο </th>";
        echo "<th>ΑΦΜ </th>";
        echo "<th>Τηλέφωνο </th>";
        echo "<th>Email </th>";
        echo "<th>ID </th>";
        echo "</tr>";
        while($row = mysqli_fetch_array($result)){
            echo "<tr>";
            echo "<td>" . $row['uname'] . "</td>";
            echo "<td>" . $row['lname'] . "</td>";
            echo "<td>" . $row['afm'] . "</td>";
            echo "<td>" . $row['tilefono'] . "</td>";
            echo "<td>" . $row['username'] . "</td>";
            echo "<td>" . $row['id'] . "</td>";
            echo "<form action='edit-user.php' method='GET'>";
            echo "<input type='hidden' name='user_id' value='". $row['id'] . "'>";
            echo "<td>";
            echo "<input type='submit' value='Επεξεργασία'>";
            echo "</form>";
            echo "</td>";
            echo "</tr>";
        }
        echo "</table>";
        echo "</div>";
        // Free result set
        mysqli_free_result($result);
    } else{
        echo "<div align='center' class='section_title'>Δεν βρέθηκε ο πελάτης.</div><br/>";
    }
}
```

Και για τις τρεις περιπτώσεις, στην σελίδα θα παρουσιαστεί ένας πίνακας που θα προβάλλει το Όνομα, το Επώνυμο, το τηλέφωνο, το Email, το ID του χρήστη καθώς και ένα κουμπί “Επεξεργασία”. Το κουμπί αυτό θα οδηγεί στη σελίδα edit-user.php μέσω GET και με όρισμα το ID του χρήστη.

Στη σελίδα edit-user.php λοιπόν, γίνεται λεπτομερής προβολή των στοιχείων του χρήστη, δυνατότητα να μεταβληθούν τα υπάρχοντα στοιχεία του, δυνατότητα να οριστεί ως πιστοποιημένος συνεργάτης (επίπεδο admin=1) και τέλος να γίνει μεταφόρτωση μιας εικόνας

για το τμήμα της παραλίας του χρήστη που θα προβάλλεται στα αποτελέσματα της αναζήτησης τμημάτων παραλίας για κράτηση.

```
$sql = " SELECT * from users where id ='$user_id'";  
$result = $link->query($sql);  
  
if ($result->num_rows > 0) {  
    // output data of each row  
    while($row = $result->fetch_assoc()) {  
  
        $uname = $row["uname"] ;  
        $lname = $row["lname"] ;  
        $nomos = $row["nomos"] ;  
        $paralia = $row["paralia"] ;  
        $afm =$row["afm"] ;  
        $username =$row["username"] ;  
        $tilefono = $row["tilefono"] ;  
        $adminright =$row["admin"] ;  
        $created = $row["created_at"] ;  
    }  
}
```

Με το παραπάνω κομμάτι κώδικα, παίρνουμε τις πληροφορίες του χρήστη από τον πίνακα users και τις εκχωρούμε σε τοπικές μεταβλητές για να τις προβάλλουμε στη συνέχεια.

Με το παρακάτω κομμάτι κώδικα, θα μεταφορτώσουν την επιλεγθείσα εικόνα που μας έδωσε ο διαχειριστής και θα την αποθηκεύσουμε μόνιμα στον πίνακα images ορίζοντας σαν

μεταβλητή \$postoid το ID του τμήματος παραλίας του συνεργάτη.

```
$image="";
if(isset($_POST['upload']))
{
    $target = "images/".basename($_FILES["image"]["name"]);
    $image = $_FILES['image']['name'];

    $sql = "UPDATE images SET image='$image' WHERE postoid = '$paralia' ";
    mysqli_query($link, $sql);

    if (move_uploaded_file($_FILES['image']['tmp_name'], $target)) {
        $msg = "Η φωτογραφία σας ενημερώθηκε επιτυχώς!";
    }else{
        $msg = "Υπήρξε πρόβλημα κατά το ανέβασμα της φωτογραφίας σας. Παρακαλώ προσπαθήστε ξανά.";
    }
}
```

Εδώ, θα δείτε ότι γίνεται η προβολή των στοιχείων του χρήστη που τραβήξαμε από τον πίνακα προηγούμενως:

```
<div class="row animate-box">
  <div class="col-md-6">
    <h3 class="section-title">Υφιστάμενα Στοιχεία</h3>
    <ul class="contact-info">
      <li><i class="icon-location-pin"></i><b>ΟΝΟΜΑ:<?php echo $uname ?></b></li>
      <li><i class="icon-location-pin"></i><b>ΕΠΩΝΥΜΟ:<?php echo $lname ?></b></li>
      <li><i class="icon-location-pin"></i><b>ΑΦΜ:<?php echo $afm ?></b></li>
      <li><i class="icon-phone2"></i><b>ΤΗΛΕΦΩΝΟ:<?php echo $telefono ?></b></li>
      <li><i class="icon-mail"></i><b>EMAIL:<?php echo $username ?></b></li>
      <li><i class="icon-location-pin"></i><b>ΝΟΜΟΣ:<?php echo $nomos ?></b></li>
      <li><i class="icon-globe2"></i><b>ΠΑΡΑΛΙΑ(Κωδ. Τμήματος):<?php echo $paralia ?></b></li>
      <?php if ($adminright == 0) {
        $hi = 'Not Admin';
      }
      elseif ($adminright == 1) {
        $hi = 'Admin';
      }
      elseif ($adminright == 2) {
        $hi = 'Super Admin';
      }
      else {
        $hi = $adminright;
      }
      ?>
      <li><i class="icon-globe2"></i><b>Δικαιώματα:<?php echo $hi ?></b></li>
      <li><i class="icon-location-pin"></i><b>ΗΜ/ΝΙΑ ΕΓΓΡΑΦΗΣ:<?php echo $created ?></b></li>
```

Και τέλος μέσω της παρακάτω φόρμας μπορούμε να μεταβάλλουμε τα στοιχεία του συνεργάτη όπως για παράδειγμα: όνομα, επώνυμο, ΑΦΜ, τηλέφωνο, email, κωδικό τμήματος παραλίας, Νομό και τέλος επίπεδο διαχείρισης.

```

<h3 class="section-title">Ανάκληση στοιχείων</h3>
<form action="insert_user.php" method="POST">
  <input type="hidden" name="user_id" value="<?php echo $user_id ?>" >
  <div class="row">
    <div class="col-md-6">
      <div class="form-group">
        <input type="text" name="uname1" class="form-control" placeholder="Όνομα" required>
      </div>
    </div>
    <div class="col-md-6">
      <div class="form-group">
        <input type="text" name="lname1" class="form-control" placeholder="Επώνυμο" required>
      </div>
    </div>
    <div class="col-md-6">
      <div class="form-group">
        <input type="text" name="afm1" class="form-control" placeholder="ΑΦΜ" required>
      </div>
    </div>
    <div class="col-md-6">
      <div class="form-group">
        <input type="text" name="telefon1" class="form-control" placeholder="Τηλέφωνο" required>
      </div>
    </div>
    <div class="col-md-6">
      <div class="form-group">
        <input type="email" name="username1" class="form-control" placeholder="Email" required>
      </div>
    </div>
    <div class="col-md-6">
      <div class="form-group">
        <input type="text" name="paralia1" class="form-control" placeholder="Παραλία" required>
      </div>
    </div>
    <div class="col-md-6">
      <label>Νομός</label>
      <select name="nomos1" class="form-control" required="true">
        <option value="">Επιλέξτε νομό</option>
        <option value="ΑΓΙΟ ΟΡΟΣ">ΑΓΙΟ ΟΡΟΣ</option>
        <option value="ΑΤΤΙΚΗΣ">ΑΤΤΙΚΗΣ</option>
        <option value="ΑΙΤΩΛΟΑΚΑΡΝΑΝΙΑΣ">ΑΙΤΩΛΟΑΚΑΡΝΑΝΙΑΣ</option>
      </select>
    </div>
  </div>
  <div class="col-md-6">
    <div class="form-group">
      <label>Δικαιώματα</label>
      <select name="admin1" value="Admin" class="form-control" required="true">
        <option value="" > </option>
        <option value="0">Not Admin</option>
        <option value="1">Admin</option>
        <option value="2">Super Admin</option>
      </select>
    </div>
  </div>
  <div class="col-md-12">
    <div class="form-group" align="center">
      <input type="submit" value="ΥΠΟΒΟΛΗ" class="btn btn-primary">
    </div>
  </div>
</form>
<form method="post" align="center" enctype="multipart/form-data">
  Ανέβασμα Φωτογραφίας για το τμήμα παραλίας: <?php echo $paralia; ?>
  <br>
  <br>
  <input type="hidden" name="size" value="1000000">
  <div>
    <input type="file" name="image" required="true" accept="image/*">
  </div>
  <br>
  <br>
  <div>
    <input type="submit" name="upload" value="Upload" class="btn btn-one" >
  </div>
</form>

```

Ο διαχειριστής εφόσον συμπληρώσει τα απαραίτητα στοιχεία από την παραπάνω φόρμα που θέλει να μεταβάλλει, τότε μεταφέρεται στην κρυφή σελίδα insert_user.php.

Στην σελίδα αυτή, όπως θα δείτε παρακάτω, γίνεται η αποθήκευση των τιμών του POST σε τοπικές μεταβλητές όπου στη συνέχεια χρησιμοποιώντας την \$update θα γίνει η αλλαγή του στον πίνακα users της βάσης δεδομένων.

```
require_once 'config.php';
session_start();
if(!isset($_SESSION["admin"]) && $_SESSION["admin"] === 2))
{
    header("location: index.php");
    exit;
}
$user_id = $_POST['user_id'];
$username = $_POST['username1'];
$lname = $_POST['lname1'];
$afm = $_POST['afm1'];
$tilefono1 = $_POST['tilefono1'];
$username1 = $_POST['username1'];
$paralia1 = $_POST['paralia1'];
$nomos1 = $_POST['nomos1'];
$admin1 = $_POST['admin1'];

$update = "UPDATE users set uname='$username', lname='$lname', afm='$afm', tilefono='$tilefono1', username='$username1', paralia='$paralia1',
nomos='$nomos1', admin='$admin1' where id='$user_id' ";
mysqli_query($link,$update);

header("location: edit-user.php?user_id=$user_id");
?>
```

3.11 Εμφάνιση κρατήσεων πλατφόρμας διαχειριστή

Μια από τις δυνατότητες του διαχειριστή είναι να έχει πρόσβαση και να διαχειρίζεται όλες τις κρατήσεις από όλα τα τμήματα παραλιών που υπάρχουν. Γι' αυτό το λόγο υπάρχει η σελίδα all-bookings.php.

Η σελίδα αυτή μοιάζει με την σελίδα my-bookings.php που έχουν εξηγήσει παραπάνω, με τη μόνη διαφορά όμως ότι δεν επιστρέφει αποτελέσματα με βάση τον user της συνεδρίας.

```

$sql = " SELECT bookingid, sunbeds, umbrellas, date, time, name, surname, email, phone, cost, paid, accepted, comments from bookings ORDER BY
        bookingid DESC";
$result = $link->query($sql);

$result = mysqli_query($link, $sql); // First parameter is just return of "mysqli_connect()" function
if ($result->num_rows > 0) {
    echo "<tr>";
    echo "<th>BookingID</th>";
    echo "<th>Ξαπλώστρες</th>";
    echo "<th>Ομπρέλες</th>";
    echo "<th>Ημερομηνία</th>";
    echo "<th>Ωρα</th>";
    echo "<th>Όνομα</th>";
    echo "<th>Επώνυμο</th>";
    echo "<th>Email</th>";
    echo "<th>Τηλέφωνο</th>";
    echo "<th>Συνολικό Ποσό</th>";
    echo "<th>Πληρώθηκε</th>";
    echo "<th>Αποδοχή</th>";
    echo "<th>Σχόλια</th>";
    echo "<th>Επεξεργασία</th>";
    echo "</tr>";
    while ($row = mysqli_fetch_assoc($result)) { // Important line !!! Check summary get row on array ..
        foreach ($row as $field => $value) { // I you want you can right this line like this: foreach($row as $value) {
            echo "<td>";
            echo " $value "; // I just did not use "htmlspecialchars()" function.
            echo "</td>";
        }
        echo "<form action='edit-booking.php' method='GET'>";
        echo "<input type='hidden' name='bookingid' value='". $row['bookingid'] .'>";
        echo "<td>";
        echo "<input type='submit' value='Επεξεργασία' ";
        echo "</td>";
        echo "</form>";
        echo "</tr>";
    }
}

```

Από το παραπάνω κομμάτι κώδικα καταλαβαίνουμε, ότι οι δύο αυτές παρόμοιες σελίδες διαφέρουν στην \$sql εντολή τους. Η εντολή λοιπόν στη σελίδα των διαχειριστών δεν έχει τον περιορισμό του “where postoid = '\$name'” οπότε μας επιστρέφει όλες τις κρατήσεις που υπάρχουν στον πίνακα booking της βάσης.

3.12 Εξαγωγή στατιστικών πλατφόρμας διαχειριστή

Μια επίσης έξτρα δυνατότητα που έχουν οι διαχειριστές, είναι αυτή της εξαγωγής στατιστικών στοιχείων σε μορφή .xls για τις κρατήσεις των τμημάτων παραλίας (συνεργατών) και για μια παραλία σε σύνολο.


```

        <form action="export_posto.php" method="POST">
        <h3 class="section-title">Εξαγωγή Στατιστικών Πόστου</h3>
        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <label for="date-start">Επιλογή Πόστου:</label>
                    <?php
                    echo "<select id='postoid' name='postoid' class='form-group' required>";
                    echo '<option value="">Επιλογή Πόστου</option>';

                    while ($row = mysqli_fetch_array($result1)) {
                        unset($postoid, $postoname);
                        $postoid = $row['postoid'];
                        $postoname = $row['postoname'];

                        echo '<option value="'. $postoid. '">'. $postoname. '</option>';
                    }
                    echo "</select>";
                <?php
            </div>
        </div>
        <div class="col-md-6">
            <div class="form-group align="center">
                <input type="submit" value="ΥΠΟΒΟΛΗ" class="btn btn-primary">
            </div>
        </div>
    </form>
</div>
</div>

    <div class="col-md-6">
        <h3 class="section-title">Εξαγωγή Στατιστικών Παραλίας</h3>
        <form action="export_beaches.php" method="POST">

        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <label for="date-start">Επιλογή Παραλίας:</label>
                    <?php
                    echo "<select id='beachid' name='beachid' class='form-group' required>";
                    echo '<option value="">Επιλογή Παραλίας</option>';

                    while ($row = mysqli_fetch_array($result2)) {
                        unset($beachid, $beachname);
                        $beachid = $row['beachid'];
                        $beachname = $row['beachname'];

                        echo '<option value="'. $beachid. '">'. $beachname. '</option>';
                    }
                    echo "</select>";
                <?php
            </div>
        </div>
    </form>
</div>

```

Από τον παραπάνω κώδικα καταλαβαίνουμε, ότι υπάρχουν δύο διαφορετικές φόρμες επιλογής στατιστικών εξαγωγής. Η πρώτη επιστρέφει σε ένα `select` τα αποτελέσματα του κώδικα `$sql1` και η δεύτερη του κώδικα `$sql2` αντίστοιχα.

```

$sql1 = " SELECT postoid,postoname from posto";
$result1 = $link->query($sql1);

$sql2 = " SELECT beachid,beachname from beaches";
$result2 = $link->query($sql2);

```

Και οι δύο φόρμες βέβαια, αφού υποβληθούν μπορούμε να δούμε ότι μας ανακατευθύνουν στη κρυφή σελίδα export_posto.php. Εδώ γίνεται η ουσιαστική εξαγωγή των δεδομένων σε μορφή .xls όπως θα δείτε παρακάτω:

```
$date = date('Y-m-d-H:i');
// $date1 = $_POST['date1'];
// $date2 = $_POST['date2'];
$postoid = $_POST['postoid'];
$filename = "report";
$filename .= $date;
$filename .= ".xls";
function filterData(&$str){
    $str = preg_replace("/\t/", "\\t", $str);
    $str = preg_replace("/\r?\n/", "\\n", $str);
    if(strstr($str, '"')) $str = '"' . str_replace('"', '""', $str) . '"';
}

$fields = array('BOOKINGID', 'CREATED', 'POSTOID', 'BEACHID', 'SUNBEDS', 'UMNRELLAS', 'DATE', 'TIME', 'NAME', 'SURNAME', 'EMAIL', 'PHONE', 'COST', 'PAID', 'ACCEPTED', 'COMMENTS');

// Display column names as first row
$excelData = implode("\t", array_values($fields)) . "\n";

// Get records from the database
$query = $link->query("SELECT * FROM bookings WHERE postoid = '$postoid'");
if($query->num_rows > 0){
    // Output each row of the data
    $i=0;
    while($row = $query->fetch_assoc()){
        $rowData = array($row['bookingid'], $row['created_at'], $row['postoid'], $row['beachid'], $row['sunbeds'], $row['umbrellas'], $row['date'], $row['time'], $row['name'], $row['surname'], $row['email'], $row['phone'], $row['cost'], $row['paid'], $row['accepted'], $row['comments']);
        array_walk($rowData, 'filterData');
        $excelData .= implode("\t", array_values($rowData)) . "\n";
    }
} else {
    $excelData .= "No records found...\n";
}
```

```
// Headers for download
header("Content-Disposition: attachment; filename=\"$filename\"");
header("Content-Type: application/vnd.ms-excel");

// Render excel data
echo $excelData;

exit;

header("location: export-data.php");
```

Τέλος, αφού τα στοιχεία του πίνακα bookings εξαχθούν, δημιουργείται ένα αρχείο με όνομα report και την σημερινή ημερομηνία και αποθηκεύεται τοπικά στον υπολογιστή του διαχειριστή. Αφού γίνει λοιπόν η αποθήκευση, ο διαχειριστής ανακατευθύνεται στην ίδια σελίδα (refresh) και μπορεί να ανοίξει το αρχείο για προβολή.

3.13 Προσθήκη νέα παραλίας από διαχειριστή

Ο διαχειριστής έχει τη δυνατότητα να επιλέξει από τη σελίδα adminportal.php να μπει στη σελίδα add-beach.php. Από εκεί συμπληρώνοντας την παρακάτω φόρμα, μπορεί να καταχωρήσει μια νέα παραλία σε έναν Νόμο.

```
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">

<div class="col-md-6">

    <select name="nomos" class="form-control" required="true">
    <option value="" >Νομός</option>
    <option value="Άγιο Όρος">Άγιο Όρος</option>
    <option value="ΑΘΗΝΩΝ">ΑΘΗΝΩΝ</option>
    <option value="Αιτωλοακαρνανίας">Αιτωλοακαρνανίας</option>
    <option value="Ανατολικής Αττικής">Ανατολικής Αττικής</option>
    <option value="Αργολίδας">Αργολίδας</option>
    <option value="Αρκαδίας">Αρκαδίας</option>
    <option value="Άρτας">Άρτας</option>
    <option value="Αχαΐας">Αχαΐας</option>
    <option value="Βοιωτίας">Βοιωτίας</option>
    <option value="Γρεβενών">Γρεβενών</option>
    <option value="Δράμας">Δράμας</option>
    <option value="Δυτικής Αττικής">Δυτικής Αττικής</option>
    <option value="Δωδεκανήσου">Δωδεκανήσου</option>
    <option value="Εβρου">Εβρου</option>
    <option value="Ευβοίας">Ευβοίας</option>
    <option value="Ευρυτανίας">Ευρυτανίας</option>
    <option value="Ζακύνθου">Ζακύνθου</option>
    <option value="Ηλείας">Ηλείας</option>
    <option value="Ημαθίας">Ημαθίας</option>
    <option value="Ηρακλείου">Ηρακλείου</option>
    <option value="Θεσπρωτίας">Θεσπρωτίας</option>
    <option value="Θεσσαλονίκης">Θεσσαλονίκης</option>
    <option value="Ιωαννίνων">Ιωαννίνων</option>
    <option value="Καβάλας">Καβάλας</option>
    <option value="Καρδίτσας">Καρδίτσας</option>
    <option value="Καστοριάς">Καστοριάς</option>
    <option value="Κέρκυρας">Κέρκυρας</option>
    <option value="Κεφαλληνίας">Κεφαλληνίας</option>
    <option value="Κιλκίς">Κιλκίς</option>
    <option value="Κοζάνης">Κοζάνης</option>
    <option value="Κορινθίας">Κορινθίας</option>
    <option value="Κυκλάδων">Κυκλάδων</option>
    <option value="Λακωνίας">Λακωνίας</option>
    <option value="Λάρισας">Λάρισας</option>
    <option value="Λασιθίου">Λασιθίου</option>
    <option value="Λέσβου">Λέσβου</option>
    <option value="Λευκάδας">Λευκάδας</option>
    <option value="Μαγνησίας">Μαγνησίας</option>
    <option value="Μεσσηνίας">Μεσσηνίας</option>
    <option value="Ξάνθης">Ξάνθης</option>
    <option value="Πειραιώς">Πειραιώς</option>
    <option value="Πέλλας">Πέλλας</option>
    <option value="Πιερίας">Πιερίας</option>
    <option value="Πρέβεζας">Πρέβεζας</option>
    <option value="Ρεθύμνης">Ρεθύμνης</option>
    <option value="Ροδόπης">Ροδόπης</option>
    <option value="Σάμου">Σάμου</option>
    <option value="Σερρών">Σερρών</option>
```

Στο παραπάνω κώδικα, όπως θα δείτε χρησιμοποιούμε μια φόρμα POST όπου υπάρχει ένα select με προ συμπληρωμένους όλους τους Νομούς της Ελλάδος και υπάρχει και ένα input text όπου ο χρήστης συμπληρώνει το όνομα της παραλίας που επιθυμεί να προσθέσει.

```
if($_SERVER["REQUEST_METHOD"] == "POST"){  
    // Prepare a select statement  
    $sql = "SELECT beachname FROM beaches WHERE beachname = ? and nomos like'$nomos';";  
  
    if($stmt = mysqli_prepare($link, $sql)){  
        // Bind variables to the prepared statement as parameters  
        mysqli_stmt_bind_param($stmt, "s", $beachname);  
  
        // Set parameters  
        $param_name = trim($_POST["beachname"]);  
  
        // Attempt to execute the prepared statement  
        if(mysqli_stmt_execute($stmt)){  
            /* store result */  
            mysqli_stmt_store_result($stmt);  
  
            if(mysqli_stmt_num_rows($stmt) == 1){  
                $name_err = "Αυτή η Παραλία υπάρχει ήδη!";  
            } else{  
                $beachname = trim($_POST["beachname"]);  
            }  
        } else{  
            echo "Ωπα! Κάτι πήγε στραβά. Παρακαλώ προσπαθήστε ξανά αργότερα.";  
        }  
    }  
}
```

Με το \$sql επιλέγουμε τις υπάρχουσες καταχωρήσεις από τον πίνακα beaches και ελέγχουμε ότι η παραλία που εισήγαγε ο χρήστης δεν υπάρχει ξανά καταχωρημένη στον ίδιο Νομό. Εφόσον η συνθήκη δεν ισχύει (δεν υπάρχει ξανά η παραλία) τότε εκτελείται ο παρακάτω κώδικας που μέσω της \$sql κάνει insert τα δεδομένα στη βάση.

```

if(empty($name_err) ){

    // Prepare an insert statement
    $sql = "INSERT INTO beaches (beachname, nomos) VALUES (?,?)";
    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "ss", $param_name, $param_nomos);

        // Set parameters
        $param_name = $beachname;
        $param_nomos = $nomos;

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){

            header("location: adminportal.php");
        } else{
            echo "Ωπα! Κάτι πήγε στραβά. Παρακαλώ προσπαθήστε ξανά αργότερα.";
        }
    }
}

```

3.14 Προσθήκη τμήματος παραλίας από διαχειριστή

Για να μπορέσουμε να συνδέσουμε έναν συνεργάτη με μία παραλία, ο διαχειριστής έχει την δυνατότητα να δημιουργήσει ένα νέο τμήμα παραλίας το οποίο ουσιαστικά θα ανήκει στον συνεργάτη. Από την σελίδα adminportal.php λοιπόν μπορεί να επιλέξει να μεταβεί στη σελίδα add-posto.php.

```

session_start();
$userid = $admin = $beachname = $name_err = $param_name = $param_nomos = "";
if(! (isset($_SESSION["admin"]) && $_SESSION["admin"] === 2) )
{
    header("location: index.php");
    exit;
}

$nomos = $image = "";
$sql1 = "SELECT beachid, beachname, nomos FROM beaches ORDER BY nomos ASC";
$result1 = $link->query($sql1);
$sql2 = "SELECT id, uname, lname, afm, paralia FROM users where paralia='' ORDER BY created_at DESC";
$result2 = $link->query($sql2);
if(isset($_POST['upload']))
{
    $target = "images/" . basename($_FILES["image"]["name"]);
    $image = $_FILES['image']['name'];
}

```

Εκεί, όπως θα δείτε αφού γίνει έλεγχος ότι ο συνδεδεμένος χρήστης είναι όντως διαχειριστής, με χρήση των \$sql1 και \$sql2 θα πάρουμε από τον πίνακα beaches τις διαθέσιμες παραλίες

αλφαβητικά ταξινομημένες και από τον πίνακα users τα στοιχεία των χρηστών οι οποίοι όμως δεν έχουν κάποια ήδη δηλωμένη παραλία (where paralia='').

Τα αποτελέσματα των παραπάνω εντολών τα βάζουμε στις μεταβλητές \$result1 και \$result2 αντίστοιχα όπου θα τα χρησιμοποιήσουμε παρακάτω για να εμφανιστούν σαν επιλογές στους δύο select μηχανισμούς που έχουμε στη φόρμα μας.

```
<form action="insert_posto.php" method="post" align="center">

    <div class="col-md-6">
        <div class="form-group">
            <?php
            echo "<select id='userid' name='username' class='form-group' required>";
            echo '<option value="">Επιλογή Συνεργάτη</option>';

            while ($row = mysqli_fetch_array($result2)) {
                unset($userid, $username);
                $userid= $row['id'];
                $username.= $row['uname'];
                $username.= " ";
                $username.= $row['lname'];
                $username.= " ";
                $username.= $row['afm'];

                echo '<option value="'. $userid. '">'. $username. '</option>';
            }
            echo "</select>";
            ?>
        </div>
        <div class="form-group" align="center">

            <?php
            echo "<select id='beachid' name='beachname' class='form-group' required>";
            echo '<option value="">Επιλογή Παραλίας</option>';

            while ($row = mysqli_fetch_array($result1)) {
                unset($beachid, $beachname);
                $beachid = $row['beachid'];
                $beachname.= $row['nomos'];
                $beachname.= " ";
                $beachname.= $row['beachname'];

                echo '<option value="'. $beachid. '">'. $beachname. '</option>';
            }
            echo "</select>";
            ?>
        </div>
```

Όπως θα δείτε στη παραπάνω εικόνα, έχουμε δημιουργήσει μια φόρμα τύπου POST η οποία όταν υποβληθεί θα μας ανακατευθύνει στη κρυφή σελίδα insert_posto.php.

```

<div class="form-group">
  <input type="number" name="latitude" class="form-control" placeholder="Latitude" step="any" required>
</div>

<div class="form-group">
  <input type="number" name="longitude" class="form-control" placeholder="Longitude" step="any" required>
</div>
<div class="form-group">
  <input type="number" name="xaplostra" class="form-control" placeholder="Τιμή Ξαπλώστρας" required>
</div>
<div class="form-group">
  <input type="number" name="ombrella" class="form-control" placeholder="Τιμή Ομπρέλας" required>
</div>
<div class="form-group">
  <input type="text" name="postoname" class="form-control" placeholder="Διακριτό Όνομα" required>
</div>

<div class="col-md-12">
  <div class="form-group align="center">
    <input type="submit" value="ΚΑΤΑΧΩΡΗΣΗ" class="btn btn-primary">
  </div>
</div>
</div>

```

Εκτός από τα δύο select, ο διαχειριστής θα πρέπει να συμπληρώσει όπως βλέπετε από τον παραπάνω κώδικα τα πεδία: latitude και longitude όπου αντιστοιχούν στις συντεταγμένες του τμήματος παραλίας του συνεργάτη (πόστο), την αρχική τιμή της ξαπλώστρας και της ομπρέλας και τέλος το διακριτό όνομα του συνεργάτη (επωνυμία).

Αφού συμπληρωθούν όλα τα παραπάνω, μεταβαίνουμε στη κρυφή για τον διαχειριστή σελίδα insert_posto.php.

```

$name = "";
$postoname = $_POST['postoname'];
$username = $_POST['username'];
$beachname = $_POST['beachname'];
$latitude = $_POST['latitude'];
$longitude = $_POST['longitude'];
$xaplostra = $_POST['xaplostra'];
$ombrella = $_POST['ombrella'];

$update1 = "INSERT INTO posto (userid,beachid,postoname,latitude,longitude,xaplostra,ombrella) VALUES ('$username','$beachname','$postoname','$latitude','$longitude','$xaplostra','$ombrella')";
mysqli_query($link,$update1);

$sql1 = "SELECT postoid FROM posto WHERE userid = '$username' ";
$result = mysqli_query($link, $sql1);
while($row = $result->fetch_assoc()) {
  $name = $row['postoid'];
}
$update2 = "UPDATE users set paralia = '$name' where id='$username' ";
mysqli_query($link,$update2);

```

Όπως βλέπετε στον παραπάνω κώδικα, αρχικά παίρνουμε τα δεδομένα των POST μεταβλητών και τα αποθηκεύουμε σε τοπικές μεταβλητές με τα αντίστοιχα ονόματα. Με χρήση της εντολής \$update1 κάνουμε εισαγωγή των στοιχείων αυτών στο πίνακα posto και στη συνέχεια με τη

βοήθεια της εντολής \$sql1 παίρνουμε το postoid του χρήστη. Ο λόγος που το κάνουμε αυτό είναι γιατί μέσω της \$update2 θα εισάγουμε τον αριθμό αυτό στον πίνακα users, ώστε να γνωρίζουμε από εδώ και πέρα ότι ο χρήστης μπορεί να διαχειρίζεται το συγκεκριμένο τμήμα παραλίας έπειτα από ταυτοποίηση. Προσοχή, επειδή η συνεδρία μας βασίζεται απόλυτα σε αυτή την πληροφορία για να προβάλει την σελίδα adminportal.php , αν ο συνεργάτης είναι ήδη συνδεδεμένος με ένα session στο λογαριασμό του και του κάνουμε ταυτόχρονα προσθήκη το τμήμα παραλίας, αυτό δεν θα εμφανιστεί στο συνεργάτη μέχρι να πραγματοποιήσει ξανά είσοδο. Αυτό είναι απόλυτα φυσιολογικό καθώς τα δεδομένα που ελέγχει η συνεδρία μέσω του browser, γίνονται ανάκτηση από τη βάση μόνο κατά την είσοδο του χρήστη και δεν ανανεώνονται όσο παραμένει συνδεδεμένος.

3.15 Διαγραφή τμήματος παραλίας από διαχειριστή

Η τελευταία δυνατότητα που προσφέρετε στον διαχειριστή, είναι να διαγράψει ένα τμήμα παραλίας δηλαδή ένα πόστο συνεργάτη. Αφού λοιπόν επιλέξει τη σελίδα delete-posto.php από την πλατφόρμα συνεργάτη, θα του εμφανιστεί η παρακάτω φόρμα όπου θα χρειαστεί να επιλέξει έναν συνεργάτη προς διαγραφή.

```
$sql2 = "SELECT id, uname, lname, afm, paralia FROM users where paralia!='' ORDER BY created_at DESC";  
$result2 = $link->query($sql2);
```

Από τον παραπάνω κώδικα θα δείτε ότι η \$sql2 μας επιστρέφει στην \$result2 τα στοιχεία των χρηστών από τον πίνακα users οι οποίοι όμως έχουν ήδη εγγεγραμμένο τμήμα παραλίας (paralia not = '').


```

<form action="delete_posto.php" method="post">

<div class="col-md-6">
  <div class="form-group">

    echo "<select id='userid' name='userid' class='form-group' required>";
    echo '<option value="">Επιλογή Συνεργάτη</option>';

    while ($row = mysqli_fetch_array($result2)) {
      unset($userid, $username);
      $userid= $row['id'];
      $username.= $row['uname'];
      $username.= " ";
      $username.= $row['lname'];
      $username.= " ";
      $username.= $row['afm'];

      echo '<option value="'. $userid. '">'. $username. '</option>';
    }
    echo "</select>";
    ?>

    <div class="col-md-12">
      <div class="form-group" align="center">
        <input type="submit" value="ΚΑΤΑΧΩΡΗΣΗ" class="btn btn-primary">
      </div>
    </div>
  </div>
</div>
</form>

```

Με χρήση της παραπάνω φόρμας, παρουσιάζουμε τα στοιχεία των συνεργατών μέσω μιας select και αφού ο διαχειριστής κάνει την επιλογή του, με το κουμπί “ΚΑΤΑΧΩΡΗΣΗ” μεταφερόμαστε στη κρυφή σελίδα delete_posto.php .

Εκεί, όπως θα δείτε στη παρακάτω εικόνα, μέσω της \$sql0 βρίσκουμε από τον πίνακα users το id του τμήματος της παραλίας που ανήκει. Αυτό αποθηκεύεται στην τοπική μεταβλητή \$posto, η οποία χρησιμοποιείται από την \$sql1 και πηγαίνει στον πίνακα posto και διαγράφει όλη την εγγραφή αυτή.

```

require_once "config.php";
session_start();
$posto = '';
if(!(isset($_SESSION["admin"]) && $_SESSION["admin"] === 2) )
{
    header("location: index.php");
    exit;
}

$userid = $_POST['userid'];

$sql0 = "SELECT paralia FROM users WHERE id = '$userid' ";
$result = mysqli_query($link, $sql0);
while($row = $result->fetch_assoc()) {
    $posto = $row['paralia'];
}
$sql1 = "DELETE FROM posto WHERE postoid = '$posto' ";
mysqli_query($link, $sql1);

// $sql2 = "DELETE FROM images WHERE postoid = '$posto' ";
// mysqli_query($link, $sql2)

header("location: adminportal.php");
?>

```

Τέλος, αφότου πραγματοποιηθεί η διαγραφή και χωρίς να έχει δει κάποια γραφική αλλαγή ο διαχειριστής, τον ανακατευθύνει στη σελίδα adminportal.php εκ νέου.

3.16 Ενσωμάτωση πλατφόρμας PayPal

Όταν ο συνεργάτης αποδεχθεί μια κράτηση, τότε όπως έχουμε αναφέρει και παραπάνω αποστέλλεται ένα γραπτό μήνυμα sms στον χρήστη για να προβεί στην πληρωμή. Το sms αυτό περιέχει ένα link (vresxaplostra.site/checkout.php?bookingid=1021) όπου οδηγεί στην σελίδα checkout.php περνώντας κάθε φορά ως GET παράμετρο το booking id.

```
require_once "config.php";
session_start();
$userid = $admin = $bookiid = $sunbeds = $umbrellas = $amount = $comments = $name = $surname = "";
$bookingid = $_GET['bookingid'];

$sql = " SELECT * from bookings where bookingid ='$bookingid'";
$result = $link->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {

        $sunbeds = $row["sunbeds"] ;
        $umbrellas = $row["umbrellas"] ;
        $date = $row["date"] ;
        $time = $row["time"] ;
        $name = $row["name"] ;
        $surname = $row["surname"] ;
        $amount = $row["cost"] ;
        $comments = $row["comments"] ;
    }
}

?>
```

Όπως θα δείτε στον παραπάνω κώδικα, με χρήση της \$sql καταχωρούμε τα στοιχεία της κράτησης σε τοπικές μεταβλητές. Πολύ σημαντική είναι η μεταβλητή \$amount όπου θα χρησιμοποιήσει το script του PayPal για να γίνει η πληρωμή. Σε αυτό το σημείο, έχουμε δημιουργήσει έναν λογαριασμό PayPal σαν επαγγελματίας ώστε να μπορούμε να λαμβάνουμε χρήματα. Περισσότερες πληροφορίες γι' αυτό θα βρείτε εδώ : <https://www.paypal.com/gr/business>. Κάθε λογαριασμός έχει και ένα μοναδικό αναγνωριστικό που πρέπει να περάσουμε σαν παράμετρο στο script της ενσωμάτωσης όπως θα δείτε παρακάτω.

```
<script src="/js/main.js"></script>  
<script src="https://www.paypal.com/sdk/js?client-id=XXXXXXXXXXXXX" ></script>  
<script src="index.js"></script>  
<script>  
    var payableAmount = <?php echo $amount; ?>;  
</script>  
<script>  
    var bookingid = <?php echo $bookingid; ?>;  
</script>  
</body>  
</html>
```

Αφού αποκτήσουμε και ενσωματώσουμε το script του Paypal σειρά έχει να δημιουργήσουμε και να συνδέσουμε ένα αρχείο JavaScript όπως αυτό που βλέπετε παραπάνω (index.js).

Μέσα στο αρχείο αυτό, όπως θα δείτε στον κώδικα που ακολουθεί, θα καθορίσουμε τις ενέργειες που θέλουμε να γίνουν όταν ο χρήστης ολοκληρώσει επιτυχώς την συναλλαγή του ή αν ακυρωθεί.

```
paypal.Buttons( {  
  createOrder: function (data,actions) {  
    return actions.order.create({  
      purchase_units: [{  
        amount:{  
          value:payableAmount  
        }  
      }]  
    });  
  },  
  onApprove: function(data,actions) {  
    return actions.order.capture().then(function(details){  
      console.log(details)  
      window.location.replace("payment-success.php")  
      document.cookie = "payment = " + bookingid;  
      console.log(document.cookie);  
    })  
  },  
  onCancel: function(data,actions) {  
    window.location.replace("payment-cancelled.php")  
  }  
}).render('#paypal-payment-button');
```

Από τον κώδικα που βρίσκεται παραπάνω, θα δείτε ότι περνάμε στο script το amount, δηλαδή την μεταβλητή που αντιστοιχεί στο ποσό που θέλουμε να πληρώσει ο χρήστης. Επίσης θέλουμε να πάρουμε και πληροφορία για το τελικό αποτέλεσμα της συναλλαγής, γι' αυτό το λόγο έχουμε δηλώσει τα onApprove και onCancel. Τι κάνουν αυτές οι δύο συναρτήσεις λίγο πολύ το καταλαβαίνουμε και από το όνομα τους. Η onApprove ανακατευθύνει τον χρήστη στη σελίδα payment-success.php δημιουργώντας παράλληλα και ένα cookie. Στο cookie αυτό,

που το έχουμε ονομάσει payment, αποθηκεύεται το bookingid της συναλλαγής που θα το χρειαστούμε για να ανανεώσουμε την εικόνα της συναλλαγής στη βάση μας. Η onCancel από την άλλη μεριά δεν κάνει κάτι παραπάνω από μια απλή ανακατεύθυνση του χρήστη στη σελίδα payment-cancelled.php όπου εκεί ο χρήστης ενημερώνεται ότι η συναλλαγή του για κάποιο λόγο δεν έχει ολοκληρωθεί και πρέπει να προσπαθήσει ξανά.

Για την περίπτωση λοιπόν που η συναλλαγή ολοκληρωθεί επιτυχώς και αφού έχει δημιουργηθεί το payment cookie μας, ο χρήστης ανακατευθύνεται στη σελίδα payment-success.php όπως είπαμε και παραπάνω. Εκεί, όπως θα δείτε στον τμήμα του κώδικα που ακολουθεί, μεταφέρουμε την πληροφορία του cookie μας στη τοπική μεταβλητή payment.

```
<div class="col-md-8 col-md-offset-2 text-center heading-section animate-box">
  <h3>Η συναλλαγή σας ολοκληρώθηκε!</h3>
  <p>Σας ευχαριστούμε για την πληρωμή.</p>
  <?php if(isset($_COOKIE['payment'])) {
    echo $payment ;
    $payment = $_COOKIE['payment'];
    $sql = "UPDATE bookings set paid='YES' where bookingid='$payment'";
    mysqli_query($link,$sql);
    setcookie("payment", "", time()-3600);
  } ?>
  
</div>
```

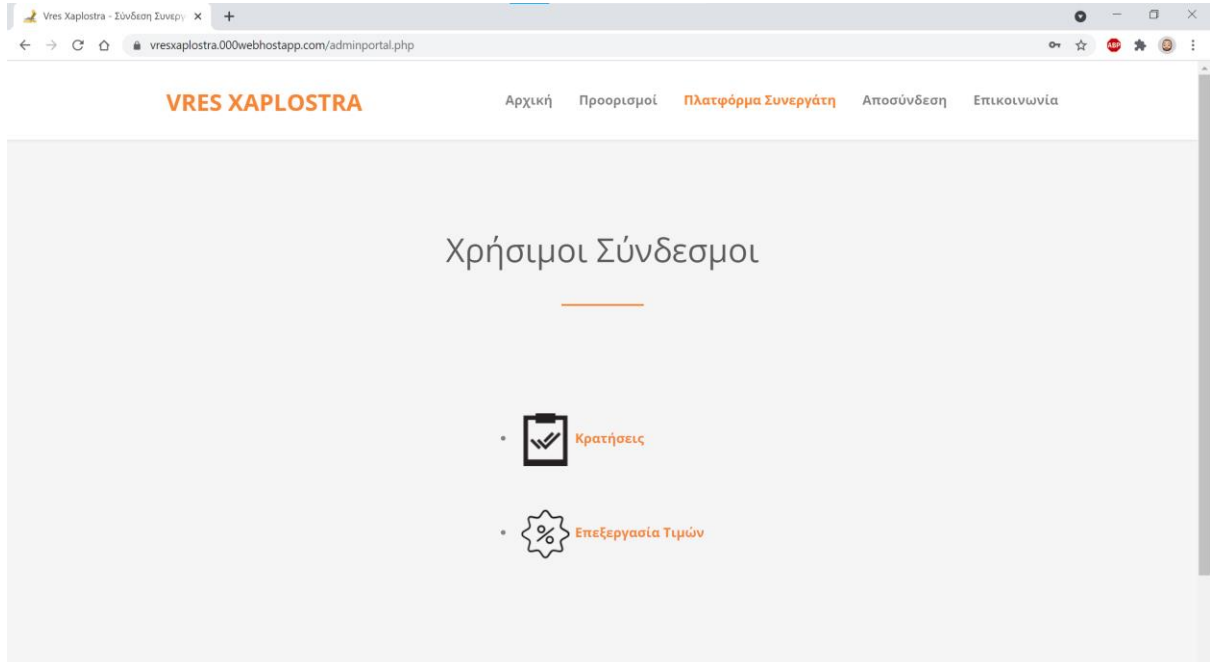
Στη συνέχεια με χρήση της εντολής \$sql κάνουμε update την τιμή paid της κράτησης σε “YES” στον πίνακα bookings.

Τέλος, για να είμαστε σίγουροι ότι ο χρήστης μας δεν θα πραγματοποιήσει ξανά πληρωμή για την ίδια κράτηση εφόσον έχει ολοκληρωθεί επιτυχώς, προσθέτουμε το παρακάτω τμήμα κώδικα στη σελίδα checkout.php έτσι ώστε να ρωτά τη βάση μας αν το συγκεκριμένο bookingid είναι πληρωμένο, έτσι ώστε αν είναι να τον κάνει κατευθείαν ανακατεύθυνση στη σελίδα payment-success.php χωρίς να τον αφήνει να πληρώσει ξανά.

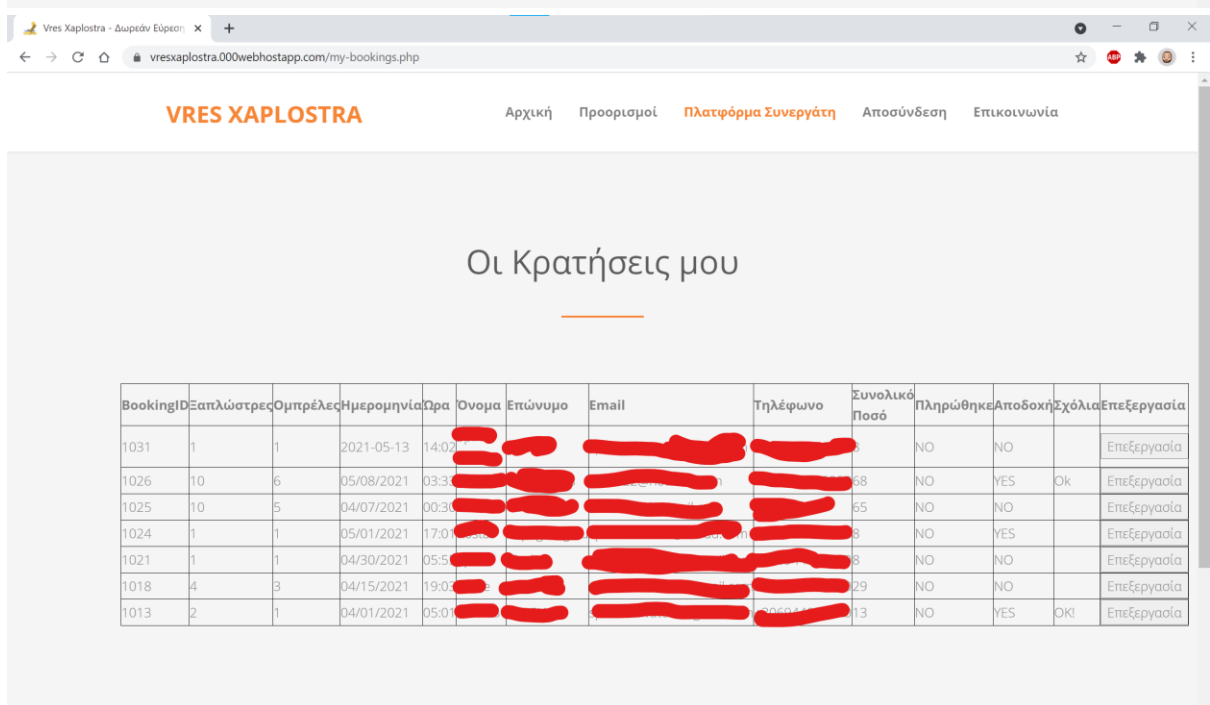
```
if( $paid == 'YES')
{
    header("location: payment-success.php");
    exit;
}
?>
```

Κεφάλαιο 4ο : Παρουσίαση λειτουργίας εργασίας

4.1 Παρουσίαση πλατφόρμας σαν Συνεργάτης



The screenshot shows the admin portal of VRES XAPLOSTRA. The header includes the logo and navigation links: Αρχική, Προορισμοί, Πλατφόρμα Συνεργάτη (highlighted), Αποσύνδεση, and Επικοινωνία. The main content area is titled 'Χρήσιμοι Σύνδεσμοι' (Useful Links) and contains two links: 'Κρατήσεις' (Bookings) with a clipboard icon and 'Επεξεργασία Τιμών' (Price Adjustment) with a percentage icon.



The screenshot shows the 'Οι Κρατήσεις μου' (My Bookings) section. It displays a table with the following columns: BookingID, Διαπλώστρες, Ομπρέλες, Ημερομηνία, Ωρα, Όνομα, Επώνυμο, Email, Τηλέφωνο, Συνολικό Ποσό, Πληρώθηκε, Αποδοχή, Σχόλια, and Επεξεργασία. The table contains 7 rows of booking data, with some fields redacted for privacy.

BookingID	Διαπλώστρες	Ομπρέλες	Ημερομηνία	Ωρα	Όνομα	Επώνυμο	Email	Τηλέφωνο	Συνολικό Ποσό	Πληρώθηκε	Αποδοχή	Σχόλια	Επεξεργασία
1031	1	1	2021-05-13	14:02	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	68	NO	NO		Επεξεργασία
1026	10	6	05/08/2021	03:30	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	68	NO	YES	OK	Επεξεργασία
1025	10	5	04/07/2021	00:30	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	65	NO	NO		Επεξεργασία
1024	1	1	05/01/2021	17:00	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	68	NO	YES		Επεξεργασία
1021	1	1	04/30/2021	05:50	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	68	NO	NO		Επεξεργασία
1018	4	3	04/15/2021	19:00	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	29	NO	NO		Επεξεργασία
1013	2	1	04/01/2021	05:00	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	13	NO	YES	OK!	Επεξεργασία

Vres Xaplostra - Δωριάν Εύρεση x +

vresxaplostra.000webhostapp.com/edit-booking.php?bookingid=1018

VRES XAPLOSTRA Αρχική Προορισμοί Πλατφόρμα Συνεργάτη Αποσύνδεση Επικοινωνία

Στοιχεία Κράτησης

📍 ΟΝΟΜΑ: name

📍 ΕΠΩΝΥΜΟ: surname

📞 ΤΗΛΕΦΩΝΟ: [REDACTED]

✉ EMAIL: [REDACTED]

🌐 ΠΑΡΑΛΙΑ: ΑΡΙΛΛΑΣ

📅 ΗΜΕΡΟΜΗΝΙΑ: 04/15/2021

🕒 ΩΡΑ: 19:03

🍴 ΞΑΠΛΩΣΤΡΕΣ: 4

👤 ΟΜΠΡΕΛΕΣ: 3

💰 ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ: 29

📦 ΠΛΗΡΩΘΗΚΕ: NO

📄 ΑΠΟΔΟΧΗ: NO

💬 ΣΧΟΛΙΑ:

Αλλαγή Κράτησης

Κατάσταση Αποδοχής

Σχόλια *Μόνο στα Αγγλικά ή με Λατινικούς χαρακτήρες*

ΥΠΟΒΟΛΗ

Vres Xaplostra - Δωριάν Εύρεση x +

vresxaplostra.000webhostapp.com/edit-price.php

VRES XAPLOSTRA Αρχική Προορισμοί Πλατφόρμα Συνεργάτη Αποσύνδεση Επικοινωνία

Επεξεργασία Τιμών

Εδώ μπορείτε να μεταβάλετε τις τιμές σας

Υφιστάμενες Τιμές

Ξαπλώστρα: 5€ Ομπρέλα: 3€

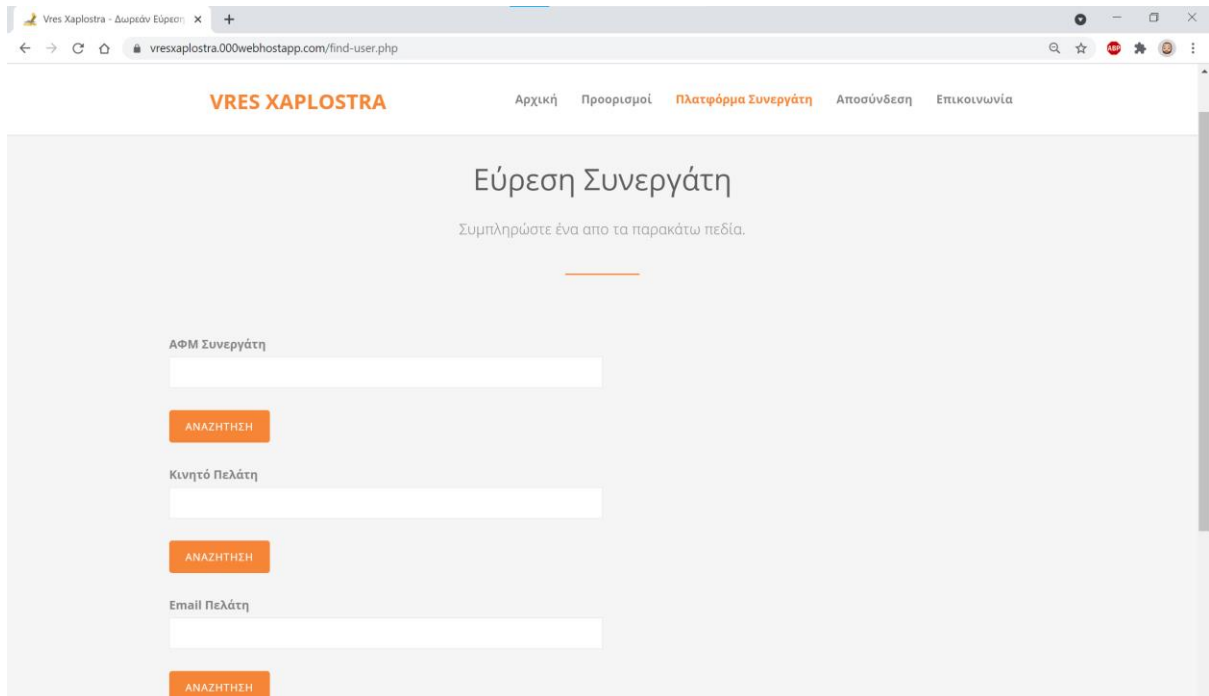
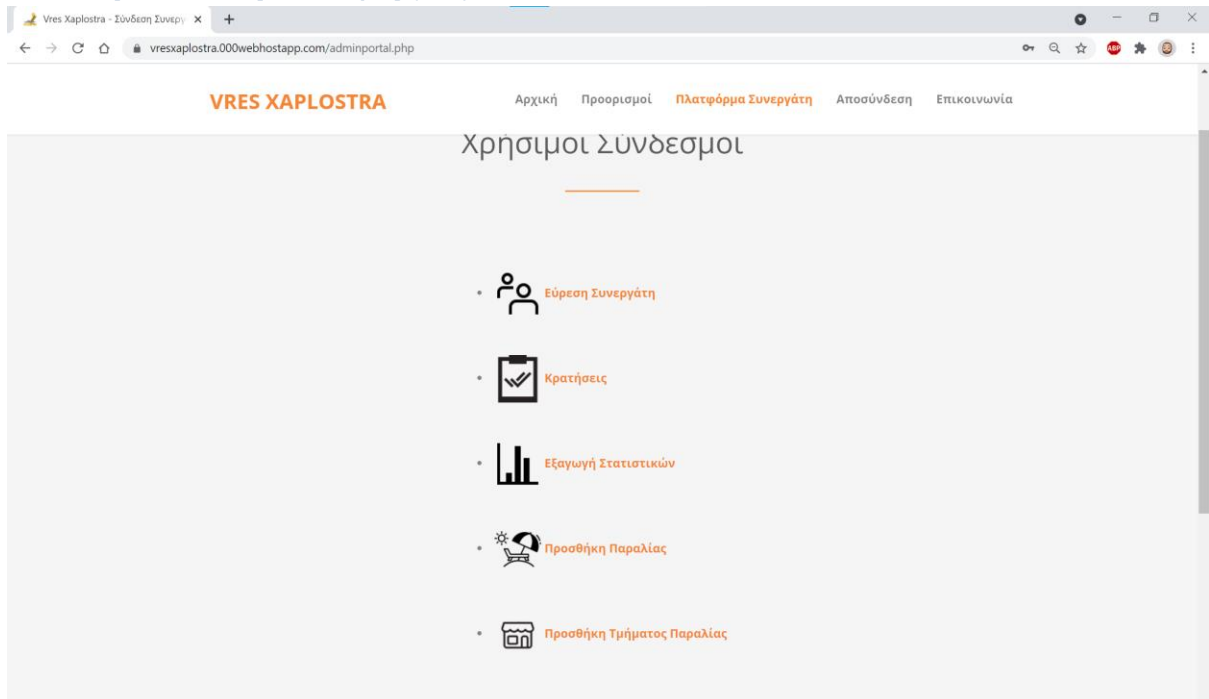
Τελευτ. επεξεργασία: 2021-04-14 22:34:51

Νέα Τιμή Ξαπλώστρας

Νέα Τιμή Ομπρέλας

ΥΠΟΒΟΛΗ

4.2 Παρουσίαση πλατφόρμας σαν Administrator



Vres Xaplostra - Δωριόν Εύρεση

vresxaplostra.000webhostapp.com/edit-user.php?user_id=32

VRES XAPLOSTRA Αρχική Προορισμοί Πλατφόρμα Συνεργάτη Αποσύνδεση Επικοινωνία

Υφιστάμενα Στοιχεία

- ΟΝΟΜΑ: Σπύρος
- ΕΠΩΝΥΜΟ: Βλάχος
- ΑΦΜ: [REDACTED]
- ΤΗΛΕΦΩΝΟ: [REDACTED]
- EMAIL: [REDACTED]
- ΝΟΜΟΣ: ΚΕΡΚΥΡΑΣ
- ΠΑΡΑΛΙΑ(Κωδ. Τμήματος): 2
- Δικαιώματα: Admin
- ΗΜ/ΝΙΑ ΕΓΓΡΑΦΗΣ: 2019-11-16 23:45:17

Αλλαγή Στοιχείων

Όνομα	Επώνυμο
ΑΦΜ	Τηλέφωνο
Email	Παραλία
Νομός Επιλέξτε νομό	Δικαιώματα

ΥΠΟΒΟΛΗ

Ανέβασμα φωτογραφίας για το τμήμα παραλίας 2

Choose File No file chosen

UPLOAD

Vres Xaplostra - Δωριόν Εύρεση

vresxaplostra.000webhostapp.com/export-data.php

VRES XAPLOSTRA Αρχική Προορισμοί Πλατφόρμα Συνεργάτη Αποσύνδεση Επικοινωνία

Εξαγωγή Στατιστικών

Εδώ μπορείτε να κατεβάσετε αναφορές σε excel

Εξαγωγή Στατιστικών Πόστου

Επιλογή Πόστου: [Επιλογή Πόστου]

ΥΠΟΒΟΛΗ

Εξαγωγή Στατιστικών Παραλίας

Επιλογή Παραλίας: [Επιλογή Παραλίας]

ΥΠΟΒΟΛΗ

Twitter Instagram

Vres Xaplostra - Διαρίων Εύρεση

vresxaplostra.000webhostapp.com/add-beach.php

VRES XAPLOSTRA

Αρχική Προορισμοί Πλατφόρμα Συνεργάτη Αποσύνδεση Επικοινωνία

Προσθήκη Παραλίας

Εδώ μπορείτε να καταχωρήσετε μια νέα παραλία στη βάση

Νουός

Όνομα

ΚΑΤΑΧΩΡΗΣΗ

Twitter Instagram

Vres Xaplostra - Διαρίων Εύρεση

vresxaplostra.000webhostapp.com/add-posto.php

VRES XAPLOSTRA

Αρχική Προορισμοί Πλατφόρμα Συνεργάτη Αποσύνδεση Επικοινωνία

Προσθήκη Τμήματος Παραλίας

Εδώ μπορείτε να καταχωρήσετε νέο τμήμα σε μια Παραλία

Επιλογή Συνεργάτη

Επιλογή Παραλίας

Latitude

Longitude

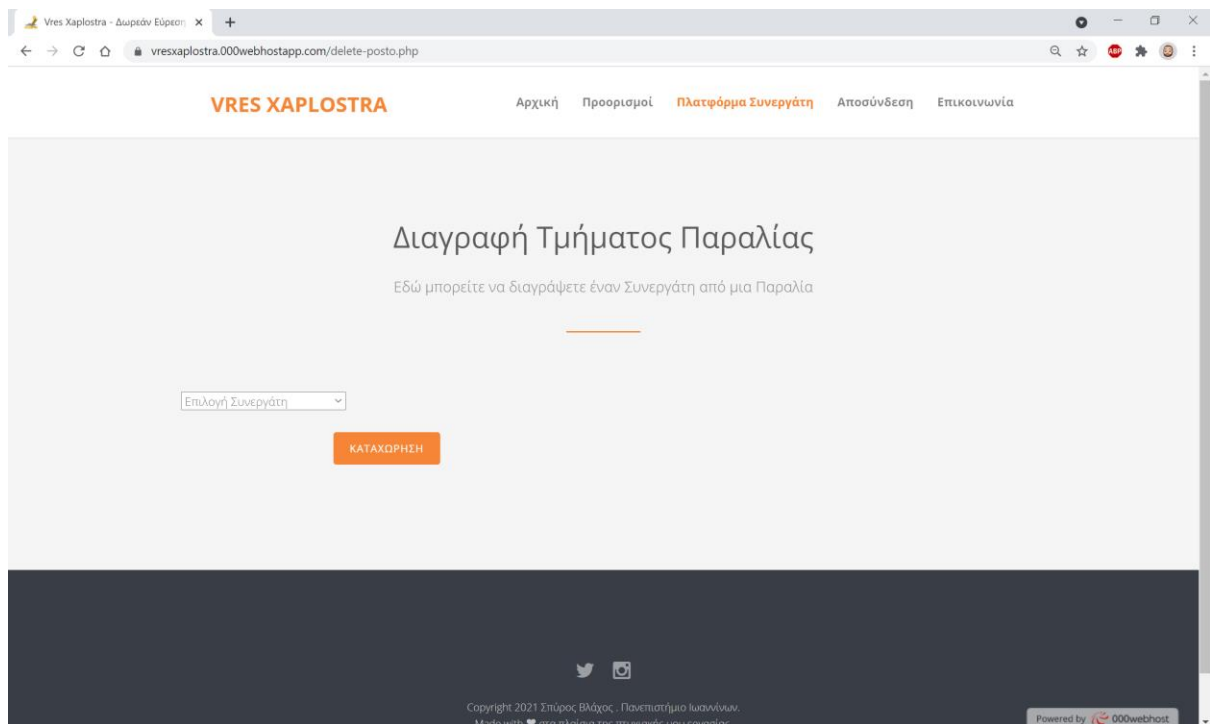
Τιμή Ξαπλώστρας

Τιμή Ομπρέλας

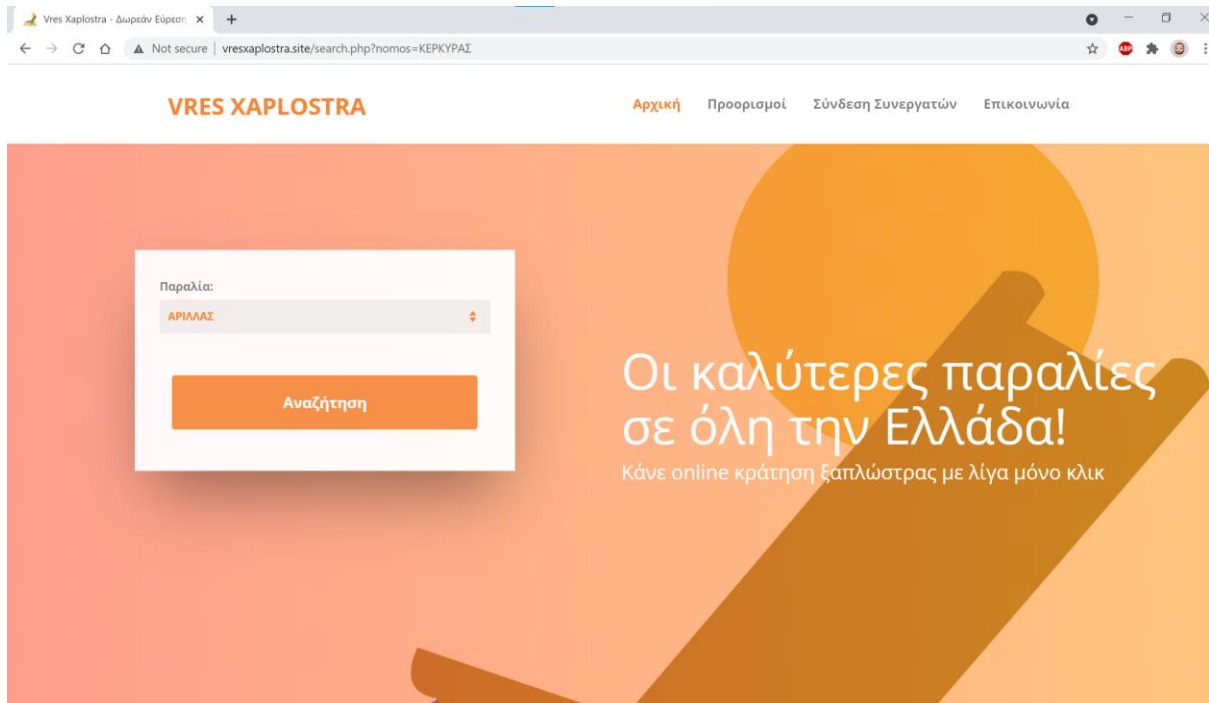
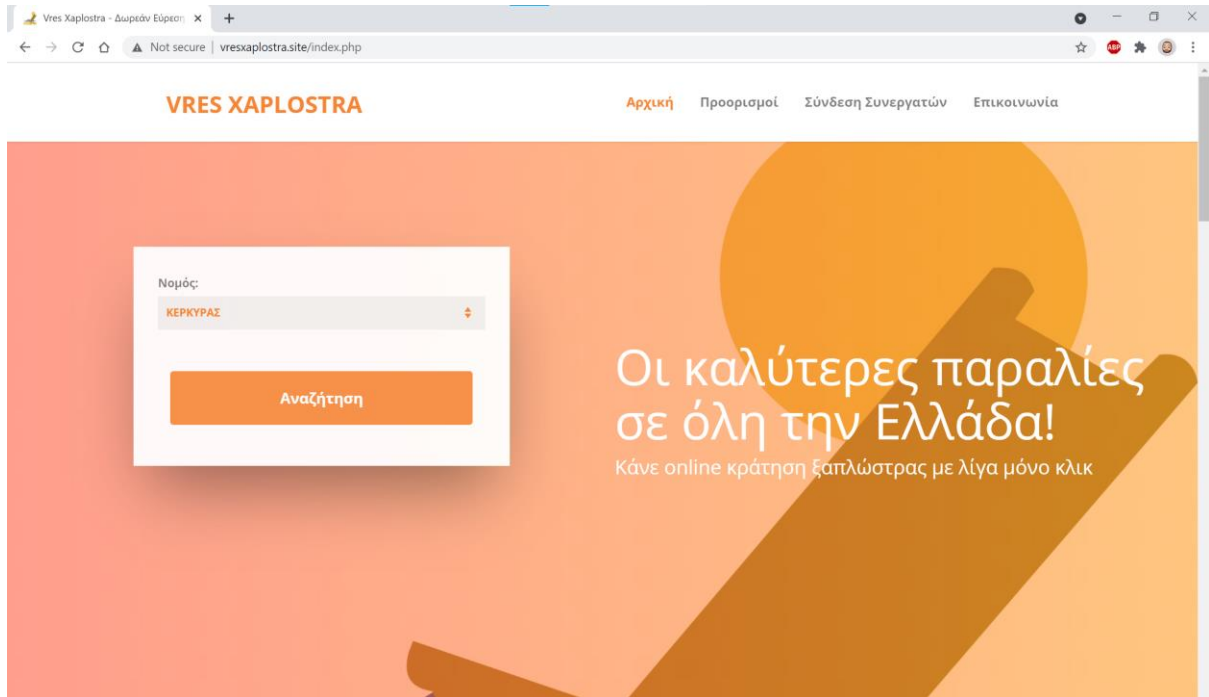
Διακριτό Όνομα

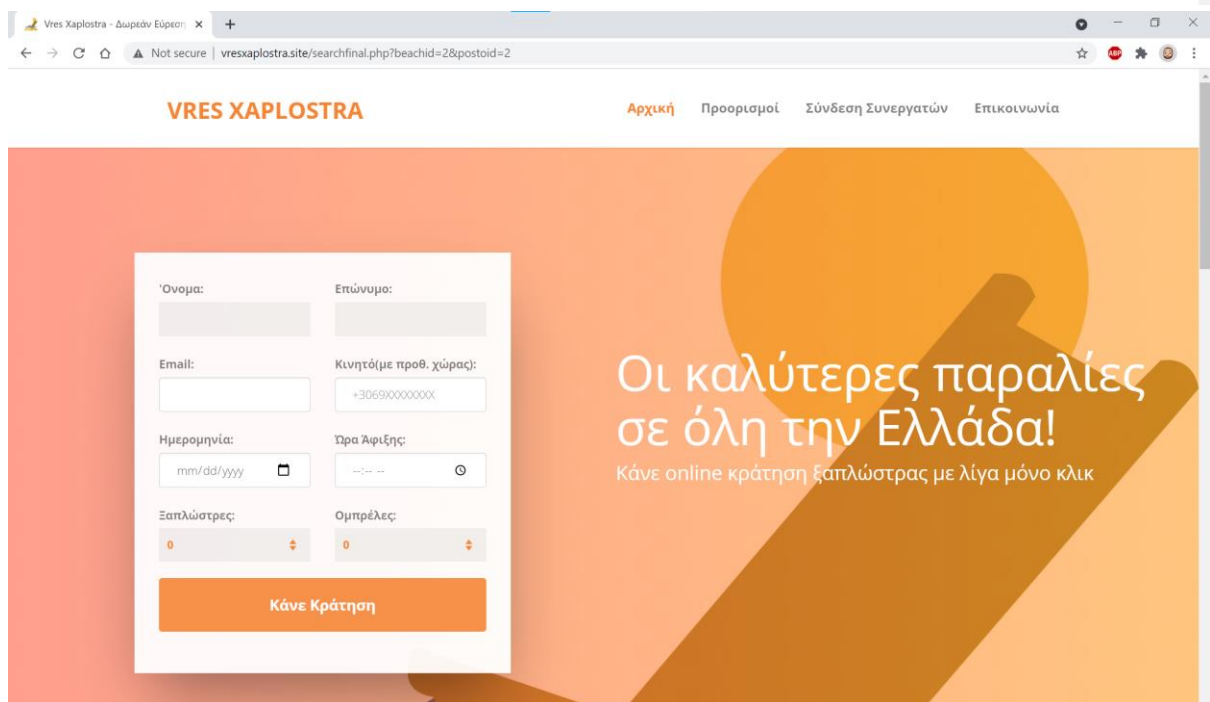
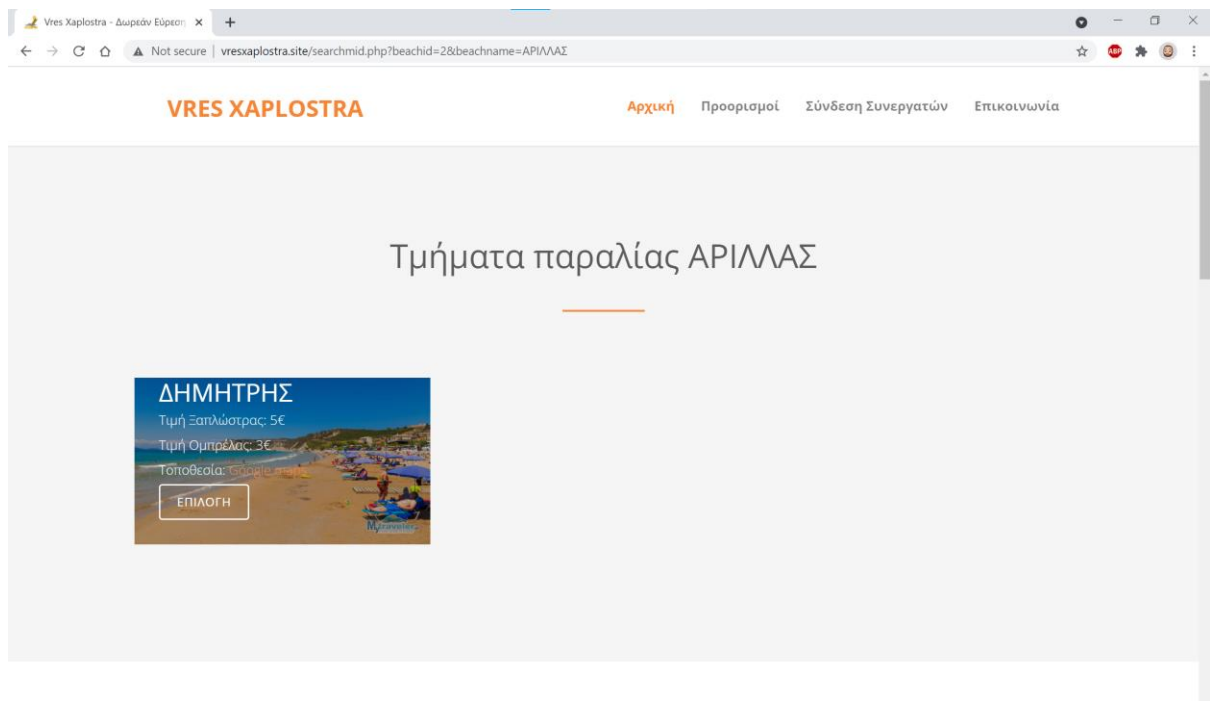
ΚΑΤΑΧΩΡΗΣΗ

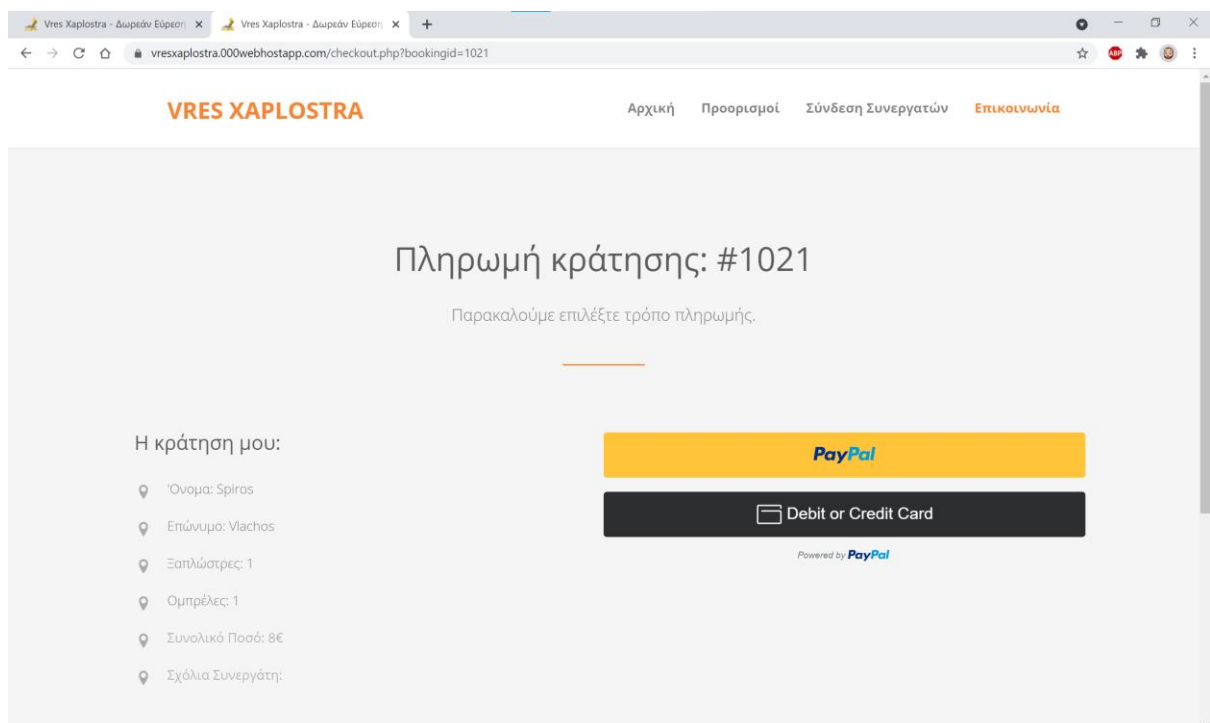
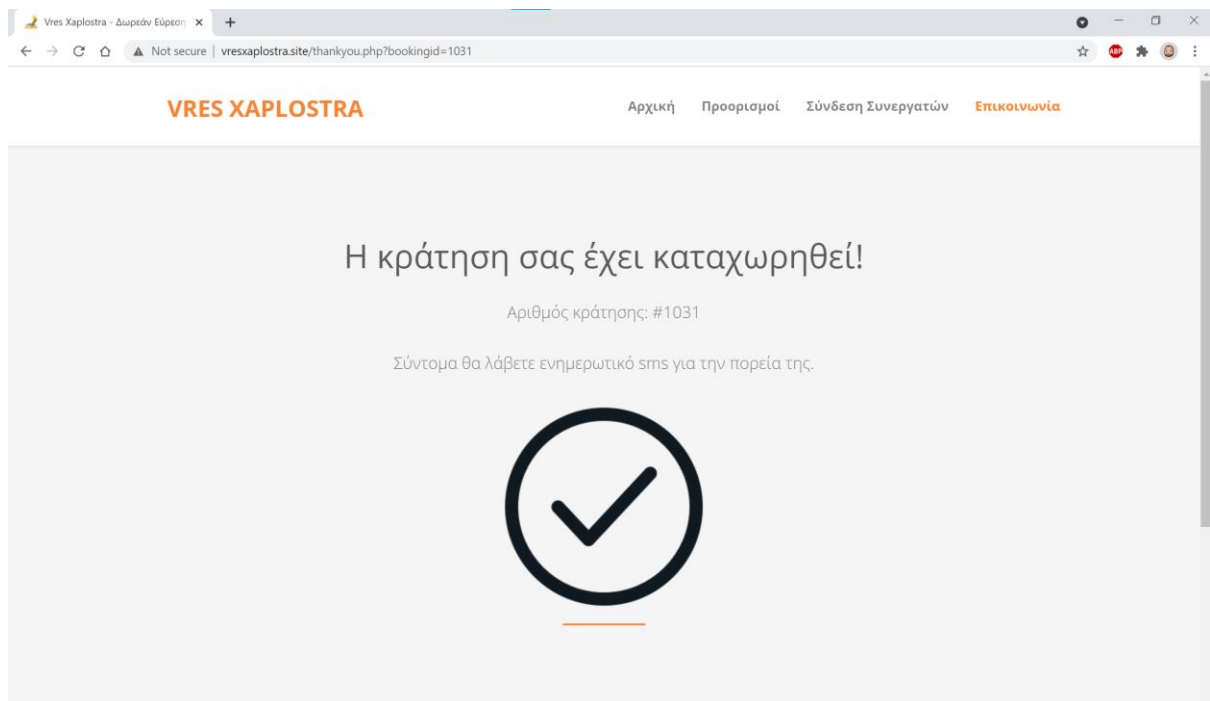
Powered by 000webhost



4.3 Παρουσίαση πλατφόρμας σαν Χρήστης







ΣΥΜΠΕΡΑΣΜΑΤΑ

Κατά την εκπόνηση της παρούσας Πτυχιακής Εργασίας δημιουργήσαμε μια καινοτόμα πλατφόρμα online κράτησης ξαπλωστρών. Αρχικά μελετήσαμε κάποια βασικά στοιχεία για τις γλώσσες php, html, JavaScript, bootstrap και sql.

Στη συνέχεια μάθαμε την ιστορία των γλωσσών προγραμματισμού, την εξέλιξη τους και πώς μπορεί κάποιος να αποκτήσει ένα δικό του όνομα domain καθώς και έναν hosting server ώστε να φιλοξενήσει την ιστοσελίδα του.

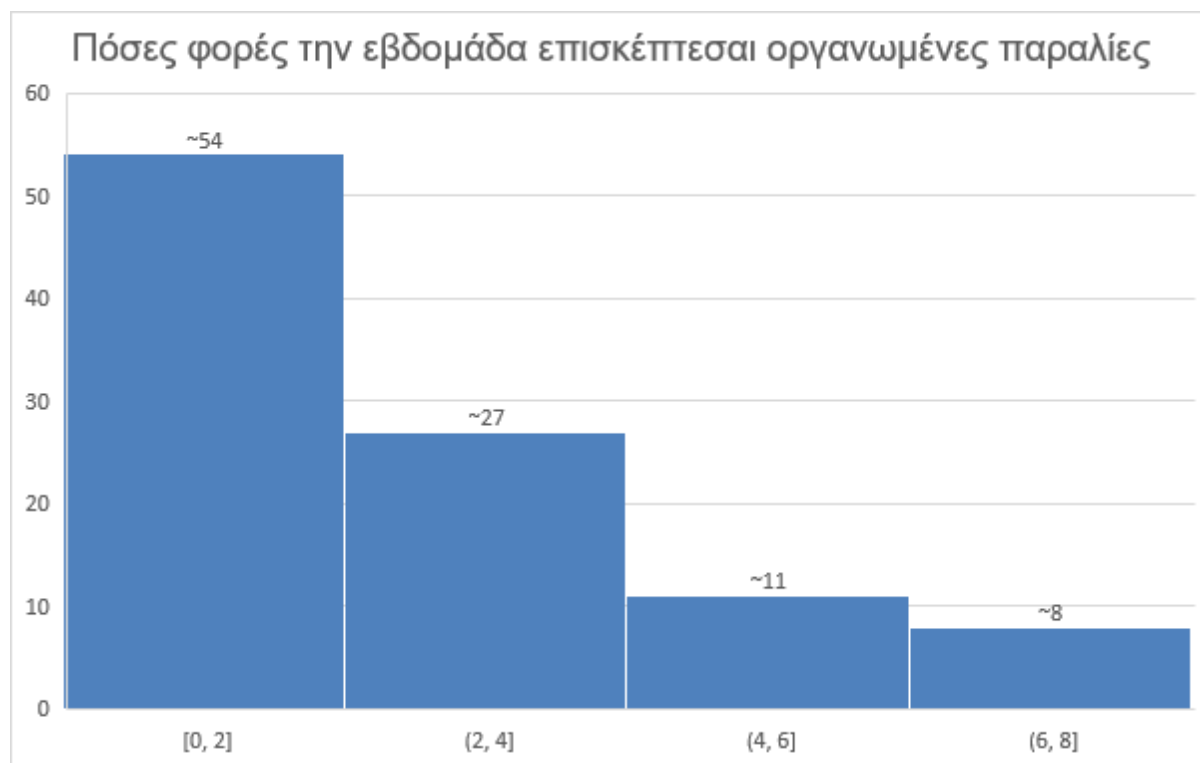
Στο τρίτο κεφάλαιο γνωρίσαμε την αρχιτεκτονική και τον τρόπο προγραμματισμού που βρίσκεται πίσω από την πλατφόρμα αυτή. Μάθαμε πρακτικές και μεθόδους, οι οποίες είναι χρήσιμες και χρησιμοποιούνται σε πολλές ιστοσελίδες στον έξω κόσμο.

Στο τελευταίο κεφάλαιο της παρούσας Πτυχιακής Εργασίας παρουσιάσαμε κάποια στιγμιότυπα από το γραφικό περιβάλλον της πλατφόρμας. Είδατε την εικόνα που έχει ένας απλός χρήστης που θα επισκεφτεί το www.vresxaplostra.site, την εικόνα που έχει ένας συνεργάτης και τέλος την εικόνα που έχει ένας διαχειριστής της πλατφόρμας.

Τέλος, από την παρακάτω έρευνα την οποία πραγματοποιήσα, συμπεραίνουμε ότι υπάρχει ένα μεγάλο ποσοστό ζήτησης στην αγορά για μια τέτοιου είδους πλατφόρμα σε ποσοστό μάλιστα που αγγίζει το 77% (πιθανό έως απόλυτα πιθανό) από τους ερωτηθέντες συμμετέχοντες.

ΕΡΕΥΝΑ

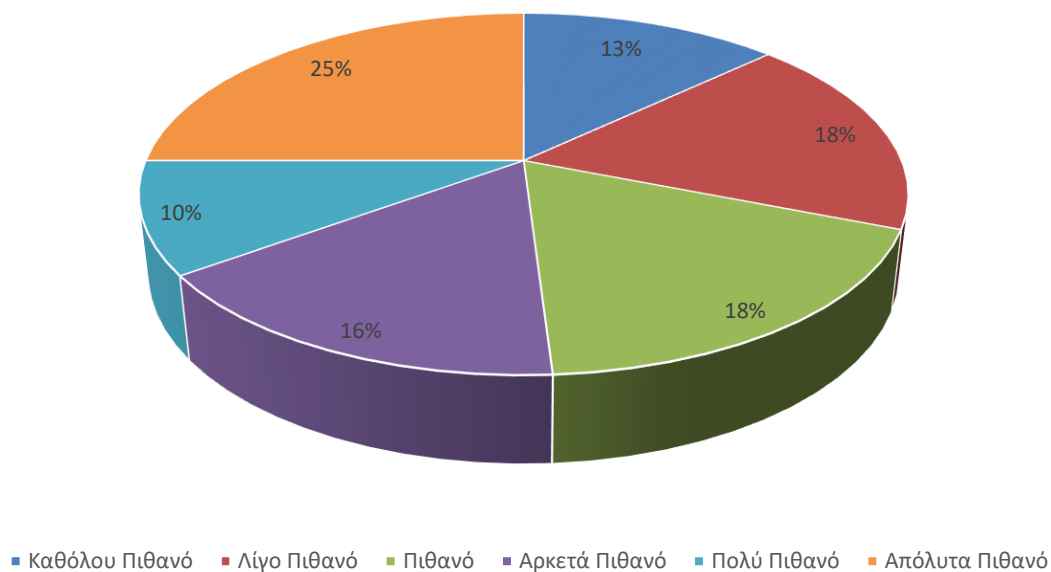
Τα παρακάτω στοιχεία, αποτελούν αποτέλεσμα έρευνας που πραγματοποιήθηκε σε 100 άτομα.



Γράφημα 1.

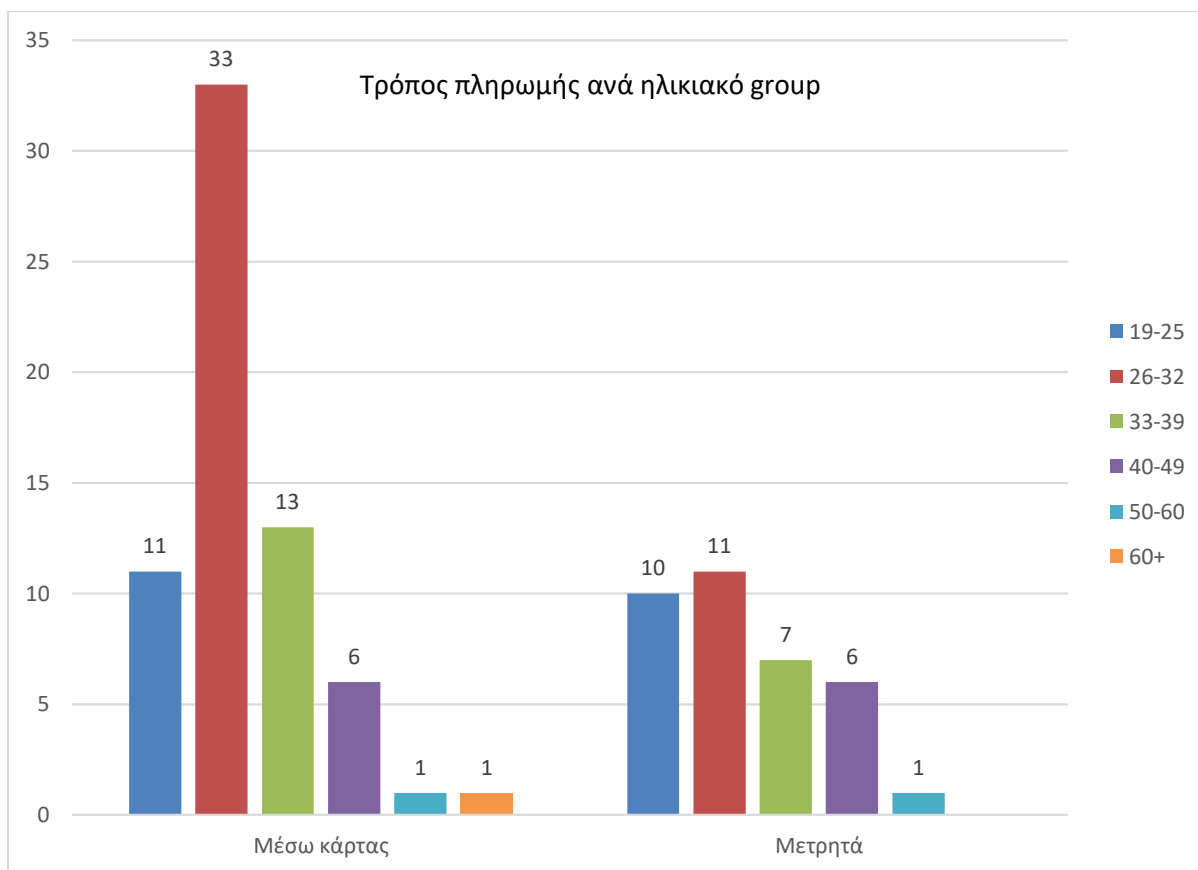
Από το παραπάνω γράφημα (1), συμπεραίνουμε τη συχνότητα επισκεψιμότητας των ερωτηθέντων ατόμων σε οργανωμένες παραλίες ανά εβδομάδα. Όπως θα δείτε, το 54% επισκέπτεται από 0-2 φορές την εβδομάδα την παραλία την περίοδο του καλοκαιριού, το 27% από 2-4 φορές, το 11% από 4-6 φορές ενώ ένα 8% από 6-7 φορές.

Πόσο πιθανό είναι να χρησιμοποιήσεις μια πλατφόρμα
κράτησης ξαπλωστών;



Γράφημα 2.

Από το παραπάνω γράφημα (2), συμπεραίνουμε την πιθανότητα χρήσης της πλατφόρμας μας. Το ποσοστό του 25% θεωρεί απόλυτα πιθανό να τη χρησιμοποιούσε, το ποσοστό του 10% θεωρεί πολύ πιθανό, το ποσοστό του 18% θεωρεί πιθανό, το ποσοστό του 18% θεωρεί λίγο πιθανό, ενώ το ποσοστό του 13% το θεωρεί καθόλου πιθανό.



Γράφημα 3.

Τέλος, από το γράφημα (3) συμπεραίνουμε τον προτιμητέο τρόπο πληρωμής ανά ηλικιακό γκρουπ. Όπως φαίνεται και παραπάνω, το 65% προτιμάει την πληρωμή μέσω κάρτας, ενώ μόλις το 35% των ερωτηθέντων θα πραγματοποιούσε την πληρωμή του με χρήση μετρητών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Elizabeth Castro (2003), Εισαγωγή στην HTML για τον παγκόσμιο ιστό, Εκδόσεις Κλειδάριθμος
- [2] Sams Publishing - Luke Welling and Laura Thomson (2001), PHP and MySQL Web Development
- [3] Julie C. Meloni (2003), Μάθετε PHP, MySQL και Apache σε 24 Ώρες, Εκδόσεις Μ. Γκιούρδας
- [4] ALEXANDRE OUELLETTE - [HTTPS://CAREERFOUNDRY.COM/EN/BLOG/WEB-DEVELOPMENT/WHAT-IS-BOOTSTRAP-A-BEGINNERS-GUIDE/](https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/)
https://www.w3schools.com/whatis/whatis_bootstrap.asp
- [5] Domantas G (Μάρτιος 2021) - <https://www.hostinger.com/tutorials/what-is-a-domain-name>
- [6] <https://www.ics.forth.gr/about-ics?lang=el>
- [7] <https://www.w3schools.com/tags/default.asp>