



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη Web εφαρμογής για συσχέτιση γονιδίων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αναγνώστου Νικόλαος

Επιβλέπων : Διονύσιος-Δημήτριος Κουτσούρης

Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη Web εφαρμογής για συσχέτιση γονιδίων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αναγνώστου Νικόλαος

Επιβλέπων : Διονύσιος-Δημήτριος Κουτσούρης

Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Οκτωβρίου 2014

.....
Διονύσιος Δημήτριος Κουτσούρης
Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Ματσόπουλος
Επ.Καθηγητής Ε.Μ.Π.

.....
Δημήτριος Φωτιάδης
Καθ.Παν.Ιωαν.

Αθήνα , Οκτώβριος 2014

.....

Αναγνώστου Νικόλαος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Αναγνώστου Νικόλαος, 2014

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, θα ήθελα αρχικά να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Κουτσούρη Διονύσιο-Δημήτριο για την ανάθεσή της, καθώς και τον υποψήφιο διδάκτορα κ. Αναστασίου Αθανάσιο για την καθοδήγησή του και για το γεγονός ότι μου έδωσε τη δυνατότητα να ασχοληθώ με αυτό το εξαιρετικά ενδιαφέρον θέμα.

Τέλος, οφείλω ιδιαίτερες ευχαριστίες στους γονείς μου, στα πολυαγαπημένα μου αδέρφια και στους φίλους μου για τη στήριξη και τη βοήθειά τους καθ'όλη τη διάρκεια των σπουδών μου.

Περίληψη

Η παρούσα διπλωματική έχει ως στόχο την ανάπτυξη μιας δυναμικής ιστοσελίδας, η οποία θα σχετίζεται με το χώρο της ηλεκτρονικής υγείας (E-Health). Η ιστοσελίδα θα παρέχει πληροφορίες σχετικά με το EurHEALTHAgeing project και θα είναι δυναμική. Θα μπορεί δηλαδή οποιοσδήποτε χρήστης να εγγράφεται σε αυτή, να αναζητά πληροφορίες για συγκεκριμένα γονίδια, να παρεμβαίνει στα ήδη υπάρχοντα, ακόμα και να προσθέτει καινούργια. Για το σκοπό αυτό χρησιμοποιήσα κατά κύριο λόγο τη γλώσσα προγραμματισμού PHP, μια γλώσσα ειδικά σχεδιασμένη για τη δημιουργία δυναμικών ιστοσελίδων, και τον εξυπηρετητή βάσεων δεδομένων MySQL.

Η μελέτη των γονιδίων θα συμβάλλει στο γενικότερο σκοπό αυτού του project, όπου είναι η ανάγκη του προσδιορισμού διάφορων γενετικών παραγόντων που καθορίζουν την πορεία της υγείας ενός ατόμου. Η αποτελεσματική πολιτική που ακολουθείται για την υγεία καθ' όλη τη διάρκεια της ζωής, μπορεί να υποστηρίξει σε μεγάλο βαθμό την υγιή γήρανση του πληθυσμού. Για αυτό το λόγο υπάρχει επιτακτική ανάγκη, να εντοπίζονται διάφορα γεγονότα από τα πρώτα χρόνια της ζωής ενός ατόμου.

Λέξεις-κλειδιά

δυναμική ιστοσελίδα, ηλεκτρονική υγεία, MySQL, PHP, Apache, βάσεις δεδομένων, διαδίκτυο, προγραμματισμός, πίνακες

Abstract

The scope of this thesis is the development of a dynamic web page, which belongs to the area of E-Health. The web page will provide information about the EurHealthAgeing project and will be dynamic. That means, every user could register to the webpage, search for information about specific genes, modify existing genes or add new to the database. For that purpose the PHP programming language is mainly used, a language specifically designed for the creation of dynamic web pages and also the MySQL database management system.

The study of genes will contribute to the general purpose of this project, which is the necessity to determine different genetic factors that define the health of a person, in the future. The effective policy that is followed for the health, throughout the lifetime, could support in a high level the healthy aging of the population. For that reason there is a necessity to identify facts from the early years of lifetime.

Key-words

dynamic web page, E-health, MySQL, PHP, Apache , databases, internet, programming, tables

Πίνακας περιεχομένων

1 Εισαγωγή.....	1
1.1 Αντικείμενο της διπλωματικής.....	1
2 Ηλεκτρονική υγεία , διαδίκτυο και εφαρμογές βάσεων δεδομένων.....	2
2.1 Ηλεκτρονική υγεία.....	2
2.2 Δυναμική ιστοσελίδα.....	2
2.3 Το Διαδίκτυο (Internet).....	3
2.4 Η αρχιτεκτονική πελάτη-εξυπηρετητή (client-server).....	4
2.5 Η αρχιτεκτονική 3 στρωμάτων (3-Tier Architecture).....	4
2.6 Το στρώμα πελάτη.....	5
3 Το μεσαίο στρώμα (Middle Tier).....	6
3.1 Γενικά - Ρόλος.....	6
3.2 Εξυπηρετητές Δικτύου.....	6
3.3 Πρωτόκολλο HTTP.....	6
3.4 Γενικά για τον Apache και τις τρέχουσες εκδόσεις του.....	7
3.4.1 Επιλογή της μεθόδου εγκατάστασης.....	7
3.4.2 Εγκατάσταση του Apache Web Server στα Windows.....	7
3.4.3 Το αρχείο διαμόρφωσης του Apache.....	11
4 MySQL.....	12
4.1 Εισαγωγή στη MySQL.....	12
4.2 Εγκατάσταση και Διαμόρφωση της Mysql.....	12
4.3 Εγκατάσταση της MySQL σε συστήματα Linux/Unix.....	12
4.4 Εγκατάσταση της MySQL στα Windows.....	13
4.5 Πιθανά προβλήματα κατά την εγκατάσταση.....	15
4.6 Σημαντικές οδηγίες ασφαλείας.....	15
4.7 Σύστημα δικαιωμάτων της MySql.....	15
4.8 Διαχείριση των χρηστών.....	16
4.9 Τύποι δεδομένων της MySql.....	16
4.9.1 Αριθμητικοί τύποι δεδομένων.....	17
4.9.2 Τύποι δεδομένων για τιμές ημερομηνίας-ώρας.....	17

4.9.3 Αλφαριθμητικοί τύποι.....	17
4.10 Εισαγωγή στην SQL.....	18
4.11 Εντολές της SQL και χρήση τους.....	18
5 Δικτυακός προγραμματισμός με την PHP.....	20
5.1 Εισαγωγή στην PHP.....	20
5.2 Εγκατάσταση της PHP.....	20
5.2.1 Εγκατάσταση της PHP σε συστήματα Linux/Unix.....	20
5.2.2 Εγκατάσταση της PHP στα Windows.....	21
5.3 Τα βασικά της PHP.....	23
5.4 Σύνδεση της PHP με τη MySQL.....	23
6 HTML.....	25
6.1 Γενικά χαρακτηριστικά.....	25
6.2 Επεξεργαστής κειμένου.....	25
6.3 Δομή ενός εγγράφου HTML.....	25
6.4 Χρήση της HTML μαζί με PHP.....	27
7 Υλοποίηση της δυναμικής ιστοσελίδας.....	28
7.1 Δημιουργία νέου χρήστη και απόκτηση δικαιωμάτων.....	28
7.2 Διάγραμμα βάσης δεδομένων και δημιουργία πινάκων.....	28
7.3 Ρυθμίσεις στον Apache Server.....	33
7.4 Παρουσίαση της δυναμικής ιστοσελίδας.....	33
Κώδικας.....	47
Βιβλιογραφία.....	98

1

Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζουμε το αντικείμενο της διπλωματικής.

1.1 Αντικείμενο της διπλωματικής

Αντικείμενο της παρούσας διπλωματικής είναι η δημιουργία μιας δυναμικής ιστοσελίδας (web site), που έχει ως στόχο τη συσχέτιση γονιδίων, τα οποία καθορίζουν την πορεία της υγείας ενός ατόμου . Στον δικτυακό αυτό τόπο παρουσιάζονται πληροφορίες σχετικά με το EurHEALTHAgeing project, τι περιλαμβάνει και το σκοπό δημιουργίας του. Επίσης ο χρήστης μπορεί να αναζητά πληροφορίες για τα υπάρχοντα Entity Types, Entities και associations αλλά και να προσθέτει καινούργια , να διορθώνει και να διαγράφει όποια δεν επιθυμεί. Όλες οι πληροφορίες που παρουσιάζονται στην ιστοσελίδα αποθηκεύονται σε μία βάση δεδομένων, που δημιουργήθηκε στον MySQL Server, προκειμένου το περιεχόμενο του site να είναι δυναμικό και να μεταβάλλεται εύκολα. Το υλικό αυτό θα είναι προσβάσιμο, μόνο σε όσους χρήστες κάνουν εγγραφή στην ιστοσελίδα μας.

2

Ηλεκτρονική υγεία , διαδίκτυο και εφαρμογές βάσεων δεδομένων

2.1 Ηλεκτρονική υγεία (E-health)

Τα τελευταία χρόνια η έλευση νέων τεχνολογιών δεν έχει αφήσει ανεπηρέαστη την υγεία. Είναι πολύ πιθανό να έχετε ακούσει τον όρο ηλεκτρονική υγεία (e-health). Πρόκειται ουσιαστικά για ένα νέο μοντέλο περίθαλψης που έχει επίκεντρο τον πολίτη. Χρησιμοποιείται πλέον με μεγάλη επιτυχία σε ορισμένες χώρες της Ευρώπης ενώ γίνονται αρκετές προσπάθειες να εφαρμοστεί και στην Ελλάδα. Δεν υπάρχει ακριβής ορισμός για την ηλεκτρονική υγεία. Πρακτικά θα μπορούσαμε να πούμε ότι αναφέρεται στη μετεξέλιξη των παραδοσιακών συστημάτων υγείας, στα οποία έχουν ειησέλθει νέες τεχνολογίες και το διαδίκτυο.

Οι νέες αυτές τεχνολογίες πληροφοριών που έχουν προσαρμοστεί στην υγεία, στοχεύουν στην καλύτερη πρόληψη, διάγνωση, θεραπεία παρακολούθηση και διαχείριση της υγείας και του τρόπου ζωής. Ειδικά σε περιόδους οικονομικής κρίσης όπως περνάει η χώρα μας, η ανάπτυξη ηλεκτρονικών υπηρεσιών υγείας μπορεί να συμβάλει στην ανάπτυξη της ανταγωνιστικότητας, να εξασφαλίσει τη βιωσιμότητα των συστημάτων υγείας, χωρίς όμως να επηρεαστεί το κόστος και η ποιότητα της υγειονομικής περίθαλψης.

Ορισμένες εφαρμογές της ηλεκτρονικής υγείας, όπου ίσως να είναι γνωστές σε μερικούς, είναι ο ηλεκτρονικός ιατρικός φάκελος, ο οποίος περιέχει πληροφορίες για κάποιον ασθενή όπως ιατρικό ιστορικό, εγχειρήσεις, ιατροφαρμακευτική περίθαλψη και νοσηλείες οι οποίες μπορούν να μελετηθούν και να αξιοποιηθούν από κατάλληλο προσωπικό όποτε είναι απαραίτητο. Άλλη μια εφαρμογή είναι τα online φαρμακεία τα οποία επιτρέπουν τη διεκπεραίωση συναλλαγών μέσω Internet. Τέλος τα τελευταία χρόνια συναντάται στη χώρα μας η ηλεκτρονική συνταγογράφηση, όπου πρόκειται για σύνταξη και καταχώρηση ιατρικών συνταγών με χρήση υπολογιστικών συστημάτων και συγκεκριμένου ιατρικού λογισμικού. [8]

2.2 Δυναμική ιστοσελίδα

Δυναμική ιστοσελίδα λέγεται η ιστοσελίδα η οποία έχει μενού διαχείρισης, όπου μπορούμε να συνδεθούμε και να αλλάξουμε το περιεχόμενο της. Μια δυναμική ιστοσελίδα ή αλλιώς ένα δυναμικό Website έχει πολλές περισσότερες δυνατότητες από μια στατική ιστοσελίδα, διότι δεν είναι απλά ένα ηλεκτρονικό έγγραφο κώδικα προγραμματισμού αλλά μια ολοκληρωμένη διαδικτυακή εφαρμογή. Το δυναμικό Website χρησιμοποιεί ένα συνδυασμό από βάσεις δεδομένων στην περίπτωση μας την MySQL , γλώσσες προγραμματισμού για server σαν την ASP και την PHP μαζί με πρότυπα ιστοσελίδων. Το σημαντικό είναι ότι το περιεχόμενο των ιστοσελίδων του website αποθηκεύεται σε μια βάση δεδομένων. Το μέγεθος των δεδομένων όπως για παράδειγμα οι περιεχόμενες ιστοσελίδες που μπορούν να αποθηκευθούν σε αυτές τις βάσεις δεδομένων είναι θεωρητικά απεριόριστο, βλέπουμε δηλαδή ότι ο όγκος της ιστοσελίδας μεταβάλλεται άρα είναι “δυναμικός”.

Όταν ένας επισκέπτης καλεί μια ιστοσελίδα, το περιεχόμενο της δημιουργείται με τον συνδυασμό κειμένου και προεπιλεγμένου προτύπου. Η δυναμική ιστοσελίδα αλληλεπιδρά με τους επισκέπτες της, δίνοντας τους τη δυνατότητα για παράδειγμα να σχολιάσουν, να συμπληρώσουν φόρμες και να λαμβάνουν ενημερωτικά emails. Έτσι η αλληλεπίδραση αυτή που υφίσταται μεταξύ της ιστοσελίδας και

των επισκεπτών τους εντυπωσιάζει και είναι πιο εύκολο πλέον να επισκεφτούν ξανά την ιστοσελίδα. Τέλος μια σημαντική ακόμη ιδιότητα της δυναμικής ιστοσελίδας είναι η δυνατότητα που δίνεται στον δημιουργό να χρησιμοποιήσει σύγχρονες τεχνικές marketing, όπως για παράδειγμα ένα newsletter δηλαδή διαφημιστικά νέα μέσω email. Μια κατηγοριοποίηση που θα μπορούσαμε να κάνουμε στα δυναμικά websites ανάλογα με τη λειτουργικότητα τους είναι η παρακάτω :

- 1) Συστήματα διαχείρισης περιεχομένου γνωστά και ως συστήματα CMS.
- 2) Ηλεκτρονικά καταστήματα.
- 3) Forums
- 4) Blogs

Βέβαια υπάρχουν περισσότερες κατηγορίες αλλά οι τέσσερις παραπάνω που ανέφερα αποτελούν την πλειοψηφία των διαθέσιμων ιστοσελίδων.

Η διαφορά των δυναμικών ιστοσελίδων από τις στατικές είναι κυρίως το γεγονός ότι είναι αρκετά πιο δύσκολη η κατασκευή τους και ανάλογα με τις δυνατότητες που προσφέρει προς τους χρήστες αυξάνεται και ο βαθμός δυσκολίας τους. Μια στατική ιστοσελίδα η οποία περιέχει μοναδικό περιεχόμενο, για να μπορέσει να υποστεί οποιαδήποτε επεξεργασία απαιτεί χειροκίνητη παρέμβαση από το δημιουργό της. Τέλος μια ακόμη σημαντική διαφορά είναι ότι οι δυναμικές ιστοσελίδες έχουν καλύτερη απόδοση και είναι αρκετά πιο εντυπωσιακές στην εμφάνιση τους από ότι οι στατικές.

2.3 Διαδίκτυο

Το διαδίκτυο (Internet) είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν μια ομάδα πρωτοκόλλων, η οποία είναι γνωστή ως "TCP/IP" και εξυπηρετεί εκατομμύρια χρήστες σε όλο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας και ανταλλάσσουν μηνύματα με τη χρήση διαφόρων πρωτοκόλλων. Το κοινό αυτό δίκτυο καλείται διαδίκτυο. Αποτελεί ουσιαστικά ένα μέσο με το οποίο διακινούνται καθημερινά τεράστιες ποσότητες πληροφοριών. Η συνεχόμενη ανάπτυξη του με αλματώδεις ρυθμούς τα τελευταία χρόνια το έχουν καθιερώσει ως ένα από τα πρωτεύοντα μέσα επικοινωνίας. Το Internet αποτελεί μια τεράστια "δεξαμενή" πληροφοριών που διαρκώς εμπλουτίζεται και ενημερώνεται με νέα στοιχεία. Οι εκατομμύρια χρήστες σε όλο τον κόσμο το χρησιμοποιούν για διάφορους λόγους, όπως ψυχαγωγικούς, ενημερωτικούς, επαγγελματικούς και μορφωτικούς. Ο κάθε χρήστης που χρησιμοποιεί το Internet μπορεί να επικοινωνήσει με οποιονδήποτε άλλο διασυνδεδεμένο χρήστη, με το γνωστό σε όλους email αλλά και να ενημερωθεί για πληθώρα θεμάτων που τον αφορά.

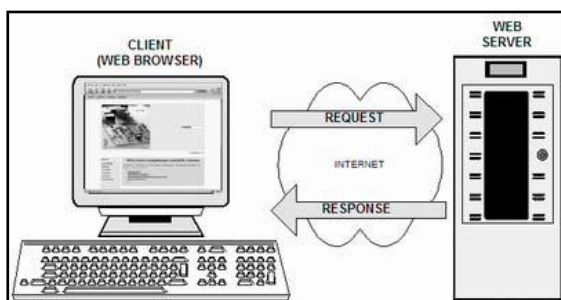
Όταν μιλάμε για το Internet ουσιαστικά αναφερόμαστε στην δημοφιλέστερη μορφή του το WWW (World Wide Web). Οι πληροφορίες στο Web παρουσιάζονται με τη μορφή σελίδων, οι οποίες ομαδοποιούνται στα γνωστά στους περισσότερους Web Sites. Κάθε Site έχει τη δική του μοναδική διεύθυνση, μέσω της οποίας μπορεί ο κάθε χρήστης από οποιοδήποτε σημείο του κόσμου βρίσκεται να συνδεθεί και να πλοηγηθεί στο περιεχόμενο του. Όταν λοιπόν κάποια εταιρεία, ένας οργανισμός ή ένας ιδιώτης θέλουν να παρουσιάσουν κάποιες πληροφορίες στο Web, τότε θα πρέπει να δημιουργήσουν ένα Web Site και να δομήσουν τις πληροφορίες αυτές σε μορφή σελίδων Web, γνωστές και ως ιστοσελίδες και έπειτα να τις εκδώσουν στο Internet. Επίσης το Web όσο περνάει ο καιρός γίνεται όλο και πιο δημοφιλές και για αυτό το λόγο χρησιμοποιείται ως μέσο διαφήμισης από διάφορες εταιρείες και επιχειρήσεις.

Για να μπορέσει μέχρι σήμερα κάποιος να κατασκευάσει ένα Web Site έστω και απλό θα πρέπει να έχει κάποιες γνώσεις γύρω από Web programming (html, java, javascript), ή έστω να χειρίζεται κάποια

προγράμματα κατασκευής ιστοσελίδων (joomla, frontpage, CoffeCup) όπου αρκετά από αυτά κυκλοφορούν δωρεάν στο Web.

2.4 Η αρχιτεκτονική πελάτη –εξυπηρετητή (client-server)

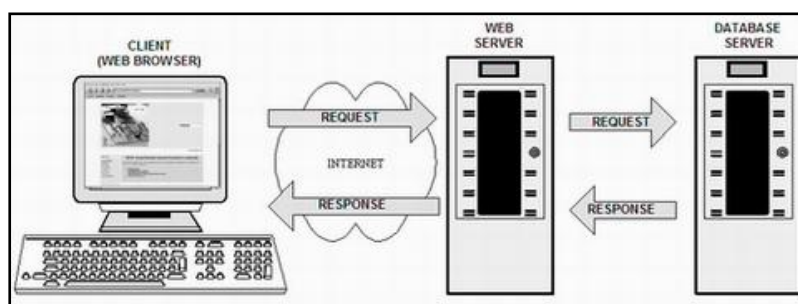
Κατά την περιήγηση ενός χρήστη στο Internet το πρόγραμμα περιήγησης (web browser) που χρησιμοποιεί, αιτείται κάποιες πληροφορίες από ένα εξυπηρετητή δικτύου (web server) και ο server αποκρίνεται με τις ζητούμενες πληροφορίες, οι οποίες προβάλλονται κατάλληλα στον πελάτη (client) μέσω του web browser. Το μοντέλο που περιγράψαμε παραπάνω αποτελεί την κλασική αρχιτεκτονική 2 στρωμάτων ή αλλιώς αρχιτεκτονική πελάτη-εξυπηρετητή (client-server) που χρησιμοποιείται συχνά στο διαδίκτυο.



Εικόνα 2.1 : Η αρχιτεκτονική πελάτη-εξυπηρετητή

2.5 Η αρχιτεκτονική 3 στρωμάτων (3 Tier-Architecture)

Το μοντέλο πελάτη-εξυπηρετητή επικρατούσε στις δικτυακές εφαρμογές όσο οι ιστοσελίδες ήταν κυρίως στατικές και δεν υπήρχαν ιδιαίτερες απαιτήσεις για την εξυπηρέτηση πολλών πελατών ταυτόχρονα. Με την έκρηξη όμως δημοτικότητας του Internet οι περισσότερες εφαρμογές που χρησιμοποιούν πλέον οι χρήστες είναι πολύπλοκες, με δυναμικό κυρίως περιεχόμενο και πολύπλοκες λειτουργίες σε δεδομένα, οι οποίες είναι προτιμότερο να διενεργούνται από ένα ξεχωριστό σύστημα διαχείρισης βάσεων δεδομένων. Αυτό οδήγησε στη μετάβαση από την αρχιτεκτονική πελάτη-εξυπηρετητή στην αρχιτεκτονική 3 στρωμάτων. Πρόκειται για μια αρχιτεκτονική στη οποία ο web browser εξακολουθεί να είναι το στρώμα πελάτη το οποίο επικοινωνεί με τον εξυπηρετητή δικτύου. Ο εξυπηρετητής δικτύου γίνεται το μεσαίο στρώμα, το οποίο περιέχει τη δικτυακή εφαρμογή και προστίθεται ένας εξυπηρετητής βάσης δεδομένων (Database server) στο τρίτο στρώμα, ο οποίος αναλαμβάνει την αποτελεσματικότερη αποθήκευση των δεδομένων και των ερωτήσεων της δικτυακής εφαρμογής προς τη βάση.[9]



Εικόνα 2.2 : Η αρχιτεκτονική 3-στρωμάτων

2.6 Το στρώμα πελάτη

Το στρώμα πελάτη σε μια δικτυακή εφαρμογή αποτελεί το πρόγραμμα περιήγησης ιστοσελίδων (**web browser**) που βρίσκεται εγκατεστημένο σε έναν υπολογιστή που έχει πρόσβαση στο διαδίκτυο. Οι αρμοδιότητες του στρώματος αυτού, είναι η αποστολή αιτήσεων για κάποια πληροφορία προς κάποιο εξυπηρετητή δικτύου και η εμφάνιση της πληροφορίας που αυτός επιστρέφει με τη μορφή ιστοσελίδας. Η επεξεργασία της πληροφορίας πραγματοποιείται από το μεσαίο στρώμα σε συνεργασία με το στρώμα δεδομένων. Αυτό πρακτικά σημαίνει ότι κατά την ανάπτυξη της εφαρμογής δεν μας ενδιαφέρει η εγκατάσταση ή η ρύθμιση του στρώματος πελάτη.

Οποιοσδήποτε χρήστης μπορεί να χρησιμοποιήσει την ιστοσελίδα μας και να πάρει τις πληροφορίες που θέλει από τη βάση δεδομένων μας χωρίς να μας ενδιαφέρει το λειτουργικό του σύστημα. Αυτός είναι και ένας λόγος που κάνει την ανάπτυξη δικτυακών εφαρμογών με το μοντέλο τριών στρωμάτων και τους web browsers ως στρώμα πελάτη αρκετά δημοφιλείς.

3

Μεσαίο στρώμα (Middle Tier)

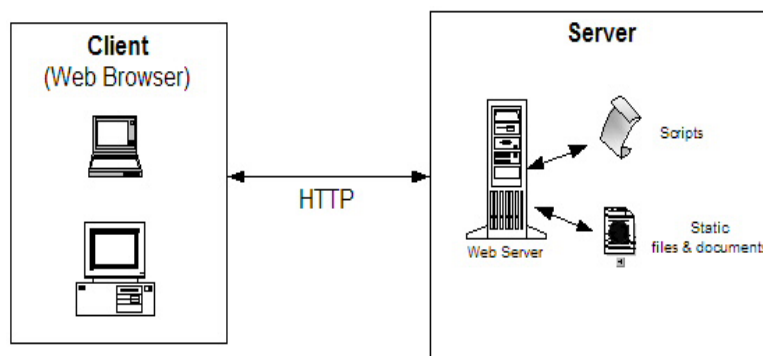
3.1 Γενικά - Ρόλος

Το μεσαίο στρώμα στην αρχιτεκτονική τριών στρωμάτων που περιγράφουμε, αποτελούν οι εξυπηρετητές δικτύου. Στην εφαρμογή μας χρησιμοποιείται ο εξυπηρετητής δικτύου Apache. Το μεσαίο στρώμα έχει κεντρικό ρόλο σε μια δικτυακή εφαρμογή που ακολουθεί την αρχιτεκτονική τριών στρωμάτων. Είναι το στρώμα που επιτρέπει στα άλλα δυο στρώματα να επικοινωνήσουν μεταξύ τους. Εξυπηρετεί τις συνδέσεις των χρηστών και παράγει το περιεχόμενο των δεδομένων που παρουσιάζονται στο στρώμα πελάτη. Στο στρώμα αυτό ανήκει και το προγραμματιστικό μέρος της εφαρμογή μας.

3.2 Εξυπηρετητές Δικτύου

Εξυπηρετητές δικτύου (**Web Server**) λέγεται το πρόγραμμα που εξυπηρετεί τις δικτυακές αιτήσεις των πελατών (**clients**), την προβολή μιας ιστοσελίδας, την εκτέλεση ενός προγράμματος (script) κ.α. Το μηχάνημα στο οποίο εγκαθίσταται το πρόγραμμα αυτό και εννοείται να είναι συνδεδεμένο με το Internet, αποκαλείται επίσης Web Server.

Οι εξυπηρετητές δικτύου αναπτύσσονται σύμφωνα με το μοντέλο προγραμματισμού client-server και χρησιμοποιούν το πρωτόκολλο μεταφοράς δεδομένων HTTP, προκειμένου να αποστείλουν στους χρήστες web τα αρχεία και τις ιστοσελίδες τα οποία αιτούνται. Οι αιτήσεις των χρηστών στέλνονται και αυτές με τη βοήθεια του πρωτοκόλλου HTTP όταν ο χρήστης αιτείται κάποια διεύθυνση URL στο web browser που χρησιμοποιεί.



Εικόνα 3.1 : Η αρχιτεκτονική του client-server στο διαδίκτυο

3.3 Πρωτόκολλο HTTP

Το πρωτόκολλο HTTP ή πρωτόκολλο μεταφοράς υπερκειμένου, είναι το στάνταρ πρωτόκολλο που χρησιμοποιείται για την επικοινωνία μεταξύ του web browser και του web server, δηλαδή των δύο

πρώτων στρωμάτων του μοντέλου 3-Tier των δικτυακών εφαρμογών. Η λειτουργία του πρωτοκόλλου HTTP είναι η εξής : ένας πελάτης (web browser) αιτείται κάποιο πόρο (μια σελίδα html, ένα αρχείο) από ένα εξυπηρετητή δικτύου και αυτός απαντάει στέλνοντας τον ζητούμενο πόρο, ή ένα μήνυμα λάθους αν η αίτηση δεν μπορεί να πραγματοποιηθεί. Για κάθε αίτηση (request) υπάρχει μια απάντηση (response).

Μια αίτηση HTTP αποτελείται από το όνομα της μεθόδου που χρησιμοποιείται, την περιγραφή του ζητούμενου πόρου, την έκδοση του πρωτοκόλλου και ένα προαιρετικό αριθμό παραμέτρων που δίνουν πληροφορίες για το πώς πρέπει αυτός να επιστραφεί.[10]

3.4 Γενικά για τον Apache και τις τρέχουσες εκδόσεις του

Ο Apache θεωρείται ως ένας από τους πιο δημοφιλείς Web Server. Αρχικά ο λόγος για τον οποίο δημιουργήθηκε ήταν για τους κεντρικούς υπολογιστές Unix. Αργότερα χρησιμοποιήθηκε για τα Windows και άλλα λειτουργικά συστήματα. Το Apache είναι ελεύθερο λογισμικό όπου διανέμεται από το Apache Software Foundation, το οποίο προωθεί διάφορες ανοικτές πηγές τεχνολογίας Ιστού, γνωστές ως Web technologies. Η διεύθυνση <http://httpd.apache.org> περιέχει ανακοινώσεις για τις διάφορες εκδόσεις του Apache όπως είναι η Apache 1.3.x, Apache 2.0.x και Apache 2.2.x. Το Apache 1.3.x είναι η παλαιότερη από τις εκδόσεις. Το Apache Software Foundation διατηρεί και τις τρεις παραπάνω εκδόσεις, τα χαρακτηριστικά όμως του Apache 2.2.x περιλαμβάνουν περισσότερη υποστήριξη για caching, φιλτράρισμα και για άλλες λειτουργίες του συστήματος. Το Apache Software Foundation χρησιμοποιεί αριθμούς δευτερεύουσας έκδοσης για να χαρακτηρίσει τις ενημερώσεις που περιέχουν βελτιώσεις ή διορθώσεις σφαλμάτων σχετιζόμενες με την ασφάλεια. Όταν προστίθενται βελτιώσεις ή διορθώσεις στον κώδικα που αφορούν τις δευτερεύουσες εκδόσεις, τότε το Apache Software Foundation κυκλοφορεί μια νέα έκδοση του λογισμικού με νέο αριθμό δευτερεύουσας έκδοσης. [1]

3.4.1 Επιλογή της μεθόδου εγκατάστασης

Για να πραγματοποιηθεί η εγκατάσταση υπάρχουν αρκετές επιλογές στη διάθεση του κάθε χρήστη. Το Apache είναι “ανοιχτό λογισμικό”, πράγμα το οποίο σημαίνει ότι οποιοσδήποτε χρήστης μπορεί να έχει πρόσβαση σε όλο τον πηγαίο κώδικα του. Αυτό πρακτικά μας επιτρέπει να δημιουργήσουμε μια εξειδικευμένη έκδοση του server σύμφωνα με το τι θέλουμε να κάνουμε. Επίσης υπάρχουν διανομές του Apache με προ-μεταγλωττισμένα δυαδικά αρχεία για τις περισσότερες πλατφόρμες UNIX. Η δημιουργία εκτελέσιμων αρχείων από τον πηγαίο κώδικα του Apache server μας παρέχει ευελιξία, διότι μπορούμε να δημιουργήσουμε την δική μας έκδοση server. Επίσης έχουμε τη δυνατότητα να αναβαθμιστούμε στις πιο πρόσφατες εκδόσεις και να ενημερώσουμε τις ρυθμίσεις για την ασφάλεια.

3.4.2 Εγκατάσταση του Apache στα Windows

Προτού ξεκινήσουμε την εγκατάσταση του πρέπει να είμαστε σίγουροι ότι στο σύστημα μας δεν τρέχει οιοσδήποτε άλλος Web Server. Εάν ήδη τρέχουμε κάποιον άλλο καλό θα είναι να τον καταργήσουμε από το σύστημα μας ή να τον απενεργοποιήσουμε. Ο μόνος τρόπος για να μπορέσουμε να τρέξουμε πολλαπλούς Web servers, είναι να τους τρέχουμε σε διαφορετικούς συνδυασμούς διευθύνσεων και θυρών. Το Apache 2.2.11 τρέχει στις περισσότερες πλατφόρμες Windows και παρέχει περισσότερα χαρακτηριστικά και μεγαλύτερη σταθερότητα έναντι των εκδόσεων 2.0 και 1.3. Για να ξεκινήσουμε τη διαδικασία εγκατάστασης του Apache στον υπολογιστή μας επισκεπτόμαστε τη σελίδα: <http://httpd.apache.org/download.cgi>. Επίσης καλό θα ήταν ο χρήστης να επισκεφτεί την σελίδα <http://www.apache.org/dist/httpd/binaries/win32/README.html>

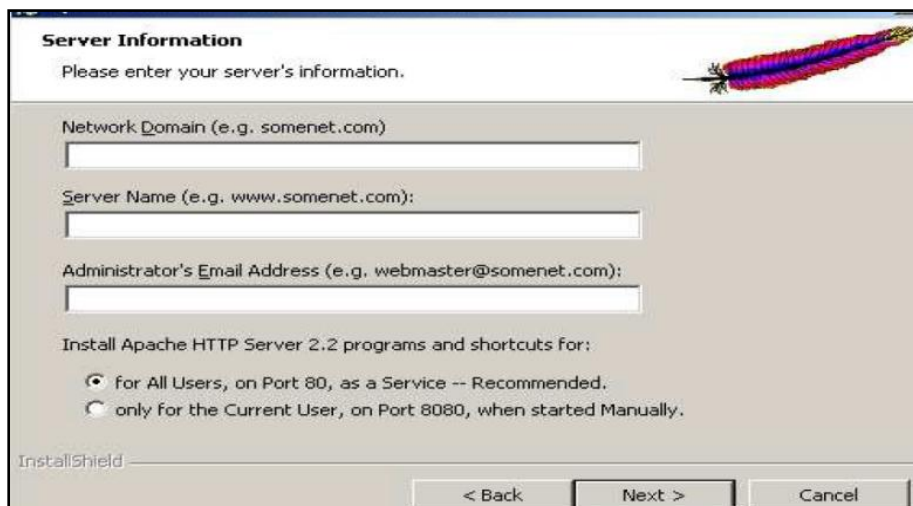
η οποία παρέχει αρκετές χρηστικές πληροφορίες για όλες τις εκδόσεις των Windows που χρησιμοποιούνται σήμερα .Η ιστοσελίδα αυτή διατηρείται από το Apache Software Foundation ως βοήθημα για τους χρήστες που τρέχουν μια έκδοση του Apache Server.Εφόσον λοιπόν ο χρήστης είναι έτοιμος, θα σας παρουσιάσω παρακάτω αναλυτικά τα βήματα της διαδικασίας εγκατάστασης του Apache 2. στα Windows.

1)Αναζητούμε τη σύνδεση που φέρει τον τίτλο Win32Binary(MSI Installer).Κάνουμε διπλό κλικ πάνω του για να ξεκινήσουμε τον οδηγό εγκατάστασης (Installation Wizard).Διαβάζουμε τις σχετικές πληροφορίες που παρατίθενται και στην πορεία η εγκατάσταση μας ενημερώνει για την προστασία του λογισμικού, τους όρους χρήσης και διάφορες λεπτομέρειες σχετικά με την τρέχουσα έκδοση του Apache HTTP Server.Μας ζητείται να αποδεχτούμε τους όρους συμφωνίας άδειας χρήσης του Apache , όπου ουσιαστικά μας αναφέρει ότι μπορούμε να χειριστούμε το λογισμικό όπως θέλουμε, να κάνουμε διάφορες προσαρμογές που μας εξυπηρετούν, αλλά απαγορεύεται να ισχυριστούμε ότι είναι δικό μας κάποιο κομμάτι του.

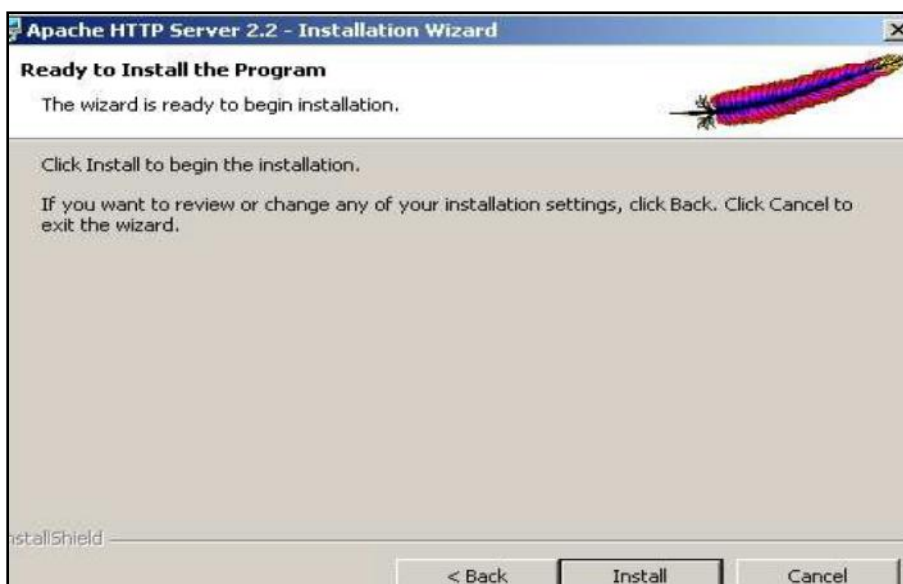
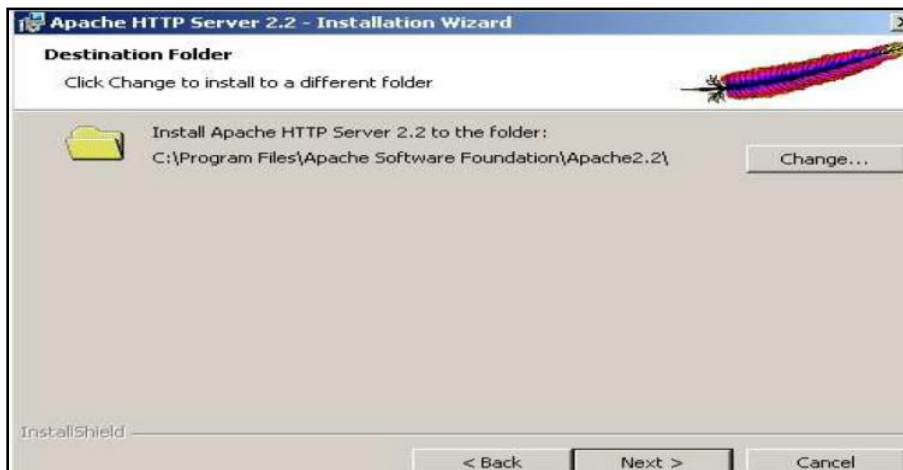




2) Το επόμενο παράθυρο εγκατάστασης που εμφανίζεται, μας ζητάει να ορίσουμε κάποιες βασικές πληροφορίες σχετικά με το σύστημα μας, όπως είναι η πλήρης διεύθυνση (Network Domain) του δικτύου για τον server καθώς και η ηλεκτρονική διεύθυνση (email) του διαχειριστή (administrator). Το όνομα του server (Server Name) θα είναι αυτό για το οποίο θα έχουν πρόσβαση οι χρήστες (Clients) στον server. Το e-mail του διαχειριστή μας ζητείται να το δηλώσουμε, διότι σε περίπτωση που προκύψει κάποιο μήνυμα σφάλματος, να εμφανίζετε στην οθόνη του φυλλομετρητή και έτσι να μπορούν οι επισκέπτες να επικοινωνήσουν μαζί του για να διορθωθεί το οποιοδήποτε πρόβλημα. Στη συνέχεια δηλώνουμε αν θα εγκαταστήσουμε τον Apache σαν υπηρεσία (service), ή αν προτιμάμε να ξεκινάει χειροκίνητα. Στην περίπτωση που επιλέξουμε να τον εγκαταστήσουμε σαν υπηρεσία, ο server θα ξεκινά κάθε φορά που ξεκινάνε τα Windows και έτσι θα μπορεί ο χρήστης να πραγματοποιεί τον έλεγχο του από τα εργαλεία διαχείρισης υπηρεσιών που περιλαμβάνουν τα Windows.



3) Το τρίτο παράθυρο που εμφανίζεται μας ζητάει να επιλέξουμε ποιον τύπο εγκατάστασης θέλουμε, το typical ή τον custom. Η διαφορά τους είναι ότι στην τυπική (typical) εγκατάσταση εγκαθίστανται τα δυαδικά αρχεία του Apache και τα αρχεία πληροφοριών τεκμηρίωσης, αλλά δεν εγκαθίστανται τα header αρχεία, ενώ η εξειδικευμένη (Custom) εγκατάσταση μας αφήνει το ελεύθερο να επιλέξουμε αν θα εγκαταστήσουμε τα header αρχεία, ή τα αρχεία πληροφοριών τεκμηρίωσης. Έπειτα επιλέγουμε τον φάκελο όπου θα γίνει η εγκατάσταση και έτσι αρχίζει η διαδικασία αντιγραφής αρχείων και ρυθμίσεων του Apache. Για να είμαστε σίγουροι ότι όλα έχουν πάει καλά και ότι όντως ο Apache εγκαταστάθηκε σωστά, καλό θα ήταν για να τον δοκιμάσουμε, να ανοίξουμε ένα φυλλομετρητή (Browser) και να πληκτρολογήσουμε τη διεύθυνση: <http://127.0.0.1>. Αν εμφανιστεί το μήνυμα "It works!" τότε όλα έγιναν σωστά και από εδώ και πέρα μπορούμε να φορτώσουμε τις ιστοσελίδες μας στον φάκελο htdocs που βρίσκεται μέσα στον Apache.



Τέλος μόλις ολοκληρωθεί η διαδικασία με επιτυχία, εμφανίζεται και ένα μήνυμα που το επιβεβαιώνει.

3.4.3 Το αρχείο διαμόρφωσης του Apache

Το Apache διατηρεί όλες τις πληροφορίες των ρυθμίσεων του σε αρχεία απλού κειμένου (text files), όπου το κυριότερο από αυτά είναι το αρχείο **httpd.conf**. Το αρχείο αυτό περιλαμβάνει ντιρεκτίβες (directives) και υποδοχείς (containers). Οι ντιρεκτίβες καθορίζουν βασικές λειτουργίες του Apache, όπως για παράδειγμα εξουσιοδοτήσεις (authorization) και διάφορες παραμέτρους για την απόδοση και τη λειτουργία του δικτύου. Οι περιέκτες από τη μεριά τους καθορίζουν το περιβάλλον στο οποίο αναφέρονται αυτές οι ρυθμίσεις, όπως για παράδειγμα είναι οι ρυθμίσεις εξουσιοδοτήσεων, που αναφέρονται σαν ένας ολόκληρος φάκελος ή σαν ένα μεμονωμένο αρχείο κειμένου.

4

MySQL

4.1 Εισαγωγή στη MySQL

Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων ανοιχτού κώδικα (Open Source) και διανέμεται δωρεάν από την εταιρεία MySQL AB. Αυτό σημαίνει πως πρόκειται για ένα λογισμικό το οποίο διατίθεται ελεύθερα και μπορεί ο οποιοσδήποτε χρήστης να το χρησιμοποιήσει δωρεάν, και να παρέμβει σε αυτό για να το τροποποιήσει. Μια σχεσιακή βάση δεδομένων (Relational Database) παριστάνει και αποθηκεύει τα δεδομένα σε πίνακες. Ο MySQL server δημιουργήθηκε με σκοπό να διαχειρίζεται μεγάλες βάσεις δεδομένων και λειτουργεί κυρίως σε συστήματα πελάτη – εξυπηρετητή.

4.2 Εγκατάσταση και Διαμόρφωση της Mysql

Οι οδηγίες εγκατάστασης που θα παρουσιάσω παρακάτω αναφέρονται στην έκδοση της MySQL 5.1. Η MySQL QB είναι η εταιρεία η οποία παράγει και διανέμει την MySQL και στην περίπτωση που προκύψουν ενημερώσεις στον κώδικα οι οποίες πιθανόν να σχετίζονται με την ασφάλεια ή με πιθανά σφάλματα, τότε η εταιρεία κυκλοφορεί μια καινούργια έκδοση λογισμικού η οποία περιέχει νέο αριθμό δευτερεύουσας έκδοσης. Όπως βλέπουμε οι δευτερεύουσες εκδόσεις δεν ακολουθούν κάποια προκαθορισμένη περίοδο, αλλά εκδίδονται όταν υπάρχει ανάγκη. Είναι απαραίτητο ο κάθε χρήστης να είναι ενήμερος με το λογισμικό που χρησιμοποιεί, γιατί μπορεί να υπάρξουν τροποποιήσεις ή να αλλάξει ο δευτερεύων αριθμός έκδοσης σε σύντομο χρονικό διάστημα αφότου το έχει εγκαταστήσει. Για αυτό καλό θα ήταν να επισκέπτεται τη σελίδα <http://dev.mysql.com/doc/relnotes/mysql/en/5.0/en/> για να ενημερώνεται για τυχόν αλλαγές που έχουν γίνει και οποίες μπορεί να αφορούν τη διαμόρφωση του λογισμικού ή τη διαδικασία εγκατάστασης. Συνήθως η διαδικασία εγκατάστασης του λογισμικού δεν αλλάζει, έτσι σε περίπτωση που προκύψει καινούργια δευτερεύουσα έκδοση, ο χρήστης μπορεί να ενημερώνει το λογισμικό χωρίς να συναντήσει κάποια ιδιαίτερη δυσκολία. Η MYSQL AB είναι η εταιρεία η οποία διανέμει την MySQL στην ιστοσελίδα <http://www.mysql.com>, όπου εκεί μπορεί ο κάθε χρήστης να βρει κώδικα για όλες τις πλατφόρμες και επίσης RPM και αρχεία πηγαίου κώδικα για πλατφόρμες Linux/Unix. [1]

4.3 Εγκατάσταση της MySQL σε συστήματα Linux/Unix

Η εγκατάσταση της MySQL σε πλατφόρμα Linux/Unix γίνεται με δυο τρόπους, είτε χρησιμοποιώντας RPM είτε εγκαθιστώντας τα εκτελέσιμα αρχεία. Στην περίπτωση που επιλεγθεί εγκατάσταση από RPM τότε είναι απαραίτητα τα δυο παρακάτω αρχεία:

α) MYSQL-server-type-VERSIONNUMBER.PLATFORM.rpm- Ο MySQL server

β)MYSQL-client-type-VERSIONNUMBER.PLATFORM.rpm- Οι βιβλιοθήκες πελάτη της MySQL

Για να δούμε αν πραγματοποιείται εγκατάσταση από RPM αρκεί να πληκτρολογήσουμε την εξής εντολή:
rpm -I MySQL- server- VERSION.I1386.RPM MySQL- CLIENT –version.i386.rpm

Αν προτιμήσουμε το δεύτερο τρόπο, να εγκαταστήσουμε δηλαδή την MySQL εγκαθιστώντας τα εκτελέσιμα αρχεία, τότε είναι απαραίτητο να έχουμε τα βοηθήματα gunzip και tar για να προβούμε σε αποσυμπίεση των δυαδικών αρχείων. Η πρώτη σειρά εντολών προσθέτει μια ομάδα και ένα χρήστη και αποσυμπιέζει τα αρχεία ως εξής:

```
# groupadd mysql
# useradd -g mysql mysql
# cd /usr/local
# gunzip < /path/to/mysql-VERSION-PLATFORM.tar.gz
```

Στη συνέχεια αναφέρεται πως πρέπει να δημιουργηθεί μια σύνδεση με ένα μικρότερο όνομα:

```
# ln -s mysql-VERSION-PLATFORM mysql
# cd mysql
```

Έπειτα πρέπει να αποσυμπίεστούν τα αρχεία README ΚΑΙ INSTALL και αυτά με τη σειρά τους οδηγούν το χρήστη στην ολοκλήρωση της εγκατάστασης. Ουσιαστικά θα χρησιμοποιηθεί η παρακάτω σειρά εντολών:

```
# scripts/msql_install_db - -user=mysql
# chown -R root
# chown -R mysql mysql_data
# chgrp -R mysql
# bin/mysqld_safe - -user=mysql
```

Μετά από όλα αυτά ο server είναι έτοιμος να εκκινήσει. Αν για οποιοδήποτε λόγο παρουσιαστεί πρόβλημα συνίσταται ο χρήστης να ανατρέξει στην ενότητα “Αντιμετώπιση Προβλημάτων κατά την εγκατάσταση”.

4.4 Εγκατάσταση της MySQL στα Windows

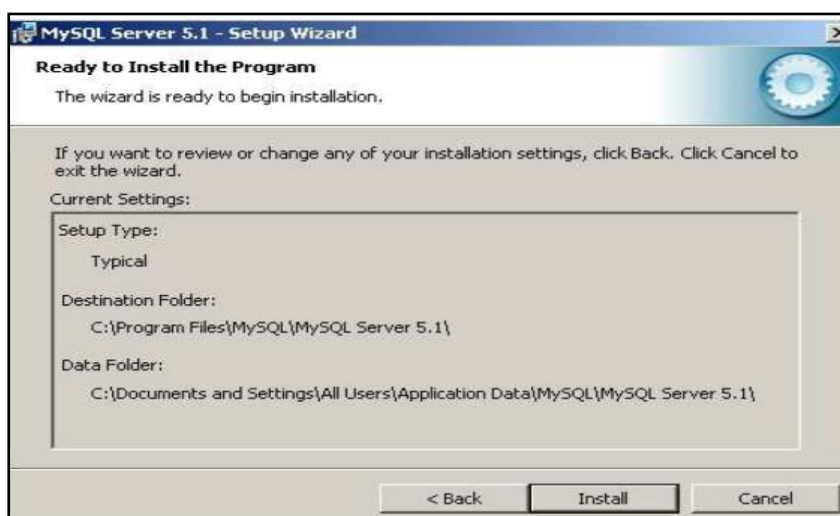
Η εγκατάσταση της MySql στα windows είναι μια απλή διαδικασία. Η εταιρεία MySql AB παράγει και διανέμει το πρόγραμμα εγκατάστασης σε ένα συμπιεσμένο αρχείο zip. Το αρχείο αυτό το μεταφέρουμε στον υπολογιστή μας, το αποσυμπιέζουμε και εκτελούμε το πρόγραμμα εγκατάστασης setup.exe.

Παρακάτω γίνεται αναλυτική παρουσίαση εγκατάστασης της MySql 5.1 στα Windows.

- 1) Κάνουμε διπλό κλικ πάνω στο αρχείο setup.exe. για να ξεκινήσει η εγκατάσταση. Ο οδηγός αναφέρει κάποιες γενικές πληροφορίες για το πρόγραμμα και ενημερώνει το χρήστη πως για να προχωρήσει η εγκατάσταση του προγράμματος, πρέπει να κλείσει οποιαδήποτε εφαρμογή χρησιμοποιεί εκείνη την ώρα.



- 2) Το επόμενο παράθυρο ενημερώνει το χρήστη για τις τρεις μεθόδους εγκατάστασης- Typical, Complete και Custom. Συνιστάται ο χρήστης να επιλέξει την τυπική μέθοδο εγκατάστασης (Typical), διότι είναι κατάλληλη για τους περισσότερους επειδή περιλαμβάνει τον server, τον client και διάφορα χρηστικά εργαλεία για την MySQL. Στη συνέχεια γίνεται κλικ στο Next για να συνεχίσει η εγκατάσταση.



- 3) Αφότου ο χρήστης πατήσει το Next, ξεκινάει η διαδικασία εγκατάστασης και μόλις ολοκληρωθεί εμφανίζεται στην οθόνη μήνυμα που ενημερώνει το χρήστη ότι η διαδικασία έχει ολοκληρωθεί.
- 4) Τέλος ο χρήστης κάνει κλικ στην επιλογή Finish και το πρόγραμμα από εδώ και πέρα είναι έτοιμο για χρήση.

Επιπλέον η MySQL παρέχει την εφαρμογή winmysqladmin.exe η οποία όταν προστελαστεί πρώτη φορά από τον χρήστη, του ζητάει να ορίσει ένα όνομα χρήστη (username) και ένα κωδικό πρόσβασης (password). Η εφαρμογή αυτή είναι αρκετά χρήσιμη, διότι δίνει σημαντικές πληροφορίες για το περιβάλλον, όπως επίσης για τη διεύθυνση IP και το όνομα του υπολογιστή.

4.5 Πιθανά προβλήματα κατά την εγκατάσταση

Υπάρχει περίπτωση κατά τη διάρκεια της εγκατάστασης να παρουσιαστεί κάποιο πρόβλημα. Τότε συνίσταται ο χρήστης να ανατρέξει στο βοηθητικό εγχειρίδιο της MySQL το οποίο βρίσκεται στη διεύθυνση <http://dev.mysql.com/doc/refman/5.0/en/problems.html> και δίνει λύσεις σε συγκεκριμένα σφάλματα που μπορεί να παρουσιαστούν.

α) Αν κατά τη σύνδεση στην MySQL εμφανιστεί στον χρήστη το μήνυμα Access denied, τότε είναι πιθανό να έχει πληκτρολογήσει λανθασμένα στοιχεία στο όνομα χρήστη (user name) και στο κωδικό πρόσβασης (password).

β) Αν εμφανιστεί το μήνυμα Can't connect to server τότε πρέπει να βεβαιωθεί ο χρήστης ότι η MySQL τρέχει στο σύστημα.

Αν παρ'όλα αυτά ο χρήστης συνεχίζει να έχει πρόβλημα και να μην μπορεί συνδεθεί στο πρόγραμμα τότε μπορεί να στείλει e-mail στη διεύθυνση <http://lists.mysql.com/> .

4.6 Σημαντικές οδηγίες ασφαλείας

Είναι πολύ σημαντικό ο χρήστης να κατανοήσει τις βασικές οδηγίες ασφαλείας γιατί πολλές φορές μπορεί να είναι εκτεθειμένος σε κινδύνους χωρίς να το γνωρίζει. Αν ο χρήστης συνδέεται στην MySQL μέσω ενός παρόχου υπηρεσιών Internet (Internet Explorer, Mozilla Firefox κτλ.), τότε υπάρχουν σημαντικά θέματα ασφαλείας του server, τα οποία εφόσον δεν είναι χρήστης root, δεν μπορεί να τα τροποποιήσει, να παρέμβει δηλαδή σε αυτά για να είναι πιο ασφαλής.

4.7 Σύστημα δικαιωμάτων της MySQL

Κάθε φορά που ο χρήστης συνδέεται η MySQL ελέγχει τα εξής.

- α) Από πού γίνεται η πρόσβαση, δηλαδή από ποιον υπολογιστή host
- β) Την ταυτότητα του χρήστη, το όνομα του δηλαδή και τον κωδικό πρόσβασης
- γ) Τι μπορεί να κάνει ο χρήστης, ποια είναι δηλαδή τα δικαιώματα του.

Οι πληροφορίες αυτές αποθηκεύονται στη βάση δεδομένων mysql, η οποία δημιουργείται αυτόματα κατά την εγκατάσταση της MySQL. Στη συγκεκριμένη βάση δεδομένων συναντώνται αρκετοί πίνακες, μερικούς από τους οποίους θα παρουσιάσω παρακάτω.

α) users : Στον πίνακα αυτό ορίζονται τα δικαιώματα που έχει ο χρήστης για τις εντολές.

β) db: Ο πίνακας αυτός ορίζει τα δικαιώματα για όλες τις βάσεις δεδομένων στον server.

γ) columns_priv : Ο πίνακας αυτός ορίζει τα δικαιώματα που έχει ο χρήστης για συγκεκριμένα πεδία μέσα σε ένα πίνακα.

δ) tables_priv : Ο πίνακας αυτός ορίζει τα δικαιώματα που έχει ο χρήστης για συγκεκριμένους πίνακες μέσα σε μια βάση δεδομένων.

ε) host: Ο πίνακας αυτός ορίζει ποιοι υπολογιστές είναι αποδεκτοί, ώστε να συνδέονται σε μια συγκεκριμένη βάση δεδομένων.

4.8 Διαχείριση των χρηστών

Όταν προσπελάζουμε την MySQL τότε έχουμε στη διάθεση μας μόνο ένα χρήστη και μια βάση δεδομένων. Ο χρήστης αυτός μπορεί να έχει πρόσβαση σε όλους τους πίνακες της βάσης δεδομένων και έχει τη δυνατότητα να εκτελεί οποιαδήποτε εντολή θέλει. Υπάρχει όμως περίπτωση κάποιος να είναι διαχειριστής του δικού του server και έτσι να έχει την ευχέρεια να προσθέσει επιπλέον βάσεις δεδομένων και χρήστες. Η πιο απλή μέθοδος που υπάρχει για να μπορέσει κάποιος να προσθέσει καινούργιο χρήστη είναι με την εντολή GRANT, όπου συνδεδεμένος στην MySQL σαν χρήστης, πληκτρολογεί αυτή την εντολή και έτσι καθορίζει ένα νέο χρήστη. Ένας δεύτερος τρόπος αλλά πιο δύσκολος είναι με την εντολή INSERT, όπου πρέπει ο χρήστης να την δώσει σε όλους τους πίνακες που σχετίζονται με τη βάση δεδομένων mysql, πράγμα όμως που προϋποθέτει να γνωρίζει όλα τα πεδία πινάκων. Επίσης είναι αρκετά χρήσιμο ο χρήστης να ανατρέξει στην ιστοσελίδα <http://dev.mysql.com/doc/refman/5.0/en/grant.html>, όπου εκεί θα βρεί μια πλήρη λίστα για τα δικαιώματα και τι ακριβώς μπορεί να κάνει με την εντολή GRANT. Ορισμένα από τα πιο συνηθισμένα δικαιώματα είναι τα παρακάτω :

CREATE: Με την εντολή CREATE ο χρήστης μπορεί να δημιουργεί βάσεις δεδομένων και πίνακες.

DELETE: Με την εντολή DELETE ο χρήστης μπορεί να διαγράφει εγγραφές από πίνακες.

DROP : Με την εντολή DROP ο χρήστης μπορεί να διαγράφει πίνακες και βάσεις δεδομένων.

INSERT : Με την εντολή INSERT ο χρήστης μπορεί να προσθέτει εγγραφές σε πίνακες.

SELECT : Με την εντολή SELECT ο χρήστης μπορεί να επιλέγει εγγραφές από πίνακες.

4.9 Τύποι δεδομένων της MySQL

Είναι πολύ σημαντικό ο χρήστης να χρησιμοποιεί τον σωστό τύπο δεδομένων και να ορίζει το μέγεθος που πρέπει για κάθε πεδίο του πίνακα, έτσι ώστε να έχει την βέλτιστη απόδοση η βάση δεδομένων του. Η MySQL χρησιμοποιεί τρεις διαφορετικούς τύπους δεδομένων. Τους αριθμητικούς, τους ημερομηνίας ώρας και τους αλφαριθμητικούς. Παρακάτω θα παρουσιάσω τους πιο κοινούς τύπους δεδομένων για την κάθε κατηγορία.

4.9.1 Αριθμητικοί τύποι δεδομένων

INT: Ακέραιος κανονικού μεγέθους, όπου το μέγιστο μήκος του είναι 11 ψηφία.

TINYINT: Πολύ μικρός ακέραιος όπου το μέγιστο μήκος του είναι 4 ψηφία.

SMALLINT: Μικρός ακέραιος όπου το μέγιστο μήκος του είναι 5 ψηφία.

MEDIUMINT: Μεσαίου μεγέθους ακέραιος όπου το μέγιστο μήκος του είναι 9 ψηφία.

BIGINT: Μεγάλος ακέραιος όπου το μέγιστο μήκος του είναι 11 ψηφία.

FLOAT(M,D) : Αριθμός κινητής υποδιαστολής όπου ο χρήστης ορίζει με το M το μέγεθος εμφάνισης ,το συνολικό αριθμό δηλαδή των ψηφίων και με το D ορίζει το πλήθος των δεκαδικών. Αν δεν ορίσει τιμές ο χρήστης τότε το M παίρνει την τιμή 10 και το D την τιμή 2.

DOUBLE(M,D): Αριθμός κινητής υποδιαστολής διπλής ακρίβειας όπου αντίστοιχα με τον FLOAT αν δεν ορίσει τιμές ο χρήστης τότε το M παίρνει την τιμή 16 και το D την τιμή 4.

DECIMAL(M,D) : Ένας μη συμπιεσμένος αριθμός κινητής υποδιαστολής ο οποίος δεν μπορεί να είναι μη-προσημασμένος. Στους αριθμούς αυτούς κάθε δεκαδικό ψηφίο αντιστοιχεί σε ένα byte και είναι υποχρεωτικό ο χρήστης να ορίσει τιμή για το μέγεθος εμφάνισης (M) και για το πλήθος των δεκαδικών (D).

4.9.2 Τύποι δεδομένων για τιμές ημερομηνίας /ώρας

Η MySQL υποστηρίζει αρκετούς τύπους δεδομένων για την αποθήκευση τιμών ημερομηνίας/ ώρας και οι τύποι αυτοί είναι αρκετά ευέλικτοι για την πλευρά του χρήστη. Παρακάτω θα αναφέρω τους τύπους δεδομένων που υποστηρίζει η MySQL:

DATE : Ο τύπος αυτός αναφέρεται σε μια τιμή ημερομηνίας με την μορφή ΕΕΕ-ΜΜ-ΗΗ μεταξύ 1000-01-01 και 9999-12-31.

DATETIME : Ο τύπος αυτός αναφέρεται σε ένα συνδυασμό τιμών ημερομηνίας και ώρας με την μορφή ΜΜ-ΗΗ ΩΩ:ΛΛ:ΔΔ μεταξύ 1000-01-01 00:00:00 και 9999-12-31 23:59:59

TIMESTAMP : Μια χρονική ένδειξη ανάμεσα στα μεσάνυχτα της 1^{ης} Ιανουαρίου 1970 και στο έτος 2037. Μπορούν να οριστούν πολλαπλά μήκη στο πεδίο TIMESTAMP.

TIME : Αποθηκεύει την ώρα σε μορφή ΩΩ:ΛΛ:ΔΔ.

YEAR(M) : Αποθηκεύει ένα έτος σε διψήφια ή τετραψήφια μορφή. Το M είναι παράμετρος μεγέθους όπου αν έχει την τιμή 2 το έτος μπορεί να είναι από 1970 έως 2069 ενώ αν έχει την τιμή 4 το έτος μπορεί να είναι από 1901 έως 2155.

4.9.3 Αλφαριθμητικοί τύποι

Η MySQL υποστηρίζει τους παρακάτω πιο συνηθισμένους τύπους για αλφαριθμητικά:

CHAR(M) : Πρόκειται για ένα αλφαριθμητικό με σταθερό μέγεθος μεταξύ 1 και 255 χαρακτήρες. Δεν είναι απαραίτητο ο χρήστης να ορίσει το μήκος διότι ορίζεται ως προεπιλεγμένο μήκος 1.

VARCHAR(M) : Πρόκειται για ένα αλφαριθμητικό μεταβλητού μεγέθους, με μήκος που κυμαίνεται μεταξύ 1 και 255 χαρακτήρες όπου εδώ είναι απαραίτητο ο χρήστης να ορίσει το μήκος.

BLOB ή TEXT : Πρόκειται για ένα πεδίο με μέγιστο μέγεθος 65.535 χαρακτήρες. Ο τύπος BLOB χρησιμοποιείται για την αποθήκευση μεγάλων δεδομένων όπως είναι οι εικόνες. Ο τύπος TEXT χρησιμοποιείται και αυτός για την αποθήκευση μεγάλων δεδομένων, η διαφορά τους όμως είναι ότι ενώ στον τύπο BLOB γίνεται διάκριση μεταξύ κεφαλαίων και πεζών χαρακτήρων κάτι τέτοιο δεν ισχύει στον τύπο TEXT. Τέλος και στις δυο τύπους δεν είναι υποχρεωτικό να ορίσει ο χρήστης το μέγεθος.

ENUM : Μια λίστα στοιχείων από τα οποία πρέπει να επιλεγθεί μια τιμή η οποία όμως μπορεί να είναι και κενή(NULL). Ένας τύπος ENUM μπορεί να περιέχει μέχρι 65535 διαφορετικές τιμές.[1]

4.10 Εισαγωγή στην SQL

Η SQL προέρχεται από τα αρχικά των λέξεων Structured Query Language και πρόκειται για την πιο διαδεδομένη γλώσσα διαχείρισης βάσεων δεδομένων. Όταν δημιουργούμε μια δικτυακή εφαρμογή η οποία χρησιμοποιεί βάση δεδομένων, το πιο ουσιώδες σημείο είναι η διεπαφή (interface) που χρησιμοποιούμε για να έχουμε πρόσβαση στη βάση αυτή. Τη διεπαφή αυτή τη προσπελάζουμε μέσω της γλώσσας SQL. Τέλος η SQL χρησιμοποιείται σε βάσεις δεδομένων που χρησιμοποιούν το σχεσιακό μοντέλο δεδομένων.

4.11 Εντολές της SQL και χρήση τους

Στο σημείο αυτό θα παρουσιάσουμε τις βασικές εντολές της SQL, που χρησιμοποιούνται για τη δημιουργία και τη διαχείριση μιας βάσης δεδομένων.

Εντολή CREATE : Η εντολή αυτή χρησιμοποιείται για τη δημιουργία ενός πίνακα. Η σύνταξη για την εντολή δημιουργίας πίνακα είναι :

```
CREATE TABLE όνομα-πίνακα (όνομα_στήλης, τύπος_στήλης);
```

Όπως βλέπουμε από τη σύνταξη της εντολής CREATE απαιτείται να δώσει ο χρήστης ένα όνομα δικής του επιλογής για τον πίνακα, να ορίσει τα ονόματα των πεδίων-στηλών και να δώσει ένα ορισμό για κάθε πεδίο.

Εντολή INSERT : Η εντολή INSERT χρησιμοποιείται για να προσθέσει ο χρήστης νέες εγγραφές στους πίνακες. Η σύνταξη της εντολής αυτής είναι :

```
INSERT INTO όνομα_πίνακα (λίστα_στηλών ) VALUES (τιμές_στηλών);
```

Είναι υποχρεωτικό στη λίστα τιμών μέσα στις παρενθέσεις τα αλφαριθμητικά να περικλείονται σε εισαγωγικά.

Εντολή SELECT : Η εντολή SELECT χρησιμοποιείται για την ανάκτηση εγγραφών από ένα πίνακα. Η σύνταξη της εντολής αυτής είναι η εξής :

```
SELECT εκφράσεις_και_στήλες FROM όνομα_πίνακα  
[WHERE συνθήκη_αληθείας]  
[ORDER BY κάποια_στήλη [ASC| DESC]]  
[LIMIT μετάθεση, γραμμές ]
```

Μια σημαντική έκφραση της εντολής αυτής είναι η μορφή :

SELECT * FROM όνομα_πίνακα;

Με αυτή την εντολή ο χρήστης επιλέγει όλες τις γραμμές και τις στήλες από το συγκεκριμένο πίνακα. Αν όμως θέλει να επιλέξει συγκεκριμένες στήλες τότε ο χαρακτήρας * αντικαθίσταται με τα ονόματα των στηλών διαχωρισμένα με κόμματα. Τα αποτελέσματα που εκτελούνται με την εντολή SELECT σε ένα πίνακα μιας βάσης δεδομένων, πραγματοποιούνται με τη σειρά με την οποία ο χρήστης εισάγει τις εγγραφές στον πίνακα. Αν τώρα θέλει να τα ταξινομήσει με αύξουσα σειρά, τη δουλειά αυτή την κάνει η πρόταση ORDER BY σε συνδυασμό με την εντολή SELECT και έχει τη μορφή :

```
SELECT εκφράσεις_και_στήλες FROM όνομα_πίνακα  
order by κάποια_στήλη;
```

Με την πρόταση αυτή τα αποτελέσματα ταξινομούνται κατά αύξουσα σειρά. Μπορεί όμως να τα ταξινομήσει και κατά φθίνουσα σειρά με τη δεσμευμένη λέξη DESC και η εντολή έχει τη μορφή :

```
SELECT εκφράσεις_και_στήλες FROM όνομα_πίνακα  
order by κάποια_στήλη DESC;
```

Εντολή LIMIT : Με την εντολή LIMIT ο χρήστης μπορεί να ανακτήσει συγκεκριμένες εγγραφές που χρειάζεται. Για να χρησιμοποιήσει αυτή την εντολή είναι απαραίτητο να καθορίσει δυο παραμέτρους, την μετάθεση(offset) δηλαδή την έναρξη και το πλήθος των γραμμών.

Εντολή WHERE : Η εντολή WHERE χρησιμοποιείται για να μπορεί ο χρήστης εκτός από συγκεκριμένες στήλες από πίνακες μιας βάσης δεδομένων να ανακτά και συγκεκριμένες γραμμές (εγγραφές). Η σύνταξη της εντολής WHERE είναι η εξής :

```
SELECT εκφράσεις_και_στήλες FROM όνομα_πίνακα  
[WHERE συνθήκη_αληθείας ]
```

5

Δικτυακός προγραμματισμός με την PHP

5.1 Εισαγωγή στην PHP

Η PHP είναι μια γλώσσα συγγραφής σεναρίων (scripting language), η οποία ενσωματώνεται μέσα στον κώδικα της HTML και εκτελείται στην πλευρά του server (server-side scripting). Θεωρείται από τις πιο δημοφιλείς γλώσσες για τη δημιουργία δικτυακών εφαρμογών. Πρόκειται για μια γλώσσα προγραμματισμού ανοιχτού κώδικα, πράγμα που σημαίνει ότι ο οποιοσδήποτε μπορεί να τη χρησιμοποιήσει για να δημιουργήσει κώδικα (scripts), που τρέχει στο web server και να κατασκευάσει κάποια εφαρμογή η οποία θα χρησιμοποιεί βάσεις δεδομένων. Τα PHP scripts χρησιμοποιούν κυρίως server-side scripting, ο οποίος απαιτεί τρία πράγματα για να δουλέψει. Ένα συντάκτη κειμένου όπου γράφουμε κώδικα, ένα web server και ένα web browser. Αρκεί να ξεκινήσει η λειτουργία του web server και να γίνει η σύνδεση με την PHP. Έτσι ο χρήστης θα μπορεί πλέον να προσπελάσει το αποτέλεσμα ενός script που έχει δημιουργήσει σε PHP με ένα server, βλέποντας τη σελίδα που έχει δημιουργήσει μέσα από τον server. [1]

5.2 Εγκατάσταση της PHP

Οι οδηγίες εγκατάστασης που θα παρουσιάσω παρακάτω αναφέρονται στην έκδοση 5.2.9 της PHP. Η εταιρία που είναι υπεύθυνη για τη δημιουργία και τη διανομή του λογισμικού της η PHP Group χρησιμοποιεί αριθμούς “ δευτερεύουσας έκδοσης “ για να χαρακτηρίσει τις ενημερώσεις της κύριας έκδοσης του λογισμικού. Οι δευτερεύουσες εκδόσεις λογισμικού κυκλοφορούν όταν προκύψουν ενημερώσεις στον κώδικα ,δεν είναι προκαθορισμένος δηλαδή ο χρόνος που θα κυκλοφορήσουν, για αυτό συνίσταται ο χρήστης να ανατρέχει σε τακτά χρονικά διαστήματα στην ιστοσελίδα <http://www.php.net/ChangeLog-5.php> για να ενημερώνεται για τυχόν νέες εκδόσεις του λογισμικού.

5.2.1 Εγκατάσταση της PHP σε συστήματα Linux/Unix

Η εγκατάσταση της PHP σε ένα σύστημα Linux/Unix δεν είναι δύσκολη διαδικασία. Αρκεί να μεταβεί ο χρήστης στην ιστοσελίδα <http://www.php.net> και να ακολουθήσει την ενότητα Downloads. Στη συνέχεια μεταφέρει το αρχείο αυτό σε ένα κατάλογο ο οποίος περιέχει αρχεία πηγαίου κώδικα, το αποσυμπιέζει και έπειτα μεταβαίνει στον κατάλογο της PHP. Εκεί υπάρχει ένα script με το όνομα configure, το οποίο δέχεται πληροφορίες όταν το τρέχει ο χρήστης από τη γραμμή εντολής, και τα ορίσματα των γραμμών εντολής ελέγχουν ποιες δυνατότητες θα υποστηρίξει η PHP. Όταν ολοκληρωθεί η εκτέλεση του script ο χρήστης εισάγει την εντολή make και στην πορεία την εντολή make install. Με την πληκτρολόγηση των εντολών αυτών, ολοκληρώνεται η διαδικασία εγκατάστασης της PHP. Τέλος είναι πολύ σημαντικό ο χρήστης, για να είναι σίγουρος ότι λειτουργεί σωστά η PHP και συνεργάζεται σωστά με τον Apache, να αντιγράψει το αρχείο php.ini στην προκαθορισμένη θέση του και το object αρχείο του στον κατάλογο εγκατάστασης του Apache.

5.2.2 Εγκατάσταση της PHP στα Windows

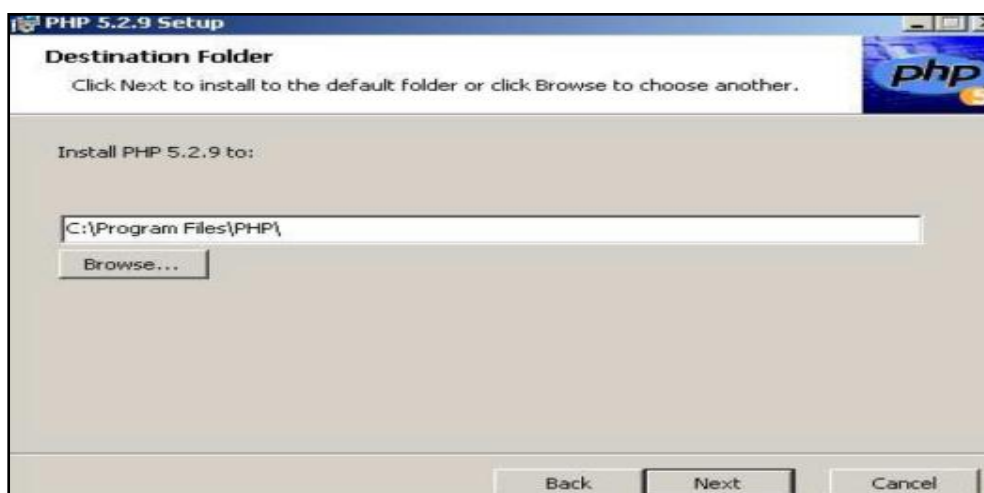
Η εγκατάσταση της PHP είναι μια εύκολη διαδικασία. Αρκεί να μεταβεί ο χρήστης στην ιστοσελίδα <http://www.php.net> και να κατεβάσει από εκεί ένα PHP installer. Η έκδοση που θα χρησιμοποιήσουμε είναι η 5.2.9 . Παρακάτω θα περιγράψω αναλυτικά τα βήματα που θα χρειαστούν για την εγκατάσταση της στο σύστημα μας.

1) Κάνουμε διπλό κλικ στο αρχείο εγκατάστασης. Το πρώτο παράθυρο που εμφανίζεται μας ενημερώνει για την έκδοση της PHP που πρόκειται να εγκαταστήσουμε στο σύστημα μας, όπως επίσης μας προτρέπει να απενεργοποιήσουμε κάποιο πρόγραμμα anti-virus που πιθανόν να χρησιμοποιούμε, ή οποιοδήποτε άλλο λογισμικό τρέχουμε κατά τη διάρκεια της εγκατάστασης.

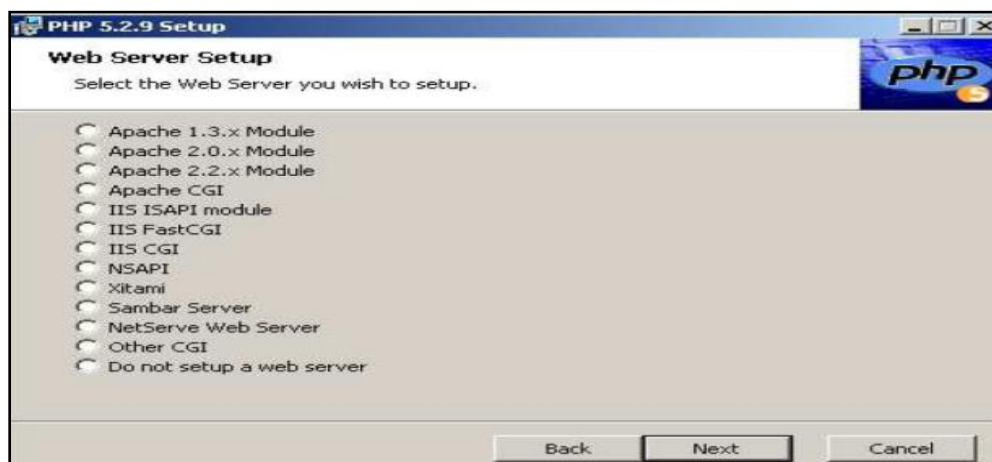


Στο δεύτερο παράθυρο ο οδηγός εγκατάστασης μας δίνει κάποιες πληροφορίες σχετικά με την άδεια χρήσης της συγκεκριμένης έκδοσης που θα χρησιμοποιήσουμε, και στην πορεία μας ζητάει να αποδεχτούμε κάποιους όρους που θέτει για να μπορέσουμε να προχωρήσουμε.

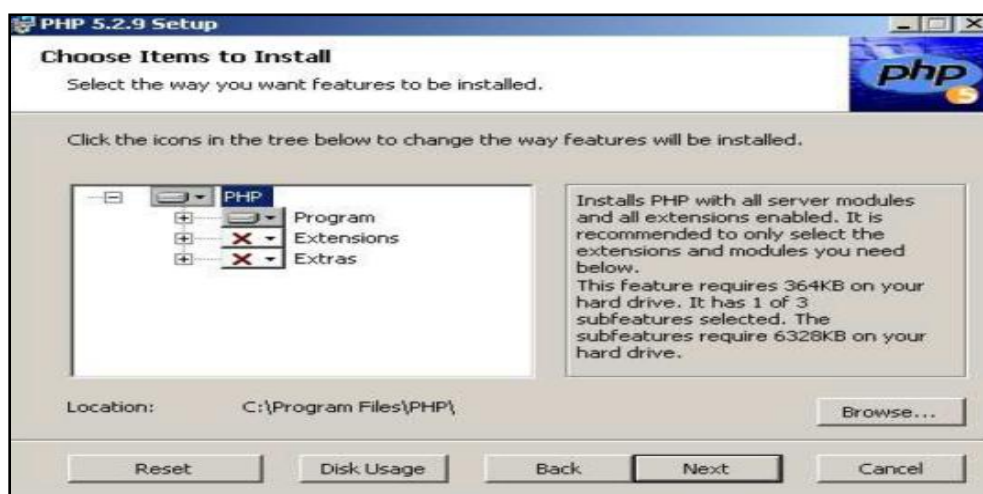
2) Επιλέγουμε το φάκελο όπου θα γίνει η εγκατάσταση της PHP και έπειτα πατάμε Next.



3) Επιλέγουμε τον Server τον οποίο τρέχουμε στο σύστημα μας καθώς και τον φάκελο ο οποίος περιέχει τα αρχεία διαμόρφωσης του Apache Web Server και κάνουμε κλικ στο κουμπί Next.



4) Επιλέγουμε ποια στοιχεία θέλουμε να εγκατασταθούν και έπειτα ο οδηγός μας προτρέπει να πατήσουμε την επιλογή Install για να ξεκινήσει η εγκατάσταση.



5) Όταν ολοκληρωθεί η εγκατάσταση της PHP στο σύστημα μας, ο οδηγός μας στέλνει ένα μήνυμα ότι η διαδικασία έχει πραγματοποιηθεί. Κάνουμε κλικ στην επιλογή finish για να βγούμε από τον οδηγό εγκατάστασης και πλέον ο χρήστης είναι έτοιμος να ξεκινήσει να προγραμματίζει γράφοντας τα πρώτα του script.

Ένα πολύ σημαντικό αρχείο της PHP είναι το **php.ini** . Στο αρχείο αυτό μπορεί να παρέμβει ο χρήστης και να κάνει διάφορες τροποποιήσεις στην PHP. Επιβάλλεται όμως να επανεκκινήσει τον server για να εφαρμοστούν οι αλλαγές που πραγματοποίησε. Βρίσκεται μέσα στον κατάλογο των Windows, στο αρχείο **httpd.conf** .Το αρχείο αυτό περιλαμβάνει τις λεγόμενες ντιρεκτίβες οι οποίες έχουν δυο μορφές: τιμές (values) και σημάνσεις (flags). Οι πιθανές τιμές που έχουν οι ντιρεκτίβες διαφέρουν από τη μια στην άλλη.

5.3 Τα βασικά της PHP

Για να ξεκινήσουμε να τρέχουμε τα πρώτα script σε PHP αρκεί να γνωρίζουμε ότι τα δημιουργούμε με οποιονδήποτε συντάκτη κειμένων θέλουμε, διότι τα αρχεία της PHP αποτελούνται από απλό κείμενο. Αφότου δημιουργήσουμε το πρώτο μας script, το αποθηκεύουμε στην κατάλληλη θέση με ένα όνομα που θέλουμε, ακολουθούμενο από την κατάληξη .php . Πλέον το script μας μπορούμε να το προσπελάσουμε με οποιαδήποτε εφαρμογή Web browser θέλουμε, για να δούμε τι θα εμφανίσει. Ο κώδικας PHP που γράφουμε οριοθετείται στην αρχή, από το tag αρχής (**<?php**) και στο τέλος από το tag τέλους (**?>**). Αυτό γίνεται για να μπορεί να τον ξεχωρίζει ο Web browser από τον κώδικα html.

Επίσης καλό είναι ο χρήστης όταν γράφει κώδικα να προσθέτει και κάποια σχόλια τα οποία θα τον κάνουν πιο ευανάγνωστο και κατανοητό από τον ίδιο αλλά και από τους τρίτους. Ένα σχόλιο είναι κείμενο και αγνοείται από το μηχανισμό εκτέλεσης της PHP . Ένα σχόλιο μιας γραμμής ξεκινά με δυο χαρακτήρες // ή με ένα χαρακτήρα #. Τα σχόλια τα οποία καταλαμβάνουν περισσότερες γραμμές ξεκινάνε με τους χαρακτήρες /* και τελειώνουν με τους ίδιους χαρακτήρες.

5.4 Σύνδεση της PHP με τη MySQL

Η PHP όπως αναφέραμε νωρίτερα θεωρείται ιδανική γλώσσα για τη δημιουργία δυναμικών ιστοσελίδων οι οποίες επικοινωνούν με βάσεις δεδομένων, στην παρούσα διπλωματική με τη MySQL, διότι διαθέτει ενσωματωμένες συναρτήσεις με τη βοήθεια των οποίων γίνεται η επικοινωνία με τις βάσεις δεδομένων. Παρακάτω θα παραθέσω ένα κομμάτι κώδικα από τα script που έχω δημιουργήσει, για να παρουσιάσω τις συναρτήσεις της PHP που χρησιμοποιούνται για τη σύνδεση με τη βάση δεδομένων **metabolites_second** που έχω δημιουργήσει, τη δημιουργία ερωτημάτων (query) και την ανάκτηση αποτελεσμάτων. Έστω το ακόλουθο κομμάτι κώδικα :

```
//Δημιουργία σύνδεσης με τη MySQL

$db = mysql_connect('localhost', 'admin', 'admin') or

//Εμφάνιση σφάλματος σε περίπτωση αποτυχημένης σύνδεσης

die ('Unable to connect. Check your connection parameters.');
```

```
//Επιλογή της βάσης δεδομένων metabolites_second

mysql_select_db('metabolites_second', $db) or die(mysql_error($db));
```

```
//Δημιουργία μιας επερώτησης (query) προς τη βάση δεδομένων

$result = mysql_query($query, $db) or die(mysql_error());
```

```
//Κλείσιμο της σύνδεσης με τη MySQL

mysql_close($connection);
```

```
//Επιστρέφει μια γραμμή από ένα σύνολο εγγραφών
```

```
while ($row = mysql_fetch_assoc($result)) {  
echo ($odd == true) ? '<tr class="odd_row">' : '<tr class="even_row">';
```

Επεξήγηση συναρτήσεων της mysql

- Η συνάρτηση **mysql_connect** χρησιμοποιείται για τη σύνδεση με τη MySQL και δέχεται τρεις παραμέτρους: το hostname, το όνομα του χρήστη που έχει πρόσβαση στο server και το κωδικό ασφαλείας του.
- Η συνάρτηση **mysql_error** επιστρέφει μήνυμα λάθους σε περίπτωση αποτυχημένης σύνδεσης προς την MySQL.
- Η συνάρτηση **mysql_select_db** επιλέγει μια συγκεκριμένη βάση δεδομένων την οποία έχει δημιουργήσει ήδη ο χρήστης στη MySQL. Οι παράμετροι που δέχεται είναι το όνομα της βάσης και η τιμή που επιστράφηκε από τη **mysql_connect()**.
- Η συνάρτηση **mysql_query** χρησιμοποιείται για τη αποστολή ενός ερωτήματος (query) προς τη βάση δεδομένων.
- Η συνάρτηση **mysql_close** χρησιμοποιείται για να κλείσει η σύνδεση με τη MySQL.
- Η συνάρτηση **mysql_fetch_assoc** δέχεται σαν όρισμα το αποτέλεσμα ενός ερωτήματος (query) στη βάση δεδομένων και επιστρέφει μια γραμμή από ένα σύνολο εγγραφών σε μορφή συσχετιστικού πίνακα.

6

HTML

6.1 Γενικά χαρακτηριστικά

Η HTML είναι η κύρια γλώσσα σήμανσης που χρησιμοποιείται για τη δημιουργία ιστοσελίδων και τα αρχικά της σημαίνουν Hyper Text Markup Language. Δεν είναι μια πραγματική γλώσσα προγραμματισμού, διότι δεν διαθέτει τις συνηθισμένες εντολές που χαρακτηρίζουν τις γλώσσες προγραμματισμού. Χρησιμοποιείται ουσιαστικά για τον προσδιορισμό του περιεχομένου ενός εγγράφου, καθώς και για τον τρόπο με τον οποίο το έγγραφο αυτό θα παρουσιαστεί στο web. Περιγράφει με ακρίβεια το περιεχόμενο ενός εγγράφου, αλλά ο τρόπος με τον οποίο θα εμφανισθεί το αποτέλεσμα στο web εξαρτάται από το χρησιμοποιούμενο browser. Ουσιαστικά οι browsers διαβάζουν τα αρχεία και εμφανίζουν το αποτέλεσμά του κώδικα HTML στην οθόνη μας.

Ο κώδικας HTML αποτελείται από κάποια βασικά δομικά στοιχεία, τα οποία είναι οι ετικέτες (**tags**) και οι ιδιότητες (**attributes**). Οι ετικέτες είναι κώδικας HTML που περικλείεται ανάμεσα στα σύμβολα μικρότερο (<) και μεγαλύτερο (>) και πρακτικά περιγράφουν το περιεχόμενο μιας ιστοσελίδας. Επίσης οι ετικέτες μπορεί να περιέχουν και άλλα στοιχεία HTML τα οποία καλούνται ιδιότητες (attributes). Οι ιδιότητες είναι ένα κομμάτι κώδικα, που προσδιορίζει κάποια επιπλέον χαρακτηριστικά του περιεχομένου της ετικέτας.[3]

6.2 Επεξεργαστής κειμένου

Τα αρχεία HTML είναι απλά αρχεία κειμένου, οπότε οποιοσδήποτε επεξεργαστής κειμένου είναι αρκετός. Για το λειτουργικό των Windows ο επεξεργαστής κειμένου Notepad μπορεί να χρησιμοποιηθεί αποτελεσματικά. Όταν δημιουργηθεί ένα αρχείο κειμένου με οποιοσδήποτε κειμενογράφο θα πρέπει απαραίτητα να αποθηκευθεί με την απόληξη (.html) ή (.htm) για να μπορέσει στη συνέχεια να διαβασθεί από τον browser.

6.3 Δομή ενός εγγράφου HTML

Ένα έγγραφο HTML πρέπει να έχει την παρακάτω δομή :

```
<HTML>
<HEAD>
<TITLE>          </TITLE>
<BODY>
</BODY>
</HTML>
```

Η πρώτη ετικέτα που βλέπουμε η <HTML> είναι το πρώτο στοιχείο ενός εγγράφου το οποίο δηλώνει πως πρόκειται για έγγραφο HTML. Η ετικέτα έναρξης <HTML> τοποθετείται στην αρχή του αρχείου, ενώ η ετικέτα τερματισμού </HTML> τοποθετείται στο τέλος του αρχείου. Η ετικέτα <HEAD> περιέχει

πληροφορίες σχετικά με το αρχείο HTML. Η ετικέτα <TITLE> όπως βλέπουμε τοποθετείται εντός της ετικέτας <HEAD> και μέσα σε αυτή τοποθετείται κείμενο το οποίο εμφανίζεται σαν τίτλος στη γραμμή τίτλου του browser. Η ετικέτα <BODY> περιέχει όλα εκείνα τα στοιχεία που αποτελούν το κυρίως σώμα του εγγράφου. Αποτελεί ένα βρόγχο και το τέλος του καθορίζεται με την ετικέτα τερματισμού </BODY> η οποία όπως βλέπουμε τοποθετείται πριν την ετικέτα </HTML> στο τέλος του εγγράφου.

Ένα παράδειγμα μιας εντολής html η οποία δηλώνει ότι το κείμενο που περιέχεται ανάμεσα στα tags <p> και </p> θα πρέπει να απεικονιστεί σαν παράγραφος είναι το παρακάτω :

```
<p> This is the first paragraph of the text. </p>
```

Κώδικας 6.1 : Παράδειγμα εντολής HTML

Επίσης όπως αναφέραμε νωρίτερα οι εντολές της γλώσσας HTML δέχονται ιδιότητες (attributes), οι οποίες καθορίζουν την εμφάνιση ενός συγκεκριμένου τμήματος της σελίδας. Παράδειγμα χρήσης μιας τέτοιας ιδιότητας είναι ο κώδικας παρακάτω :

```
<p color= "green"> This is the the first paragraph of the text. </p>
```

Κώδικας 6.2 : Παράδειγμα εντολής HTML με ιδιότητα (attribute)

Στην περίπτωση αυτή ο browser διαβάζει την εντολή και μορφοποιεί το κείμενο της παραγράφου να έχει πράσινο χρώμα. Η HTML παρέχει μια πληθώρα εντολών τις οποίες μπορεί να χρησιμοποιήσει ο προγραμματιστής για να απεικονίσει με τον τρόπο που εκείνος θέλει την πληροφορία στον browser. Σκοπός της διπλωματικής δεν είναι να παρουσιάσω διεξοδικά τις εντολές αυτές, απλά πριν προχωρήσω καλό θα ήταν να αναφέρω ορισμένα από τα πιο συνηθισμένα tags και attributes που χρησιμοποιώ, στα scripts που δημιούργησα:

<p> : Ορίζει μια νέα παράγραφο και προσδιορίζει το κείμενο που βρίσκεται μέσα σε αυτή.

**
** : Καθορίζει αλλαγή γραμμής.

<h1>, <h2>, <h3>, <h4>, <h5>, <h6> : Καθορίζει επικεφαλίδες σε ένα εγγράφο, με το νούμερο 1 να αντιστοιχεί στο μεγαλύτερο μέγεθος γραμμάτων και το νούμερο 6 στο μικρότερο μέγεθος γραμμάτων που απεικονίζει η επικεφαλίδα.

<table> : Ορίζει γραμμές και στήλες και τις αντίστοιχες επικεφαλίδες τους που αποτελούν έναν πίνακα.

<form>: Ορίζει μια φόρμα η οποία αποτελείται από διάφορα πεδία και συμπληρώνεται από το χρήστη για τη παροχή δεδομένων στην εφαρμογή.

<div> : Διαχωρίζει μια ενότητα ενός εγγράφου για ειδική διαμόρφωση.

<a> : Η ετικέτα a χρησιμοποιείται σαν βάση για τη δημιουργία κάθε είδους υπερσύνδεσμου και διαθέτει την ιδιότητα **href** που καθορίζει τη διεύθυνση του αρχείου με το οποίο θα γίνει η σύνδεση.

Τέλος αξίζει να επισημάνουμε πως παρόλο που με τη βοήθεια της html μπορούμε να περιγράψουμε με ακρίβεια το περιεχόμενο ενός εγγράφου, δεν σημαίνει πως θα εμφανισθεί με απόλυτη ακρίβεια και το αποτέλεσμα στο web, διότι εξαρτάται σε μεγάλο βαθμό από το χρησιμοποιούμενο browser.[3]

6.4 Χρήση της HTML μαζί με PHP

Πολλές φορές ο PHP κώδικας συνδυάζεται με τον HTML κώδικα. Στα περισσότερα script κατά τη διάρκεια της διπλωματικής έχω συμπεριλάβει κώδικα PHP μέσα σε κώδικα HTML, οι οποίοι συνδυάζονται ώστε να σχηματίζουν ένα μεμονωμένο script. Παρακάτω παραθέτω ένα παράδειγμα για το πώς ενσωματώνεται η PHP σε ένα αρχείο html.

```
<html>
<head>
  <title> A new comic </title>
<body>
  <?php
    $output ="Read a new comic";
    echo $output;
  ?>
</body>
</html>
```

Κώδικας 6.3 : Παράδειγμα ενσωμάτωσης PHP σε κώδικα html

Ο κώδικας PHP ξεχωρίζει εύκολα διότι περικλείεται ανάμεσα από τις ετικέτες αρχής και τέλους(<?php και ?>). Έτσι ο προγραμματιστής αντί να γράφει ένα πρόγραμμα με αρκετές εντολές για να παράγει html, γράφει ένα html script με ενσωματωμένο κώδικα php για να πραγματοποιήσει αυτό που θέλει.

7

Υλοποίηση της δυναμικής ιστοσελίδας

7.1 Δημιουργία νέου χρήστη και απόκτηση δικαιωμάτων

Η ανάγκη για την εμφάνιση δυναμικού περιεχομένου σε ιστοσελίδες είχε ως φυσικό επακόλουθο τη χρησιμοποίηση των βάσεων δεδομένων στις δικτυακές εφαρμογές. Με αυτό τον τρόπο αυτό γίνεται πιο εύκολη η διαχείριση και προβολή του περιεχομένου μιας δυναμικής ιστοσελίδας. Όλο το περιεχόμενο που βρίσκεται στην ιστοσελίδα μας αποθηκεύεται σε μια βάση δεδομένων που δημιουργήσαμε στον MySQL Server.

Όμως προτού μια φτιάξουμε αυτή τη βάση δεδομένων έπρεπε να έχουμε δικαίωμα από τον διαχειριστή να το κάνουμε. Δικαιώματα δημιουργίας και διαγραφής σε μια βάση δεδομένων έχει μόνο ο υπερχρήστης. Για αυτό το λόγο δημιουργήσαμε έναν νέο χρήστη πληκτρολογώντας την εξής εντολή στο τερματικό :

```
CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

(όπου στην περίπτωση μας επιλέξαμεμε newuser: **admin** και password: **admin**)

Στην συνέχεια επιλέξαμε τις πληροφορίες στις οποίες θέλουμε να έχει πρόσβαση ο χρήστης. Στην προκειμένη περίπτωση με την παρακάτω εντολή έχουμε πλήρη πρόσβαση.

```
GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

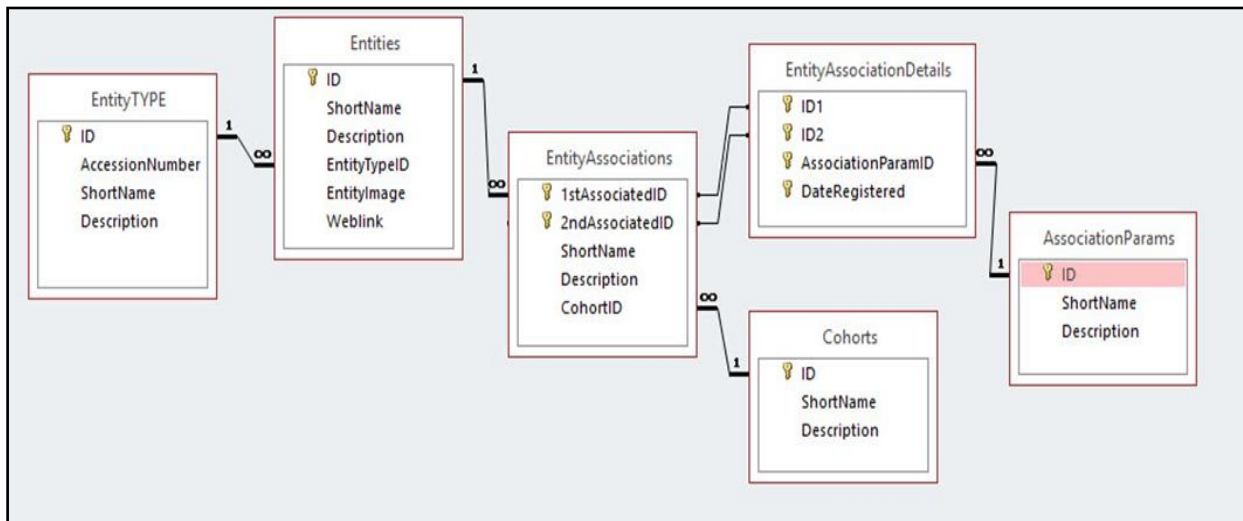
(οι αστερίσκοι αναφέρονται στην βάση δεδομένων και στον πίνακα που επιτρέπεται η πρόσβαση)

Έπειτα από τις δυο παραπάνω ενέργειες δημιουργήσαμε τη καινούργια βάση δεδομένων στην οποία καταχωρήσαμε όλους τους πίνακες που μου είναι απαραίτητοι για την ιστοσελίδα μας και της δώσαμε το όνομα **metabolites_second**. Η εντολή SQL για τη δημιουργία αυτής της βάσης είναι η εξής :

```
CREATE DATABASE metabolites_second;
```

7.2 Διάγραμμα βάσης δεδομένων και δημιουργία πινάκων

Στην παρακάτω εικόνα παρουσιάζω το διάγραμμα της βάσης δεδομένων EuHAP, το οποίο αποτελείται από 6 πίνακες. Στη συνέχεια εξηγώ συνοπτικά τις στήλες από τις οποίες αποτελείται ο κάθε ένας πίνακας και παραθέτω τις εντολές SQL με τις οποίες τους πρόσθεσα στη βάση δεδομένων μας metabolites_second.



Σχήμα 7.1 : Διάγραμμα βάσης δεδομένων EuHAP

Ο πίνακας Entity Type

Στον πίνακα EntityType έχουμε τέσσερα χαρακτηριστικά, ή αλλιώς τέσσερις στήλες :

- ID : Το ID όπου είναι το πρωτεύον κλειδί (**primary key**) του πίνακα. Για παράδειγμα μπορεί να έχουμε το Entity Type Methylation Markers με ID ίσο με 1. Τώρα αν έχουμε ένα άλλο Entity Type για παράδειγμα τα Metabolites, το ID αυτού του Entity δεν μπορεί να είναι ίσο με 1 αλλά θα έχει άλλη αριθμητική τιμή. Αυτό το χαρακτηριστικό συνδέεται με το EntityTypeID (δευτερεύον κλειδί) του πίνακα Entities.
- AccessionNumber : Είναι ο αριθμός που μπορεί να εισαγάγει ο χρήστης για το Entity Type.
- ShortName : Είναι το όνομα του Entity Type που προσθέτουμε.
- Description : Είναι η περιγραφή που κάνουμε για κάθε Entity.

Οι εντολές SQL που χρησιμοποιήσαμε για τη δημιουργία του πίνακα EntityType είναι οι παρακάτω:

```
CREATE TABLE entity_type (
  id int(11) NOT NULL AUTO_INCREMENT,
  accession_number int(11) DEFAULT NULL,
  short_name varchar(100) DEFAULT NULL,
  description text,
  PRIMARY KEY (id)
```

Πίνακας 7.1 : Εντολές SQL για τη δημιουργία του πίνακα EntityType

Ο πίνακας Entities

Στον πίνακα Entities έχουμε τα έξι ακόλουθα χαρακτηριστικά :

- ID : Το ID είναι το πρωτεύον κλειδί (**primary key**) αυτού του πίνακα. Για παράδειγμα μπορεί να έχουμε το Entity Aspartate με ID ίσο με 1. Τώρα αν έχουμε ένα άλλο Entity για παράδειγμα το Glycoxine ,το ID αυτού του Entity δεν μπορεί να είναι ίσο με 1 αλλά θα έχει άλλο νούμερο διαφορετικό του 1.
- ShortName : Είναι το όνομα του Entity που προσθέτουμε στο site μας.
- Description : Είναι η περιγραφή για κάθε Entity.

- EntityTypeID : Είναι δευτερεύον κλειδί (**foreign key**) που παραπέμπει στο ID του πίνακα EntityType. Για παράδειγμα αν το ID ενός EntityType πχ. του PTM είναι 4 και σχετίζεται με το Entity Aspartate τότε το EntityTypeID στον πίνακα θα είναι 4.
- Image : Είναι μια εικόνα την οποία μπορεί ο χρήστης να προσθέσει σε κάποιο Entity.
- Weblink : Μπορεί να είναι μια διεύθυνση URL στο Internet, η οποία θα περιέχει πληροφορίες που θα σχετίζονται με κάποιο Entity.

Οι εντολές SQL που χρησιμοποίησα για τη δημιουργία του πίνακα Entities είναι οι παρακάτω :

```
CREATE TABLE entity (
  id int(11) NOT NULL AUTO_INCREMENT,
  short_name varchar(100) DEFAULT NULL,
  description text,
  entity_type_id int(11) NOT NULL,
  entity_image mediumblob,
  weblink text,
  PRIMARY KEY (id),
  KEY EntityTypeID (entity_type_id),
  CONSTRAINT entity_ibfk_1 FOREIGN KEY (entity_type_id) REFERENCES entity_type (id) ON
  DELETE CASCADE ON UPDATE CASCADE
```

Πίνακας 7.2 : Εντολές SQL για τη δημιουργία του πίνακα Entities

Ο πίνακας EntityAssociations

Στον πίνακα EntityAssociations έχουμε τα 5 ακόλουθα χαρακτηριστικά :

- 1st Associated Entity ID : Αναφέρεται στο πρώτο Entity που θέλουμε να συνδέσουμε με κάποιο άλλο.
- 2nd Associated Entity ID : Αναφέρεται στο δεύτερο Entity που θέλουμε να συνδέσουμε με κάποιο άλλο
- Description : Είναι η περιγραφή για κάθε association που δημιουργούμε.
- Cohort ID : Αυτό το ID είναι δευτερεύον κλειδί (**foreign key**) και αναφέρεται στο μοναδικό ID του πίνακα Cohorts.

Τέλος η τριπλέτα 1st Associated Entity ID , 2nd Associated Entity ID και Cohort ID συμπεριφέρεται σαν πρωτεύον κλειδί (**primary key**) σε αυτό τον πίνακα.

```
CREATE TABLE entity_association (
  entity_id_1 int(11) NOT NULL,
  entity_id_2 int(11) NOT NULL,
  description text,
  cohort_id int(11) NOT NULL DEFAULT 0,
  PRIMARY KEY (entity_id_1,entity_id_2),
  KEY entity (entity_id_1),
  KEY entity2 (entity_id_2),
  CONSTRAINT entity FOREIGN KEY (entity_id_1) REFERENCES entity (id) ON DELETE          CASCADE ON
  UPDATE NO ACTION,
  CONSTRAINT `entity2` FOREIGN KEY (`entity_id_2`) REFERENCES `entity` (`id`) ON DELETE  CASCADE
  ON UPDATE NO ACTION
```

Πίνακας 7.3: Εντολές SQL για τη δημιουργία του πίνακα EntityAssociations

Ο πίνακας EntityAssociationDetails

Στον πίνακα EntityAssociationDetails έχουμε τα ακόλουθα χαρακτηριστικά.

- ID1 : Είναι το δευτερεύον κλειδί (**foreign key**) που σχετίζεται με το 1st Associated Entity ID από τον πίνακα EntityAssociations.
- ID2 : Είναι το δευτερεύον κλειδί (**foreign key**) που σχετίζεται με το 2nd Associated Entity ID από τον πίνακα EntityAssociations.
- Association Parameters ID : Είναι δευτερεύον κλειδί(**foreign key**) και σχετίζεται με το ID από τον πίνακα Association Parameters.
- Date Registered : Είναι η ημερομηνία που δημιουργήθηκε το Association.

Τέλος το πρωτεύον κλειδί (**primary key**) αυτού του πίνακα είναι η τετράδα ID1, ID2, AssociationParametersID και DateRegistered.

Οι εντολές SQL που χρησιμοποίησα για τη δημιουργία του πίνακα EntityAssociationDetails είναι οι παρακάτω :

```
CREATE TABLE entity_association_details (  
id_1 int(11) NOT NULL,  
id_2 int(11) NOT NULL,  
assoc_param_id int(11) NOT NULL,  
date_registered timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
PRIMARY KEY (id_1, id_2, assoc_param_id ),  
KEY id1 (id_1),  
KEY id2 (id_2),  
KEY param_id (assoc_param_id ),  
CONSTRAINT id1 FOREIGN KEY (id_1) REFERENCES entity_association (entity_id_1) ON DELETE CASCADE  
ON UPDATE NO ACTION,  
CONSTRAINT id2 FOREIGN KEY (id_2) REFERENCES entity_association (entity_id_2) ON DELETE CASCADE  
ON UPDATE NO ACTION,  
CONSTRAINT param_id FOREIGN KEY (assoc_param_id) REFERENCES association_parameter (ID) ON  
DELETE CASCADE ON UPDATE NO ACTION
```

Πίνακας 7.4 : Εντολές SQL για τη δημιουργία του πίνακα EntityAssociationDetails

Ο πίνακας Cohorts

Στον πίνακα Cohorts έχουμε τα ακόλουθα χαρακτηριστικά.

- ID : Είναι το πρωτεύον κλειδί (primary key) αυτού του πίνακα.
- ShortName : Είναι το όνομα για κάθε Cohort.
- Description : Είναι η περιγραφή για κάθε Cohort.

Οι εντολές SQL που χρησιμοποίησα για τη δημιουργία του πίνακα Cohorts είναι οι παρακάτω :

```
CREATE TABLE cohort (  
id int(11) NOT NULL AUTO_INCREMENT,  
short_name varchar(100) DEFAULT NULL,  
description text,  
PRIMARY KEY (id)
```

Πίνακας 7.5 : Εντολές SQL για τη δημιουργία του πίνακα Cohorts

Ο πίνακας Association Parameters

Στον πίνακα Association Parameters έχουμε τα ακόλουθα χαρακτηριστικά.

- ID : Είναι το πρωτεύον κλειδί (**primary key**) αυτού του πίνακα. Συνδέεται με το Association Parameters ID του πίνακα Association Details.
- ShortName : Είναι το όνομα για κάθε Association Parameter.
- Description : Είναι η περιγραφή για κάθε Association Parameters.

Οι εντολές SQL που χρησιμοποίησα για τη δημιουργία του πίνακα Association Parameters είναι οι παρακάτω :

```
CREATE TABLE association_parameter (  
ID int(11) NOT NULL AUTO_INCREMENT,  
ShortName varchar(100) DEFAULT NULL,  
Description text,  
PRIMARY KEY (ID)
```

Πίνακας 7.6 : Εντολές SQL για τη δημιουργία του πίνακα EntityAssociationDetails

Ο πίνακας user

Τέλος έχω δημιουργήσει τον πίνακα user, όπου εκεί καταχωρούνται τα στοιχεία των χρηστών οι οποίοι εγγράφονται στην ιστοσελίδα μας και περιλαμβάνει τις εξής στήλες:

- ID : Είναι το πρωτεύον κλειδί (**primary key**) αυτού του πίνακα.
- first_name : Είναι το μικρό όνομα του χρήστη.
- last_name: Είναι το επίθετο του χρήστη.
- email: Είναι η διεύθυνση ταχυδρομείου του χρήστη, όπου εκεί του έρχεται μήνυμα αν τυχόν έχει ξεχάσει το κωδικό πρόσβασης (password) με τον οποίο κάνει login στην ιστοσελίδα μας.
- organization: Είναι ο οργανισμός που ανήκει ο χρήστης.
- username: Είναι το όνομα χρήστη με το οποίο ο επισκέπτης κάνει login στην ιστοσελίδα.
- password: Είναι ο κωδικός πρόσβασης με τον οποίο ο χρήστης κάνει login στην ιστοσελίδα.

Οι εντολές SQL που χρησιμοποίησα για τη δημιουργία του πίνακα user είναι οι παρακάτω :

```
CREATE TABLE user (  
id int(11) NOT NULL AUTO_INCREMENT,  
first_name varchar(30) NOT NULL,  
last_name varchar(30) NOT NULL,  
email varchar(40) NOT NULL,  
organization varchar(40) NOT NULL,  
username varchar(40) NOT NULL,  
password varchar(100) NOT NULL,  
PRIMARY KEY (id),  
UNIQUE KEY username_UNIQUE (username),  
UNIQUE KEY email_UNIQUE (email)
```

Πίνακας 7.7 : Εντολές SQL για τη δημιουργία του πίνακα user

7.3 Απαραίτητες ρυθμίσεις στον Apache για να “ανεβάσω” τη δυναμική ιστοσελίδα

Το τελευταίο βήμα αφότου δημιούργησα τα απαραίτητα scripts για τη δημιουργία της ιστοσελίδας, ήταν να προβώ στις απαραίτητες ρυθμίσεις, ώστε να “ανεβάσω” τις σελίδες μας στον Apache server. Αυτό πραγματοποιήθηκε με τη βοήθεια του **Virtual hosting** (εικονική φιλοξενία) . Πρόκειται για χρήση πολλαπλών “εικονικών συστημάτων” (virtual hosts) που φιλοξενούν πολλαπλά sites στον ίδιο web server. Ένας virtual host μπορεί να είναι IP-based ή name-based. Στον παρακάτω πίνακα βλέπουμε τις απαραίτητες ρυθμίσεις που πραγματοποιήσαμε στο αρχείο **httpd-vhosts** που βρίσκεται στα αρχεία διαμόρφωσης του Apache.

```
<VirtualHost *:80>
DocumentRoot "C:/xampp/finalproject/"
ServerName eurohealth.com

<Directory "C:/xampp/finalproject/">
Options All
AllowOverride All
Require all granted
</Directory>
</VirtualHost>
```

Πίνακας 7.8 : Ρυθμίσεις στο αρχείο httpd-vhosts του Apache

Όπως βλέπουμε έχει δηλωθεί η πόρτα 80 για την επικοινωνία του server, το μέρος δηλαδή όπου βρίσκονται τα αρχεία μας καθώς και το όνομα του server μας (*eurohealth.com*). Άλλη μια ρύθμιση απαιτείται στο αρχείο **hosts**, του συστήματος των windows, ώστε να κάνουμε την αντιστοιχία της IP διεύθυνσης με το όνομα του server: *127.0.0.1 eurohealth.com*. Η διεύθυνση **127.0.0.1** δηλώνει τον τοπικό υπολογιστή μας (*localhost*).

7.4 Παρουσίαση της δυναμικής ιστοσελίδας

Ετικέτα Home

Η πρώτη σελίδα που αντικρίζει ο χρήστης όταν επισκέπτεται την ιστοσελίδα μας είναι η παρακάτω :

Welcome to EurHEALTHAgeing Database Webpage

[Home](#) [Project Information](#) [Partners](#) [Login](#)

EurHEALTHAgeing is a **multidisciplinary project** linking studies of **early developmental processes** with those on **longevity** and **ageing** by combining use of cutting edge technologies- **post-translational modifications, DNA methylation, metabolomics** and **genomics** with unique cohorts with multiple ageing phenotypes. We will use **four key European study cohorts** with rich early life data and age-related health outcomes. We will explore a unique twin dataset, two birth cohorts with extensive maternal, pre- and peri-natal data and a population based cohort of subjects in their 60s-70s. By analysing specific age related metabolomics, epigenetic (methylation and post-translational modification) data and correlating it to early life events, genetics and ageing outcomes we will provide **major insights into developmental processes that influence longevity and ageing**. By focussing on molecular mechanisms involved in both early life events and ageing we will **find genes and metabolic and lifecourse pathways that are relevant in both early development and adult life**.

This will also allow us to develop **biomarkers** of ageing that reflect the role of early development on ageing; potentially **identifying pathways for therapeutic intervention when the process is still reversible**.

Εικόνα 7.1 : Αρχικό παράθυρο (HOME) της ιστοσελίδας

Στο πρώτο παράθυρο (HOME) ο επισκέπτης ενημερώνεται για το το EurHEALTHAgeing project. Πρόκειται για ένα διεπιστημονικό πρόγραμμα που συνδέει τις μελέτες για πρώιμες αναπτυξιακές διαδικασίες, με εκείνες που αφορούν τη μακροζωία και τη γήρανση, συνδυάζοντας τη χρήση τεχνολογιών αιχμής, μεταφραστικές τροποποιήσεις, μεθυλίωση του DNA και μοναδικές ομάδες (cohorts) που διαθέτουν πολλαπλούς φαινότυπους γήρανσης. Χρησιμοποιεί τέσσερις βασικές Ευρωπαϊκές ομάδες μελέτης, με πλούσια τα στοιχεία πρώιμης ζωής και αποτελέσματα της υγείας σχετικά με την ηλικία. Επίσης διερευνά ένα μοναδικό διπλό σύνολο δεδομένων, δυο κλάσεις γεννήσεων με εκτεταμένα γενετικά δεδομένα και μια βάση πληθυσμού βασισμένη σε ηλικίες 60-70 ετών. Με το να αναλύουμε τα δεδομένα που σχετίζονται με συγκεκριμένη ηλικία, και συσχετίζοντας τα με πρώιμα συμβάντα της ζωής θα παρέχουμε σημαντικές γνώσεις σε αναπτυξιακές διεργασίες που επηρεάζουν τη μακροβιότητα και τη γήρανση. Με το να επικεντρωνόμαστε σε μοριακούς μηχανισμούς που περιλαμβάνονται και στη πρώιμη ανάπτυξη και στην ενήλικη ζωή, θα βρούμε γονίδια και οδούς που είναι σχετικά και στα πρώτα χρόνια ζωής ενός ατόμου αλλά και στην ενήλικη ζωή. Αυτό θα μας επιτρέψει να αναπτύξουμε βιοδείκτες γήρανσης που απεικονίζουν το ρόλο της πρώιμης ανάπτυξης. Ενδεχομένως να βρούμε μονοπάτια για θεραπευτική παρέμβαση όταν η εξέλιξη θα είναι ακόμα αντιστρέψιμη.

Υπάρχουν αυξανόμενες ενδείξεις ότι βιολογικές διαταραχές κατά διάρκεια των πρώτων χρόνων ζωής ενός ατόμου μπορεί να οδηγήσουν σε ασθένειες με την πάροδο του χρόνου. Σύμφωνα με μελέτες έως το 2050 ο αριθμός των ατόμων στην Ευρωπαϊκή Ένωση που θα ξεπεράσουν την ηλικία των 80 ετών θα αυξηθεί κατά 170%, πράγμα το οποίο με τη σειρά του θα οδηγήσει σε αύξηση των δαπανών που σχετίζονται με την υγειονομική περίθαλψη. Το προσδόκιμο υγιούς ζωής πρέπει να εξελίσσεται παράλληλα με αυτές τις αλλαγές, ενώ οι το κόστος δαπανών πρέπει να περιορίζεται. Η αποτελεσματική πολιτική που ακολουθείται για την υγεία καθ' όλη τη διάρκεια της ζωής, μπορεί να υποστηρίξει σε μεγάλο βαθμό την υγιή γήρανση του πληθυσμού. Για αυτό το λόγο υπάρχει επιτακτική ανάγκη να εντοπίζονται γεγονότα από τα πρώτα χρόνια της ζωής ενός ατόμου, που καθορίζουν την πορεία της υγείας του με την πάροδο των χρόνων. Αυτός είναι και ο λόγος που δημιουργήθηκε το EurHEALTHAgeing project.

Ετικέτα Project Information

Όταν ο επισκέπτης επιλέξει την ετικέτα Project Information του εμφανίζεται το παρακάτω παράθυρο.

Welcome to EurHEALTHAgeing Database Webpage

[Home](#) [Project Information](#) [Partners](#) [Login](#)

Objectives of EurHEALTHAgeing

- 1) To **identify genetic factors** influencing **birthweight** (measures of size/shape: birthweight and length relative to gestational age, ponderal index) and other early postnatal determinants of healthy ageing (e.g. growth patterns/velocity) by carrying out a genomewide association scan (**GWAS**) on these traits.
- 2) To **identify** using **NextGen Sequencing (NGS)- rarer variants** i.e. single base pair changes, insertions, deletions and breakpoints of rarer copy number variations (CNVs), and assess their potential to influence transcripts' and proteins' functional capabilities and their **influence on birthweight and growth measures and ageing phenotypes**
- 3) To **identify DNA methylation, post-translational modifications and metabolite profiles** that correlate with prenatal/early post natal determinants of health-span (e.g. birthweight, weight and weight-gain of mother, smoking and alcohol drinking of mother, maternal pregnancy related diseases and complications).
- 4) To **discover novel biomarkers** of ageing using a multi-disciplinary approach incorporating pathway analysis, biochemistry, genetics, metabolomics, epidemiology, DNA methylation and bioinformatics.
- 5) To **identify**, using pathway analysis, the **molecular pathways** involved early life event variables that result in healthy ageing (musculoskeletal, cardiovascular, metabolic) by incorporating early life events, methylation, post-translational modification, metabolomic and genotypic data.
- 6) To **identify lifecourse trajectories leading to healthy aging** by incorporating early life events, methylation, post-translational modification, metabolomic and genotypic data.
- 7) To **generate an extensive database** of the correlations between early life events and ageing outcomes linking datapoints to methylation, metabolomic, genetic and post-translational modification (PTM) results, allowing future comparison with data derived from other studies. Results can then be employed in tests for healthy ageing.

Εικόνα 7.2 : Ενημερώνει τον χρήστη για τον σκοπό δημιουργίας της EurHEALTH

Στη σελίδα αυτή ο επισκέπτης ενημερώνεται για τους στόχους που έχει θέσει και θέλει να πραγματοποιήσει το EurHEALTHAGEING project. Οι στόχοι αυτοί είναι οι παρακάτω :

- 1) Να προσδιορίσει την ταυτότητα των γενετικών παραγόντων που επηρεάζουν το βάρος γέννησης, με τη διεξαγωγή γονιδιακής σάρωσης (GWAS) για αυτά τα χαρακτηριστικά.
- 2) Να προσδιορίσει χρησιμοποιώντας την αλληλουχία NextGen Sequencing (NGS), τις λειτουργικές ικανότητες των πρωτεϊνών και την επιρροή τους στο βάρος γέννησης, στις μετρήσεις ανάπτυξης και στους φαινότυπους γήρανσης.
- 3) Να προσδιορίσει την ταυτότητα του DNA.
- 4) Να ανακαλύψει νέους βιοδείκτες γήρανσης χρησιμοποιώντας μια διεπιστημονική προσέγγιση .
- 5) Να αναγνωρίσει τις μοριακές οδούς που εμπλέκονται στις μεταβλητές γεγονότων της πρώιμης ζωής, οι οποίες έχουν ως αποτέλεσμα την υγιή γήρανση.
- 6) Να αναγνωρίσει πτυχές στην πορεία της ζωής που οδηγούν στην υγιή γήρανση ενσωματώνοντας γεγονότα της πρώιμης ζωής.
- 7) Να παράγει μια εκτενή βάση δεδομένων για τις συσχετίσεις μεταξύ των γεγονότων της πρώιμης ζωής και των αποτελεσμάτων της γήρανσης, συνδέοντας δεδομένα για τον προσδιορισμό της ταυτότητας του DNA, επιτρέποντας έτσι τη μελλοντική σύγκριση με δεδομένα που προήλθαν από άλλες μελέτες. Τα αποτελέσματα αυτά θα μπορούν να χρησιμοποιηθούν στην πορεία σε δοκιμές για την υγιή γήρανση του πληθυσμού.

Ετικέτα Partners

Όταν ο επισκέπτης επιλέξει την ετικέτα Partners τότε του εμφανίζεται το παρακάτω παράθυρο :

Welcome to EurHEALTHAgeing Database Webpage

[Home](#) [Project Information](#) [Partners](#) [Login](#)

1) **King's College London** is one of the top 30 universities in the world (2011/2012 QS international world rankings), the Sunday Times 'University of the Year 2010/11' and the fourth oldest in England.

2) The **Department of Twin Research and Genetic Epidemiology** is the biggest UK adult twin registry of 12,000 twins used to study the genetic and environmental aetiology of age related complex traits and diseases. TwinsUK was set up by Professor Tim Spector, Consultant Rheumatologist in 1992 using grant funding from the Wellcome Trust and Arthritis Research Campaign (ARUK) and it is currently based at the Department of Twin Research, King's College London.

3) **Nordic Bioscience** discovers and develops innovative medical treatments to help millions of people around the world suffering from serious, chronic, degenerative diseases such as osteoporosis and osteoarthritis. Since its incorporation in 1989, Nordic Bioscience has strived to revolutionize in vitro diagnostics enabling the Company to discover and design innovative drug development programme.

4) The **MRC Lifecourse Epidemiology Unit (LEU)** was established in 2010, under the direction of Professor Cyrus Cooper, following a reconfiguration of the forerunning Epidemiology Resource Centre (ERC). The mission of the LEU is to provide a centre of excellence which uses epidemiological methods to promote human health.

5) The **University of Oulu** is an international research and innovation university engaged in multidisciplinary basic research and academic education. The University of Oulu is one of the largest universities in Finland with an exceptionally wide academic base. Internationally pioneering research is conducted as a collaboration of different disciplines.

6) The **Institute of Communications and Computer Systems (ICCS)** is a non profit Academic Research Body established in 1989 by the Ministry of Education in order to carry research and development activities in the fields of all diverse aspects of telecommunications , computer systems and techniques and their application in a variety of areas such as transceivers, radar and generally electromagnetic sensors, satellite and wireless communications, electromagnetic phenomena modelling, neural networks, systems, software and hardware engineering, Telematics and multimedia applications, Transport applications, Control systems, Robotics, Biomedical engineering, Electric power Renewable Energy Sources and Distributed Generation and management systems.

7) The **AIT Austrian Institute of Technology** is an Austrian research institute with a European format and focuses on the key infrastructure issues of the future. The AIT, which comprises five independent and performance-driven departments (Energy, Mobility, Health & Environment, Safety & Security and Foresight & Policy Development), works in close collaboration with industry and customers from public institutions, striving to increase their added value through innovation and new technologies.

8) Founded in 2001 by pioneers in qPCR, **TATAA Biocenter** is Europe's leading provider of gene expression profiling services and world's prime organizer of hands-on training in quantitative real-time PCR (qPCR) and other molecular techniques to quantify nucleic acids. The company offer comprises the entire field of qPCR services, including commissioned services and research projects, hands-on training, and highest quality products.

9) Located near Vienna, **Seibersdorf Laboratories GmbH** is a wholly-owned subsidiary of AIT Austrian Institute of Technology and comprises a selected team of long term experienced experts. The history of the company dates back to the 1950s. Under the brand name Seibersdorf Laboratories the team provides high-quality laboratory services as well as solution-driven Research & Development. Experts in the areas of ionizing radiation, radiation protection, electromagnetic compatibility, laser and optical radiation, radio-frequency engineering, chemical analytics and toxicology cultivate an environment of interdisciplinary cooperation and offer one-stop solutions.

Εικόνα 7.3 : Παρουσιάζει τους συνεργάτες του EurHEALTH project

Στη σελίδα αυτή ο επισκέπτης βλέπει ποιοι συνεργάζονται για τη δημιουργία και την υποστήριξη αυτού του project. Πρόκειται για μερικά μεγάλα πανεπιστήμια όπως είναι το **King's College London**, το **University of Oulu** και διάφορα ινστιτούτα όπως το **Institute of Communications and Computer Systems (ICCS)**, το **AIT Austrian Institute of Technology** κ.α.

Ετικέτα Login

Επιλέγοντας την ετικέτα Log In ο επισκέπτης μπορεί να έχει πρόσβαση στην οθόνη Login. Σε αυτή την οθόνη έχει τρεις επιλογές.

- 1) Να συνδεθεί στην ιστοσελίδα μας κάνοντας Login, πληκτρολογώντας όνομα χρήστη (username) και κωδικό πρόσβασης (password) εάν είναι ήδη εγγεγραμμένος .

Welcome to EurHEALTHAgeing Database Webpage

[Home](#) [Project Information](#) [Partners](#) [Login](#)

Username:

Password:

Click here to [Register](#) if you have not done so already.

[Forgot Password?](#)

Εικόνα 7.4 : Παράθυρο εισόδου (Login) στην εφαρμογή μας

- 2) Εάν δεν είναι εγγεγραμμένος, επιλέγει να κάνει εγγραφή (Register) και μόλις το επιλέξει μεταφέρεται σε άλλη οθόνη όπου και συμπληρώνει ορισμένα πεδία που του ζητούνται.

Welcome to EurHEALTHAgeing Database Webpage

[Home](#) [Project Information](#) [Partners](#) [Login](#)

First Name:

Last Name:

Email:

Organization:

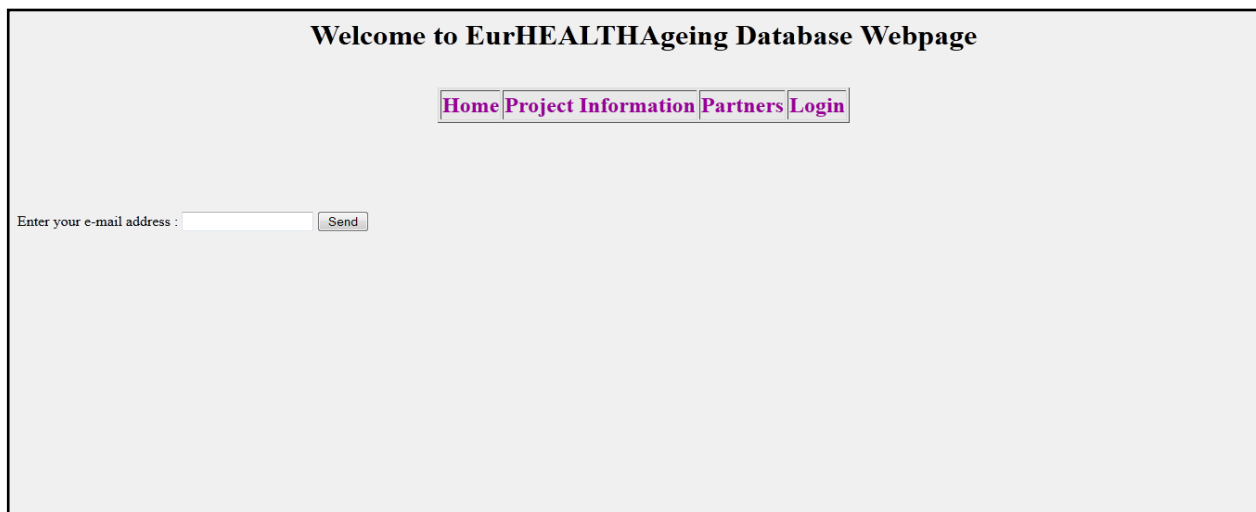
Username:

Password: (min 6 characters)

Εικόνα 7.5 : Σελίδα εγγραφής (Register) στην εφαρμογή μας

Στο σημείο αυτό έχω μεριμνήσει, ώστε για περιπτώσεις όπου ο χρήστης δεν συμπληρώσει όλα τα πεδία, ή όταν εισάγει λανθασμένο e-mail (το e-mail πρέπει απαραίτητα να περιλαμβάνει το χαρακτήρα @), ή e-mail το οποίο υπάρχει ήδη, να μην εκτελείτε το συγκεκριμένο ερώτημα (query) στη βάση δεδομένων μας. Στην περίπτωση που τα στοιχεία του χρήστη είναι σωστά, τότε εισάγονται και αποθηκεύονται στον πίνακα *users* της βάσης δεδομένων μας (*metabolites_second*).

- 3) Εάν είναι εγγεγραμμένος αλλά έχει ξεχάσει τον κωδικό πρόσβασης (password) επιλέγει το σημείο *Forgot Password*, όπου μεταφέρεται σε άλλο παράθυρο και εκεί του ζητείται να πληκτρολογήσει τη διεύθυνση του email του (e-mail address). Τότε αυτόματα στο email του στέλνεται μήνυμα, όπου του υπενθυμίζει τον κωδικό πρόσβασης (password). Έτσι μπορεί πλέον να κάνει *Login* για να περιηγηθεί στη σελίδα.



Εικόνα 7.6 : Σελίδα υπενθύμισης του κωδικού πρόσβασης του χρήστη

Πρωτόκολλο SMTP

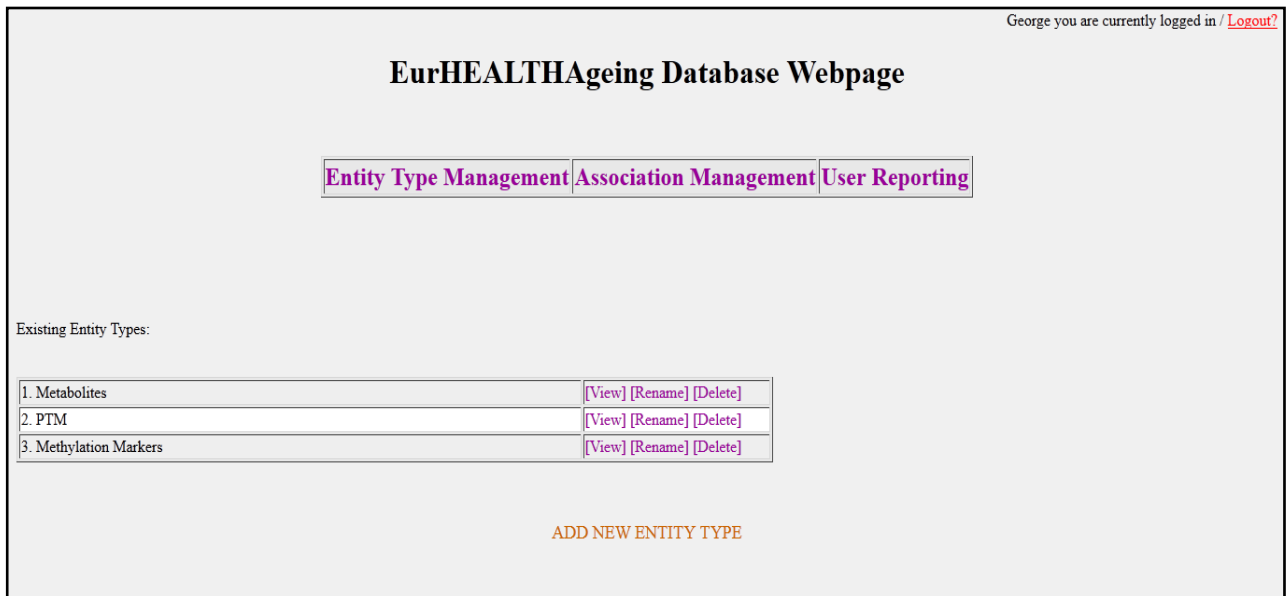
Η συγκεκριμένη λειτουργία που αναφέραμε παραπάνω βασίζεται στο πρωτόκολλο SMTP (Simple Mail Transfer Protocol) το οποίο έχει καθιερωθεί για την μετάδοση μηνυμάτων ηλεκτρονικού ταχυδρομείου στο Διαδίκτυο. Για την συγκεκριμένη εργασία, στο αρχείο `sendmail.ini` επιλέχτηκε ο SMTP Server: `smtp.gmail.com` και η πόρτα η οποία χρησιμοποιήθηκε για την επικοινωνία είναι η 465. Επίσης στο αρχείο `php.ini` επιλέξαμε `sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"`.

Το πρόγραμμα **Sendmail** ήταν ένα από τα πρώτα προγράμματα που υλοποίησε το SMTP, ενώ σήμερα υπάρχει μία πληθώρα τέτοιων προγραμμάτων όπως για παράδειγμα τα Postfix, gmail, Novell GroupWise, Exim, Novell NetMail και άλλα. Σε μία μέτρηση που έγινε το 2001 βρέθηκαν τουλάχιστον 50 προγράμματα τα οποία υλοποιούσαν το πρωτόκολλο SMTP είτε ως client (δηλαδή αποστολείς ηλεκτρονικών μηνυμάτων) είτε ως server (δηλαδή παραλήπτες ηλεκτρονικών μηνυμάτων).

Το αρχικό SMTP υποστήριζε κατά βάση μονάχα μηνύματα απλού κειμένου και όχι μετάδοση αρχείων (πχ. εικόνες, εκτελέσιμα, μουσική κοκ). Στην συνέχεια όμως αναπτύχθηκαν διάφορα standards που επέτρεπαν την εισαγωγή αρχείων στα ηλεκτρονικά μηνύματα. Ένα από αυτά τα standards είναι και το Multipurpose Internet Mail Extensions (MIME), το οποίο κωδικοποιεί τα αρχεία με τέτοιο τρόπο ούτως ώστε να μπορούν να μεταδοθούν σε απλά μηνύματα SMTP.

Πεδίο Entity Type Management

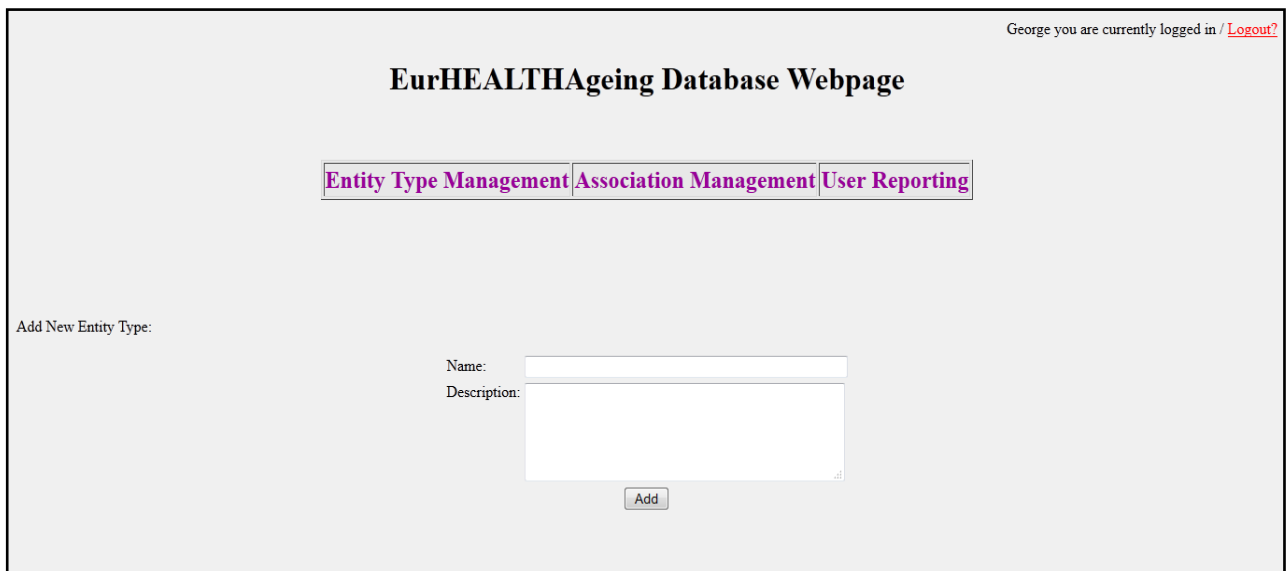
Το παράθυρο που αντικρίζει ο χρήστης όταν συνδεθεί επιτυχώς, κάνοντας login είναι το παρακάτω :



Εικόνα 7.7 : Σελίδα παρουσίασης των αποθηκευμένων Entity Types

Του εμφανίζεται μια λίστα με τα υπάρχοντα entity types (Metabolites, PTM, Methylation Markers). Εκεί έχει αρκετές επιλογές. Αρχικά μπορεί να προσθέσει ένα καινούργιο entity type αρκεί να πατήσει την επιλογή ADD NEW ENTITY TYPE.

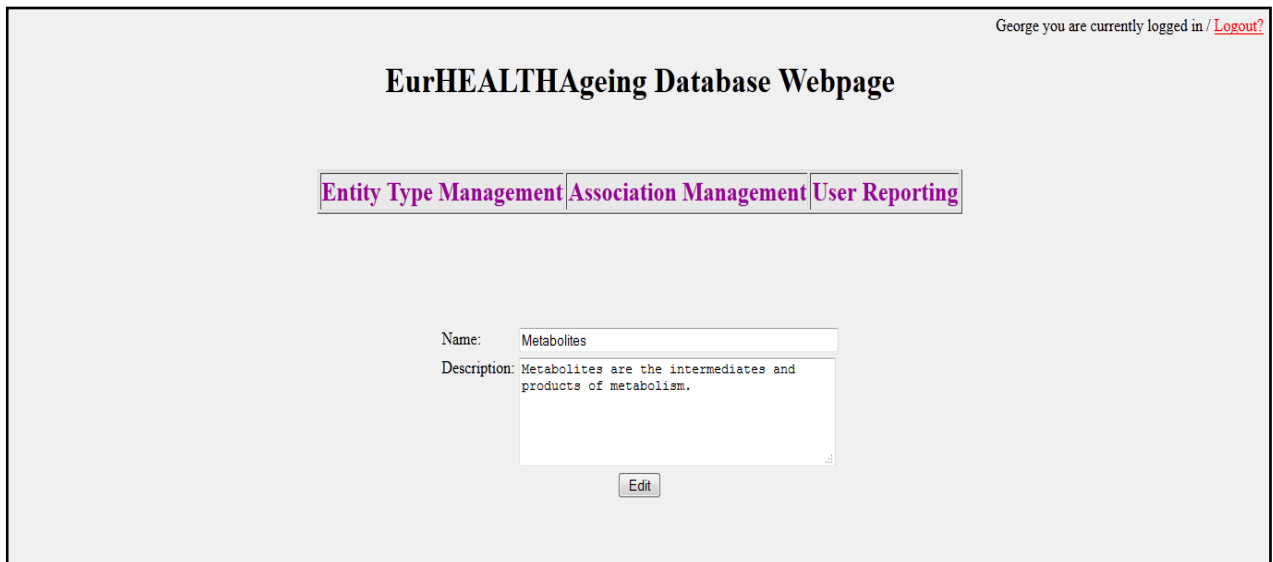
Αν προβεί σε αυτή την κίνηση του εμφανίζεται το παρακάτω παράθυρο όπου πρέπει να συμπληρώσει το όνομα του συγκεκριμένου entity type και να δώσει μια σύντομη περιγραφή για αυτό.



Εικόνα 7.8 : Σελίδα προσθήκης ενός καινούργιου Entity Type

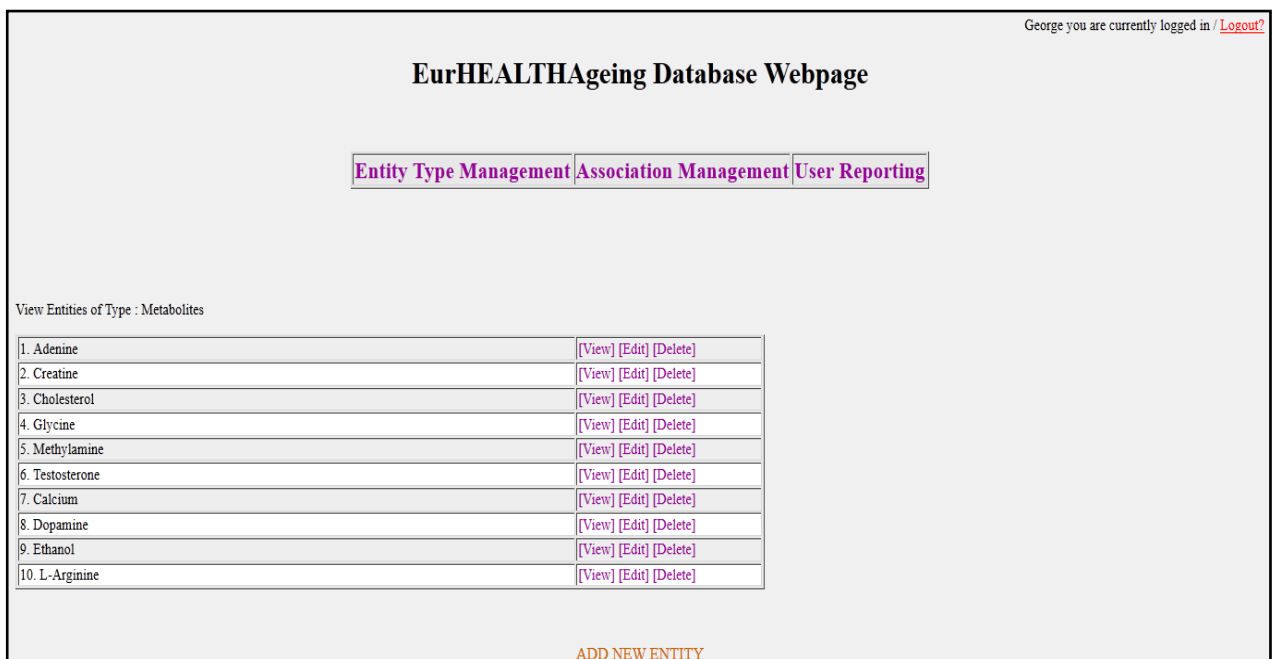
Αφότου συμπληρώσει τα δυο αυτά πεδία επιλέγει ADD και του εμφανίζεται μήνυμα επιτυχημένης προσθήκης (Successful entry). Τότε αυτόματα αυτό το καινούργιο Entity Type προστίθεται στη βάση δεδομένων μας metabolites_second.

Αν επιλέξει να διορθώσει κάποιο υπάρχων Entity Type, έστω το metabolites τότε του εμφανίζεται το παρακάτω παράθυρο.



Εικόνα 7.9 : Σελίδα διόρθωσης (edit) ενός υπάρχων Entity Type

Αν επιλέξει να διαγράψει (delete) κάποιο Entity Type, τότε αυτό αυτόματα διαγράφεται από τη βάση δεδομένων μας. Έπειτα μπορεί να δει τα entities, που περιλαμβάνει κάθε ένα από τα τρία υπάρχοντα entity types. Αν επιλέξει να δει τα entities που είναι αποθηκευμένα στο entity type metabolites, τότε του εμφανίζεται το παρακάτω παράθυρο.



Εικόνα 7.10 : Σελίδα παρουσίασης των Entities που ανήκουν στο Entity Type Metabolites

Αν επιλέξει να δει τα entities που είναι αποθηκευμένα στο entity type PTM του εμφανίζεται το παρακάτω παράθυρο.

George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage

[Entity Type Management](#)
[Association Management](#)
[User Reporting](#)

View Entities of Type : PTM

1. Acetylation	[View] [Edit] [Delete]
2. Allylsine	[View] [Edit] [Delete]
3. Biotin	[View] [Edit] [Delete]
4. Chromophore	[View] [Edit] [Delete]
5. Cyclopeptide	[View] [Edit] [Delete]
6. D-amino acid	[View] [Edit] [Delete]
7. Decarboxylation	[View] [Edit] [Delete]
8. Lipoprotein	[View] [Edit] [Delete]
9. Nitration	[View] [Edit] [Delete]
10. Pyridoxal phosphate	[View] [Edit] [Delete]

[ADD NEW ENTITY](#)

Εικόνα 7.11 : Σελίδα παρουσίασης των Entities που ανήκουν στο Entity Type PTM

Αν επιλέξει να δει τα entities που είναι αποθηκευμένα στο entity type Methylation Markers του εμφανίζεται το παρακάτω παράθυρο.

George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage

[Entity Type Management](#)
[Association Management](#)
[User Reporting](#)

View Entities of Type : Methylation Markers

1. bcell	[View] [Edit] [Delete]
2. h1_mesendoderm	[View] [Edit] [Delete]
3. h9_fibroblast	[View] [Edit] [Delete]
4. frontalcortex_alz	[View] [Edit] [Delete]
5. h1_npc	[View] [Edit] [Delete]
6. h1_mesenchymal	[View] [Edit] [Delete]
7. colon_cancer	[View] [Edit] [Delete]
8. foreskin_fibroblast	[View] [Edit] [Delete]
9. hmec	[View] [Edit] [Delete]
10. pbmc	[View] [Edit] [Delete]

[ADD NEW ENTITY](#)

Εικόνα 7.12 : Σελίδα παρουσίασης των Entities που ανήκουν στο Entity Type Methylation Markers

Αν ο χρήστης τώρα θέλει να προσθέσει κάποιο καινούργιο entity, τότε επιλέγει την ετικέτα ADD NEW ENTITY και του εμφανίζεται το παρακάτω παράθυρο. Στο παράθυρο αυτό εκτός από το όνομα του συγκεκριμένου entity μπορεί να προσθέσει και μια διεύθυνση (link) η οποία θα παρέχει περισσότερες πληροφορίες, όπως επίσης να ανεβάσει και κάποια εικόνα σχετική με το συγκεκριμένο entity.

George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage

[Entity Type Management](#)
[Association Management](#)
[User Reporting](#)

Add New Entity:

Name:

Description:

Links:

Image: No file selected.
(max size 1Mb)

Εικόνα 7.13 :Σελίδα προσθήκης ενός καινούργιου Entity

Αν επιλέξει να δει (view) κάποιο συγκεκριμένο Entity, έστω ότι επιλέγει την Adenine τότε του εμφανίζεται το παρακάτω παράθυρο. Σε περίπτωση που ο χρήστης έχει ανεβάσει κάποια εικόνα για το συγκεκριμένο entity, τότε εκείνη εμφανίζεται στο πεδίο image. Αντίστοιχα, ένα ο χρήστης έχει εισάγει κάποια διεύθυνση, εκείνη θα εμφανίζεται στο πεδίο Link.

George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage

[Entity Type Management](#)
[Association Management](#)
[User Reporting](#)

Entity Name : Adenine

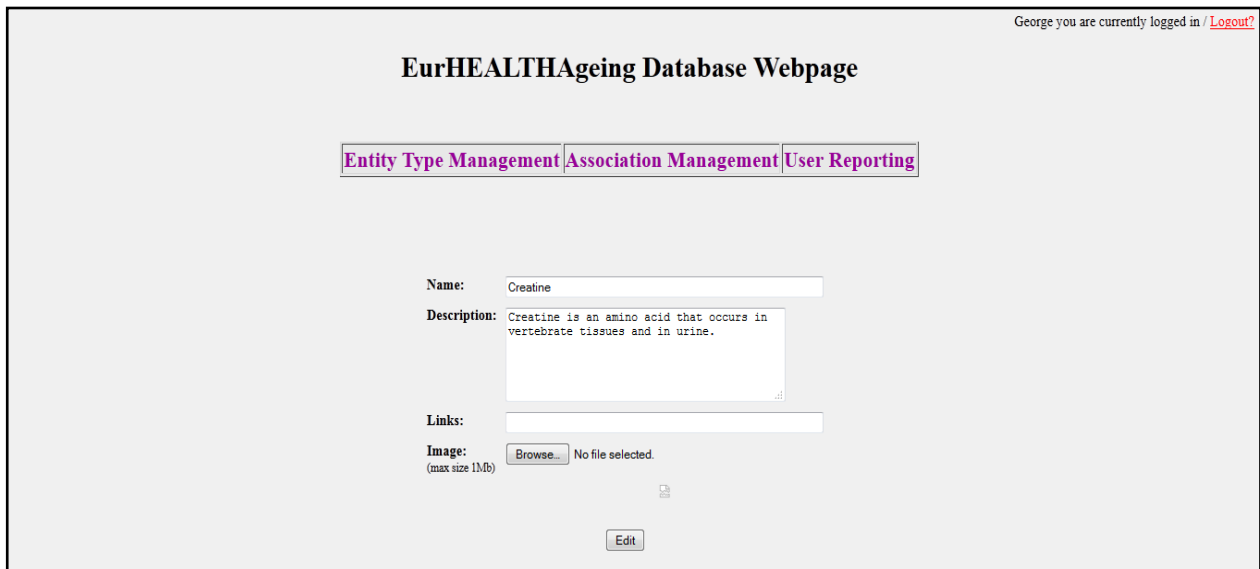
Description : Adenine is a purine base. Adenine is found in both DNA and RNA.

Links :

Image :

Εικόνα 7.14 : Σελίδα προβολής στοιχείων ενός Entity.

Αν επιλέξει να διορθώσει (edit) κάποιο Entity , έστω την Creatine τότε του εμφανίζεται το παρακάτω παράθυρο.

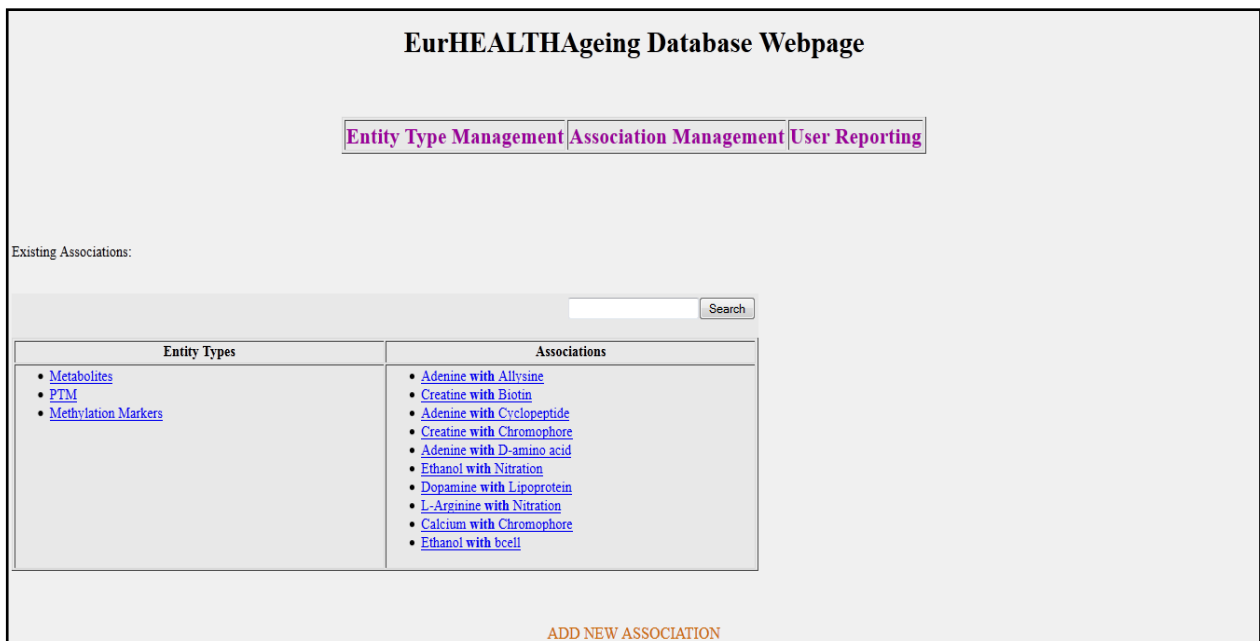


Εικόνα 7:15 : Σελίδα διόρθωσης (edit) στοιχείων ενός Entity.

Αν ο χρήστης επιλέξει να διαγράψει κάποιο Entity τότε αυτό αυτόματα διαγράφεται από βάση δεδομένων μας όπως συμβαίνει και με τα Entity Types που ανέφερα παραπάνω.

Ετικέτα association management

Όταν ο χρήστης επιλέξει την ετικέτα association management, τότε μεταφέρεται στο παρακάτω παράθυρο. Εκεί του εμφανίζονται τα υπάρχοντα associations και εκτός από τη δυνατότητα που έχει να προσθέσει ένα καινούργιο association μπορεί να κάνει αναζήτηση (search), για να του εμφανιστούν συγκεκριμένες πληροφορίες που θέλει.



Εικόνα 7.16 :Σελίδα εμφάνισης των υπάρχόντων associations

Αν ο χρήστης θέλει να διορθώσει ένα υπάρχων association ή να το διαγράψει , το επιλέγει από την προηγούμενη σελίδα και τότε του εμφανίζεται το παρακάτω παράθυρο.

George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage

[Entity Type Management](#) | [Association Management](#) | [User Reporting](#)

Association Adenine with Allylsine

Description	
-------------	--

[Edit](#)
[Delete](#)

Εικόνα 7.17 : Σελίδα διόρθωσης ή διαγραφής ενός υπάρχοντος Association

Αν επιλέξει να προσθέσει ένα καινούργιο association τότε μεταφέρεται στο παρακάτω παράθυρο. Εκεί σαν πρώτο βήμα του ζητείται να συνδυάσει δυο Entity Types για να προχωρήσει στο επόμενο βήμα. Έστω ότι επιλέγει να συνδυάσει τα Metabolites με τα PTM.

George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage

[Entity Type Management](#) | [Association Management](#) | [User Reporting](#)

Add new association Step 1/2:

Choose entity type 1	Choose entity type 2
Metabolites ▾	PTM ▾

[Next](#)

Εικόνα 7.18 : Σελίδα επιλογής δυο Entity Types για τη δημιουργία ενός καινούργιου association

Στο επόμενο παράθυρο, δεύτερο βήμα για την προσθήκη ενός καινούργιου association, ζητείται από το χρήστη να επιλέξει δυο Entities, ένα για το κάθε ένα από τα δυο Entity Types που είχε επιλέξει προηγουμένως.

George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage

Entity Type Management
Association Management
User Reporting

Add new association Step 2/2:

Choose entity of type Metabolites	Choose entity of type PTM
Adenine ▾	Acetylation ▾
Give description for association	
<input type="button" value="Create Association"/>	

Εικόνα7.19 : Σελίδα επιλογής δυο Entities για τη δημιουργία ενός καινούργιου association

Ετικέτα User Reporting

Αν ο χρήστης επιλέξει την ετικέτα User Reporting τότε μεταφέρεται στο παρακάτω παράθυρο. Εκεί του ζητείται να επιλέξει ποιο από τα δυο γραφήματα θέλει να δει.

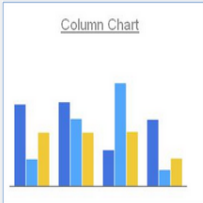
George you are currently logged in / [Logout?](#)

EurHEALTHAgeing Database Webpage


Entity Type Management
Association Management
User Reporting

Select the type of Chart to view

Column Chart



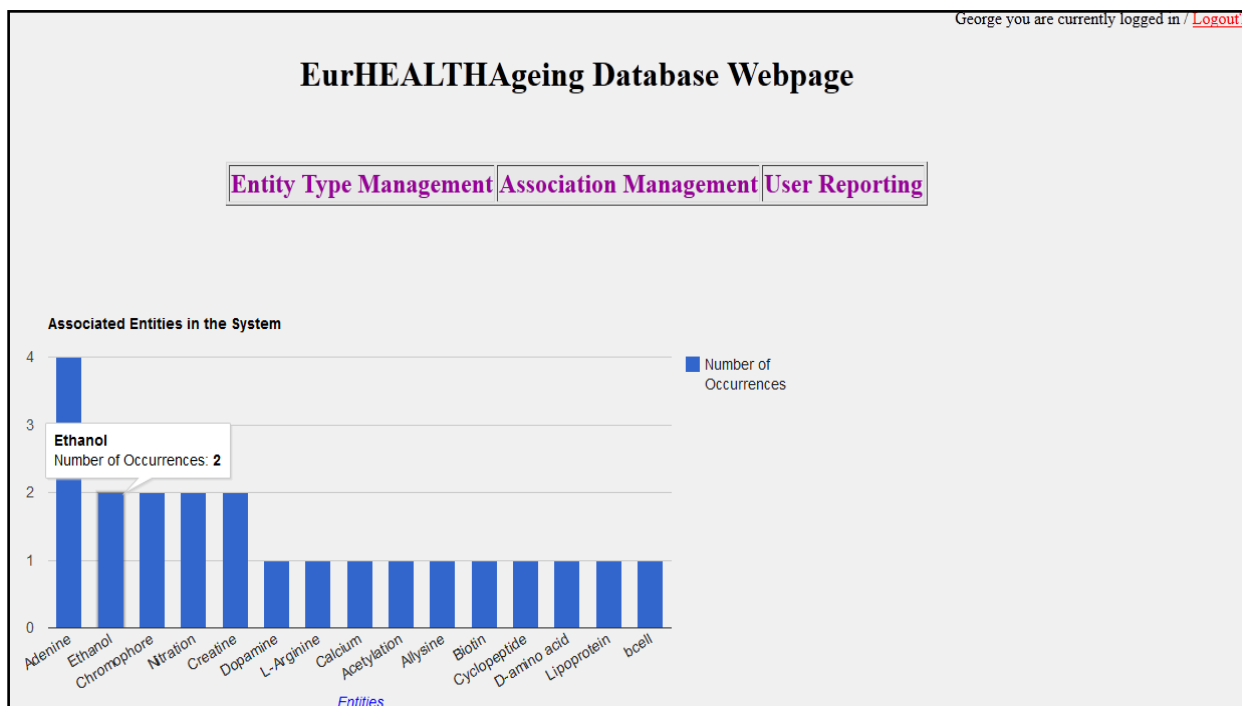
Pie Chart



Εικόνα 7.20 : Σελίδα επιλογής γραφήματος

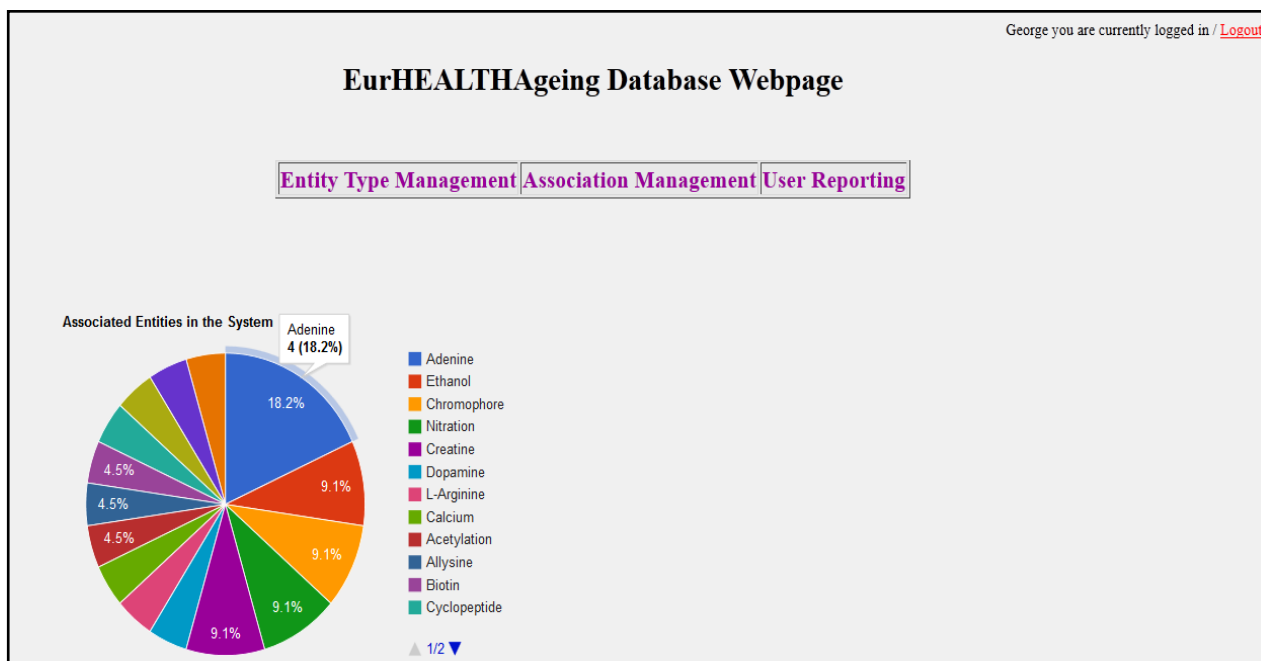
Αν επιλέξει το Column Chart τότε του εμφανίζεται το παρακάτω παράθυρο. Περιλαμβάνει γράφημα σε άξονες χ,ψ , όπου ο άξονας των χ απεικονίζει τα entities και ο άξονας των ψ τον συνολικό αριθμό που εμφανίζεται κάθε entity, στα υπάρχοντα associations. Επίσης όταν ο χρήστης τοποθετεί τον κέρσορα

πάνω σε μια μπάρα, στο παράδειγμα μας επιλέγω την Ethanol ,του αναφέρει τα παραπάνω στοιχεία χωρίς να χρειάζεται να κοιτάξει το σύνολο του γραφήματος.



Εικόνα 7.21 : Προβολή γραφήματος-Column Chart

Αν επιλέξει το Pie Chart του εμφανίζεται το παρακάτω παράθυρο. Εδώ πρόκειται για ένα κυκλικό γράφημα, όπου απεικονίζει το ποσοστό που συμμετέχει κάθε entity στο σύνολο των associations, με ξεχωριστό χρώμα για το κάθε ένα entity. Επίσης όταν ο χρήστης τοποθετήσει τον κέρσορα πάνω σε ένα entity, στο παράδειγμα μας επιλέγω την Adenine, εκτός από το ποσοστό του εμφανίζει και τον συνολικό αριθμό που συναντάται στο σύνολο των associations.



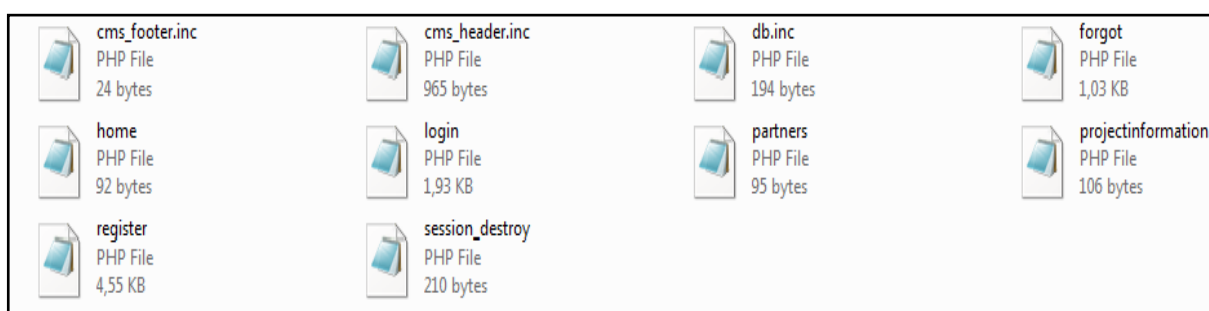
Εικόνα 7.22 : Προβολή γραφήματος- Pie Chart

Παράρτημα

Δομή καταλόγων

Αρχικά δημιούργησα έναν κατάλογο που τον ονόμασα **final project**, όπου εκεί περιέχονται όλα τα script με τα οποία δημιούργησα τη δυναμική ιστοσελίδα. Ο κατάλογος αυτός περιέχει 4 φακέλους τον homepage, τον entity_type_management, τον associations και τον graphs. Ο κάθε ένας από τους τέσσερις αυτούς φακέλους περιέχει τα αντίστοιχα scripts, με τα οποία έφτιαξα ξεχωριστά κάθε τμήμα της ιστοσελίδας.

- 1) Το πρώτο τμήμα της ιστοσελίδας δημιουργήθηκε με τα script που περιέχονται στον φάκελο **homepage**.



Κώδικας

K-1.1 Κώδικας cms_header.inc

Το βασικό script με τα οποίο έφτιαξα την αρχική μορφή του site είναι το **cms_header.inc**, όπου παραθέτω τον κώδικα παρακάτω:

```
<?php session_start(); ?>

<html>

<head>

<title> CMS </title>

<style type="text/css">

td {vertical-align: top;}

</style>

</head>

<body bgcolor="#F0F0F0">

<h1 align="center"> Welcome to EurHEALTHAgeing Database Webpage </h1>

<br/>

<?php
```

```

if (isset($_SESSION['name'])) {
echo '<p> You are currently logged in as: ' . $_SESSION['name'] . ' </p>';
}
?>
</div>
<style>
table,th,td
th
{
background-color:#E8E8E8;
}
</style>
<table border = "1" align="center">
<th><a href="home.php" style="text-decoration:none; color: #990099"><font size="5">Home</font></a></th>
<th><a href="projectinformation.php" style="text-decoration:none; color: #990099"><font size="5">Project
Information</font></a></th>
<th><a href="partners.php" style="text-decoration:none; color: #990099"><font size="5">Partners</font></a></th>
<th><a href="login.php" style="text-decoration:none; color: #990099"><font size="5">Login</font></a></th>
</table>
<br/><br/><br/>

```

Κλάση session

Αξιοπρόσεκτη είναι η συνάρτηση **session_start** της PHP στην αρχή του κώδικα, η οποία χρησιμοποιείται για να ξεκινήσει μια συνεδρία και γενικά τα sessions λειτουργούν με το να δημιουργούν ένα μοναδικό αναγνωριστικό ID για κάθε επισκέπτη . Η σημασία των sessions χρήζει ιδιαίτερης σημασίας διότι αν δεν υπήρχαν θα δημιουργείτο το παρακάτω πρόβλημα. Ο browser που χρησιμοποιεί ο χρήστης για να προσπελάσει την ιστοσελίδα δεν διατηρεί την πληροφορία που του εισάγει (στην περίπτωση μας το username και το password) από σελίδα σε σελίδα, με αποτέλεσμα όταν θα ήθελε να περιηγηθεί σε μια άλλη ενότητα, να μην γνώριζε ο browser ότι έχει πραγματοποιηθεί η σύνδεση και έτσι να χρειάζοταν να πληκτρολογεί κάθε φορά τα στοιχεία του. Έτσι λοιπόν τη λύση τη δίνουν τα sessions τα οποία κάθε φορά που ένας χρήστης θα επισκέπτεται την ιστοσελίδα μας, ο server δημιουργεί αυτόματα ένα μοναδικό αναγνωριστικό ID το οποίο αποθηκεύεται στον browser του χρήστη. Το κλείσιμο όμως μιας session δεν γίνεται αυτόματα, επειδή είναι δύσκολο για το σύστημα να καταλάβει πότε ο χρήστης αποχωρεί. Αυτό το πραγματοποιήσαμε με την βοήθεια της εντολής `session_destroy()`.

K-1.2 Κώδικας home.php

Το πρώτο παράθυρο (Εικόνα 7.1) που αντικρίζει ο χρήστης όταν επισκεφτεί την ιστοσελίδα μας δημιουργήθηκε με το script home και παρακάτω παραθέτω τον κώδικα του.

```
<?php include 'cms_header.inc.php'; ?>

<style>

p {

    text-indent: 50px;

}

</style>

<p style="font-family:arial; font-size:17px; padding-left:140px;" align="justify">

EurHEALTHAgeing is a <b>multidisciplinary project</b> linking studies of <b>early developmental processes</b> with those on <b>longevity</b> and<br>

<b>ageing</b> by combining use of cutting edge technologies- <b>post-translational modifications, DNA methylation, metabolomics</b> and<br>

<b>genomics</b> with unique cohorts with multiple ageing phenotypes. We will use <b>four key European study cohorts</b> with rich early life data<br>

and age-related health outcomes. We will explore a unique twin dataset, two birth cohorts with extensive maternal, pre- and peri-natal<br>

data and a population based cohort of subjects in their 60s-70s. By analysing specific age related metabolomics, epigenetic (methylation<br>

and post-translational modification) data and correlating it to early life events, genetics and ageing outcomes we will provide <b>major</b>

insights into developmental processes that influence longevity and ageing</b>. By focussing on molecular mechanisms involved in<br>

both early life events and ageing we will <b>find genes and metabolic and lifecourse pathways that are relevant in both early</b>

development and adult life</b>.</br></p>

<p style="font-family:arial; font-size:17px; padding-left:140px;" align="justify">

This will also allow us to develop <b>biomarkers</b> of ageing that reflect the role of early development on ageing; potentially <b>identifying</b>

<br>

pathways for therapeutic intervention when the process is still reversible</b>.</br>

</p>

</p>

<?php include 'cms_footer.inc.php'; ?>
```

K-1.3 Κώδικας projectinformation.php

Το παράθυρο που αντικρίζει ο χρήστης όταν επιλέξει την ετικέτα Project Information (Εικόνα 7.2) δημιουργήθηκε με το script projectinformation και παρακάτω παραθέτω τον κώδικα του.

```
<?php include 'cms_header.inc.php'; ?>

<style>

p {

    text-indent: 50px;

}


```

</style>

<p style="font-family:arial; font-size:17px; " align="center">Objectives of EurHEALTHAgeing</p>

</br>

<p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

1) To identify genetic factors influencing birthweight (measures of size/shape: birthweight and length relative to gestational age, ponderal index) and</br>

other early postnatal determinants of healthy ageing (e.g. growth patterns/velocity) by carrying out a genomewide association scan (GWAS) on these traits.</br>

</p>

<p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

2) To identify using NextGen Sequencing (NGS)- rarer variants i.e. single base pair changes, insertions, deletions and breakpoints of rarer copy number</br>

variations (CNVs), and assess their potential to influence transcripts' and proteins' functional capabilities and their influence on birthweight and growth

measures and ageing phenotypes</br>

</p>

<p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

3) To identify DNA methylation, post-translational modifications and metabolite profiles that correlate with prenatal/early post natal determinants of </br>

health-span (e.g. birthweight, weight and weight-gain of mother, smoking and alcohol drinking of mother, maternal pregnancy related diseases and complications).</br>

</p>

<p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

4) To discover novel biomarkers of ageing using a multi-disciplinary approach incorporating pathway analysis, biochemistry, genetics, metabolomics,</br>

epidemiology, DNA methylation and bioinformatics.</br>

</p>

<p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

5) To identify, using pathway analysis, the molecular pathways involved early life event variables that result in healthy ageing (musculoskeletal, </br>

cardiovascular, metabolic) by incorporating early life events, methylation, post-translational modification, metabolomic and genotypic data.</br>

</p>

<p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

6) To identify lifecourse trajectories leading to healthy aging by incorporating early life events, methylation, post-translational modification,</br>

metabolomic and genotypic data.</br>

</p>

<p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

7)To generate an extensive database of the correlations between early life events and ageing outcomes linking datapoints to methylation, metabolomic,</br>

genetic and post-translational modification (PTM) results, allowing future comparison with data derived from other studies. Results can then be employed</br>

in tests for healthy ageing.</br>

</p>

<?php include 'cms_footer.inc.php'; ?>

K-1.4 Κώδικας partners.php

Το παράθυρο που αντικρίζει ο χρήστης όταν επιλέξει την ετικέτα Partners (Εικόνα 7.3) δημιουργήθηκε με το script partners και παρακάτω παραθέτω τον κώδικα του.

<?php include 'cms_header.inc.php'; ?>

<style>

p {

text-indent: 50px;

}

</style>

<p><p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

1) King's College London is one of the top 30 universities in the world (2011/2012 QS international world rankings), the Sunday Times 'University</br>

of the Year 2010/11' and the fourth oldest in England.</br>

</p>

<p><p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

2) The Department of Twin Research and Genetic Epidemiology is the biggest UK adult twin registry of 12,000 twins used to study the genetic and</br>

environmental aetiology of age related complex traits and diseases. TwinsUK was set up by Professor Tim Spector, Consultant Rheumatologist in 1992</br>

using grant funding from the Wellcome Trust and Arthritis Research Campaign (ARUK) and it is currently based at the Department of Twin Research,</br>

King's College London.</br>

</p>

<p><p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

3) Nordic Bioscience discovers and develops innovative medical treatments to help millions of people around the world suffering from serious, chronic,</br>

degenerative diseases such as osteoporosis and osteoarthritis. Since its incorporation in 1989, Nordic Bioscience has strived to revolutionize in vitro</br>

diagnostics enabling the Company to discover and design innovative drug development programme.</br>

</p>

<p><p style="font-family:arial; font-size:17px; padding-left:90px;" align="justify">

4) The MRC Lifecourse Epidemiology Unit (LEU) was established in 2010, under the direction of Professor Cyrus Cooper, following a reconfiguration of the</br>

forerunning Epidemiology Resource Centre (ERC). The mission of the LEU is to provide a centre of excellence which uses epidemiological methods to promote

human health.

5) The **University of Oulu** is an international research and innovation university engaged in multidisciplinary basic research and academic education.

The University of Oulu is one of the largest universities in Finland with an exceptionally wide academic base. Internationally pioneering research is

conducted as a collaboration of different disciplines.

6) The **Institute of Communications and Computer Systems** (ICCS) is a non profit Academic Research Body established in 1989 by the Ministry of

Education in order to carry research and development activities in the fields of all diverse aspects of telecommunications , computer systems and

techniques and their application in a variety of areas such as transceivers, radar and generally electromagnetic sensors, satellite and wireless

communications, electromagnetic phenomena modelling, neural networks, systems, software and hardware engineering, Telematics and multimedia applications,

Transport applications, Control systems, Robotics, Biomedical engineering, Electric power Renewable Energy Sources and Distributed Generation and

management systems.

7) The **AIT Austrian Institute of Technology** is an Austrian research institute with a European format and focuses on the key infrastructure issues of the

future. The AIT, which comprises five independent and performance-driven departments (Energy, Mobility, Health & Environment, Safety & Security and

Foresight & Policy Development), works in close collaboration with industry and customers from public institutions, striving to increase their added

value through innovation and new technologies.

8) Founded in 2001 by pioneers in qPCR, **TATAA Biocenter** is Europe's leading provider of gene expression profiling services and world's

organizer of hands-on training in quantitative real-time PCR (qPCR) and other molecular techniques to quantify nucleic acids. The company offer

comprises the entire field of qPCR services, including commissioned services and research projects, hands-on training, and highest quality products.

9) Located near Vienna, **Seibersdorf Laboratories GmbH** is a wholly-owned subsidiary of AIT Austrian Institute of Technology and comprises a

selected team of long term experienced experts. The history of the company dates back to the 1950s. Under the brand name Seibersdorf Laboratories the team

provides high-quality laboratory services as well as solution-driven Research & Development. Experts in the areas of ionizing radiation, radiation

protection, electromagnetic compatibility, laser and optical radiation, radio-frequency engineering, chemical analytics and toxicology cultivate an

environment of interdisciplinary cooperation and offer one-stop solutions.

```
<?php include 'cms_footer.inc.php'; ?>
```

K-1.5Κώδικας login.php

Στη συνέχεια όταν ο επισκέπτης επιλέξει να κάνει Login (Εικόνα 7.4) , το script που το πραγματοποιεί αυτό είναι το Login και ο κώδικας είναι ο παρακάτω:

```
<?php include 'cms_header.inc.php'; ?>

<?php
include 'db.inc.php';

// filter incoming values
$username = (isset($_POST['username'])) ? trim($_POST['username']) : '';
$password = (isset($_POST['password'])) ? $_POST['password'] : '';
$redirect = (isset($_REQUEST['redirect'])) ? $_REQUEST['redirect'] :
'main.php';

if (isset($_POST['submit'])) {
$query = 'SELECT username FROM user WHERE ' .
'username = "' . mysql_real_escape_string($username, $db) . "' AND ' .
'password = "' . mysql_real_escape_string($password, $db) . "'";
$result = mysql_query($query, $db) or die(mysql_error($db));
if (mysql_num_rows($result) > 0) {
$_SESSION['username'] = $username;
$_SESSION['logged'] = 1;
header ('Location: /entity_type_management/tab1.php');
die();
echo '<p> You will be redirected to your original page request.</p>';
echo '<p> If your browser doesn\'t redirect you properly
automatically, ' .
'<a href="' . $redirect . "' > click here </a> . </p>';
die();
} else {
```

```

$error = '

<strong> You have supplied an invalid username and/or ' .
'password!</strong></p>';
}
}
?>
<html>
<head>
<title> Login </title>
</head>
<body>
<?php
if (isset($error)) {
echo $error;
}
?>
<form action="login.php" method="post">
<table style = "background-color:#F0F0F0;">
<tr>
<td> Username: </td>
<td><input type="text" name="username" maxlength="20" size="20"
value="<?php echo $username; ?>"/></td>
</tr><tr>
<td> Password: </td>
<td><input type="password" name="password" maxlength="20" size="20"
value="<?php echo $password; ?>"/></td>
</tr><tr>
<td></td>
<td>
<input type="hidden" name="redirect" value="<?php echo $redirect ?>"/>
<input type="submit" name="submit" value="Login"/>
</tr>
</table>
</form>
<br/>
<p> Click here to <a href="register.php">Register</a> if you have not done so already.</p>
<br/>


```



```
<p align="center"><a href="forgot.php">Forgot Password?</a></p>
```

```
</body>
```

```
</html>
```

```
<?php include 'cms_footer.inc.php'; ?>
```

Το αρχείο **db.inc.php** που αναφέρεται στην τρίτη γραμμή του παραπάνω script περιλαμβάνει τον εξής κώδικα :

```
$db = mysql_connect('localhost', 'admin', 'admin') or  
die ('Unable to connect. Check your connection parameters.');
```

```
mysql_select_db('metabolites_second', $db) or die(mysql_error($db));
```

Με τον παραπάνω κώδικα πραγματοποιείται η σύνδεση της PHP με τη MySQL και τη βάση δεδομένων μας `metabolites_second`. Είναι μια τεχνική για να μην γράφω συνέχεια αυτές τις γραμμές κώδικα στα περισσότερα scripts όπου χρειάζονται ,να τις κάνω include, που σημαίνει ότι περιλαμβάνονται και αυτές απλά είναι αποθηκευμένες σε ξεχωριστό αρχείο.

K-1.6 Κώδικας register.php

Το script που εμφανίζει το παράθυρο εγγραφής (Εικόνα 7.5) στην ιστοσελίδα μας είναι το register. Παρακάτω παραθέτω τον κώδικα του.

```
<?php include 'cms_header.inc.php'; ?>
```

```
<?php
```

```
$db = mysql_connect('localhost', 'admin', 'admin') or
```

```
die ('Unable to connect. Check your connection parameters.');
```

```
mysql_select_db('metabolites_second', $db) or die(mysql_error($db));
```

```
// filter incoming values
```

```
$username = (isset($_POST['username'])) ? trim($_POST['username']) : '';
```

```
$password = (isset($_POST['password'])) ? $_POST['password'] : '';
```

```
$first_name = (isset($_POST['first_name'])) ? trim($_POST['first_name']) : '';
```

```
$last_name = (isset($_POST['last_name'])) ? trim($_POST['last_name']) : '';
```

```
$email = (isset($_POST['email'])) ? trim($_POST['email']) : '';
```

```
$organization = (isset($_POST['organization'])) ? trim($_POST['organization']) : '';
```

```
if (isset($_POST['submit']) && $_POST['submit'] == 'Register') {
```

```
$errors = array();
```

```
// make sure mandatory fields have been entered
```

```
if (empty($username)) {
```

```
$errors[] = 'Username cannot be blank.';
```

```
}
```

```
// check if username already is registered
```

```
$query = 'SELECT username FROM user WHERE username = "' . $username . "'";
```

```

$result = mysql_query($query, $db) or die(mysql_error());

if (mysql_num_rows($result) > 0) {

$errors[] = 'Username ' . $username . ' is already registered.';

$username = '';

}

mysql_free_result($result);

if (empty($password)) {

$errors[] = 'Password cannot be blank.';

}

if (empty($first_name)) {

$errors[] = 'First name cannot be blank.';

}

if (empty($last_name)) {

$errors[] = 'Last name cannot be blank.';

}

if (empty($email)) {

$errors[] = 'Email address cannot be blank.';

}

if (empty($organization)) {

$errors[] = 'Organization cannot be blank.';

}

if(!filter_var($email, FILTER_VALIDATE_EMAIL))

{

$errors[] = 'Invalid Email address.';

}

if (strlen($password)<6) {

$errors[] = 'Password is less than 6 characters.';

}

if (count($errors) > 0) {

echo '<p><strong style="color:#FF000;"> Unable to process your ' . ' registration.</strong></p>';

echo '<p>Please fix the following:</p>';

echo '<ul>';

foreach ($errors as $error) {

echo '<li>' . $error . '</li>';

}

echo '</ul>';

```

```

} else {

// No errors so enter the information into the database.

$query = 'INSERT INTO user

(id, first_name, last_name, email, organization, username, password)

VALUES

(NULL,

"' . mysql_real_escape_string($first_name, $db) . "',

"' . mysql_real_escape_string($last_name, $db) . "',

"' . mysql_real_escape_string($email, $db) . "',

"' . mysql_real_escape_string($organization, $db) . "',

"' . mysql_real_escape_string($username, $db) . "',

"' . mysql_real_escape_string($password, $db) . "')';

$result = mysql_query($query, $db) or die(mysql_error());

$user_id = mysql_insert_id($db);

$_SESSION['logged'] = 1;

$_SESSION['username'] = $username;

header('Refresh: 2; URL=login.php');

?>

<html>

<head>

<title> Register </title>

</head>

<body>

<p><strong>Thank you <?php echo $username; ?> for registering!</strong></p>

<p> Your registration is complete! If your browser doesn't redirect properly after 5 seconds,

<a href="main.php">click here</a>.</p>

</body>

</html>

<?php

    die();

}

}

?>

<html>

<head>

<title>Register</title>

```

```

<style type="text/css">
td { vertical-align: top; }
</style>
</head>
<body>
<form action="register.php" method="post">
<table cellspacing="10px" style = "background-color:#F0F0F0;">
<tr>
<td><label for="first_name">First Name:</label></td>
<td><input type="text" name="first_name" id="first_name" size="25"
maxlength="20" value="<?php echo $first_name; ?>"/></td>
</tr><tr>
<td><label for="last_name">Last Name:</label></td>
<td><input type="text" name="last_name" id="last_name" size="25"
maxlength="20" value="<?php echo $last_name; ?>"/></td>
</tr><tr>
<td><label for="email">Email:</label></td>
<td><input type="text" name="email" id="email" size="25"
maxlength="25" value="<?php echo $email; ?>"/></td>
</tr><tr>
<td><label for="organization">Organization:</label></td>
<td><input type="text" name="organization" id="organization" size="25"
maxlength="20" value="<?php echo $organization; ?>"/></td>
</tr><tr>
<td><label for="username">Username:</label></td>
<td><input type="text" name="username" id="username" size="25"
maxlength="20" value="<?php echo $username; ?>"/></td>
</tr><tr>
<td><label for="password">Password:</label></td>
<td><input type="password" name="password" id="password" size="25"
maxlength="20" value="<?php echo $password; ?>"/></td>
<td><small>(min 6 characters)</small></td>
</tr><tr>
<td></td>
<td><input type="submit" name="submit" value="Register"/></td>
</tr>

```

```

</table>

</form>

</body>

</html>

<?php include 'cms_footer.inc.php'; ?>

```

K-1.7 Κώδικας forgot.php

Υπάρχει πιθανότητα ο χρήστης να θέλει να περιηγηθεί στην ιστοσελίδα , να έχει κάνει παλιότερα εγγραφή, αλλά να έχει ξεχάσει τον κωδικό πρόσβασης. Τότε στην περίπτωση αυτή η ιστοσελίδα του παρέχει τη δυνατότητα, πληκτρολογώντας το username του, να του στείλει στο email του τον κωδικό πρόσβασης. Το script που το πραγματοποιεί αυτή τη διαδικασία και εμφανίζει την Εικόνα 7.6 στην ιστοσελίδα μας είναι το forgot και ο κώδικας είναι ο παρακάτω :

```

<?php include 'cms_header.inc.php'; ?>

<?php

error_reporting(0);

$email=$_POST['email'];

if($_POST['submit']=='Send')

{

$db = mysql_connect('localhost', 'admin', 'admin') or

die ('Unable to connect. Check your connection parameters.');
```

mysql_select_db('metabolites_second', \$db) or die(mysql_error(\$db));

```

$query="select * from user where email='$email'";

$result=mysql_query($query) or die(error);

if(mysql_num_rows($result))

{

$row = mysql_fetch_assoc($result);

//Create message to sent

$message = ' Dear ' . $row['username'];

$message .= ' your password is: ' . $row['password'];

$code=rand(100,999);

mail($email, " EurHEALTHAgeingDatabase - Password Reminder", $message);

echo '<b>Email sent. Check your Inbox</b>';

header('Refresh: 2; URL=login.php');

}

else

{

echo "No user exist with this email id";

```

```

}
}
?>
<form action="forgot.php" method="post">
<br/><br/>
Enter your e-mail address : <input type="text" name="email">
<input type="submit" name="submit" value="Send">
</form>
<?php include 'cms_footer.inc.php'; ?>

```

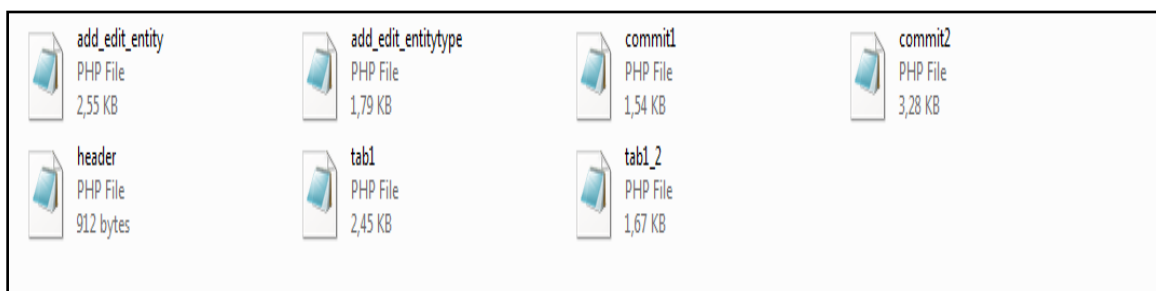
Αξιοσημείωτη στον παραπάνω κώδικα forgot.php, είναι η χρήση της ενδογενούς συνάρτησης **mail()** της PHP για την αποστολή των στοιχείων του χρήστη.

```

$message = ' Dear ' . $row['username'];
$message .= ' your password is: ' . $row['password'];
$code=rand(100,999);
mail($email, " EurHEALTHAgeingDatabase - Password Reminder", $message);

```

2) Το δεύτερο μέρος της ιστοσελίδας δημιουργήθηκε με τα scripts που περιέχονται στον φάκελο **entity_type_management**.



K-1.8 Κώδικας header.php

Η εμφάνιση των τριών νέων ετικετών (Entity Type Management, Association Management, User Reporting) που εμφανίζονται στον χρήστη όταν κάνει Login στην ιστοσελίδα μας, δημιουργήθηκε με το script header και παρακάτω παραθέτω τον κώδικα του.

```

<html>
<head>
<title> CMS </title>
<style type="text/css">

```

```

td {vertical-align: top;}

</style>

</head>

<body bgcolor="#F0F0F0">

<table style="width:100%;">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

<br/><br/>

<?php

if (isset($_SESSION['name'])) {

echo '<p> You are currently logged in as: ' . $_SESSION['name'] . ' </p>';

}

?>

</div>

<style>

table,th,td

th

{

background-color:#E8E8E8;

}

</style>

<table border = "1" align="center">

<th><a href="tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type Management</font></a></th>

<th><a href="../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association Management</font></a></th>

<th><a href="../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>

</table>

<br/><br/><br/><br/><br/>

```

K-1.9 Κώδικας tab1.php

Το παράθυρο που εμφανίζει τα υπάρχοντα Entity Types (Εικόνα 7.7) τα οποία είναι αποθηκευμένα στη βάση δεδομένων μας, δημιουργήθηκε με το script tab1 και παραθέτω τον κώδικα παρακάτω:

```

<?php

session_start();

//check to see if user has logged in with a valid password

if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

```

```

exit();

}

echo ' <p align="right"> ' . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';

?>

<?php include './db.inc.php';?>

<html>

<head>

<title> CMS </title>

<style type="text/css">

td {vertical-align: top;}

</style>

</head>

<body bgcolor="#F0F0F0">

<table style="width:100%;">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

<br/><br/>

<?php

if (isset($_SESSION['name'])) {

echo ' <p> You are currently logged in as: ' . $_SESSION['name'] . ' </p>';

}

?>

</div>

<style>

table,th,td

th

{

background-color:#E8E8E8;

}

</style>

<table border = "1" align="center">

<th><a href="tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type Management</font></a></th>

<th><a href=" ../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association
Management</font></a></th>

<th><a href=" ../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>

</table>

<br/><br/><br/><br/><br/>

```



```

<p>Existing Entity Types:</p><br/>

<style type="text/css">

.odd_row {background-color:#EEE;}

.even_row {background-color:#FFF;}

</style>

<table border="1" style="width:60%;">

<?php

$query = 'SELECT * FROM entity_type';

$result = mysql_query($query, $db) or die(mysql_error($db));

$odd = true;

$i=1;

while ($row = mysql_fetch_assoc($result)) {

echo ($odd == true) ? '<tr class="odd_row">' : '<tr class="even_row">';

$odd = !$odd;

echo '<td style="width:75%;">';

echo $i++;

echo ' .';

echo $row['short_name'];

echo '</td><td>';

echo ' <a href="tab1_2.php?action=view&name=' . $row['short_name'] . ' &id=' . $row['id'] . '" style="text-decoration:none; color: #990099">
[View] </a>';

echo ' <a href="add_edit_entitytype.php?action=edit&id=' . $row['id'] . '" style="text-decoration:none; color: #990099"> [Rename] </a>';

echo ' <a href="commit1.php?action=delete&id=' . $row['id'] . '" style="text-decoration:none; color: #990099"> [Delete] </a>';

echo '</td></tr>';

}

?>

</table>

<br/><br/>

<p align="center"><a href="add_edit_entitytype.php?action=add" style="text-decoration:none; color: CC6600"><font size="4">ADD NEW
ENTITY TYPE</font></a></p>

</body>

</html>

```

K-1.10 Κώδικας tab1_2.php

Το παράθυρο που εμφανίζει τα υπάρχοντα Entities (Εικόνα 7.10, Εικόνα 7.11, Εικόνα 7.12) τα οποία είναι αποθηκευμένα στη βάση δεδομένων μας, δημιουργήθηκε με το script tab1_2 και παρακάτω παραθέτω τον κώδικα του :

```

<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

echo ' <p align="right"> ' . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style=" color: #FF0000">
Logout? </a></p>';

?>

<?php include 'header.php';

include '../db.inc.php';

?>

<html>

<body>

<p>View Entities of Type : <?php echo $_GET['name'];?></p>

<style type="text/css">

.odd_row {background-color:#EEE;}

.even_row {background-color:#FFF;}

</style>

<table border="1" style="width:60%;">

<?php

$counter = 1;

$query = 'SELECT * FROM entity

WHERE entity_type_id = ' . $_GET['id'];

$result = mysql_query($query, $db) or die(mysql_error($db));

$odd = true;

while ($row = mysql_fetch_assoc($result)) {

echo ($odd == true) ? '<tr class="odd_row">' : '<tr class="even_row">';

$odd = !$odd;

echo '<td style="width:75%;">';

echo $counter;

++$counter;

echo ' . ';

echo $row['short_name'];

echo '</td><td>';

echo ' <a href="commit2.php?action=view&id=' . $row['id'] . '" style="text-decoration:none; color: #990099"> [View] </a>';

```

```

echo ' <a href="add_edit_entity.php?action=edit&id=' . $row['id'] . '" style="text-decoration:none; color: #990099"> [Edit] </a>';

echo ' <a href="commit2.php?action=delete&id=' . $row['id'] . '" style="text-decoration:none; color: #990099"> [Delete] </a>';

echo '</td></tr>';

}

?>

</table>

<br/><br/>

<?php

echo '<p align="center"><a href="add_edit_entity.php?action=add&id=' . $_GET['id'] . '" style="text-decoration:none; color: CC6600"><font
size="4">ADD NEW ENTITY</font></a></p>'

?>

</body>

</html>

```

K-1.11 Κώδικας add_edit_entitytype.php

Το παράθυρο όπου μπορεί ο χρήστης να προσθέσει ένα καινούργιο Entity Type (Εικόνα 7.8) και το παράθυρο όπου μπορεί να διορθώσει κάποιο υπάρχων Entity Type (Εικόνα 7.9), δημιουργήθηκε με το script add_edit_entitytype. Παρακάτω παραθέτω τον κώδικα του.

```

<?php

session_start();

//check to see if user has logged in with a valid password

if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

echo ' <p align="right">' . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';

?>

<?php

include 'header.php';

// This file, modifies the input form according to what is selected add/edit

?>

<?php

include '../db.inc.php';

if ($_GET['action'] == 'edit') {

//retrieve the record's information

$query = 'SELECT

```

```

        short_name, description
FROM
        entity_type
WHERE
id = ' . $_GET['id'];
$result = mysql_query($query, $db) or die(mysql_error($db));
$data = mysql_fetch_assoc($result);
$ShortName = $data["short_name"];
$Description = $data["description"];
} else {
//set values to blank
$ShortName = "";
$Description = "";
}
if ($_GET['action'] == 'add'){
echo '<p> Add New Entity Type:</p>';}
?>
<html>
<head>
<title><?php echo ucfirst($_GET['action']); ?> Entity Type</title>
</head>
<body>
<form action="commit1.php?action=<?php echo $_GET['action']; ?>" method="post">
<table style = "background-color:#F0F0F0;" align="center">
<tr>
<td> Name: </td>
<td><input type="text" name="ShortName" size="54"
value="<?php echo $ShortName; ?>" /></td>
</tr><tr>
<td> Description: </td>
<td><textarea name="Description" cols="40" rows="5">
<?php echo $Description; ?></textarea></td>
</tr><tr>
<td colspan="2" style="text-align: center;">
<?php
if ($_GET['action'] == 'edit') {

```

```

echo '<input type="hidden" value="" . $_GET['id'] . "' name="id" />';
}
?>

<input type="submit" name="submit"
value="<?php echo ucfirst($_GET['action']); ?>" />
</td>
</tr>
</table>
</form>
</body>
</html>

```

K-1.12 Κώδικας add_edit_entity.php

Το παράθυρο όπου μπορεί ο χρήστης να προσθέσει ένα καινούργιο Entity (Εικόνα 7.13) και το παράθυρο όπου μπορεί να διορθώσει κάποιο υπάρχων Entity (Εικόνα 7.15) δημιουργήθηκε με το script add_edit_entity. Παρακάτω παραθέτω τον κώδικα του.

```

<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['logged'] != 1) {
echo 'Sorry, but you don\'t have permission to view this page!';
exit();
}

echo ' <p align="right"> ' . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style=" color: #FF0000">
Logout? </a></p>';

?>

<?php include 'header.php';

// This file, modifies the input form according to what is selected add/edit

?>

<?php
include '../db.inc.php';

if ($_GET['action'] == 'edit') {

//retrieve the record's information

$query = 'SELECT
        short_name, description, entity_type_id, entity_image, weblink
FROM
        entity
WHERE

```

```

id = ' . $_GET['id'];

$result = mysql_query($query, $db) or die(mysql_error($db));

$data = mysql_fetch_assoc($result);

$ShortName = $data['short_name'];

$Description = $data['description'];

$EntityTypeID = $data['entity_type_id'];

$EntityImage = $data['entity_image'];

$Weblink = $data['weblink'];

} else {

//set values to blank

$ShortName = "";

$Description = "";

$EntityTypeID = "";

$EntityImage = "";

$Weblink = "";

}

if ($_GET['action'] == 'add'){

echo '<p> Add New Entity:</p>';}

?>

<html>

<head>

<title><?php echo ucfirst($_GET['action']); ?> Entity </title>

</head>

<body>

<form action="commit2.php?action=<?php echo $_GET['action']; ?>" method="post" enctype="multipart/form-data">

<table cellspacing="8px" style = "background-color:#F0F0F0;" align="center">

<tr>

<td><b> Name: </b></td>

<td><input type="text" name="ShortName" size="54"

value="<?php echo $ShortName; ?>" /></td>

</tr><tr>

<td><b> Description:</b></td>

<td><textarea name="Description" cols="40" rows="5">

<?php echo $Description; ?></textarea></td>

</tr><tr>

<td><b> Links: </b></td>

```

```

<td><input type="url" name="Weblink" size="54"
value="<?php echo $Weblink; ?>" /></td>
</tr><tr>
<td><b>Image:</b> </br><small>(max size 1Mb)</small></td>
<td><input type="file" name="EntityImage" />
<?php if ($_GET['action'] == 'edit'){ echo '<p align="center"></p>';}
echo '</td><td>';?></td>
</tr>
<td colspan="2" style="text-align: center;">
<?php
if ($_GET['action'] == 'edit') {
echo '<input type="hidden" value="' . $_GET['id'] . '" name="id" />';
}
if ($_GET['action'] == 'add') {
echo '<input type="hidden" value="' . $_GET['id'] . '" name="id" />';
}
?>
</br>
<input type="submit" name="submit"
value="<?php echo ucfirst($_GET['action']); ?>" />
</td>
</tr>
</table>
</form>
</body>
</html>

```

K-1.13 Κώδικας commit1.php

Το script commit1 πραγματοποιεί τις αλλαγές που υφίσταται η βάση δεδομένων μας και συγκεκριμένα ο πίνακας Entity Type . Οι αλλαγές αυτές αφορούν προσθήκη νέου Entity Type και διόρθωση ή διαγραφή κάποιου υπάρχοντος. Παρακάτω παραθέτω τον κώδικα του.

```

<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['logged'] != 1) {
echo 'Sorry, but you don\'t have permission to view this page!';

```

```

exit();
}
?>
<?php include 'header.php';?>
<?php
include '../db.inc.php';
?>
<html>
<head>
<title> Commit </title>
</head>
<body>
<?php
switch ($_GET['action']) {
//Add Entity Type
case 'add':
$query1 = 'INSERT INTO
entity_type
(id, short_name, description)
VALUES
(NULL,
" . $_POST['ShortName'] . "',
" . $_POST['Description'] . "')';
break;
//Edit Entity Type
case 'edit':
$query2 = 'UPDATE entity_type SET
short_name = " . $_POST['ShortName'] . "',
description = " . $_POST['Description'] . "'
WHERE
id = ' . $_POST['id'];
break;
//Delete Entity Type
case 'delete':
$query3 = 'DELETE FROM entity_type
WHERE

```



```

id = ' . $_GET['id'];

break;

}

//According to the specific query, do the appropriate actions to the database

//add element to the database

if (isset($query1)) {

$result1 = mysql_query($query1, $db) or die(mysql_error($db));

echo "<p align=center>Successful Entry !!</p>";

header("Refresh: 1; URL=tab1.php");

}

//Edit the Database Element

if (isset($query2)) {

$result2 = mysql_query($query2, $db) or die(mysql_error($db));

echo "<p align=center>Successful Edit !!</p>";

header("Refresh: 1; URL=tab1.php");

}

//Delete the Database Element

if (isset($query3)) {

$result3 = mysql_query($query3, $db) or die(mysql_error($db));

header("Refresh: 0; URL=tab1.php");

}

?>

</body>

</html>

```

K-1.14 Κώδικας commit2.php

Το script commit2 πραγματοποιεί τις αλλαγές που υφίσταται η βάση δεδομένων μας και συγκεκριμένα ο πίνακας Entity . Οι αλλαγές αυτές αφορούν προσθήκη νέου Entity και διόρθωση ή διαγραφή κάποιου υπάρχοντος. Παρακάτω παραθέτω τον κώδικα του.

```

<?php

session_start();

//check to see if user has logged in with a valid password

if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

```

```

echo ' <p align="right"> . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style=" color: #FF0000">
Logout? </a></p>';

?>

<?php include 'header.php';?>

<?php

include '../db.inc.php';

?>

<html>

<head>

<title> Commit </title>

</head>

<body>

<?php

if (empty($_FILES['EntityImage']['tmp_name'])){

$image = NULL;}

else{

$image = mysql_real_escape_string(file_get_contents($_FILES['EntityImage']['tmp_name']));

}

switch ($_GET['action']) {

//Add Entity Type

case 'add':

$query1 = "INSERT INTO

entity

(id, short_name, description, entity_type_id, entity_image, weblink)

VALUES

(NULL,

" . $_POST['ShortName'] . ",

" . $_POST['Description'] . ",

" . $_POST['id'] . ",

'$image',

" . $_POST['Weblink'] . ")";

break;

//Edit Entity Type

case 'edit':

if (empty($_FILES['EntityImage']['tmp_name'])) {

$query2 = "UPDATE entity SET

short_name = " . $_POST['ShortName'] . ",

```


</html>

3) Το τρίτο μέρος της ιστοσελίδας δημιουργήθηκε με τα scripts που περιέχονται στο φάκελο associations.



K-1.15 Κώδικας index.php

Το παράθυρο όπου εμφανίζει τα υπάρχοντα associations (Εικόνα7.16) δημιουργήθηκε με το script index. Παρακάτω παραθέτω τον κώδικα του.

```
<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['logged'] != 1) {
echo 'Sorry, but you don\'t have permission to view this page!';
exit();
}

echo ' <p align="right"> . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';
?>

<?php

//Find associations

include './db.inc.php';

$query =

"SELECT a.id as e1_id,
           a.short_name as e1_short_name,
           b.id as e2_id,
           b.short_name as e2_short_name

FROM entity_association

JOIN entity as a ON entity_association.entity_id_1 = a.id

JOIN entity as b ON entity_association.entity_id_2 = b.id
```

```

JOIN entity_type as et1 ON et1.id = a.entity_type_id
JOIN entity_type as et2 ON et2.id = b.entity_type_id ";

//Selected entity type

    if (@$_GET["entity-type"]) {

        $entityTypeId = @$_GET['entity-type'];

        $query .=

        sprintf("WHERE a.entity_type_id = %d

                OR b.entity_type_id = %d ",$entityTypeId,$entityTypeId);

        // Search

    } elseif (@$_GET['q']) {

        $search = @$_GET['q'] . '%';

        $query.=

        sprintf("WHERE a.short_name like '%s'

                OR b.short_name like '%s'

                                OR      et1.short_name like '%s'

                                OR      et2.short_name like '%s' ",

                $search,$search,$search,$search);

    }

    $query.= " ORDER BY date_registered";

    $result = mysql_query($query, $db) or die(mysql_error($db));

    $associations = mysql_fetch_assoc($result);

    $query =

"SELECT *

FROM entity_type ";

    $result2 = mysql_query($query, $db) or die(mysql_error($db));

?>

<html>

<head>

<title> CMS </title>

<style type="text/css">

td {vertical-align: top;}

</style>

</head>

<body bgcolor="#F0F0F0">

<table style="width:100%;">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

```

```

<br/><br/>

<?php
if (isset($_SESSION['name'])) {
echo '<p> You are currently logged in as: ' . $_SESSION['name'] . ' </p>';
}
?>
</div>

<style>
table,th,td
th
{
background-color:#E8E8E8;
}
</style>

<table border = "1" align="center">

<th><a href="../entity_type_management/tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type
Management</font></a></th>

<th><a href="../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association
Management</font></a></th>

<th><a href="../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>
</table>

<br/><br/><br/><br/>

<p>Existing Associations:</p><br/>

<style type="text/css">
.odd_row {background-color:#EEE;}
.even_row {background-color:#FFF;}
</style>

<table border="0" style="width:60%;">
<tr><td align="right">
<form action="" method="get">
<input type="text" name="q"/><input type="submit" value="Search"/>
</form>
</td></tr>
</table>

<table border="1" style="width:60%;">
<thead>
<tr>

```

```

<th style="width:50%;">Entity Types</th>

<th style="width:50%;">Associations</th>

</tr>

</thead>

<tr>

<td style="width:50%;">

<ul>

<?php

while($entityType = mysql_fetch_assoc($result2)){

echo "<li><a href='index.php?entity-type=" . $entityType['id']. "'>";

    echo $entityType['short_name'];

    echo "</a></li>";

}

?>

</ul>

</td>

<td style="width:50%;">

<ul>

<?php

while($association = mysql_fetch_assoc($result)){

$url="view.php?entity-id-1=%d&entity-id-2=%d";

    $url = sprintf($url,$association['e1_id'],$association['e2_id'] );

    echo "<li><a href='$url'>";

    echo $association['e1_short_name'] . " <strong>with</strong> " . $association['e2_short_name'];

    echo "</a></li>";

}

?>

</ul>

</td>

</tr>

<?php

?>

</table>

<br/><br/>

<p align="center"><a href="step1.php" style="text-decoration:none; color: CC6600"><font size="4">ADD NEW ASSOCIATION</font></a></p>

</body>

</html>

```


K-1.16 Κώδικας view.php

Το παράθυρο όπου μπορεί ο χρήστης να διορθώσει ή να διαγράψει κάποιο association (Εικόνα 7.17), δημιουργήθηκε με το script view. Παρακάτω παραθέτω τον κώδικα του.

```
<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['logged'] != 1) {
echo 'Sorry, but you don\'t have permission to view this page!';

exit();
}

echo ' <p align="right"> . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';

?>

<?php

//Find associations

include './db.inc.php';

$query =

"SELECT a.id as e1_id,

                a.short_name as e1_short_name,

b.id as e2_id,

                b.short_name as e2_short_name,

                ea.description

FROM entity_association ea

JOIN entity as a ON ea.entity_id_1 = a.id

JOIN entity as b ON ea.entity_id_2 = b.id

WHERE ea.entity_id_1 = %d AND ea.entity_id_2 = %d ";

$result = mysql_query(sprintf($query,$_GET['entity-id-1'],$_GET['entity-id-2']));

$association = mysql_fetch_assoc($result);

?>

<html>

<head>

<title> CMS </title>

<style type="text/css">

td {vertical-align: top;}

</style>

</head>

<body bgcolor="#F0F0F0">
```

```

<table style="width:100%;">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

<br/><br/>

<?php

if (isset($_SESSION['name'])) {

echo '<p> You are currently logged in as: ' . $_SESSION['name'] . ' </p>';

}

?>

</div>

<style>

table,th,td

th

{

background-color:#E8E8E8;

}

</style>

<table border = "1" align="center">

<th><a href=" ../entity_type_management/tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type
Management</font></a></th>

<th><a href=" ../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association
Management</font></a></th>

<th><a href=" ../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>

</table>

<br/><br/><br/><br/><br/>

<h2>Association <?php echo $association['e1_short_name'] ?> with <?php echo $association['e2_short_name'] ?></h2>

<table border="1" style="width:60%;">

<tr>

<td style="width:30%;">Description</td>

<td ><textarea style="width:100%" rows="10" ><?php echo htmlspecialchars ($association['description'])?></textarea></td>

</tr>

</table>

<?php

$url="associations.php?action=delete&entity-id-1=%d&entity-id-2=%d";

$url = sprintf($url,$association['e1_id'],$association['e2_id'] );

$editUrl="step2.php?action=edit&entity-id-1=%d&entity-id-2=%d";

$editUrl = sprintf($editUrl,$association['e1_id'],$association['e2_id'] );

?>

```

```

<p align="center"><a href="<?php echo $editUrl?>" style="text-decoration:none; color: CC6600"><font size="4">Edit </font></a></p>
<p align="center"><a href="<?php echo $url?>" style="text-decoration:none; color: CC6600"><font size="4">Delete </font></a></p>
</body>
</html>

```

K-1.17 Κώδικας step1.php

Το παράθυρο που είναι το πρώτο βήμα για την προσθήκη ενός καινούργιου association (Εικόνα 7.18), όπου εκεί ο χρήστης συνδυάζει δυο διαφορετικά entity types δημιουργήθηκε με το script step1. Παρακάτω παραθέτω τον κώδικα του.

```

<?php
session_start();

//check to see if user has logged in with a valid password

if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

echo ' <p align="right"> . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';

?>

<?php

//Find associations

include './db.inc.php';

$query =

"SELECT *

FROM entity_type ";

$result2 = mysql_query($query, $db) or die(mysql_error($db));

?>

<html>

<head>

<title> CMS </title>

<style type="text/css">

td {vertical-align: top;}

</style>

</head>

<body bgcolor="#F0F0F0">

<table style="width:100%;">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

```

```

<br/><br/>

<?php
if (isset($_SESSION['name'])) {
echo '<p> You are currently logged in as: ' . $_SESSION['name'] . ' </p>';
}
?>

</div>

<style>
table,th,td
th
{
background-color:#E8E8E8;
}
</style>

<table border = "1" align="center">

<th><a href="../entity_type_management/tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type
Management</font></a></th>

<th><a href="../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association
Management</font></a></th>

<th><a href="../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>

</table>

<br/><br/><br/><br/><br/>

<p>Add new association Step 1/2:</p><br/>

<form action="step2.php" method="get">

<style type="text/css">
.odd_row {background-color:#EEE;}
.even_row {background-color:#FFF;}
</style>

<table border="0" style="width:60%;">

<tr><td align="right">

</td></tr>

</table>

<table border="1" style="width:60%;">

<thead>

<tr>

<th style="width:50%;">Choose entity type 1</th>

<th style="width:50%;">Choose entity type 2</th>

```

```

</tr>

</thead>

<tr>

<td style="width:50%;">

<select name="entity_type_id_1">

<?php
while($entityType = mysql_fetch_assoc($result2)){
echo '<option value="' . $entityType['id'] . "'>' . $entityType['short_name'] . '</option>';
}
?>

</select>

</td>

<td style="width:50%;">

<select name="entity_type_id_2">

<?php
mysql_data_seek($result2,0);
while($entityType = mysql_fetch_assoc($result2)){
echo '<option value="' . $entityType['id'] . "'>' . $entityType['short_name'] . '</option>';
}
?>

</select>

</td>

</tr>

<?php
?>

</table>

<br/><br/>

<input type="submit" value="Next"/>

</form>

</body>

</html>

```

K-1.18 Κώδικας step2.php

Το παράθυρο που είναι το δεύτερο βήμα για την προσθήκη ενός καινούργιου association (Εικόνα 7.19), όπου εκεί ο χρήστης συνδυάζει δυο διαφορετικά entities δημιουργήθηκε με το script step2. Παρακάτω παραθέτω τον κώδικα του.

```

<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

echo ' <p align="right"> ' . $_SESSION['username'] . ' you are currently logged in / <a href="./session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';

?>

<?php

//Find associations
include './db.inc.php';

    $association = null;

    if(@$_REQUEST['action'] == 'edit'){

        $query =

"SELECT a.id as e1_id,

        a.entity_type_id as e1_entity_type_id,

        a.short_name as e1_short_name,

        b.id as e2_id,

        b.entity_type_id as e2_entity_type_id,

        b.short_name as e2_short_name,

        ea.description

FROM entity_association ea

JOIN entity as a ON ea.entity_id_1 = a.id

JOIN entity as b ON ea.entity_id_2 = b.id

WHERE ea.entity_id_1 = %d AND ea.entity_id_2 = %d ";

        $result = mysql_query(sprintf($query,$_GET['entity-id-1'],$_GET['entity-id-2']));

        $association = mysql_fetch_assoc($result);

        $entityType1 = $association["e1_entity_type_id"];

        $entityType2 = $association["e2_entity_type_id"];

    }

    else{

        $entityType1 = $_GET['entity_type_id_1'];

        $entityType2 = $_GET['entity_type_id_2'];

    }

    $queryEntityType =

```

```

"SELECT short_name
FROM entity_type
WHERE id = %d " ;

$resultET1 = mysql_query(
sprintf($queryEntityType,$entityTypeId1) , $db) or die(mysql_error($db));
$resultET1 = mysql_fetch_row($resultET1);
$resultET2 = mysql_query(
sprintf($queryEntityType,$entityTypeId2) , $db) or die(mysql_error($db));
$resultET2 = mysql_fetch_row($resultET2);

$queryEntity =
"SELECT id,short_name
FROM entity
WHERE entity_type_id = %d " ;
$resultE1 = mysql_query(
sprintf($queryEntity,$entityTypeId1) , $db) or die(mysql_error($db));
$resultE2 = mysql_query(
sprintf($queryEntity,$entityTypeId2) , $db) or die(mysql_error($db));
?>

<html>
<head>
<title> CMS </title>
<style type="text/css">
td {vertical-align: top;}
</style>
</head>
<body bgcolor="#F0F0F0">
<table style="width:100%;">
<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>
<br/><br/>
<?php
if (isset($_SESSION['name'])) {
echo '<p> You are currently logged in as: ' . $_SESSION['name'] . ' </p>';
}
?>
</div>
<style>

```

```

table,th,td

th

{

background-color:#E8E8E8;

}

</style>

<table border = "1" align="center">

<th><a href="../entity_type_management/tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type
Management</font></a></th>

<th><a href="../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association
Management</font></a></th>

<th><a href="../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>

</table>

<br/><br/><br/><br/>

<p>Add new association Step 2/2:</p><br/>

<?php

if(@$_REQUEST['action'] == 'edit'){

?>

<form action="associations.php?action=edit" method="post">

<input type="hidden" name="cur_entity_id_1" value=" <?php echo @$association['e1_id']?>"/>

<input type="hidden" name="cur_entity_id_2" value="<?php echo @$association['e2_id']?>"/>

<?php }else { ?>

<form action="associations.php?action=add" method="post">

<?php }?>

<input type="hidden" name="entity_type_id_1" value=" <?php echo entityType1?>"/>

<input type="hidden" name="entity_type_id_2" value="<?php echo entityType2?>"/>

<style type="text/css">

.odd_row {background-color:#EEE;}

.even_row {background-color:#FFF;}

</style>

<table border="0" style="width:60%;">

<tr><td align="right">

</td></tr>

</table>

<table border="1" style="width:60%;">

<thead>

<tr>

```



```

<th style="width:50%;">Choose entity of type <?php echo $resultET1[0] ?> </th>
<th style="width:50%;">Choose entity of type <?php echo $resultET2[0] ?></th>
</tr>
</thead>
<tr>
<td style="width:50%;">
<select name="entity_id_1">
<?php
while($entity = mysql_fetch_assoc($resultE1)){
echo '<option value="" . $entity['id'] . ""';
if(@$association['e1_id'] == $entity['id']){
echo ' selected="selected" ';
}
echo '>'. $entity['short_name'] . '</option>';
}
?>
</select>
</td>
<td style="width:50%;">
<select name="entity_id_2">
<?php
while($entity = mysql_fetch_assoc($resultE2)){
echo '<option value="" . $entity['id'] . ""';
if(@$association['e2_id'] == $entity['id']){
echo ' selected="selected" ';
}
echo '>'. $entity['short_name'] . '</option>';
}
?>
</select>
</td>
</tr>
<tr>
<td colspan="2">
<h4 for="description">Give description for association</h4>
<textarea id="description" rows="7" cols="50" name="description" ><?php echo htmlentities(@$association['description'])?></textarea>

```

```

</td>

</tr>

<?php
?>

</table>

<br/><br/>

<input type="submit" value="Create Association"/>

</form>

</body>

</html>

```

K-1.19 Κώδικας associations.php

Το script που πραγματοποιεί τις αλλαγές που συμβαίνουν στη βάση δεδομένων μας, στο πεδίο αυτό και συγκεκριμένα στον πίνακα entity_association είναι το associations. Οι αλλαγές αφορούν προσθήκη ενός καινούργιου association και διόρθωση ή διαγραφή ενός υπάρχοντος. Παρακάτω παραθέτω τον κώδικα ΤΟΥ.

```

<?php

session_start();

//check to see if user has logged in with a valid password

if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

?>

<?php

//error_reporting(E_ALL);

//ini_set('display_errors', 1);

include './db.inc.php';

//if add, check for the association, if already exist in the database

if($_REQUEST['action'] == 'add'){

//edo bazo thn epilogh tou xrhsth gia ta duo entities(einai sunduasmenh san enas akeraios arithmos)

$combined_entities = $_POST['entity_id_1']. $_POST['entity_id_2'];

//epilego ola ta associations se enan pinaka list_associations (einai akeraioi arithmoi anapoda:prota entity_id_2 meta entity_id_1)

$query = ' SELECT CONCAT( entity_id_2, entity_id_1 ) AS list_associations

FROM entity_association';

$result = mysql_query($query, $db) or die(mysql_error($db));

while ($outcome = mysql_fetch_assoc($result)){

if($combined_entities == $outcome['list_associations']){

```

```

exit("<b>This Association Already Exist!</b>");
}
}

// In case of add or edit, Check if Entities are the same => if same don't proceed
if(($_REQUEST['action']!= 'delete') && $_POST['entity_id_1'] == $_POST['entity_id_2']){
exit("<b>Error:</b> You try to associate identical Entities");
}

switch ($_REQUEST['action']) {

//Add Entity Type
case 'add':

$query1 = "INSERT INTO
entity_association
(entity_id_1, entity_id_2, description, date_registered)
VALUES
(%d,
%d,
'%s',
NOW())";

$query1= sprintf(
$query1,$_POST['entity_id_1'],$_POST['entity_id_2'],$_POST['description']);
$success = mysql_query($query1);

if($success){
header("Location: /associations/index.php");
die();
}

else{

if (mysql_errno() == 1062) {
echo '<b>This Association Already exist!</b><br>';
}

echo mysql_error();
}

break;

//Edit Entity Type
case 'edit':

$query2 =
"UPDATE entity_association

```

```

SET

entity_id_1 = %d,

entity_id_2 = %d,

description = '%s'

WHERE

entity_id_1 = %d

AND

entity_id_2 = %d

";

$query2= sprintf(

$query2,

$_POST['entity_id_1'],$_POST['entity_id_2'],$_POST['description'],

$_POST['cur_entity_id_1'],$_POST['cur_entity_id_2']);

$success = mysql_query($query2);

if($success){

header("Location: /associations/index.php");

die();

}

else{

if (mysql_errno() == 1062) {

    echo '<b>This Association Already exist!</b><br>';

}

echo mysql_error();

}

break;

//Delete Entity Type

case 'delete':

$query3 =

"DELETE FROM entity_association

WHERE

entity_id_1 = %d

AND

entity_id_2 = %d";

$query3= sprintf(

$query3,$_GET['entity-id-1'],$_GET['entity-id-2']);

mysql_query($query3);

```

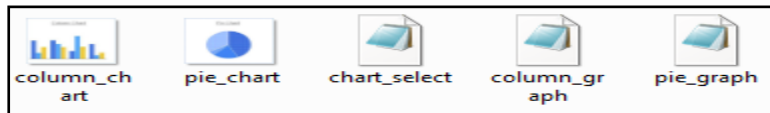
```
header('Location: /associations/index.php');

die();

break;

}
```

4) Το τελευταίο μέρος της ιστοσελίδας παρουσιάζει σε δυο γραφήματα(column chart, pie chart) το ποσοστό συμμετοχής κάθε ενός entity, σε σύγκριση με το σύνολο των entities για τη δημιουργία των associations . Το κομμάτι αυτό δημιουργήθηκε με τα scripts που περιέχονται στο φάκελο graphs.



K-1.20 Κώδικας chart_select.php

Το παράθυρο που δίνει τη δυνατότητα στο χρήστη να επιλέξει ποιο από τα δυο είδη γραφήματος θέλει να δει (Εικόνα 7.20), δημιουργήθηκε με το script chart_select και παρακάτω παραθέτω τον κώδικα του:

```
<?php

session_start();

//check to see if user has logged in with a valid password

if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

echo ' <p align="right"> ' . $_SESSION['username'] . ' you are currently logged in / <a href=" ../session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';

?>

<html>

<head>

<title> Select Chart </title>

</head>

<body bgcolor="#F0F0F0">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

<br/><br/>

<style>

table,th,td

th

{

background-color:#E8E8E8;
```



```

$query = 'SELECT
entity_id_1, COUNT(*) cnt
FROM
(SELECT entity_id_1 FROM entity_association
UNION ALL
SELECT entity_id_2 FROM entity_association) a
GROUP BY entity_id_1
ORDER BY cnt DESC ';
$result = mysql_query($query, $db) or die(mysql_error($db));
//Initialize the 2Dimensional Array
$my_array = array('entity', 'Number of Occurrences');
$props[] = $my_array;
//Select short_name(from table entity) according to specific id(table entity_association)
while ($data = mysql_fetch_array($result)){
$value = $data['entity_id_1'];
$query1 = 'SELECT short_name FROM entity
WHERE id = ' . $value;
$result1 = mysql_query($query1, $db) or die(mysql_error($db));
$entity_name = mysql_result($result1, 0);
//Fill the 2Dimensional Array $Props[]
$my_array = array($entity_name, intval($data['cnt']));
$props[] = $my_array;
}
//Encode the 2Dimensional Array to send to Javascript
$json = json_encode($props);
?>
<html>
<head>
<title> Graphical Representation </title>
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
google.load("visualization", "1", {packages:["corechart", 'table']});
google.setOnLoadCallback(drawChart);
function drawChart() {
var data = new google.visualization.arrayToDataTable

```

```

(<?php echo $json; ?>);

var options = {
    title: 'Associated Entities in the System',
        backgroundColor: '#F0F0F0',
    hAxis: {title: 'Entities', titleTextStyle: {color: 'blue'}}
};

var chart = new google.visualization.ColumnChart(document.getElementById('chart_div'));

chart.draw(data, options);
}

</script>
</head>

<body bgcolor="#F0F0F0">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

<br/><br/>

<style>

table,th,td

th

{

background-color:#E8E8E8;

}

</style>

<table border = "1" align="center">

<th><a href="../entity_type_management/tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type
Management</font></a></th>

<th><a href="../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association
Management</font></a></th>

<th><a href="../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>

</table>

<br/><br/><br/>

<div id="chart_div" style="width: 900px; height: 400px;"></div>

</body>

</html>

```

K-1.22 Κώδικας pie_graph.php

Το script που εμφανίζει το γράφημα pie_chart (Εικόνα 7.22), είναι το pie_graph και παρακάτω παρουσιάζω τον κώδικα του.

```

<?php
session_start();

```



```

//check to see if user has logged in with a valid password

if ($_SESSION['logged'] != 1) {

echo 'Sorry, but you don\'t have permission to view this page!';

exit();

}

echo ' <p align="right"> ' . $_SESSION['username'] . ' you are currently logged in / <a href=".."../session_destroy.php?" style="color: #FF0000">
Logout? </a></p>';

?>

<?php

$db = mysql_connect('localhost', 'admin', 'admin') or

die ('Unable to connect. Check your connection parameters.');
```

```

mysql_select_db('metabolites_second', $db) or die(mysql_error($db));

?>

<?php

//Select entity_id_1, entity_id_2 under one table(2columns => entity_id_1 and cnt) and count the occurrences of each entity

$query = 'SELECT

entity_id_1, COUNT(*) cnt

FROM

(SELECT entity_id_1 FROM entity_association

UNION ALL

SELECT entity_id_2 FROM entity_association) a

GROUP BY entity_id_1

ORDER BY cnt DESC ';

$result = mysql_query($query, $db) or die(mysql_error($db));

//Initialize the 2Dimensional Array

$my_array = array('entity', 'Number of Occurrences');
```

```

$props[] = $my_array;

//Select short_name(from table entity) according to specific id(table entity_association)

while ($data = mysql_fetch_array($result)){

$value = $data['entity_id_1'];

$query1 = 'SELECT short_name FROM entity

WHERE id = ' . $value;

$result1 = mysql_query($query1, $db) or die(mysql_error($db));

$entity_name = mysql_result($result1, 0);

//Fill the 2Dimensional Array $Props[]

$my_array = array($entity_name, intval($data['cnt']));

$props[] = $my_array;

```

```

}

//Encode the 2Dimensional Array to send to Javascript

$json = json_encode($props);

?>

<html>

<head>

<title> Graphical Representation </title>

<script type="text/javascript" src="https://www.google.com/jsapi"></script>

<script type="text/javascript">

google.load("visualization", "1", {packages:["corechart", 'table']});

google.setOnLoadCallback(drawChart);

function drawChart() {

    var data = new google.visualization.arrayToDataTable(<?php echo $json; ?>);

var options = {

    title: 'Associated Entities in the System',

        backgroundColor: '#F0F0F0'

    };

var chart = new google.visualization.PieChart(document.getElementById('piechart'));

    chart.draw(data, options);

}

</script>

</head>

<body bgcolor="#F0F0F0">

<h1 align="center"> EurHEALTHAgeing Database Webpage </h1>

<br/><br/>

<style>

table,th,td

th

{

background-color:#E8E8E8;

}

</style>

<table border = "1" align="center">

<th><a href="../entity_type_management/tab1.php" style="text-decoration:none; color: #990099"><font size="5">Entity Type
Management</font></a></th>

<th><a href="../associations/index.php" style="text-decoration:none; color: #990099"><font size="5">Association
Management</font></a></th>

```

```
<th><a href="../../associations/chart_select.php" style="text-decoration:none; color: #990099"><font size="5">User Reporting</font></a></th>
</table>
<br><br><br>
<div id="piechart" style="width: 900px; height: 500px;"></div>
</body>
</html>
```

Google Charts

Στο σημείο αυτό πρέπει να αναφέρω πως στα δυο παραπάνω scripts (column_graph.php, pie_graph.php), έγινε χρήση των google charts. Τα google charts παρέχουν κώδικα γραμμένο σε javascript και χρησιμοποιούνται για την απεικόνιση διαφόρων ειδών γραφημάτων. Τον κώδικα αυτό τον ενσωμάτωσα στα δικά μου δεδομένα, για να μπορέσω να απεικονίσω τα δυο γραφήματα (column_graph, pie_graph). Η βασική δομή και των δυο scripts, είναι η τοποθέτηση των προς απεικόνιση δεδομένων μας σε έναν δυσδιάστατο πίνακα. Με την χρήση της συνάρτησης **json_encode** της javascript, ο πίνακας διαμορφώνεται ώστε τα δεδομένα μας να είναι προσπελάσιμα από τον κώδικα των google charts.

Βιβλιογραφία

- [1] Julie C Meloni. (2008). ΜΑΘΕΤΕ PHP, MySQL και Apache. Αθήνα, Ελλάδα : Μ.Γκιούρδας.
- [2] Atkinson L.Suraski Z.(2004). Πλήρης οδηγός της PHP 5. Αθήνα , Ελλάδα : Μ.Γκιούρδας
- [3] Χρυσάνθη Αγγελή. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ WEB -HTML 4 & ASP. Αθήνα, Ελλάδα. Εκδόσεις Σύγχρονη Εκδοτική (2005).
- [4] Timothy Boronczyk, Elizabeth Naramore, Jason Gerner , Yann Le Scouarnec, Jeremy Stolz, Michael K.Glass. Beginning PHP6, Apache, MySQL Web Development. Wrox An Imprint of Wiley.
- [5] Ιστότοπος για τη δικτυακή γλώσσα PHP <http://www.php.net/>
- [6] Ιστότοπος για τη βάση δεδομένων MySQL <http://www.mysql.com>
- [7] Ιστότοπος για τον server Apache <http://www.apache.org/>
- [8] Pagliari C., Sloan D., Gregor P., Sullivan F., Detmer D., Kahan JP., Oortwijn W., MacGillivray S. “What is eHealth: a scoping exercise to map the field”, J Med Internet Res. 31 March 2005
- [9] Eckerson, Wayne W., “Three Tier Client/Server Architecture: Achieving Scalability, Performance and Efficiency in Client Server Applications.”, January 1995
- [10] “Hypertext Transfer Protocol -HTTP/1.1”
(<http://www.w3.org/Protocols/rfc2616/rfc2616.txt>),
- [11] Hugh E. Williams and David Lane, “Web Database Applications with PHP and MySQL, 2d Edition”, O’Reilly Media Inc, 2004
- [12] Larry Ullman: *PHP and MySQL for dynamic web sites*, 2nd Edition, Peachpit Press