



Τμήμα Μηχανικών Πληροφορικής

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: ΤΟΥΡΙΣΤΙΚΟΣ ΟΔΗΓΟΣ ΕΛΑΣΣΟΝΑΣ ΜΕ ΧΡΗΣΗ PHP FRAMEWORK

ΣΠΟΥΔΑΣΤΗΣ: ΜΑΘΙΟΥΔΑΚΗΣ ΜΗΝΑΣ Α.Μ.:1834

ΣΠΟΥΔΑΣΤΗΣ: ΖΙΑΚΟΥΛΗΣ ΚΥΡΙΑΚΟΣ Α.Μ.:1729

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΠΑΠΑΔΑΚΗΣ ΝΙΚΟΛΑΟΣ



Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον καθηγητή μας κο. Νικόλαο Παπαδάκη για την καθοδήγηση και την βοήθεια που μας προσέφερε σε όλα τα στάδια της εκπόνησης της πτυχιακής μας εργασίας.

Τέλος θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας για όλη την στήριξη που μας παρείχαν.

Abstract

The purpose of this thesis is to create a Tourist Guide for the very city of Ellassona, providing useful information for the visitors and the locals, using Html, Php, Css, Js , JQuery,Xampp, FuelPhp programs and technologies.

There is a thorough description of the FuelPhp, Xampp installation as well as a brief explanation of their selection.

Σύνοψη

Η πτυχιακή αυτή εργασία έχει ως σκοπό την δημιουργία μίας ιστοσελίδας(Τουριστικού οδηγού) για την πόλη της Ελασσόνας με σκοπό την ενημέρωση των επισκεπτών και των ντόπιων κατοίκων.

Στις παρακάτω σελίδες θα έχετε την ευκαιρία να δείτε με λεπτομέρεια το κάθε βήμα που κάναμε για να φέρουμε την ιστοσελίδα μας στην τελική της μορφή χρησιμοποιώντας html,css,js,jquery,ajax , ενώ θα υπάρξει και αναλυτική περιγραφή της εγκατάστασης του Xampp και του FuelPhp και του γιατί επιλέχθηκαν.

Περιεχόμενα

Σύνοψη	4
1.1 Περίληψη.....	8
1.4 Δομή Εργασίας.....	9
2. Μεθοδολογία Υλοποίησης	9
3. Σχέδιο Δράσης.....	10
4. Κύριο μέρος Πτυχιακής	11
4.1 Ανάλυση προβλήματος.....	11
4.1.2 Απαιτήσεις συστήματος	12
4.2 Σχεδιασμός υλοποίησης	12
4.3 Υλοποίηση	17
4.3.1 Σχεδιασμός βάσης δεδομένων.....	17
4.3.2 Εγκατάσταση του framework.....	21
5. Αποτελέσματα.....	65
5.1 Συμπεράσματα.....	65
5.2 Μελλοντική χρήση	65
Βιβλιογραφία.....	66
Παράρτημα.....	66
Controller reservation	66
Class ReserveQueries	77
Class MyRules.....	80
App/Bootstrap.php.....	80
Controller_Users	81

Πίνακας Εικόνων.

Εικόνα 3	15
Εικόνα 4	16
Εικόνα 5	18
Εικόνα 6	20
Εικόνα 7	20
Εικόνα 8	20
Εικόνα 9	20
Εικόνα 10	20
Εικόνα 11	20
Εικόνα 12	20
Εικόνα 13	20
Εικόνα 14	20
Εικόνα 15	21
Εικόνα 16	21
Εικόνα 17	25
Εικόνα 18	26
Εικόνα 19	27
Εικόνα 20	28
Εικόνα 21	28
Εικόνα 22	28
Εικόνα 23	29
Εικόνα 24	29
Εικόνα 25	30
Εικόνα 26	30
Εικόνα 27	31
Εικόνα 28	31
Εικόνα 29	32
Εικόνα 30	32
Εικόνα 31	33
Εικόνα 32	34
Εικόνα 33	35

Εικόνα 34	37
Εικόνα 35	38
Εικόνα 36	39
Εικόνα 37	40
Εικόνα 38	41
Εικόνα 39	42
Εικόνα 40	43
Εικόνα 41	44
Εικόνα 42	44
Εικόνα 43	45
Εικόνα 44	46
Εικόνα 45	51
Εικόνα 46	51
Εικόνα 47	52
Εικόνα 48	53
Εικόνα 49	53
Εικόνα 50	54
Εικόνα 51	54
Εικόνα 52	55
Εικόνα 53	56
Εικόνα 54	57
Εικόνα 55	57
Εικόνα 56	58
Εικόνα 57	58
Εικόνα 58	59
Εικόνα 59	59
Εικόνα 60	60
Εικόνα 61	63
Εικόνα 62	64

1.1 Περίληψη

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός ιστότοπου για την πόλη της Ελασσόνας. Πρόκειται ουσιαστικά για έναν τουριστικό οδηγό που στοχεύει τόσο στην διευκόλυνση των επισκεπτών(τουριστών) παρέχοντας τους υπηρεσίες εστίασης, μεταφοράς κ.α. καθώς επίσης και στην ενημέρωση των ντόπιων κατοίκων της Ελασσόνας για τα διάφορα κοινωνικά δρώμενα που λαμβάνουν μέρος στην πόλη.

Όλοι οι χρήστες της ιστοσελίδας θα έχουν δικαίωμα να εγγραφούν σε αυτή και να συμμετέχουν ως τον βαθμό που τους επιτρέπεται είτε με σχόλια σε διάφορα αντικείμενα είτε μέσω του μενού επικοινωνίας με τους διαχειριστές. Θα μπορούν επίσης να ρυθμίζουν τα προσωπικά τους στοιχεία και θα έχουν το δικαίωμα να κάνουν εύκολες και γρήγορες online κρατήσεις εφόσον το επιθυμούν.

Το περιεχόμενο της σελίδας μας έχει την δυνατότητα να ανανεώνεται δυναμικά από τους διαχειριστές του έχοντας ως αποτέλεσμα την ευζωία και αναβάθμιση των υπηρεσιών που παρέχει στο ευρύτερο κοινό, καθιστώντας το ως ένα πολύ δημοφιλές και δυνατό μέσο επικοινωνίας και ενημέρωσης.

Βάση του τρόπου κατασκευής της μας είναι εξαιρετικά εύκολο να αναβαθμίσουμε την λειτουργία της και να δημιουργήσουμε περαιτέρω ενότητες που θα καλύπτουν ενδεχόμενα μελλοντικά ζητήματα, όπως παραδείγματος χάρη την προβολή των τοπικών προϊόντων και των παραγωγών τους.

1.2 Κίνητρο για την διεξαγωγή της εργασίας

Αδιαμφισβήτητο το μεγαλύτερο κίνητρο για την εργασία αυτή ήταν η προσαρμογή μας σε ένα ρεαλιστικό περιβάλλον εργασίας χρησιμοποιώντας τις πιο σύγχρονες/επίκαιρες τεχνολογίες που υπάρχουν αυτή την στιγμή στην αγορά και που στον τομέα του web development είναι το κτίσιμο μιας ιστοσελίδας χρησιμοποιώντας php framework.

1.3 Σκοπός και Στόχοι της εργασίας

Σκοπός της ΠΕ μας ήταν να δημιουργήσουμε ένα εύχρηστο και φιλικό περιβάλλον τόσο για τους επισκέπτες όσο και για τους διαχειριστές του. Προβλέπονται επίσης η βελτιστοποίηση της σελίδας μας α) σε σχέση με όλους τους σύγχρονους τουριστικού οδηγούς που υπάρχουν αυτή την στιγμή στο διαδίκτυο και β) η ωριοποίηση του user interface σε επίπεδο αισθητικής και εμφάνισης. Εξίσου σημαντικό είναι να βρεθούν λύσεις που να υποστηρίζουν σημαντικές αρχές όπως παραδείγματος χάρη

- Cross-Browser functionality (Chrome, Mozilla Firefox, Opera, IE, etc)
- Cross-Platform functionality (Windows, Mac OS, Linux)
- Cross-Device functionality (tablets, cellphones, etc)

1.4 Δομή Εργασίας

Στα ακόλουθα κεφάλαια θα επεκταθούμε στην επεξήγηση του κώδικα μας, στα περιθώρια βελτιστοποίησης της ιστοσελίδας μας καθώς επίσης θα παραθέσουμε αρκετές εικόνες με βήμα βήμα το στήσιμο της σελίδας.

2. Μεθοδολογία Υλοποίησης

Για την εκπόνηση της Π.Ε. χρησιμοποιήθηκε το mvc framework FuelPhp το οποίο δίνει εύκολες και εύχρηστες λύσεις στο γράψιμο κώδικα σε γλώσσα php και ένα Lampp stack το οποίο αποτελείται από

- Apache Http server 2.2 (Εξυπηρετητής παγκόσμιου ιστού)
- MySQL 5.5.40 Database (Βάση δεδομένων)
- MySQL engine InnoDB (Για υποστήριξη foreign keys)
- Php 5.5.9 (Εξέχουσα γλώσσα προγραμματισμού για web development)
- Adminer (Software tool φτιαγμένο για την διαχείριση της βάσης δεδομένων)

Όλος ο κώδικας της ιστοσελίδας έχει γραφτεί στο Sublime text editor λόγω των απίστευτα εξυπηρετικών features που παρέχει όπως

- Goto Anything - ταχύτητα εύρεσης αρχείων με quick search μέσα στο project που δουλεύουμε
- Multiple selections - για να αλλάζει το όνομα μιας μεταβλητής παντού στον κώδικα μας.
- Split editing – Μπορούμε να έχουμε 2 η παραπάνω παράθυρα ανοιχτά για να συγκρίνουμε κώδικα.

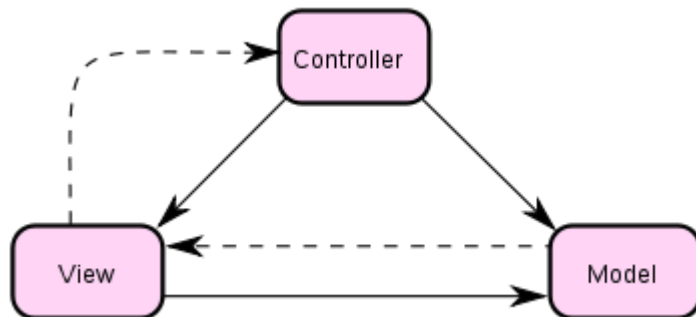
Τέλος χρησιμοποιήσαμε το **Git** για version control και το <https://bitbucket.org> για το hosting του κώδικα μας, πράγμα που βελτίωσε σε μεγάλο βαθμό την αποδοτικότητα και το επίπεδο της συνεργασίας μας.

2.1 Πρότυπα σχεδιασμού

MVC

Το M.V.C. αποτελεί ένα πρότυπο σχεδιασμού το οποίο βασίζεται στην ιδέα του να διαχωρίζουμε την λογική μιας εφαρμογής από το κομμάτι της παρουσίασης της.

Το MVC μας προτρέπει να χωρίζουμε την εφαρμογή μας σε 3 κομμάτια γνωστά ως Model View και Controller.



Όπως παρατηρούμε στην παραπάνω εικόνα, με τις έντονες γραμμές επισημαίνεται η σχέση των κομματιών αυτών ενώ με τις διακεκομμένες γραμμές διαφαίνονται οι σχέσεις που φαίνονται προφανείς από την πλευρά του χρήστη χωρίς αυτές να έχουν συμπεριληφθεί στο κομμάτι του σχεδιασμού της εφαρμογής.

Ενδεικτικά αναφέρουμε ένα παράδειγμα για να γίνει πιο κατανοητή η ιδέα του MVC

- Ο χρήστης αλληλεπιδρά με το view – κάνοντας κλικ σε ένα link ή κάνοντας submit σε μια φόρμα.
- Ο Controller διαχειρίζεται το input του χρήστη και μεταφέρει την πληροφορία στο Model
- Το Model λαμβάνει την πληροφορία και αναβαθμίζει την κατάσταση της (π.χ. προσθέτοντας δεδομένα σε μια βάση)
- Το View κάνει έλεγχο για την κατάσταση του Model και δρα ανάλογα
- Το View περιμένει για την επόμενη αλληλεπίδραση του χρήστη

3. Σχέδιο Δράσης

Γνωρίζοντας την ιστορία των frameworks που ξεκίνησε με το Ruby on rails στο χώρο του web development άρχισε να επικρατεί μια νέα τάση. Αρχίσανε να γράφοντα πολλά frameworks για php γεγονός που κάνει τον κώδικα πιο οργανωμένο και κατανοητό. Ένας από τους λόγους για την ραγδαία εξάπλωση των frameworks στον χώρο του web development είναι και η χρήση του μοντέλου αρχιτεκτονικής λογισμικού mvc (model view controller).

Ορμώμενοι των παραπάνω αποφασίσαμε να υλοποιήσουμε την πτυχιακή μας εργασία βασισμένοι σε αυτόν τον τρόπο λειτουργίας.

Όσον αφορά το framework και την χρήση του αυτό καθ' αυτό δεν μπορούμε να πούμε πως τα πράγματα γίνονται καλύτερα ή χειρότερα, ωστόσο η ίδια έρχεται με μερικά πλεονεκτήματα και κάποια μειονεκτήματα τα οποία βρίσκονται στην κρίση του εκάστοτε Developer για να εκτιμήσει.

Πλεονεκτήματα

- **Αποτελεσματικότητα,**
Projects τα οποία θα έπαιρναν ώρες και ατελείωτες γραμμές κώδικα για να υλοποιηθούν τώρα γίνονται σε μερικά μόνο λεπτά χρησιμοποιώντας τις έτοιμες συναρτήσεις που παρέχει το εκάστοτε framework.
- **Ασφάλεια,**
Ένα ευρέως διαδεδομένο framework έχει σπουδαίες εφαρμογές ασφάλειας. Επίσης λόγω της διασημότητας του έχει ένα εξίσου μεγάλο community από χρήστες που θα ενημερώσουν την ομάδα διαχείρισης σε περίπτωση που εντοπίσουν κάποιο κενό ασφαλείας προκειμένου αυτό να διορθωθεί.
- **Κόστος**
Τα περισσότερα και δημοφιλέστερα framework είναι free για το κοινό, πράγμα που συμβάλει και στην τελική τιμή μια εργασίας.
- **Υποστήριξη**
Τα frameworks έχουν πάντα documentation για τους χρήστες, ομάδες υποστήριξης προβλημάτων και αρκετά forum επικοινωνίας που μας παρέχουν γρήγορη και ουσιαστική βοήθεια.

Μειονεκτήματα

- **Μαθαίνεις το Framework και όχι την Γλώσσα**
- **Περιορισμοί**
Γνωρίζοντας ότι ο πυρήνας του Framework δεν μπορεί να τροποποιηθεί πρέπει να είμαστε προσεκτικοί και ενήμεροι για τις δυνατότητες και τα όρια του επιλέγοντας την χρήση του μόνο σε περίπτωση που είμαστε σίγουροι ότι εκπληρώνει όλες μας τις ανάγκες. Public Code
Ο κώδικας είναι προσβάσιμος από όλους και ιδιαίτερα από άτομα με «κακές προθέσεις».

4. Κύριο μέρος Πτυχιακής

4.1 Ανάλυση προβλήματος

Για να μπορέσουμε να κεντρίσουμε περισσότερο το τουριστικό ενδιαφέρον ώστε να μπορέσει να ανέβει η κοινότητα της Ελασσόνας δημιουργήσαμε μια δυναμική ιστοσελίδα η οποία παραθέτει τις δυνατότητες που μπορεί να παρέχει στους πολίτες και στους επισκέπτες της.

Πιο αναλυτικά αναφέρονται δρομολόγια, δραστηριότητες, ξενοδοχεία καθώς και κέντρα ενοικιάσεων οχημάτων. Μέσα από την σελίδα ο χρήστης μπορεί να βρει πληροφορίες σχετικά με την πόλη της Ελασσόνας και να γίνει ένα ενεργό μέρος στα δρώμενα κοινοποιώντας την δική του άποψη αλλά και να διαβάσει άλλων ώστε να ενημερωθεί για τα εκάστοτε ενδιαφέροντα του.

Παρόλα αυτά η κύρια λειτουργία τις ιστοσελίδας μας επικεντρώνεται στην online αναζήτηση και κράτηση δωματίων στα ξενοδοχεία της Ελασσόνας, παράλληλα προσφέρεται η δυνατότητα ενοικίασης οχημάτων.

4.1.2 Απαιτήσεις συστήματος

- Δυναμική διαχείριση περιεχομένου μέσω του administration panel
- Responsive templating για την εμφάνιση του περιεχομένου της σελίδας μας σε κινητά τηλέφωνα και tablets
- Εύκολη αναζήτηση μέσω του plugin auto complete της jquery
- Ατομική σελίδα χρήστη για εμφάνιση εξειδικευμένου περιεχομένου
- Αναζήτηση στην βάση δεδομένων για τα διαθέσιμα δωμάτια από την λίστα των ξενοδοχείων με βάση των αναγκών του χρήστη
- Δυνατότητα μετάφρασης της σελίδας μας σε παραπάνω από μια γλώσσες
- Φόρμα επικοινωνίας με τους administrator της σελίδας

4.2 Σχεδιασμός υλοποίησης

Μετά από αναζήτηση επιλέξαμε το unify template για την παρουσίαση του frontend της ιστοσελίδας μας το οποίο είναι βασισμένο σε bootstrap prototype template engine και είναι responsive.

Για το backend (administrator page) χρησιμοποιήσαμε το start bootstrap-SB ADMIN template.

Στη συνέχεια εγκαταστήσαμε έναν profiler για php τον quick php profiler ο οποίος μας βοηθάει να πάρουμε χρήσιμες πληροφορίες όπως

- Load Time
μας ενημερώνει για το πόσο γρήγορα φορτώνουνε τα διάφορα κομμάτια της σελίδας μας
- Query viewer
δυνατότητα οπτικής παρουσίασης των queries που τρέχουν πίσω από την κάθε σελίδα καθώς και τον χρόνο εκτέλεσης τους.
- Memory used
Το μέγεθος των δεδομένων που αποθηκεύονται στην μνήμη
- Session variables
μπορούμε να δούμε κάθε φορά της μεταβλητές που έχουμε αποθηκεύσει στο session
- Get Variables
προβολή των μεταβλητών που έχουν δημιουργηθεί από την http μέθοδο get
- Post Variables
προβολή των μεταβλητών που έχουν δημιουργηθεί από την http μέθοδο post

Εγκαθιστώντας το **datepicker** plugin του jquery-ui και παραμετροποιώντας το πετύχαμε μια καλαίσθητη παρουσίαση και επιλογή των ημερομηνιών από τον χρήστη.

Παράθεση κώδικα

```
$('#start').datepicker({
  dateFormat: 'yy-mm-dd',
  prevText: '<i class="fa fa-angle-left"></i>',
  nextText: '<i class="fa fa-angle-right"></i>',
  minDate: 0,
  onSelect: function( selectedDate )
  {
    var date = $(this).datepicker('getDate');
    if (date) {
      date.setDate(date.getDate() + 1);
    }
    $('#finish').datepicker('option', 'minDate', date );
  }
});
$('#finish').datepicker({
  dateFormat: 'yy-mm-dd',
  prevText: '<i class="fa fa-angle-left"></i>',
  nextText: '<i class="fa fa-angle-right"></i>',
  minDate: 1,
  onSelect: function( selectedDate )
  {
    //$('#start').datepicker('option', 'maxDate', selectedDate);
    $('#searchbtn').removeAttr('disabled');
  }
});
```

1 datepicker

Με αυτή την παραμετροποίηση αποκλείουμε το ενδεχόμενο εκχώρησης ημερομηνιών προγενέστερων της σημερινής, αυτόματα απενεργοποιούνται οι προηγούμενες ημερομηνίες έτσι ώστε ο χρήστης να κάνει σωστή καταχώρηση για ένα εύρος ημερομηνιών.

Κάναμε εγκατάσταση του plugin **Autocomplete – jquery.js** ώστε να γίνεται αυτόματη συμπλήρωση των ονομάτων των ξενοδοχείων και των εταιριών ενοικιάσεων οχημάτων για την μεγαλύτερη διευκόλυνση του χρήστη

```

$.ready(function(){
  var options, a;
  jQuery(function(){
    a = $('#query').autocomplete(options);
  });
});
var a = $('#query').autocomplete({
  serviceUrl:rootUrl+'/search/'+foo+'.php',
  minChars:1,
  delimiter: /(,|;)\s*/, // regex or character
  maxHeight:400,
  width:201,
  zIndex: 9999,
  deferRequestBy: 0, //milliseconds
  noCache: false, //default is false, set to true to disable caching
  showNoSuggestionNotice: true,
});

```

1 autocomplete

Μία ακόμα τεχνική που χρησιμοποιήσαμε για την εξασφάλιση γρήγορης ανταπόκρισης του συστήματος σε πολύπλοκα ερωτήματα, είναι το **Eager loading**. Επειδή κάνουμε χρήση πολλών συσχετιζόμενων πινάκων σε αντίθεση με το lazy loading το eager loading αποστέλλει λιγότερα ερωτήματα στην βάση.

Χρησιμοποιούμε την Κλάση **Lang** του framework ώστε να πετύχουμε εύκολα την μετάφραση της σελίδας σε παραπάνω από μια γλώσσες.

```

return array(
  'welcome' => 'Χαίρόμαστε πάντα να μαθαίνουμε την άποψή σας και υπάρχουν πολλοί
               τρόποι να επικοινωνήσετε μαζί μας.',
  'comment' => 'Αφήστε μας σχόλιο',
  'error' => 'Κάτι πήγε στραβά. Δοκιμάστε ξανά',
  'contact' => 'Επικοινωνία',
  'adress' => 'Οδός Ερμή, Ελασσόνα 40200',
  'open' => 'Ωρες λειτουργίας',
  'open' => 'Ωράριο',
  'mondayfriday' => 'Δευτ.-Παρ.: ',
  'time' => '10:00 με 20:00',
  'saturday' => 'Σάββατο',
  'timesat' => ' 11:00-13:00',
  'thanks' => 'Ευχαριστούμε',
  'asap' => 'Θα επικοινωνίσουμε μαζί σας το συντομότερο δυνατό',
  'namehold' => 'Το όνομα σας',
  'emailhold' => 'Το email σας',
  'messagehold' => 'Το μήνυμά σας',
  'nameLabel' => 'Όνομα',
  'message' => 'Μήνυμα',
  'submit' => 'Υποβολή',
);

```

Εικόνα 2

Μέσω των μεθόδων lang::load και lang::get

Για να είναι πιο ευδιάκριτος ο κάθε χρήστης του συστήματος χρησιμοποιήσαμε την σελίδα του **Gravatar.com** μέσω του πακέτου <https://github.com/excalibur1976/fuelphp-gravatar> για το fuel

```
$options['image'] = urlencode('monsterid');  
$options['size'] = 872;  
$gravatar = \Gravatar::forge($this->current_user->email,$options);  
$view->gravatar = $gravatar;
```

Εικόνα 3

Πλεονασμός δεδομένων

Ο πλεονασμός δεδομένων είναι υπεύθυνος για ορισμένα προβλήματα που επηρεάζουν την βάση δεδομένων μας.

Πιο συγκεκριμένα δημιουργούν πρόβλημα:

- Στην ενημέρωση των δεδομένων
- Στην διαγραφή των δεδομένων
- Στην Εισαγωγή των δεδομένων

Κανονικοποίηση

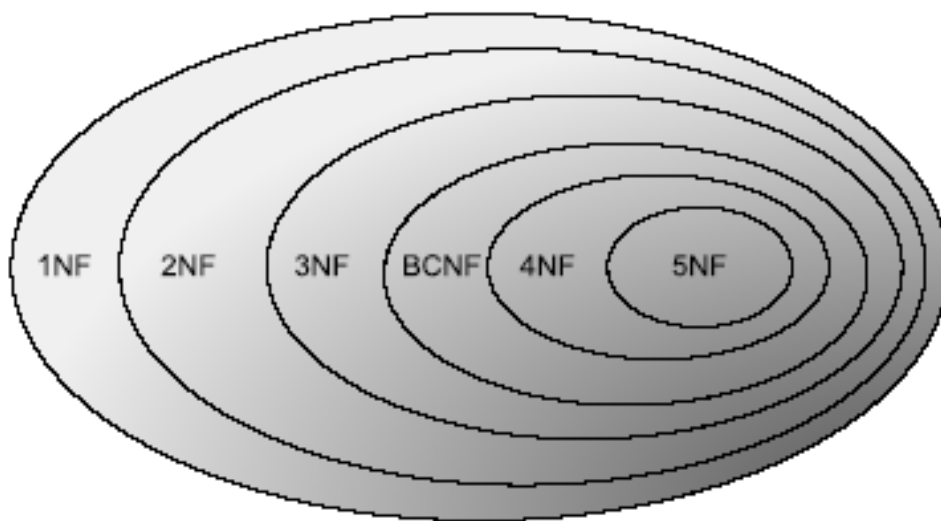
Ουσιαστικά με την κανονικοποίηση επιτυγχάνουμε την διάσπαση πολύπλοκων σχημάτων σε μικρότερα σχήματα πιο σωστά ορισμένα ώστε να αποφεύγονται τελείως οι συνθήκες πλεονασμού.

Με την κανονικοποίηση ουσιαστικά καταφέρνουμε να εξαλείψουμε τις ανωμαλίες που προκύπτουν κατά την εισαγωγή, διαγραφή και ενημέρωση στοιχείων στην βάση μας.

1^η Κανονική μορφή

Η πρώτη σχεσιακή μορφή είναι εξαιρετικά κρίσιμη για το σχεσιακό μας μοντέλο γιατί αποτελεί την αρχή της αποδόμησης όπου παρουσιάζονται όλες οι ενότητες και τα χαρακτηριστικά τους .

Σημαντικό επίσης είναι να αναφερθεί πως εάν ένας πίνακας βρίσκεται σε κανονική μορφή τότε βρίσκεται και σε όλες τις προηγούμενες μορφές.



Εικόνα 4

Μια σχέση βρίσκεται σε πρώτη κανονική μορφή όταν κάθε στήλη του πίνακα δεν παίρνει πολλαπλές ή σύνθετες τιμές και όταν κάθε κελί περιέχει μια μονάχα τιμή.

2^η Κανονική μορφή

Ένας πίνακας που βρίσκεται σε πρώτη κανονική μορφή βρίσκεται και σε δεύτερη κανονική μορφή όταν πληρούνται οι παρακάτω προδιαγραφές.

Το πρωτεύον κλειδί αποτελείται από ένα μόνο χαρακτηριστικό

Ο πίνακας δεν έχει χαρακτηριστικά που δεν ανήκουν στο κλειδί

Κάθε χαρακτηριστικό που δεν είναι κλειδί, είναι πλήρως συναρτησιακά εξαρτώμενο από το πρωτεύων κλειδί

3^η Κανονική μορφή

Ο πίνακας πρέπει να είναι στην δεύτερη κανονική μορφή και πρέπει επίσης όλα τα γνωρίσματα τα οποία δεν είναι κύρια να είναι άμεσα εξαρτώμενα από το πρωτεύων κλειδί.

Θα πρέπει όλοι οι πίνακες της βάσης μας να είναι στην τρίτη κανονική μορφή.

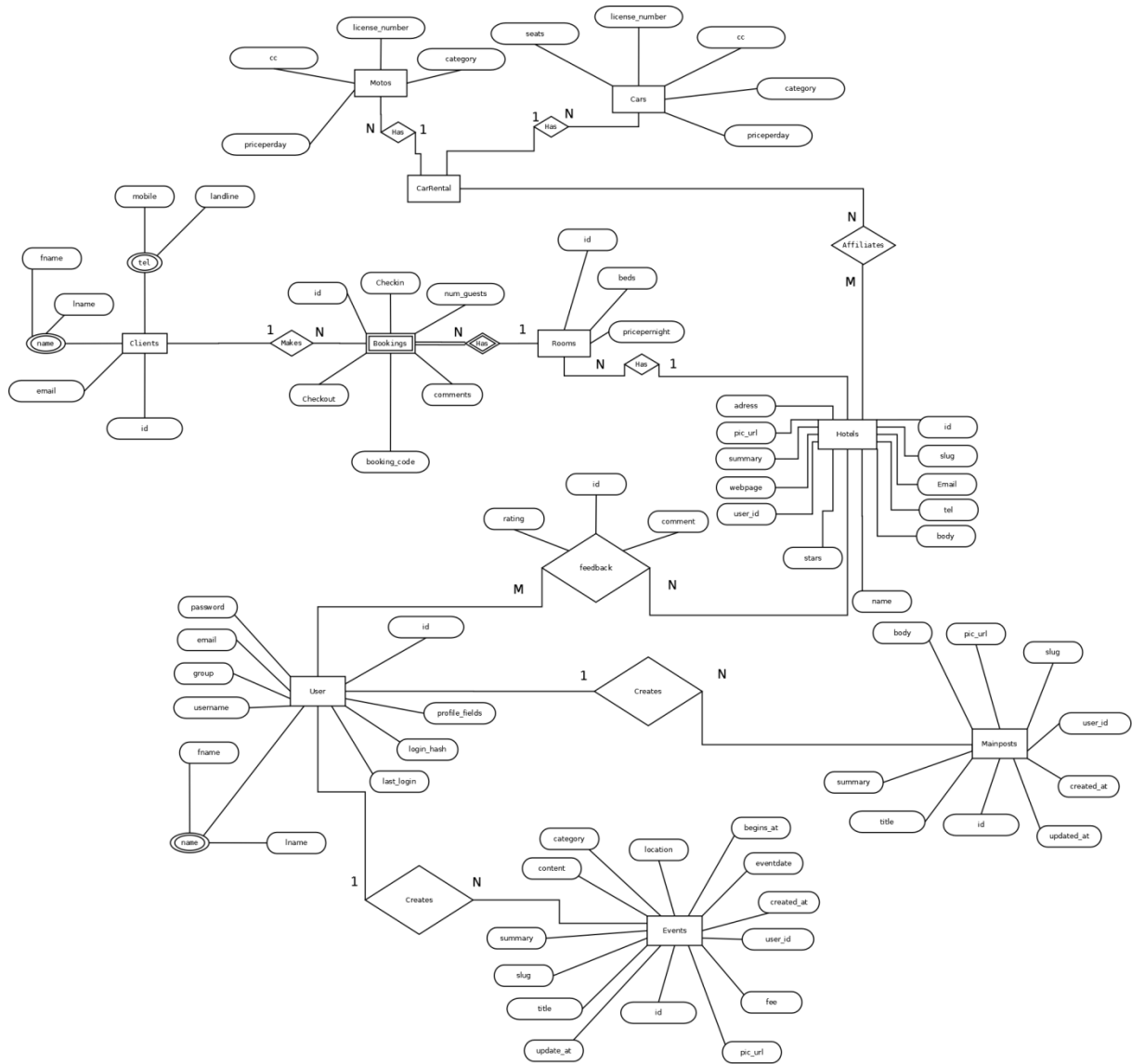
4.3 Υλοποίηση

4.3.1 Σχεδιασμός βάσης δεδομένων

Μετά από προσεκτικό σχεδιασμό της βάσης μας καταλήξαμε στις παρακάτω οντότητες

- users
- hotels
- events
- rentals
- rooms
- vehicles
- comments
- facts
- bookings

Διάγραμμα οντοτήτων συσχετίσεων



Εικόνα 5

ERM to Relational model

1. Hotels(id, name, address, email, tel, stars, webpage, summary, body, pic_url, slug)
- 2.
3. Rooms(hotel_name, id, beds, pricepernight)
- 4.
5. HotelComments(id ,user_id, hotel_id , rating, body, parent_id)
- 6.
7. Users(id, email, password, group, username, fname, lname, last_login, login_hash, profile_fields)
- 8.
9. Events(id, user_id, title, slug, summary, content, category, location, begins_at, eventdate, created_at,updated_at, user_id, fee, pic_url)
- 10.
11. Booking(id ,user_id, rooms_id, checkin, checkout, num_guests, comments, total)
12. Rentals(id, name, address, email, tel, webpage, summary, body, pic_url, slug)
- 13.
14. CarBookings(id ,user_id, rooms_id, checkin, checkout, num_guests, comments, total)
15. RentalComments(id ,user_id, hotel_id , rating, body, parent_id)
- 16.
17. Vehicle(id, brand, license, cc, category, seats, rental_id, price)

Σε όλους τους πίνακες της βάσης μας ως πρωτεύων κλειδί χρησιμοποιήθηκε το πεδίο id.

Περιορισμοί πληθικότητας

Παρακάτω παρατίθεται ο κώδικας που περιγράφει τους περιορισμούς πληθικότητας όπως ακριβώς τον γράψαμε στην κλάση του κάθε μοντέλου. Όπου δεν αναφέρονται ξένα κλειδιά μέσω σύμβασης του framework χρησιμοποιείται το όνομα του table και underscore id, π.χ.

Το user_id στον πίνακα carbookings είναι ξένο κλειδί που αναφέρεται στον πίνακα users.

```
protected static $_belongs_to = array('rooms','users');
```

Εικόνα 6

```
protected static $_belongs_to = array('vehicles','users');
```

Εικόνα 7

```
protected static $_belongs_to = array('user');
```

Εικόνα 8

```
protected static $_has_many = array('rooms' => array(  
    'model_to' => 'Model_Room',  
    'key_from' => 'id',  
    'key_to' => 'hotel_id',  
    'cascade_save' => true,  
    'cascade_delete' => true,  
),  
    'hotelcomments');
```

Εικόνα 9

```
protected static $_belongs_to = array('users','hotels');
```

Εικόνα 10

```
protected static $_has_many = array('vehicles','rentalcomments');
```

Εικόνα 11

```
protected static $_belongs_to = array('users','rentals');
```

Εικόνα 12

```
protected static $_belongs_to = array('hotels');  
protected static $_has_many = array('bookings');
```

```
protected static $_has_many = array('mainposts','events','hotelcomments','bookings','carbookings');
```

Εικόνα 13

```
protected static $_belongs_to = array('rentals');  
protected static $_has_many = array('carbookings');
```

Εικόνα 14

Ένα μη προφανές γνώρισμα είναι το `profile_fields` του πίνακα `users`.

Το πεδίο αυτό είναι ένα associative array το οποίο χρησιμοποιούμε για να αποθηκεύσουμε περισσότερες πληροφορίες για τον χρήστη όπως ονοματεπώνυμο, λογαριασμό social media κ.τ.λ.

Το μοντέλο μας βρίσκεται ήδη στην Τρίτη κανονική μορφή (3NF)

4.3.2 Εγκατάσταση του framework

Windows

Απλώς αντιγράφουμε το τον φάκελο του FuelPhp στο path του τοπικού server xampp που έχουμε στήσει και δημιουργούμε νέο project. Έπειτα τρέχουμε τις εντολές που θέλουμε στο σχετικό directory είτε με cmd είτε με κάποιο άλλο πρόγραμμα όπως το git bash στην περίπτωση μας

Nix system Installation (Linux, OS X, Unix, etc)

Για να εξοικονομήσουμε χρόνο κατεβάζουμε μια βασική έκδοση της oil της βιβλιοθήκης curl για να μπορούμε να γράφουμε στο project μας oil αντί για php oil

```
$ curl get.fuelphp.com/oil | sh
```

Εικόνα 15

```
$ curl get.fuelphp.com/oil | sh
```

Επόμενο βήμα η δημιουργία ενός φακέλου στο directory που βρισκόμαστε

```
$ oil create <project_name>
```

Εικόνα 16

Note: Αυτό θα τρέξει επίσης παρέχοντας δυνατότητα εγγραφής στα απαραίτητα directories

```
$ oil refine install
```

 και

```
$ php composer.phar update
```

 για να τραβήξει τα composer dependencies

<http://fuelphp.com/docs/installation/instructions.html#command>

File Structure

Στο root του application μας θα πρέπει να υπάρχουν 3 βασικά αντικείμενα

- **Fuel/** - Εκεί όπου θα υπάρχει όλος ο κώδικας της php μας

Μέσα στον φάκελο του **fuel** υπάρχουν μερικοί άλλοι σημαντικοί φάκελοι.

- **app/** - όλος ο κώδικας php του application μας πάει εδώ συμπεριλαμβάνοντας τα module, views, controllers.
- **core/** - εκεί που ζει το fuel.
- **packages /** - το fuel ξεχωρίζει συγκεκριμένες λογικές σε διαφορετικά πακέτα για να κρατάει το core ανέπαφο. Από default το fuel περιέχει 3 packages, το oil, το orm και το auth τα οποία πρέπει να ενεργοποιηθούν, το πώς γίνεται αυτό θα το δούμε παρακάτω.
- **Public/** - Ότι θέλουμε να είναι άμεσα προσβάσιμο από τον browser π.χ. JS, CSS, images κ.τ.λ.
- **Oil/** - Χρησιμοποιείτε κυρίως για debugging και η χρήση του είναι προαιρετική

Στο **app/** folder υπάρχουν μερικά πολύ βασικά αρχεία

- **Config/** - Αρχεία configuration για αρκετές κλάσεις και το γενικό config.php βρίσκονται εδώ.
- **Classes/** - Εδώ βρίσκονται όλα τα controllers, models, helper classes, business logic, libraries κ.τ.λ. Αν χρειαστεί να γράψουμε μια κλάση για να χρησιμοποιήσουμε στον κώδικα μας θα γραφτεί σε αυτόν εδώ τον φάκελο.
- **Classes/controller/** - Εδώ τοποθετούνται οι controllers
- **Classes/model/** - Εδώ τοποθετούνται τα models
- **Views/** - Εδώ τοποθετούμε τα αρχεία view η επειδή δεν υπάρχουν συμβάσεις ονομασίας μπορούμε απλά να τα τοποθετήσουμε στο root.

Configuration

Database Configuration

Στις παρακάτω γραμμές του κώδικα μας βλέπουμε τις αρχικές ρυθμίσεις στο directory

fuel/app/config/development/db.php

```
1 <?php
2 /**
3  * The development database settings. These get merged with the global settings.
4  */
5
6  return array(
7      'default' => array(
8          'connection' => array(
9              'dsn'      => 'mysql:host=localhost;dbname=fuel_dev',
10             'username' => 'root',
11             'password' => 'root',
12         ),
13     ),
14 );
```

Εμείς πρέπει απλά να συσχετίσουμε την βάση δεδομένων που φτιάξαμε και να αλλάξουμε τον κωδικό

```
1 <?php
2 /**
3  * The development database settings. These get merged with the global settings.
4  */
5
6  return array(
7      'default' => array(
8          'connection' => array(
9              'dsn'      => 'mysql:host=localhost;dbname=Ptyxiaki_db',
10             'username' => 'root',
11             'password' => '3KPyHRJCQBt4yQFF',
12         ),
13     ),
14 );
```

Enabling Auth & Orm Packages

Λίγα λόγια για τα Orm & Auth πακέτα

Το **ORM** σημαίνει Object Relational Mapper και κάνει δύο βασικά πράγματα

- Θεωρεί τις σειρές των πινάκων της βάσης μας σαν αντικείμενα και μας επιτρέπει να δημιουργήσουμε σχέσεις ανάμεσα τους
- Ακολουθεί πιστά το Active Record Pattern (αποθήκευση δεδομένων σε relational databases – κρατώντας πληροφορίες για τα δεδομένα αλλά και για τις συσχετίσεις τους)

Το πακέτο **Auth**

- παρέχει ένα τυποποιημένο interface πιστοποίησης για το fuelphp. Αυτό επιτρέπει σε όσους το χρησιμοποιούν να γράψουν τους δικούς τους drivers και να τους ενσωματώσουν για να λειτουργούν με τον παλιό τους κώδικα κρατώντας τις βασικές μεθόδους σταθερές.

Ενεργοποίηση των πακέτων συμπεριλαμβανομένου και του πακέτου email

```
261 'always_load' => array(
262
263     /**
264      * These packages are loaded on Fuel's startup.
265      * You can specify them in the following manner:
266      *
267      * array('auth'); // This will assume the packages are in PKGPATH
268      *
269      * // Use this format to specify the path to the package explicitly
270      * array(
271      *     array('auth' => PKGPATH.'auth/')
272      * );
273      */
274     'packages' => array(
275         'orm',
276         'auth',
277         'email',
278     ),
```


Development

Χρησιμοποιώντας λειτουργικό Windows εκτελούμε τα επόμενα βήματα όπως ακριβώς περιγράφονται.

- Ανοίγουμε το gitbash [console app] και πηγαίνουμε στον φάκελο στον οποίο θέλουμε να δημιουργήσουμε το project μας, στην περίπτωση μας το path είναι `cd /c/xampp/htdocs`
- Χρησιμοποιούμε την εντολή `$ oil create Ptyxiaki` για να δημιουργήσουμε ένα νέο project με το όνομα Ptyxiaki

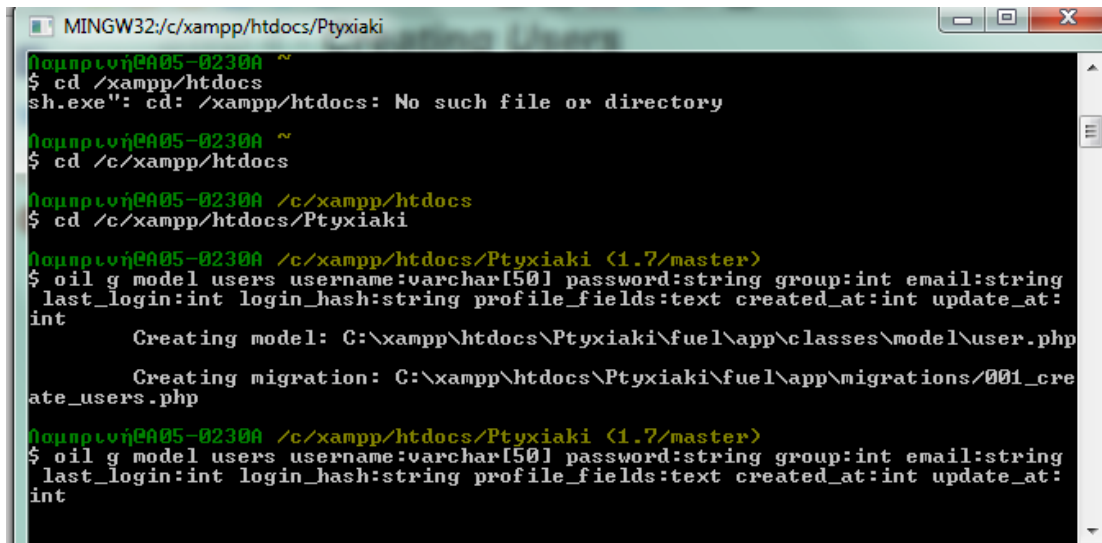
Αφού δημιουργήσαμε πλέον το project μας θα συνεχίσουμε φτιάχνοντας την κλάση **Create_users**

Απλά τρέχουμε την εντολή

```
oil g model users username:varchar[50] password:string group:int email:string last_login:int login_hash:string profile_fields:text created_at:int updated_at:int
```

Στην παρακάτω εικόνα βλέπουμε τα αρχεία που δημιουργήθηκαν με την εντολή που γράψαμε

- Δημιουργήθηκε ένα model στον φάκελο `fuel/app/classes/model/user.php`
- Δημιουργήθηκε ένα migration στον φάκελο `fuel/app/migrations/001_create_users.php`



```
MINGW32/c:/xampp/htdocs/Ptyxiaki
Θαμπριλή@A05-0230A ~
$ cd /xampp/htdocs
sh.exe": cd: /xampp/htdocs: No such file or directory
Θαμπριλή@A05-0230A ~
$ cd /c/xampp/htdocs
Θαμπριλή@A05-0230A /c/xampp/htdocs
$ cd /c/xampp/htdocs/Ptyxiaki
Θαμπριλή@A05-0230A /c/xampp/htdocs/Ptyxiaki (1.7/master)
$ oil g model users username:varchar[50] password:string group:int email:string last_login:int login_hash:string profile_fields:text created_at:int update_at:int
Creating model: C:\xampp\htdocs\Ptyxiaki\fuel\app\classes\model\user.php
Creating migration: C:\xampp\htdocs\Ptyxiaki\fuel\app\migrations\001_create_users.php
Θαμπριλή@A05-0230A /c/xampp/htdocs/Ptyxiaki (1.7/master)
$ oil g model users username:varchar[50] password:string group:int email:string last_login:int login_hash:string profile_fields:text created_at:int update_at:int
```

Εικόνα 17

Για να ενημερώσουμε το σύστημα μας και να μπορούμε να δούμε τις αλλαγές που κάναμε στο γράφουμε την εντολή `$oil r migrate`

Αμέσως μετά θα δημιουργήσουμε έναν administrator user γράφοντας τις εντολές

```
$oil console
$Auth::create_user('admin', '123456', 'mail@mail.com', 100);
```

name, password, email, group_id

το response 1 που παίρνουμε υποδεικνύει ότι το user_id του admin έχει την τιμή 1

```
MINGW32:/c/xampp/htdocs/Ptyxiaki
Parsing Error - syntax error, unexpected ';', expecting ')' in APPPATH/config/co
nfig.php on line 315
hampry@0A05-0230A /c/xampp/htdocs/Ptyxiaki (1.7/master)
$ oil console
Parse error: syntax error, unexpected ';', expecting ')' in C:\xampp\htdocs\Ptyx
iaki\fuel\app\config\config.php on line 316
Parsing Error - syntax error, unexpected ';', expecting ')' in APPPATH/config/co
nfig.php on line 316
hampry@0A05-0230A /c/xampp/htdocs/Ptyxiaki (1.7/master)
$ oil console
Parse error: syntax error, unexpected end of file, expecting ')' in C:\xampp\htd
ocs\Ptyxiaki\fuel\app\config\config.php on line 317
Parsing Error - syntax error, unexpected end of file, expecting ')' in APPPATH/c
onfig/config.php on line 317
hampry@0A05-0230A /c/xampp/htdocs/Ptyxiaki (1.7/master)
$ oil console
Fuel 1.7.2 - PHP 5.5.11 (cli) (Apr  8 2014 15:01:59) [WINNT]
>>> Auth::create_user('admin','123456','mail@mail.com',100);
1
>>>
```

Εικόνα 18

Κάνοντας τα παραπάνω έχουμε στήσει ήδη την βάση της εφαρμογής μας. Συνεχίζουμε χρησιμοποιώντας την εντολή

```
oil generate admin table_name fields:type --skip
```

να φτιάχνουμε με scaffolding admin controllers models και βασικά views για Crud (create read update delete) λειτουργίες για όλες της βασικές οντότητες της βάσης μας.

Πιο συγκεκριμένα για τους πίνακες Hotels, Rentals, Bookings, CarBookings, Hotelcomments, Carcomments, Rooms, Vehicles και Events.

Administration Page / Backend

Website Preview

Αρχίζοντας με τα βασικά features του διαχειριστικού μας συστήματος στην πάνω αριστερά γωνία, στο 2^ο tab θα βρούμε ένα preview button το οποίο όταν επιλεγθεί μας μεταφέρει στο front end της ιστοσελίδας μας.

Admin logout

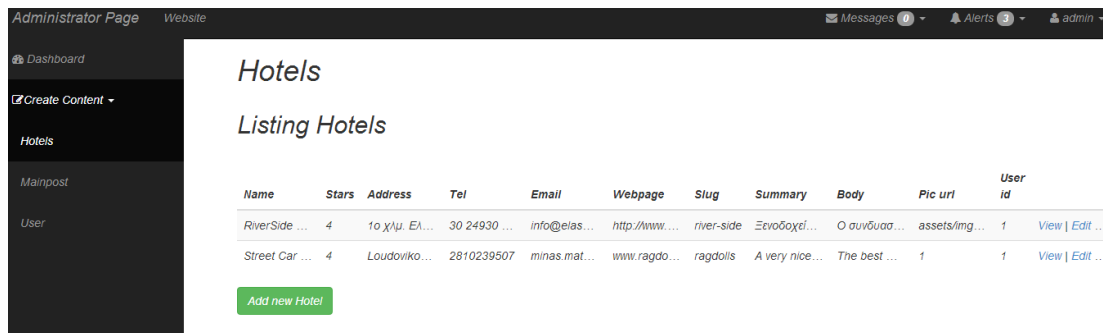
Στην πάνω δεξιά πλευρά της ιστοσελίδας μας υπάρχει ένα drop-down menu με το όνομα admin, όταν πατήσουμε πάνω σε αυτό εμφανίζονται διάφορες επιλογές. Για να αποσυνδεθούμε από την ιστοσελίδα πατάμε logout

Στην Αριστερή πλευρά στην σελίδα του Διαχειριστή έχουμε ένα Dropdown menu με το όνομα **Create content** όταν το επιλέξουμε μας εμφανίζεται η λίστα με τα περιεχόμενα του, πρόκειται για περιεχόμενα τα οποία μπορούν να δημιουργηθούν ή να μεταβληθούν δυναμικά

DROP-DOWN MENU

1)Hotels

Στην παρακάτω διαφάνεια βλέπουμε μια εικόνα από το tab **Hotels**



The screenshot shows the Admin Panel interface. On the left is a sidebar with a dark theme containing 'Dashboard', 'Create Content', 'Hotels', 'Mainpost', and 'User'. The main content area is titled 'Hotels' and 'Listing Hotels'. It features a table with columns: Name, Stars, Address, Tel, Email, Webpage, Slug, Summary, Body, Pic url, and User id. Two hotel entries are visible: 'RiverSide ...' and 'Street Car ...'. Below the table is a green 'Add new Hotel' button.

Name	Stars	Address	Tel	Email	Webpage	Slug	Summary	Body	Pic url	User id
RiverSide ...	4	1ο χ/μ. Ελ...	30 24930 ...	info@elas...	http://www...	river-side	Ξενοδοχί...	Ο συνδυασ...	assets/img...	1
Street Car ...	4	Loudoviko...	2810239507	minas.mat...	www.ragdo...	ragdolls	A very nice...	The best ...	1	1

Εικόνα 19

Όπως φαίνεται στην δεξιά πλευρά των ήδη υπαρχόντων ξενοδοχείων (αντικειμένων) υπάρχει η δυνατότητα παραμετροποίησης τους, όπως παραδείγματος χάρη η προεπισκόπηση τους η μεταβολή των περιεχομένων τους η ακόμα και η διαγραφή τους.

Ακόμη όπως παρατηρούμε υπάρχει ένα πράσινο κουμπί **Add new hotel**.

Πατώντας αυτό το κουμπί δημιουργούμε ένα νέο item (hotel) το οποίο προκειμένου να δημιουργηθεί θα πρέπει να του δώσουμε τις κατάλληλες παραμέτρους και στον σωστό τύπο, π.χ.

Hotels

New Hotel

Name
Name

Stars
Stars

Address
Address

Tel
Tel

Email
Email

Webpage
Webpage

Εικόνα 20

Tel: integer
 Address: String
 κ.ο.κ.

Όταν τελειώσουμε με την παραμετροποίηση του αντικειμένου πατάμε το κουμπί save, ενώ για να πάμε ένα βήμα πίσω πατάμε το κουμπί **back**.

Pic url
1

User id
1

Save

Back

Εικόνα 21

Εφόσον τώρα σώσαμε το αντικείμενο που δημιουργήσαμε και του δώσαμε τις κατάλληλες τιμές σε όλα του τα γνωρίσματα μπορούμε πλέον να το δούμε και να το συμπεριλάβουμε στην κατηγορία των ξενοδοχείων.

Hotels

Listing Hotels

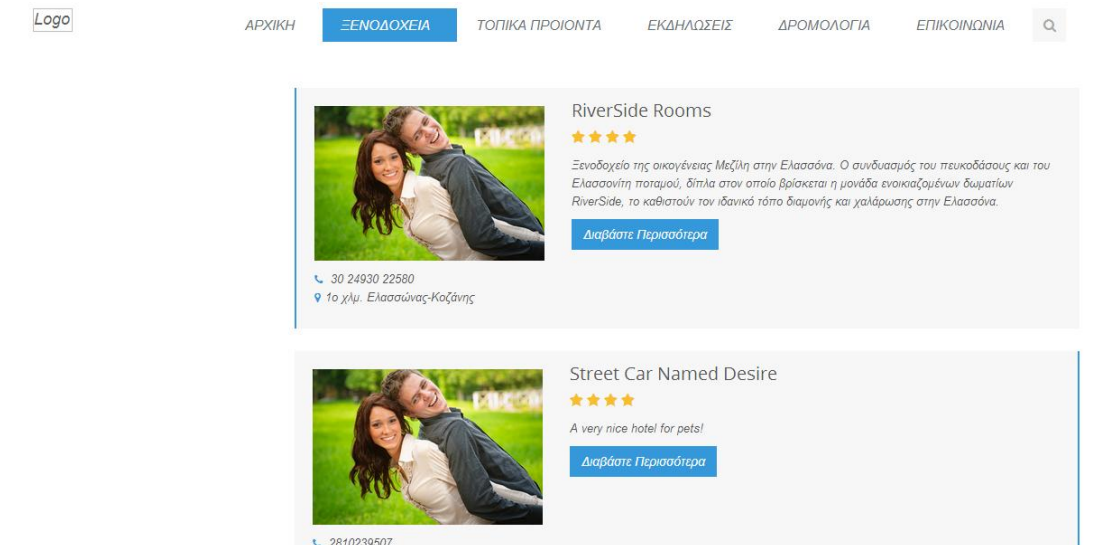
Name	Stars	Address	Tel	Email	Webpage	Slug	Summary	Body	Pic url	User id
RiverSide ...	4	1ο χιλμ. Ελ...	30 24930 ...	info@elias...	http://www...	river-side	Ξενοδοχεί...	Ο συνδυασ...	assets/mg...	1
Street Car ...	4	Loudoviko...	2810239507	minas.mat...	www.ragdo...	ragdolls	A very nice...	The best ...	1	1
Megaro	5	Bofor 8	2810239123	info@meg...	www.mega...	megaro	Its a nice h...	VEry very ...	1	1

Add new Hotel

Εικόνα 22

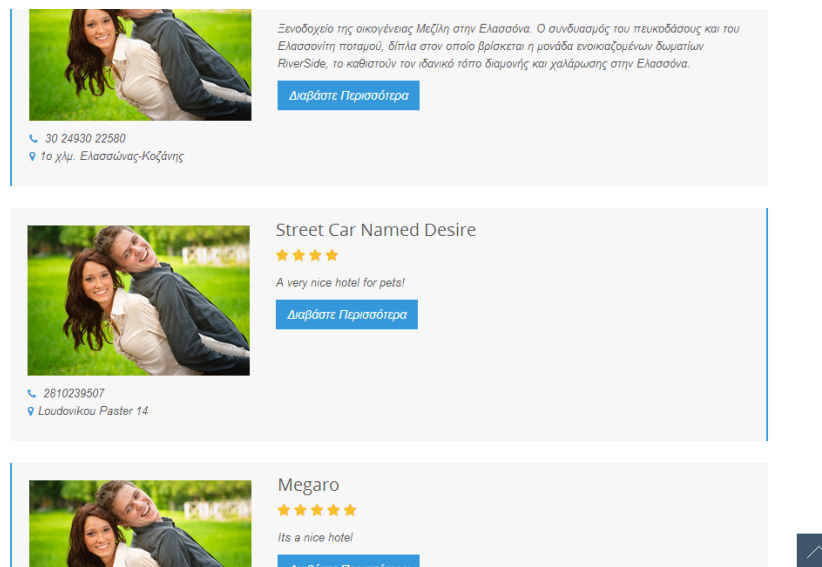
Για να σιγουρευτούμε τώρα για το πώς φαίνεται η ιστοσελίδα μας πηγαίνουμε στο Front end (tab ξενοδοχεία) και βλέπουμε το αντικείμενο που φτιάξαμε.

(Πριν)



Εικόνα 23

(Μετά)



Εικόνα 24

Όπως βλέπουμε το αντικείμενο μας δημιουργήθηκε και τώρα δεν μας μένει παρά να το κοιτάξουμε για μια ακόμη φορά και να βεβαιωθούμε ότι όλα του τα στοιχεία παρουσιάζονται σωστά.

Αφού διαβάσουμε τις πληροφορίες που φαίνονται κάτω από το όνομα του ξενοδοχείου στο Front end πατάμε το μπλε κουμπί διαβάστε περισσότερα για να δούμε εάν και τα υπόλοιπα **attributes** συμπληρώθηκαν σωστά

Rooms

Εφόσον οι ανάγκες ενός ξενοδοχείου αναβαθμίζονται συνέχεια και για περαιτέρω λόγους που θα δούμε παρακάτω κρίναμε σκόπιμο να δημιουργήσουμε ένα ακόμη **feature** το **Rooms**. Πρόκειται ουσιαστικά για ένα feature που είναι υπεύθυνο για την δημιουργία ενός δωματίου με συγκεκριμένα χαρακτηριστικά όπως φαίνεται στην εικόνα παρακάτω

Hotels

Listing Hotels

Name	Stars	Summary	Created At	
RiverSide Rooms	4	Ξενοδοχείο της οικογένειας ...	05/01/14,13:34:46	Rooms View Edit Delete
Street Car Named Desire	4	A very nice hotel for pets!	05/16/14,17:04:32	Rooms View Edit Delete

[Add new Hotel](#)

Εικόνα 25

Κάνουμε κλικ στο Rooms που βρίσκεται στην κάτω γωνία δεξιά και οδηγούμαστε στο επόμενο μενού

Hotel: RiverSide Rooms

Listing Rooms

id	beds	Hotel id	Created At	
11	3	1	05/30/14,12:41:46	View Edit Delete
12	5	1	05/30/14,23:26:19	View Edit Delete

[Add new Room](#)

[Back](#)

Εικόνα 26

Εδώ βλέπουμε διάφορες πληροφορίες για τα ήδη υπάρχοντα δωμάτια όπως τον αριθμό id τους το νούμερο των κρεβατιών που υπάρχουν στο συγκεκριμένο δωμάτιο, το Hotel_id που μας υποδεικνύει σε πιο ξενοδοχεία ανήκει το κάθε δωμάτιο. Τέλος υπάρχει ένα crud μενού (create, update, edit & delete).

Εάν θέλουμε να δημιουργήσουμε ένα νέο δωμάτιο πατάμε στο κουμπί Add new Room και επεξεργαζόμαστε με προσοχή τα πεδία τιμών που έχουμε δημιουργήσει.

Εφόσον είμαστε χαρούμενοι με τις επιλογές μας πατάμε στο κουμπί **save** για να σωθούν οι επιλογές μας και να δούμε πως φαίνεται το νέο αντικείμενο που δημιουργήσαμε

New Room

Roomtype

Beds

Price Per Night

Save

[Back](#)

Εικόνα 27

Έχοντας σώσει το νέο μας αντικείμενο μπορούμε να επαληθεύσουμε ότι έχει δημιουργηθεί βλέποντας την λίστα των δωματίων του εν λόγω ξενοδοχείου

Hotel: RiverSide Rooms

Listing Rooms

id	beds	Hotel id	Created At	
11	3	1	05/30/14,12:41:46	View Edit Delete
12	5	1	05/30/14,23:26:19	View Edit Delete
13	2	1	07/16/14,17:23:32	View Edit Delete

[Add new Room](#)

[Back](#)

Εικόνα 28

2)Mainposts

Μια άλλη επιλογή του Drop-down menu είναι το **mainposts**

Διατηρώντας την λογική της δημιουργίας item της κατηγορίας hotel μας δίδεται η δυνατότητα να δημιουργήσουμε item(s) της κατηγορίας Mainposts.

Mainposts

Listing Mainposts

Id ↕	Title ↕	Slug ↕	Summary ↕	Body ↕	Pic url ↕	Author ↕	
2	Ipsum	ipsum	"Lorem ipsum ...	"Lorem ipsum ...	assets/img/pic.jpg	admin	View Edit Del...
3	Ipsum Lorem	ipsum-lorem	The field Slug i...	The field Slug i...	assets/img/pic3...	admin	View Edit Del...
4	The field Sum...	th-field-summary	The field Slug i...	The field Slug i...	assets/img/pic3...	admin	View Edit Del...

Add new Mainpost

Εικόνα 29

Κάνοντας κλικ στο **Add new Mainpost** μας εμφανίζεται το ακόλουθο μενού

Mainposts

New Mainpost

Title

Slug
Summary

Εικόνα 30

Αφού συμπληρώσουμε όλα τα απαιτούμενα πεδία αποθηκεύουμε τις επιλογές μας.

Το μοντέλο αυτό υποστηρίζει το μοντέλο διαχείρισης Crud.

3)Event

Η Τρίτη επιλογή από το drop-down menu (Manage Content) είναι το **event**. Διατηρώντας ανέπαφη την ως τώρα λογική μας μπορούμε να δημιουργήσουμε items της κατηγορίας event και να τα αποθηκεύσουμε με σκοπό την προβολή τους στο front end της ιστοσελίδας μας.

Events

Listing Events

Id	Title	Summary	Category	When	Created At	
1	Open αγώνας ορει...	Με ιδιαίτερη ικανοπ...	sports	2014-10-09	05/17/14, 15:51:38	View Edit Delete
2	Olympus Marathon	Πρόκειται για έναν ...	sports	2014-05-20	05/17/14, 15:54:16	View Edit Delete
3	Luffy vs Warden	GOMU GOMU NO...	Entertainment	2014-05-18	05/17/14, 22:32:58	View Edit Delete

[Add new Event](#)

Εικόνα 31

Συμπληρώνοντας τα πεδία του event ενημερώνουμε το κοινό με σημαντικές πληροφορίες όπως π.χ. την ημερομηνία έναρξης της συγκεκριμένης εκδήλωσης τα participation fees και διάφορες άλλες πληροφορίες σημαντικές για το περιεχόμενο της εκδήλωσης και την κατηγορία στην οποία αυτό ανήκει.

Το event υποστηρίζει επίσης το μοντέλο διαχείρισης Crud

4)Users

Η τέταρτη επιλογή από το drop-down menu (Manage Content) είναι το **users** που είναι preview only category.

User

Listing Users

Id	username	group	email	Last Login	Created At	
1	admin	100	mail@mail.com	1405518711	1398797286	View
2	mod	50	mail1@mail.com	1398932381	1398880600	View
3	user	1	mail2@mail.com	0	1398880624	View

Εικόνα 32

5)Facts

Τα Facts είναι και αυτά μια ξεχωριστή κατηγορία που έχουμε προσθέσει στην ιστοσελίδα μας και εμφανίζονται σε μορφή carousel στην αρχική μας σελίδα.

Η χρηστική τους αξία περιορίζεται στην ωραιοποίηση του homepage και στην παροχή χαριτωμένων πληροφοριών.

Όπως και στις προαναφερθείσες κατηγορίες υποστηρίζεται το μοντέλο crud ενώ οι λειτουργίες προσθήκης νέων fact γίνεται με τον ίδιο κλασικό και απaráλλακτο ως τώρα τρόπο.

WYSIWYG

Τέλος άξιο αναφοράς είναι το wysiwyg(What You See Is What You Get) editor που προσθέσαμε για την σωστή μορφοποίηση και ωραιοποίηση των κειμένων μας.

Υποστηρίζει πληθώρα επιλογών όπως την δημιουργία πινάκων δημιουργία link και γενικές ρυθμίσεις μορφοποίησης του κειμένου μας.

User interface / Frontend

Layout

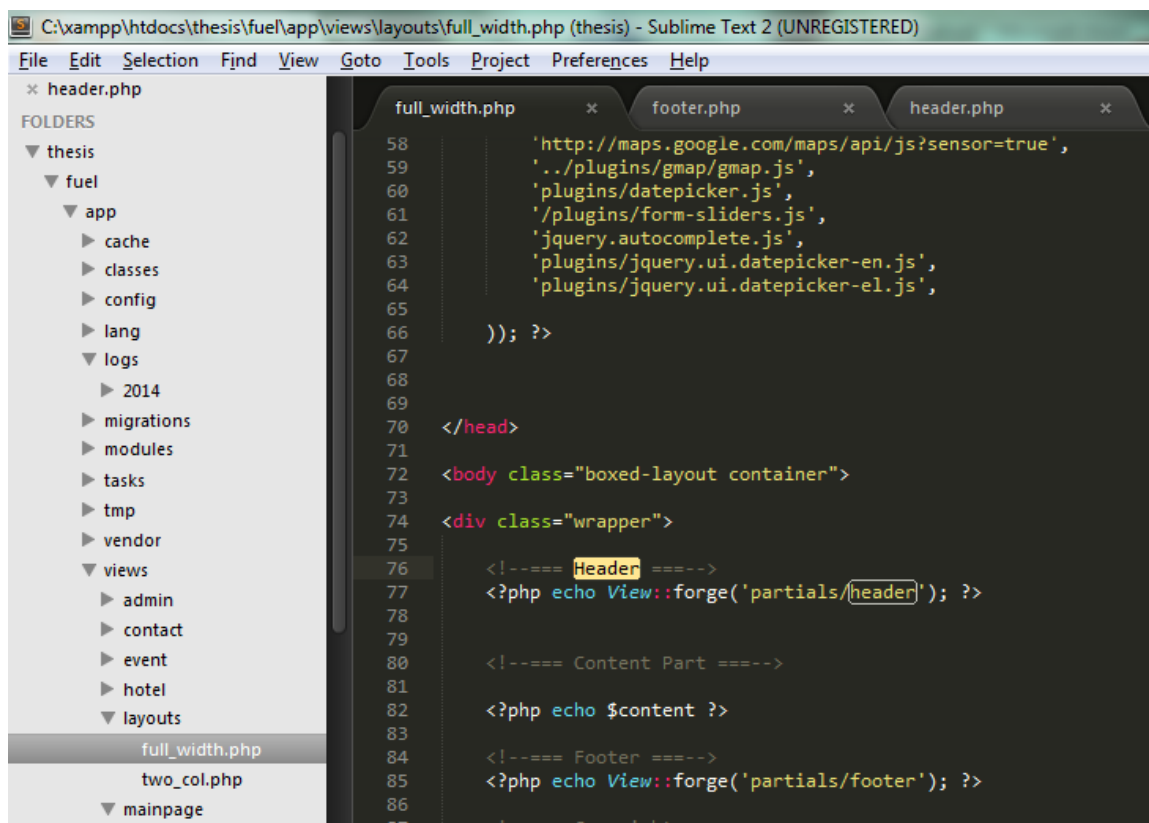
Με το Fuelphp μας δίνεται η δυνατότητα να δημιουργήσουμε layouts, κομμάτια κώδικα τα οποία χρησιμοποιούν οι controllers με κλήση rhp για την παρουσίαση των δεδομένων της σελίδας μας.

Ας πάρουμε για παράδειγμα τον controller hotel

```
1.
2. class Controller_Hotel extends Controller_Base
3. {
   public $template = 'layouts/
```

παρατηρούμε πως καλούμε το layout/ full_width το οποίο βρίσκεται στον φάκελο layouts του fuelphp directory μας.

Πηγαίνοντας στο αρχείο full_width βλέπουμε πως καλούμε εσωτερικά το header και το footer που περιέχουν στοιχεία όπως την αλλαγή γλώσσας, login κ.α. και θα εξηγήσουμε αναλυτικά παρακάτω.



```
C:\xampp\htdocs\thesis\fuel\app\views\layouts\full_width.php (thesis) - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
x header.php
FOLDERS
thesis
fuel
app
  cache
  classes
  config
  lang
  logs
  2014
  migrations
  modules
  tasks
  tmp
  vendor
  views
    admin
    contact
    event
    hotel
    layouts
      full_width.php
      two_col.php
    mainpage
  full_width.php
  footer.php
  header.php
58 'http://maps.google.com/maps/api/js?sensor=true',
59 '../plugins/gmap/gmap.js',
60 'plugins/datepicker.js',
61 '/plugins/form-sliders.js',
62 'jquery.autocomplete.js',
63 'plugins/jquery.ui.datepicker-en.js',
64 'plugins/jquery.ui.datepicker-el.js',
65
66 )); ?>
67
68
69
70 </head>
71
72 <body class="boxed-layout container">
73
74 <div class="wrapper">
75
76 <!--==== Header ---->
77 <?php echo View::forge('partials/header'); ?>
78
79
80 <!--==== Content Part ---->
81
82 <?php echo $content ?>
83
84 <!--==== Footer ---->
85 <?php echo View::forge('partials/footer'); ?>
86
87 <!--==== Copyright ---->
```

Εικόνα 33

Το header και το footer της σελίδας μας βρίσκονται στα ομώνυμα files στον φάκελο partials.

Header

Multilingual navigation menu

Όπως παρατηρούμε στην πάνω δεξιά γωνία της ιστοσελίδας μας υπάρχει η επιλογή «Γλώσσα», κάνοντας κλικ μπορούμε να αλλάξουμε την γλώσσα προβολής των αντικειμένων μας από ελληνικά σε Αγγλικά, ενώ η επιλογή αυτή μπορεί να επεκταθεί και σε περαιτέρω γλώσσες για να κρατήσουμε το σύστημα μας απλό και καθαρό αποφασίσαμε να χρησιμοποιήσουμε μόνο αυτές τις δύο.

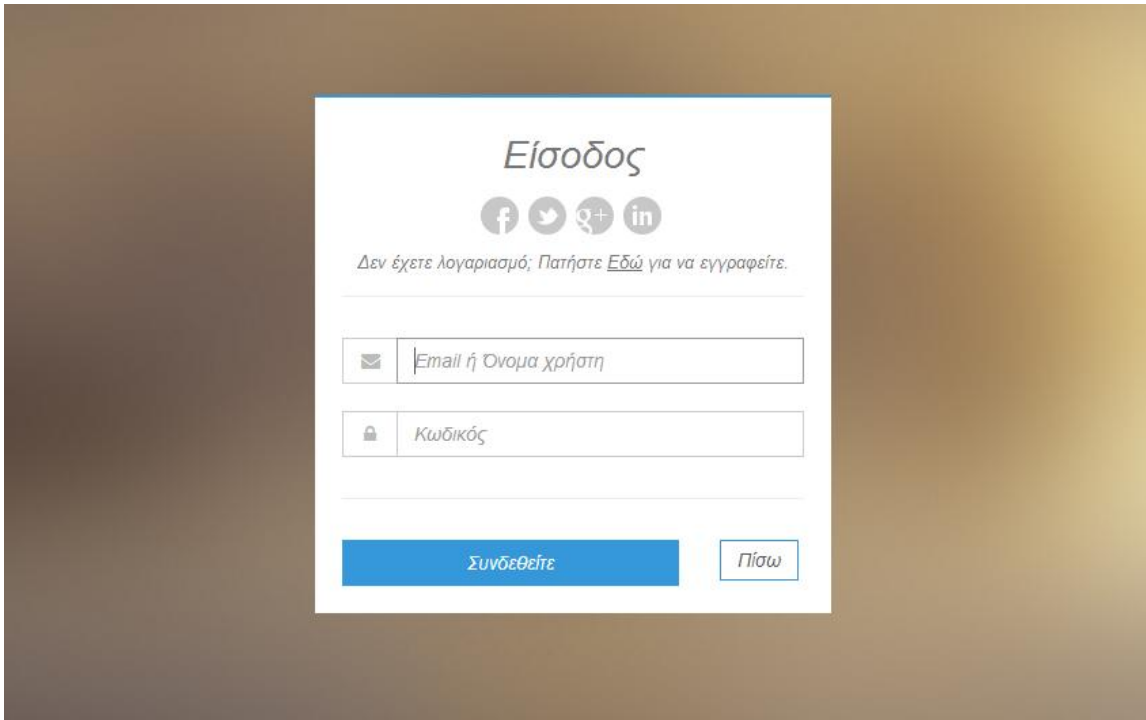
Κάνοντας λοιπόν την αλλαγή γλώσσας παρατηρούμε ότι αλλάζει το σύστημα πλοήγησης και όλες οι επιλογές είναι πλέον στα αγγλικά, για να επιστρέψουμε στην προβολή στα ελληνικά αρκεί να επιλέξουμε το languages (πλεον) και έπειτα το «ελληνικά» .

Quick contact

Στην συνέχεια όπως παρατηρούμε στην πάνω δεξιά γωνία δίπλα στις γλώσσες υπάρχει ένας φάκελος(Glyphicon) ο οποίος όταν επιλεγεί μας παραπέμπει στο μενού επικοινωνίας το οποίο θα αναλύσουμε παρακάτω τι ακριβώς κάνει.

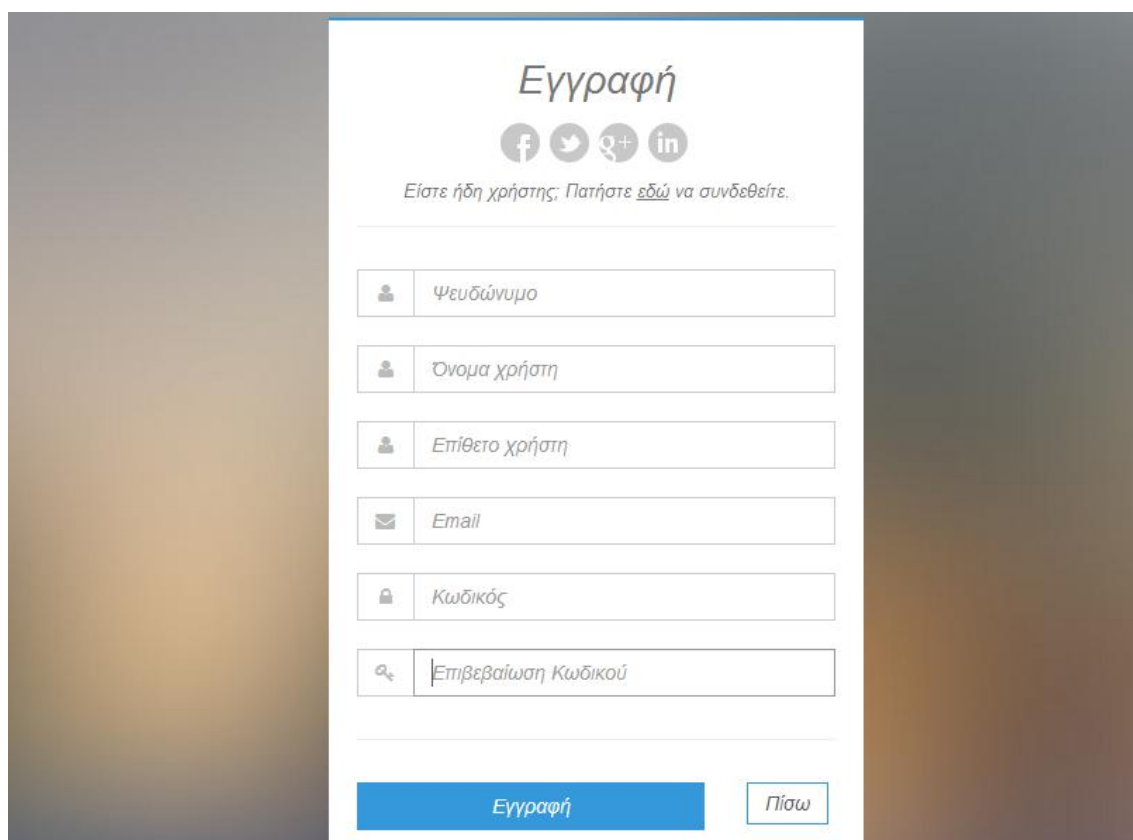
Login

Συνεχίζοντας προς τα δεξιά παρατηρούμε ένα ακόμα feature τις ιστοσελίδας μας που είναι το login, κάνοντας κλικ σε αυτό μπορούμε να συνδεθούμε χρησιμοποιώντας τον username και το password μας.



Εικόνα 34

Σε περίπτωση που δεν είμαστε users στο site μπορούμε να δημιουργήσουμε ένα νέο κωδικό χρήστη συμπληρώνοντας πάντα τις απαραίτητες πληροφορίες που φαίνονται στην παρακάτω εικόνα.



Εγγραφή

f t g+ in

Είστε ήδη χρήστης; Πατήστε [εδώ](#) να συνδεθείτε.

Ψευδώνυμο

Όνομα χρήστη

Επίθετο χρήστη

Email

Κωδικός

Επιβεβαίωση Κωδικού

Εγγραφή Πίσω

Εικόνα 35

Ο διαχειριστής της ιστοσελίδας με το username admin μπορεί να συνδεθεί με το

Username: admin

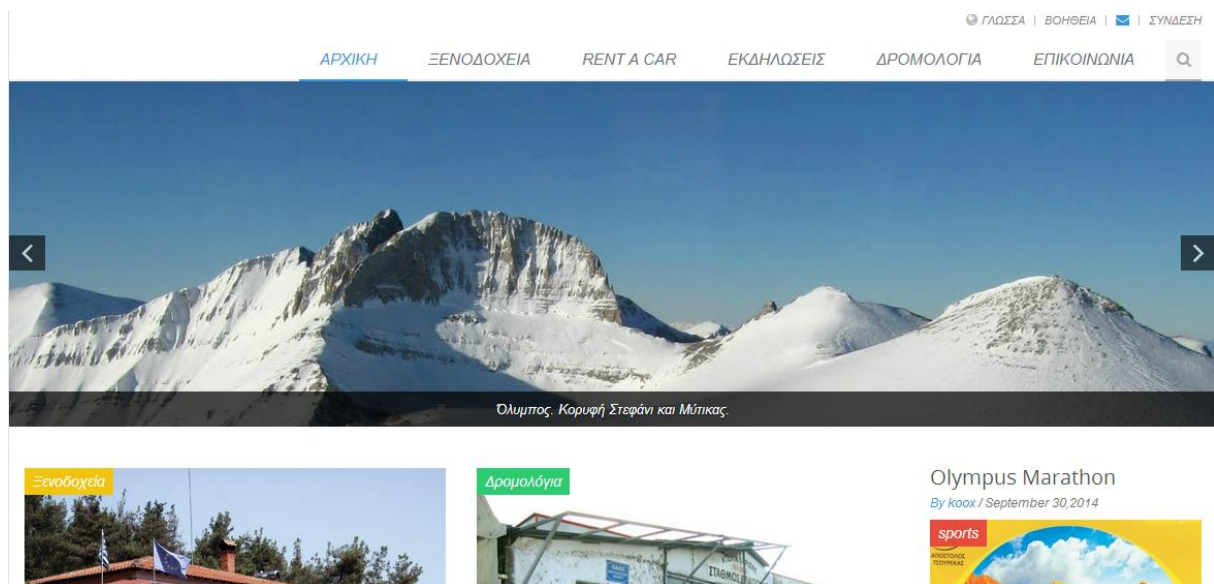
Password: 123456

Footer

Στο footer μας στην κάτω αριστερά πλευρά βλέπουμε πως έχουμε τοποθετήσει social icons τα οποία μας παραπέμπουν σε μέσα μαζικής δικτύωσης για ευνόητους λόγους οι σελίδες αυτές δεν έχουν δημιουργηθεί.

Mainpage-Αρχική

Ξεκινάμε την περιγραφή μας για την αρχική μας σελίδα (homepage) στην οποία υπάρχουν αρκετά features που πρέπει να εξηγηθούν.



Εικόνα 36

Slider

Όπως παρατηρούμε καθώς επισκεπτόμαστε τον τουριστικό οδηγό της Ελασσόνας στην αρχική σελίδα υπάρχει ένα επιβλητικό slider το οποίο εναλλάσσει διάφορες εικόνες της πόλης της Ελασσόνας κάθε 5 δευτερόλεπτα, ενώ μπορεί ο καθένας να πλοηγηθεί στις εικόνες γρηγορότερα χρησιμοποιώντας τα δυο βελάκια στα πλάγια. Κάθε εικόνα έχει ένα ξεχωριστό τίτλο που μας βοηθάει να καταλάβουμε περισσότερα για το αντικείμενο τις εικόνες.

Random Hotel

Ακριβώς κάτω από το Slider μας προβάλλουμε ένα random item της κατηγορίας hotel όπως φαίνεται και στο tag τις φωτογραφίας . Εάν κάνουμε ανανέωση στην ιστοσελίδα μας βλέπουμε πως το αντικείμενο αυτό αλλάζει.



Olympion Hotel

Ισαύρων 2 / +30 24930 22278

Το ξενοδοχείο Olympion βρίσκεται στην πόλη της Ελασσόνας στην Θεσσαλία. Διαθέτει άνετα δωμάτια, πλήρως εξοπλισμένα και όμορφα διακοσμημένα με θέα...

Εικόνα 37

Κάτω από την φωτογραφία βλέπουμε το summary του εκάστοτε ξενοδοχείου.

Η ρύθμιση αυτή γίνεται με τον κώδικα

```
1. $view->hotel = Model_Hotel::query()->order_by(DB::expr('RAND()'))->get_one();
```

που βρίσκεται στον controller της mainpage μας. Όπως παρατηρούμε εμφανίζεται ένα ξενοδοχείο διότι χρησιμοποιούμε μετά την συνάρτηση rand την get_one.

Δρομολόγια

Ακριβώς δίπλα από το random hotel υπάρχει ένα link το οποίο μας παραπέμπει στα δρομολόγια της Ελασσόνας.

Επερχόμενες εκδηλώσεις



› dignissimos ducimus qui
niti atque corrupti quos dolores
cati cupiditate...

Olympus Marathon

By koox / September 30, 2014



Ορειν αγώνας ορειβατικού

ΣΚΙ

By admin / October 09, 2014



Εικόνα 38

Δίπλα από τα δρομολόγια υπάρχει ένα menu που μας προβάλλει τις δυο πιο κοντινές εκδηλώσεις σε σχέση με την ημερομηνία διεξαγωγής τους προβάλλοντας τα αποτελέσματα τους με αύξουσα σειρά.

Για να γίνει εφικτή η παραπάνω λειτουργία προσθέτουμε στον controller της mainpage το παρακάτω κομμάτι κώδικα.

```
1. $view-  
>events = Model_Event::find('all', array('order_by' => array('eventdate' => 'asc')  
, 'limit' => 2));
```

Βλέπουμε πως γίνεται orderby eventdate και τα αποτελέσματα εμφανίζονται με αύξουσα σειρά ενώ το limit είναι δυο για αυτό και εμφανίζονται στην αρχική μας σελίδα δυο εκδηλώσεις. Το νούμερο αυτό μπορεί να αλλάξει και να προβληθούν αντίστοιχου αριθμού άρθρα αρκεί να υπάρχουν στην βάση μας.

Facts



runt

qui
dolores

Το 2006 με τροπολογία που ψηφίστηκε στη Βουλή αναγνωρίστηκαν επίσημα ως ξεχωριστές κοινότητες τα Ζωνιανά Ρεθύμνου, το Βραχάσι Λασιθίου και η Τσαριτσάνη Λάρισας.



wikipedia.com



Φωτογραφίες



Τελευταία Tweets

@htmlstream At vero seos etodela
ccusamus et iusto odio dignissimos.
<http://t.co/sBav7dm>
5 hours ago

Εικόνα 39

Κάτω από τις επερχόμενες εκδηλώσεις και λίγο πιο αριστερά βρίσκονται τα facts μικρές πληροφορίες ενημερωτικού χαρακτήρα οι οποίες αλλάζουν κάθε 5 δευτερόλεπτα και καλούνται όπως είναι λογικό στον controller της mainpage με τον εξής κώδικα.

```
1. $view->facts = Model_Fact::query()->order_by(DB::expr('RAND()'))->get();
```

Σε αντίθεση με το get_one των ξενοδοχείων το get μας δίνει την δυνατότητα να επιλέξουμε τυχαία ανάμεσα σε όλα μας τα αντικείμενα.

Ξενοδοχεία

Comments

Στο section comments στο view του ξενοδοχείου παραθέτουμε τα σχόλια των χρηστών. Το model των σχολίων έχει ένα πεδίο parent_id που υποδηλώνει όπως το λέει άλλωστε και η λέξη αν αυτό το σχόλιο αποτελεί απάντηση σε κάποιο άλλο σχόλιο.

Comments



byadmin

May 19, 2014 / [Reply](#)

Staff very helpful, welcoming, full breakfast, and excellent location Bathrooms could use a bit of updating (depends on the room you get)



bykoox

May 17, 2014 / [Reply](#)

Location was easy to walk to everything. Really good restaurants within a very short walking distance. Breakfast included and it was really good. Coffee/tea/biscuits offered almost all day. Comfortable sitting area. Exceptionally helpful and friendly staff. We just loved this hotel!! Would always stay here!! Bed was a bit hard but the put a pad on for us. They are very accommodating.



byadmin

May 17, 2014 / [Reply](#)

Location Rooftop restaurant Room size Good selection for breakfast Wi-Fi access Executive room - needs updating Lack of space in shower Room temperature can't be manual adjusted Mini Fridge not cold

Εικόνα 40

Μια από τις πολλές λειτουργίες που δημιουργήσαμε είναι η εμφάνιση αυτών των σχολίων με συγκεκριμένη μορφοποίηση ώστε να υποδηλώνει εμφανώς αυτό το id. Για αυτό τον λόγο γράψαμε την κλάση NestedList

```

public function make_list($arr)
{
    $options['image']=urlencode('monsterid');
    self::$count ++;

    $return = '<div class="media">';
    foreach ($arr as $item)
    {
        $return .=
        <hr>
        <div class="media">
            <a class="pull-left" href="'. $item->url .'">
                
            </a>
            <div class="media-body">
                <h4 class="media-heading"><em>by </em>
                    . $item->users->username.
                    <span>'.date('F d,Y',$item->created_at).' /<a href="'. "#". $item->id.'"> Reply </a></span>
                </h4>
                <p>
                    . $item->body.'
                </p>';

        $return .= (($item['childs']) ? NestedList::make_list($item['childs']).'</div>' : '</div>' );
    }
    $return .= '</div></div>';
    return $return;
}
}

```

Εικόνα 41

```

class NestedList{

    private $items;

    public function __construct($items) {
        $this->items = $items;
    }
    private static $count = 1;

    //επιστρέφει array με τα παιδιά ενός parent
    public function buildTree() {

        $childs = array();

        foreach($this->items as $item)
        {
            $childs[$item->parent_id][] = $item;
        }

        foreach($this->items as $item) if (isset($childs[$item->id]))
        {
            $item->childs = $childs[$item->id];
        }

        return $childs[0];
    }
}

```

Εικόνα 42

Η function buildTree επιστρέφει ένα array με τα παιδιά ενός parent id ενώ η function make_list είναι μια αναδρομική συνάρτηση η οποία επιστρέφει το κατάλληλα διαμορφωμένο κείμενο.

Breadcrumbs

Όπως παρατηρούμε τα πεδία των ξενοδοχείων, στα δεξιά μας έχουμε τοποθετήσει τα breadcrumbs, πρόκειται για ένα σύστημα ταξινόμησης που μας δείχνει που βρισκόμαστε κάθε φορά μέσα στην ιστοσελίδα.

The screenshot shows a website interface with a navigation menu at the top: ΑΡΧΙΚΗ, ΞΕΝΟΔΟΧΕΙΑ, RENT A CAR, ΕΚΔΗΛΩΣΕΙΣ, ΔΡΟΜΟΛΟΓΙΑ, ΕΠΙΚΟΙΝΩΝΙΑ, and a search icon. Below the menu, the page title is 'Ξενοδοχεία' and the breadcrumb is 'Αρχική / Ξενοδοχεία'. On the left, there is a search form titled 'Αναζήτηση Δωματίου' with fields for 'Ξενοδοχείο', 'Ημερομηνία Check in', and 'Ημερομηνία Check Out'. Below the form, it says 'Δείξει/Απέκρυψε παραπάνω επιλογές'. On the right, there are two hotel listings. The first is 'RiverSide Rooms' with a 3.5 star rating, a photo of a building, and a description: 'Ξενοδοχείο της οικογένειας Μεζίλη στην Ελασσάνα. Ο συνδυασμός του πευκοδάσους και του Ελασσονίτη ποταμού, δίπλα στον οποίο βρίσκεται η μονάδα ενοικιαζομένων δωματίων RiverSide, το καθιστούν τον ιδανικό τόπο διαμονής και χαλάρωσης στην Ελασσάνα.' Below the description is a button 'Διαβάστε Περισσότερα'. The second listing is 'Street Car Named Desire' with a 0 star rating, a photo of a person, and a description: 'A very nice hotel for pets!' Below the description is a button 'Διαβάστε Περισσότερα'. In the bottom right corner, there is a 'Code Profi' logo.

Εικόνα 43

Φίλτρα αναζήτησης

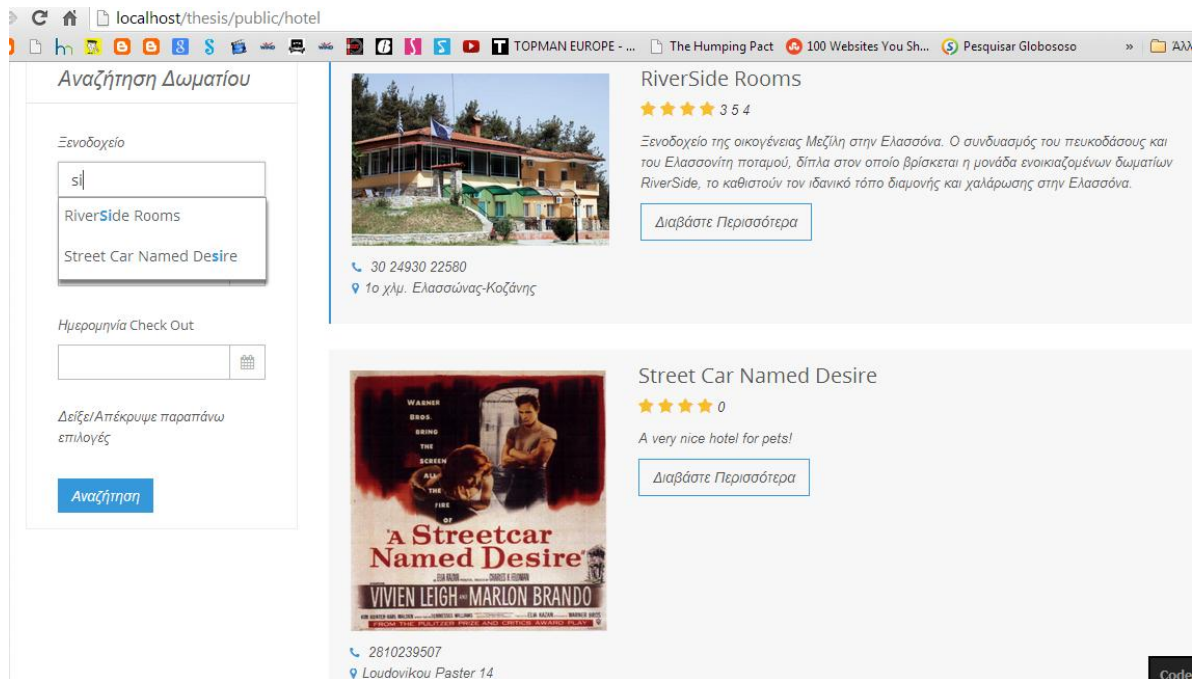
Στο αριστερό sidebar υπάρχει μια φόρμα αναζήτησης που μας βοηθάει να προσαρμόσουμε τα κριτήρια της έρευνας μας.

Ας εξηγήσουμε όμως σταδιακά και αναλυτικά όλα τα στοιχεία της φόρμας.

Ξενοδοχείο

Στο πεδίο αυτό εισάγουμε το όνομα του ξενοδοχείου που ψάχνουμε, αν φυσικά το γνωρίζουμε αλλιώς προτείνεται να μείνει κενό. Μπορούμε πάντα με την τεχνική autocomplete που έχουμε δημιουργήσει εάν γράψουμε κάποια γράμματα να πάρουμε κάποιες προτάσεις από ξενοδοχεία που περιλαμβάνουν τους χαρακτήρες αυτούς.

Π.χ.



The screenshot shows a web browser window with the URL localhost/thesis/public/hotel. The page title is "Αναζήτηση Δωματίου". On the left, there is a search form with a text input containing "sil", a dropdown menu showing suggestions "RiverSide Rooms" and "Street Car Named Desire", and a "Ημερομηνία Check Out" field. Below the form is a "Δείξε/Απέκρυψε παραπάνω επιλογές" link and a blue "Αναζήτηση" button. The main content area displays two search results. The first is "RiverSide Rooms" with a 3.5 star rating, a photo of a building, a description in Greek, a phone number (30 24930 22580), and a location (1ο χλμ. Ελασσώνας-Κοζάνης). The second is "Street Car Named Desire" with a 0 star rating, a movie poster image, a description "A very nice hotel for pets!", a phone number (2810239507), and a location (Loudavikou Paster 14). A "Code" label is visible in the bottom right corner of the results area.

Εικόνα 44

Ημερομηνία Check in / Check out

Στα πεδία αυτά συμπληρώνουμε τις ημερομηνίες για τις οποίες θέλουμε να κάνουμε την κράτηση μας. Οι ημερομηνίες που μπορούμε να κάνουμε checkin εμφανίζονται από την τρέχουσα σημερινή ημερομηνία και εκείνες του checkout από την επόμενη της τρέχουσα και έπειτα. Για την επιλογή των ημερομηνιών αποφασίσαμε να χρησιμοποιήσουμε το date picker plugin

Show/Hide more options

Ακριβώς κάτω από τις ημερομηνίες checkin checkout βρίσκεται η επιλογή «Δείξε/Απέκρυψε παραπάνω επιλογές η οποία με mouse over υπογραμμίζεται με μπλε underline. Κάνοντας κλικ επάνω του εμφανίζονται διάφορα input types, πιο συγκεκριμένα


Αριθμός ατόμων

που δηλώνει ουσιαστικά τον αριθμό κρεβατιών που θέλουμε.

3 άτομα 3 κρεβάτια κ.ο.κ.

Input range

Που καθορίζει το μέγιστο ποσό το οποίο διατίθεται ο πελάτης να πληρώσει.



RiverSide Rooms ★ ★ ★ ★

📍 1ο χλμ. Ελασσώνας-Κοζάνης 📞 30 24930 22580 ✉ info@elassonahotel.gr

Ξενοδοχείο της οικογένειας Μεζίλη στην Ελασσόνα. Ο συνδυασμός του πευκοδάσους και του Ελασσονίτη ποταμού, δίπλα στον οποίο βρίσκεται η μονάδα ενοικιαζομένων δωματίων RiverSide, το καθιστούν τον ιδανικό τόπο διαμονής και χαλάρωσης στην Ελασσόνα.

Βρήκαμε 3 διαθέσιμα δωμάτια με αυτά τα κριτήρια. Από: 10€

[Κάντε κράτηση](#)



Gorgolaini Hotel ★ ★ ★ ★ ★

📍 gorgolaini 32, Herakleio 📞 2810331549 ✉ cooxlee@hotmail.com

Situated at the foot of the Acropolis, Gorgolaini Hotel has elegantly decorated air-conditioned rooms. It has a beautiful roof garden with lounge chairs and 2 hot bathtubs with spectacular views of Herakleio.

Βρήκαμε 3 διαθέσιμα δωμάτια με αυτά τα κριτήρια. Από: 15€

[Κάντε κράτηση](#)

Στο fuelrhr υπάρχει η κλάση Validation και εμείς στην ουσία χρησιμοποιούμε τις μεθόδους της για να βελτιστοποιήσουμε την ποιότητα της αναζήτησης μας.

Για να εξασφαλίσουμε ότι οι χρήστες θα θέτουν σωστά το ερώτημα αναζήτησης τους στην βάση μας πραγματοποιούμε μια σειρά από validations τα οποία διέπονται από ορισμένους κανόνες και οι οποίοι ανάλογα τις συνθήκες που πληρούνται η όχι πυροδοτούν συγκεκριμένες αντιδράσεις.

Πιο συγκεκριμένα θα αναλύσουμε τα validations που γίνονται στην αναζήτηση ξενοδοχείου.

```
1. $val = Validation::forge();
2.
3.     $val->add('start', 'Check In')
4.         ->add_rule('required')
5.         ->add_rule('valid_date')
6.         ->add_rule('numeric_between', $today , Input::post('finish'));
7.
8.     $val->add('finish', 'Check Out')
9.         ->add_rule('required_with', 'start')
10.        ->add_rule('valid_date');
11.
12.    $val->add('hotel', 'Hotel')
13.        ->add_rule('match_value', $hotelnames);
```

Στην συνέχεια και αφού περάσουν τα Validation αρχίζουμε το χτίσιμο του query μας,

Αρχικά χρησιμοποιώντας την τεχνική eager loading, δηλαδή κάνοντας left join τα table που σχετίζονται με τα ξενοδοχεία (rooms και bookings) το οποίο σχετίζεται με το hotel μέσω του table rooms) δημιουργούμε ένα αντικείμενο \$hotel το οποίο έχει όλες τις συσχετίσεις που χρειαζόμαστε για να κάνουμε την αναζήτηση.

```
1. $hotel = Model_Hotel::query()
2.     ->related('rooms')
3.     ->related('rooms.bookings');
```

Με αυτό τον τρόπο τρέχοντας ένα query έχουμε όλα τα πεδία τα οποία χρειαζόμαστε. Το query το χτίζουμε σύμφωνα με τα input του user.

Αφού δημιουργήσαμε την κλάση **ReserveQueries** και τις συναρτήσεις που εμπεριέχονται σε αυτή μπορούμε απλά να κάνουμε instantiate ένα object της κλάσης ReserveQueries μέσα στον controller reservations και να χρησιμοποιήσουμε όλες τις functions της πρώτης.

Ας αναλύσουμε ωστόσο τις functions που χρησιμοποιούμε μέσα στην κλάση ReserveQueries

get_table_names

```
1. public function get_table_names($table) {
2.
3.     $query = DB::select('name')->from($table)->execute()->as_array('name');
4.     $names = array();
5.     foreach ($query as $key => $value)
6.     {
7.         $names[$key] = $value['name'];
8.     }
9.     return $names;
10. }
```

Η συνάρτηση `get_table_names` επιστρέφει ένα array με όλα τα ονόματα των ξενοδοχείων

search_hotel

```
1. public function search_hotel($hotel){
2.
3.     $checkin = Session::get('dates.0');
4.     $checkout = Session::get('dates.1');
5.
6.     if(!is_null(Session::get('hotel')))
7.     {
8.         $hotel = $hotel
9.             ->where('name',Session::get('hotel'));
10.    }
11.
12.    $hotel = $hotel
13.        ->related('rooms')
14.        ->related('rooms.bookings')
15.        ->and_where_open();
16.    $hotel = $hotel
17.        ->or_where_open()
18.        ->where('rooms.bookings.checkIn', '>', $checkin)
19.        ->where('rooms.bookings.checkIn', '>=', $checkout)
20.        ->or_where_close()
21.        ->or_where_open()
22.        ->where('rooms.bookings.checkOut', '<=', $checkin)
23.        ->where('rooms.bookings.checkOut', '<', $checkout)
24.        ->or_where_close()
25.        ->or_where('rooms.bookings.id', '=', NULL)
26.        ->and_where_close();
27.
28.    if(!is_null(Session::get('guests')))
29.    {
30.        $hotel = $hotel-
>where('rooms.beds', '>=', Session::get('guests'));
```

```

31.     }
32.     else
33.     {
34.         Session::delete('guests');
35.     }
36.
37.     if(!is_null(Session::get('price')))
38.     {
39.         $hotel = $hotel-
>where('rooms.pricepernight', '<=', Session::get('price'));     }
40.         else
41.         {
42.             Session::delete('price');
43.         }
44.         //Επειδή έχουμε ένα πεδίο που είναι κλειδί
45.         $hotel = $hotel->where('rooms.id', '!=', NULL)-
>order_by('rooms.pricepernight', 'ASC');
46.
47.         return $hotel;
48.     }

```

Η συνάρτηση `search_hotel` επιστρέφει το/α ξενοδοχείο/α με διαθέσιμα δωμάτια ανάλογα με τα κριτήρια της αναζήτησης μας με αύξουσα σειρά, δηλαδή από το πιο φθινό στο πιο ακριβό.

`get_best_comment`

```

1. public function get_best_comment($id) {
2.
3.     $best_comment = Model_Hotelcomment::query()
4.         ->where('hotel_id', $id)
5.         ->order_by('rating', 'DESC')
6.         ->order_by('created_at', 'DESC')
7.         ->related('users')
8.         ->get_one();
9.
10.    return $best_comment;
11. }

```

Η συνάρτηση `get_best_comment` επιστρέφει το πιο πρόσφατο σχόλιο από αυτά που έχουν την υψηλότερη βαθμολογία

Επιπροσθέτως στο view του εκάστοτε hotel που έχουμε επιλέξει δημιουργήσαμε μια επιπλέον φόρμα η οποία σαφώς κρατάει τα κριτήρια που επιλέξαμε νωρίτερα αλλά παράλληλα μας δίνει την δυνατότητα να τα αλλάξουμε σε περίπτωση που π.χ. θέλουμε να επιλέξουμε διαφορετικές ημερομηνίες κ.α.

#	Room Type	Price per Night	Action
11	1	10	Κάντε κράτηση
12	1	13	Κάντε κράτηση
13	1	14	Κάντε κράτηση

Εικόνα 45

Μόλις διαλέξουμε το δωμάτιο της αρεσκείας μας προχωράμε ένα βήμα παρακάτω στην κράτηση μας η οποία ολοκληρώνεται όπως θα δούμε παρακάτω step by step.

Κράτηση

RiverSide Rooms ★★★★★

1ο χλμ. Ελασσώνας-Κοζάνης ☎ 30 24930 22580 ✉ info@elassonahotel.gr

Ξενοδοχείο της οικογένειας Μεζίλη στην Ελασσόνα. Ο συνδυασμός του πευκοδάσους και του Ελασσονίτη ποταμού, δίπλα στον οποίο βρίσκεται η μονάδα ενοικιαζομένων δωματίων RiverSide, το καθιστούν τον ιδανικό τόπο διαμονής και χαλάρωσης στην Ελασσόνα.

Βρήκαμε 3 διαθέσιμα δωμάτια με αυτά τα κριτήρια. Από: 10€

Κάντε κράτηση

Εικόνα 46

Προκειμένου να είναι σε θέση ο χρήστης να ολοκληρώσει την κράτηση του θα πρέπει όπως αναφέραμε παραπάνω να συμπληρώσει και κάποια περαιτέρω menu.

Τα τρία αυτά menu εμφανίζονται σταδιακά και μπορούμε να επέμβουμε στο επόμενο μόνο εάν έχουμε συμπληρώσει όλες τις τιμές που θέλουμε για την κράτηση μας σωστά.

Βήμα Πρώτο: Edit profile

The screenshot shows a web interface for editing a user profile. At the top, there is a navigation bar with three tabs: 'Edit Profile' (active), 'Details', and 'Invoice'. Below the navigation bar is a form titled 'Manage your Name, ID and Email Addresses'. The form contains the following fields and elements:

- Username:** Input field with a user icon on the right.
- First name:** Input field with a user icon on the right.
- Last name:** Input field with a user icon on the right.
- Address:** Input field with an envelope icon on the right.
- Email address:** Input field with an envelope icon on the right.
- Enter your Password:** Input field with a lock icon on the right.
- Re-type Password:** Input field with a lock icon on the right.
- I agree with the Terms and Conditions
- Reset:** Button
- Save:** Button

At the bottom right of the form, there is a small button labeled 'Code Profile'.

Εικόνα 47


Στην φόρμα του edit profile εάν δεν είμαστε εγγεγραμμένοι χρήστες συμπληρώνοντας τα στοιχεία που ζητούνται δημιουργούμε έναν καινούριο user στην βάση μας με τα στοιχεία με τα στοιχεία που δόθηκαν στα input fields. Τώρα εάν είμαστε ήδη συνδεδεμένοι την ώρα που κάνουμε την κράτηση μας το σύστημα αφενός μας δίνει την δυνατότητα να κάνουμε την κράτηση μας με τα στοιχεία του user που είναι συνδεδεμένος και υπάρχουν ήδη στην βάση δεδομένων μας. Σημειώστε πως αν παρόλα αυτά θέλουμε να κάνουμε μια κράτηση με διαφορετικά στοιχεία αυτό είναι εφικτό λόγω της προαναφερθείσας τεχνοτροπίας.

Βήμα Δεύτερο: Details

[Edit Profile](#) [Details](#) [Invoice](#)

Preview your booking

Hotel: RiverSide Rooms



Ξενοδοχείο της οικογένειας Μεζίλη στην Ελασσόνα. Ο συνδυασμός του πευκοδάσους και του Ελασσονίτη ποταμού, δίπλα στον οποίο βρίσκεται η μονάδα ενοικιαζομένων δωματίων RiverSide, το καθιστούν τον ιδανικό τόπο διαμονής και χαλάρωσης στην Ελασσόνα.

Room / Reservation Details

User details

[Next](#)


Εικόνα 48

[Edit Profile](#) [Details](#) [Invoice](#)

Preview your booking

Hotel: RiverSide Rooms

Room / Reservation Details

#	Guests	Check In	Check out	Price	Remove
11	1	2014-11-13	2014-11-27	10	

User details

[Next](#)







Εικόνα 49

Hotel: RiverSide Rooms
Room / Reservation Details
User details
<p><i>Check if your account details are correct.</i></p> <p>Your name AKRIVI</p> <hr/> <p>Your last name KATIFORI</p> <hr/> <p>Primary Email Address minas.mathew@gmail.com</p> <hr/> <p>Address DRAKOULI 24</p> <hr/> <p>Change</p>

[Next](#)

Εικόνα 50

Σε περίπτωση που είχαμε κάνει και κράτηση οχήματος η φόρμα θα ήταν ως εξής.

Edit Profile	Details	Invoice													
<p>Preview your booking</p>															
<table border="1"> <tr> <td> <p>Hotel: RiverSide Rooms</p>  </td> <td> <p>Ξενοδοχείο της οικογένειας Μεζίλη στην Ελασσόνα. Ο συνδυασμός του πευκοδάσους και του Ελασσονίτη ποταμού, δίπλα στον οποίο βρίσκεται η μονάδα ενοικιαζόμενων δωματίων RiverSide, το καθιστούν τον ιδανικό τόπο διαμονής και χαλάρωσης στην Ελασσόνα.</p> </td> </tr> <tr> <td colspan="2"> <p>Room / Reservation Details</p> </td> </tr> <tr> <td colspan="2"> <p>Rental Provider: Avis</p>  </td> <td> <p>We're an operating division of Avis Budget EMEA Ltd, the leading car rental company in Europe, Africa, the Middle East and Asia, with more than 30 years' experience in the car rental industry. Avis Europe serves over 8 million customers annually through a network spanning 168 countries.</p> </td> </tr> <tr> <td colspan="2"> <p>Vehicle / Reservation Details</p> </td> <td></td> </tr> <tr> <td colspan="2"> <p>User details</p> </td> <td></td> </tr> </table>			<p>Hotel: RiverSide Rooms</p> 	<p>Ξενοδοχείο της οικογένειας Μεζίλη στην Ελασσόνα. Ο συνδυασμός του πευκοδάσους και του Ελασσονίτη ποταμού, δίπλα στον οποίο βρίσκεται η μονάδα ενοικιαζόμενων δωματίων RiverSide, το καθιστούν τον ιδανικό τόπο διαμονής και χαλάρωσης στην Ελασσόνα.</p>	<p>Room / Reservation Details</p>		<p>Rental Provider: Avis</p> 		<p>We're an operating division of Avis Budget EMEA Ltd, the leading car rental company in Europe, Africa, the Middle East and Asia, with more than 30 years' experience in the car rental industry. Avis Europe serves over 8 million customers annually through a network spanning 168 countries.</p>	<p>Vehicle / Reservation Details</p>			<p>User details</p>		
<p>Hotel: RiverSide Rooms</p> 	<p>Ξενοδοχείο της οικογένειας Μεζίλη στην Ελασσόνα. Ο συνδυασμός του πευκοδάσους και του Ελασσονίτη ποταμού, δίπλα στον οποίο βρίσκεται η μονάδα ενοικιαζόμενων δωματίων RiverSide, το καθιστούν τον ιδανικό τόπο διαμονής και χαλάρωσης στην Ελασσόνα.</p>														
<p>Room / Reservation Details</p>															
<p>Rental Provider: Avis</p> 		<p>We're an operating division of Avis Budget EMEA Ltd, the leading car rental company in Europe, Africa, the Middle East and Asia, with more than 30 years' experience in the car rental industry. Avis Europe serves over 8 million customers annually through a network spanning 168 countries.</p>													
<p>Vehicle / Reservation Details</p>															
<p>User details</p>															
<p>Next</p>															


Εικόνα 51

Προχωρώντας στο επόμενο βήμα details φτάνουμε σε ένα collapsible menu το οποίο φροντίζει να μας ενημερώσει για τα 1)Ξενοδοχείου που διαλέξαμε 2)Το δωμάτιο του ξενοδοχείου καθώς επίσης και για τις λεπτομέρειες της κράτησης μας και 3)Για τα στοιχεία του χρήστη που διαλέξαμε. Σαφώς και όπως αναφέραμε προηγουμένως όποια εάν θέλουμε π.χ. να αλλάξουμε τα στοιχεία του χρήστη γιατί θέλουμε να κάνουμε την κράτηση με διαφορετικό όνομα μπορούμε πατώντας το κουμπί change στο user details να αλλάξουμε τα στοιχεία μας και να προχωρήσουμε ξανά στην κράτηση μας. Αν όλα είναι εντάξει και επιθυμούμε να προχωρήσουμε με την κράτηση μας πατάμε το κουμπί save changes.

Βήμα τρίτο : Invoice

Κράτηση Αρχική / Κράτηση

[Edit Profile](#) [Details](#) [Invoice](#)

 #123456789 / 20, October 2013
Auiderer facilis consectetur

Client Information:
First Name: AKRIVI
Last Name: KATIFORI
Email: MINAS.MATHEW@GMAIL.COM
Address: DRAKOULI 24

Payment Details:
Bank Name: FOO
Account Number: BAR
SWIFT Code: FOO
V.A.T Reg #: BAR

Payment Invoice

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est

[Code Prof](#)

Εικόνα 52

Αφού φτάσουμε στο τρίτο και τελευταίο μενού το Invoice μας παρουσιάζονται όλα τα στοιχεία της κράτησης που διαλέξαμε να κάνουμε, από το όνομα στο οποίο έγινε η κράτηση το ξενοδοχείου, δωμάτιο που επιλέχθηκε κ.α. Εάν θέλουμε να οριστικοποιήσουμε την κράτηση μας πατάμε το κουμπί **submit the invoice** και η κράτηση μας ολοκληρώνεται. Στο μενού αυτό έχουμε επίσης την δυνατότητα να τυπώσουμε το invoice μας. Αφού τελειώσουμε με μια κράτηση μεταφερόμαστε στο προφίλ του εκάστοτε χρήστη όπου μπορούμε να δούμε τις κρατήσεις του και να ακυρώσουμε εκείνες που δεν έχουν γίνει σε προγενέστερες ημερομηνίες από την τρέχουσα ή ακόμα και αυτές που ανήκουν στην τρέχουσα ημερομηνία το πώς ακριβώς λειτουργεί αυτός ο μηχανισμός το εξηγούμε παρακάτω.

Προφίλ χρήστη

Όταν ένας χρήστης είναι συνδεδεμένος μπορούμε κάνοντας κλικ στο όνομα του να επισκεφτούμε το προφίλ του. Στο προφίλ κάθε χρήστη υπάρχουν τα εξής τρία μενού.

Reservation history

Στο μενού αυτό μπορούμε να δούμε το ιστορικό των κρατήσεων και διάφορα στοιχεία γύρω από αυτές όπως παραδείγματος χάρη τον κωδικό κράτησης ο οποίος είναι μοναδικός, το δωμάτιο του ξενοδοχείου που έχουμε κλείσει κ.α. Μπορούμε εάν το επιθυμούμε να διαγράψουμε μια κράτηση πατώντας το κουμπί cancel αφού πρώτα επαληθεύσουμε ότι διαγράφουμε την σωστή κράτηση σε περίπτωση δυο η παραπάνω κρατήσεων μέσω του ίδιου χρήστη.

Πιο συγκεκριμένα το μενού του reservation history χωρίζεται σε 2 επιμέρους κατηγορίες

Κράτηση Ξενοδοχείου

Προφίλ χρήστη Αρχική / Προφίλ χρήστη

Κρατήσεις Δωματιων Κρατήσεις Οχημάτων

Επικείμενες κρατήσεις

#	Description	Booking Code	Checkin	Checkout	Status
1	δωμάτιο #11 στο ξενοδοχείο RiverSide Rooms	YHMVYA	2014-11-12 0...	2014-11-15 0...	Cancel
2	δωμάτιο #17 στο ξενοδοχείο Gorgolaini Hotel	YEJ7SZ	2014-11-11 0...	2014-11-20 0...	Cancel

Τρεχουσες κρατήσεις

#	Description	Booking Code	Checkin	Checkout
1	δωμάτιο #12 στο ξενοδοχείο RiverSide Rooms	9RHVVX	2014-11-02 00:00:00	2014-11-08 00:00:00

Ιστορικο κρατήσεων

#	Description	Booking Code	Checkin	Checkout	Code Prof
---	-------------	--------------	---------	----------	-----------

Εικόνα 53

Στην κράτηση ξενοδοχείου μπορούμε να δούμε όλες μας τις κρατήσεις σύμφωνα με τις τρεις υποκατηγορίες που εξηγούνται παρακάτω.

Κράτηση Οχήματος

The screenshot shows a user profile page titled "Προφιλ χρήστη" (User Profile) with a navigation bar containing "Αρχική / Προφιλ χρήστη". The profile picture is a cartoon crab. The main content area is divided into two tabs: "Κρατήσεις Δωματιών" (Room Bookings) and "Κρατήσεις Οχημάτων" (Car Bookings), with the latter being active. Under "Επικείμενες κρατήσεις" (Upcoming bookings), there is a message: "Δεν έχετε επικείμενες κρατήσεις." (You have no upcoming bookings). Under "Τρεχουσες κρατήσεις" (Active bookings), there is one entry: # 1, Description: οχημα #asd του Rental Provider Avis, Booking Code: asdfgv, Checkin: 2014-11-02 23:56:10, Checkout: 2014-11-04 00:00:00. Under "Ιστορικο κρατήσεων" (Booking history), there is a message: "Δεν έχετε κάνει κράτηση πριν απο σήμερα." (You have not made a booking before today). A sidebar on the left contains "Reservation History", "Comments", and "Settings". At the bottom, there is a calendar for November 2014 and a "Code Profile" button.

Εικόνα 54

Στην κράτηση οχήματος μπορούμε να δούμε όλες μας τις κρατήσεις σύμφωνα με τις τρεις υποκατηγορίες που εξηγούνται παρακάτω.

Επικείμενες κρατήσεις

The screenshot shows the "Επικείμενες κρατήσεις" (Upcoming bookings) section. It contains a table with the following data:

#	Description	Booking Code	Checkin	Checkout	Status
1	δωμάτιο #11 στο ξενοδοχείο RiverSide Rooms	YHMYVA	2014-11-12 0...	2014-11-15 0...	Cancel
2	δωμάτιο #17 στο ξενοδοχείο Gorgolaini Hotel	YEJTSZ	2014-11-11 0...	2014-11-20 0...	Cancel

Εικόνα 55

Οι επικείμενες κρατήσεις αφορούν κρατήσεις σε ημερομηνίες μελλοντικές σε σχέση με την τρέχουσα(σημερινή) που μπορούν να διαγραφούν

Παλαιότερες κρατήσεις

Ιστορικό κρατήσεων

#	Description	Booking Code	Checkin	Checkout
Δεν έχετε κάνει κράτηση πριν απο σήμερα.				

Εικόνα 56

Στην κατηγορία αυτή ο κάθε χρήστης μπορεί να δει μια λίστα από τις κρατήσεις που έχει κάνει και αφορούν ημερομηνίες προγενέστερες της τρέχουσας. Όπως είναι λογικό εφόσον οι κρατήσεις αυτές είναι προγενέστερες δεν υπάρχει νόημα να μπορεί να ακυρωθούν.

Τρέχουσες κρατήσεις

Τρέχουσες κρατήσεις

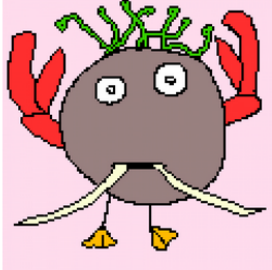
#	Description	Booking Code	Checkin	Checkout
1	δωμάτιο #12 στο ξενοδοχείο RiverSide Rooms	9RHVVX	2014-11-02 00:00:00	2014-11-08 00:00:00

Εικόνα 57

Στις τρέχουσες κρατήσεις θα βρούμε τις κρατήσεις που έχουν γίνει από τον χρήστη μας και είναι σε ισχύ αυτή την στιγμή δηλαδή βρίσκονται ήδη μέσα στο περιθώριο του check in / check out της κράτησης που έκαναν. Στην κατηγορία αυτή επίσης δεν υπάρχει νόημα να μπορεί να ακυρωθεί η κράτηση αφού είναι ήδη σε ισχύ.

Comments

Προφιλ χρήστη Αρχική / Προφιλ χρήστη



- Reservation History
- Comments**
- Settings

< November 2014 >

User Comments In Hotels

- admin** @RiverSide Rooms 6 μήνες πριν
Staff very helpful, welcoming, full breakfast, and excellent location Bathrooms could use a bit of updating (depends on the room you get)
25 Likes 10 Share
- admin** @RiverSide Rooms 6 μήνες πριν
Location Rooftop restaurant Room size Good selection for breakfast W-Fi access Executive room - needs updating Lack of space in shower Room temperature can't be manual adjusted Mini Fridge not cold
25 Likes 10 Share
- admin** @RiverSide Rooms 13 ώρες πριν
ορα
25 Likes 10 Share

Code Profi

Εικόνα 58

Στο μενού comments μπορούμε να δούμε όλα τα σχόλια που έχουν γίνει από τον εν λόγο χρήστη για ένα ξενοδοχείο καθώς επίσης και την ημερομηνία που έγιναν αυτά.

Contact

Χαιρόμαστε πάντα να μαθαίνουμε την άποψή σας και υπάρχουν πολλοί τρόποι να επικοινωνήσετε μαζί μας.



Επικοινωνία

📍 Οδός Ερμής, Ελασσόνα 40200
✉ info@elassona.gr
☎ +30) 24930 24389
🌐 <http://www.elassona.gr>

Ωράριο

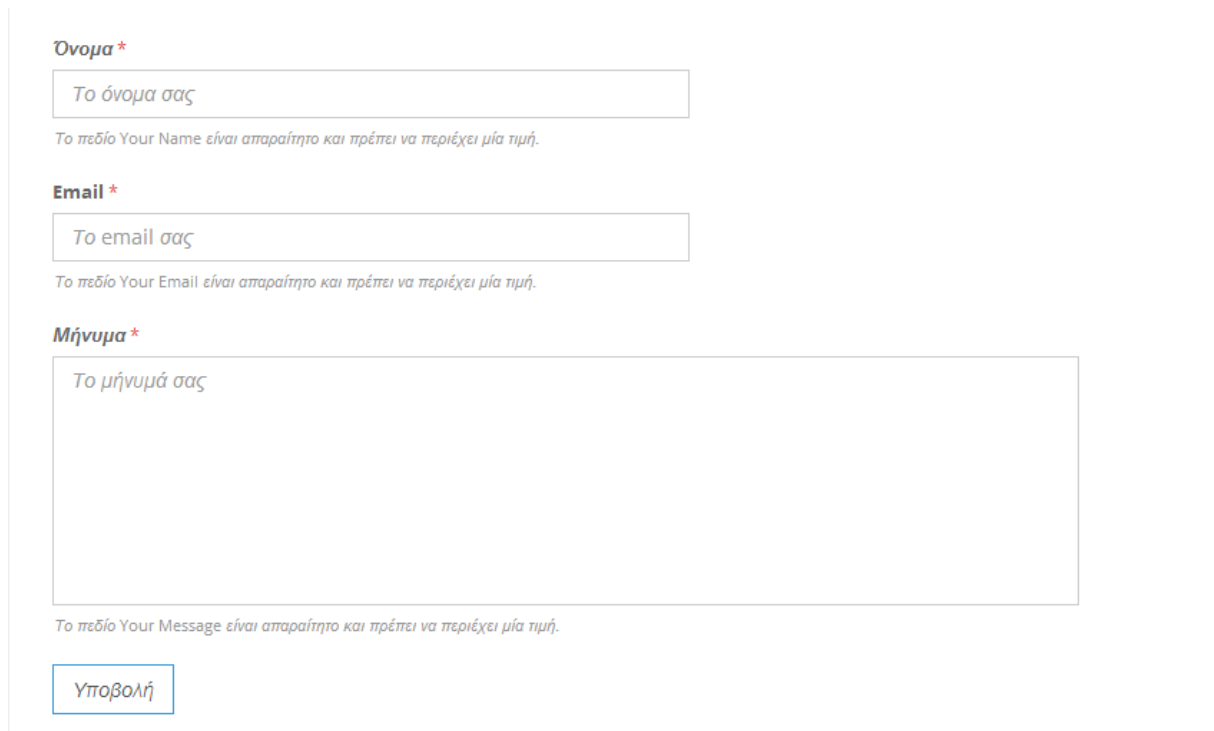
Δευτ.-Παρ.: 10:00 με 20:00
Σάββατο 11:00-13:00

Αφήστε μας σχόλιο

Εικόνα 59

Κάτω από τον χάρτη της Ελασσόνας που βρίσκεται στο κέντρο της σελίδας υπάρχει μια φόρμα την οποία μπορούν να χρησιμοποιήσουν οι επισκέπτες για να επικοινωνήσουν μαζί μας.

Όλα τα πεδία της φόρμας είναι απαραίτητο να συμπληρωθούν προκειμένου να αποσταλεί το οποιοδήποτε μήνυμα.



The image shows a contact form with three input fields and a submit button. The first field is labeled 'Όνομα *' (Name) and contains the placeholder text 'Το όνομά σας'. Below it is a small note: 'Το πεδίο Your Name είναι απαραίτητο και πρέπει να περιέχει μία τιμή.' The second field is labeled 'Email *' and contains the placeholder text 'Το email σας'. Below it is a small note: 'Το πεδίο Your Email είναι απαραίτητο και πρέπει να περιέχει μία τιμή.' The third field is labeled 'Μήνυμα *' (Message) and contains the placeholder text 'Το μήνυμά σας'. Below it is a small note: 'Το πεδίο Your Message είναι απαραίτητο και πρέπει να περιέχει μία τιμή.' At the bottom of the form is a button labeled 'Υποβολή' (Submit).

Εικόνα 60

Κάθε φορά που γίνεται υποβολή χωρίς να είναι κάποιο από τα πεδία συμπληρωμένο μας εμφανίζεται μήνυμα μέσω της τεχνολογίας Ajax που μας ειδοποιεί για τις ενέργειες που πρέπει να γίνουν προκειμένου να είναι σε θέση το μήνυμά μας να αποσταλεί.

Ενδεικτικά αναφέρουμε πως το ajax πρόκειται για μια client side τεχνολογία στην οποία ο χρήστης μπορεί να στέλνει και να παίρνει από τον server μας πληροφορίες ασύγχρονα, χωρίς να παρεμβαίνει στην συμπεριφορά και τον τρόπο προβολής της εκάστοτε σελίδας.

Ας ριζούμε όμως και μια σύντομη ματιά στον κώδικά μας και να εξηγήσουμε τι συμβαίνει και με ποια σειρά. Αρχικά βλέπουμε την function **action_send**

```

1. public function action_send(){
2.
3.     $response = Response::forge();
4.     $val = Validation::forge();
5.     $response->title = "";
6.     if (Input::method() !== 'POST' or !Input::is_ajax())
7.     {
8.         $response->set_status(400);
9.         return $response;
10.    }
11.    $val->add('name','Your Name')
12.        ->add_rule('required')
13.        ->add_rule('max_length',255)
14.        ->add_rule('min_length',2);
15.
16.    $val->add('email','Your Email')
17.        ->add_rule('required')
18.        ->add_rule('max_length',255)
19.        ->add_rule('valid_email');
20.
21.    $val->add('message','Your Message')
22.        ->add_rule('required');
23.
24.    if($val->run())
25.    {
26.        try
27.        {
28.            Email::forge()
29.                ->to('cooxlee@hotmail.com')
30.                ->from(Input::post('email'),Input::post('name'))
31.                ->subject('Contact Form')
32.                ->body(Input::post('message'))
33.                ->send();
34.
35.            $response->body(json_encode(array(
36.                'status' => true,
37.
38.            )));
39.        }
40.
41.        catch (EmailSendingFailedException $e)
42.        {
43.            $response->set_status(500);
44.        }
45.    }
46.    else
47.    {
48.        $response->body(json_encode(array(
49.            'status' => false,
50.            'validation' => array(
51.                'name' => $val->error('name') ? $val->error('name')-
>get_message() : null,
52.                'email' => $val->error('email') ? $val->error('email')-
>get_message() : null,
53.                'message' => $val->error('message') ? $val->error('message')-
>get_message() : null ,
54.            )
55.        )));
56.    }
57.    return $response;
58. }

```

Αρχικά τρέχει το validation για να βεβαιωθούμε ότι όλα τα fields της φόρμας μας έχουν συμπληρωθεί σωστά. Εάν όντως όλες οι πληροφορίες είναι σωστά περασμένες στέλνουμε ένα json response με status success. Αν για οποιονδήποτε λόγο ενώ οι πληροφορίες είναι σωστά περασμένες και δεν φτάσουν στο script μας προσπαθούμε να κάνουμε catch το exception μας, αν αυτό δεν γίνει εφικτό επιστρέφουμε internal server error 500, διαφορετικά θα πρέπει να αναλάβει δράση η κλάση validation να ενημερώσει τον χρήστη με μήνυμα στα input fields για τα λάθη που έχει κάνει.

```
1. Contact.js file
$(function(){
2.
3.   $('form#contact').submit(function(e){
4.
5.     e.preventDefault();
6.
7.     $.post(this.action, $(this).serialize(), function(data){
8.
9.
10.      if(data.status === true)
11.      {
12.        alert("Σωστά!");
13.      }
14.      else
15.      {
16.
17.        $('.validation').text('');
18.        $('.color-red').text('');
19.
20.
21.        if (data.validation.name){
22.          $('#val-name').text(data.validation.name);
23.          $('#red-name').text(' *');
24.        }
25.        if (data.validation.email){
26.          $('#val-email').text(data.validation.email);
27.          $('#red-email').text(' *');
28.        }
29.        if (data.validation.message){
30.          $('#val-message').text(data.validation.message);
31.          $('#red-message').text(' *');
32.        }
33.      }
34.    }, 'json').error(function(){
35.
36.      alert('error');
37.
38.    });
39.
40.  });
41.
42. });
```

Όνομα *

Το όνομα σας

Το πεδίο Your Name είναι απαραίτητο και πρέπει να περιέχει μία τιμή.

Email *

Το email σας

Το πεδίο Your Email είναι απαραίτητο και πρέπει να περιέχει μία τιμή.

Μήνυμα *

Το μήνυμά σας

Το πεδίο Your Message είναι απαραίτητο και πρέπει να περιέχει μία τιμή.

Υποβολή

Εικόνα 61

Όπως βλέπουμε και στην παραπάνω εικόνα τα πεδία της φόρμας που δεν συμπληρώθηκαν έχουν ένα τύπου σχόλιο από κάτω που μας ενημερώνει ότι είναι απαραίτητο να συμπληρωθούν. Η λειτουργία αυτή υποστηρίζεται και στην Αγγλική γλώσσα με ίδιου τύπου comments.

©2014 Δεδομένα χάρτη Google 200 m

Leave a comment

Name *

The field Your Name is required and must contain a value.

Email *

The field Your Email is required and must contain a value.

Message *

The field Your Message is required and must contain a value.

Εικόνα 62

Όπως είναι λογικό για να υποβάλουμε το μήνυμά μας πατάμε στο κουμπί υποβολή και το μήνυμά μας θα σταλεί εφόσον πληρούνται όλες οι προδιαγραφές.

5. Αποτελέσματα

Μετά την ολοκλήρωση της πτυχιακής εργασίας καταφέραμε να βελτιωθούμε σε προσωπικό επίπεδο όσον αφορά την επέκταση των γνώσεων μας πάνω στο web development και να αποκτήσουμε μια πιο σφαιρική άποψη στον τομέα αυτό.

Με αυτή την σελίδα συμβάλουμε στην βελτίωση του τουριστικού επιπέδου της περιοχής της Ελασσόνας προσφέροντας περαιτέρω πληροφορίες και υπηρεσίες για τους ενδιαφερόμενους που θα ήθελαν να την επισκεφτούν.

5.1 Συμπεράσματα

Μέσα σε μια κοινωνία η οποία εξελίσσεται ραγδαία η εξοικείωση με τα διαδικτυακά μέσα κρίνεται αναγκαία. Συγκεκριμένα στον τομέα του τουρισμού, η χρήση μιας ιστοσελίδας καθιστά εύκολη και εύχρηστη την πρόσβαση και απόκτηση πληροφοριών από κάποιον ενδιαφερόμενο.

Μέσα από την ανάγκη της εξέλιξης όχι μόνο πετύχαμε τον στόχο μας να προωθήσουμε έναν όχι και τόσο διάσημο προορισμό αλλά καταφέραμε να διευρύνουμε τις γνώσεις μας χρησιμοποιώντας ένα από τα πιο σύγχρονα μέσα του web development, το fuel php framework.

5.2 Μελλοντική χρήση

Μια μελλοντική χρήση της πτυχιακής εργασίας αυτή θα ήταν η δημιουργία μιας εφαρμογής για λειτουργικά android. Προφανώς ανάλογα με τις εκάστοτε ανάγκες υπάρχει χώρος και διάθεση για βελτίωση και επέκταση διάφορων εφαρμογών για την ακόμα καλύτερη εξυπηρέτηση του κύριου στόχου.

Βιβλιογραφία

Ηλεκτρονική βιβλιογραφία

- [1] <http://code.tutsplus.com/tutorials/build-an-admin-panel-with-the-fuel-php-framework--net-23186>
- [2] <http://code.tutsplus.com/tutorials/getting-started-with-the-fuel-php-framework--net-21334>
- [3] <http://ucf.github.io/fuelphp-crash-course/>
- [4] <http://delab.csd.auth.gr/courses/dpmsdb/fd-nf.ppt>
- [5] <http://fuelphp.com/docs/installation/instructions.html>
- [6] <http://stackoverflow.com/>
- [7] http://www.phpbuilder.com/columns/FuelPHP_Gilmore_08-25-2011.php3
- [8] <http://php.net/docs.php>

Παράρτημα

Controller reservation

```
1. <?php
2.
3. class Controller_Reservations extends Controller_Base
4. {
5.     public $template = 'layouts/full_width';
6.
7.     public function before(){
8.
9.         parent::before();
10.
11.         //
12.         //Εάν ο χρήστης είναι συνδεδεμένος και έχει κάνει κράτηση
13.         //για ένα δωμάτιο, τότε οφείλουμε να ελέγξουμε αν ο
14.         //χρήστης έχει κάνει κράτηση για ένα αυτοκίνητο.
15.         //Αν όχι, τότε οφείλουμε να τον ενημερώσουμε να
16.         //κάνει κράτηση για ένα αυτοκίνητο.
17.
18.         if(Input::post('room'))
19.         {
20.             Session::set('room', Input::post('room'));
21.         }
22.         if(Input::post('vehicle'))
23.         {
24.             Session::set('rental', array('id' => Input::post('vehicle')));
25.         }
26.         $method = Request::active()->uri->segment(2);
```

```

22.     //επιχειρηματίας μπορεί να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση
    ενοίκιο ή να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση σε άλλο δωμάτιο
    ενοίκιο ή να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση σε άλλο δωμάτιο
23.     if ($method == 'account' && (is_null(Session::get('room')) && is_null(Se
    ssion::get('vehicle.id'))))
24.     {
25.         Response::redirect(Session::get('previous_url'), 'refresh');
26.     }
27.     }
28.     //επιχειρηματίας μπορεί να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση
    reservations/details (επιχειρηματίας μπορεί να κάνει κράτηση σε δωμάτιο που έχει
    ήδη κάνει κράτηση σε άλλο δωμάτιο)
29.     if(Input::post('allcheck'))
30.     {
31.         Session::set('allcheck', Input::post('allcheck'));
32.     }
33.     }
34.     //επιχειρηματίας μπορεί να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση
    % ενοίκιο ή να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση σε άλλο
    δωμάτιο ή να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση σε άλλο
    δωμάτιο ή να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει κράτηση σε άλλο
    δωμάτιο
35.     if ($method == 'invoice' && is_null(Session::get('allcheck')))
36.     {
37.         Response::redirect('reservations/details', 'refresh');
38.     }
39.     elseif ($method != 'invoice')
40.     {
41.         Session::delete('allcheck');
42.     }
43.     }
44.     }
45.     }
46.     }
47.     public function action_index()
48.     {
49.     }
50.     $view = View::forge('reservation/index');
51.     }
52.     if (Input::method() === 'POST')
53.     {
54.         //επιχειρηματίας μπορεί να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει
    κράτηση σε άλλο δωμάτιο ή να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει
    κράτηση σε άλλο δωμάτιο ή να κάνει κράτηση σε δωμάτιο που έχει ήδη κάνει
    κράτηση σε άλλο δωμάτιο
55.         $res = new ReserveQueries;
56.         $names = $res->get_table_names('hotels');
57.         $today = date("Y-m-d");
58.     }
59.     }
60.     $val = Validation::forge();
61.     }
62.     $val->add('start', 'Check In')
63.         ->add_rule('required')
64.         ->add_rule('valid_date')
65.         ->add_rule('numeric_between', $today , Input::post('finish'));
66.     }
67.     $val->add('finish', 'Check Out')
68.         ->add_rule('required_with', 'start')
69.         ->add_rule('valid_date');
70.     }
71.     }
72.     $val->add('hotel', 'Hotel')

```

```

73.         ->add_rule('match_value',$names);
74.
75.
76.     if ($val->run())
77.     {
78.
79.         //Οἱ εἰσῆχοντες εἰσὶν ἰσῆχοντες
80.         $checkin = Input::post('start');
81.         $checkout = Input::post('finish');
82.         $beds = Input::post('beds');
83.         $guests = Input::post('persons');
84.         $price = Input::post('price');
85.         $hotelName = Input::post('hotel');
86.
87.         Session::set('dates',array($checkin,$checkout));
88.
89.         if (!is_null(Session::get('dates')) )
90.         {
91.             //Ἐὰν οὐκ ἔσῃ ἡμεῖς ἐν τῇ σессии.
92.             if ($hotelName != "") { Session::set('hotel', $hotelName); }
93.             else { Session::delete('hotel'); }
94.             if ($guests != "-") { Session::set('guests', $guests); }
95.             else { Session::delete('guests'); }
96.             if ($price != 0) { Session::set('price', $price); }
97.             else { Session::delete('price'); }
98.
99.             //Ἐὰν οὐκ ἔσῃ ἡμεῖς ἐν τῇ σессии. Reserve::search_hotel()
100.            $hotel = Model_Hotel::query();
101.
102.            //function οἱ εἰσῆχοντες εἰσὶν οἱ εἰσῆχοντες Reserve εἰς τὴν εἰσῆχοντες εἰσὶν οἱ εἰσῆχοντες.
103.            $hotel = $res->search_hotel($hotel);
104.        }
105.        //Οἱ οἱ εἰσῆχοντες οἱ εἰσῆχοντες
106.        $view->hotels = $hotel->get();
107.
108.    }
109.    else
110.    {
111.        Session::set_flash('error', $val->error());
112.    }
113.
114.    }
115.
116.    $this->template->title = 'Hotels';
117.    $this->template->content = $view;
118.    }
119.
120.    public function action_account(){
121.
122.        $view = View::forge('reservation/customers/create');
123.
124.        If (Auth::check())
125.        {
126.            $id_info = Auth::get_user_id();
127.            $user = Model_User::query()
128.                ->where('id' , $id_info[1])
129.                ->get_one();
130.            $view->profile = Auth::get_profile_fields('profile_fields');

```

```

131.         $view->user = $user;
132.
133.     }
134.
135.     $this->template->title = 'Account Settings';
136.     $this->template->content = $view;
137. }
138.
139. //validation ajax (account.js). redirect 0 0
    reservations/details
140.     public function action_val_account(){
141.
142.         $response = Response::forge();
143.         $response->title = "";
144.
145.         $val = Validation::forge();
146.         $val->add_callable(new MyRules());
147.
148.         if (Input::method() !== 'POST' or !Input::is_ajax())
149.         {
150.             $response->set_status(400);
151.             return $response;
152.         }
153.         if (!Auth::check())
154.         {
155.
156.             $val->add('username', 'username')
157.                 ->add_rule('required')
158.                 ->add_rule('max_length', 255)
159.                 ->add_rule('min_length', 2)
160.                 ->add_rule('unique', 'users.username');
161.             $val->add('email', 'Email')
162.                 ->add_rule('required')
163.                 ->add_rule('max_length', 255)
164.                 ->add_rule('valid_email');
165.
166.             $val->add('fname', 'First name')
167.                 ->add_rule('required')
168.                 ->add_rule('max_length', 255)
169.                 ->add_rule('min_length', 2);
170.
171.             $val->add('lname', 'Last name')
172.                 ->add_rule('required')
173.                 ->add_rule('max_length', 255)
174.                 ->add_rule('min_length', 2);
175.
176.             $val->add('password1', 'Password')
177.                 ->add_rule('required');
178.
179.             $val->add('password2', 'Repeat Password')
180.                 ->add_rule('required')
181.                 ->add_rule('required_with', 'password1')
182.                 ->add_rule('match_field', 'password1');
183.         }
184.
185.         $val->add('terms', 'Terms and Conditions')
186.             ->add_rule('required');
187.
188.         if($val->run())

```

```

189.     {
190.         $auth = Auth::instance();
191.         Session::set('profile',array(
192.             'username' => Input::post('username'),
193.             'fname'    => Input::post('fname'),
194.             'lname'    => Input::post('lname'),
195.             'email'    => Input::post('email'),
196.             'address'  => Input::post('address')
197.         ));
198.         if ( !Auth::check())
199.         {
200.             try
201.             {
202.                 $create_process = Auth::create_user(
203.                     Input::post('username'),
204.                     Input::post('password1'),
205.                     Input::post('email'),
206.                     1,
207.                     array(
208.                         'profile_fields' => array(
209.                             'fname' => Input::post('fname'),
210.                             'lname' => Input::post('lname'),
211.                         ))
212.                 );
213.
214.                 if ($create_process)
215.                 {
216.
217.                     $auth-
218. >login(Input::post('username'), Input::post('password1'));
219.                     $response->body(json_encode(array(
220.                         'status' => true,
221.                     )));
222.                 }
223.
224.                 catch (Exception $e)
225.                 {
226.                     $error = $e->getMessage();
227.                 }
228.             }
229.             elseif (Auth::check())
230.             {
231.                 try
232.                 {
233.                     $update_process = Auth::update_user(
234.                         array(
235.                             'email' => Input::post('email'),
236.                             'profile_fields' => array(
237.                                 'fname' => Input::post('fname'),
238.                                 'lname' => Input::post('lname'),
239.                             )
240.                         ));
241.
242.                     if ($update_process)
243.                     {
244.                         //$auth-
245. >login(Input::post('username'), Input::post('password1'));

```

```

246.         $response->body(json_encode(array(
247.             'status' => true,
248.         )));
249.     }
250. }
251. catch (Exception $e)
252. {
253.     $error = $e->getMessage();
254. }
255. }
256. }
257. else
258. {
259.     $response->body(json_encode(
260.         array(
261.             'status' => false,
262.             'validation' => array(
263.                 'username' => $val->error('username') ? $val-
264. >error('username')->get_message() : null,
265.                 'fname' => $val->error('fname') ? $val->error('fname')-
266. >get_message() : null,
267.                 'lname' => $val->error('lname') ? $val->error('lname')-
268. >get_message() : null,
269.                 'email' => $val->error('email') ? $val->error('email')-
270. >get_message() : null,
271.                 'adress' => $val->error('adress') ? $val-
272. >error('adress')->get_message() : null,
273.                 'terms' => $val->error('terms') ? $val->error('terms')-
274. >get_message() : null ,
275.                 'password1' => $val->error('password1') ? $val-
276. >error('password1')->get_message() : null,
277.                 'password2' => $val->error('password2') ? $val-
278. >error('password2')->get_message() : null,
279.             )
280.         )
281.     ));
282. }
283. return $response;
284. }
285.
286. public function action_details(){
287.
288.     $view = View::forge('reservation/customers/details');
289.     if(Session::get('room'))
290.     {
291.         $room_id = Session::get('room');
292.         $room = Model_Room::query()
293.             ->where('id' , $room_id)
294.             ->related('hotels')->get_one();
295.         $view->room = $room;
296.     }
297.     if(Session::get('rental.id'))
298.     {
299.         $vehicle_id = Session::get('rental.id');
300.         $vehicle = Model_Vehicle::query()
301.             ->where('id' , $vehicle_id)
302.             ->related('rentals')->get_one();
303.         $view->vehicle = $vehicle;

```

```

297.     }
298.
299.     $this->template->title = 'Booking Details';
300.     $this->template->content = $view;
301.     }
302.     public function action_invoice(){
303.
304.         $view = View::forge('reservation/customers/invoice');
305.         $res = new ReserveQueries;
306.
307.         if(!is_null(Session::get('room')))
308.         {
309.             $room_id = Session::get('room');
310.             $room = Model_Room::query()
311.                 ->where('id' , $room_id)
312.                 ->related('hotels')->get_one();
313.             $code = $res-
314. >get_unique(Str::random('distinct' , 6) , 'bookings' , 'bookingCode');
315.             $view->room = $room;
316.         }
317.         if(!is_null(Session::get('rental.id')))
318.         {
319.             $vehicle_id = Session::get('rental.id');
320.             $vehicle = Model_Vehicle::query()
321.                 ->where('id' , $vehicle_id)
322.                 ->related('rentals')->get_one();
323.             $vehcode = $res-
324. >get_unique(Str::random('distinct' , 6) , 'carbookings' , 'bookingCode');
325.             $view->vehicle = $vehicle;
326.         }
327.         //unique bookingCode.
328.
329.         if (Input::method() === 'POST' && Input::post('finish') && Auth::ch
330. eck())
331.         {
332.             if (isset($room) && isset($vehicle)):
333.                 // typical transaction code flow
334.                 try
335.                 {
336.                     DB::start_transaction();
337.
338.                     // some query ...
339.                     $query = DB::insert('bookings');
340.                     $query->set(array(
341.                         'bookingCode' => $code,
342.                         'guests' => '1',
343.                         'checkIn' => Session::get('dates.0'),
344.                         'checkOut' => Session::get('dates.1'),
345.                         'room_id' => Session::get('room'),
346.                         'user_id' => $this->current_user->id,
347.                         'created_at' => Date::time()->get_timestamp(),
348.                     ));
349.                     $query->execute();
350.
351.                     $query2 = DB::insert('carbookings');
352.                     $query2->set(array(

```



```

353.         'bookingCode' => $vehcode,
354.         'checkIn' => Session::get('dates.0'),
355.         'checkOut' => Session::get('dates.1'),
356.         'vehicle_id' => Session::get('rental.id'),
357.         'user_id' => $this->current_user->id,
358.         'created_at' =>Date::time()->get_timestamp(),
359.     ));
360.     $query2->execute();
361.     DB::commit_transaction();
362.
363.     Session::delete('profile');
364.     Session::delete('dates');
365.     Session::delete('room');
366.     Session::delete('allcheck');
367.     Session::delete('hotel');
368.     Session::delete('rental');
369.
370.     Session::set_flash('success', e('Added Booking'));
371.
372.     Response::redirect('users/profile');
373.
374.     // return query result
375. }
376. catch (Exception $e)
377. {
378.     // rollback pending transactional queries
379.     DB::rollback_transaction();
380.     Session::delete('profile');
381.     Session::delete('dates');
382.     Session::delete('room');
383.     Session::delete('allcheck');
384.     Session::delete('hotel');
385.     Session::delete('rental');
386.
387.     Response::redirect('users/profile');
388.
389.     Session::set_flash('error', e('Could not make Booking'));
390.
391. }
392.
393.
394. else:
395.
396.     if(isset($room))
397.     {
398.         try{
399.             $query = DB::insert('bookings');
400.             $query->set(array(
401.                 'bookingCode' => $code,
402.                 'guests' => '1',
403.                 'checkIn' => Session::get('dates.0'),
404.                 'checkOut' => Session::get('dates.1'),
405.                 'room_id' => Session::get('room'),
406.                 'user_id' => $this->current_user->id,
407.                 'created_at' =>Date::time()->get_timestamp(),
408.             ));
409.             $query->execute();
410.
411.             Session::delete('profile');

```

```

412.         Session::delete('dates');
413.         Session::delete('room');
414.         Session::delete('allcheck');
415.         Session::delete('hotel');
416.         Session::delete('rental');
417.
418.         Session::set_flash('success', e('Added Booking'));
419.         Response::redirect('users/profile');
420.     }
421.     catch (Exception $e)
422.     {
423.         Session::delete('profile');
424.         Session::delete('dates');
425.         Session::delete('room');
426.         Session::delete('allcheck');
427.         Session::delete('hotel');
428.         Session::delete('rental');
429.         Session::set_flash('error', e('Could not make Booking'));
430.     }
431. }
432. if(isset($vehicle))
433. {
434.
435.     try
436.     {
437.         $query2 = DB::insert('carbookings');
438.         $query2->set(array(
439.             'bookingCode' => $vehcode,
440.             'checkIn' => Session::get('dates.0'),
441.             'checkOut' => Session::get('dates.1'),
442.             'vehicle_id' => Session::get('rental.id'),
443.             'user_id' => $this->current_user->id,
444.             'created_at' =>Date::time()->get_timestamp(),
445.         ));
446.         $query2->execute();
447.         Session::delete('profile');
448.         Session::delete('dates');
449.         Session::delete('room');
450.         Session::delete('allcheck');
451.         Session::delete('hotel');
452.         Session::delete('rental');
453.
454.         Session::set_flash('success', e('Added Booking'));
455.         Response::redirect('users/profile');
456.     }
457.     catch (Exception $e)
458.     {
459.         Session::delete('profile');
460.         Session::delete('dates');
461.         Session::delete('room');
462.         Session::delete('allcheck');
463.         Session::delete('hotel');
464.         Session::delete('rental');
465.         Session::set_flash('error', e('Could not make Booking'));
466.     }
467.
468. }
469.
470. endif;

```

```

471.
472.     }
473.
474.     $this->template->title = 'Invoice';
475.     $this->template->content = $view;
476.     }
477.     public function action_delete()
478.     {
479.         if (Input::method() === 'POST')
480.         {
481.             $id = \Input::post('booking');
482.             if ($booking = Model_Booking::find($id))
483.             {
484.                 $booking->delete();
485.
486.                 Session::set_flash('success', e('Deleted booking #'.$id));
487.             }
488.
489.             else
490.             {
491.                 Session::set_flash('error', e('Could not delete booking #'.$id
492.             ));
493.             }
494.             Response::redirect('users/profile');
495.         }
496.     }
497.
498.     public function action_rental()
499.     {
500.
501.         $view = View::forge('reservation/rental');
502.
503.         if (Input::method() === 'POST')
504.         {
505.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
506.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
507.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
508.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
509.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
510.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
511.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
512.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
513.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
514.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
515.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
516.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
517.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
518.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
519.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
520.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
521.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
522.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
523.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
524.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
525.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
526.             // ΕΙΣΑΓΩΓΗ ΕΙΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

```

```

527.         if ($val->run())
528.         {
529.
530.             //Ορίστε τα input
531.             $checkin = Input::post('pickup');
532.             $checkout = Input::post('return');
533.             $category = Input::post('category');
534.             $seats = Input::post('seat');
535.             $price = Input::post('price');
536.             $rentalName = Input::post('name');
537.
538.             Session::set('dates', array($checkin,$checkout));
539.
540.             if (!is_null(Session::get('dates')) )
541.             {
542.                 //Εάν ο χρήστης έχει ήδη κάνει κράτηση
543.                 if ($rentalName != "") { Session::set('rental', array('name
' => $rentalName)); }
544.                 else { Session::delete('rental.name'); }
545.                 if ($seats != "-
") { Session::set('rental', array('seats' => $seats)); }
546.                 else { Session::delete('rental.seats'); }
547.                 if ($price != 0) { Session::set('rental', array('price' =>
$price)); }
548.                 else { Session::delete('rental.price'); }
549.
550.                 //Εάν ο χρήστης έχει κάνει κράτηση
551.                 $rental = Model_Rental::query();
552.
553.                 //function που επιστρέφει τα αποτελέσματα της
κράτησης
554.                 $rental = $res->search_rental($rental);
555.             }
556.             //Ορίστε την query
557.             $view->rentals = $rental->get();
558.
559.         }
560.         else
561.         {
562.             Session::set_flash('error', $val->error());
563.         }
564.
565.     }
566.
567.     $this->template->title = 'Rental Providers';
568.     $this->template->content = $view;
569. }
570.
571.
572.
573.
574.     // echo "<pre>";
575.     // var_dump($view->room);
576.     // echo "</pre>";
577.     // die;

```

Class ReserveQueries

```
1. <?php
2.
3. class ReserveQueries {
4.
5.     /**
6.      * @return array
7.      */
8.     public function get_table_names($table) {
9.
10.         $query = DB::select('name')->from($table)->execute()->as_array('name');
11.         $names = array();
12.         foreach ($query as $key => $value)
13.         {
14.             $names[$key] = $value['name'];
15.         }
16.         return $names;
17.     }
18.
19.     /**
20.      * @return Model_Hotelcomment
21.      */
22.     public function get_best_comment($id) {
23.
24.         $best_comment = Model_Hotelcomment::query()
25.             ->where('hotel_id', $id)
26.             ->order_by('rating', 'DESC')
27.             ->order_by('created_at', 'DESC')
28.             ->related('users')
29.             ->get_one();
30.
31.         return $best_comment;
32.     }
33.
34.     /**
35.      * @return Model_Hotel
36.      */
37.     public function search_hotel($hotel){
38.
39.         $checkin = Session::get('dates.0');
40.         $checkout = Session::get('dates.1');
41.
42.         if(!is_null(Session::get('hotel'))){
43.             $hotel = $hotel
44.                 ->where('name', Session::get('hotel'));
45.         }
46.
47.         $hotel = $hotel
48.             ->related('rooms')
49.             ->related('rooms.bookings')
50.             ->and_where_open();
51.         $hotel = $hotel
```

```

46.         ->or_where_open()
47.         ->where('rooms.bookings.checkIn', '>', $checkin)
48.         ->where('rooms.bookings.checkIn', '>=', $checkout)
49.         ->or_where_close()
50.         ->or_where_open()
51.         ->where('rooms.bookings.checkOut', '<=', $checkin)
52.         ->where('rooms.bookings.checkOut', '<', $checkout)
53.         ->or_where_close()
54.         ->or_where('rooms.bookings.id', '=', NULL)
55.     ->and_where_close();
56.
57.     if(!is_null(Session::get('guests')))
58.     {
59.         $hotel = $hotel-
>where('rooms.beds', '>=', Session::get('guests'));
60.     }
61.     else
62.     {
63.         Session::delete('guests');
64.     }
65.
66.     if(!is_null(Session::get('price')))
67.     {
68.         $hotel = $hotel-
>where('rooms.pricepernight', '<=', Session::get('price'));
69.     }
70.     else
71.     {
72.         Session::delete('price');
73.     }
74.     //ε±0 εΝ0 εΝ0 εΕε·ε%εΝ0 0 ε0 εΗεΝε±εΙ0 εΗε²εΝ.
75.     $hotel = $hotel->where('rooms.id', '!=', NULL)-
>order_by('rooms.pricepernight', 'ASC');
76.
77.     return $hotel;
78. }
79.
80. public function search_rental($rental){
81.
82.     $checkin = Session::get('dates.0');
83.     $checkout = Session::get('dates.1');
84.
85.     if(!is_null(Session::get('rental.name')))
86.     {
87.         $rental = $rental
88.             ->where('name', Session::get('rental.name'));
89.     }
90.
91.     $rental = $rental
92.         ->related('vehicles')
93.         ->related('vehicles.carbookings')
94.         ->and_where_open();
95.     $rental = $rental
96.         ->or_where_open()
97.         ->where('vehicles.carbookings.checkIn', '>', $checkin)
98.         ->where('vehicles.carbookings.checkIn', '>=', $checkout)
99.         ->or_where_close()
100.        ->or_where_open()

```



```

5.
6. // \Session::set('previous_url', \Input::referrer());
7. \Event::register('shutdown', function() {
8.     if (Uri::base()===Uri::base(Input::referrer())){
9.         \Session::set('previous_url', \Input::referrer());
10.    }
11.});

```

Controller_Users

```

1. <?php
2.
3. class Controller_Users extends Controller_Base
4. {
5.     public $template = 'layouts/full_width';
6.
7.     public function action_login()
8.     {
9.         // Already logged in
10.        Auth::check() and Response::redirect_back('/', 'refresh');
11.
12.        $val = Validation::forge();
13.
14.        if (Input::method() == 'POST')
15.        {
16.
17.            $val->add('email', 'Email or Username')
18.                ->add_rule('required');
19.
20.            $val->add('password', 'Password')
21.                ->add_rule('required');
22.
23.            if ($val->run())
24.            {
25.                $auth = Auth::instance();
26.
27.                // check the credentials. This assumes that you have the pr
28.                evious table created
29.                if (Auth::check() or $auth-
30.                >login(Input::post('email'), Input::post('password')))
31.                {
32.                    // credentials ok, go right in
33.                    if (Config::get('auth.driver', 'Simpleauth') == 'Ormaut
34.                    h')

```

```

32.         {
33.             $current_user = Model\Auth_User::find_by_username(A
    uth::get_screen_name());
34.         }
35.         else
36.         {
37.             $current_user = Model_User::find_by_username(Auth::
    get_screen_name());
38.         }
39.         //Session::set_flash('success', e('Welcome, '.$current_
    user->username));
40.         Response::redirect(Session::get('previous_url'), 'refre
    sh');
41.     }
42.     else
43.     {
44.         $this->template->set_global('login_error', 'Fail');
45.     }
46. }
47. }
48.
49. $this->template->title = 'ΕΞΕΛΕΞΕΤΕ ΣΕΝΑ';
50. $this->template->subtitle = '';
51. $this->template-
    >content = View::forge('users/login', array('val' => $val), false);
52. }
53.
54. public function action_logout()
55. {
56.     Auth::logout();
57.     Response::redirect_back('/', 'refresh');
58. }
59.
60. public function action_val_registration()
61. {
62.     $response = Response::forge();
63.     $val = Validation::forge();
64.     $val->add_callable(new MyRules());
65.     if (Input::method() !== 'POST' or !Input::is_ajax())
66.     {
67.         $response->set_status(400);
68.         return $response;
69.     }
70.
71.
72.     $val->add('username', 'nickname')
73.         ->add_rule('required')
74.         ->add_rule('max_length', 255)
75.         ->add_rule('min_length', 2)
76.         ->add_rule('unique', 'users.username');
77.
78.     $val->add('email', 'Email')
79.         ->add_rule('required')
80.         ->add_rule('max_length', 255)
81.         ->add_rule('valid_email')
82.         ->add_rule('unique', 'users.email');
83.
84.     $val->add('password', 'Password')
85.         ->add_rule('required');

```

```

86.
87.     $val->add('password2', 'Repeat Password')
88.         ->add_rule('required')
89.         ->add_rule('required_with','password')
90.         ->add_rule('match_field','password', $strict = false);
91.
92.     $val->add('fname','name')
93.         ->add_rule('required')
94.         ->add_rule('max_length',255)
95.         ->add_rule('min_length',2);
96.
97.     $val->add('lname','lastname')
98.         ->add_rule('required')
99.         ->add_rule('max_length',255)
100.        ->add_rule('min_length',2);
101.
102.        if($val->run())
103.        {
104.            try
105.            {
106.                $create_process = Auth::create_user(Input::post('
username'), Input::post('password'), Input::post('email'), Input::post('fna
me'), Input::post('lname'));
107.                if ($create_process)
108.                {
109.                    $response->body(json_encode(array(
110.                        'status' => true,
111.                    )));
112.                }
113.            }
114.            catch (Exception $e)
115.            {
116.                $error = $e->getMessage();
117.            }
118.        }
119.        else
120.        {
121.            $response->body(json_encode(array(
122.                'status' => false,
123.                'validation' => array(
124.                    'username' => $val->error('username') ? $val-
>error('username')->get_message() : null,
125.                    'email' => $val->error('email') ? $val-
>error('email')->get_message() : null,
126.                    'password' => $val->error('password') ? $val-
>error('password')->get_message() : null,
127.                    'password2' => $val->error('password2') ? $val-
>error('password2')->get_message() : null,
128.                    'fname' => $val->error('fname') ? $val-
>error('fname')->get_message() : null,
129.                    'lname' => $val->error('lname') ? $val-
>error('lname')->get_message() : null,
130.                )
131.            )));
132.        }
133.
134.        return $response;
135.
136.

```

```

137.
138.     }
139.     public function action_register()
140.     {
141.         $view = View::forge('users/register');
142.         $this->template->title = 'Είσοδος';
143.         $this->template->subtitle = '';
144.         $this->template->content = $view;
145.     }
146.
147.     public function action_profile()
148.     {
149.         if(Auth::check())
150.         {
151.
152.             $view = View::forge('users/index');
153.             $id_info = Auth::get_user_id();
154.
155.             $user = Model_User::query()
156.                 ->where('id' , $id_info[1])
157.                 ->related('bookings')
158.                 ->related('bookings.rooms')
159.                 ->related('bookings.rooms.hotels')
160.                 ->get_one();
161.             //Είσοδος
162.             $view->bookings_future = Model_Booking::query()
163.                 ->where('user_id', $id_info[1])
164.                 ->where('checkIn', '>' ,Date::time()-
165.                 >format('mysql'))
166.                 ->get();
167.             $view->bookings_past = Model_Booking::query()
168.                 ->where('user_id', $id_info[1])
169.                 ->where('checkOut', '<' ,Date::time()-
170.                 >format('mysql'))
171.                 ->get();
172.             $view->bookings_present = Model_Booking::query()
173.                 ->where('user_id', $id_info[1])
174.                 ->where('checkOut', '>' ,Date::time()-
175.                 >format('mysql'))
176.                 ->where('checkIn', '<=' ,Date::time()-
177.                 >format('mysql'))
178.                 ->get();
179.             //Είσοδος
180.             $view->carbookings_future = Model_Carbooking::query()
181.                 ->where('user_id', $id_info[1])
182.                 ->where('checkIn', '>' ,Date::time()-
183.                 >format('mysql'))
184.                 ->get();
185.             $view->carbookings_past = Model_Carbooking::query()
186.                 ->where('user_id', $id_info[1])
187.                 ->where('checkOut', '<' ,Date::time()-
188.                 >format('mysql'))
189.                 ->get();
190.             $view->carbookings_present = Model_Carbooking::query()

```

```

190.             ->where('user_id', $id_info[1])
191.             ->where('checkOut', '>' ,Date::time()-
    >format('mysql'))
192.             ->where('checkIn', '<=' ,Date::time()-
    >format('mysql'))
193.             ->get();
194.
195.             // echo "<pre>";
196.             // var_dump(Date::time()->format('mysql'));
197.             // echo "</pre>";
198.             // die;
199.
200.             $view->user = $user;
201.
202.
203.             $options['image'] = urlencode('monsterid');
204.             $options['size'] = 872;
205.             $gravatar = \Gravatar::forge($this->current_user-
    >email,$options);
206.             $view->gravatar = $gravatar;
207.
208.             $this->template->title = $this->current_user->username;
209.             $this->template->subtitle = '';
210.             $this->template->content = $view;
211.
212.         }
213.
214.         else Response::redirect('/');
215.     }
216.
217.     public function action_comments()
218.     {
219.         if(Auth::check())
220.         {
221.             $view = View::forge('users/comments');
222.             $id_info = Auth::get_user_id();
223.             $user = Model_User::query()
224.                 ->where('id' , $id_info[1])
225.                 ->related('hotelcomments')
226.                 ->related('hotelcomments.hotels')
227.                 ->get_one();
228.
229.             // echo "<pre>";
230.             // var_dump($user);
231.             // echo "</pre>";
232.             // die;
233.
234.             $view->user = $user;
235.
236.
237.             $options['image'] = urlencode('monsterid');
238.             $options['size'] = 872;
239.             $gravatar = \Gravatar::forge($this->current_user-
    >email,$options);
240.             $view->gravatar = $gravatar;
241.
242.             $this->template->title = $this->current_user->username;
243.             $this->template->subtitle = '';
244.             $this->template->content = $view;

```

```

245.
246.     }
247.
248.     else Response::redirect('/');
249. }
250. public function action_settings()
251. {
252.     if(Auth::check())
253.     {
254.         $view = View::forge('users/settings');
255.         $id_info = Auth::get_user_id();
256.         $user = Model_User::query()
257.             ->where('id' , $id_info[1])
258.             ->get_one();
259.
260.         // echo "<pre>";
261.         // var_dump($user);
262.         // echo "</pre>";
263.         // die;
264.
265.         $view->user = $user;
266.
267.
268.         $options['image'] = urlencode('monsterid');
269.         $options['size'] = 872;
270.         $gravatar = \Gravatar::forge($this->current_user->
271. >email,$options);
272.         $view->gravatar = $gravatar;
273.
274.         $this->template->title = $this->current_user->username;
275.         $this->template->subtitle = '';
276.         $this->template->content = $view;
277.     }
278.
279.     else Response::redirect('/');
280. }
281.
282. }
283.
284.
285.
286.
287.
288.     // echo "<pre>";
289.     // var_dump($view->room);
290.     // echo "</pre>";
291.     // die;

```