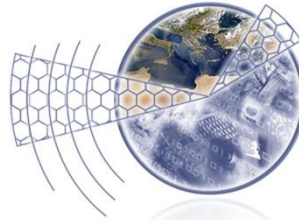




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & πολυμέσων**



Πτυχιακή εργασία

Τίτλος: Ηλεκτρονικό ινστιτούτο αισθητικής με χρήση MVC PHP Framework

Σταυγιανουδάκης Ιωάννης (ΑΜ:1226)

Επιβλέπων Καθηγήτρια: Λογκάκη Μαρία

Επιτροπή Αξιολόγησης: Βιδάκης Νικόλαος, Βαρδιάμπασης Δημήτριος

Ημερομηνία παρουσίασης:

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τους γονείς μου για την οικονομική, ψυχολογική και ηθική υποστήριξη που μου παρείχαν καθ' όλη τη διάρκεια της ακαδημαϊκής μου θητείας και ειδικότερα την τελευταία χρονιά που χρειάστηκε να εστιάσω ακόμα περισσότερο στην επίτευξη των στόχων μου.

Ευχαριστώ την καθηγήτρια μου, κύρια Λογκάκη Μαρία για την αποτελεσματική καθοδήγηση που μου παρείχε κατά την εκπόνηση της πτυχιακής μου εργασίας! Επίσης, εκτιμώ ιδιαίτερα την ελαστικότητα που έδειχνε και την προσαρμοστικότητά του κάθε φορά που άλλαζα πορεία.

Ευχαριστώ πολύ όλο το διδακτικό προσωπικό το Α.Τ.Ε.Ι. Ηρακλείου που με δίδαξε τόσα χρόνια και αποκόμισα την υπέροχη αυτή ακαδημαϊκή εμπειρία ολοκληρώνοντας το πρόγραμμα σπουδών της σχολής Τεχνολογικών εφαρμογών και του τμήματος *Εφαρμοσμένης Πληροφορικής και Πολυμέσων*.

Abstract

Online beauty salon using MVC (Model-View-Controller) open source PHP Framework.

This is a dynamic website that will set up an online beauty salon to provide visitors with information and beauty services. I used to implement one of the most popular Open Source PHP frameworks, CodeIgniter. It is based on architecture MVC (Model-View-Controller). This model helps us keep organized our code so as to be easier to understand and easily cultivated.

The Model is the piece which manages operations on a data base (e.g. CRUD functions), the View has to do with what appears to the user, is the pages of our application and the Controller monitors the user actions, identifies them and carry at the appropriate point in the code where we accomplished a specific function. It also receives data generated by Models and transfer the appropriate View.

The system will consist of a base where MySQL is used to store user information, products and services to be provided in the online store. When a visitor enters the home page of beauty salon will be able to browse for products where they can see information and to add a shopping cart to buy them in different ways of payment and to receive either home or the physical store of the beauty institute.

Also, if the users wants to be able to access all the pages for these actions, it will be a complete user authorization and authentication system. Logging in if you are a member or register if not, it will be possible to e-appointment, which the users could make an appointment for services provided by highly reactive manner, friendly and convenient for any discerning visitor.

There will be a management page where you can use to see new information and purchase history or services generally can manage their profile.

On the side of the owner (administrator), will be able to check orders or appointments details. Information for clients, ability to publish articles, news, adding new products and services, manage categories of existing products and services,

for the successful integration I used technologies and tools such as:

- MVC Framework (CodeIgniter)
- XHTML, CSS
- PHP5, MySQL • XML • AJAX (jQuery), Photoshop

Σύνοψη

Ηλεκτρονικό ινστιτούτο αισθητικής με την χρήση MVC Opensource PHP Framework.

Πρόκειται για μία δυναμική ιστοσελίδα στην οποία θα στηθεί ένα ηλεκτρονικό ινστιτούτο αισθητικής για την παροχή στους επισκέπτες πληροφορίες και υπηρεσίες ομορφιάς.

Θα χρησιμοποιηθεί για την υλοποίηση του ένα από τα πιο διάσημα Opensource PHP frameworks, το **CodeIgniter**. Είναι βασισμένο στην αρχιτεκτονική **MVC (Model-View-Controller)**. Το πρότυπο αυτό μας βοηθά να κρατάμε οργανωμένο τον κώδικά σας έτσι ώστε να είναι πιο κατανοητός και πιο εύκολα συντηρούμενος.

Το Model είναι το κομμάτι που διαχειρίζεται λειτουργίες πάνω σε δεδομένα της βάσης (πχ CRUD functions), το View έχει να κάνει με το τι φαίνεται στο χρήστη, δηλαδή οι σελίδες της εφαρμογής μας και το Controller ελέγχει ενέργειες του χρήστη, τις αναγνωρίζει και τις μεταφέρει στο κατάλληλο σημείο του κώδικα όπου θα επιτελεστεί μία συγκεκριμένη λειτουργία. Επίσης, δέχεται τα δεδομένα που προκύπτουν από τα Models και τα μεταφέρει στο κατάλληλο View.

Το σύστημα θα αποτελείται από μία βάση **MySQL**, όπου θα χρησιμοποιηθεί για να αποθηκεύσουμε στοιχεία των χρηστών, των προϊόντων και των υπηρεσιών που θα παρέχονται στο ηλεκτρονικό κατάστημα. Όταν ο επισκέπτης εισέλθει στην αρχική σελίδα του ινστιτούτου αισθητικής θα έχει τη δυνατότητα να περιηγηθεί σε προϊόντα, όπου μπορεί να δει πληροφορίες, και να τα προσθέσει σε ένα shopping cart έτσι ώστε να τα αγοράσει με διάφορους τρόπου πληρωμής και να τα παραλάβει είτε στο σπίτι του είτε από το φυσικό κατάστημα του ινστιτούτου.

Επίσης για να μπορέσει να έχει πρόσβαση σε όλες της σελίδες, και για παραπάνω ενέργειες, θα υπάρχει ένα πλήρες σύστημα χρήστη. Κάνοντας login αν είναι μέλος, register αν δεν είναι, θα υπάρχει δυνατότητα e-appointment, δηλαδή θα μπορεί να κλείσει ραντεβού για τις υπηρεσίες που παρέχονται, με άκρως διαδραστικό τρόπο, φιλικό και εύχρηστο για κάθε απαιτητικό επισκέπτη.

Θα υπάρχει σελίδα διαχείρισης του χρήστη όπου θα μπορεί να βλέπει νέα, πληροφορίες, ιστορικό αγορών ή υπηρεσιών και γενικότερα να μπορεί να διαχειριστεί το προφίλ του.

Από την μεριά του ιδιοκτήτη (administrator), θα μπορεί να βλέπει παραγγελίες ή ραντεβού που έχουν κλείσει πελάτες. Πληροφορίες για πελάτες, όπως και θα έχει την δυνατότητα να δημοσιεύει άρθρα, νέα, να προσθέτει καινούργια προϊόντα, υπηρεσίες ή να διαχειρίζεται είδη υπάρχοντα προϊόντα και υπηρεσίες. Για την επιτυχημένη ολοκλήρωση θα πρέπει να χρησιμοποιηθούν τεχνολογίες και εργαλεία όπως: MVC Framework (CodeIgniter), xHTML, CSS PHP5 ,MySQL XML AJAX (jQuery) Photoshop κ.α.

Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ.....	7
1.1	ΠΕΡΙΛΗΨΗ.....	7
1.2	ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΕΡΓΑΣΙΑΣ	7
2	ΤΕΧΝΟΛΟΓΙΕΣ & ΕΡΓΑΛΕΙΑ	8
2.1	ΔΥΝΑΜΙΚΗ ΙΣΤΟΣΕΛΙΔΑ ⁽¹⁾	8
2.2	ΤΙ ΕΙΝΑΙ PHP MYSQL ΚΑΙ ΤΙ ΡΗΡΜΥADMIN. ⁽²⁾	9
2.3	ΤΙ ΕΙΝΑΙ Ο ΑΡΑΧΕ ΚΑΙ Η XML. ⁽²⁾	10
2.4	ΗΛΕΚΤΡΟΝΙΚΟ ΚΑΤΑΣΤΗΜΑ (E-SHOP) ⁽¹⁾	11
2.5	ΤΙ ΕΙΝΑΙ ΤΑ PHP FRAMEWORKS ⁽²⁾	12
2.6	FIREBUG ⁽³⁾	13
3	ΤΟ CODEIGNITER	14
3.1	ΤΙ ΕΙΝΑΙ ΤΟ CODEIGNITER ⁽⁴⁾	14
3.2	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ CODEIGNITER ⁽⁴⁾	15
3.3	ΒΙΒΛΙΟΘΗΚΕΣ, HELPERS & FUNCTIONS ΤΟΥ CODEIGNITER ⁽⁴⁾	17
3.4	ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΚΑΙ MVC ⁽⁴⁾	19
3.5	ΠΑΡΑΔΕΙΓΜΑΤΑ ⁽⁴⁾	20
	ΠΑΡΑΔΕΙΓΜΑ 1: ΑΠΟΣΤΟΛΗ E-MAIL ΜΕ CODEIGNITER.....	20
	ΠΑΡΑΔΕΙΓΜΑ 2: ΣΥΜΠΙΕΣΗ ΑΡΧΕΙΩΝ ⁽⁴⁾	20
4	ΜΕΡΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	21
4.1	ΑΝΑΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ	21
4.2	ΑΠΑΙΤΗΣΕΙΣ ΕΦΑΡΜΟΓΗΣ	21
4.3	ΥΛΟΠΟΙΗΣΗ	22
4.3.1	ΕΓΚΑΤΑΣΤΑΣΗ WAMP ⁽⁸⁾	22
4.3.2	ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ CODEIGNITER ⁽⁴⁾	23
4.3.3	ECLIPSE PHP ⁽²⁾	25
4.4	Η ΔΟΜΗ ΤΩΝ ΑΡΧΕΙΩΝ ⁽⁴⁾	27
4.5	ΔΗΜΙΟΥΡΓΙΑ .HTACCESS & VIRTUAL HOSTS ⁽⁹⁾	28
4.6	ΟΙ ΠΙΝΑΚΕΣ ΤΗΣ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΜΑΣ	30
4.7	ΤΟ FRONTEND (ΒΑΣΙΚΗ ΣΕΛΙΔΑ ΕΠΙΣΚΕΠΤΩΝ ΚΑΙ ΜΕΛΩΝ)	31
4.7.1	Ο MAIN CONTROLLER ΤΟΥ ΙΝΣΤΙΤΟΥΤΟΥ ⁽⁴⁾	32
4.7.2	ΤΟ CUSTOM TEMPLATE ΤΟΥ FRONT END ΜΑΣ. ⁽¹⁰⁾	40
4.7.3	ΤΑ VIEWS ΜΑΣ ⁽⁷⁾	42

5	ΕΠΕΚΤΑΣΗ ΤΟΥ CODEIGNITER.....	46
5.1	ΤΟ BACKOFFICE (STAVGIAN CUSTOM MANAGEMENT SYSTEM)	46
5.2	ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ STAVGIANCMS	46
5.3	SCREENSHOTS.....	47
5.4	ΈΝΑ ΔΕΙΓΜΑ ΑΠΟ ΤΑ ΒΑΣΙΚΑ ΜΟΝΤΕΛΑ ΜΑΣ. (MODELS) ⁽¹¹⁾	49
5.4.1	BEAUTY_MODEL.PHP (ΓΕΝΙΚΟ ΜΟΝΤΕΛΟ ΤΟΥ ΙΝΣΤΙΤΟΥΤΟΥ ΑΙΣΘΗΤΙΚΗΣ)	49
5.4.2	PRODUCTS_MODEL(ΓΙΑ ΠΡΟΪΟΝΤΑ ΚΑΙ ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΪΟΝΤΩΝ) ⁽¹¹⁾	55
5.5	CUSTOM ΒΙΒΛΙΟΘΗΚΕΣ (LIBRARIES) ⁽¹¹⁾	57
5.5.1	MY_CONTROLLER (CUSTOM LIBRARY ΓΕΝΙΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ)	57
5.5.2	ION_AUTH.PHP (CUSTOM USER AUTHORIZATION & AUTHENTICATION) ⁽¹³⁾	65
5.5.3	CNODE.PHP (CUSTOM LIBRARY ΓΙΑ ΤΗΣ ΚΑΤΗΓΟΡΙΕΣ) ⁽¹¹⁾	76
5.5.4	CRUD.PHP (CUSTOM LIBRARY ΓΙΑ ΤΗΣ CREATE-READ-UPDATE-DELETE) ⁽¹¹⁾	84
6	ΑΠΟΤΕΛΕΣΜΑ.....	91
6.1	ΣΥΜΠΕΡΑΣΜΑΤΑ	91
6.2	ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ	92
	ΒΙΒΛΙΟΓΡΑΦΙΑ	93
	ΠΑΡΑΡΤΗΜΑΤΑ.....	94
	ΠΑΡΑΡΤΗΜΑ Α: ΠΑΡΟΥΣΙΑΣΗ ΠΤΥΧΙΑΚΗΣ ΣΕ POWERPOINT SLIDES.	94

Πίνακας Εικόνων

Εικόνα 1 : Διάφορα PHP Frameworks	12
Εικόνα 2 : Διάγραμμα ροής CodeIgniter	19
Εικόνα 3 : Διάγραμμα ροής CodeIgniter.....	19
Εικόνα 4 : Διάγραμμα MVC	19
Εικόνα 5 : Εγκατάσταση wamp	22
Εικόνα 6 : Downloading wamp	22
Εικόνα 7 : Downloading CodeIgniter.....	23
Εικόνα 8 : Welcome Screen του CodeIgniter.....	24
Εικόνα 9 : Αρχική οθόνη Eclipse IDE	25
Εικόνα 10 : Βασική οθόνη eclipse IDE με PDT plugin	26
Εικόνα 11 : Δομή αρχείων αρχικής εγκατάστασης	27
Εικόνα 12 : Πίνακες στη βάση δεδομένων.....	30
Εικόνα 13 : Αρχική σελίδα ιστοτιούτου αισθητικής	31
Εικόνα 14 : BackOffice - Custom CMS Αρχική	46
Εικόνα 15 : Crud παραγγελιών με Χρήση JQGRID	47
Εικόνα 16 : Διαχείριση κατηγοριών.....	47
Εικόνα 17 : Πίνακας διαχείρισης προϊόντων	48
Εικόνα 18 : Φόρμα του CRUD για ένα προϊόν	48

1 Εισαγωγή

1.1 Περίληψη

Η παρούσα πτυχιακή πραγματοποιήθηκε από τον Σταυγιανουδάκη Ιωάννη, φοιτητή του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Κρήτης, στη σχολή τεχνολογικών εφαρμογών, στο τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων. Επιβλέπων καθηγητής ήταν η κυρία Λογκάκη Μαρία. Αντικείμενο της εργασίας αυτής είναι η σχεδίαση και η ανάπτυξη ενός ηλεκτρονικού ινστιτούτου αισθητικής με την χρήση της τεχνολογίας mvc με ένα από τα διασημότερα php framework της αγοράς εν ονόματι CodeIgniter.

Θα δημιουργήσουμε ένα δυναμικό web application όπου θα μπορεί να στηθεί ένα ηλεκτρονικό κατάστημα. Στην περίπτωση μας είναι ένα ινστιτούτο αισθητικής, το οποίο θα λειτουργεί σαν ηλεκτρονικό κατάστημα για αγορές προϊόντων ομορφιάς μέσω διαδικτύου και επιπλέον ο χρήστης που θα το επισκέπτεται θα μπορεί να κάνει κράτηση δυναμικά, για κάποια υπηρεσία που θα παρέχετε στο φυσικό κατάστημα. Ο χρήστης θα έχει την δυνατότητα να γίνει μέλος του ινστιτούτου. Να βλέπει το ιστορικό των αγορών του και των υπηρεσιών που έχει επιλέξει στο παρελθόν. Θα μπορεί να βλέπει νέα και ανακοινώσεις. Στην αρχική σελίδα θα εμφανίζονται δημοφιλή προϊόντα ανάλογα με στατιστικά όπως και νέα προϊόντα ανάλογα την ημερομηνία. Θα μπορεί να επικοινωνεί εύκολα με την εταιρία με την χρήση φόρμας επικοινωνίας και θα μπορεί να λαμβάνει ενημερωτικά e-mail.

Από την μεριά του ιδιοκτήτη – administrator θα στήσουμε ένα custom Content Management System. Θα το ονομάσουμε stavgianCMS. Από εκεί θα μπορεί να διαχειρίζεται κατά ένα μεγάλο ποσοστό της σελίδας του. Το stavgianCMS θα δίνει την δυνατότητα στους χρήστες με τα κατάλληλα δικαιώματα να βλέπουν και να διαχειρίζονται κατηγορίες, προϊόντα, galleries, χρήστες, newsletters, faq's, σελίδες, ραντεβού ,αγορές κα.

Δεν θα χρησιμοποιηθεί κάποιο έτοιμο CMS αλλά ένα προσωπικό cms που φτιάχτηκε με τη χρήση του CodeIgniter και της php σε συνεργασία με JQuery , ajax , json, xml και θα λειτουργήσει σαν backoffice του ηλεκτρονικού μας ινστιτούτου.

Για την επίτευξη αυτής της πτυχιακής εργασίας χρησιμοποιήθηκαν ακόμα χρήσιμα εργαλεία και την σχεδίαση και ανάπτυξη του ηλεκτρονικού μας ινστιτούτου ομορφιάς όπως το :

1. Eclipse PHP (για την συγγραφή κώδικα)
2. Navicat MySQL (για διαχείριση βάσεων δεδομένων)
3. PhpMyAdmin (για διαχείριση βάσεων δεδομένων)
4. WampServer (MySQL,Apache,PHP 5)
5. Photoshop (διαμόρφωση γραφικών)
6. Firefox & firebug (Browser)

1.2 Σκοπός και Στόχοι εργασίας

Στόχος της εργασίας αυτής είναι η δημιουργία μιας ιστοσελίδας(web application), ινστιτούτου αισθητικής με την διαδικασία εγγραφής χρηστών και κράτηση ραντεβού. Ακόμα στόχο έχει οι διάφοροι χρήστες να μπορούν να υποβάλουν κρατήσεις και να αγοράζουν προϊόντα, εύκολα γρήγορα και να μπορούν να επικοινωνούν με το ινστιτούτο με μία φόρμα επικοινωνίας για οποιαδήποτε απορία και πληροφορία.

Σκοπός της εργασίας αυτής είναι να μάθουμε για το σχεδιασμό και προγραμματισμό μιας ιστοσελίδας χρησιμοποιώντας το εργαλείο ανοικτού κώδικα CodeIgniter και η κατανόηση του τρόπου λειτουργίας του. Επίσης την κατανόηση κάποιων βασικών γλωσσών προγραμματισμού που απαιτούνται για την ανάπτυξη PHP Framework, όπως το CodeIgniter.

2 Τεχνολογίες & εργαλεία

2.1 Δυναμική ιστοσελίδα ⁽¹⁾

Δυναμική ιστοσελίδα είναι η ιστοσελίδα που τα στοιχεία της αλληλεπιδρούν με τον χρήστη. Οι δυναμικές ιστοσελίδες, σε αντίθεση με τις στατικές ιστοσελίδες, δεν είναι απλά HTML έγγραφα, αλλά συμπεριλαμβάνουν προγραμματισμό (σε μία γλώσσα προγραμματισμού κατάλληλη για το διαδίκτυο, όπως π.χ. είναι η php) και ουσιαστικά είναι web εφαρμογές.

Η κατασκευή δυναμικών ιστοσελίδων είναι προφανώς πιο πολύπλοκη από τις απλές στατικές ιστοσελίδες, και ο βαθμός δυσκολίας τους εξαρτάται φυσικά από τις λειτουργίες και τις δυνατότητες που περιλαμβάνει η εκάστοτε web εφαρμογή.

Στις δυναμικές ιστοσελίδες, το περιεχόμενο της ιστοσελίδας, αποθηκεύεται και αντλείται δυναμικά από μία ή περισσότερες βάσεις δεδομένων (π.χ. MySQL), ενώ διαθέτουν εκτός από το frontend (user interface) και το backend (administration area) μέσω του οποίου γίνεται εύκολα η διαχείριση του περιεχομένου της ιστοσελίδας. Έτσι συνολικά η web εφαρμογή συναντάται συνήθως ως CMS (content management system), δηλαδή σύστημα διαχείρισης περιεχομένου.

Πλεονεκτήματα από μια σωστά σχεδιασμένη ιστοσελίδα:

1. δημιουργία παρουσίας επιχείρησης στο διαδίκτυο
2. αύξηση αναγνωρισιμότητας επιχείρησης ή/και προϊόντων
3. προώθηση προϊόντων μέσω διαφημίσεων και διαδικτυακών εκστρατειών
4. αύξηση πωλήσεων
5. αύξηση του δικτύου πωλήσεων, πελατών και συνεργατών
6. υποστήριξη πελατών, εύκολη επικοινωνία με πελάτες
7. παροχή πληροφοριών σχετικά με νέα και ανακοινώσεις κάθε τύπου
8. προβολή της επιχείρησής σας ως σύγχρονη και τεχνολογικά προηγμένη

Χαρακτηριστικά δυναμικών ιστοσελίδων:

1. ειδικής κατασκευής σχεδιασμός ιστοσελίδας (custom made web design)
2. Σύστημα Διαχείρισης Περιεχομένου (Content Management System – CMS)
3. απεριόριστος αριθμός σελίδων, απεριόριστος αριθμός καταχωρήσεων
4. συμβατότητα με πολλαπλούς browsers, συμβατότητα με πρότυπα διαδικτύου
5. βελτιστοποίηση χρόνου απόκρισης (response time optimized)
6. καταχώρηση στις Μηχανές Αναζήτησης, βασικό Search Engine Optimization (SEO)
7. νέα & ανακοινώσεις, εγγραφή διαδικτυακού ονόματος (domain name) για 1 χρόνο
8. διαδικτυακή φιλοξενία (web hosting) για 1 χρόνο, στατιστικά στοιχεία
9. εκπαίδευση χρηστών

2.2 Τι είναι PHP mysql και τι phpmyadmin. (2)

Η **PHP** είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π. χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Η ιστορία της PHP ξεκινά από το 1995, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

Η **MySQL** είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) το οποίο μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά του από την κόρη του Μόντυ Βιντένιους, την Μάι. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, σήμερα θυγατρική της Sun Microsystems. (2)

Το **PhpMyAdmin** είναι ένα εργαλείο γραμμένο σε php με το οποίο διαχειριζόμαστε τις βάσεις δεδομένων που έχουμε μέσω web. Το phpMyAdmin μπορεί να χειρίζεται πλήρως βάσεις δεδομένων, πίνακες, πεδία πινάκων αλλά και ένα ολόκληρο MySQL Server. Υποστηρίζει 54 γλώσσες, μεταξύ των οποίων και τα ελληνικά και είναι λογισμικό ανοιχτού κώδικα. Δημιουργεί και να διαγράφει βάσεις δεδομένων, δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες, κάνει συντήρηση της βάσης, προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων, εκτελεί ερωτήματα SQL ακόμα και ομαδικά (batch), διαχειρίζεται κλειδιά σε πεδία, φορτώνει αρχεία κειμένου σε πίνακες, δημιουργεί και διαβάζει πίνακες (που προέρχονται από dump βάσης), εξάγει δεδομένα σε μορφή CVS, XML και LATEX, διαχειρίζεται πολλούς διακομιστές, διαχειρίζεται τους χρήστες MySQL και τα δικαιώματά τους, ελέγχει την αναφορική δραστηριότητα των δεδομένων των MyISAM πινάκων, δημιουργεί PDF γραφικών του layout της βάσης δεδομένων, εκτελεί αναζητήσεις σε όλη τη βάση δεδομένων ή μέρος αυτής, υποστηρίζει πίνακες InnoDB και ξένα κλειδιά, υποστηρίζει MySQLi, μια βελτιωμένη επέκταση του MySQL.



2.3 Τι είναι ο Apache και η xml. (2)



Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οπο ίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Συντηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle MySQL

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση.

Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET . Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους.

Η XML είναι, δηλαδή, ένα σύνολο κανόνων (ή διαφορετικά ένα πακέτο κατευθυντήριων γραμμών ή συμβάσεων) για το σχεδιασμό μορφών κειμένου οι οποίες διευκολύνουν τη δόμηση των δεδομένων σας. Η XML δεν είναι γλώσσα προγραμματισμού. Δε χρειάζεται, επομένως, να είστε προγραμματιστής για να τη χρησιμοποιήσετε ή να τη μάθετε. Η XML διευκολύνει τον υπολογιστή να παράγει δεδομένα, να διαβάζει δεδομένα και να εξασφαλίζει τη σαφήνεια της δομής των δεδομένων.

Η XML αποφεύγει τις συνήθεις παγίδες του σχεδιασμού γλωσσών: είναι επεκτάσιμη, ανεξάρτητη συστήματος υλικού και μπορεί να υποστηρίξει διεθνείς και τοπικές προσαρμογές. Είναι πλήρως συμβατή με Unicode.

2.4 Ηλεκτρονικό κατάστημα (e-shop) ⁽¹⁾

Ηλεκτρονικό κατάστημα (e-shop) είναι ο όρος που χρησιμοποιείται για να αναφερθεί κάποιος σε ένα διαδικτυακό τόπο (site) μέσω του οποίου πραγματοποιούνται πωλήσεις διαφόρων ειδών. Ορισμένες πλατφόρμες δημοπρασιών προσφέρουν και συστήματα ηλεκτρονικών καταστημάτων. Η Επανάσταση της Πληροφορικής άλλαξε σημαντικά τον τρόπο ζωής των πολιτών, επιφέροντας μια σειρά αλλαγών, που επηρεάζουν και τις εμπορικές επιχειρήσεις. Αυτές οι αλλαγές μπορούν να αποτελέσουν σημαντικό όπλο στα χέρια των επιχειρήσεων που θέλουν να ανταποκριθούν στις απαιτήσεις ενός νέου περιβάλλοντος που χαρακτηρίζεται από συνεχώς μεταβαλλόμενες συνθήκες, διεθνοποίηση και εντατικοποίηση του ανταγωνισμού, κ.α.



Οι επιχειρήσεις που θα "επιβιώσουν" στον ανταγωνισμό είναι αυτές που στον παρόντα χρόνο θα κάνουν τις στρατηγικές επιλογές για την ενσωμάτωση των νέων τεχνολογιών στις πρακτικές τους. Παλαιότερα, η ενσωμάτωση αυτή περιλάμβανε μόνο την ηλεκτρονική παρουσίαση των καταστημάτων, όχι όμως και όλων των ειδών τους και, πολύ περισσότερο, δεν υπήρχε δυνατότητα άμεσης παραγγελίας κάποιου είδους.

Με λίγα λόγια: Πωλήσεις σε οποιονδήποτε, οπουδήποτε, οποτεδήποτε. Έτσι, αναπτύσσονται διεθνώς, αλλά και στην Ελλάδα, τα ηλεκτρονικά καταστήματα, που προσφέρουν ημερησίως χιλιάδες προϊόντα που υπόσχονται χαμηλότερες τιμές. Ανάλογα με τα προσφερόμενα είδη, ο μελλοντικός πελάτης μπορεί να αναζητήσει ανάμεσα σε πολλά ομοειδή το συγκεκριμένο είδος που επιθυμεί, να μάθει την τιμή και τον χρόνο αποστολής (εάν το παραγγείλει), να το δει σε εικόνες (ορισμένες φορές και σε βίντεο) και να κάνει και σχετικές συγκρίσεις τιμών. Οι τιμές στα ηλεκτρονικά καταστήματα είναι φθηνότερες, γιατί ένα τέτοιο κατάστημα δεν διατηρεί σημεία πώλησης με υψηλό ενοίκιο, δεν απασχολεί αριθμητικά το ίδιο προσωπικό με ένα συμβατικό και παραμένει "ανοικτό" σε 24ωρη βάση και για 365 μέρες ετησίως. Ο μέλλον πελάτης μπορεί ακόμη να βρει και να παραγγείλει είδη που δεν υπάρχουν στα συμβατικά καταστήματα της πόλεως ή της χώρας του και μπορεί να πληρώσει μέσω της πιστωτικής του κάρτας ή με την χρήση της αντικαταβολής ή paypal.

2.5 Τι είναι τα PHP Frameworks ⁽²⁾

Η PHP είναι μια ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού για προγραμματισμό ιστοσελίδων. Ενώ υπάρχουν πάρα πολλές άλλες γλώσσες προγραμματισμού για το διαδίκτυο, όπως είναι για παράδειγμα η ASP, JSP, Ruby και άλλες, η **PHP** εξακολουθεί να είναι η πιο δημοφιλής από όλες.

Τι είναι όμως αυτό που κάνει πραγματικά την **PHP** τόσο δημοφιλή; Η PHP είναι η πιο δημοφιλής από όλες τις άλλες γλώσσες γιατί είναι σχετικά εύκολο να την μάθει κάποιος σε αντίθεση με όλες τις άλλες. Επιπλέον υπάρχουν πάρα πολλά άρθρα διαθέσιμα στο Ιντερνέτ για αρχάριους που βοηθάνε πολύ στην εκμάθησή της.

Παρόλο όμως που υπάρχουν πάρα πολλές πηγές στο διαδίκτυο για να μάθουμε την **PHP** και να βρούμε λύσεις στα προβλήματα μας, είναι δύσκολο να κατασκευάσουμε μια ολοκληρωμένη διαδικτυακή εφαρμογή ή μια σελίδα από την αρχή με την PHP. Ευτυχώς για εμάς υπάρχουν αρκετά αξιόπιστα **PHP Frameworks** που βοηθούν στην μείωση του χρόνου ανάπτυξης κάθε εφαρμογής. Αυτά τα Frameworks υποστηρίζονται από τεράστιες κοινότητες που θα σας βοηθήσουν όταν αντιμετωπίζεται προβλήματα.

Γνωστά PHP Frameworks ⁽¹⁾



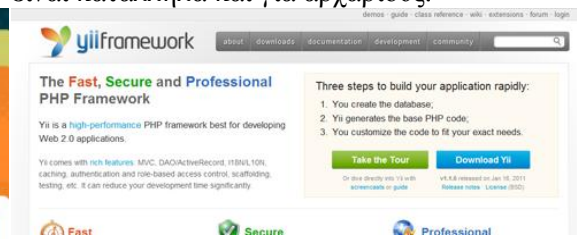
Το Zend Framework είναι ένα αντικειμενοστραφές framework γραμμένο σε PHP 5. Είναι βασισμένο στην απλότητα, με φιλική προς τις εταιρείες άδεια χρήσης. Η αφαιρετικά συνδεδεμένη αρχιτεκτονική της, επιτρέπει στους προγραμματιστές να χρησιμοποιούν στοιχεία του Framework μέσα σε άλλα frameworks



CakePHP: Χρησιμοποιεί κοινά μοτίβα σχεδίασης όπως είναι το MVC και ORM. Το CakePHP μειώνει το κόστος ανάπτυξης και βοηθάει τους προγραμματιστές να γράψουν λιγότερο κώδικα.



Το Symfony είναι ένα **PHP 5 Framework**, που παρέχει μια αρχιτεκτονική, στοιχεία και εργαλεία στους προγραμματιστές για ανάπτυξη σύνθετων εφαρμογών διαδικτύου γρήγορα. Στον επίσημο διαδικτυακό χώρο του Framework θα βρείτε επίσης και πρακτικά μαθήματα για εκμάθηση του Framework που είναι κατάλληλα και για αρχάριους.



Το Yii PHP Framework είναι ένα framework βασισμένο σε components υψηλής απόδοσης ιδανικό για εφαρμογές μεγάλου μεγέθους. Έχει μια μεγάλη λίστα από χαρακτηριστικά και είναι αυστηρά αντικειμενοστραφές.

Εικόνα 1 Διάφορα PHP Frameworks

2.6 Firebug⁽³⁾



Το Firebug είναι ένα Firefox addon που με ένα κλικ σας παρέχει μια σειρά εργαλείων web development, επιτρέποντάς σας να επεξεργαστείτε οποιαδήποτε σελίδα σε πραγματικό χρόνο. «Κάθεται» στη μπάρα (statusbar) του Firefox και ενεργοποιείται μόνο όταν το καλέσετε. Τα βασικά χαρακτηριστικά του περιλαμβάνουν τα εξής:

Έλεγχος και επεξεργασία της HTML. Το Firebug εμφανίζει τον πλήρη HTML κώδικα οποιασδήποτε σελίδας, δίνοντάς την δυνατότητα να τον επεξεργαστείτε απευθείας και να δείτε τις αλλαγές σε πραγματικό χρόνο. Σας βγάζει, μάλιστα, από τον κόπο να βρείτε τον κώδικα που αντιστοιχεί σε μια συγκεκριμένη περιοχή της

σελίδας, αφού όταν επιλέξετε την περιοχή εμφανίζεται μόνο η HTML που την αφορά. Ασφαλώς, οι αλλαγές χάνονται την επόμενη φορά που θα πατήσετε ανανέωση (refresh) της σελίδας.

Έλεγχος και επεξεργασία των CSS. Επιλέγοντας μια συγκεκριμένη περιοχή, το Firebug εμφανίζει, πέρα από την HTML, τα CSS που της αντιστοιχούν. Και σε αυτή την περίπτωση, σας δίνει τη δυνατότητα να τα επεξεργαστείτε και να δείτε απευθείας τα αποτελέσματα. Για την ακρίβεια, μπορείτε να αλλάξετε ολοκληρωτικά τα CSS μιας σελίδας και να δείτε πώς θα φαινόταν. Καθώς συμπληρώνετε, μάλιστα, μια τιμή, το Firebug εμφανίζει προτάσεις με τις τιμές που θα μπορούσε να περιλαμβάνει, μεγάλο βοήθημα στην περίπτωση που δεν είσατε σίγουροι για την σωστή σύνταξη μιας καταχώρησης.

Εργαλεία ελέγχου και απασφαλμάτωσης. Το Firebug περιλαμβάνει μια σειρά εργαλείων για τον έλεγχο της σελίδας σας και τη διόρθωση ενδεχομένων λαθών: javascript debugger, γρήγορο εντοπισμό λαθών σε Javascript, CSS και XML, DOM Inspector, ακόμα και παρακολούθηση της δικτυακής δραστηριότητας ώστε να βρείτε τους λόγους για τους οποίους η σελίδα σας αργεί να φορτώσει κ.α.

Επεκτασιμότητα. Το Firebug μπορεί να δεχτεί δεκάδες επιπλέον extensions. Δεν πρόκειται για επεκτάσεις του Firefox, αλλά για επεκτάσεις του ίδιου του Firebug, που του προσθέτουν επιπλέον λειτουργικότητες και που μπορούν να σας φανούν χρήσιμες ανάλογα με την τεχνολογία την οποία χρησιμοποιείτε. Π.χ. υπάρχουν, μεταξύ άλλων, επεκτάσεις για development σε jQuery, Flash, PHP, αλλά και επεκτάσεις όπως το Page Speed που δείχνει την ταχύτητα φόρτωσης μιας σελίδας.

Συνεχής ανάπτυξη. Το Firebug αναπτύσσεται συνεχώς, βγάζοντας κάθε τόσο νέες, ενημερωμένες εκδόσεις.

Ηλεκτρονική διεύθυνση: <http://getfirebug.com/>

Firebug extensions: <http://getfirebug.com/downloads#extensions>

3 Το CodeIgniter



3.1 Τι είναι το CodeIgniter⁽⁴⁾

Το CodeIgniter είναι ένα **Opensource PHP framework**. Είναι ένα εργαλείο για τους ανθρώπους που αναπτύσσουν web εφαρμογές χρησιμοποιώντας PHP. Στόχος του είναι να επιτρέψει να αναπτυχθούν σχέδια πολύ πιο γρήγορα, από ό, τι θα μπορούσατε να συντάσσοντας κώδικα από την αρχή, μαζί με το πακέτο παρέχεται ένα πλούσιο σύνολο βιβλιοθηκών(libraries) για γενικότερα απαιτούμενα καθήκοντα, καθώς και ένα απλό interface και λογική δομή για την πρόσβαση σε αυτές τις βιβλιοθήκες. Το CodeIgniter μας επιτρέπει να επικεντρωθούμε δημιουργικά στο έργο μας ελαχιστοποιώντας τις γραμμές κώδικα που προϋποθέτει μια συγκεκριμένη εργασία.

Αν είστε ήδη PHP web developer το CodeIgniter θα μας βοηθήσει να κάνουμε καλύτερα και πιο εύκολα αυτό που θέλουμε. Θα περιορίσει αισθητά τον κώδικα, που στην ουσία χρειάζεται να γραφτεί από τον προγραμματιστή. Τα script μας θα είναι πολύ πιο εύκολα αναγνώσιμα και ανανεώσιμα. Θα μας βοηθήσει να δώσουμε σε μεγάλους ιστότοπου μία συνεπής δομή. Θα πάρει πρωτοβουλίες για πράγματα τα οποία για εμάς θα είναι απλά μία ρουτίνα. Το CodeIgniter δεν είναι :

- Για αυτούς που δεν έχουν επαρκή γνώση της PHP & HTML.
- Αυτούς που θέλουν να κάνουν ένα CMS γρήγορα και απλά, με ελάχιστο κώδικα.
- Για αυτούς που θέλουν απλά να φτιάχνουν απλές σελίδες με λίγα χαρακτηριστικά.

Το CodeIgniter απευθύνετε σε αυτούς που θέλουν:

- ένα framework με ένα μικρό ίχνος.
- εξαιρετική απόδοση.
- ευρεία συμβατότητα με τυποποιημένους hosting accounts που τρέχουν ποικιλία εκδόσεων και configuration της PHP.
- ένα framework που απαιτεί σχεδόν μηδαμινή διαμόρφωση.
- ένα framework που δεν απαιτεί χρήση γραμμή εντολών.
- ένα framework που δεν έχει περιορισμούς κωδικοποίησης.
- να αποφεύγουν την πολυπλοκότητα, που ευνοεί τις απλές λύσεις.
- σαφή, λεπτομερή documentation με απίστευτο μεγάλο community .

3.2 Χαρακτηριστικά του CodeIgniter⁽⁴⁾

Τα χαρακτηριστικά γνωρίσματα είναι ένας πολύ φτωχός τρόπος να κριθεί μια εφαρμογή δεδομένου ότι δεν λαμβάνεται υπόψη η εμπειρία των χρηστών, για το πόσο διαισθητικά ή έξυπνα είναι σχεδιασμένη αυτή η εφαρμογή.

Τα χαρακτηριστικά γνωρίσματα δεν αποκαλύπτουν τίποτα για την ποιότητα του κώδικα, ή την απόδοση, ή την προσοχή στη λεπτομέρεια, ή τις πρακτικές ασφάλειας. Ο μόνος τρόπος να κριθεί πραγματικά ένα application είναι να δοκιμαστεί από τον developer ανακαλύπτοντας τις δυνατότητες του κώδικα.

To CodeIgniter είναι application framework.

Είναι ένα εργαλείο για web developers που φτιάχνουν εφαρμογές PHP. Ο στόχος είναι να μας επιτρέψει να αναπτύσσουμε προγράμματα πολύ γρηγορότερα από το να γράφαμε τον κώδικα από την αρχή με την παροχή μεγάλης ποικιλίας βιβλιοθηκών (libraries) για συνήθεις διεργασίες που απαιτούν επαναληπτικό κώδικα. Μας δίνει τη δυνατότητα να εστιάσουμε δημιουργικά στο project μας με ελάχιστο γράψιμο κώδικα που απαιτείται για μία διεργασία.

Παρέχετε Δωρεάν

Έχει χορηγηθεί άδεια ελεύθερης χρήσης λογισμικού ανοικτού κώδικα από την Apache/BSD

Αληθινά ελαφρύ

Το σύστημα πυρήνων (core System) απαιτεί μόνο μερικές πολύ μικρές βιβλιοθήκες. Αυτό είναι μία αντίθεση σε άλλα application frameworks που απαιτούν περισσότερους πόρους. Οι πρόσθετες βιβλιοθήκες φορτώνονται δυναμικά κατόπιν αιτήσεως, βασισμένο στις ανάγκες σας για μια δεδομένη διαδικασία, έτσι το σύστημα βάσεων είναι πολύ εύκαμπτο και αρκετά γρήγορο.

Γρήγορο

Το CodeIgniter μας προκαλεί να βρούμε ένα γρηγορότερο όπως αναφέρετε στο documentation. Και όλο αυτό επειδή κάνει χρήση MVC.

CodeIgniter παράγει καθαρά URLs

Τα URLs που παράγονται από το CodeIgniter είναι καθαρά και φιλικά στις μηχανές αναζήτησης.

example.com/news/article/345

Σημείωση: Εξ ορισμού το αρχείο `index.php` συμπεριλαμβάνεται στο URL αλλά μπορεί να αφαιρεθεί χρησιμοποιώντας ένα απλό αρχείο `.htaccess`.

To CodeIgniter είναι ένα δυνατό πακέτο.

Το CodeIgniter έρχεται μαζί με μία πλήρη σειρά από βιβλιοθήκες που επιτρέπουν στους συνήθεις αναγκαίους στόχους όπως επικοινωνία με μία βάση, αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου, επικύρωση δεδομένων (validation), διαχείριση sessions, διαχείριση εικόνων, διαχείριση XML – RPC και πολλά άλλα.

Μπορεί να επεκταθεί εύκολα.

Μπορούμε να επεκτείνουμε εύκολα μέσω των βιβλιοθηκών, κάποιων helpers και επίσης δεν απαιτείται template Engine. Δεν είμαστε αναγκασμένοι να χρησιμοποιήσουμε όλες τις δυνατότητες και όλες τις βιβλιοθήκες.

Το CodeIgniter είναι λεπτομερώς τεκμηριωμένο

Οι προγραμματιστές αγαπούν να κωδικοποιούν και να μισούν να γράφουν την τεκμηρίωση. Δεδομένου ότι η τεκμηρίωση είναι τόσο σημαντική όσο ο ίδιος ο κώδικας, το CodeIgniter δεσμεύτηκε να το κάνει και το έκανε.. Ο κώδικας μας είναι εξαιρετικά καθαρός και καλά σχολιασμένος επίσης.

Το CodeIgniter έχει μια φιλική Κοινότητα των χρηστών

Η αυξανόμενη κοινότητα χρηστών μπορεί να φανεί ενεργά από χρήστες που συμμετέχουν στα κοινοτικά φόρουμ.

3.3 Βιβλιοθήκες, helpers & driver functions του CodeIgniter⁽⁴⁾

Library Functions Reference

<p>Benchmark mark() elapsed_time() memory_usage()</p> <p>Calendar generate()</p> <p>Config item() load() set_item() site_url() system_url()</p> <p>Cart insert() update() total() total_items() contents() has_options() product_options() destroy()</p> <p>Email from() to() cc() bcc() subject() message() send() print_debugger() initialize() reply_to() set_alt_message() clear() attach()</p> <p>Encryption encode() decode() set_cipher() set_mode() sha1() encode_from_legacy()</p> <p>Form Validation (Helper) form_error() validation_errors() set_value() set_rules() set_select() set_checkbox() set_message() set_radio() run()</p>	<p>FTP connect() upload() close() list_files() mirror() download() rename() move() delete_file() delete_dir() list_files() chmod() mkdir()</p> <p>File Uploading do_upload() data() display_errors()</p> <p>HTML Table generate() set_caption() set_heading() add_row() make_columns() set_template() set_empty() clear() function()</p> <p>Image Manipulation resize() crop() rotate() clear() watermark() display_errors()</p> <p>Input & Security post() get() get_post() cookie() server() set_cookie() get_cookie() ip_address() valid_ip() user_agent() request_headers() get_request_header() is_ajax_request()</p>	<p>Java Script event() effect() hide() show() toggle() animate() click() fadeIn() fadeOut() toggleClass() slideUp() slideDown() slideToggle() corner() tablesorter() modal() calendar()</p> <p>Loader Class view() model() library() database() vars() helper() plugin() file() config() language() add_package_path() remove_package_path()</p> <p>Language line() load()</p> <p>Output set_output() get_output() append_output() set_header() set_status_header() enable_profiler() set_profiler_sections() cache() parse_exec_vars()</p> <p>Session userdata() set_userdata() unset_userdata() set_flashdata() flashdata() keep_flashdata() sess_destroy() flashdata() set_flashdata() keep_flashdata() sess_destroy()</p>	<p>Security xss_clean() sanitize_filename()</p> <p>Template Parser parse() parse_string()</p> <p>Pagination initialize() create_links()</p> <p>Typography auto_typography() format_characters() nl2br_except_pre() protect_braced_quotes()</p> <p>Unit Testing run() report() result() use_strict() active() set_test_items() set_template()</p> <p>URI segment() rsegment() slash_segment() slash_rsegment() uri_to_assoc() ruri_to_assoc() assoc_to_uri() uri_string() ruri_string() total_segments() total_rsegments() segment_array() rsegment_array()</p> <p>User Agent is_browser() is_mobile() is_robot() is_referral() browser() version() mobile() robot() platform() referrer() agent_string() accept_string() accept_charset()</p>	<p>Trackback send() display_errors() send_error() send_success() receive() data()</p> <p>XML - RPC server() timeout() method() request() send_request() set_debug() display_error() display_response() send_error_message() send_response()</p> <p>Zip Encoding add_data() add_dir() read_file() read_dir() archive() download() get_zip() clear_data()</p>
---	---	--	--	--



Helper Functions Reference

Array Helper
 element()
 random_element()
 elements()

Captcha
 captcha_create()

Cookie Helper
 set_cookie()
 get_cookie()
 delete_cookie()

Date Helper
 now()
 mdate()
 standard_date()
 local_to_gmt()
 gmt_to_local()
 mysql_to_unix()
 unix_to_human()
 human_to_unix()
 timespan()
 days_in_month()
 timezones()
 timezone_menu()

Directory Helper
 directory_map()

Download Helper
 force_download()

Email Helper
 valid_email()
 send_email()

File Helper
 read_file()
 write_file()
 delete_files()
 get_filenames()
 get_dir_file_info()
 get_file_info()
 get_mime_by_extension()
 symbolic_permissions()
 octal_permissions()
 br()
 heading()
 img()
 link_tag()
 nbs()
 ol()
 ul()
 meta()
 doctype()

Form Helper
 form_open()
 form_open_multipart()
 form_hidden()
 form_input()
 form_password()
 form_upload()
 form_textarea()
 form_dropdown()
 form_fieldset()
 form_fieldset_close()
 form_checkbox()
 form_multiselect()
 form_radio()
 form_submit()
 form_label()
 form_reset()
 form_button()
 form_close()
 form_prep()
 set_value()
 set_select()
 set_checkbox()
 set_radio()

HTML Helper
 br()
 heading()
 img()
 link_tag()
 nbs()
 ol()
 ul()
 meta()
 doctype()

Inflector Helper
 singular()
 plural()
 camelize()
 underscore()
 humanize()

Language Helper
 lang()

Number helper
 byte_format()

Path Helper
 set_realpath()

Security Helper
 xss_clean()
 dohash()
 strip_image_tags()
 sanitize_filename()
 encode_php_tags()

Smiley Helper
 smiley_js()
 parse_smileys()
 get_clickable_smileys()

String Helper
 random_string()
 alternator()
 repeater()
 reduce_double_slashes()
 trim_slashes()
 reduce_multiples()
 quotes_to_entities()
 strip_quotes()

Text Helper
 word_limiter()
 character_limiter()
 ascii_to_entities()
 entities_to_ascii()
 convert_accented_characters()
 word_censor()
 highlight_code()
 highlight_phrase()
 word_wrap()
 ellipsis()

Typography Helper
 auto_typography()
 nl2br_except_pre()

URL Helper
 site_url()
 base_url()
 current_url()
 uri_string()
 index_page()
 anchor()
 anchor_popup()
 mailto()
 safe_mailto()
 auto_link()
 url_title()
 prep_url()
 redirect()

XML Helper
 xml_convert()

Driver Functions Reference

Caching
 is_supported()
 get()
 save()
 delete()
 clean()
 cache_info()
 get_metadata()
 apc->save()
 file->save()
 memcached->save()

Database - Queries
 query()
 simple_query()
 protect_identifiers()
 escape()
 escape_str()
 escape_like_str()
 insert_string()
 update_string()

Database - Active Records

get()
 get_where()
 select()
 select_max()
 select_min()
 select_avg()
 select_sum()
 from()
 join()
 where()
 or_where()
 where_in()
 or_where_in()
 where_not_in()
 or_where_not_in()
 like()
 or_like()
 not_like()
 or_not_like()
 group_by()
 distinct()
 having()
 or_having()
 order_by()
 limit()
 count_all_results()
 count_all()
 insert()
 set()
 update()
 delete()
 empty_table()
 truncate()

Database - Table
 list_tables()
 table_exists()

Database - Transactions
 trans_start()
 trans_complete()
 trans_off()
 trans_begin()
 trans_status()
 trans_rollback()
 trans_commit()
 trans_strict()

Database - Forge
 create_database()
 drop_database()
 add_field()
 add_key()
 create_table()
 drop_table()
 rename_table()
 add_column()
 drop_column()
 modify_column()
 some_function()

Database - Utility
 list_databases()
 database_exists()
 optimize_table()
 repair_table()
 optimize_database()
 csv_from_result()
 xml_from_result()
 backup()

Database - Results
 result()
 result_array()
 row()
 row_array()
 num_rows()
 num_fields()
 free_result()
 affected_rows()
 insert_id()
 count_all()
 platform()
 version()
 last_query()

Active Record Caching
 start_cache()
 stop_cache()
 flush_cache()

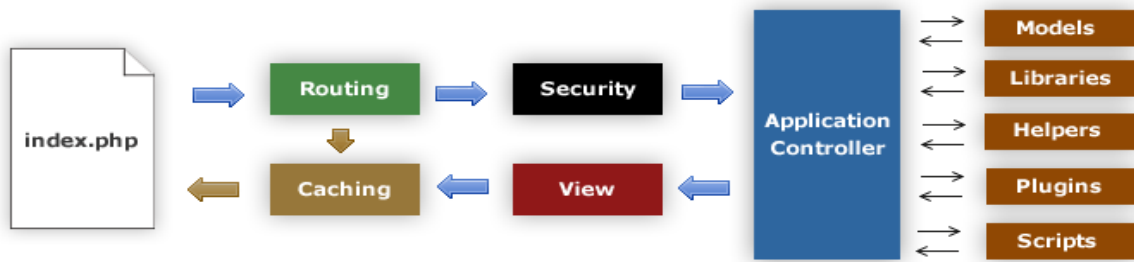
Database - Fields
 list_fields()
 field_exists()
 field_data()

Database - Custom
 call_function()

Database - caching
 cache_on()
 cache_off()
 cache_delete()
 cache_delete_all()

JavaScript
 event()
 effect()
 hide()
 show()
 toggle()
 animate()
 fadeIn()
 fadeOut()
 toggleClass()
 slideUp()
 slideDown()
 slideToggle()
 corner()
 tablesorter()
 modal()
 calendar()

3.4 Διάγραμμα ροής και MVC ⁽⁴⁾



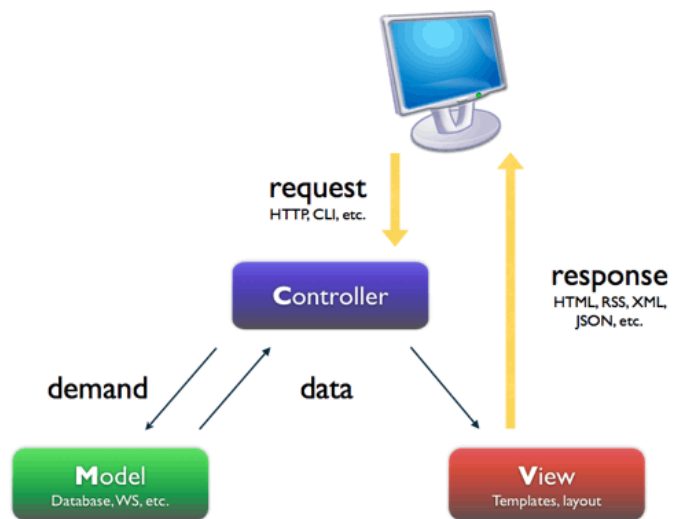
Εικόνα 2 : Διάγραμμα ροής Codeigniter

Το CodeIgniter είναι βασισμένο στο σχέδιο ανάπτυξης **MVC** από τα αρχικά Model – View – Controller. Είναι μια προσέγγιση λογισμικού που χωρίζει τη λογική της εφαρμογής από την λογική της παρουσίαση.

Το **MODEL** αντιπροσωπεύει τις δομές δεδομένων μας. Χαρακτηριστικά οι κλάσεις των μοντέλων περιέχουν τις λειτουργίες που μας βοηθούν να ανακτήσουμε, να παρεμβάλουμε, και να ενημερώσουμε τις πληροφορίες στη βάση δεδομένων μας.

Το **VIEW** είναι οι πληροφορίες που παρουσιάζονται σε έναν χρήστη. Ένα View θα μπορούσε να είναι μία ολοκληρωμένη ιστοσελίδα, αλλά στο CodeIgniter, ένα view μπορεί επίσης να είναι ένα μέρος σελίδων όπως μια επιγραφή ή μια υποσημείωση, ένα μενού. Μπορεί επίσης να είναι μια σελίδα RSS, ή οποιοσδήποτε τύπος "σελίδας".

Ο **CONTROLLER** χρησιμεύει ως ένας μεσάζων μεταξύ του model, view και οτιδήποτε απαιτείται για να γίνει επεξεργασία ενός αιτήματος HTTP και να παραχθεί μία ιστοσελίδα.



Εικόνα 3 : Διάγραμμα MVC

3.5 Παραδείγματα ⁽⁴⁾

Παράδειγμα 1: Αποστολή e-mail με CodeIgniter

```
$this->load->library('email');  
$this->email->from('your@your-site.com', 'Your Name');  
$this->email->subject('Email Test');  
$this->email->message('Testing the email class.');
```

```
$this->email->send();
```

Παράδειγμα 2: Συμπίεση αρχείων ⁽⁴⁾

```
$name = 'mydata1.txt';  
$data = 'the contents of my file.....';  
$this->zip->add_data($name, $data);  
$this->zip->archive('c:/my_backup.zip');
```

4 Μέρος πτυχιακής εργασίας

4.1 Ανάλυση προβλήματος

Για τις ανάγκες επίλυσης του προβλήματος μας χρειάζεται να μάθουμε κάποιες βασικές αρχές κάποιων βασικών γλωσσών προγραμματισμού, μελέτη για το περιβάλλον και δομή των τύπων για το πώς θα δομηθεί μια κράτηση, η γνώση για το Codeigniter και την βασική ρύθμιση και διαχείριση του. Βασική γνώση PHP MySQL και οργανωτικό πνεύμα και τάση στον αντικειμενοστραφή προγραμματισμό. Η βασική μας θεωρία για την υλοποίηση αυτής της ιστοσελίδας είναι το όσο το πιο δυνατό να καταφέρουμε να επιλύσουμε το πρόβλημα μας με απλό και κατανοητό τρόπο για όλους τους πιθανούς χρήστες.

4.2 Απαιτήσεις Εφαρμογής

Η Εφαρμογή μας πρέπει να είναι:

1. Ελκυστική προς τον περιηγητή
2. Εύκολη στην χρήση
3. Πλούσια με χρήσιμες πληροφορίες
4. Κατανοητή προς τον χρήστη
5. Να προσφέρει αμεσότητα και ποικιλία

Εγγραφή πελάτη Περιγραφή Απαίτησης

- Αρχικά ζητείται από τον πελάτη να συμπληρώσει τα στοιχεία του. Τα απαραίτητα προς συμπλήρωση στοιχεία είναι τα εξής: Όνομα χρήστη, διεύθυνση email, συνθηματικό, επιβεβαίωση συνθηματικού, πόλη, διεύθυνση, τηλέφωνο, φύλλο κα.
- Τα στοιχεία θα ελέγχονται για την εγκυρότητά τους και θα εμφανίζεται κατάλληλο μήνυμα σε περίπτωση σφάλματος.
- Θα αποθηκεύονται τα στοιχεία του πελάτη και τα στοιχεία της εγγραφής που θα πραγματοποιήσει.
- Θα στέλνει το σύστημα email για να μπορέσει να ενεργοποιηθεί ο λογαριασμός του.
- Θα μπορεί να αλλάζει στοιχεία – κωδικό – πάντα με ταυτοποίηση.

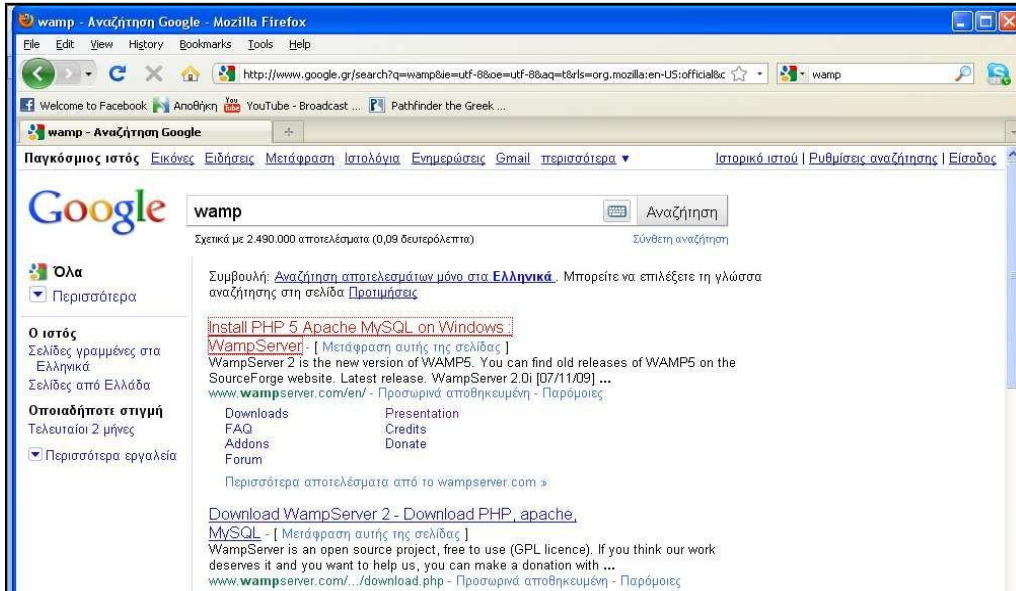
Κλείσιμο ραντεβού πελάτη Περιγραφή Απαίτησης

- Ζητείται από τον πελάτη να συμπληρώσει τη φόρμα κλεισίματος ραντεβού όταν θα έχει ήδη επιλέξει την μέρα και την ώρα που επιθυμεί να κλείσει το ραντεβού. Τα στοιχεία θα ελέγχονται για την εγκυρότητά τους και θα εμφανίζεται κατάλληλο μήνυμα σε περίπτωση σφάλματος.
- Θα αποθηκεύονται τα στοιχεία του πελάτη και τα στοιχεία του ραντεβού που θα πραγματοποιήσει.

4.3 Υλοποίηση

4.3.1 Εγκατάσταση Wamp⁽⁵⁾

Πολύ εύκολα, πατώντας Wamp στο Google εμφανίζεται το παρακάτω παράθυρο από το οποίο επιλέγουμε και κατεβάζουμε το πακέτο Wamp.



Εικόνα 5 : downloading wamp

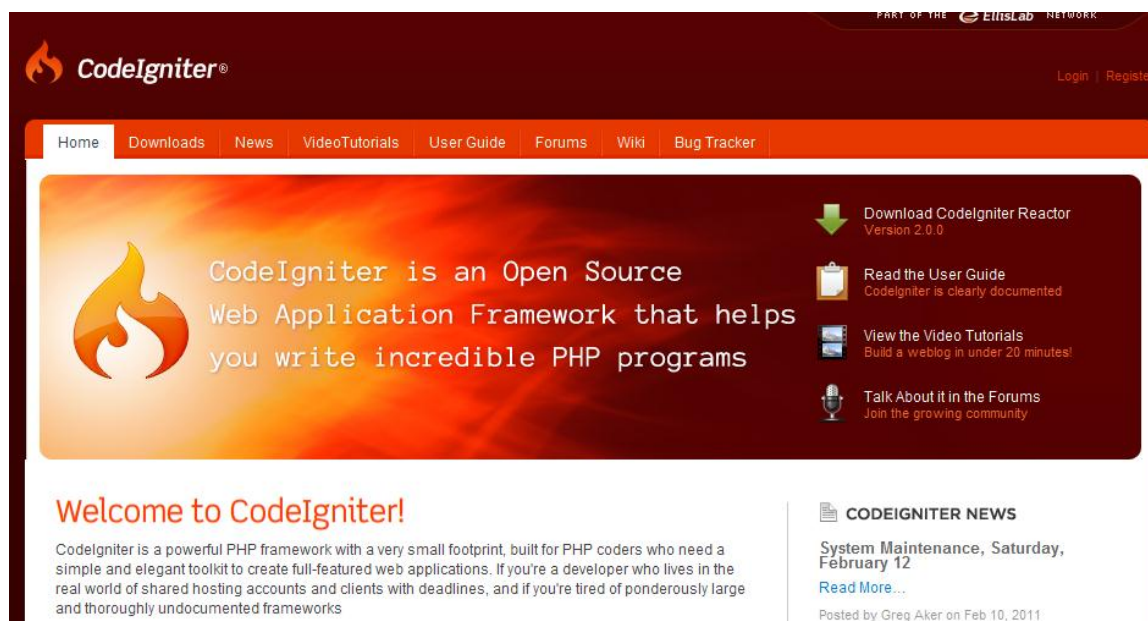


Εικόνα 4 : Εγκατάσταση wamp

Ακολουθώντας τις οδηγίες γίνεται η εγκατάσταση του Wamp χωρίς ιδιαίτερη δυσκολία

4.3.2 Εγκατάσταση του CodeIgniter⁽⁴⁾

Πηγαίνουμε στην σελίδα <http://www.CodeIgniter.com> και πατάμε download.



Εικόνα 6 : Downloading Codeigniter

Μετά την μεταφόρτωση των αρχείων του CodeIgniter αποσυμπιέζουμε τα αρχεία στο www folder του Wamp τον φάκελο που μετονομάσαμε σε beauty-plus.gr. Αν πληκτρολογήσουμε τότε στον browser το <http://localhost/beauty-plus.gr/welcome> τότε θα μεταβούμε στο welcome του CodeIgniter όπου είναι ένα απλό παράδειγμα του μίας δυναμικής σελίδας που φορτώνετε από έναν controller welcome που χρησιμοποιεί το view welcome_message.php. Όπως δείχνει και το url δεν φορτώνουμε το welcome_message.php αλλά τον controller welcome.

View : welcome_message.php

Welcome to CodeIgniter!

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you'll find it located at:

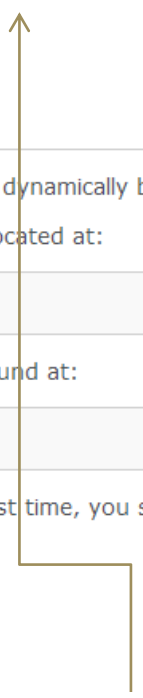
`system/application/views/welcome_message.php`

The corresponding controller for this page is found at:

`system/application/controllers/welcome.php`

If you are exploring CodeIgniter for the very first time, you should start by reading the [User Guide](#).

Page rendered in 0.1966 seconds



Εικόνα 7 : Welcome Screen του CodeIgniter

```
CONTROLLER : welcome.php
<?php

class Welcome extends Controller {

    function Welcome()
    {
        parent::Controller();
    }

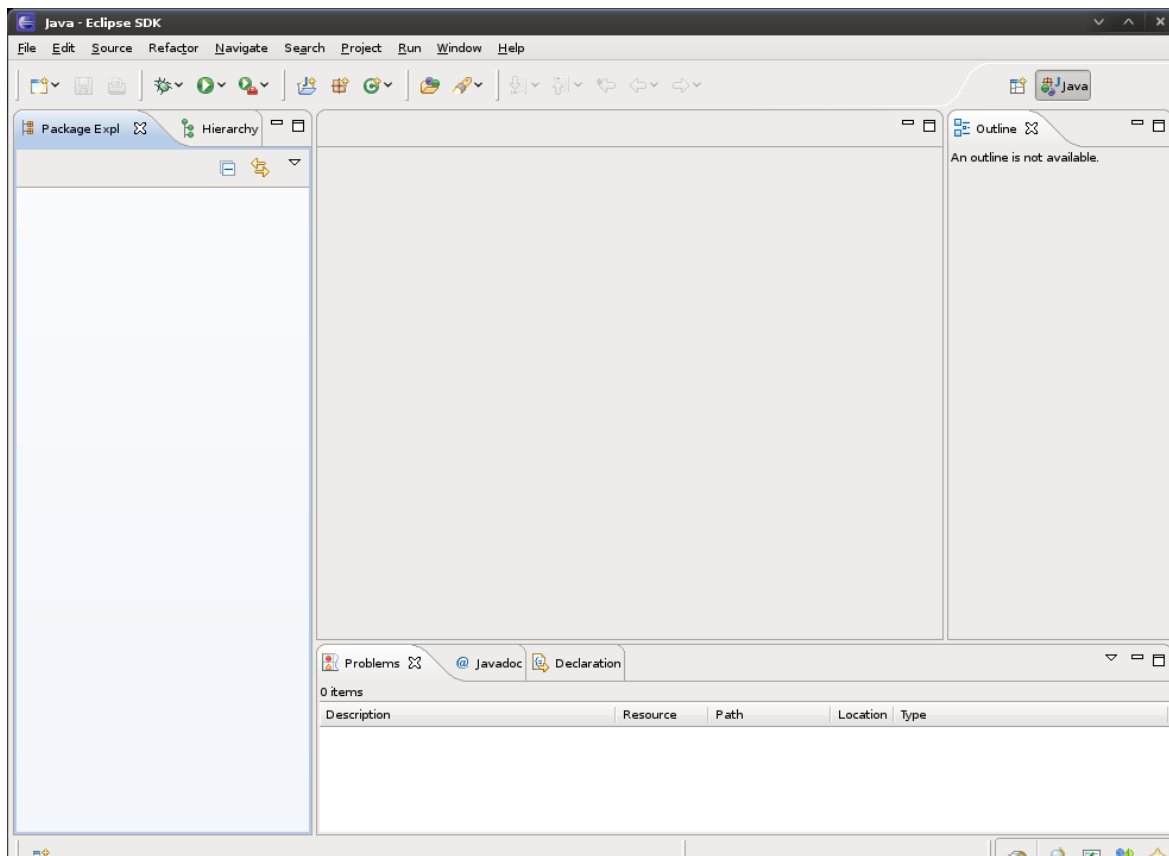
    function index()
    {
        $this->load->view('welcome_message');
    }
}

/* End of file welcome.php */
/* Location: ./system/application/controllers/welcome.php */
```



4.3.3 Eclipse PHP (2)

Θεωρούμε καθ'όλη την έκταση του κεφαλαίου αυτού ότι έχουμε σαν χώρο εργασίας (workspace) τον φάκελο ~/www/. Και σ' αυτόν έχουμε έναν φάκελο για κάθε εργασία μας. Βέβαια χρειάζεται προσεκτική ρύθμιση των δικαιωμάτων πρόσβασης προκειμένου να λειτουργεί σωστά και ο εξυπηρετητής http αλλά και να μπορούμε να επεξεργαστούμε τα αρχεία.



Εικόνα 8 : Αρχική οθόνη Eclipse IDE

Αρχικά θα δούμε την οθόνη δεξιά. Αυτό που θα παρατηρήσουμε είναι ότι είναι ανοιχτή μια java όψη-προοπτική (perspective). Μια από τις πρώτες μας εργασίες είναι να αλλάξουμε σε php όψη και να κλείσουμε την java προοπτική. Αυτό θα το καταφέρουμε με δεξί κλικ στην java όψη πάνω δεξιά και επιλέγοντας *close* στο μενού αντικειμένου και κατόπιν αριστερό κλικ στο κουμπί *open perspective* και στο μενού που εμφανίζεται 'other' και κατόπιν επιλέγουμε την όψη PHP.

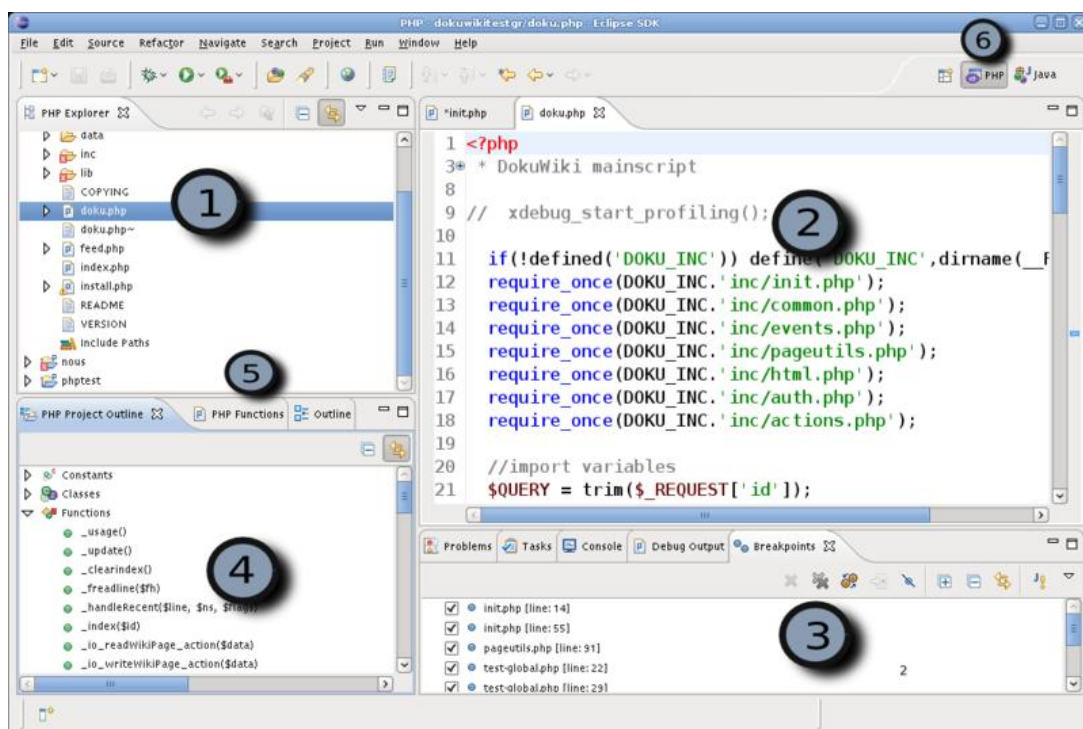
Το επόμενο βήμα μας είναι να εισάγουμε τυχόν υπαρκτές εργασίες που έχουμε δημιουργήσει.

1. Από την κεντρική γραμμή μενού επιλέξτε *File > Import...* και ανοίξει ο καθοδηγητής Εισαγωγής.
2. Επέλεξε *General > Existing Project into Workspace* και κάνε κλικ στο Next.
3. Επέλεξε είτε *Select root directory* ή *Select archive file* και κάνε κλικ στο σχετικό πλοηγό για να βρεις τον κατάλογο ή το αρχείο που περιέχει το project.
4. Κάτω από τα εγχειρήματα επέλεξε το/τα εγχειρήματα που θα ήθελες να εισάγεις
5. Κάνε κλικ στο Finish για να αρχίσει η εισαγωγή.

Σε περίπτωση που δεν γίνεται η εισαγωγή του έργου σας δημιουργήστε εκ νέου ένα έργο και μετακινήστε εκεί τα αρχεία του.

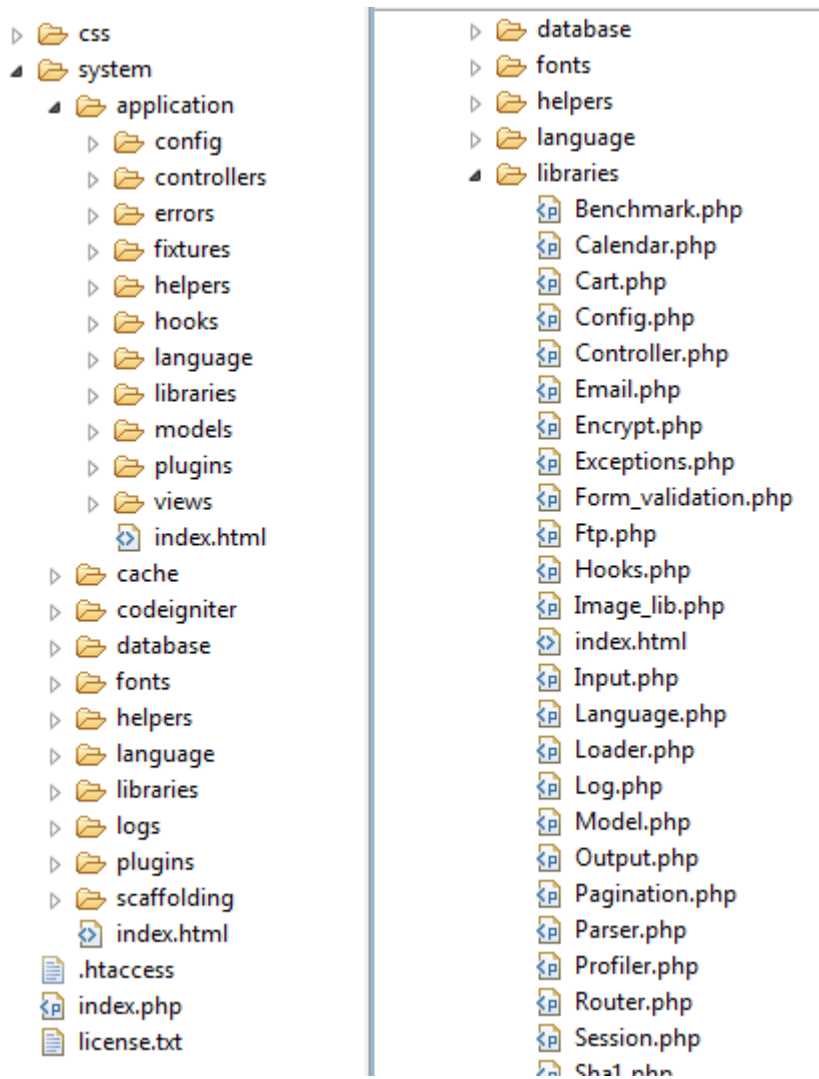
Αριθμημένα τα βασικά μέρη της php προοπτικής eclipse :

1. Περιγητής καταλόγων του επιλεγμένου php έργου (php explorer).
2. Πολλαπλές καρτέλες τμημάτων επεξεργασίας κειμένου.
3. Καρτέλες βοηθητικών όψεων.
4. Περίγραμμα των συναρτήσεων , κλάσεων κτλ που περιέχονται στο έργο μας
5. Καρτέλα (που δεν είναι ενεργή) όλων των διαθέσιμων php συναρτήσεων. (php functions)



Εικόνα 9 : Βασική οθόνη eclipse IDE με PDT plugin

4.4 Η δομή των αρχείων⁽⁴⁾



Εικόνα 10 : Δομή αρχείων αρχικής εγκατάστασης

4.5 Δημιουργία `.htaccess` & *Virtual Hosts* ⁽⁶⁾

Έχουμε τοποθετήσει τον φάκελο με τα αρχεία του CodeIgniter στο φάκελο όπου έχουμε ορίσει σαν root μέσω του Wamp. Στην περίπτωση μας χρειάστηκε να δημιουργήσουμε ένα αρχείο `.htaccess` για να μπορούμε να αφαιρέσουμε κατά την εκφόρτωση της σελίδας μας την κατάληξη `index.php`. Έχουμε ονομάσει τον φάκελο του project **beauty-plus.gr**. Μέσα στο φάκελο `beauty-plus.gr/system` υπάρχει ένας φάκελος `application` όπου είναι αυτός ο φάκελος στον οποίο θα δημιουργήσουμε την σελίδα μας. Για λόγους ευκολίας και λειτουργικότητας τον μεταφέραμε στο root μας.

Δημιουργήσαμε το `.htaccess` αρχείο μας που περιέχει μέσα τις ακόλουθες γραμμές κώδικα όπου είναι εντολές προς τον Apache Web server για το που και σε τι ακριβώς να δώσει πρόσβαση όπως και τι να κάνει και πώς να συμπεριφερθεί σε διάφορες περιπτώσεις.

.htaccess

```
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteBase /

  #Removes access to the system folder by users.
  #Additionally this will allow you to create a System.php controller,
  #previously this would not have been possible.
  #'system' can be replaced if you have renamed your system folder.
  RewriteCond %{REQUEST_URI} ^system.*
  RewriteRule ^(.*)$ /index.php?/$1 [L]

  #When your application folder isn't in the system folder
  #This snippet prevents user access to the application folder
  #Rename 'application' to your applications folder name.
  RewriteCond %{REQUEST_URI} ^application.*
  RewriteRule ^(.*)$ /index.php?/$1 [L]

  #Checks to see if the user is attempting to access a valid file,
  #such as an image or css document, if this isn't true it sends the
  #request to index.php
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^(.*)$ index.php?/$1 [L]
</IfModule>

<IfModule !mod_rewrite.c>
  # If we don't have mod_rewrite installed, all 404's
  # can be sent to index.php, and everything works as normal.
  # Submitted by: ElliotHaughin
  ErrorDocument 404 /index.php

</IfModule>
```

Για να δουλεύει σωστά .htaccess αρχείο μας θα πρέπει επίσης να ενεργοποιήσουμε ένα module το οποίο ονομάζεται mod_rewrite από τις ρυθμίσεις του Apache. Ακόμα για περισσότερη ευκολία και λειτουργικότητα αλλάξαμε κάποιες ρυθμίσεις για να μην αναγκάζομαστε να γράφουμε συνεχώς στο URL το

```
http://localhost/beauty-plus.gr/index.php/main
```

μετατρέποντας το απλά σε:

```
http://beauty-plus.gr/
```

Αυτό το πετυχαίνουμε προσθέτοντας στον αρχείο hosts που βρίσκετε στην τοποθεσία C:\Windows\System32\drivers\etc\hosts\hosts την παρακάτω γραμμή κώδικα:

```
127.0.0.1 beauty-plus.gr
```

Και ακόμα στο αρχείο httpd-vhosts.conf που βρίσκετε στο C:\wamp\bin\apache\Apache2.2.11\conf\extra προσθέσαμε τις παρακάτω γραμμές κώδικα:

```
<VirtualHost *:80>
  DocumentRoot "C:/wamp/www/"
  ServerName localhost
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "C:/wamp/www/phpmyadmin"
  ServerName phpmyadmin
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "E:/www/beauty-plus.gr/"
  ServerName beauty-plus.gr
</VirtualHost>
```

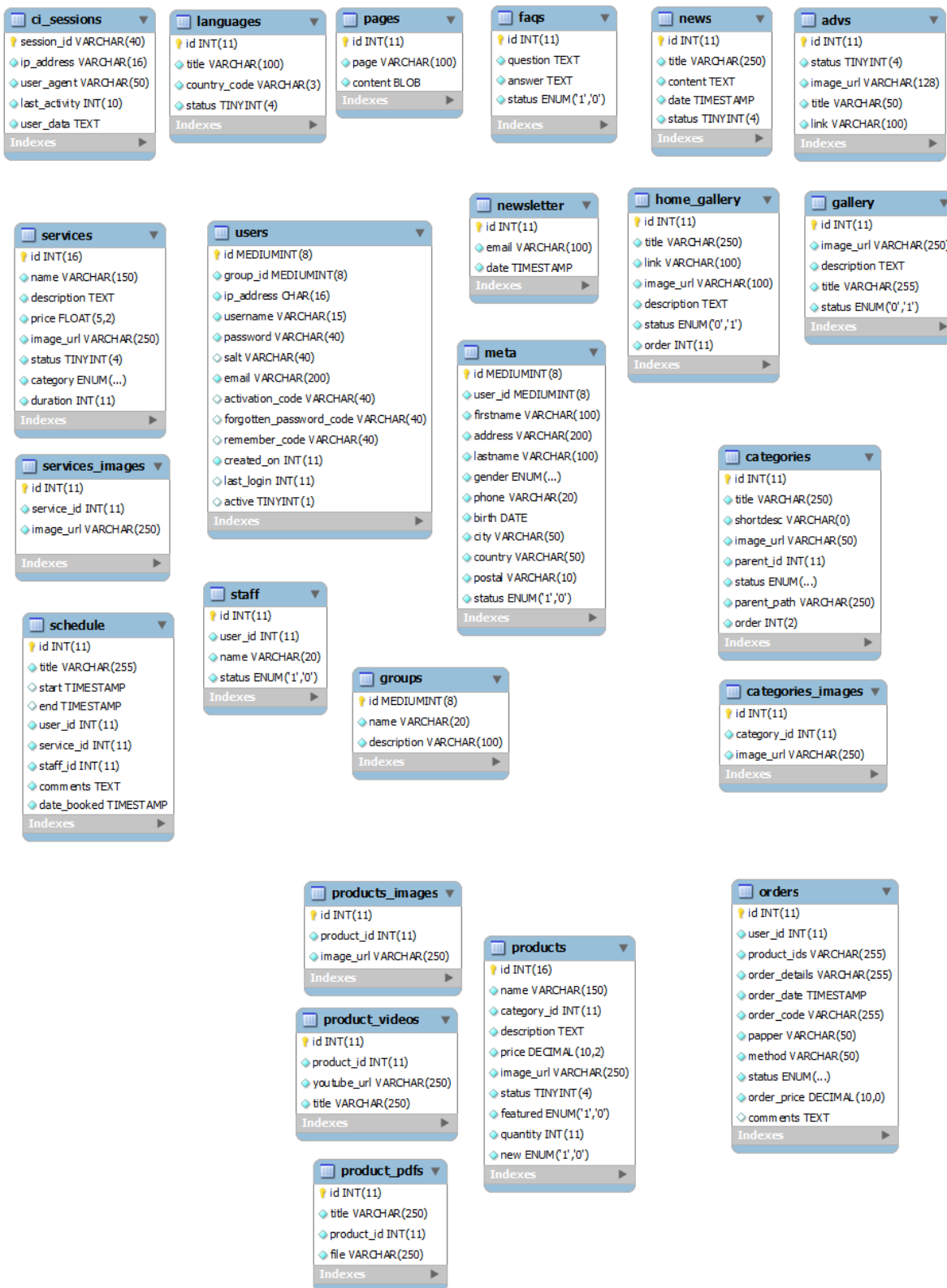
Με αυτόν τον τρόπο δημιουργήσαμε Virtual Hosts έτσι ώστε ο Apache να ξεγελάσει τον browser μας και να του φορτώσει αυτό που του λέμε εμείς έχοντας τακτοποιημένα τα μελλοντικά μας project και γενικότερα θα έχουμε μία πλήρη άποψη του τι συμβαίνει και πως τίποτα δεν είναι μαγικό.

Επίσης έχουμε δημιουργήσει στην ρίζα της εφαρμογής μας ένα φάκελο public και μέσα έχουμε δημιουργήσει τέσσερις υπό-φακέλους.

1. CSS : Όπου θα αποθηκεύονται τα στυλ και θα φορτώνονται δυναμικά σε κάθε view
2. Images: Όπου θα τοποθετούνται οι φωτογραφίες μας
3. JS : Όπου θα αποθηκεύονται Plugins της JQuery και custom scripts
4. Uploads: Οτιδήποτε γίνεται Upload μέσω της εφαρμογής μας.

***Στο root της εφαρμογής έχουμε ακόμα το φάκελο FCKeditor που περιέχει τον editor που θα χρησιμοποιηθεί στο backoffice και ένα φάκελο private.*

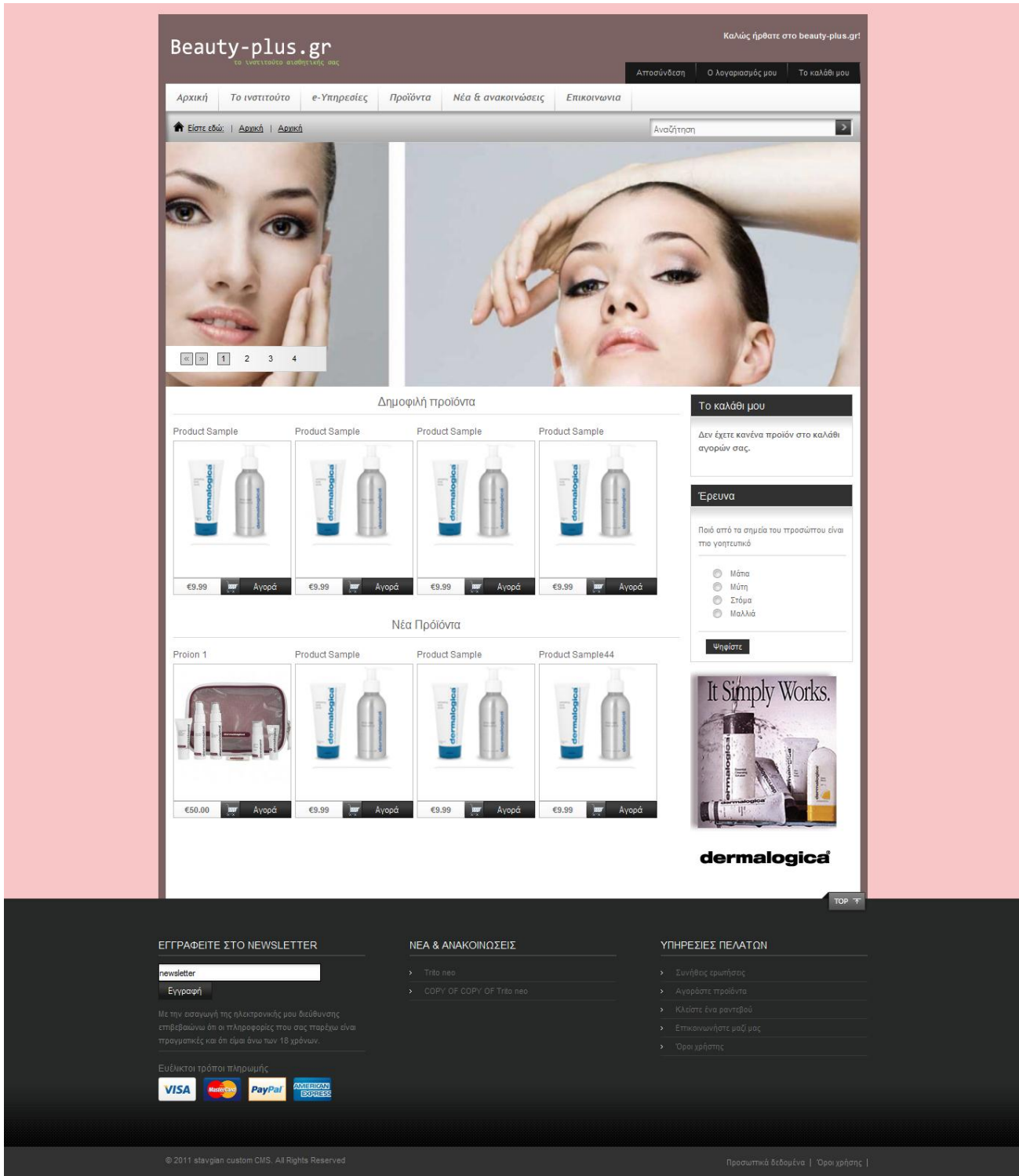
4.6. Οι πίνακες της βάση δεδομένων μας



Εικόνα 11 : Πίνακες στη βάση δεδομένων.

4.7 Το Frontend (Βασική σελίδα Επισκεπτών και μελών)

Η αρχική μας σελίδα η οποία είναι ένα Custom Template το οποίο φτιάχτηκε καθαρά από CSS. Ο CSS κώδικας βρίσκεται στο CD στο εσωτερικό του εξώφυλλου εμπεριέχετε , στο φάκελο X:\\beauty-plus.gr\\public\\css



Εικόνα 12 : Αρχική σελίδα ινστιτούτου αισθητικής

4.7.1 Ο main Controller του ινστιτούτου ⁽⁴⁾

Ο κεντρικός controller μας ο οποίος φορτώνει τις βασικές λειτουργίες – μενού του frontend σε συνεργασία πάντα με κάποιες βασικές βιβλιοθήκες και μοντέλα.

```
<?php
require_once APPPATH . "controllers/media.php";
require_once APPPATH . "libraries/cnode.php";
class Main extends Media {

function __construct(){
parent::__construct();
$this->_SetTemplate('beauty');
$this->load->helper('html');
$this->load->helper('common');
$this->_append_title('beauty-plus.gr');
error_reporting(E_ALL ^ (E_NOTICE | E_WARNING));
$this->load->model('products_model');
$this->load->model('beauty_model');
$data = (object)array();
}

function index()
{
$this->_prepend_title('Αρχική σελίδα');
$this->_setOutputData('breadCrumb', array((object)array(
'title'=> 'Αρχική',
'link'=>site_url('main/index')
)));
$data->services = $this->beauty_model->getAllServices();
$data->gallery = $this->beauty_model->home_gallery();
$data->footer_news=$this->beauty_model->footer_news();
$data->featured = $this->products_model->getMainFeature();
$data->advs = $this->beauty_model->advs();
$data->newest = $this->products_model->getNewProducts();

$this->load->view('inc/home',$data);
}

function search()
{
$this->_prepend_title('Αποτελέσματα αναζήτησης');
$this->_setOutputData('breadCrumb', array((object)array(
'title'=> 'Αποτελέσματα αναζήτησης',
'link'=>NULL
)));

if ($this->input->post('term'))
{
$data->results = $this->beauty_model->search($this->input->post('term'));
foreach ($data->results as $ok)
$data->parent = $this->db->get_where('categories',array('id'=>$ok['category_id']))->row();
}else redirect('main/index');

$data->services = $this->beauty_model->getAllServices();
$data->footer_news=$this->beauty_model->footer_news();

$this->load->view('inc/search',$data);
}

function institute()
{
$this->_prepend_title('To ινστιτούτο');
$this->_setOutputData('breadCrumb', array((object)array(
'title'=> 'To ινστιτούτο',
'link'=>site_url('main/institute')
)));
}
```

```

$data->gallery = $this->db->get('gallery', (object)array('gallery_id'))->result();
$data->services = $this->beauty_model->getAllServices();
$data->featured = $this->products_model->getMainFeature();
$data->footer_news=$this->beauty_model->footer_news();
$data->content= $this->db->get_where('pages', array('id'=>'1'))->row();
$this->load->view('inc/institute', $data);
}

function news()
{
    $data->services = $this->beauty_model->getAllServices();
    $data->featured = $this->products_model->getMainFeature();
    $data->footer_news=$this->beauty_model->footer_news();
    $data->advs=$this->beauty_model->advs();

    $data->posts=$this->db->get_where('news', array('status'=>'1'), $this->db->order_by('date', 'DESC'))->result();

    $this->prepend_title('Νέα & Ανακοινώσεις');
    $this->setOutputData('breadCrumb', array((object)array(
        'title'=> 'Νέα & Ανακοινώσεις',
        'link'=>site_url('main/news')
    )));

    $num_reviews = $this->db->count_all_results();

    /*Pagination */
    $this->load->library('pagination');
    $config['base_url'] = site_url().'/main/.'.__FUNCTION__.'.'/';
    $config['page_query_string'] = FALSE;
    $config['total_rows'] = $num_reviews;
    $config['per_page'] = 4;
    $config['uri_segment'] = 2;
    $this->config->set_item('enable_query_strings', 'FALSE');
    $this->pagination->initialize($config);

    $this->load->view('inc/news', $data);
}

function view_new($post_title=null, $post_id=null)
{
    if(empty($post_id))
        show_404();

    $this->load->helper('text');
    $data = (object)array();
    $data->services = $this->beauty_model->getAllServices();
    $data->advs = $this->beauty_model->advs();
    $data->footer_news=$this->beauty_model->footer_news();
    $data->featured = $this->products_model->getMainFeature();
    $this->db->select('*');
    $this->db->where('id', $post_id);
    $this->db->from('news');
    $data->posts = $this->db->get()->row();
    $this->append_title($data->posts->title);
    $this->setOutputData('breadCrumb', array(
        (object)array(
            'title'=> 'Νέα - Ανακοινώσεις',
            'link'=>site_url('main/news')
        ),
        (object)array(
            'title'=> $data->posts->title,
            'link'=>site_url('main/view_new/'.$post_id)
        )
    ));

    $this->load->view('inc/view_new', $data);
}

```

```

function products($category='Κατάλογος Προϊόντων', $category_id=1, $offset=0)
{
$this->load->model('products_model');
$data = (object)array();
$data->cart_products = $this->db->get('products')->result();
$table = 'categories';
$node = new cNode($category_id, $table);
$data->children = $node->getChildren();
foreach($data->children as $subCategory)
{
$ok = $this->db->get_where('categories_images', array('category_id'=>$subCategory->nodeId));
if($ok->num_rows > 0)
$subCategory->image = $ok->row()->image_url;
else
$subCategory->image = 'no_image1.jpg';
}

$breadcrumbStart = array((object)array(
    'title' => 'Κατάλογος Προϊόντων',
    'id' => 1
));

$data->breadcrumb = array_merge($breadcrumbStart, $node->getBreadcrumb($category_id));

//ola ta proionta tis katigorias(mazi me tis mesa katigories i oxi)
//$this->db->where('category_id', $category_id);

$childNodes = new cNode( $category_id, 'categories');
$nodes = $childNodes->getNodesToRoot();
$parent_path = implode("_", array_reverse($nodes));
$this->db->like('parent_path', $parent_path);
$cats = $this->db->get('categories');
if($cats->num_rows > 0)
{
    foreach($cats->result() as $my_category)
        $category_ids[] = $my_category->id;

    $this->db->where_in('category_id', $category_ids);
}

$this->db->where('status', '1');
$this->db->from('products');
$num_reviews = $this->db->count_all_results();

$this->load->library('pagination');

$config['base_url'] = site_url().'/main/.'.__FUNCTION__.'/.urlize($category)./'.'.$category_id;
$config['page_query_string'] = FALSE;
$config['total_rows'] = $num_reviews;
$config['per_page'] = 8;
$config['uri_segment'] = 5;
$this->config->set_item('enable_query_strings', 'FALSE');
$this->pagination->initialize($config);

$data->products = $this->products_model->getProduct($category_id, $config['per_page'], $offset = 0);
foreach($data->products as $prod)
{
$ok = $this->db->get_where('products', array('id'=>$prod->id));
if($ok->num_rows > 0)
$prod->image = $ok->row()->image_url;
else
$prod->image = 'no_image3.jpg';
}

foreach($data->breadcrumb as $branch)
$branch->link = site_url().'/main/products/'.urlize($branch->title)./'.'.$branch->id);

```

```

foreach($data->children as $branch)
$branch->link = site_url('/main/products/'.urlize($branch->title).'/'. $branch->nodeId);

$this->_setOutputData('breadCrumb', $data->breadCrumb);

$this->_prepend_title($data->breadCrumb[count($data->breadCrumb)-1]->title);
$data->services = $this->beauty_model->getAllServices();
$data->footer_news=$this->beauty_model->footer_news();
$data->advs = $this->beauty_model->advs();
$this->load->view('inc/products', $data);
}

function viewProduct($name = NULL,$new_id=null)
{
if(empty($new_id))
show_404();

$this->load->helper('text');
$data = (object)array();
$data->footer_news=$this->beauty_model->footer_news();
$data->services = $this->beauty_model->getAllServices();

$this->db->select('*');
$this->db->where('id', $new_id);
$this->db->from('products');
$data->new = $this->db->get()->row();
$data->images = $this->db->get_where('products_images', array('product_id'=>$new_id))->result();
$data->similar = $this->products_model->getSimilar($new_id);
$data->advs = $this->beauty_model->advs();

/* Δημιουργία breadcrumb tree κατηγοριών προϊόντων */
$node = new cNode($data->new->category_id, 'categories');
$breadCrumbStart = array((object)array(
    'title' => 'Κατάλογος Προϊόντων',
    'id' => 1
));
$breadCrumbEnd = array((object)array(
    'title' => $data->new->name,
    'id' => $data->new->id
));

$data->breadCrumb=array_merge($breadCrumbStart,$node->getBreadcrumb($data->new->category_id),$breadCrumbEnd);

$this->_prepend_title($data->new->name);
$this->_setOutputData('breadCrumb', $data->breadCrumb);

$this->load->view('inc/view_product', $data);
}

function viewService($name = NULL,$new_id=null)
{
if(empty($new_id))
show_404();

$this->load->helper('text');
$data = (object)array();
$data->footer_news=$this->beauty_model->footer_news();
$data->services = $this->beauty_model->getAllServices();

$this->db->select('*');
$this->db->where('id', $new_id);
$this->db->from('services');
$data->new = $this->db->get()->row();
$data->images = $this->db->get_where('services_images', array('service_id'=>$data->new->id))->result();

$this->_prepend_title($data->new->name);
$this->_setOutputData('breadCrumb', array(
(object)array(

```

```

        'title'=> 'e-Υπηρεσίες',
        'link'=> site_url('eservices/index')
    ),
    (object)array(
        'title'=> $data->new->name,
        'link'=> site_url('main/viewService/' . $data->new->name . '/' . $data->new->id)
    )
));
$this->load->view('inc/view_service', $data);
}

function faqs()
{
    $this->_prepend_title('Συχνές ερωτήσεις');
    $this->_setOutputData('breadCrumb', array((object)array(
        'title'=> 'Συχνές ερωτήσεις',
        'link'=>site_url('inc/faqs')
    )));
    $data->services = $this->beauty_model->getAllServices();
    $data->footer_news=$this->beauty_model->footer_news();
    $data->faqs = $this->db->get('faqs', (object)array('faqs_id'))->result();

    $this->load->view('inc/faqs', $data);
}

function terms()
{
    $this->db->where('id', 2);
    $data->content = $this->db->get('pages')->row();

    $data->services = $this->beauty_model->getAllServices();
    $data->footer_news=$this->beauty_model->footer_news();

    $this->_prepend_title($data->content->page);
    $this->_setOutputData('breadCrumb', array((object)array(
        'title'=> 'Όροι χρήσης',
        'link'=>site_url('main/contact')
    )));
    $this->load->view('inc/page', $data);
}

function contact()
{
    $this->db->where('id', 3);
    $data->content = $this->db->get('pages')->row();
    $data->services = $this->beauty_model->getAllServices();
    $data->footer_news=$this->beauty_model->footer_news();

    $this->_prepend_title($data->content->page);
    $this->_setOutputData('breadCrumb', array((object)array(
        'title'=> 'Επικοινωνία',
        'link'=>site_url('main/contact')
    )));
    /* Αν έχει πατησει να σταλεί μήνυμα */
    if ($this->is_ajax())
    {
        $this->load->helper('html');
        $message = '';
        foreach($_POST as $post => $value)
        {
            $data->{$post} = $this->input->post($post);
            if($post != 'code' && !empty($data->{$post}))
            $message .= "<b>".strtoupper($post)."</b>".br().$data->{$post}.br(2);
        }
        $fonts = array('./system/fonts/Jester.ttf');
        $this->load->library('PhpCaptcha', $fonts);
        $emailData = (object)array(
            'fromName' => $data->name,
            'fromEmail' => $data->email,
            'toName' => 'Yiannis',

```

```

        'toEmail' => 'stavgian@gmail.com',
        'subject' => 'Φόρμα επικοινωνίας | beauty-plus.gr',
        'message' => $message
    );

    if ($this->sendEmail($emailData))
    {
        $this->load->view('controls/send_success');
    }
    else
    {
        $this->load->view('controls/send_failure');
        $this->load->view('controls/reload_captcha');
    }
    }
    else $this->load->view('inc/contact', $data);
    }

    function sendContact()
    {
        $this->_setOutputMode(self::OUTPUT_NORMAL);

        $this->load->helper('html');
        $message = '';
        foreach($_POST as $post => $value)
        {
            $data->{$post} = $this->input->post($post);
            if($post != 'code' && !empty($data->{$post}))
            $message .= "<b>".strtoupper($post)."</b>".br().$data->{$post}.br(2);
        }

        $fonts = array('./system/fonts/Jester.ttf');
        $this->load->library('PhpCaptcha', $fonts);
        if($this->phpcaptcha->Validate($data->code) && check_email_address($data->email))
        {
            $emailData = (object)array(
                'fromName' => $data->name,
                'fromEmail' => $data->email,
                'toName' => 'Yiannis',
                'toEmail' => 'stavgian@gmail.com',
                'subject' => 'Φόρμα επικοινωνίας | beauty-plus.gr',
                'message' => $message
            );
            if ($this->sendEmail($emailData))
            {
                $this->load->view('controls/send_success');
            }
            else
            {
                $this->load->view('controls/send_failure');
                $this->load->view('controls/reload_captcha');
            }
        }
        else
        {
            $this->load->view('controls/send_failure');
            $this->load->view('controls/reload_captcha');
        }
    }

    function sendNewsletter($email = NULL)
    {
        $this->_setOutputMode(self::OUTPUT_NORMAL);
        //$this->_setOutputMode(NULL);
        $email = $this->input->post('email');
        if(check_email_address($email))
        {
            $exist = $this->db->get_where('newsletter', array('email'=>$email));
            if($exist->num_rows() == 0)
            $this->db->insert('newsletter', array('email'=>$email));
            $message = "To e-mail: $email έχει ήδη γραφτεί στο Newsletter.";
        }
    }

```

```

        $emailData = (object)array(
            'fromName' => 'Beauty-plus.gr Newsletter',
            'fromEmail' => $email,
            'toName' => 'Yiannis Stav',
            'toEmail' => 'stavgian@gmail.com',
            'subject' => 'Newsletter Form | beauty-plus.gr',
            'message' => $message
        );
        if ($this->sendEmail($emailData))
            $this->load->view('controls/newsletter_success');
        else $this->load->view('controls/newsletter_failure');
    }else $this->load->view('controls/newsletter_failure');
}

function unsubscribeNewsletter()
{
    $this->_setOutputMode(self::OUTPUT_NORMAL);
    //$this->_setOutputMode(NULL);
    $email = $this->input->post('email');
    $this->db->get_where('newsletter', array('email'=>$email));
    $this->db->delete('newsletter', array('email'=>$email));
}

function featured_products()
{
    $this->_prepend_title('Δημοφιλή προϊόντα');
    $this->_setOutputData('breadCrumb', array((object)array(
        'title'=> 'Δημοφιλή προϊόντα',
        'link'=>site_url('main/featured_products')
    )));
    //$data->featured = $this->products_model->getMainFeature();
    $data->footer_news=$this->beauty_model->footer_news();
    $data->services = $this->beauty_model->getAllServices();
    $data->advs = $this->beauty_model->advs();

    $this->db->where('status', '1');
    $this->db->where('featured', '1');
    $this->db->from('products');
    $num_reviews = $this->db->count_all_results();

    $this->load->library('pagination');

    $config['base_url'] = site_url().'main/.__FUNCTION__';
    $config['page_query_string'] = FALSE;
    $config['total_rows'] = $num_reviews;
    $config['per_page'] = 8;
    $config['uri_segment'] = 3;
    $this->config->set_item('enable_query_strings', 'FALSE');
    $this->pagination->initialize($config);
    $this->db->order_by('id', 'RANDOM');

    $data->featured = $this->db->get('products', $config['per_page'], $this->uri->segment(3))->result();
    $this->load->view('inc/featured_products',$data);
}

function newest_products()
{
    $this->_prepend_title('Νέα Προϊόντα');
    $this->_setOutputData('breadCrumb', array((object)array(
        'title'=> 'Νέα προϊόντα',
        'link'=>site_url('main/newest_products')
    )));

    //$data->newest = $this->products_model->getNewProducts();
    $data->featured = $this->products_model->getMainFeature();
    $data->footer_news=$this->beauty_model->footer_news();
    $data->services = $this->beauty_model->getAllServices();
    $data->advs = $this->beauty_model->advs();
    $this->db->where(array('status'=>'1', 'new'=>'1'));
    $this->db->from('products');
}

```



```
$num_reviews = $this->db->count_all_results();
$this->load->library('pagination');

$config['base_url'] = site_url().'/main/'.__FUNCTION__;
$config['page_query_string'] = FALSE;
$config['total_rows'] = $num_reviews;
$config['per_page'] = 8;
$config['uri_segment'] = 3;
$this->config->set_item('enable_query_strings', 'FALSE');
$this->pagination->initialize($config);

$this->db->order_by('id', 'ASC');
$data->newest = $this->db->get('products', $config['per_page'], $this->uri->segment(3))->result();

$this->load->view('inc/newest_products',$data);
}
}
```

4.7.2 To Custom Template του Front end μας. ⁽⁷⁾

- To HEAD

```

<!-- Loading the Doctype From HTML Library -->
<?php echo doctype('html1-trans');?>
<html>
<!-- Start Of Head -->
<head>
<!-- FAVICON for our Page -->
<link href="<?php echo base_url();?>public/images/favicon/favicon.ico" rel="icon" type="image/x-
icon" />
<!-- Charset Used -->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- Site Title -->
<title><?php echo $title;?></title>
<!-- CSS USED -->
<link href="<?php echo base_url();?>public/css/style.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/960.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/footer.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/default.advanced.css" rel="stylesheet" type="text/css"
/>
<link href="<?php echo base_url();?>public/css/ui-lightness/jquery-ui-1.8.7.custom.css" rel="stylesheet"
type="text/css" />
<link href="<?php echo base_url();?>public/css/tooltip.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/slider.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/jquery.ad-gallery.css" rel="stylesheet" type="text/css"
/>
<link href="<?php echo base_url();?>public/css/fancy.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/contentslider.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/jquery.weekcalendar.css" rel="stylesheet" type="text/css"
/>
<link href="<?php echo base_url();?>public/css/home_rotator.css" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url();?>public/css/fullcalendar.css" rel="stylesheet" type="text/css" />
<!-- JQuery -->
<!-- load jQuery 1.1.3 -->
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery-1.5.1.min.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery-1.4.4.min.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery-1.4.3.min.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery-ui-
1.8.7.custom.min.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.easing.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/contentslider.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/anythingslider.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.anythingslider.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/cufon-yui.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.ad-gallery.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/thumbgallery.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/cufon.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.validate.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.form.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.jBreadCrumb.1.1.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.fancybox.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/ddaccordion.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.ad-gallery.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.tools.min.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/fullcalendar.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.blockUI.js"></script>
<!-- Wizard -->
<script type="text/javascript" src="<?php echo base_url();?>public/js/hoverIntent.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/wizardPro.min.js"></script>
<script type="text/javascript" src="<?php echo base_url();?>public/js/wizardPro.js"></script>
<link href="<?php echo base_url();?>public/css/wizardPro.css" rel="stylesheet" type="text/css" /> <!--
Topsy -->
<script type="text/javascript" src="<?php echo base_url();?>public/js/jquery.tipsy.js"></script>
<link href="<?php echo base_url();?>public/css/tipsy.css" rel="stylesheet" type="text/css" />

```

```
<?php foreach($js as $js_file): ?>
<script type="text/javascript" src="<?php echo $js_file; ?>"></script>
<?php endforeach; ?>

<script type="text/javascript" language="javascript">
var baseURL = "<?php echo base_url()?>";
var message = "<?php echo mysql_real_escape_string($this->load->view('loading/default', array(),
true));?>";
var cssRules = {
    border: 'none',
    width: '200px',
    margin: 'auto',
    padding: '15px',
    backgroundColor: '#000',
    '-webkit-border-radius': '10px',
    '-moz-border-radius': '10px',
    opacity: '.5',
    color: '#fff'
};
var block_options = {
    message: message,
    css: cssRules
};
var myURL = location.href;
</script>
<!--[if IE 6]>
<style>
    img, div, td { behavior: url(iepngfix.htc); }
</style>
<![endif]-->
<!-- End of Head -->
</head>
```

- **TO BODY*****

```
<body>
<a name="top"></a>
<!-- Start Of Main -->
<div id="wrapper_sec">
    <!-- Loading inc/header -->
    <?php $this->load->view('inc/header');?>
    <div id="content_sec">
        <?php echo $output;?>
    </div>
</div><!-- End Of wrapper_sec -->
<div class="clear"></div>
<!-- Loading inc/footer -->
<div id="footer">
<?php echo $this->load->view('inc/footer');?>
</div>
</body>
</html>
```

****Όπως μπορούμε να δούμε το head είναι πολύ μεγαλύτερο από το body αλλά επειδή γενικότερα δουλεύουμε περισσότερο στο body μας είναι πολύ πιο εύκολο να δουλέψουμε πάνω σε 10 γραμμές κώδικα.*

4.7.3 Τα Views μας ⁽⁸⁾

- **Inc/header.php**

Περιέχει το πάνω μέρος της σελίδας και τα κάποια JQuery Plugins αλλά και scripts

```

<script type="text/javascript">
$(document).ready(function(){
    //EXPOSE
    // execute your scripts when the DOM is ready. this is a good habit
    // expose the form when it's clicked or cursor is focused
    var form = $(".expose").bind("click keydown", function() {
    $(this).expose({
    // when exposing is done, change form's background color
    onLoad: function() {
    form.css({backgroundColor: '#c7f8ff'});
    },
    // when "unexposed", return to original background color
    onClose: function() {
    form.css({backgroundColor: null});
    }
    });
});

$(".small_tooltip").tipsy({gravity: 's',fade: true,trigger: 'focus'});
var link = "<?php echo site_url()?>";
$("#radio").buttonset();
$(".a.fancybox").fancybox();
$("#dialog:ui-dialog").dialog("destroy");

$("#dialog-AddCart").dialog({
    modal: true,
    disabled: true,
    autoOpen: false,
    buttons: {
        Ok: function() {
            location.reload();
            $(this).dialog("close");
            //location.href="<?php echo site_url('cart/index'); ?>";
        }
    }
});

$(".addForm").submit(function() {
    // Get the product ID and the quantity
    var id = $(this).find('input[name=product_id]').val();

    $.ajax({

        url: "<?php echo site_url('cart/add_cart_item/'); ?>",
        type: 'POST',
        data: {product_id: id, ajax: '1'},
        success: function() {

            $('#dialog-AddCart').dialog('open');

        }
    });
    return false;
});
});

</script>
<div id="dialog-AddCart" title="Προσθήκη στο καλάθι">
<p>
<span class="ui-icon ui-icon-circle-check" style="float:left; margin:0 7px 50px 0;"></span>
Το προϊόν προστέθηκε στο καλάθι σας.
</p>
</div>
<div id="head"><!-- Start Of Header -->

```

```

<div class="logo"><!-- Logo -->
  <a href="<?php echo site_url(); ?>"><img alt="<?php echo $title;?>" src="<?php echo
base_url();?>public/images/logo.png"></a>
</div>
  <div class="rightnavi">
    <p class="bold right">Καλώς ήρθατε στο beauty-plus.gr!</p>
<div class="clear"></div>
<ul>
<?php if ($this->session->userdata('email')):?>
<li><a href="<?php echo site_url('auth/logout');?>" class="small_tooltip" title="Αποσύνδεση - <?php
echo $this->session->userdata('email');?>">Αποσύνδεση</a></li>
<li><a href="<?php echo site_url('members/index');?>" class="small_tooltip" title="Μετάβαση στην
περιοχή μέλους">Ο λογαριασμός μου</a></li>
<?php endif;?>

<li><a href="<?php echo site_url('cart');?>" class="small_tooltip" title="<?php echo $this->cart-
>total_items();?> Προϊόντα">Το καλάθι μου</a></li>
<?php if (!$this->session->userdata('email')):?>
<li><a href="<?php echo site_url('auth/create_user');?>">Δημιουργία λογαριασμού</a></li>
<li><a href="<?php echo site_url('auth/index');?>">Είσοδος Χρήστη</a></li>
<?php endif;?>
</ul>
  </div>
  <div class="clear"></div>

<div class="navigation">
<ul id="nav" class="dropdown dropdown-linear dropdown-columnar">
  <li><a href="<?php echo site_url('main/index');?>">Αρχική</a></li>
  <li><a href="<?php echo site_url('main/institute');?>">Το ινστιτούτο</a></li>
  <li class="dir"><a href="<?php echo site_url('eservices/index');?>">e-Υπηρεσίες</a>
</li>
<li class="dir"><h6>Πρόσωπο</h6>
  <ul>
    <?php foreach ($services as $service):?>
      <?php if ($service->category == 'face'):?>
<li><a href="<?php echo site_url('main/viewService/'.urlize($service->name).'/'. $service->id);?>">
<?php echo $service->name;?></a></li>
      <?php endif;?>
    <?php endforeach;?>
  </ul>
</li>
  <li class="dir"><h6>Σώμα</h6>
  <ul>
    <?php foreach ($services as $service):?>
      <?php if ($service->category == 'body'):?>
<li><a href="<?php echo site_url('main/viewService/'.urlize($service->name).'/'. $service->id);?>">
<?php echo $service->name;?></a></li>
      <?php endif;?>
    <?php endforeach;?>
  </ul>
</li>
  <li class="dir"><h6>Αποτριχώση</h6>
  <ul>
    <?php foreach ($services as $service):?>
      <?php if ($service->category == 'hairs'):?>
<li><a href="<?php echo site_url('main/viewService/'.urlize($service->name).'/'. $service->id);?>">
<?php echo $service->name;?></a></li>
      <?php endif;?>
    <?php endforeach;?>
  </ul>
</li>
  <li class="dir"><a href="<?php echo site_url('main/products/Katalogos-
Proiontw/1');?>">Προϊόντα</a>
  <ul>
  </ul>
</li>
  <li><a href="<?php echo site_url('main/contact');?>">Επικοινωνία</a></li>
</ul>

```

```

</div>

</div><!-- End Of Head -->
<div class="clear"></div>
<div id="crumb"
<ul class="left">
  <li><p>Είστε εδώ:</p></li>
  <li><a href="<?php echo site_url()?>" title="Αρχική"> Αρχική </a></li>
  <?php foreach($breadCrumb as $node):?>
  <li><a href="<?php echo $node->link?>" <?php echo $node->title?> </a></li>
  <?php endforeach;?>
</ul>
<ul class="search right">
<form action="<?php echo site_url('main/search');?>" method="post">
<li><input name="term" value="Αναζήτηση" class="bar small_tooltip" title="Αναζήτηση περιεχομένου"
type="text" onblur="if(this.value == '') { this.value = 'Αναζήτηση'; }" onfocus="if(this.value ==
'Αναζήτηση') { this.value = ''; }"></li>
<li><input class="go" value="" type="submit"></li>
</form>
</ul>
</div>
<div class="clear"></div>

```

- **\$output** ⁽⁹⁾

Περιέχει την έξοδο και το αποτέλεσμα που θα έχουμε οπτικά και γενικότερα για τις λειτουργίες της σελίδας μας. Προέρχεται από τον Controller όπου εξαρτάται από Μοντέλα – Βιβλιοθήκες – Config files – Helpers – hooks - plugins και πολλά άλλα.

- **Inc/footer.php** ⁽¹⁰⁾

Το κάτω μέρος της σελίδας μας με ενσωματωμένες λειτουργίες όπου νέα και ανακοινώσεις, newsletter, και επιπλέον μενού πλοήγησης.

```

<div class="foot_inr">
  <div class="topdiv">
    <a href="#top" class="top">&nbsp;</a>
  </div>
  <script>
    $(document).ready(function(){
      $('#newsletter_form').validate({
        errorLabelContainer: "#newsletter_error_messages",
        submitHandler: function(form){
          $("#newsletter_messages").html('');
          $("#newsletter_messages").slideDown();
          $(form).ajaxSubmit({
            success: function(responseText, textStatus){
              $("#newsletter_messages").html(responseText);
            }
          });
          $(form).resetForm();
          return false;
        },
        showErrors: function(errorMap, errorList) {
          $("#newsletter_messages").slideUp();
          this.defaultShowErrors();
        }
      });
    });
  </script>
  <div class="emailsignup">
    <h5>Εγγραφείτε στο newsletter</h5>

```

```

<form action="<?php echo site_url('main/sendNewsletter')?>"method="POST" id="newsletter_form"
name="newsletter_form">
  <ul class="inp">
    <li><input name="email" type="text" class="bar required small_tooltip" title="Παρακαλώ εισάγετε ένα
έγκυρο e-mail" value="newsletter" onblur="if(this.value == '') { this.value = 'newsletter'; }"
onfocus="if(this.value == 'newsletter') { this.value = ''; }" /></li>
    <li><input type="submit" class="simplebtn" value="Εγγραφή" /></li>
  </ul>
  <div id="newsletter_error_messages"></div>
  <div id="newsletter_messages"></div>
</form>
<div class="clear"></div>
<p class="signtxt">
  Με την εισαγωγή της ηλεκτρονικής μου διεύθυνσης επιβεβαιώνω ότι οι πληροφορίες που σας
παρέχω είναι πραγματικές και ότι είμαι άνω των 18 χρόνων.
</p>
<h6>Ευέλικτοι τρόποι πληρωμής</h6>
<ul class="cards">
  <li><a href="#"></a></li>
  <li><a href="#"></a></li>
  <li><a href="#"></a></li>
  <li><a href="#"></a></li>
</ul>
</div>

<div class="partners">
<?php $this->load->view('inc/footer_news');?>
</div>
<div class="partners">
<h5>Υπηρεσίες πελατών</h5>
  <ul>
    <li><a href="<?php echo site_url('main/faqs');?>">Συνήθειες ερωτήσεων</a></li>
    <li><a href="<?php echo site_url('main/products');?>">Αγοράστε προϊόντα</a></li>
    <li><a href="<?php echo site_url('eservices');?>">Κλείστε ένα ραντεβού</a></li>
    <li><a href="<?php echo site_url('main/contact');?>">Επικοινωνήστε μαζί μας</a></li>
    <li><a href="<?php echo site_url('main/terms');?>">Όροι χρήστης</a></li>
  </ul>
</div>
<div class="clear"></div>
</div>
<div class="clear"></div>
<div class="copyrights">
<p>© 2011 Stavgianoudakis Yiannis. All Rights Reserved</p>
  <div class="footer_links">
    <ul>
      <li><a href="<?php echo site_url('main/terms');?>">Προσωπικά δεδομένα</a></li>
      <li><a href="<?php echo site_url('main/terms');?>">Όροι χρήσης</a></li>
    </ul>
  </div>
</div>
<div class="clear"></div>
</div>

```


5 Επέκταση του Codelgniter

5.1 Το BackOffice (Stavgian Custom Management System)

Εικόνα 13 : BackOffice - Custom CMS Αρχική

5.2 Βασικά Χαρακτηριστικά StavgianCMS

- Προβολή στατιστικών στοιχείων
- Εύκολη και γρήγορη πλοήγηση
- Δημιουργήθηκε από το CodeIgniter
- Διαχείριση Ραντεβού
- Διαχείριση Παραγγελιών
- Διαχείριση Προϊόντων και κατηγοριών
- Διαχείριση Gallery και φωτογραφιών κατηγοριών και προϊόντων
- Διαχείριση Newsletter
- Διαχείριση Χρηστών
- Διαχείριση περιεχομένου σελίδων
- Εξαγωγή αναφορών με κριτήρια σε pdf excel csv κ.α

5.3 Screenshots

Παραγγελίες

Διαχείριση παραγγελιών									
ID	Κωδικός	Ημ/νία	Κατάσταση	Όνομα	Τηλ.επκ	Μέθοδος	Παραστατικό	Λεπτομέρειες	Σχόλια
66	reM3U	2011-03-25 18:19:29	waiting	Stavgianoudakis	6941665	Πιστωτική κάρτα	Απόδειξη λιστ	Yiannis,Stavgianoudakis,St.Kelaidi	GIANNIS
67	fSH6O	2011-03-25 22:56:00	completed	Stavgianoudakis	6941665	Αντικαταβολή	Απόδειξη λιστ	Yiannis,Stavgianoudakis,Ano Valsan	Giannis
68	oSGol	0000-00-00 00:00:00	shopping	Stavgianoudakis	6941665			Yiannis,Stavgianoudakis,St.Kelaidi	
69	sesv6	0000-00-00 00:00:00	shopping	Stavgianoudakis	6941665			Yiannis,Stavgianoudakis,St.Kelaidi	
70	9aEYf	0000-00-00 00:00:00	shopping	Stavgianoudakis	6941665			Yiannis,Stavgianoudakis,St.Kelaidi	
71	Gc9Jh	0000-00-00 00:00:00	shopping	Stavgianoudakis	6941665			Yiannis,Stavgianoudakis,St.Kelaidi	
72	vX4Wj	0000-00-00 00:00:00	shopping	Stavgianoudakis	6941665			Yiannis,Stavgianoudakis,St.Kelaidi	
73	zIXYX	2011-03-27 15:09:46	completed	Stavgianoudakis	6941665	Πιστωτική κάρτα	Τιμολόγιο	Yiannis,Stavgianoudakis,Ano Valsan	ffdgfggh

Εικόνα 14 : Crud παραγγελιών με Χρήση JQGRID

Πληροφορίες:
Κάθε αλλαγή που γίνει επιτρέπει και έχει άμεση συσχέτιση με την βάση δεδομένων μας.

Δημιουργία κατηγορίας

όνομα κατηγορίας Προσθήκη

Διαχείριση κατηγοριών προϊόντων Μετακίνηση Διαγραφή

Κατηγορία

- Κατάλογος Προϊόντων
- Φροντίδα σώματος
 - Περιποίηση άκρων
 - Μανικαούρ
 - Πεντικαούρ
- Καλυντικά
 - Ανδρική σειρά
 - Ξύρισμα
- Συσκευασίες δώρου
 - Αρωματικά κερία
- Μαγιιάζ
- Copsiller 1
- Πούδρες

Εικόνα 15 : Διαχείριση κατηγοριών

5.4 Ένα δείγμα από τα βασικά μοντέλα μας. (MODELS)⁽⁹⁾

5.4.1 Beauty_model.php (Γενικό μοντέλο του Ινστιτούτου Αισθητικής)

```

<?php
class Beauty_model extends Model
{
    function __construct()
    {
        parent::Model();
    }

    function search($term)
    {
        $data = array();
        $this->db->select("*");
        $this->db->like('name',$term);
        $this->db->orlike('description',$term);
        $this->db->orlike('id',$term);
        $this->db->orderby('price','asc');
        $this->db->where('status','1');
        $this->db->limit(25);
        $Q = $this->db->get('products');
        if ($Q->num_rows() > 0){
            foreach ($Q->result_array() as $row){
                $data[] = $row;
            }
        }
        $Q->free_result();
        return $data;
    }

    //Custom Query return row()
    function CustomQuery($sql)
    {
        $query = $this->db->query($sql);
        return $query;
    }

    function footer_news()
    {
        $this->db->select("*");
        $this->db->orderby('id','asc');
        $this->db->where('status','1');
        $this->db->limit(7);
        $query = $this->db->get('news');
        if ($query->num_rows() > 0)
        {
            return $query->result();
        }
    }

    function home_gallery()
    {
        $this->db->select("*");
        $this->db->orderby('order','asc');
        $this->db->where('status','1');
        $query = $this->db->get('home_gallery');
        if ($query->num_rows() > 0)
        {
            return $query->result();
        }
    }

    function getAllServices()
    {
        $this->db->select('*');
        $this->db->where('status','1');
        return $result=$this->db->get('services')->result();
    }
}

```

```

    }

    function getStaff()
    {
        $this->db->select('*');
        $this->db->where('status','1');
        return $result = $this->db->get('staff')->result();
    }

    function advs()
    {
        $this->db->select('*');
        $this->db->where('status','1');
        $this->db->limit(1);
        $this->db->order_by('id','RANDOM');
        return $result = $this->db->get('advs')->result();
    }
}

```

Crud_model (Συνεργάζεται με την βιβλιοθήκη Crud με στόχο crud functions στο Backoffice) ⁽⁹⁾
 <?php

```

class Crud_model extends Model {

    function __construct(){
        parent::Model();
    }

    function add($table, $data=null, $id=null)
    {
        if(!empty($_FILES))
        {
            if(isset($_FILES['image-file'])) // to vazei o rich text editor!
                unset($_FILES['image-file']);

            $upload_status = '<br />';
            foreach($_FILES as $name => $file)
            {
                $upload_response = $this->upload_file($name);
                if($upload_response->success == true)
                {
                    $data->{$name} = $upload_response->message['file_name'];
                    $upload_status .= "<b>[".$name."</b>: ".$upload_response->message['orig_name']. " έχει  

ανέβει!<br />";
                }
                else
                {
                    $upload_status .= "<b>[".$name."</b>: ".strip_tags($upload_response->message). "<br />";
                }
            }
            $upload_status .= '<br />';
        }

        $this->db->trans_start();

        if(isset($data->lang_fields))
        {
            $lang_table = $table.'_lang';
            $lang_fields = $data->lang_fields;
            unset($data->lang_fields);
        }

        if(isset($lang_fields))
            foreach($lang_fields as $language=>$fields)
                foreach($fields as $field=>$value)
                    if($language == 'english')
                        {
                            if(empty($value))
                                $value = 'empty';
                            $data->{$field} = trim($value);
                        }
        }
    }
}

```

```

    }

    $this->db->insert($table, $data);
    $last_id = $this->db->insert_id();

    if(isset($lang_fields))
    {
        foreach($lang_fields as $language => $language_fields)
        {
            $lang_data = (object)array();
            $lang_data->lang_id = $this->db->get_where('languages', array('title'=>$language))->row()->id;
            $lang_data->field_id = $last_id;
            foreach($language_fields as $field_name => $field_value)
                $lang_data->{$field_name} = trim($field_value);

            $this->db->where('field_id', $lang_data->field_id);
            $this->db->where('lang_id', $lang_data->lang_id);
            $check = $this->db->get($lang_table);

            if($check->num_rows() > 0)
            {
                $this->db->where('field_id', $lang_data->field_id);
                $this->db->where('lang_id', $lang_data->lang_id);
                $this->db->update($lang_table, $lang_data);
            }else
                $this->db->insert($lang_table, $lang_data);
        }
    }

    $this->db->trans_complete();

    $this->session->set_flashdata($table.'Highlight', $last_id);
    show_alert($table,'Επιτυχία',$upload_status.'Εγινε εισαγωγή στην βάση δεδομένων!');
}

function edit($table, $data=null, $id=null)
{
    if(!empty($_FILES))
    {
        if(isset($_FILES['image-file'])) // to vazai o rich text editor!
            unset($_FILES['image-file']);

        $upload_status = '<br />';
        foreach($_FILES as $name => $file)
        {
            $upload_response = $this->upload_file($name);
            if($upload_response->success == true)
            {
                $data->{$name} = $upload_response->message['file_name'];
                $upload_status .= "<b>[".$name."</b>: ".$upload_response->message['orig_name']." έχει  
ανέβει!<br />";
            }else
                $upload_status .= "<b>[".$name."</b>: ".strip_tags($upload_response->message)."<br />";
        }
        $upload_status .= '<br />';
    }
    $this->db->trans_start();

    if(isset($data->lang_fields))
    {
        $lang_table = $table.'_lang';
        $lang_fields = $data->lang_fields;
        unset($data->lang_fields);

        foreach($lang_fields as $language => $language_fields)
        {
            $lang_data = (object)array();
            $lang_data->lang_id = $this->db->get_where('languages', array('title'=>$language))-
>row()->id;

```

```

$lang_data->field_id = $id;
foreach($language_fields as $field_name => $field_value)
    {
        if($language == 'english')
            $data->{$field_name} = trim($field_value);
            $lang_data->{$field_name} = trim($field_value);
    }

$this->db->where('field_id', $lang_data->field_id);
$this->db->where('lang_id', $lang_data->lang_id);
$check = $this->db->get($lang_table);

if($check->num_rows() > 0)
    {
        $this->db->where('field_id', $lang_data->field_id);
        $this->db->where('lang_id', $lang_data->lang_id);
        $this->db->update($lang_table, $lang_data);
    }else
        $this->db->insert($lang_table, $lang_data);
    }

}

if(count(object_to_array($data)) != 0 && !is_null($id))
{
    $this->db->where('id', $id);
    $this->db->update($table, $data);
}

$this->db->trans_complete();

$this->session->set_flashdata($table.'Highlight', $id);

show_alert($table, 'Επιτυχία', $upload_status.' Η εγγραφή ανανεώθηκε!');
}

function delete($table, $data=null, $id=null)
{
    $delete_status = '';
    $count = 0;

    if(is_null($id) && empty($_POST))
        show_alert($table, 'Αποτυχία', 'Δεν επιλέξατε εγγραφές για διαγραφή!');
    else if(!is_null($id))
    {
        $count = 1;
        $delete_status .= $this->delete_row($table, $id);
    }else if(is_null($id) && !empty($_POST))
        foreach($_POST as $id => $value)
        {
            $count++;
            $delete_status .= $this->delete_row($table, $id);
        }
    }
    else
        show_error('Improper use of delete mode');

    show_alert($table, 'Επιτυχία', $delete_status.$count.' εγγραφές έχουν σβηστεί!');
}

function delete_row($table, $id=null)
{
    $lang_table = $table.'_lang';
    $fields = $this->db->field_data($table);

    $this->db->where('id', $id);
    $row = $this->db->get($table)->row();

    foreach($fields as $field)
        if(strpos($field->name, '_url') || strpos($field->name, '_file'))
        {
            $file = $row->{$field->name};
            if($this->delete_file($file))

```



```

        $delete_status .= "<b>[".$field->name."</b>: ".$file." διαγράφηκε!<br />";
    else
    $delete_status .= "<b>[".$field->name."</b>: ".$file." δεν είναι δυνατή η διαγραφή!<br />";
    }

    $this->db->trans_start();

    if($this->db->table_exists($lang_table))
    {
        $this->db->where('field_id', $id);
        $this->db->delete($lang_table);
    }

    $this->db->where('id', $id);
    $this->db->delete($table);

    $this->db->trans_complete();

    return $delete_status;
}

function duplicate($table, $data=null, $id=null)
{
    $lang_table = $table.'_lang';

    $fields = $this->db->field_data($table);

    $this->db->where('id', $id);
    $row = $this->db->get($table)->row();

    foreach($fields as $field)
    {
        if(strpos($field->name, '_url') || strpos($field->name, '_file'))
        {
            $file = $row->{$field->name};
            $new_file = $this->duplicate_file($file);
            if($new_file !== false)
            {
                $duplicate_status .= "<b>[".$field->name."</b>: ".$file." κλωνοποιήθηκε!<br />";
                $row->{$field->name} = $new_file;
            }else
            $duplicate_status .= "<b>[".$field->name."</b>: ".$file." δεν είναι δυνατή η κλωνοποίηση!<br
/>";
        }
        if($field->name == title || $field->name == name)
        {
            $row->{$field->name} = 'COPY OF '.$row->{$field->name};
        }
    }

    unset($row->id);

    $this->db->trans_start();

    $this->db->insert($table, $row);
    $duplicate_id = $this->db->insert_id();

    if($this->db->table_exists($lang_table))
    {
        $this->db->where('field_id', $id);
        $lang_rows = $this->db->get($lang_table)->result();

        foreach($lang_rows as $lang_row)
        {
            unset($lang_row->id);
            $lang_row->field_id = $duplicate_id;

            $this->db->insert($lang_table, $lang_row);
        }
    }
}

```

```

$this->db->trans_complete();

$this->session->set_flashdata($table.'Highlight', $duplicate_id);
show_alert($table, 'Επιτυχία', $duplicate_status.'η εγγραφή κλωνοποιήθηκε!');
}

function changeStatus($table, $id=null)
{
    $this->db->where('id', $id);
    $status = $this->db->get($table)->row()->status;

    $this->db->where('id', $id);
    if($status == '1')
        $statusData = array('status'=>'0');
    else
        $statusData = array('status'=>'1');

    $this->db->update($table, $statusData);

    die(json_encode($statusData));
}

function upload_file($file)
{
    $config['upload_path'] = './public/uploads/';
    $config['allowed_types'] = 'gif|jpg|png|pdf|doc|html|flv|mpg|mpeg|mpe|qt|mov|avi|movie';
    $config['max_size'] = '100000';
    $config['encrypt_name'] = true;
    $config['remove_spaces'] = true;

    $this->load->library('upload', $config);
    $data = (object)array(
        'success'=>true,
        'message'=>'
    );

    if (!$this->upload->do_upload($file))
    {
        $data->message = $this->upload->display_errors();
        $data->success = false;
    }else
        $data->message = $this->upload->data();

    return $data;
}

function delete_file($file)
{
    $folder = './public/uploads/';

    if(unlink($folder.$file))
        return true;

    return false;
}

function duplicate_file($file)
{
    $folder = './public/uploads/';
    $x = explode('.', $file);
    $extension = ".".end($x);
    $this->load->helper('string');

    $newfile = random_string('unique');

    if (!copy($folder.$file, $folder.$newfile.$extension))
        return false;

    return $newfile.$extension;
}

```



```
}

function getMainFeature(){

    $this->db->select("*");
    $this->db->where('featured', 'true');
    $this->db->where('status', '1');
    $this->db->orderby("rand()");
    $this->db->limit(30);
    $query = $this->db->get('products');
    if ($query->num_rows() > 0)
    {
        return $query->result();
    }
    else return FALSE;
}

function getRandomProducts($limit)
{
    $this->db->select("*");
    $this->db->orderby('id', "asc");
    $this->db->limit(30);
    $Q = $this->db->get('products');
    if ($Q->num_rows() > 0)
    {
        return $Q->result();
    }
    }else return FALSE;
}

}
```

5.5 Custom Βιβλιοθήκες (Libraries)⁽⁹⁾

Για το αποτέλεσμα που θα δείτε χρειάστηκε να δημιουργηθούν κάποιες libraries από το μηδέν για να μπορέσουμε να κάνουμε κάποια πράγματα σχετικά με την διαχείριση και την σωστή λειτουργία του ηλεκτρονικού ινστιτούτου αισθητικής μας.

5.5.1 MY_Controller (custom library γενικών λειτουργιών)

```

<?php
class MY_Controller extends Controller {

    /* Output control constants */
    const OUTPUT_TEMPLATE = 10;
    const OUTPUT_NORMAL = 11;

    /* Private properties */

    //the default template
    protected $template_view = 'default';

    //the default public folder this means that content inside this folder will be
    accessed directly
    //without using the routing.
    //Note!!! This folder must be enabled from the .htaccess file.
    private $public_folder = 'public/';

    //the default location for the templates inside the views folder this means
    (views/templates/)
    private $template_folder = 'templates/';

    //the default css location for the css files inside the $public_folder("public/" by
    default) (public/css/)
    private $css_folder = 'css/';

    //the default js location for the css files inside the $public_folder("public/" by
    default) (public/js/)
    private $js_folder = 'js/';

    //Inline scripting (js, css etc)
    private $inline_scripting = '';

    private $modules = array(); //An array that contains the module output.

    private $charset = ''; //The page charset

    private $title = ''; //The page Title

    //Media files and data
    private $media = array('css'=>array(),
                           'js'=>array(),
                           'meta'=>array(),
                           'rdf'=>array());

    //An instance of the SEOContent Class
    /**
     *
     * @var SEOContent
     */
    protected $metaObj = null;

    //
    protected $folder_path = '';
    //The requested controller to be loaded
    protected $controller = '';
    //The requested method to be called
    protected $method = '';

    private $_output_data = array();
    
```

```

public $language = 'english';

/**
 * The MY_Controller constructor method.
 */
function __construct(){

    parent::__Controller();

    //Initializing the controller
    //Get the default charset from the config file.
    $this->charset = $this->config->item('charset');

    //Set the default mode to use a template view
    $this->_setOutputMode(self::OUTPUT_TEMPLATE);

    //Passing as paramaters the controller and the method that should be used.
    //This is useful for the SEOContent object above.
    $this->controller = $this->uri->rsegment(1);
    $this->method     = $this->uri->rsegment(2);

    $index = array_search($this->controller, $this->uri->segment_array());

    if($index > 1){
        for($i=1; $i < $index; $i++){
            $this->folder_path[] = $this->uri->segment($i);
        }
        $this->folder_path = implode("/", $this->folder_path) . "/";
    }

    $this->language = ($this->uri->segment(1) == '')
    ? 'english'
    : $this->uri->segment(1);
    $languages = array();

    $langs = $this->db->where('status','1')->get('languages')->result();

    foreach($langs as $lang)
    $languages[] = $lang->title;

    if(array_search($this->language, $languages) === false)
    $this->language = 'english';

    $this->config->set_item('language', $this->language);
    //$this->load->language('template');

    //Initialize the controller according the ni_controller config file.
    $this->init_controller();

}

private function init_controller(){
    if($this->load->config('MY_controller', true, true)){

    $default_conf = $this->config->item("default", "my_controller");
    $controller_conf = $this->config->item($this->controller, "my_controller");

    if($controller_conf !== false){
    $default_conf = array_merge_recursive($default_conf, $controller_conf);
    }

    $this->_SetTemplate($default_conf['template']);

    foreach($default_conf['module_positions'] as $position){
        $this->_register_module_position($position);
    }

    foreach ($default_conf['modules'] as $pos=>$modules){

```

```

        foreach($modules as $module){
            $this->_load_module($pos, $module);
        }

        foreach($default_conf['libraries'] as $library){
            $this->load->library($library);
        }

        foreach($default_conf['models'] as $model){
            $this->load->model($model);
        }

        foreach($default_conf['helpers'] as $helper){
            $this->load->helper($helper);
        }
    }
}

/**
 * Caching the inline scripting.
 *
 * @return void
 */
function inlinejs($checksum=null){
    $this->_setOutputMode(self::OUTPUT_NORMAL);

    if(is_null($checksum))
        show_404();

    $checksum = str_replace(".js", "", $checksum);

    $this->output->cache(20*24*60);

    $content = $this->session->userdata($checksum);
    $this->session->unset_userdata($checksum);

    header('Content-type: text/javascript');
    header("Cache-Control: public");
    header("Pragma: public");
    $expire_date = date("D, d M Y H:i:s e", time() + (20*24*60*60) );
    header("Expires: {$expire_date}");
    header("Etag: {$checksum}");

    echo $this->load->view( "templates/blank",array("content"=>$content) , false);

    return;
}

/**
 * CodeIgniter magic method that controls the output of a controller.
 * You can't call it directly.
 *
 * @see http://codeigniter.com/user_guide/general/controllers.html#output
 * @final this method cannot be overloaded
 * @param string $output the controller output
 * @return void
 */
final function _output($output){

    if($this->inline_scripting != ''){
        $checksum = md5($this->inline_scripting, false);
        $this->session->set_userdata($checksum, $this->inline_scripting);
        $this->_add_js_file(site_url("{ $this->folder_path }{$this->
>controller}/inlinejs/{ $checksum }.js"), true);
    }

    switch($this->mode){
        //Use the template
        case self::OUTPUT_TEMPLATE:
    }
}

```



```

        $data = array( 'meta'=>$this->media['meta'],
                    'rdf'=>$this->media['rdf'],
                    'js'=>$this->media['js'],
                    'css'=>$this->media['css'],
                    'title'=>$this->title,
                    'charset'=>$this->charset,
                    'output'=>$output,
                    'modules'=>(object)$this-
>modules,
                    //just for compatibility issues
                    'inline_scripting'=>
""); //<script type="text/javascript"> . $this->inline_scripting . "</script>");

        //Merge the data arrays
        $data = array_merge($data, $this->_output_data);

        //Load the final output
        $out = $this->load->view($this->template_folder . $this-
>template_view, $data, TRUE);

        //The End
        $this->_compress_output($out);
        break;
        //or just echo output.
        case self::OUTPUT_NORMAL:
        default:
            $this->_compress_output($output);
            break;
    }
}

/**
 * PHP Speedy compression.
 *
 */
private function _compress_output($output){
    //ini_set('memory_limit', '64M');
    //require_once(APPPATH . 'external_libraries/compressor/php_speedy.php');
    echo $output;
    //$compressor->finish();
}

/**
 * Pass data on the final output.
 *
 * @param string $paramName the parameter name.
 * @param mixed $value $the value of the parameter
 */
public function _setOutputData($paramName, $value){
    $this->_output_data[$paramName] = $value;
}

/**
 * This method sets the output mode.
 *
 * @access protected
 * @param int $mode One of the constants self::OUTPUT_TEMPLATE, self::OUTPUT_NORMAL
 * @return void
 */
public function _setOutputMode($mode){
    $this->mode = $mode;
}

/**
 * Sets the template that will be used at the final output.
 *
 * @deprecated
 * @access protected
 * @param string $template
 * @return bool
 */

```

```

        */
        public function _template_to_use($template){
            trigger_error("Method " . __METHOD__ . " has marked as deprecated. Please use
" . __CLASS__ . "::_SetTemplate() instead.", E_USER_WARNING );
            $this->_SetTemplate($template);
        }

        protected function _SetTemplate($template){
            $filepath = APPPATH . "views/" . $this->template_folder .
str_replace('.php','',$template) . ".php";

            if(!$this->_is_file_exists($filepath)){
                throw new Exception("Cannot locate template file <tt>$template</tt>");
                return false;
            }

            $this->template_view = $template;
            return true;
        }

        /**
         * Adds a Javascript file into the template.
         *
         * @access protected
         * @param string $file a js file located inside the public/js/ folder or an url.
         * @param boolean $custom_url Default FALSE. A flag to determine if the given $file is
an url or just a file insite the public/js/ folder.
         * @return bool
         */
        public function _add_js_file($file, $custom_url=false){

            if($custom_url===false){
                $filepath = $this->public_folder . $this->js_folder .
str_replace('.js','',$file) . ".js";

                if(!$this->_is_file_exists($filepath)){
                    throw new Exception("Cannot locate javascript file
<tt>$filepath</tt>");
                    return false;
                }

                $filepath = base_url() . $filepath;
            }else{
                $filepath = $file;
            }

            if(array_search($filepath, $this->media['js']) === false)
                $this->media['js'][] = $filepath;
            else
                return false;

            return true;
        }

        /**
         * Adds a CSS file into the template.
         *
         * @access protected
         * @param string $file a css file located inside the public/js/ folder or an url.
         * @param boolean $custom_url Default FALSE. A flag to determine if the given $file is
an url or just a file insite the public/js/ folder.
         * @return bool
         */

        /**
         * Adds a CSS file to the template page.
         *
         * @access protected
         * @param string $file a css file located inside the public/css/ folder.
         * @return bool
         */
    
```

```

        */
        public function _add_css_file($file, $custom_url=false){
            if(!$custom_url){
                $filepath = $this->public_folder . $this->css_folder .
str_replace('.css', '', $file) . ".css";
                if(!$this->is_file_exists($filepath)){
                    show_error("Cannot locate css file <tt>$filepath</tt>");
                    return false;
                }

                $filepath = base_url() . $filepath;
            }else{
                $filepath = $file;
            }

            if(array_search($filepath, $this->media['css']) === false)
                $this->media['css'][] = $filepath;
            else return false;

            return true;
        }

        /**
         * Sets the default charset
         *
         * @access protected
         * @param string $charset
         * @return void
         */
        public function _set_charset($charset){
            $this->charset = $charset;
        }

        /**
         * Sets the page title
         *
         * @access protected
         * @param string $new_title
         * @return void
         */
        public function _set_title($new_title){
            $this->title = $new_title;
        }

        /**
         * prepends a string at the title text
         *
         * @access protected
         * @param string $title
         * @return void
         */
        public function _prepend_title($title){
            $this->title = $title." :: ".$this->title;
        }

        /**
         * Appends a string at the title text
         *
         * @access protected
         * @param string $title
         * @return void
         */
        public function _append_title($title){
            $this->title .= " - $title";
        }

        /**
         * Adds meta tags.
    
```

```

*
* @access protected
* @param string $name the name of the meta tag
* @param string $content the content of the meta tag
* @return bool
*/
public function _add_meta($name, $content){
    // if(array_key_exists($name, $this->media['meta']))
    // show_error("Duplicate usage of meta tag file
<tt>$name</tt>.");

    $this->media['meta'][$name] = $content;
    return true;
}

/**
 * Adds RDF meta tags (3rd generation meta tags).
 *
 * @access protected
 * @param string $name the name of the meta tag
 * @param string $content the content of the meta tag
 * @return bool
 */
public function _add_RDF($name, $content){
    if(array_key_exists($name, $this->media['rdf']))
        show_error("Duplicate usage of meta tag file <tt>$name</tt>.");

    $this->media['rdf'][$name] = $content;
    return true;
}

/**
 * Registers module positions
 *
 * @access protected
 * @param string $position_name the name of the position (no special chars or spaces
are allowed)
 * @return bool
 */
public function _register_module_position($position_name){
    if(array_key_exists($position_name, $this->modules))
        show_error("Module position failed because position <tt>$position_name</tt>
has already been registred.");

    //Check for illegal characters.
    if(!preg_match("/[a-zA-Z0-9]*/", $position_name))
        show_error("Position name <tt>$position_name</tt> contains some illegal
characters. Only letters or numbers are allowed.");

    $this->modules[$position_name] = array();

    return true;
}

/**
 * Loads a view file (module) in a certain position.
 *
 * @access protected
 * @param string $position the module position
 * @param string $view_file the view file path.
 * @param array $params the parameter to be passed in the view file.
 * @return bool
 */
public function _load_module($position, $view_file, $params = array(), $index=null){
    if(!array_key_exists($position, $this->modules))
        show_error("Module position <tt>$position</tt> hasn't ever been registred.");

    if(!is_null($index))
        $this->modules[$position][$index] = $this->load->view($view_file, $params,
TRUE);
    else

```

```

        $this->modules[$position][] = $this->load->view($view_file, $params, TRUE);

        return true;
    }

    /**
     * Marks the beginning of inline scripting.
     * Example:
     * <?php ....
     *
     *                               $this->_start_inline_scripting();
     *     ?>
     *     <script type="text/javascript" > .... </script>
     * <?php
     *     $this->_end_inline_scripting();
     *     .....
     *
     */
    public function _start_inline_scripting(){
        ob_start();
    }

    /**
     * Marks the end of the inline scripting.
     */
    public function _end_inline_scripting(){
        $s = ob_get_clean();

        $s = preg_replace("</script.[^]*>/", '', $s);
        $s = preg_replace("</\</script>/", '', $s);

        $this->inline_scripting .= $s;
    }

    /**
     * Checks if the given file exists in the filesystem.
     *
     * @access private
     * @param string $filepath The file path, using a physical relative path
     * @return bool
     */
    private function _is_file_exists($filepath){
        return file_exists($filepath);
    }

    protected function _resetMedia(){
        $this->media = array( 'css'=>array(),
                             'js'=>array(),
                             'meta'=>array(),
                             'rdf'=>array());
    }

    protected function is_ajax() {
        return ($this->input->server('HTTP_X_REQUESTED_WITH') == 'XMLHttpRequest') ? TRUE : FALSE;
    }

}

```

5.5.2 Ion_auth.php (custom user authorization & authentication)⁽¹¹⁾

Ένα σχεδόν πλήρες σύστημα χρηστών που χρειάστηκε να κάνουμε κάποιες παραλλαγές. Βιβλιοθήκη η οποία θα διαχειριστεί τις διαδικασίες που απαιτεί ένα ολοκληρωμένο σύστημα χρηστών όπως login, register, forgotten password, remember, activation emails, και πολλά άλλα.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/**
 * Name: Ion Auth
 *
 * Author: Ben Edmunds
 *         ben.edmunds@gmail.com
 *         @benedmunds
 *
 * Added Awesomeness: Phil Sturgeon
 *
 * Location: http://github.com/benedmunds/CodeIgniter-Ion-Auth
 *
 * Created: 10.01.2009
 *
 * Description: Modified auth system based on redux_auth with extensive customization. This
is basically what Redux Auth 2 should be.
 * Original Author name has been kept but that does not mean that the method has not been
modified.
 *
 * Requirements: PHP5 or above
 */

class Ion_auth
{
    /**
     * CodeIgniter global
     *
     * @var string
     */
    protected $ci;

    /**
     * account status ('not_activated', etc ...)
     *
     * @var string
     */
    protected $status;

    /**
     * message (uses lang file)
     *
     * @var string
     */
    protected $messages;

    /**
     * error message (uses lang file)
     *
     * @var string
     */
    protected $errors = array();

    /**
     * error start delimiter
     *
     * @var string
     */
    protected $error_start_delimiter;

    /**
     * error end delimiter
     */
}
```

```

    * @var string
    **/
    protected $error_end_delimiter;

    /**
     * extra where
     *
     * @var array
     **/
    public $_extra_where = array();

    /**
     * extra set
     *
     * @var array
     **/
    public $_extra_set = array();

    /**
     * __construct
     *
     * @return void
     * @author Mathew
     **/
    public function __construct()
    {
        $this->ci =& get_instance();
        $this->ci->load->config('ion_auth', TRUE);
        $this->ci->load->library('email');
        $this->ci->load->library('session');
        $this->ci->lang->load('ion_auth');
        $this->ci->load->model('ion_auth_model');
        $this->ci->load->helper('cookie');
        $this->ci->config->load('mailer');

        $this->messages = array();
        $this->errors = array();
        $this->message_start_delimiter = $this->ci->config-
>item('message_start_delimiter', 'ion_auth');
        $this->message_end_delimiter = $this->ci->config-
>item('message_end_delimiter', 'ion_auth');
        $this->error_start_delimiter = $this->ci->config-
>item('error_start_delimiter', 'ion_auth');
        $this->error_end_delimiter = $this->ci->config-
>item('error_end_delimiter', 'ion_auth');

        //auto-login the user if they are remembered
        if (!$this->logged_in() && get_cookie('identity') &&
get_cookie('remember_code'))
        {
            $this->ci->ion_auth_model->login_remembered_user();
        }
    }

    /**
     * __call
     *
     * Acts as a simple way to call model methods without loads of stupid alias'
     *
     **/
    public function __call($method, $arguments)
    {
        if (!method_exists( $this->ci->ion_auth_model, $method ) )
        {
            throw new Exception('Undefined method Ion_auth::' . $method . '()
called');
        }

        return call_user_func_array( array($this->ci->ion_auth_model, $method),
$arguments);
    }

```



```

/**
 * Activate user.
 *
 * @return void
 * @author Mathew
 */
public function activate($id, $code=false)
{
    if ($this->ci->ion_auth_model->activate($id, $code))
    {
        $this->set_message('activate_successful');
        return TRUE;
    }

    $this->set_error('activate_unsuccessful');
    return FALSE;
}

/**
 * Deactivate user.
 *
 * @return void
 * @author Mathew
 */
public function deactivate($id)
{
    if ($this->ci->ion_auth_model->deactivate($id))
    {
        $this->set_message('deactivate_successful');
        return TRUE;
    }

    $this->set_error('deactivate_unsuccessful');
    return FALSE;
}

/**
 * Change password.
 *
 * @return void
 * @author Mathew
 */
public function change_password($identity, $old, $new)
{
    if ($this->ci->ion_auth_model->change_password($identity, $old, $new))
    {
        $this->set_message('password_change_successful');
        return TRUE;
    }

    $this->set_error('password_change_unsuccessful');
    return FALSE;
}

/**
 * forgotten password feature
 *
 * @return void
 * @author Mathew
 */
public function forgotten_password($identity) //changed $email to $identity
{
    if ( $this->ci->ion_auth_model->forgotten_password($identity) ) //changed
    {
        // Get user information
        $user = $this->get_user_by_identity($identity); //changed to
get_user_by_identity from email

        $data = array(

```

```

>item('identity', 'ion_auth')),
    'identity' => $user->{$this->ci->config-
    'forgotten_password_code' => $user->forgotten_password_code
    );
    $message = $this->ci->load->view($this->ci->config-
>item('email_templates', 'ion_auth').$this->ci->config->item('email_forgot_password', 'ion_auth'),
$data, true);
    $this->ci->email->clear();
    $this->ci->email->initialize($this->ci->config->item('email'));
    $this->ci->email->set_newline("\r\n");
    $this->ci->email->from($this->ci->config->item('admin_email',
'ion_auth'), $this->ci->config->item('site_title', 'ion_auth'));
    $this->ci->email->to($user->email);
    $this->ci->email->subject($this->ci->config->item('site_title',
'ion_auth') . ' - Forgotten Password Verification');
    $this->ci->email->message($message);
    if ($this->ci->email->send())
    {
        $this->set_message('forgot_password_successful');
        return TRUE;
    }
    else
    {
        $this->set_error('forgot_password_unsuccessful');
        return FALSE;
    }
}
else
{
    $this->set_error('forgot_password_unsuccessful');
    return FALSE;
}
}

/**
 * forgotten_password_complete
 *
 * @return void
 * @author Mathew
 */
public function forgotten_password_complete($code)
{
    $identity = $this->ci->config->item('identity', 'ion_auth');
    $profile = $this->ci->ion_auth_model->profile($code, true); //pass the code
to profile

    if (!is_object($profile))
    {
        $this->set_error('password_change_unsuccessful');
        return FALSE;
    }

    $new_password = $this->ci->ion_auth_model->forgotten_password_complete($code,
$profile->salt);

    if ($new_password)
    {
        $data = array(
            'identity' => $profile->{$identity},
            'new_password' => $new_password
        );
        $message = $this->ci->load->view($this->ci->config-
>item('email_templates',
'ion_auth').$this->ci->config->item('email_forgot_password_complete',
'ion_auth'), $data, true);

        $this->ci->email->clear();
        $this->ci->email->initialize($this->ci->config->item('email'));
        $this->ci->email->set_newline("\r\n");
    }
}

```

```

        $this->ci->email->from($this->ci->config->item('admin_email',
'ion_auth'), $this->ci->config->item('site_title', 'ion_auth'));
        $this->ci->email->to($profile->email);
        $this->ci->email->subject($this->ci->config->item('site_title',
'ion_auth') . ' - New Password');
        $this->ci->email->message($message);

        if ($this->ci->email->send())
        {
            $this->set_message('password_change_successful');
            return TRUE;
        }
        else
        {
            $this->set_error('password_change_unsuccessful');
            return FALSE;
        }
    }

    $this->set_error('password_change_unsuccessful');
    return FALSE;
}
/**
 * register
 *
 * @return void
 * @author Mathew
 */
public function register($username, $password, $email, $additional_data, $group_name =
false) //need to test email activation
{
    $email_activation = $this->ci->config->item('email_activation', 'ion_auth');

    if (!$email_activation)
    {
        $id = $this->ci->ion_auth_model->register($username, $password,
$email, $additional_data, $group_name);
        if ($id !== FALSE)
        {
            $this->set_message('account_creation_successful');
            return $id;
        }
        else
        {
            $this->set_error('account_creation_unsuccessful');
            return FALSE;
        }
    }
    else
    {
        $id = $this->ci->ion_auth_model->register($username, $password,
$email, $additional_data, $group_name);

        if (!$id)
        {
            $this->set_error('account_creation_unsuccessful');
            return FALSE;
        }

        $deactivate = $this->ci->ion_auth_model->deactivate($id);

        if (!$deactivate)
        {
            $this->set_error('deactivate_unsuccessful');
            return FALSE;
        }

        $activation_code = $this->ci->ion_auth_model->activation_code;
        $identity = $this->ci->config->item('identity', 'ion_auth');
        $user = $this->ci->ion_auth_model->get_user($id)->row();
    }
}

```

```

        $data = array(
            'identity' => $user->{$identity},
            'id' => $user->id,
            'email' => $email,
            'activation' => $activation_code,
        );

        $message = $this->ci->load->view($this->ci->config->
>item('email_templates', 'ion_auth').$this->ci->config->item('email_activate', 'ion_auth'), $data,
true);

        $this->ci->email->clear();
        $this->ci->email->initialize($this->ci->config->item('email'));
        $this->ci->email->set_newline("\r\n");
        $this->ci->email->from($this->ci->config->item('admin_email',
'ion_auth'), $this->ci->config->item('site_title', 'ion_auth'));
        $this->ci->email->to($email);
        $this->ci->email->subject($this->ci->config->item('site_title',
'ion_auth') . ' - Account Activation');
        $this->ci->email->message($message);

        if ($this->ci->email->send() == TRUE)
        {
            $this->set_message('activation_email_successful');
            return $id;
        }
        $this->set_error('activation_email_unsuccessful');
        return FALSE;
    }
}
/**
 * login
 *
 * @return void
 * @author Mathew
 */
public function login($identity, $password, $remember=false)
{
    if ($this->ci->ion_auth_model->login($identity, $password, $remember))
    {
        $this->set_message('login_successful');
        return TRUE;
    }

    $this->set_error('login_unsuccessful');
    return FALSE;
}
/**
 * logout
 *
 * @return void
 * @author Mathew
 */
public function logout()
{
    $identity = $this->ci->config->item('identity', 'ion_auth');
    $this->ci->session->unset_userdata($identity);
    $this->ci->session->unset_userdata('group');
    $this->ci->session->unset_userdata('id');
    $this->ci->session->unset_userdata('user_id');
    //delete the remember me cookies if they exist
    if (get_cookie('identity')) {
        delete_cookie('identity');
    }
    if (get_cookie('remember_code')) {
        delete_cookie('remember_code');
    }
}
    $this->ci->session->sess_destroy();
    $this->set_message('logout_successful');
    return TRUE;
}
/**
 * logged_in

```

```

*
* @return bool
* @author Mathew
**/
public function logged_in()
{
    $identity = $this->ci->config->item('identity', 'ion_auth');
    return (bool) $this->ci->session->userdata($identity);
}

/**
 * is_admin
 *
 * @return bool
 * @author Ben Edmunds
 **/
public function is_admin()
{
    $admin_group = $this->ci->config->item('admin_group', 'ion_auth');
    $user_group = $this->ci->session->userdata('group');

    return $user_group == $admin_group;
}

/**
 * is_group
 *
 * @return bool
 * @author Phil Sturgeon
 **/
public function is_group($check_group)
{
    $user_group = $this->ci->session->userdata('group');

    if(is_array($check_group))
    {
        return in_array($user_group, $check_group);
    }

    return $user_group == $check_group;
}

/**
 * Profile
 *
 * @return void
 * @author Mathew
 **/
public function profile()
{
    $session = $this->ci->config->item('identity', 'ion_auth');
    $identity = $this->ci->session->userdata($session);

    return $this->ci->ion_auth_model->profile($identity);
}

/**
 * Get Users
 *
 * @return object Users
 * @author Ben Edmunds
 **/
public function get_users($group_name=false, $limit=NULL, $offset=NULL)
{
    return $this->ci->ion_auth_model->get_users($group_name, $limit, $offset)->result();
}

/**
 * Get Users Array
 *
 * @return array Users

```

```

    * @author Ben Edmunds
    **/
    public function get_users_array($group_name=false, $limit=NULL, $offset=NULL)
    {
return $this->ci->ion_auth_model->get_users($group_name, $limit, $offset)->result_array();
    }
    /**
    * Get Newest Users
    *
    * @return object Users
    * @author Ben Edmunds
    **/
    public function get_newest_users($limit = 10)
    {
        return $this->ci->ion_auth_model->get_newest_users($limit)->result();
    }
    /**
    * Get Newest Users Array
    *
    * @return object Users
    * @author Ben Edmunds
    **/
    public function get_newest_users_array($limit = 10)
    {
        return $this->ci->ion_auth_model->get_newest_users($limit)->result_array();
    }
    /**
    * Get Active Users
    *
    * @return object Users
    * @author Ben Edmunds
    **/
    public function get_active_users($group_name = false)
    {
return $this->ci->ion_auth_model->get_active_users($group_name)->result();
    }
    /**
    * Get Active Users Array
    *
    * @return object Users
    * @author Ben Edmunds
    **/
    public function get_active_users_array($group_name = false)
    {
return $this->ci->ion_auth_model->get_active_users($group_name)->result_array();
    }
    /**
    * Get In-Active Users
    *
    * @return object Users
    * @author Ben Edmunds
    **/
    public function get_inactive_users($group_name = false)
    {
return $this->ci->ion_auth_model->get_inactive_users($group_name)->result();
    }
    /**
    * Get In-Active Users Array
    *
    * @return object Users
    * @author Ben Edmunds
    **/
    public function get_inactive_users_array($group_name = false)
    {
Return $this->ci->ion_auth_model->get_inactive_users($group_name)->result_array();
    }
    /**

```

```

* Get User
*
* @return object User
* @author Ben Edmunds
**/
public function get_user($id=false)
{
    return $this->ci->ion_auth_model->get_user($id)->row();
}

/**
 * Get User by Email
 *
 * @return object User
 * @author Ben Edmunds
 **/
public function get_user_by_email($email)
{
    return $this->ci->ion_auth_model->get_user_by_email($email)->row();
}

/**
 * Get Users by Email
 *
 * @return object Users
 * @author Ben Edmunds
 **/
public function get_users_by_email($email)
{
    return $this->ci->ion_auth_model->get_users_by_email($email)->result();
}

/**
 * Get User by Identity
 *                                     //copied from above ^
 * @return object User
 * @author jondavidjohn
 **/
public function get_user_by_identity($identity)
{
    return $this->ci->ion_auth_model->get_user_by_identity($identity)->row();
}

/**
 * Get User as Array
 *
 * @return array User
 * @author Ben Edmunds
 **/
public function get_user_array($id=false)
{
    return $this->ci->ion_auth_model->get_user($id)->row_array();
}

/**
 * update_user
 *
 * @return void
 * @author Phil Sturgeon
 **/
public function update_user($id, $data)
{
    if ($this->ci->ion_auth_model->update_user($id, $data))
    {
        $this->set_message('update_successful');
        return TRUE;
    }

    $this->set_error('update_unsuccessful');
    return FALSE;
}

/**
 * delete_user

```

```

*
* @return void
* @author Phil Sturgeon
**/
public function delete_user($id)
{
    if ($this->ci->ion_auth_model->delete_user($id))
    {
        $this->set_message('delete_successful');
        return TRUE;
    }

    $this->set_error('delete_unsuccessful');
    return FALSE;
}

/**
 * extra_where
 *
 * Crazy function that allows extra where field to be used for user fetching/unique
checking etc.
 * Basically this allows users to be unique based on one other thing than the
identifier which is helpful
 * for sites using multiple domains on a single database.
 *
 * @return void
 * @author Phil Sturgeon
 **/
public function extra_where()
{
    $where =& func_get_args();
    $this->_extra_where = count($where) == 1 ? $where[0] : array($where[0] =>
$where[1]);
}

/**
 * extra_set
 *
 * Set your extra field for registration
 *
 * @return void
 * @author Phil Sturgeon
 **/
public function extra_set()
{
    $set =& func_get_args();
    $this->_extra_set = count($set) == 1 ? $set[0] : array($set[0] => $set[1]);
}

/**
 * set_message_delimiters
 *
 * Set the message delimiters
 *
 * @return void
 * @author Ben Edmunds
 **/
public function set_message_delimiters($start_delimiter, $end_delimiter)
{
    $this->message_start_delimiter = $start_delimiter;
    $this->message_end_delimiter = $end_delimiter;
    return TRUE;
}

/**
 * set_error_delimiters *
 * Set the error delimiters *
 * @return void
 * @author Ben Edmunds
 **/
public function set_error_delimiters($start_delimiter, $end_delimiter)
{
    $this->error_start_delimiter = $start_delimiter;

```



```

        $this->error_end_delimiter = $end_delimiter;
        return TRUE;
    }

    /**
     * set_message
     *
     * Set a message
     *
     * @return void
     * @author Ben Edmunds
     */
    public function set_message($message)
    {
        $this->messages[] = $message;
        return $message;
    }

    /**
     * messages
     *
     * Get the messages
     *
     * @return void
     * @author Ben Edmunds
     */
    public function messages()
    {
        $_output = '';
        foreach ($this->messages as $message)
        {
            $_output .= $this->message_start_delimiter . $this->ci->lang-
>line($message) . $this->message_end_delimiter;
        }
        return $_output;
    }

    /**
     * set_error
     *
     * Set an error message
     *
     * @return void
     * @author Ben Edmunds
     */
    public function set_error($error)
    {
        $this->errors[] = $error;
        return $error;
    }

    /**
     * errors
     *
     * Get the error message
     *
     * @return void
     * @author Ben Edmunds
     */
    public function errors()
    {
        $_output = '';
        foreach ($this->errors as $error)
        {
            $_output .= $this->error_start_delimiter . $this->ci->lang-
>line($error) . $this->error_end_delimiter;
        }
        return $_output;
    }
}

```

5.5.3 Cnode.php (custom library για της κατηγορίες)⁽⁹⁾

Αυτή η βιβλιοθήκη δημιουργήθηκε με σκοπό να μπορούμε να διαχειριστούμε τις κατηγορίες οποιουδήποτε είδους κατηγοριών σε μία δένδροειδή μορφή που μπορεί να προσαρμοστεί οπουδήποτε αρκεί να έχει σχεδιαστεί κατάλληλα ο πίνακας στην βάση δεδομένων μας με τα κατάλληλα πεδία που απαιτούνται. Το μόνο που χρειάζεται είναι να ορίσουμε ένα `$node= new cNode ('onoma_table')` που θα παίρνει σαν όρισμα τον πίνακα που θέλουμε από τη βάση και μας βγάζει αυτόματα τις υποκατηγορίες. **Με αυτόν τον τρόπο δημιουργούμε αντικείμενα.**

```

<?php

define( 'ROOT', 0 );

class cNode
{
    public $nodeId = '';
    public $title = '';
    public $tree = '';
    public $categoriesTable = '';
    // accepts reference to database
    function __construct($node_id, $table)
    {
        $this->categoriesTable = $table;
        $ci = &get_instance();
        if( $this->checkNodeExists( $node_id ) || $node_id == ROOT)
        {
            $child = $ci->db->get_where($this->categoriesTable, array('id'=>$node_id))->row();
            $this->nodeId = $node_id;
            $this->title = $child->title;
        }else
        show_error( __METHOD__ . ' failed '.$node_id);
    }

    // accepts node id
    // returns array of child node id's
    function getChildren($mode = null)
    {
        $ci = &get_instance();

        if(!empty($this->nodeId))
            $sql = 'SELECT id FROM '.$this->categoriesTable.' WHERE parent_id = ' . $this->nodeId. ' ORDER
            BY `order` ASC';
        else
            return;

        $children = false;
        $children = $ci->db->query( $sql )->result();

        if( empty($children) )
        {
            //show_error('cNode::getChildren() failed');
            return;
        }

        foreach($children as $child)
        {
            $mychild = new cNode($child->id, $this->categoriesTable);
            if($mode == 'all')
                $childrens = $mychild->getChildren();
            $childrens[] = $mychild;
        }

        return $childrens;
    }

    function getNodeId()
    {
        return $this->nodeId;
    }
}

```

```

    }

    // accepts node id
    // returns parent node or false if none
    function getParent()
    {
        $ci = &get_instance();

        $node_id = $this->getNodeId();

        $sql = 'SELECT parent_id FROM '.$this->categoriesTable.' WHERE id = ' .
$node_id;

        $parent_node = false;
        $parent_node = $ci->db->query($sql);
        if($parent_node->num_rows() == 0)
        {
            //show_error('cNode::getParent() failed');
            return;
        }
        $parent_node = new cNode( $parent_node->row()->parent_id, $this->
>categoriesTable);

        return $parent_node;
    }

    // accepts node id
    // returns array of nodes ordered from starting node to root node or false if root
    node
    function getNodesToRoot()
    {
        $path = array();
        array_push( $path, $this->nodeId);

        if($this->getParent($this->nodeId))
        while( $this->nodeId = $this->getParent($this->nodeId)->nodeId)
        array_push( $path, $this->nodeId);

        return $path;
    }

    // returns true or dies
    function setParent( cNode $newParent )
    {
        $ci = &get_instance();

        $sql = 'UPDATE '.$this->categoriesTable.' SET parent_id = ' . $newParent->
>getNodeId() . ' WHERE id = ' . $this->getNodeId();
        $ci->db->query( $sql );

        return true;
    }

    // add child to parent
    // return the child Id
    function addChild($title=null)
    {
        $ci = &get_instance();

        $ci->db->insert($this->categoriesTable, array(
            'parent_id' => $this->nodeId ,
            'title' => $title
        ));

        // hack for the stupid non support for AUTO_INCREMENT
        $childNode = new cNode( $ci->db->insert_id(), $this->categoriesTable);
        $nodes = $childNode->getNodesToRoot();

        $parent_path = implode("_", array_reverse($nodes));
    }

```

```

        $ci->db->where('id', $ci->db->insert_id());
        $ci->db->update($this->categoriesTable, array('parent_path'=>$parent_path));

        return $childNode;
    }

    // remove a node and all children Ids
    function deleteNode( cNode $node, $removeChildren = true)
    {
        $node_id = $node->getNodeId();

        if($node_id == 1)
            return;

        $ci = &get_instance();

        $children = $node->getChildren();

        if(!empty($children))
        {
            if( $removeChildren)
            {
                foreach( $children as $child )
                {
                    $this->deleteNode( $child, true );
                }
            } else {
                $parent_node = $node->getParent();

                foreach( $children as $child )
                {
                    $child->setParent( $parent_node );
                }
            }
        }

        if(!empty($node_id))
        {
            $sql = "DELETE FROM `".$this->categoriesTable."` WHERE `id` =
'{$node_id}'";
            $result = $ci->db->query( $sql );

            if( empty($result) )
            {
                return;
            }
            return true;
        }else
        return;
    }

    // check if a node exists
    function checkNodeExists($nodeId)
    {
        $ci = &get_instance();
        $nodeOk = false;

        $sql = 'SELECT id FROM '.$this->categoriesTable.' WHERE id = ' . $nodeId;
        $result = $ci->db->query( $sql );
        if( $result->row() )
            $nodeOk = true;

        return $nodeOk;
    }

    // takes reference to a db object
    // using PEAR DB Class

    function createNodeDB()
    {

```

```

        $ci = &get_instance();
        $sql = 'CREATE TABLE `'.$this->categoriesTable.` (
                `id` int(11) NOT NULL auto_increment,
                `parent_id` int(11) NOT NULL default "0",
                PRIMARY KEY (`id`)
            ) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;';

        $ci->db->query( $sql );
    }

    function findRootID()
    {
        $ci = &get_instance();

        //lets find root
        $ci->db->limit(1);
        $ci->db->order_by('parent_id','asc');
        $root = $ci->db->get($this->categoriesTable->row()->parent_id;

        return $root;
    }

    function getTree($id=null, $mode=null, $selected_id=null, $name = 'nodeID')
    {
        // Available modes: menu, radio, null
        $ci = &get_instance();
        $tree = array();
        if(is_null($id))
        {
            $id = $this->findRootID();
            $ci->db->order_by('parent_id','asc');
            $ci->db->where('parent_id', $id);
            $children = $ci->db->get($this->categoriesTable);

            if($children->num_rows() > 0)
            {
                foreach($children->result() as $child)
                {
                    $childs[] = new cNode($child->id, $this->categoriesTable);
                    $node_id = $child->id;

                    $ok2 = $this->getTree($node_id, $mode, $selected_id, $name);
                    if($mode == 'raw')
                        $tree[$node_id] = $ok2;
                    else
                        $tree[$this->getCategoryTitle($node_id, $mode, $selected_id, $name)] = $ok2;
                }
            }

            return $tree;
        }

        $ok = new cNode($id, $this->categoriesTable);
        $childs = $ok->getChildren();
        if(!empty($childs))
        {
            foreach($childs as $child)
            {
                $node_id = $child->getNodeId();
                $ok2 = $this->getTree($node_id, $mode, $selected_id, $name);
                if($mode == 'raw')
                    $tree[$node_id] = $ok2;
                else
                    $tree[$this->getCategoryTitle($node_id, $mode, $selected_id, $name)] = $ok2;
            }
        }
        $this->tree = $tree;

        return $tree;
    }
}

```

```

function getCategoryProducts($node_id=null)
{
    $products = array();
    $ci = &get_instance();

    $ci->db->select('id');
    $ci->db->where('category_id', $node_id);
    $excursions = $ci->db->get('excursions');
    if($excursions->num_rows() > 0)
    {
        foreach($excursions->result() as $excursion)
            $products[] = $excursion->id;
    }

    return $products;
}

function getBreadcrumb($cat_id)
{
    $breadcrumb = array();

    if(!is_null($cat_id))
    {
        $ci = &get_instance();

        $thisNode = new cNode($cat_id, $this->categoriesTable);
        $path = array_reverse($thisNode->getNodesToRoot());
        unset($path[0]);
        foreach($path as $node_id)
        {
            $node = $ci->db->get_where($this->categoriesTable, array('id'=>$node_id))->row();
            $breadcrumb[] = (object)array(
                'title'=>$node->title,
                'id'=>$node_id
            );
        }

        return $breadcrumb;
    }
}

function getCategoryTitle($node_id=null, $mode=null, $selected_id=null, $name = 'nodeID')
{
    $ci = &get_instance();

    $thisNode = new cNode($node_id, $this->categoriesTable);
    $title = $thisNode->title;
    switch($mode)
    {
        case 'menu':
            $nodeName = array();
            $path = $thisNode->getNodesToRoot();
            foreach($path as $node)
            {
                if($node != 1)
                    $nodeName[] = $this->getCategoryTitle($node, 'simple');

                $parent = "";
                $thisCat = "";
                if(!empty($nodeName[1]))
                    $parent = urlencode($nodeName[1])."/";

                if(!empty($nodeName[0]))
                    $thisCat = urlencode($nodeName[0]);
            }
            $nodeTitle = $ci->db->get_where($this->categoriesTable, array('id'=>$node_id))->row();
            if($this->categoriesTable == "categories")
                $page = "products";
            else
                $page = "tech";

            $link = base_url()."home/".$page."/".urlencode($nodeTitle-
>title)."/".$node_id;
    }
}

```

```

        $title = "<a href='{$link}' title='{$title}' >{$title}</a>";
        break;

        case 'drill':    $link = $thisNode->nodeId;
        $title = "<a href='{$link}' title='{$title}' >{$title}</a>";
        break;

        case 'simple':   break;
        case 'radio':   if(!is_null($selected_id) && $node_id == $selected_id)
        $selected = " checked='checked'";
        else
        $selected = '';
        $title = "<input{$selected} id='node_{$title}' type='radio' class='checkBox' name='{$name}'
value='{$node_id}' /><span class='edit'>{$title}</span>";
        break;
        case 'nav':      $title = "<a href='#' title='{$title}'
>{$title}</a>";
        break;
        case 'default': break;
    }

    return $title;
}

function getMyTree($array=null)
{
    if(is_null($array))
    $array = $this->tree;
    if (empty($array))
    return;

    $out='<ul>';
    foreach ($array as $key => $elem)
    $out .= '<li id="'. $elem['id'] .'">' . $key . $this->getMyTree($elem) . '</li>';
    $out .= '</ul>';

    $word1 = '<li id="">subs</li>';
    $word2 = '<li id="2">id<ul></ul></li>';

    $out = str_ireplace($word1, "", $out);
    $out = str_ireplace($word2, "", $out);

    return $out;
}

function nodeExists($node_id=null)
{
    if(!is_null($node_id))
    {
        $ci = &get_instance();

        $ci->db->where('id', $node_id);
        $nodes = $ci->db->get($this->categoriesTable);

        if($nodes->num_rows() > 0)
        return true;
    }

    return false;
}

function add_node()
{
    $ci = &get_instance();
    $parent_id = $ci->input->post('nodeID');
    $title = $ci->input->post('title');

    if(!empty($title) && !empty($parent_id) && $this->nodeExists($parent_id) ==
true)
    {
        $ok = new cnode($parent_id, $this->categoriesTable);
    }
}

```

```

        $newNode = $ok->addChild($title);
        if(is_a($newNode, 'cNode'))
            echo json_encode(array('status'=>true)); die();
    }

    echo json_encode(array('status'=>false)); die();
}

function move_node($id=null, $parent_id=null)
{
    $ci = &get_instance();

    if($this->nodeExists($id) == true && $this->nodeExists($parent_id) == true)
    {
        $ci->db->where('id', $id);
        $currentNode = $ci->db->get($this->categoriesTable)->row();

        $ci->db->where('id', $id);
        $ci->db->update($this->categoriesTable, array('parent_id' =>
$parent_id));

        $childNode = new cNode( $id, $this->categoriesTable);
        $nodes = $childNode->getNodesToRoot();
        $parent_path = implode("_", array_reverse($nodes));
        $ci->db->where('id', $id);
        $ci->db->update($this->categoriesTable,
array('parent_path'=>$parent_path));

        $ci->db->where('id', $id);
        $updatedNode = $ci->db->get($this->categoriesTable)->row();

        $ci->db->like('parent_path', $currentNode->parent_path);
        $ci->db->where('id !=', $id);
        $findChildNodes = $ci->db->get($this->categoriesTable);
        if($findChildNodes->num_rows() > 0)
        {
            foreach($findChildNodes->result() as $childNode)
            {
                $new_parent_path = str_replace($currentNode-
>parent_path, $updatedNode->parent_path, $childNode->parent_path);

                $ci->db->where('id', $childNode->id);
                $ci->db->update($this->categoriesTable,
array('parent_path'=>$new_parent_path));
            }
        }

        echo json_encode(array('status'=>true)); die();
    }

    echo json_encode(array('status'=>false)); die();
}

function delete_node($id=null)
{
    $ci = &get_instance();
    if(!is_null($id) && $this->nodeExists($id) == true)
    {
        $ok = new cnode($id, $this->categoriesTable);
        $node_id = $ok->getNodeId();
        $this->deleteNode($ok);

        if($node_id != 1)
            $ci->db->delete($this->categoriesTable, array('id'=>$id));

        echo json_encode(array('status'=>true)); die();
    }

    echo json_encode(array('status'=>false)); die();
}

```



```

function edit_node()
{
    $ci = &get_instance();
    $id = $ci->input->post('cat_id');
    $title = $ci->input->post('value');
    $name = $ci->input->post('name');

    if(!is_null($title) && !is_null($id) && $this->nodeExists($id) == true && $id
!= 1)
    {
        $node = array('title'=>$title);
        $ci->db->where('id', $id);
        $ok = $ci->db->update($this->categoriesTable, $node);
        echo $title; die();
    }

    echo $name; die();
}

function edit_node_order()
{
    $ci = &get_instance();
    $sorting = 0;
    foreach($_GET as $cat_id => $sortIndex)
    {
        $sorting++;
        if(!is_null($cat_id) && $this->nodeExists($cat_id) == true && $cat_id
!= 1)
        {
            $ci->db->where('id', $cat_id);
            $ci->db->update($this->categoriesTable, array('order' =>
$sorting));
        }
        die("Sorting updated!");
    }
}
?>

```

5.5.4 crud.php (custom library για της create-read-update-delete)⁽⁹⁾

Αυτή η βιβλιοθήκη έχει φτιαχτεί επίσης **από το μηδέν**. Καλείτε με σκοπό να δημιουργηθεί ένα crud για οποιοδήποτε πίνακα στην βάση δεδομένων μας και με τον κατάλληλο κώδικα σε ένα controller και αντίστοιχα σε ένα view θα έχουμε την δυνατότητα για οποιονδήποτε πίνακα να δημιουργήσουμε – διαβάσουμε- ενημερώσουμε – διαγράψουμε εγγραφές από αυτόν τον πίνακα. Δηλαδή ένα crud library και όχι επαναληπτικό κώδικα για κάθε πίνακα. Αυτή η βιβλιοθήκη δημιουργήθηκε για να καλύψει θέματα που αφορούν τη διαχείριση του ηλεκτρονικού ινστιτούτου αισθητικής στο back office της εφαρμογής.

```
<?php

class Crud {
private $function;
private $object;
private $mode;
private $filters = array();
private $id;
private $fields;
private $lang_fields;
private $languages;
private $out_fields;
private $multi_fields_names = array();
private $view_columns = array();
private $column_names = array();
private $disabled_edit_fields = array();
private $unique_fields = array();
private $modes = array('add','edit', 'delete', 'duplicate');
private $available_modes = array('add','edit', 'delete', 'duplicate', 'gallery');
private $order_by = array('id', 'ASC');
private $mode_data;

private $results = 0;
private $per_page = 6;

function Crud(
    $function, //table name
    $object=null, //product name ex. destination
    $mode=null, // add, edit, delete
    $id=null //id for edit or delete
) {
    // get parameters
    $this->function = $function;
    if(!is_null($object))
        $this->object = $object;
    else
        $this->object = 'εγγραφής';
    $this->mode = $mode;
    $this->id = $id;

    $this->ci = &get_instance();
    $this->showAlerts($this->function);
    $this->ci->load->model('Crud_model');
    $this->fields = $this->ci->db->field_data($this->function);
    if($this->ci->db->table_exists('languages'))
        $this->languages = $this->ci->db->get('languages')->result();
    else
        $this->languages = array();
    $this->out_fields = array('id', 'lang_id', 'field_id');

    $segs = $this->ci->uri->segment_array();

    $this->ci->load->helper('text');

    $filters = array_keys($segs, 'filter');
    $order = array_keys($segs, 'orderBy');

    if(!empty($order[0]))
```

```

        $this->orderBy($segs[$order[0]+1], $segs[$order[0]+2]);

        if(!empty($filters))
            foreach($filters as $filter)
            {
                $exist = false;
                if($segs[$filter+1] != '' && $segs[$filter+2] != '')
                {
                    foreach($this->fields as $field)
                        if($field->name == $segs[$filter+1])
                            $exist = true;
                }

                if(!$exist && $segs[$filter+1] != 'drill_category_id')
                {
                    show_error("Λάθος Φίλτρο");
                }else
                {
                    $this->filters[$segs[$filter+1]] = $segs[$filter+2];
                    $this->ci->db-
>where($segs[$filter+1], $segs[$filter+2]);
                }

                if($this->ci->db->table_exists($this->function.'_lang'))
                {
                    $lang_fields = $this->ci->db->field_data($this->function.'_lang');
                    $fields_to_disable = array();
                    foreach($lang_fields as $index1=>$lang_field)
                    if(array_search($lang_field->name, array('id', 'lang_id', 'field_id')) == false)
                        foreach($this->fields as $index2=>$field)
                            if($field->name == $lang_field->name)
                                $fields_to_disable[] = $field->name;

                    $this->disable_fields($fields_to_disable);
                }

                $this->results = $this->ci->db->count_all_results($this->function);
            }

        function create()
        {
            $role = '';
            $data = '';
            $fieldsAdd = array();

            foreach($this->fields as $index=>$field)
            {
                $remove = false;
                foreach($this->disabled_edit_fields as $field_to_disable)
                    if($field_to_disable == $field->name)
                    {
                        unset($this->fields[$index]);
                        $remove = true;
                        break;
                    }

                if($remove)
                    break;

                if($field->field_type == 'connection_id')
                {
                    list($connection_table, $display_field) = split('/[-.]/', $this->multi_fields_names[$field->name]);

                    if($this->ci->db->table_exists($connection_table))
                    {
                        foreach($this->unique_fields as $unique_field)
                            if($unique_field == $field->name)
                            {
                                $alreadyInside = $this->ci->db->get($this->function)->result();
                                foreach($alreadyInside as $ready)

```

```

        $out_ids[] = $ready->{$unique_field};
        $this->ci->db->where_not_in('id', $out_ids);
    }
    $fieldData = $this->ci->db->get($connection_table)->result();
    }
    else
    {
        show_error("Ο πίνακας ".$connection_table." δεν υπάρχει!");
    }

    if(array_key_exists($field->name, $this->multi_fields_names))
    {
        foreach($fieldData as $fieldDataValue)
        $fieldDataValue->multi_title = $fieldDataValue->{$display_field};
    }else
        foreach($fieldData as $fieldDataValue)
        $fieldDataValue->multi_title = $fieldDataValue->title;

    $this->fields[$index]->multi[$field->name] = $fieldData;
}

if($field->field_type == 'status')
{
    $statusData = array(
        (object)array('id'=>'1', 'multi_title'=>'active'),
        (object)array('id'=>'0', 'multi_title'=>'inactive')
    );

    $this->fields[$index]->multi[$field->name] = $statusData;
}

if($this->ci->db->table_exists($this->function.'_lang'))
{
    $language_fields = array();
    foreach($this->languages as $language)
    {
        $lang_fields = $this->ci->db->field_data($this->function.'_lang');

        foreach($lang_fields as $index=>$lang_field)
        foreach($lang_field as $indexName=>$value)
        if(array_search($value, $this->out_fields) != false)
            unset($lang_fields[$index]);

        unset($lang_fields[0]);
    }
    foreach($lang_fields as $lang_field)
    $language_fields[] = $lang_field->name;

    if($this->ci->db->table_exists('user_languages'))
        $user_languages = '';

    foreach($this->languages as $language)
        if(array_search($field->name, $language_fields) != false)
        {
            $new_lang_field = clone $field;
            $new_lang_field->name = $field->name.'_'.$language->title;
            $this->ci->db->where('field_id', $this->id);
            $this->ci->db->where('lang_id', $language->id);
            $lang_data = $this->ci->db->get($this->function.'_lang')->row();
            $new_lang_field->value = $lang_data->{$field->name};

            if($field->field_type == 'rich_text')
            {
                $lang_field_name = "lang_fields[{$language->title}][.str_replace('_', '$language->title', ''),
                $new_lang_field->name)."]";
                $new_lang_field->cidsrte = new cidsrte(array(
                    "id"=>$new_lang_field->name,
                    "name"=>$lang_field_name,
                    "str"=>false,
                    "resizable"=>false
                ));
            }
        }
    }
}

```



```

unset($_POST['align']);

foreach($_POST as $post => $value)
{
    if(strpos($post, 'time_for'))
    {
        $datefieldName = split('_for_', $post, 2);
        $data->{$datefieldName[1]} = $data->{$datefieldName[1]}. ' '.$this->ci->input->post($post);
        }else
        $data->$post = trim($this->ci->input->post($post));
        }

        if(is_null($this->id))
            $this->mode = 'add';
        else if(!is_null($this->id))
            $this->mode = 'edit';

        $this->ci->Crud_model->{$this->mode}($this->function, $data,
$this->id);
    }else
        $this->ci->Crud_model->{$this->mode}($this->function, $data,
$this->id);
    }else
    {
        if(!is_numeric($this->mode) && $this->mode != 'all')
        {
            $this->ci->load->view('admin/results', array(
                'function'=>$this->function,
                'data'=>$data,
                'role'=>$role,
                'object'=>$this->object,
                'modes'=>$this->modes,
                'fields'=>$this->fields,
                'per_page'=>$this->per_page,
                'results'=>$this->results,
                'filters'=>$this->filters,
                'order_by'=>$this->order_by
            ));
        }else
        {
            $this->ci->_setOutputMode('OUTPUT_NORMAL');
            $this->ci->db->order_by($this->order_by[0], $this->order_by[1]);
            $limit = null;
            $offset = null;

            if($this->mode != 'all')
            {
                $offset = 0;
                for($k = 1; $k < $this->mode; $k++)
                    $offset += $this->per_page;

                $limit = $this->per_page;
            }
            foreach($this->filters as $filterName => $filterValue)
                $this->ci->db->where($filterName, $filterValue);

            $data = $this->ci->db->get($this->function, $limit, $offset)->result();

            foreach($this->fields as $index=>$field)
            {
                if($field->field_type == 'connection_id')
                {
                    list($connection_table, $display_field) = split('[-.]', $this->multi_fields_names[$field->name]);
                    if($this->ci->db->table_exists($connection_table))
                        $fieldData = $this->ci->db->get($connection_table)->result();
                    else
                        show_error("Ο πίνακας '$connection_table.' δεν υπάρχει!");
                }

                if(array_key_exists($field->name, $this->multi_fields_names))
                {
                    foreach($fieldData as $fieldDataValue)
                        $fieldDataValue->multi_title = $fieldDataValue->{$display_field};
                }
            }
        }
    }
}

```

```

}else

foreach($fieldData as $fieldDataValue)
    $fieldDataValue->multi_title = $fieldDataValue->title;

    $this->fields[$index]->multi[$field->name] = $fieldData;
}
if($field->field_type == 'status')
{
$statusData = array(
    (object)array('id'=>'1', 'multi_title'=>'active'),
    (object)array('id'=>'0', 'multi_title'=>'inactive')
);

$this->fields[$index]->multi[$field->name] = $statusData;
}
}

foreach($this->languages as $language)
    foreach($this->fields as $index => $field)
        if(strpos($field->name, '_'.$language->title) !== false )
            unset($this->fields[$index]);

    if(!empty($this->view_columns))
        foreach($this->fields as $index => $field)
            if(array_search($field->name, $this->view_columns) === false && $field->name != 'id')
                unset($this->fields[$index]);

                $this->ci->load->model('authmodel');
                $role = $this->ci->authmodel->role;
                if($role == 'user')
                {
                    $this->remove_mode('add');
                    $this->remove_mode('duplicate');
                    $this->remove_mode('delete');
                }

                $this->ci->load->view('admin/view', array(
                    'function'=>$this->function,
                    'mode_data'=>$this->mode_data,
                    'filters'=>$this->filters,
                    'data'=>$data,
                    'role'=>$role,
                    'object'=>$this->object,
                    'modes'=>$this->modes,
                    'fields'=>$this->fields,
                    'order_by'=>$this->order_by
                ));
            }
        }
    }

function set_multi_field_title($field_name, $title = '')
{
    $this->multi_fields_names[$field_name] = $title;
}

function disable_fields(array $fields_to_disable)
{
    if(is_array($fields_to_disable) && !empty($fields_to_disable))
        $this->disabled_edit_fields = $fields_to_disable;
    else
        show_error("Το πεδίο αυτό δεν υπάρχει");
}

function set_unique_fields(array $fields_to_unique)
{
    if(is_array($fields_to_unique) && !empty($fields_to_unique))
        $this->unique_fields = $fields_to_unique;
    else

```

```

        show_error("Το πεδίο αυτό δεν υπάρχει");
    }
    function set_columns()
    {
        $_1starg = func_get_arg(0);
        $columns = is_array($_1starg) ? $_1starg : func_get_args();

        $this->view_columns = $columns;
    }

    function set_types()
    {
        $_1starg = func_get_arg(0);
        $field_types = is_array($_1starg) ? $_1starg : func_get_args();

        foreach($this->fields as $index=>$field)
        {
            if(isset($field_types[$field->name]))
            {
                $field->field_type = $field_types[$field->name];
            }else if($field->name == 'status')
            {
                $field->field_type = 'status';
            }else if(strpos($field->name, '_file'))
            {
                $field->field_type = 'file';
            }else if(strpos($field->name, 'image'))
            {
                $field->field_type = 'image';
            }else if(strpos($field->name, '_id'))
            {
                $field->field_type = 'connection_id';
            }else if($field->type == 'blob' || $field->type == 'text')
            {
                $field->field_type = 'rich_text';
            }else if($field->type == 'blob' || $field->type == 'text')
            {
                $field->field_type = 'rich_text';
            }else if($field->type == 'timestamp')
            {
                $field->field_type = 'timestamp';
            }else
            {
                $field->field_type = 'text';
            }
        }
    }
}

function add_mode($mode=null, $data=null)
{
    if(!is_null($mode))
    {
        if(!is_null($data))
            $this->mode_data = $data;
        if(array_search($mode, $this->available_modes) === false)
            show_error("Δεν υπάρχει η λειτουργία: $mode");
        if(array_search($mode, $this->modes) === false)
            $this->modes[] = $mode;
        else
            show_error("Η λειτουργία αυτή είναι ήδη ενεργοποιημένη!");
    }
}

function remove_mode($mode=null)
{
    $modes = array();
    if(!is_null($mode))
    {
        if(!is_array($mode))
            $modes[] = $mode;
        else
            $modes = $mode;
    }
}

```



```

        foreach($modes as $mode)
        {
            if(array_search($mode, $this->available_modes) === false)
                show_error("Δεν υπάρχει αυτή η λειτουργία: $mode");
            $modeArrayPos = array_search($mode, $this->modes);
            if($modeArrayPos !== false)
                unset($this->modes[$modeArrayPos]);
        }
    }
}

function set_column_names(array $column_names)
{
    $this->column_names = $column_names;
}

function orderBy($column = 'id', $order = 'ASC')
{
    $this->order_by = array();
    $this->order_by[] = $column;
    $this->order_by[] = $order;
}

function set_per_page($per_page = 6)
{
    if(is_numeric($per_page) && $per_page <= 100)
        $this->per_page = $per_page;
}

private function showAlerts($function)
{
    if($this->ci->session->flashdata($function.'Heading'))
    {
        $heading = $this->ci->session->flashdata($function.'Heading');
        $message = $this->ci->session->flashdata($function.'Message');

        alertMessage($heading, $message);
    }

    if($this->ci->session->flashdata($function.'Highlight'))
        highlightRow($this->ci->session->flashdata($function.'Highlight'));
}

}

?>

```

6 Αποτέλεσμα

6.1 Συμπεράσματα

Τα αποτελέσματα αυτού του project ήταν να μπορέσουμε να δημιουργήσουμε ένα site το οποίο θα μπορεί να διαχειριστεί από χρήστες με δικαιώματα και έτσι ήταν επιτακτική ανάγκη να δημιουργηθεί ένα custom CMS για αυτό. Το stavgianCMS μπορεί πλέον να προσαρμοστεί σε οποιαδήποτε μελλοντική σελίδα και θα κάνει τη ζωή του developer πιο εύκολη.

Αυτή η πτυχιακή σίγουρα ωφέλησε εμένα προσωπικά και θα μπορέσει να βοηθήσει νέους web Developers που αναπτύσσουν εφαρμογές PHP από το μηδέν. Με τη χρήση του CodeIgniter μπορέσαμε να επικεντρωθούμε σε βασικά ζητήματα ανάπτυξης. Ίσως με λιγότερο κώδικα και σίγουρα πολύ πιο τακτοποιημένο κώδικα. Σε μελλοντικές αλλαγές η πρόσβαση στην πληροφορία που θα ζητηθεί, θα είναι σε συγκεκριμένο σημείο του κώδικα οπότε μπορούμε ανά πάσα στιγμή να ξέρουμε που θα οδηγηθούμε. Έτσι συνειδητοποιούμε τι προσφέρει το MVC.

6.2 Μελλοντική Εργασία και Επεκτάσεις

Μελλοντικά η πτυχιακή μπορεί να επεκταθεί όσο θέλουμε. Δεν υπάρχει περιορισμός. Τα πάντα μπορούν να αλλάξουν και να τροποποιηθούν όπως μας ζητηθεί.

Στο συγκεκριμένο αντικείμενο μελλοντικά θα μπορούσε να επεκταθεί κλείνοντας ραντεβού οι χρήστες, ανάλογα την διαθεσιμότητα και ο διαχειριστής να είναι αυτός που θα διαχειρίζεται την διαθεσιμότητα υπηρεσιών και γενικά να δημιουργηθεί ένας έξυπνος αλγόριθμος που θα αλλάζει συνεχώς και θα προωθούνται υπηρεσίες και προϊόντα ανάλογα με την ζήτηση από τους πελάτες ή αυτόματες ενέργειες θα είναι αυτές που θα παίρνουν αποφάσεις πάντα με τον διαχειριστή ή τον εκάστοτε ιδιοκτήτη στο παρασκήνιο.

Βιβλιογραφία

- [1. Google inc. *www.google.com*. [Ηλεκτρονικό]
2. **Wikipedia**. <http://en.wikipedia.org/wiki/Wiki>. <http://en.wikipedia.org/wiki/Wiki>. [Ηλεκτρονικό]
3. Mozilla. *www.mozilla-europe.org/el/*. [Ηλεκτρονικό]
4. **CodeIgniter**. *codeigniter.com/user_guide/*. [Ηλεκτρονικό]
5. Wamp. *www.wamp.com*. [Ηλεκτρονικό]
6. **Upton, David**. *CodeIgniter for Rapid PHP Application Development*. Birmingham, B27 6PA, UK. : Packt Publishing Ltd., 2007.
7. **net**. Net Tutorials. <http://net.tutsplus.com/>. [Ηλεκτρονικό]
8. **Dan Wellman - Packt Publishing**. *jQuery UI 1.7*. Birmingham, B27 6PA, UK. : Packt Publishing, 2009.
9. PHP.net. <http://www.php.net/manual/en/>. [Ηλεκτρονικό]
10. jQuery. *jquery.com*. [Ηλεκτρονικό]
11. **codeigniter.com**. *codeigniter.com*. *codeigniter.com*. [Online] codeigniter.com, 2011.
12. **Jose Argudo Blanco, David Upton**. *CodeIgniter 1.7*. s.l. : Packt Publishing, 2009.
13. **code.google.com**. *code.google.com*. *code.google.com*. [Ηλεκτρονικό] Goggle, 2011.
14. **Thomas Myer - Wiley Publishing, Inc**. *Professional CodeIgniter®*. Indianapolis, IN 46256 : Wiley Publishing, Inc., 2009.

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α: Παρουσίαση πτυχιακής σε powerpoint slides.

Παράρτημα Α



Α.Τ.Ε.Ι Κρήτης
Σχολή Τεχνολογικών εφαρμογών
Τμήμα εφαρμοσμένης Πληροφορικής & πολυμέσων



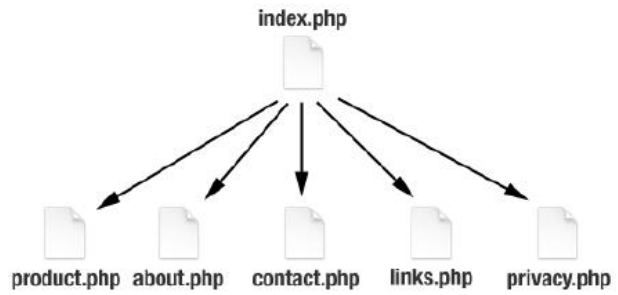
Ηλεκτρονικό ινστιτούτο αισθητικής
με χρήση Opensource MVC PHP Framework

Πτυχιακή εργασία
Σταυγιανουδάκης Ιωάννης
Α.Μ:1226

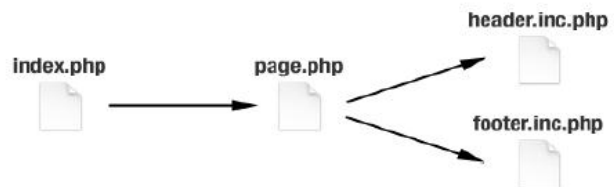


Ιστορία ανάπτυξης ιστοσελίδων

Παλαιότερα



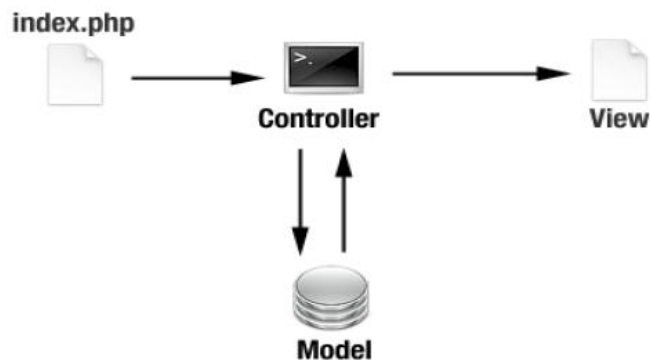
Σήμερα





Εξέλιξη ανάπτυξης ιστοσελίδων

Ανάπτυξη με χρήση Framework



Βασική ιδέα του Framework

Ένα web application framework :

- Είναι ένα Framework Λογισμικού
- Σχεδιάστηκε για την υποστήριξη ανάπτυξης:
 - ❖ Δυναμικών ιστοσελίδων
 - ❖ Web εφαρμογών
 - ❖ Web υπηρεσιών
- Στόχος είναι η αντιμετώπιση κάποιων γενικών δραστηριοτήτων που χρησιμοποιούνται στην ανάπτυξη ιστοσελίδων.
 - ❖ Βιβλιοθήκες για πρόσβαση στην Βάση δεδομένων (Database Libraries)
 - ❖ Διαχείριση Session
 - ❖ Συχνά προωθούν την επαναχρησιμοποίηση κώδικα
 - ❖ Πολλά πολλά άλλα...



Τι είναι το CodeIgniter ?

- ☆ Ένα *Web Application Framework* ανοικτού κώδικα (php)
- ☆ Παρέχετε ΔΩΡΕΑΝ
- ☆ Γρήγορο
- ☆ Είναι ένα πολύ δυνατό πακέτο.
- ☆ Μπορεί να επεκταθεί εύκολα
- ☆ Μηδαμινή παραμετροποίηση
- ☆ Λεπτομερώς τεκμηριωμένο
- ☆ Παράγει καθαρά URLs
- ☆ MVC (*Model View Controller*) αρχιτεκτονική
- ☆ Υποστηρίζει πολλαπλές βάσεις δεδομένων
- ☆ Αντικειμενοστραφής συμπεριφορά (*DB Objects*)
- ☆ Φιλική κοινότητα χρηστών

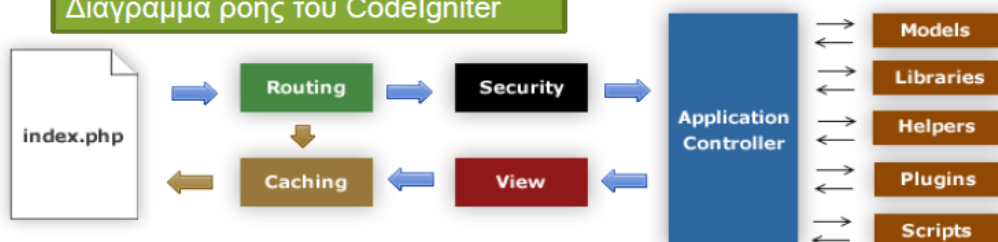


Χαρακτηριστικά CodeIgniter

Κάποια από τα χαρακτηριστικά του:

- ☆ *Active Records support*
- ☆ *Autoload class*
- ☆ *Επικύρωση δεδομένων*
- ☆ *Ασφάλεια και XSS Filtering*
- ☆ *Παρέχετε ΔΩΡΕΑΝ*
- ☆ *E-mail classes*
- ☆ *Χειρισμός εικόνων*
- ☆ *Σελιδοποίηση*
- ☆ *Συμπίεση*
- ☆ *Διαχείριση errors*
- ☆ *Helpers*

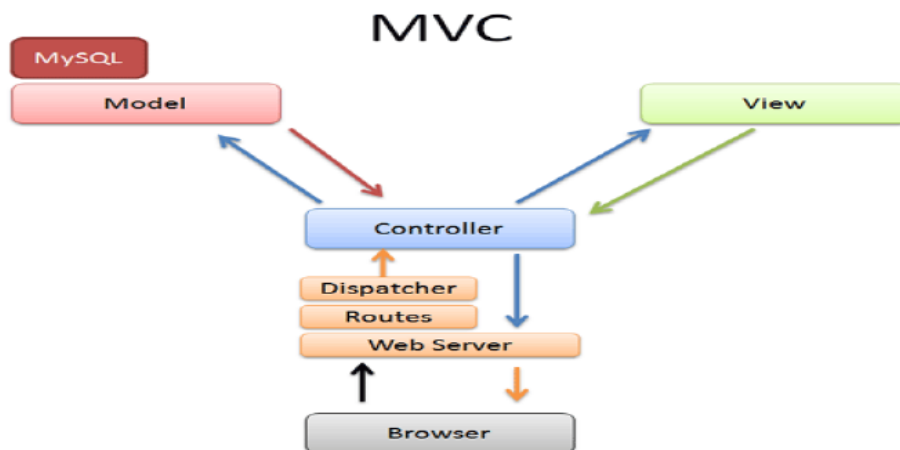
Διάγραμμα ροής του CodeIgniter



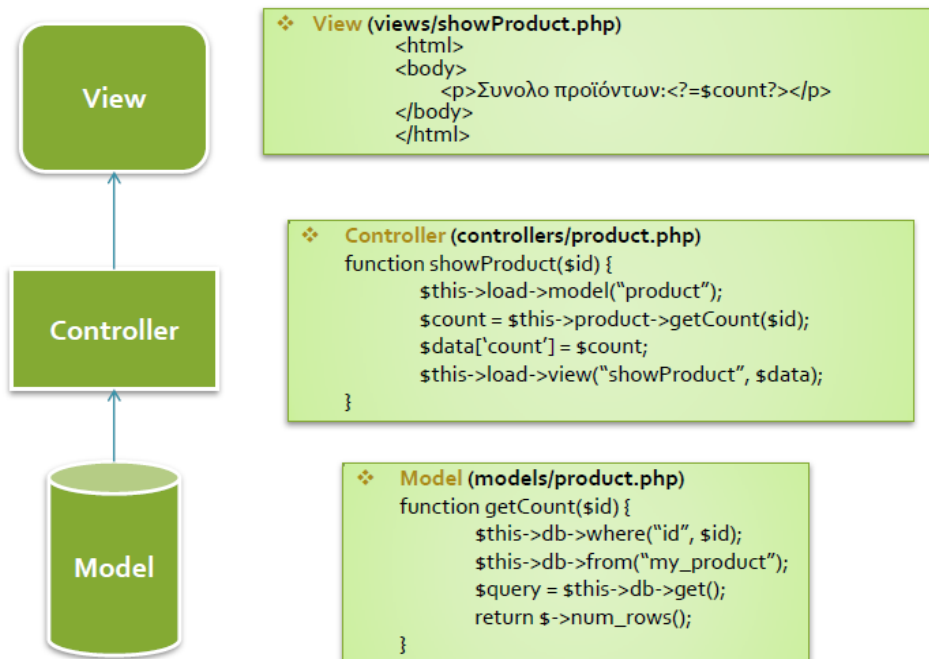


Αρχιτεκτονική MVC 1/2

- Διαχωρίζει τη διεπαφή χρήστη από το λειτουργικό κομμάτι .
- **Model**: Συμπυκνώνει τα δεδομένα της βασικής εφαρμογής
- **View** : λαμβάνει δεδομένα από το μοντέλο και τα παρουσιάζει στο χρήστη.
- **Controller** : λαμβάνει και μεταφράζει εισόδους (inputs) σε αιτήματα εισόδου σχετικά με το MODEL ή το VIEW

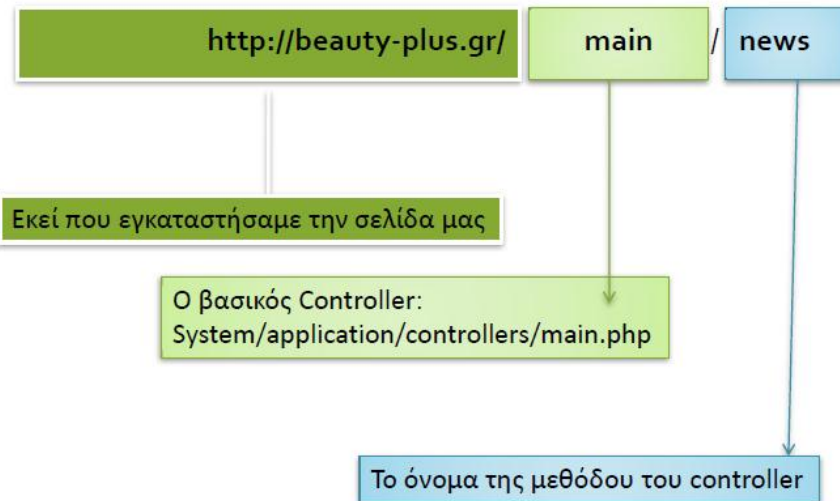


Αρχιτεκτονική MVC 2/2



CodeIgniter URL

Τα URL στο CodeIgniter είναι και τμηματικά (Segment Based).



CodeIgniter Libraries

Κλάσεις ειδικού σκοπού

| | | | |
|--------------------|-----------------|--------------------|-------------------|
| Benchmarking | Database | Encryption | Calendaring |
| FTP | Table | File Uploading | Email |
| Image Manipulation | Pagination | Input and Security | HTML |
| Cart | Parser | Session | Template |
| Unit Testing | User Agent | URI | Validation |

Χρησιμοποιώντας libraries

```
$this->load->library('database');
```



CodeIgniter Helpers

Κάποια από τα helpers που χρησιμοποιήσαμε.

| | | | | | |
|-----------|--------|-----------|------------|------------|--------|
| Array | Date | File | HTML | Smiley | Text |
| URL | Cookie | Download | Form | Security | String |
| Directory | E-mail | Inflector | XML Parser | Typography | |

Χρησιμοποιώντας ένα Helper μέσα σε ένα Controller

```

$this->load->helper('helper_name');
$this->load->helper(array('form','url'));
    
```

Form Helper

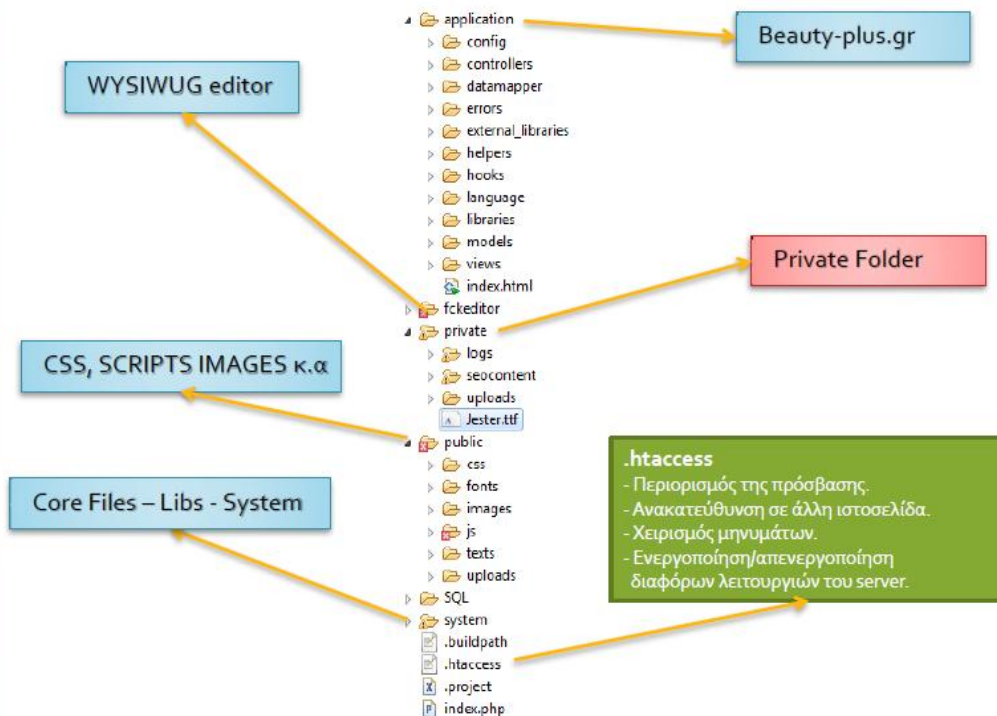
- > form_open()
- > form_open_multipart()
- > form_input()
- > form_textarea()
- > form_checkbox()
- > form_submit()
- > form_close()

URL Helper

- > site_url()
- > base_url()
- > anchor()
- > anchor_popup()
- > mailto()



Δομή Αρχείων Ιστοτιούτου Αισθητικής 1/3





Δομή Αρχείων Ινστιτούτου Αισθητικής 2/3

Models

- models
 - _template.php
 - authmodel.php
 - beauty_model.php
 - Crud_model.php
 - ion_auth_model.php
 - newsletter_model.php
 - products_model.php
 - reservation_model.php
 - services_model.php
 - users_crud.php
 - Users_relations.php
 - users.php

Views

- views
 - admin
 - form.php
 - home.php
 - newsletter.php
 - nodes.php
 - orders.php
 - results.php
 - schedule.php
 - uploading.php
 - users.php
 - view.php
 - auth
 - auth_old
 - controls
 - dmz_htmform
 - emails
 - inc
 - loading
 - member
 - templates
 - beauty.php
 - blank.php
 - default.php
 - stav.php
 - stavgianCMS.php
 - blank.php
 - index.html

Controllers

- beauty-plus
 - .settings
 - application
 - config
 - controllers
 - admin
 - newsletter.php
 - orders.php
 - schedule.php
 - users.php
 - access_denied.php
 - auth.php
 - back.php
 - cart.php
 - eservices.php
 - index.html
 - main.php
 - media.php
 - members.php



Δομή Αρχείων Ινστιτούτου Αισθητικής 3/3

Custom Libs

- libraries
 - cnode.php
 - Crud.php
 - Datagrid.php
 - datamapper.php
 - index.html
 - Ion_auth.php
 - MY_Controller.php
 - MY_Session.php
 - PhpCaptcha.php
 - TemplateController.php

Helpers

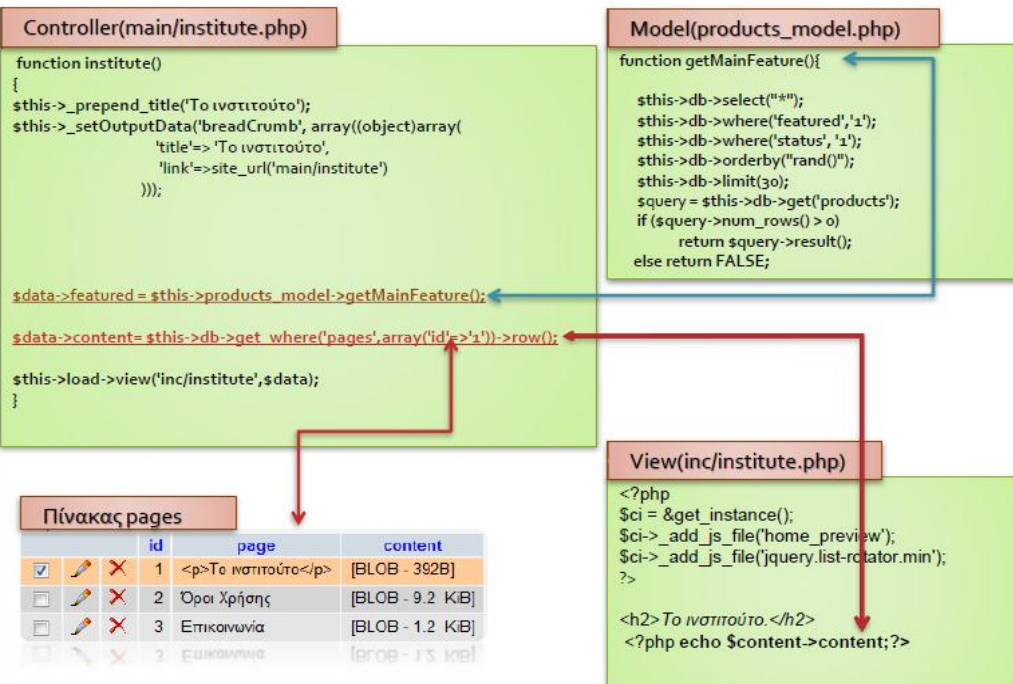
- helpers
 - array_helper.php
 - auth_helper.php
 - autoload.php
 - common_helper.php
 - cookie_helper.php
 - date_helper.php
 - exception_helper.php
 - file_helper.php
 - icons_helper.php
 - image_helper.php
 - index.html
 - inflector_helper.php
 - json_helper.php
 - MY_url_helper.php
 - print_helper.php

Config

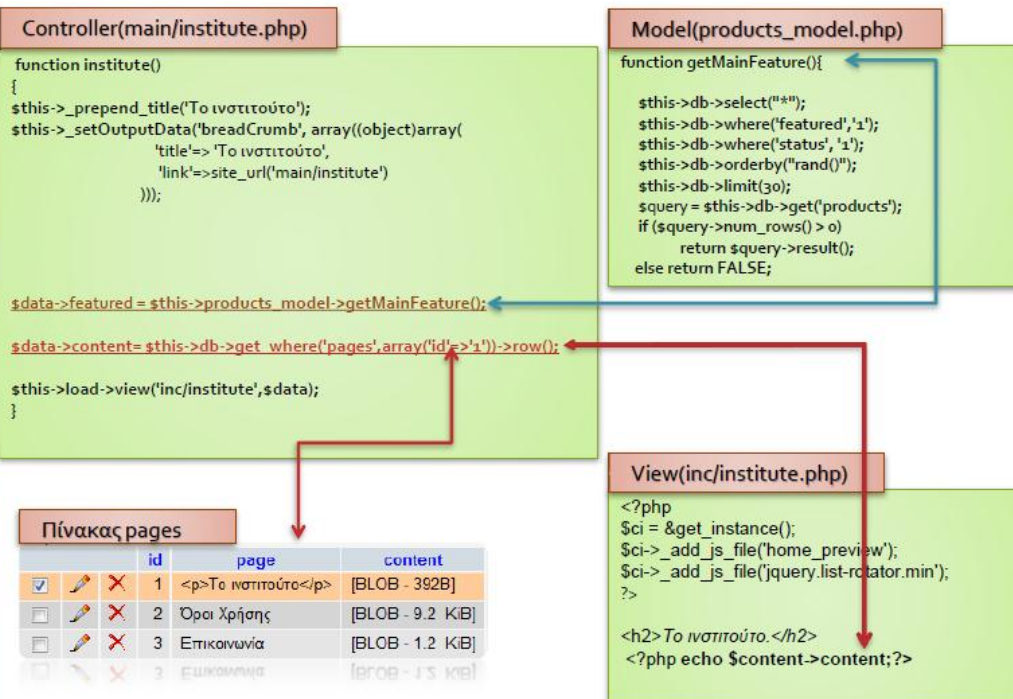
- config
 - autoload.php
 - config.php
 - constants.php
 - database.php
 - datamapper.php
 - doctypes.php
 - email.php
 - hooks.php
 - index.html
 - ion_auth.php
 - mailer.php
 - mimes.php
 - pagination.php
 - routes.php
 - smileys.php
 - user_agents.php



Παράδειγμα 1



Παράδειγμα 1





Δυνατότητες beauty-plus.gr ^{1/2}

Σαν απλός επισκέπτης

- ☆ *Επικοινωνία*
- ☆ *Δυνατότητα αγοράς προϊόντων*
- ☆ *Δυνατότητα κλεισίματος ραντεβού εύκολα*
- ☆ *Πλήρες σύστημα χρήστη (authentication authorization)*
- ☆ *Ιστορικό αγορών*
- ☆ *Ιστορικό υπηρεσιών*
- ☆ *Περιοχή χρήστη*
- ☆ *Newsletter system*
- ☆ *Νέα & ανακοινώσεις*
- ☆ *Galleries*



Δυνατότητες beauty-plus.gr ^{2/2}

Σαν διαχειριστής

- **StavqianCMS**
 - Προβολή γενικών πληροφοριών
 - Στατιστικά στοιχεία
 - Δημιουργία Event
 - Πλήρης Διαχείριση:
 - Νέων – ανακοινώσεων, παραγγελιών, Ραντεβού
 - Προϊόντων, Υπηρεσιών
 - Gallery προϊόντων, υπηρεσιών, διαφημίσεων κ.α
 - Χρηστών – Προσωπικού
 - Δικαιωμάτων
 - Περιεχόμενο σελίδων
 - Faq's και πολλά άλλα...



Τέλος

Ερωτήσεις - Απαντήσεις