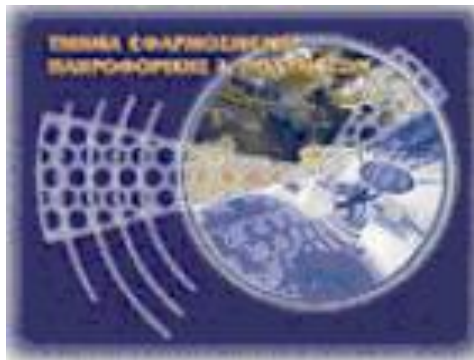




ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή Εργασία

Τίτλος : ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΓΙΑ ΛΟΓΙΣΤΙΚΟ ΓΡΑΦΕΙΟ

ΚΥΡΙΑΚΟΣ ΜΟΥΝΤΑΚΗΣ (ΑΜ: 2254)

Επιβλέπων Καθηγητής : ΠΑΠΑΔΑΚΗΣ ΝΙΚΟΛΑΟΣ

Επιτροπή Αξιολόγησης :

Ημερομηνία Παρουσίασης:

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της πτυχιακής μου εργασίας, θα ήθελα να ευχαριστήσω τους ανθρώπους οι οποίοι βοήθησαν στην περάτωση αυτής της εργασίας. Κατά κύριο λόγο, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου από το Τ.Ε.Ι Κρήτης κ.Παπαδάκη Νικόλαο, ο οποίος με υποστήριξε καθ' όλη τη διάρκεια της πτυχιακής εργασίας, όπως βέβαια και όλους τους καθηγητές για την συνεργασία, την βοήθεια και την εμπιστοσύνη που μου έδειξαν όλα αυτά τα χρόνια. Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένειά μου που με στήριξε σε καθ' όλη την διάρκεια της φοίτησής μου.

Table of Contents

ΕΥΧΑΡΙΣΤΙΕΣ	2
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....	5
1.1 ABSTRACT	5
1.2 ΣΥΝΟΨΗ.....	6
1.3 ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ.....	7
ΚΕΦΑΛΑΙΟ 2 ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ	10
2.1 ΜΕΘΟΔΟΣ ΑΝΑΛΥΣΗΣ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΠΤΥΧΙΑΚΗΣ	10
2.1.1 PHP.....	10
2.1.2 ΘΕΩΡΙΑ SERVER.....	17
2.1.2.1 WEB SERVERS.....	17
2.1.2.2 IP ADDRESSES.....	17
2.1.2.3 DOMAIN NAMES.....	18
2.1.2.4 NAME SERVERS.....	18
2.1.2.5 PORTS	18
2.1.2.6 SECURITY	18
2.1.2.7 DYNAMIC PAGES	19
2.1.3 ΘΕΩΡΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ.....	20
2.1.3.1 ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ-ΣΥΣΧΕΤΙΣΕΩΝ.....	20
2.1.3.2 ΟΝΤΟΤΗΤΑ	20
2.1.3.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	20
2.1.3.4 ΣΥΣΧΕΤΙΣΗ.....	21
2.1.3.5 ΒΑΘΜΟΣ ΣΥΣΧΕΤΙΣΕΩΝ.....	21
2.1.3.6 ΠΛΗΘΙΚΟΤΗΤΑ.....	22
2.1.3.7 ΑΣΘΕΝΗΣ ΤΥΠΟΣ ΟΝΤΟΤΗΤΑΣ.....	22
2.1.3.8 ΕΠΑΝΑΛΑΜΒΑΝΟΜΕΝΕΣ ΟΜΑΔΕΣ.....	22
2.1.3.9 ΠΛΕΙΟΤΙΜΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	23
2.1.3.10 ΥΠΟΚΛΑΣΕΙΣ ΚΑΙ ΥΠΕΡΚΛΑΣΕΙΣ.....	23
2.1.3.11 ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ	23
2.1.3.12 ΠΕΡΙΟΡΙΣΜΟΣ ΕΠΙΚΑΛΥΨΗΣ	23
2.1.4 ΜΟΝΤΕΛΑ.....	24
2.1.4.1 LAMP.....	24
2.1.4.2 SQL	24
2.1.4.3 MYSQL.....	25

2.1.4.4 PHP.....	25
2.1.4.5 HTML.....	27
2.1.4.6 CSS.....	30
2.1.4.7 Bootstrap	33
2.1.4.8 AJAX.....	35
ΚΕΦΑΛΑΙΟ 3 ΣΧΕΔΙΟ ΔΡΑΣΗΣ ΓΙΑ ΤΗΝ ΕΚΜΠΟΝΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	37
3.1 ΣΗΜΑΝΤΙΚΟΙ ΣΤΟΧΟΙ ΓΙΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ	37
3.2 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ	37
ΚΕΦΑΛΑΙΟ 4 ΚΥΡΙΟ ΜΕΡΟΣ ΠΤΥΧΙΑΚΗΣ	38
4.1 ΑΝΑΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ.....	38
4.1.2 ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ	39
4.2 ΣΧΕΔΙΑΣΜΟΣ ΥΛΟΠΟΙΗΣΗΣ	41
4.3 ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ.....	53
4.3.1 ΚΑΝΟΝΤΑΣ SIGN IN	53
4.3.2 DASHBOARD	54
4.3.3 SIDEBAR.....	54
4.3.4 ΠΡΟΣΘΗΚΗ ΝΕΟΥ ΧΡΗΣΤΗ	55
4.3.5 ΠΡΟΒΟΛΗ ΧΡΗΣΤΩΝ	56
4.3.6 ΠΡΟΣΘΗΚΗ ΠΕΛΑΤΩΝ.....	57
4.3.7 ΔΙΑΧΕΙΡΙΣΗ ΚΑΡΤΕΛΑΣ ΠΕΛΑΤΗ	58
4.3.8 ΥΠΟΛΟΓΙΣΜΟΣ ΦΠΑ	59
4.3.9 ΥΠΟΛΟΓΙΣΜΟΣ ΙΚΑ.....	59
4.3.10 ΡΥΘΜΙΣΕΙΣ ΧΡΗΣΤΗ	59
4.3.11 LOGOUT	60
ΚΕΦΑΛΑΙΟ 5 ΑΠΟΤΕΛΕΣΜΑΤΑ	61
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	61
5.2 ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ	61
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	62
ΠΑΡΑΡΤΗΜΑ 1: ΠΕΡΙΛΗΨΗ	63

ΚΕΦΑΛΑΙΟ 1 | ΕΙΣΑΓΩΓΗ

1.1 ABSTRACT

In this thesis a database was created which is a useful tool for accounting firms. There is the possibility of usage:

- by an administrator, who will be able to perform all functions of the system
- and from users who will have access to limited functions

The system gives the administrator and the users functionality over adding the accounting firm's customers tasks and generally have an overview of the client's business.

Finally a user manual was created, in which instructions are given about how anyone will be able to use the system, without any special IT skills. The description is very detailed, explaining step by step, while images of the system are also presented.

1.2 ΣΥΝΟΨΗ

Σε αυτή την πτυχιακή εργασία δημιουργήθηκε μία βάση δεδομένων η οποία είναι ένα χρήσιμο εργαλείο για τα λογιστικά γραφεία. Δίνεται η δυνατότητα χρησιμοποίησης

- από έναν διαχειριστή (administrator), ο οποίος θα μπορεί να πραγματοποιεί όλες τις λειτουργίες του συστήματος και
- από χρήστες (users) οι οποίοι θα έχουν πρόσβαση σε περιορισμένες λειτουργίες.

Στο σύστημα αυτό δίνεται η δυνατότητα στον διαχειριστή και στους χρήστες , να καταγράφουν στο σύστημα τα tasks που αφορούν τους πελάτες του λογιστικού γραφείου και γενικά να μπορούν να έχουν μία γρήγορη εικόνα της επιχείρησης του πελάτη.

Τέλος , δημιουργήθηκε ένα αναλυτικό εγχειρίδιο χρήσης του συστήματος στο οποίο δίνονται οδηγίες για το πως μπορεί ο οποιοσδήποτε να χρησιμοποιήσει το σύστημα , χωρίς δηλαδή να απαιτούνται κάποιες ιδιαίτερες γνώσεις πληροφορικής. Η περιγραφή είναι αναλυτική σε κάθε βήμα και όλες οι λειτουργίες παρουσιάζονται εικονογραφημένα.

1.3 ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1: ER διάγραμμα.....	21
Εικόνα 2: Τριμερής συσχέτιση	22
Εικόνα 3: Χρονοδιάγραμμα.....	37
Εικόνα 4: Πίνακες	41
Εικόνα 5: Εισαγωγές στον πίνακα πελατών.....	41
Εικόνα 6: Πίνακας πελατών	42
Εικόνα 7: Εισαγωγές στον πίνακα χρηστών.....	42
Εικόνα 8: Πίνακας χρηστών.....	42
Εικόνα 9: Sidebar.....	42
Εικόνα 10: Καρτέλα χρηστών	43
Εικόνα 11: Καρτέλα πελατών	43
Εικόνα 12: Dashboard	43
Εικόνα 13: index.php.....	44
Εικόνα 14: login.php.....	45
Εικόνα 15: new_customer.php	45
Εικόνα 16: insert_customer.php	46
Εικόνα 17: customers.php.....	46
Εικόνα 18: get_customers.php.....	47
Εικόνα 19: users.php	47
Εικόνα 20: get_users.php.....	48
Εικόνα 21: new_user.php.....	48
Εικόνα 22: insert_user.php	49
Εικόνα 23: Καρτέλα πελατών	49
Εικόνα 24: remove_dustomer.php	49
Εικόνα 25: remove_user.php	50
Εικόνα 26: menu.php	50
Εικόνα 27: calc_fpa.php	51
Εικόνα 28: calc_ika.php.....	51
Εικόνα 29: change_password.php	52
Εικόνα 30: Login	53
Εικόνα 31: Redirect 1	53
Εικόνα 32: Dashboard	54
Εικόνα 33: Admin Sidebar	54
Εικόνα 34: User Sidebar	55
Εικόνα 35: Εισαγωγή νέου χρήστη.....	55
Εικόνα 36: Redirect 2	56
Εικόνα 37: Καρτέλα χρηστών	56
Εικόνα 38: Redirect 3	57
Εικόνα 39: Εισαγωγή πελατών.....	57
Εικόνα 40: Καρτέλα πελάτη	58
Εικόνα 41: Redirect 4	58
Εικόνα 42: Υπολογισμός ΦΠΑ	59
Εικόνα 43: Υπολογισμός ΙΚΑ.....	59
Εικόνα 44: Αλλαγή κωδικού χρήστη	60

Εικόνα 45: Αποσύνδεση χρήστη	60
Εικόνα 46: Redirect 5	60

ΚΕΦΑΛΑΙΟ 2 | ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ

2.1 ΜΕΘΟΔΟΣ ΑΝΑΛΥΣΗΣ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΠΤΥΧΙΑΚΗΣ

2.1.1 PHP

Session

Η υποστήριξη για session στην php αποτελείται από τρόπους για τη διατήρηση δεδομένων μεταξύ διαδοχικών προσβάσεων σε ένα σύστημα. Η υποστήριξη για session μας επιτρέπει να αποθηκεύουμε δεδομένα μεταξύ requests στον `S_SESSION[]` που είναι ένας global πίνακας. Όταν ένας επισκέπτης προσπελάσει τη σελίδα μας, η PHP θα ελέγξει αυτόματα (αν η μεταβλητή `session.auto_start` είναι 1), ή με request του χρήστη (μέσω του `session_start()`), αν έχει αποσταλλεί συγκεκριμένο αναγνωριστικό μαζί με το request. Αν είναι έτσι, τότε τυχούσα αποθηκευμένη προηγούμενη συνεδρία αποκαθίσταται.

Session_start

Η `session_start ()` δημιουργεί μια συνεδρία ή συνεχίζει την τρέχουσα βαιζόμενη σε ένα αναγνωριστικό συνόδου πέρασε μέσω ενός αιτήματος GET ή POST, ή έχει περάσει μέσω ενός cookie. Όταν `session_start ()` καλείται ή όταν μια συνεδρία ξεκινά αυτόματα, η PHP θα καλέσει τους ανοιχτούς χειριστές αποθήκευσης. Αυτοί θα είναι είτε ένας ενσωματωμένος χειριστής αποθήκευσης που παρέχεται από προεπιλογή ή με επεκτάσεις PHP (όπως SQLite ή memcached) ή μπορεί να είναι προσαρμοσμένος χειριστής όπως ορίζεται από το `session_set_save_handler ()`. Το callback ανάγνωσης θα ανακτήσει όλα τα υπάρχοντα δεδομένα συνεδρίας (αποθηκεύονται σε μια ειδική συνέχεια μορφή) και θα χρησιμοποιηθεί για να συμπληρωθεί αυτόματα ο `$_SESSION` πίνακας όταν το callback ανάγνωσης επιστρέψει τα αποθηκευμένα δεδομένα συνεδρίας.

Session_destroy

Το `session_destroy` καταστρέφει όλα τα σχετιζόμενα δεδομένα της τρέχουσας συνεδρίας. Δεν σβήνει τις καθολικές μεταβλητές που έχουν χρησιμοποιηθεί από τη συνεδρία και δεν διαγράφει το cookie της συνεδρίας.

Για να σταματήσει τη συνεδρία, δηλαδή να αποσυνδέσει το χρήστη, το αναγνωριστικό του χρήστη πρέπει να διαγραφεί. Αν έχει χρησιμοποιηθεί cookie για τη μετάδοση του αναγνωριστικού του χρήστη πρέπει να διαγραφεί και αυτό. Η συνάρτηση `setcookie()` μπορεί να χρησιμοποιηθεί.

\$_SESSION[]

Ένας καθολικός πίνακας που περιέχει μεταβλητές της τρέχουσας συνεδρίας που είναι διαθέσιμες στον κώδικα.

Unset()

Η `unset` καταστρέφει τις μεταβλητές που θα της δώσουμε.

Η συμπεριφορά της `unset` μπορεί να διαφέρει, ανάλογα με τον τύπο της μεταβλητής που θέλουμε να καταστρέψουμε.

Αν μια καθολική μεταβλητή γίνει `unset()` μέσα σε μία συνάρτηση, μόνο η τοπική έκδοσή της θα καταστραφεί. Η μεταβλητή στο περιβάλλον κλήσης θα παραμείνει όπως πριν.

\$_SERVER[]

Ο `$_SERVER[]` είναι ένας πίνακας που περιέχει πληροφορίες, όπως headers, μονοπάτια και τοποθεσίες script. Οι καταχωρήσεις σε αυτόν τον πίνακα δημιουργούνται από τον διακομιστή.

Post

Η μέθοδος POST μεταφέρει δεδομένα μέσω επικεφαλίδες HTTP. Οι πληροφορίες κωδικοποιούνται όπως περιγράφεται στην περίπτωση της μεθόδου GET και τοποθετούνται σε μία επικεφαλίδα που ονομάζεται QUERY_STRING.

- Η POST μέθοδος δεν έχει περιορισμό ως προς το μέγεθος δεδομένων που θα αποσταλούν.
- Η POST μέθοδος μπορεί να χρησιμοποιηθεί για ASCII δεδομένα όπως και binary δεδομένα.
- Τα δεδομένα που αποστέλλονται από την POST μέθοδο μέσω HTTP κεφαλίδας, οπότε η ασφάλεια εξαρτάται από το πρωτόκολλο HTTP. Χρησιμοποιώντας ασφαλή σύνδεση HTTP, μπορούμε να είμαστε σίγουροι ότι τα δεδομένα μας είναι ασφαλή.
- Η PHP μας παρέχει τον καθολικό πίνακα `$_POST[]`, μέσω του οποίου μπορούμε να αποστέλλουμε δεδομένα χρησιμοποιώντας POST μέθοδο.

`$_POST[]`

Ένας καθολικός πίνακας που παρέχεται από την PHP, έτσι ώστε να μπορεί να χρησιμοποιηθεί η μέθοδος POST σε μία διαδικτυακή εφαρμογή.

Get

Η μέθοδος GET αποστέλλει την κωδικοποιημένη πληροφορία του χρήστη επικολλημένη στην αίτηση για τη σελίδα. Η αίτηση για τη σελίδα και η κωδικοποιημένη πληροφορία διαχωρίζονται με τον χαρακτήρα `?`.

- Η μέθοδος GET παράγει μία μακριά συμβολοσειρά που εμφανίζεται στα logs του διακομιστή.
- Η μέθοδος GET περιορίζει το χρήστη στους 1024 χαρακτήρες
- Ποτέ δεν χρησιμοποιούμε τη μέθοδο GET για να στείλουμε κωδικούς ή άλλα ευαίσθητα δεδομένα στον διακομιστή.
- Τα δεδομένα που στέλνονται από την GET μέθοδο μπορούν να προσπελαστούν χρησιμοποιώντας τη μεταβλητή περιβάλλοντος QUERY_STRING.
- Η PHP μας παρέχει τον καθολικό πίνακα `$_GET[]`, μέσω του οποίου μπορούμε να αποστέλλουμε δεδομένα χρησιμοποιώντας GET μέθοδο.

`$_GET[]`

Ένας καθολικός πίνακας από μεταβλητές που αποστέλλονται σε μία σελίδα μέσω του URL.

Include, require

Η εντολή include περιλαμβάνει και αξιολογεί το συγκεκριμένο αρχείο που ζητείται. Τα αρχεία περιλαμβάνονται βάσει του μονοπατιού που δίνεται, ή αν δε δίνεται, βάσει του μονοπατιού που προσδιορίζεται. Αν το αρχείο δεν βρεθεί στο μονοπάτι που έχει δοθεί, η include θα κοιτάξει στο φάκελο του αρχείου που την εκτελεί και στον τρέχον φάκελο πριν αποτύχει. Η εντολή include θα βγάλει μια προειδοποίηση αν δεν μπορεί να βρει ένα αρχείο. Αυτή είναι διαφορετική συμπεριφορά από την εντολή **require**, η οποία αν δεν βρει ένα αρχείο θα βγάλει fatal error.

Εάν έχει οριστεί μονοπάτι, είτε absolute μονοπάτι (που ξεκινάει με `\` ή γράμμα σκληρού δίσκου σε περιβάλλον Windows, είτε `/` σε περιβάλλοντα Unix/Linux), είτε σχετικό μονοπάτι με τον τρέχοντα φάκελο (που να ξεκινάει με `.` ή `..`) το `include_path` θα αγνοηθεί.

Mysqli

Η επέκταση ***mysqli*** μας επιτρέπει να έχουμε πρόσβαση σε λειτουργικότητα που παρέχεται από MySQL 4.1 και πάνω. Αντίθετα από την επέκταση ***mysql***, η ***mysqli*** δεν παρέχει ξεχωριστή συνάρτηση για άνοιγμα μόνιμης σύνδεσης. Για άνοιγμα μόνιμης σύνδεσης πρέπει να κολλήσουμε το ***p:*** στο hostname όταν πραγματοποιούμε σύνδεση.

Το πρόβλημα με τις μόνιμες συνδέσεις είναι ότι μπορεί να μείνουν σε απρόβλεπτες καταστάσεις από τους χρήστες. Για παράδειγμα ένα κλείδωμα πίνακα μπορεί να συμβεί πριν ο client τερματίσει απροσδόκητα. Ένας νέος client που θα επιχειρήσει μόνιμη σύνδεση θα την παραλάβει όπως είναι. Οποιοδήποτε "καθάρισμα" θα έπρεπε να γίνει από τον καινούριο client πριν να μπορέσει να χρησιμοποιήσει την μόνιμη σύνδεση, κάτι που θα επιβάρυνε τον προγραμματιστή.

Η μόνιμη σύνδεση της επέκτασης ***mysqli***, παρέχει παρόλα αυτά ενσωματωμένο κώδικα "καθαρισμού". Ο "καθαρισμός" πραγματοποιείται από την επέκταση ***mysqli*** και περιλαμβάνει:

- Rollback των ενεργών συναλλαγών.
- Κλείσιμο και διαγραφή των προσωρινών πινάκων.
- Ξεκλείδωμα των πινάκων.
- Επαναφορά των μεταβλητών συνεδρίας.
- Κλείσιμο των προετοιμασμένων δηλώσεων.
- Κλείσιμο του χειριστή.
- Απελευθέρωση των κλειδωμάτων που πραγματοποιήθηκαν με την εντολή GET_LOCK().

Αυτό εξασφαλίζει ότι οι μόνιμες συνδέσεις βρίσκονται σε μία "καθαρή" κατάσταση επιστρέφοβτας από τη σύνδεση, πριν ο client τις χρησιμοποιήσει.

Η επέκταση ***mysqli*** πραγματοποιεί αυτό το "καθάρισμα" καλώντας αυτόματα την συνάρτηση ***mysql_change_user()***.

Mysqli_connect

Ανοίγει μία καινούρια σύνδεση με τον MySQL server.

Mysqli_set_charset

Προσδιορίζει το προεπιλεγμένο charset που θα χρησιμοποιηθεί όταν στέλνουμε δεδομένα από και προς τον server.

Mysqli_query

Πραγματοποιεί επερώτηση προς τη βάση δεδομένων.

Mysqli_fetch_array

Επιστρέφει μία σειρά από τη βάση δεδομένων ως αποτέλεσμα επερώτησης, που έχει τη μορφή πίνακα.

Mysqli_close

Κλείνει μία σύνδεση με βάση δεδομένων που έχει ανοιχτεί πρόσφατα.

Real_escape_string

Αυτή η συνάρτηση προσθέτει έναν χαρακτήρα, το backslash, \, πριν από πιθανώς επικίνδυνους χαρακτήρες μέσα σε μία συμβολοσειρά. Οι χαρακτήρες αυτοί είναι:

- \x00
- \n
- \r
- \

- '
- "
- \x1a

Αυτό μπορεί να προλάβει επιθέσεις sql injection που συνήθως πραγματοποιούνται προσθέτοντας το χαρακτήρα ' για να επικολλήσουμε κακόβουλο κώδικα σε μία επερώτηση.

Die

Αυτή η συνάρτηση τυπώνει ένα μήνυμα και πραγματοποιεί έξοδο από τον τρέχοντα κώδικα.

Header

Η header() χρησιμοποιείται για να στείλουμε μία κεφαλίδα HTTP. Η header() πρέπει να καλεστεί πριν αποσταλεί οποιαδήποτε έξοδος, είτε από κανονικές ετικέτες HTML, κενές γραμμές σε ένα αρχείο, ή από PHP. Ένα πολύ κοινό λάθος είναι διαβάζουμε κώδικα που περιέχει **include**, **require**, συναρτήσεις, ή μία λειτουργία που να έχει πρόσβαση σε κώδικα άλλης σελίδας, ή να έχουμε κενά, ή κενές γραμμές που τυπώνονται πριν καλεστεί η header(). Το ίδιο πρόβλημα υπάρχει και όταν χρησιμοποιούμε ένα αρχείο PHP/HTML μόνο.

Echo

Εκτυπώνει όλες τις παραμέτρους. Δεν τυπώνεται επιπλέον χαρακτήρας νέας γραμμής.

Η echo δεν είναι συνάρτηση στην πραγματικότητα, οπότε δεν χρησιμοποιούμε παρενθέσεις για να την καλέσουμε. Επίσης δεν συμπεριφέρεται σαν συνάρτηση, οπότε δεν μπορεί πάντα να χρησιμοποιηθεί στο context μίας συνάρτησης. Επιπρόσθετα αν θέλουμε να περάσουμε παραπάνω από μία παραμέτρους στην echo, οι παράμετροι πρέπει να μην εσωκλείονται σε παρενθέσεις.

Η echo έχει τη σύνταξη ενός shortcut και μπορούμε να ακολουθήσουμε απευθείας το άνοιγμα με το χαρακτήρα '='.

MYSQL

SELECT

Η εντολή SELECT της MYSQL χρησιμοποιείται για να φέρει δεδομένα από μία βάση δεδομένων. Έχει τη γενική μορφή:

```
SELECT field1, field2,...fieldN  
table_name1, table_name2...
```

```
[WHERE Clause][OFFSET M ][LIMIT N]
```

- Μπορούμε να χρησιμοποιήσουμε έναν ή παραπάνω πίνακες χωρισμένους από κόμμα για να συμπεριλάβουμε διάφορες συνθήκες με μία εντολή WHERE, αλλά η WHERE είναι προαιρετικό μέρος μίας εντολής SELECT.
- Μπορούμε να φέρουμε μία ή παραπάνω στήλες με μία εντολή SELECT.
- Μπορούμε να δηλώσουμε τον χαρακτήρα * αντί για στήλες. Η SELECT θα επιστρέψει όλες τις στήλες.
- Μπορούμε να δηλώσουμε οποιαδήποτε συνθήκη χρησιμοποιώντας την εντολή WHERE.
- Μπορούμε να προσδιορίσουμε ένα offset χρησιμοποιώντας το OFFSET, από όπου η SELECT θα αρχίσει να επιστρέφει αποτελέσματα.
- Μπορούμε να περιορίσουμε τον αριθμό των αποτελεσμάτων χρησιμοποιώντας το LIMIT.

WHERE

Έχουμε δει την εντολή SELECT που χρησιμοποιείται για να φέρουμε δεδομένα από έναν πίνακα MYSQL. Μπορούμε να χρησιμοποιήσουμε μία προαιρετική εντολή WHERE για να φιλτράρουμε τα δεδομένα μας. Χρησιμοποιώντας αυτή την εντολή, μπορούμε να προσδιορίσουμε κριτήρια επιλογής για να επιλεγθούν συγκεκριμένες σειρές από έναν πίνακα.

Σύνταξη:

```
SELECT field1, field2,...fieldN  
table_name1, table_name2..  
[WHERE condition1 [AND [OR]] condition2.....
```

- Μπορούμε να χρησιμοποιήσουμε έναν ή παραπάνω πίνακες χωρισμένους από κόμμα για να συμπεριλάβουμε διάφορες συνθήκες με μία εντολή WHERE, αλλά η WHERE είναι προαιρετικό μέρος μίας εντολής SELECT.
- Μπορούμε να προσδιορίσουμε οποιαδήποτε συνθήκη χρησιμοποιώντας μία εντολή WHERE,
- Μπορούμε να προσδιορίσουμε μία ή παραπάνω συνθήκες χρησιμοποιώντας τα **AND** και **OR**.
- Η επιλογή WHERE μπορεί να χρησιμοποιηθεί επίσης και σε μία εντολή DELETE ή UPDATE.

Η επιλογή WHERE λειτουργεί περίπου όπως η εντολή if σε μία γλώσσα προγραμματισμού.

LIKE

Έχουμε δει την εντολή SELECT που χρησιμοποιείται για να φέρουμε δεδομένα από έναν πίνακα MYSQL. Μπορούμε επίσης να χρησιμοποιήσουμε την επιλογή WHERE.

Η επιλογή WHERE λειτουργεί μια χαρά όταν θέλουμε να ελέγξουμε για ακριβή ισότητα.

Όπως:

```
If "author == kazantzakis"
```

Αλλά μπορεί για παράδειγμα να θέλουμε να φιλτράρουμε τα αποτελέσματά μας όπου το author απλά να περιέχει το "akis". Αυτό μπορεί να πραγματοποιηθεί χρησιμοποιώντας την επιλογή LIKE μαζί με την επιλογή WHERE.

Αν η επιλογή LIKE χρησιμοποιηθεί με τον χαρακτήρα %, θα λειτουργήσει σαν μεταχαρακτήρας (*) σε ένα σύστημα UNIX.

Σύνταξη:

```
SELECT field1, field2,...fieldN  
table_name1, table_name2..  
WHERE field1 LIKE condition1 [AND [OR]] field2 = 'somevalue'
```

- Χρησιμοποιούμε το LIKE μαζί με το WHERE.
- Μπορούμε να χρησιμοποιήσουμε την επιλογή LIKE αντί για τη σήμα της ισότητας.
- Μπορούμε να προσδιορίσουμε μία ή παραπάνω συνθήκες χρησιμοποιώντας τα **AND** και **OR**.
- Η επιλογή WHERE... LIKE μπορεί να χρησιμοποιηθεί επίσης και σε μία εντολή DELETE ή UPDATE.

INSERT INTO

Για να εισάγουμε δεδομένα σε έναν πίνακα MYSQL, πρέπει να χρησιμοποιήσουμε την εντολή INSERT INTO. Μπορούμε να εισάγουμε δεδομένα χρησιμοποιώντας τη γραμμή εντολών mysql>, ή χρησιμοποιώντας PHP.

Σύνταξη:

```
INSERT INTO table_name ( field1, field2,...fieldN )  
VALUES ( value1, value2,...valueN );
```

UPDATE

Μπορεί να χρειαστεί να τροποποιήσουμε ήδη υπάρχοντα δεδομένα σε μία βάση δεδομένων. Αυτό μπορεί να γίνει χρησιμοποιώντας την εντολή UPDATE. Αυτή θα τροποποιήσει οποιαδήποτε τιμή, οποιουδήποτε πίνακα MYSQL.

Σύνταξη:

```
UPDATE table_name  
SET field1=new-value1, field2=new-value2  
[WHERE Clause]
```

- Μπορούμε να τροποποιήσουμε ένα ή παραπάνω πεδία ταυτόχρονα.
- Μπορούμε να προσδιορίσουμε οποιαδήποτε συνθήκη χρησιμοποιώντας την επιλογή WHERE.
- Επιτρέπεται να τροποποιήσουμε τιμές μόνο σε έναν πίνακα τη φορά.

DELETE

Αν θέλουμε να διαγράψουμε οποιαδήποτε καταχώρηση από έναν πίνακα MYSQL χρησιμοποιούμε την εντολή DELETE FROM.

Σύνταξη:

```
DELETE FROM table_name  
[WHERE Clause]
```

- Αν δεν προσδιορίσουμε τι θέλουμε να διαγραφεί με την επιλογή WHERE, τότε θα διαγραφεί ολόκληρος ο πίνακας.
- Μπορούμε να προσδιορίσουμε οποιαδήποτε συνθήκη χρησιμοποιώντας την επιλογή WHERE.
- Επιτρέπεται να τροποποιήσουμε τιμές μόνο σε έναν πίνακα τη φορά.

HTML

<form>

Η ετικέτα <form> χρησιμοποιείται για να ορίσει μία HTML φόρμα για είσοδο χρήστη.

Το στοιχείο <form> μπορεί να περιέχει ένα ή παραπάνω από τα παρακάτω στοιχεία:

- <input>
- <textarea>
- <button>
- <select>
- <option>
- <optgroup>
- <fieldset>
- <label>

<script>

Η ετικέτα <script> χρησιμοποιείται για να ορίσει κώδικα απ τη μεριά του client (javascript).

Το στοιχείο <script> είτε περιέχει απευθείας κώδικα, είτε δείχνει σε ένα άλλο αρχείο μέσω της ιδιότητας src.

Κοινές χρήσεις για τη javascript είναι η επεξεργασία εικόνας, επικύρωση φόρμας και δυναμικές αλλαγές περιεχομένου.

<div>

Η ετικέτα <div> ορίζει ένα κομμάτι ή τομέα του εγγράφου HTML.

Η ετικέτα `<div>` χρησιμοποιείται για να ομαδοποιήσει στοιχεία τα οποία θα διαμορφωθούν με CSS.

``

Η ετικέτα `` ορίζει μία μη ταξινομημένη λίστα.
Χρησιμοποιούμε την ετικέτα `` μαζί με την ετικέτα ``.

``

Η ετικέτα `` ορίζει ένα στοιχείο λίστας.
Χρησιμοποιείται σε μη ταξινομημένες λίστες, σε ταξινομημένες λίστες και σε λίστες μενού.

`<h1 - h6>`

Οι ετικέτες `<h1>` - `<h6>` χρησιμοποιούνται για να ορίσουν κεφαλίδες HTML.
Η `<h1>` ορίζει την πιο σημαντική κεφαλίδα, ενώ η `<h6>` την λιγότερο σημαντική.

`<?php?>`

Οτιδήποτε έξω από ένα ζευγάρι ετικετών ανοίγματος και κλεισίματος PHP αγνοείται από την PHP, κάτι το οποίο επιτρέπει στα αρχεία PHP να έχουν μικτό περιεχόμενο. Αυτό επιτρέπει στην PHP να ενσωματώνεται σε έγγραφα HTML, για παράδειγμα για τη δημιουργία προτύπων. Αυτό λειτουργεί αναμενόμενα, επειδή όταν ο διερμηνέας PHP συναντήσει την ετικέτα κλεισίματος `?>`, απλά αρχίζει να τυπώνει οτιδήποτε βρει (εκτός από μία καινούρια γραμμή που ακολουθεί αμέσως), μέχρι να συναντήσει ξανά την ετικέτα ανοίγματος `<?php`, εκτός αν βρίσκεται στη μέση μίας εξαρτώμενης εντολής.

`<nav>`

Η ετικέτα `<nav>` ορίζει ένα σετ από συνδέσμους περιήγησης.

Παρατηρούμε δεν μπορούν να μπουν όλοι οι σύνδεσμοι ενός εγγράφου μέσα σε ένα στοιχείο `<nav>`. Το στοιχείο `<nav>` χρησιμοποιείται μόνο στο κύριο μπλοκ συνδέσμων περιήγησης.

Περιηγητές όπως αναγνώστες οθόνης για χρήστες με ειδικές ανάγκες, μπορούν να χρησιμοποιήσουν αυτό το στοιχείο για να παραλείψουν ή όχι την αρχική απόδοση του περιεχομένου.

2.1.2 ΘΕΩΡΙΑ SERVER

2.1.2.1 WEB SERVERS

Ένας web server είναι μια τεχνολογία που επεξεργάζεται τα αιτήματα μέσω HTTP, που είναι το βασικό πρωτόκολλο δικτύου που χρησιμοποιείται για τη διανομή πληροφοριών σχετικά με τον παγκόσμιο ιστό. Ο όρος μπορεί να αναφέρεται είτε σε ολόκληρο το σύστημα του υπολογιστή, ή μια συσκευή, ή ειδικά στο λογισμικό που δέχεται και εποπτεύει τα αιτήματα HTTP.

Η κύρια λειτουργία ενός web server είναι να αποθηκεύει, να επεξεργάζεται και να παραδίδει τις ιστοσελίδες στους clients. Η επικοινωνία μεταξύ του client και του server γίνεται χρησιμοποιώντας το πρωτόκολλο Hypertext Transfer Protocol (HTTP). Οι σελίδες που παραδίδονται είναι πιο συχνά HTML έγγραφα, τα οποία μπορεί να περιλαμβάνουν εικόνες και αρχεία CSS, εκτός από το περιεχόμενο του κειμένου.

Ένας χρήστης, συνήθως ένας web browser, ξεκινά την επικοινωνία με την υποβολή αίτησης για μια συγκεκριμένη πηγή, χρησιμοποιώντας HTTP και ο διακομιστής απαντά με το περιεχόμενο του εν λόγω πόρου ή ένα μήνυμα λάθους, εάν δεν μπορεί να το πράξει. Ο πόρος είναι συνήθως ένα πραγματικό αρχείο στον δευτερογενή χώρο αποθήκευσης του διακομιστή, αλλά αυτό δεν είναι απαραίτητο και εξαρτάται από το πώς υλοποιείται ο διακομιστής.

Ενώ η κύρια λειτουργία του είναι να εξυπηρετήσει αποστέλλοντας περιεχόμενο, η πλήρης εφαρμογή του HTTP περιλαμβάνει επίσης τρόπους για τη λήψη περιεχομένου από τους clients. Η λειτουργία αυτή χρησιμοποιείται για την υποβολή φορμών ιστών, συμπεριλαμβανομένου και του φόρτωματος των αρχείων.

Σε γενικές γραμμές, όλα τα μηχανήματα στο Διαδίκτυο μπορούν να κατηγοριοποιηθούν σε δύο τύπους: τους servers και τους clients. Αυτές οι μηχανές που παρέχουν υπηρεσίες (όπως Web servers ή FTP servers) σε άλλες μηχανές είναι servers. Τα μηχανήματα που χρησιμοποιούνται για τη σύνδεση με τις υπηρεσίες αυτές είναι οι clients. Όταν συνδέεστε στο Yahoo! με www.yahoo.com για να διαβάσετε μια σελίδα, το Yahoo! παρέχει μια μηχανή (πιθανότατα ένα σύμπλεγμα πολύ μεγάλων μηχανημάτων), για χρήση στο διαδίκτυο, για να εξυπηρετήσει το αίτημά σας. Δηλαδή παρέχει ένα server. Το μηχανήμα σας, δεν μπορεί να παρέχει υπηρεσίες σε οποιονδήποτε άλλον στο Διαδίκτυο. Ως εκ τούτου, είναι ένα μηχανήμα του χρήστη, επίσης γνωστή ως client. Είναι δυνατό και συνηθισμένο για μια μηχανή να είναι και server και client, αλλά για τους σκοπούς μας εδώ μπορούμε να σκεφτούμε τα περισσότερα μηχανήματα είτε ως το ένα, είτε ως το άλλο.

Ένας server μπορεί να παρέχει μία ή περισσότερες υπηρεσίες στο Διαδίκτυο. Για παράδειγμα, μπορεί να έχει σε λειτουργία το λογισμικό που του επιτρέπει να ενεργεί ως Web server, ένα email server και έναν FTP server. Οι πελάτες που εξυπηρετούνται από ένα server έχοντας μία συγκεκριμένη εργασία, κατευθύνουν τα αιτήματά τους σε ένα συγκεκριμένο λογισμικό που τρέχει στο συνολικό server. Για παράδειγμα, εάν εκτελούμε ένα πρόγραμμα περιήγησης στο Web στον υπολογιστή μας, πιθανότατα θα θέλουμε να μιλήσουμε με το Web server. Αν εκτελούμε μία εφαρμογή Telnet θα θέλουμε να μιλήσουμε με τον Telnet server, ενώ μια εφαρμογή για e-mail θα μιλήσει στον e-mail server, και ούτω καθεξής...

2.1.2.2 IP ADDRESSES

Σε κάθε μηχανήμα στο Διαδίκτυο εκχωρείται μια μοναδική διεύθυνση που ονομάζεται διεύθυνση IP. IP, σημαίνει Internet Protocol, και αυτές οι διευθύνσεις είναι αριθμοί των 32-bit, που συνήθως εκφράζονται ως τέσσερις «οκτάδες» που είναι διακεκομμένες δεκαδικούς αριθμούς. Μια τυπική διεύθυνση IP μοιάζει με αυτό:

72.14.192.0

Η οποία αντιστοιχεί στο www.google.com.

Τα 4 γκρουπ αριθμών που βρίσκεται η διεύθυνση IP λέγονται octets επειδή μπορούν να έχουν τιμές μεταξύ 0 και 255 , το οποίο είναι 2^8 δυνατότητες ανά οκτάδα .

Κάθε μηχανήμα στο Internet έχει μια μοναδική διεύθυνση IP . Ένας server έχει μια στατική διεύθυνση IP που δεν αλλάζει πολύ συχνά . Ένα μηχανήμα στο σπίτι που καλείτε μέσα από ένα modem έχει μια διεύθυνση IP που έχει εκχωρηθεί από τον ISP . Όταν το μηχανήμα επιλέγει τη διεύθυνση IP είναι μοναδική για την εν λόγω συνεδρία . Μπορεί να είναι διαφορετική την επόμενη φορά που θα το καλέσει για συνεδρία. Έτσι, ένας ISP χρειάζεται μόνο μία διεύθυνση IP για κάθε μόντεμ που υποστηρίζει , αντί για κάθε πελάτη.

2.1.2.3 DOMAIN NAMES

Επειδή οι περισσότεροι άνθρωποι έχουμε πρόβλημα στο να θυμόμαστε τα αριθμούς που συνθέτουν διευθύνσεις IP , και επειδή οι διευθύνσεις IP μερικές φορές αλλάζουν, όλοι οι servers στο Διαδίκτυο χρησιμοποιούν ονόματα που μπορούν να διαβαστούν από τους ανθρώπους, που ονομάζονται DOMAIN NAMES. Για παράδειγμα , `www.google.com` είναι μια διεύθυνση αναγνώσιμη από τον άνθρωπο. Είναι πιο εύκολο για τους περισσότερους από εμάς να θυμόμαστε `www.google.com` απ ό,τι είναι να θυμόμαστε `72.14.192.0` .

2.1.2.4 NAME SERVERS

Μια σειρά από servers που ονομάζονται Domain Name Servers (DNS) αντιστοιχούν τα αναγνώσιμα ονόματα στις διευθύνσεις IP . Αυτοί οι servers είναι απλές βάσεις δεδομένων που αντιστοιχούν ονόματα σε IP διευθύνσεις , και διανέμονται σε όλο το Διαδίκτυο . Οι περισσότερες εταιρείες , οι ISP και τα πανεπιστήμια διατηρούν μικρούς Name Servers για την αντιστοίχιση ονομάτων κεντρικού υπολογιστή σε διευθύνσεις IP .

2.1.2.5 PORTS

Κάθε server παρέχει τις υπηρεσίες του στο διαδίκτυο χρησιμοποιώντας αριθμημένες πόρτες μία για κάθε υπηρεσία που είναι διαθέσιμες στον server. Για παράδειγμα , αν ένα μηχανήμα server που τρέχει ένα server δικτύου και έναν FTP server , συνιθίζεται ο web server να είναι διαθέσιμος στην πόρτα 80 και ο FTP στην πόρτα 21. Οι πελάτες για να συνδεθούν σε μια υπηρεσία χρησιμοποιούν συγκεκριμένη IP και συγκεκριμένη πόρτα.Κάθε μία από τις γνωστές υπηρεσίες είναι διαθέσιμες σε μία γνωστή πόρτα.

2.1.2.6 SECURITY

Οι περισσότεροι servers προσθέτουν κάποιο επίπεδο ασφάλειας στην εξυπηρέτηση διαδικασίας. Παραδείγματος χάριν, αν έχετε πάει σε μία σελίδα και ο browser έβγαλε ένα παράθυρο ζητώντας όνομα χρήστη και κωδικό , τότε έχετε συναντήσει μία σελίδα προστατευμένη με κωδικό.Ο server επιτρέπει στον ιδιοκτήτη της σελίδας να διατηρεί μία λίστα με ονόματα χρήστη και κωδικούς πρόσβασης για τους χρήστες που επιτρέπεται να έχουν πρόσβαση στην σελίδα. Οι πιά ανεπτυγμένοι server προσθέτουν παραπάνω επίπεδα ασφαλείας όπως κρυπτογραφημένη σύνδεση μεταξύ server και client έτσι ώστε ευαίσθητες πληροφορίες όπως αριθμοί πιστικής κάρτας να μπορούν να σταλούν στο διαδίκτυο.

2.1.2.7 DYNAMIC PAGES

Κάθε βιβλίο επισκεπτών σας επιτρέπει να εισάγετε ένα μήνυμα σε μορφή HTML , και την επόμενη φορά που το βιβλίο επισκεπτών προβάλλεται , η σελίδα θα περιέχει τη νέα καταχώρηση .Κάθε μηχανή αναζήτησης σας επιτρέπει να εισάγετε λέξεις-κλειδιά σε μια μορφή HTML , και στη συνέχεια δημιουργεί δυναμικά μια σελίδα με βάση τις λέξεις-κλειδιά που εισάγετε .Σε όλες αυτές τις περιπτώσεις , ο Web server δεν ψάχνει απλά ένα αρχείο . Στην πραγματικότητα επεξεργάζεται πληροφορίες και δημιουργεί μια σελίδα με βάση τις ιδιαιτερότητες του ερωτήματος. Σε όλες σχεδόν τις περιπτώσεις , ο Web server χρησιμοποιεί κάτι που ονομάζεται CGI scripts για να ολοκληρώσει αυτή την εργασία .

2.1.3 ΘΕΩΡΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

2.1.3.1 ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ-ΣΥΣΧΕΤΙΣΕΩΝ

Το μοντέλο οντοτήτων-συσχετίσεων (μοντέλο Ο/Σ - ER model) είναι ένα αφαιρετικό ιδεατό μοντέλο δεδομένων, τα οποία έχουν καθορισμένη δομή. Στη μηχανική λογισμικού χρησιμοποιείται για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίαση βάσεων δεδομένων, ως μοντέλο δεδομένων ενός συστήματος και των απαιτήσεών του με top-down προσέγγιση. Ένα διάγραμμα που δημιουργείται με αυτή τη διαδικασία σχεδίασης καλείται διάγραμμα οντοτήτων-συσχετίσεων, ή *διάγραμμα Ο/Σ* ή *ΟΣΔ* εν συντομία. Προτάθηκε αρχικά το 1976 από τον Peter Chen, ωστόσο στη συνέχεια επινοήθηκαν πολλές παραλλαγές της διαδικασίας.

Χρήση

Χρησιμοποιείται στο πρώτο στάδιο σχεδίασης ενός συστήματος πληροφοριών, κατά την ανάλυση των απαιτήσεών του. Σκοπός του είναι να περιγράψει τις αναγκαίες πληροφορίες οι οποίες πρόκειται να αποθηκευτούν στη βάση δεδομένων ή τον τύπο τους. Η μοντελοποίηση δεδομένων γίνεται για την περιγραφή των χρησιμοποιούμενων όρων και των σχέσεών τους σε έναν ορισμένο τομέα ενδιαφέροντος. Στην περίπτωση σχεδιασμού ενός συστήματος πληροφοριών, που στηρίζεται σε μια βάση δεδομένων, το εννοιολογικό μοντέλο δεδομένων χαρτογραφείται σε προχωρημένο στάδιο σε ένα λογικό μοντέλο δεδομένων, όπως το σχεσιακό μοντέλο δεδομένων. Το στάδιο αυτό ονομάζεται συνήθως στάδιο λογικού σχεδιασμού. Ύστερα, κατά τη διάρκεια του φυσικού σχεδιασμού το λογικό μοντέλο χαρτογραφείται σε κάποιο φυσικό μοντέλο. Ορισμένες φορές και οι δύο φάσεις αναφέρονται ως "φυσικός σχεδιασμός"

2.1.3.2 ΟΝΤΟΤΗΤΑ

Οντότητα (entity) είναι ένα αντικείμενο ενδιαφέροντος στον πραγματικό κόσμο το οποίο ξεχωρίζει από τα υπόλοιπα.^[5] Μια οντότητα λειτουργεί αφαιρετικά σε έναν πολύπλοκο τομέα. Οντότητες μπορεί να είναι άνθρωποι, μέρη, αντικείμενα, γεγονότα, έννοιες κλπ. **Στιγμιότυπο** (instance) μιας οντότητας είναι μια συγκεκριμένη περίπτωση ενός τύπου οντότητας

Τυπος Οντοτητας

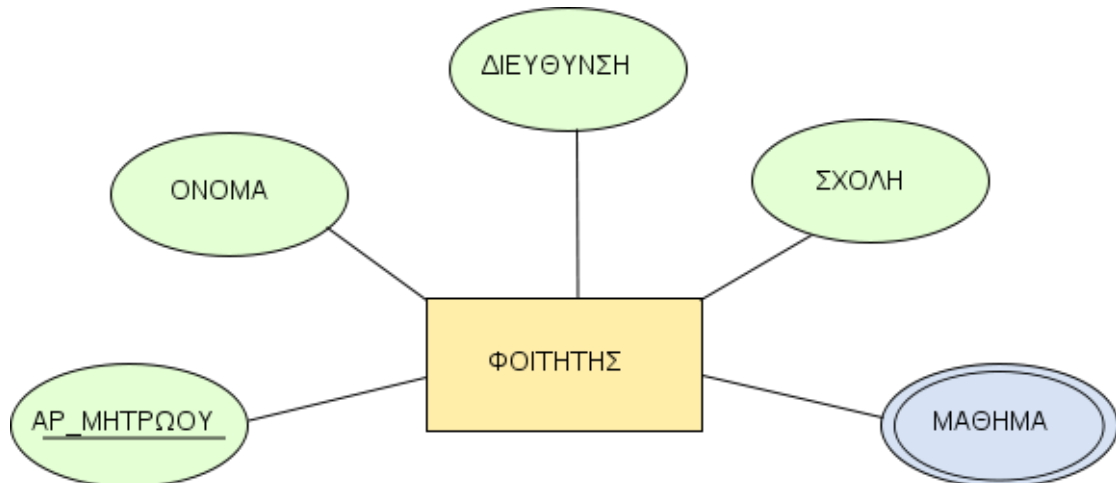
Ο **τύπος** της οντότητας είναι μια συλλογή χαρακτηριστικών που περιγράφουν την οντότητα. Οι διάφοροι τύποι οντοτήτων (π.χ. ΕΡΓΑΖΟΜΕΝΟΣ, ΦΟΙΤΗΤΗΣ) παριστάνονται στο διάγραμμα Ο/Σ με ένα ορθογώνιο.

2.1.3.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Κάθε οντότητα έχει διάφορα στοιχεία που την προσδιορίζουν. Ένα τέτοιο στοιχείο ονομάζεται **ιδιότητα** (attribute), **χαρακτηριστικό** ή **πεδίο** της οντότητας. Τα χαρακτηριστικά χωρίζονται σε:

- **μονότιμα** (single valued), τα οποία έχουν μόνο μια τιμή και
- **πλειότιμα** (multi-valued), τα οποία έχουν σύνολο από τιμές

Στο διάγραμμα Ο/Σ οι ιδιότητες που έχει μια οντότητα παριστάνονται μέσα σε έλλειψη, με υπογραμμισμένο το πρωτεύον κλειδί. Τα πλειότιμα χαρακτηριστικά μιας οντότητας παριστάνονται μέσα σε έλλειψη με διπλό περίγραμμα.



Εικόνα 1: ER διάγραμμα

2.1.3.4 ΣΥΣΧΕΤΙΣΗ

Συσχέτιση (relationship) είναι η σύνδεση δύο ή περισσότερων τύπων οντοτήτων^[5] που παρουσιάζει ενδιαφέρον για σχεδιασμό. Με συσχετίσεις μπορούν να συνδέονται και χαρακτηριστικά οντοτήτων.

Ένας τύπος συσχέτισης (σύνολο συσχετίσεων) παριστάνεται με ρόμβο. Στο εσωτερικό αναγράφεται το όνομα με μικρά γράμματα.

Υποδεικνύουμε τα όρια της συσχέτισης με ένα δείκτη. Ως όρια μπόουμε να συναντήσουμε:

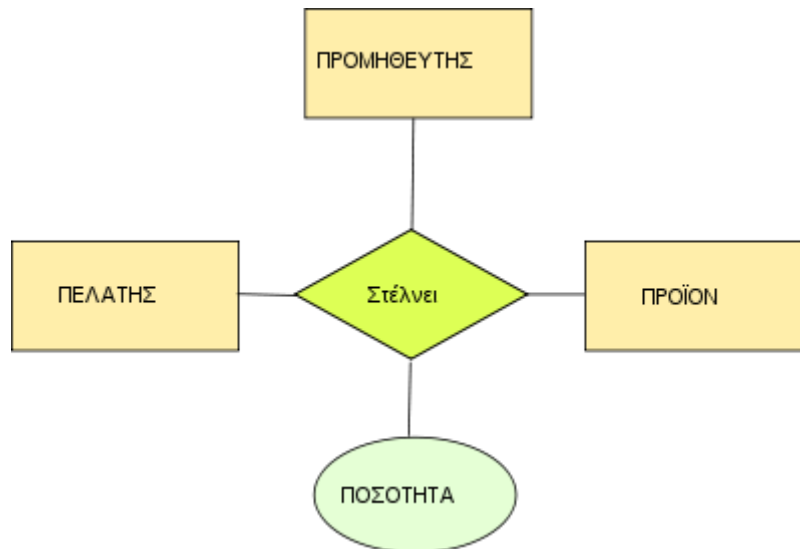
- **0 έως άπειρο** (κατώτατο όριο 0, ανώτατο όριο άπειρο)
- **τουλάχιστον 1** (κατώτατο όριο 1, ανώτατο όριο άπειρο)
- **ακριβώς 1** (κατώτατο όριο 1, ανώτατο όριο 1)
- **το πολύ 1** (κατώτατο όριο 0, ανώτατο όριο 1)

2.1.3.5 ΒΑΘΜΟΣ ΣΥΣΧΕΤΙΣΕΩΝ

Βαθμός συσχέτισης: 3 (Οι τύποι οντοτήτων ΠΡΟΜΗΘΕΥΤΗΣ, ΠΕΛΑΤΗΣ και ΠΡΟΪΟΝ παίρνουν μέρος στη συσχέτιση)

Ο βαθμός μιας συσχέτισης είναι ο αριθμός των τύπων οντοτήτων που παίρνουν μέρος στη συσχέτιση. Οι πιο συνηθισμένες συσχετίσεις είναι

- μοναδικές, ο βαθμός τους τότε είναι 1
- δυαδικές, ο βαθμός τους τότε είναι 2
- τριαδικές, ο βαθμός τους τότε είναι 3



Εικόνα 2: Τριμερής συσχέτιση

2.1.3.6 ΠΛΗΘΙΚΟΤΗΤΑ

Η **πληθικότητα** (cardinality), περιγράφει τον αριθμό στιγμιοτύπων ενός τύπου οντοτήτων που μπορούν να αντιστοιχίζονται με μία οντότητα ενός άλλου τύπου σε μια συσχέτιση.

Ο **λόγος πληθικότητας** ή **πληθικός λόγος** (cardinality ratio), είναι ο λόγος των πληθικοτήτων μιας συσχέτισης.

Μπορούμε να έχουμε συσχετίσεις με λόγο πληθικότητας:

- **1-1 (ένα-προς-ένα)** Αντιστοιχίζεται μια οντότητα ενός τύπου με το πολύ ή ακριβώς μια οντότητα ενός άλλου τύπου.
- **1-N (ένα-προς-πολλά)** Αντιστοιχίζεται μια οντότητα ενός τύπου με κανένα, ένα ή πολλά στιγμιότυπα ενός άλλου τύπου.
- **M-N (πολλά προς πολλά)** Αντιστοιχίζεται κάθε στιγμιότυπο του ενός τύπου με ένα, κανένα ή πολλά στιγμιότυπα του άλλου τύπου.

2.1.3.7 ΑΣΘΕΝΗΣ ΤΥΠΟΣ ΟΝΤΟΤΗΤΑΣ

Αδύναμη ή ασθενής οντότητα λέγεται μια οντότητα που εξαρτάται από την ύπαρξη κάποιας άλλης. Οι αδύναμες οντότητες συμμετέχουν σε συσχετίσεις μέσω ταυτοποιητικών συσχετίσεων με ισχυρή οντότητα.

Ταυτοποιητική συσχέτιση ονομάζεται η συσχέτιση στην οποία το πρωτεύον κλειδί της ισχυρής οντότητας χρησιμοποιείται ως μέρος του πρωτεύοντος κλειδιού της αδύναμης οντότητας.

Διακριτικό ή μερικό κλειδί ονομάζεται το χαρακτηριστικό της αδύναμης οντότητας το οποίο μαζί με το πρωτεύον κλειδί της ισχυρής οντότητας είναι το πρωτεύον κλειδί της αδύναμης.

Κατά την αναπαράσταση αδύναμων οντοτήτων:

- Η οντότητα παριστάνεται με διπλό ορθογώνιο.
- Η ταυτοποιητική συσχέτιση με διπλό ρόμβο.
- Το μερικό κλειδί με διακεκομμένη γραμμή.

2.1.3.8 ΕΠΑΝΑΛΑΜΒΑΝΟΜΕΝΕΣ ΟΜΑΔΕΣ

Μια **επαναλαμβανόμενη ομάδα** είναι ένα σύνολο δύο ή περισσότερων πλειότιμων γνωρισμάτων που έχουν μια λογική σχέση μεταξύ τους.

2.1.3.9 ΠΛΕΙΟΤΙΜΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Είδαμε ότι τα πλειότιμα χαρακτηριστικά παριστάνονται με μια διπλή έλλειψη. Ωστόσο, στην **ιδανική** περίπτωση πλειότιμα χαρακτηριστικά πρέπει να αφαιρούνται από το μοντέλο δεδομένων κατά τη φάση σχεδίασης. Με τον τρόπο αυτό, η σχέση βρίσκεται σε δεύτερη κανονική μορφή.

Για να επιτύχουμε κάτι τέτοιο στο μοντέλο Ο/Σ προσθέτουμε μια ακόμη συσχέτιση.

2.1.3.10 ΥΠΟΚΛΑΣΕΙΣ ΚΑΙ ΥΠΕΡΚΛΑΣΕΙΣ

Γενίκευση / Εξειδίκευση

Με την έννοια **γενίκευση** (generalization) εννοούμε τον εντοπισμό ενός συνόλου οντοτήτων (κλάση) που έχουν κοινά χαρακτηριστικά με πιο γενικευμένα αντικείμενα (υπέρκλαση). Η **εξειδίκευση** (specialization) είναι το ακριβώς αντίθετο της γενίκευσης, δηλαδή ο εντοπισμός υποσυνόλων ενός τύπου οντοτήτων με κοινά χαρακτηριστικά, τα οποία τα διαφοροποιούν από τα υπόλοιπα μέλη του.

Η συσχέτιση μεταξύ κάθε υπόκλασης και υπέρκλασης ονομάζεται ISA συσχέτιση.

2.1.3.11 ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ

Σε κάθε επίπεδο της ιεραρχίας οι τύποι οντοτήτων κληρονομούν τα χαρακτηριστικά των τύπων του αμέσως ανώτερου επιπέδου.

2.1.3.12 ΠΕΡΙΟΡΙΣΜΟΣ ΕΠΙΚΑΛΥΨΗΣ

Όταν υπάρχει περιορισμός επικάλυψης μια οντότητα δεν επιτρέπεται να ανήκει ταυτόχρονα σε δύο υποκλάσεις. (exclusive subtypes)

Ο περιορισμός επικάλυψης συμβολίζεται με μια καμπύλη γραμμή στο διάγραμμα Ο/Σ, που τέμνει την ακμή του τύπου οντοτήτων με κάθε ISA συσχέτιση.

Υπάρχει όμως περίπτωση να μην ισχύει κανένας περιορισμός. Στην περίπτωση αυτή ένα στιγμιότυπο μπορεί να ανήκει σε περισσότερες από μια υποκλάσεις. (non-exclusive subtypes).

2.1.4 ΜΟΝΤΕΛΑ

2.1.4.1 LAMP

Το LAMP είναι μια συντομία για **Linux, Apache, MySQL** και PHP, είναι μια διαδικτυακή πλατφόρμα ανάπτυξης ανοικτού κώδικα που λέγεται και Στοίβα Δικτύου (web stack) η οποία χρησιμοποιεί τα Linux ως λειτουργικό σύστημα, τον Apache ως server δικτύου, την MySQL ως RDBMS και την PHP scripting γλώσσα. (Εναλλακτικά, στην PHP μπορούν να χρησιμοποιηθούν και οι γλώσσες Perl ή Python).

Η κεντρική ιδέα πίσω από το LAMP είναι η χρήση όλων των παραπάνω. Αρχικά, αν και δεν σχεδιάστηκαν για να δουλεύουν μαζί, αυτές οι εναλλακτικές λύσεις λογισμικού ανοικτού κώδικα είναι διαθέσιμες δωρεάν εφόσον καθένα από τα κομμάτια του LAMP είναι ένα παράδειγμα λογισμικού ελεύθερου ή ανοικτού κώδικα (FOSS).

Το LAMP είναι ένα κύριο πρότυπο ανάπτυξης. Σήμερα, όλα τα προϊόντα που αποτελούν το LAMP περιέρχονται από προεπιλογή σχεδόν σε όλες τις διανομές Linux και μαζί απαρτίζουν μία δυναμική πλατφόρμα ανάπτυξης διαδικτυακών εφαρμογών.

2.1.4.2 SQL

Η SQL (αγγλ. αρκτ. από το Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Η SQL ήταν μία από τις πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του το 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων.

Γενικά, με την SQL μπορούμε να :

- Εκτελέσουμε επερωτήσεις σε μία βάση δεδομένων.
- Ανακτήσουμε δεδομένα από μία βάση δεδομένων.
- Να προσθέσουμε εγγραφές σε μία βάση δεδομένων.
- Κάνουμε ενημέρωση των εγγραφών που βρίσκονται στην βάση δεδομένων.
- Διαγράψουμε εγγραφές από την βάση δεδομένων.
- Δημιουργήσουμε νέες βάσεις δεδομένων.

Η SQL αναπτύχθηκε στην IBM από τους Andrew Richardson, Donald C. Messerly και Raymond F. Boyce, στις αρχές της δεκαετίας του 1970. Αυτή η έκδοση, αποκαλούμενη αρχικά SEQUEL, είχε ως σκοπό να χειριστεί και να ανακτήσει τα στοιχεία που αποθηκεύτηκαν στο πρώτο RDBMS της IBM, το System R.

Το πρώτο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ήταν το RDMBS που αναπτύχθηκε στο MIT, στις αρχές της δεκαετίας του 1970 και η Ingres, που αναπτύχθηκε το 1974 στο Πανεπιστήμιο Μπέρκλεϋ. Η Ingres εφάρμοσε μία γλώσσα διατύπωσης ερωτήσεων γνωστή ως QUEL, το οποίο αντικαταστάθηκε αργότερα στην αγορά από την SQL.

Προς το τέλος της δεκαετίας του 70 η Relational Software (τώρα Oracle Corporation) είδε τη δυνατότητα αυτών που περιγράφηκαν από Codd, Chamberlin, και Boyce και αναπτύξε την SQL βασισμένο στο RDBMS, με τις φιλοδοξίες πώλησης του στο Αμερικανικό ναυτικό, την Κεντρική Υπηρεσία Πληροφοριών και άλλες Αμερικανικές Υπηρεσίες.

Το καλοκαίρι του 1979, η Relational Software εισήγαγε την πρώτη διαθέσιμη στο εμπόριο εφαρμογή του SQL και νίκησε την IBM με τη διάθεση του πρώτου εμπορικού RDBMS για μερικές εβδομάδες.

Η γλώσσα SQL υποδιαιρείται σε διάφορα γλωσσικά στοιχεία, που περιλαμβάνουν:

- Clauses, οι οποίες είναι σε μερικές περιπτώσεις προαιρετικές, αλλά απαραίτητα συστατικά των δηλώσεων και ερωτήσεων.
- Expressions που μπορούν να παραγάγουν είτε τις κλιμακωτές τιμές είτε πίνακες που αποτελούνται από στήλες και σειρές στοιχείων.
- Predicates που διευκρινίζουν τους όρους που μπορούν να αξιολογηθούν σαν σωστό ή λάθος.
- Queries που ανακτούν τα στοιχεία βασισμένες σε ειδικά κριτήρια.
- Statements που μπορούν να έχουν μια επίδραση στα σχήματα και τα στοιχεία, ή που μπορούν να ελέγξουν τη ροή του προγράμματος και τις συνδέσεις από άλλα προγράμματα.
- Το κενό αγνοείται γενικά στις Statements και τις Queries SQL. Ένα κενό είναι όμως απαραίτητο για να ξεχωρίζει Statements όπως και στην κανονική γραφή κειμένων.

ΕΠΕΡΩΤΗΜΑΤΑ

Ένα επερώτημα (query) συνήθως αποτελείται από:

```
SELECT *
```

```
FROM Sygrafeas
```

```
where epitheto = 'Solomos'
```

```
and onoma = 'Dionysios'
```

αλλά μπορεί επίσης να περιλαμβάνει GROUP BY, ORDER BY και άλλα:

```
SELECT Book.title,
```

```
count(*) AS No_of_Authors
```

```
FROM Book
```

```
JOIN Book_author ON Book.isbn = Book_author.isbn
```

```
GROUP BY Book.title
```

```
having count(*) > 1
```

```
ORDER BY No_of_Authors
```

2.1.4.3 MYSQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοικτού κώδικα (RDBMS). Το ακρωνύμιο SQL σημαίνει Structured Query Language.

Η MySQL είναι είναι μια δημοφιλής επιλογή για βάσεις δεδομένων για χρήση στις υπηρεσίες web και είναι βασικό μέρος της στοίβας LAMP.

Η MySQL είναι το πιο δημοφιλές σύστημα ανοικτού κώδικα βάσεων δεδομένων επιτρέποντας την οικονομικότερη διανομή αξιόπιστων, υψηλής απόδοσης και κλιμακούμενων διαδικτυακών εφαρμογών που ενσωματώνουν βάσεις δεδομένων περιλαμβάνοντας πολλά γνωστά websites.

Πολλές από τις μεγαλύτερες και ταχύτερα αναπτυσσόμενες εταιρείες του κόσμου συμπεριλαμβανομένων των Facebook , Google , Adobe , Alcatel Lucent και Zappos βασίζονται σε MySQL για να εξοικονομήσουν χρόνο και χρήμα τροφοδοτώντας ιστοσελίδες υψηλού όγκου , και κρίσιμα για την επιχείρησή τους συστήματα και πακέτα λογισμικού .

2.1.4.4 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache HTTP εξυπηρετητής|Apache), ώστε να

παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

Ο συνδυασμός Linux/Apache/MySQL/PHP, που είναι η πιο δημοφιλής πλατφόρμα εκτέλεσης ιστοσελίδων είναι γνωστός και με το ακρωνύμιο LAMP. Παρόμοια, ο συνδυασμός */Apache/MySQL/PHP ονομάζεται *AMP, όπου το πρώτο αρχικό αντιστοιχεί στην πλατφόρμα, στην οποία εγκαθίστανται ο Apache, η MySQL και η PHP (π.χ. Windows, Mac OS X).

Ο LAMP συνήθως εγκαθίσταται και ρυθμίζεται στο Linux με τη βοήθεια του διαχειριστή πακέτων της εκάστοτε διανομής. Στην περίπτωση άλλων λειτουργικών συστημάτων, επειδή το κατέβασμα και η ρύθμιση των ξεχωριστών προγραμμάτων μπορεί να είναι πολύπλοκη, υπάρχουν έτοιμα πακέτα προς εγκατάσταση, όπως το XAMPP και το WAMP για τα Windows και το MAMP για το Mac OS X.

Το ακρωνύμιο της PHP είναι hypertext Preprocessor. Χρησιμοποιείται ευρέως ως γλώσσα scripting ανοικτού λογισμικού ,δηλαδή εκτελείται από την «πλευρά» του server.

Επίσης, η PHP είναι ελεύθερη για «κατέβασμα» και χρήση αλλά παρόλα αυτά είναι αρκετά δυνατή έτσι ώστε να αποτελεί τον πυρήνα στο δημοφιλέστερο blog σύστημα όπως επίσης και στο μεγαλύτερο κοινωνικό δίκτυο, το Facebook.

Επίσης , η php μπορεί να:

- Παράγει περιεχόμενο δυναμικών σελίδων .
- Να δημιουργήσει , να ανοίξει , να διαβάσει , να γράψει , να σβήσει και να κλείσει αρχεία σε ένα server .
- Να συλλέξει δεδομένα.
- Να πραγματοποιήσει αποστολή και λήψη cookies.
- Να προσθέσει, να σβήσει αλλά και γενικά να τροποποιήσει μια βάση δεδομένων.
- Μπορεί να χρησιμοποιηθεί στον έλεγχο πρόσβασης των χρηστών.
- Κάνει κρυπτογράφηση δεδομένων.
- Με την PHP δεν περιορίζεσαι στο να παράγεις μόνο HTML. Μπορείς να παράγεις εικόνες , αρχεία PDF ακόμη και ταινίες Flash. Επίσης μπορείς να παράγεις οποιοδήποτε τύπο κειμένου όπως XHTML και XML.

2.1.4.5 HTML

Η **HTML** (αρχικοποίηση του αγγλικού **HyperText Markup Language**, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από *ετικέτες* (tags), οι οποίες περιλαμβάνονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται *ετικέτα έναρξης* και τη δεύτερη *ετικέτα λήξης* (ή σε άλλες περιπτώσεις *ετικέτα ανοίγματος* και *ετικέτα κλεισίματος* αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Πρώτες προδιαγραφές

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML 4.

Το ίδιο το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τυπογράφων, αλλά εκτός από απλή μίμηση της τυπογραφίας προσθέτει γενικευμένη σήμανση βασισμένη σε στοιχεία, τα οποία μπορούν να εμφωλεύονται το ένα μέσα στο άλλο και να φέρουν ιδιότητες. Ακόμα, το SGML διαχωρίζει τη δομή από το περιεχόμενο, κατεύθυνση προς την οποία αργότερα κινήθηκε και η HTML, με τα CSS. Πολλά από τα στοιχεία κειμένου προέρχονται από την τεχνική αναφορά ISO TR 9537, Techniques for using SGML (τεχνικές χρήσης της SGML), η οποία με τη σειρά της καλύπτει τα χαρακτηριστικά των πρώιμων γλωσσών μορφοποίησης κειμένου που χρησιμοποιούνταν από τα TYPESET και RUNOFF, και είχαν αναπτυχθεί στις αρχές της δεκαετίας του 1960 για το λειτουργικό σύστημα CTSS.

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993[8], η οποία περιλάμβανε και έναν Ορισμό τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της

ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το ανταγωνιστικό πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.

Μετά που τα πρόχειρα HTML και HTML+ έληξαν, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασίζονταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Η αρίθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν. designation was intended to distinguish the new edition from previous drafts.

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης ενδιαφερόντων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με ανάδραση από τους δημιουργούς λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).

Ιστορικό εκδόσεων του προτύπου

Χρονικό εκδόσεων HTML

- 24 Νοεμβρίου 1995. Η HTML 2.0 δημοσιεύτηκε ως IETF RFC 1866. Επιπλέον συμπληρωματικά RFC πρόσθεσαν δυνατότητες:
- 25 Νοεμβρίου 1995: RFC 1867 (ανέβασμα αρχείο από φόρμα)
- Μάιος 1996: RFC 1942 (πίνακες)
- Αύγουστος 1996: RFC 1980 (πελατοκεντικοί χάρτες εικόνων (client-side))
- Ιανουάριος 1997: RFC 2070 (διεθνοποίηση και τοπικοποίηση). Τον Ιούνιο του 2000, όλα τα ανωτέρω χαρακτηρίστηκαν ως ξεπερασμένα ή ιστορικά, από το RFC 2854.
- Ιανουάριος 1997

Η HTML 3.2 δημοσιεύτηκε ως Σύσταση από το W3C. Ήταν η πρώτη έκδοση που αναπτύχθηκε και προτυποποιήθηκε αποκλειστικά από το W3C, μια που το IETF έκλεισε την Ομάδα Εργασίας για την HTML τον Σεπτέμβριο του 1996.

Στην HTML 3.2 αφαιρέθηκαν εντελώς οι μαθηματικοί τύποι και οι υπερκαλύψεις που προέρχονταν από επεκτάσεις του προτύπου από τρίτους φορείς. Υιοθετήθηκαν οι περισσότερες από τις ετικέτες οπτικής διαμόρφωσης του Netscape αλλά παραλείφθηκαν οι ετικέτες blink του Netscape και marquee της Microsoft, μετά από κοινή συναίνεση και των δύο εταιρειών. Για να συμπληρωθεί το κενό από την αφαίρεση της δυνατότητας αναπαράστασης μαθηματικών τύπων, προτυποποιήθηκε το MathML 14 μήνες αργότερα.

- Δεκέμβριος 1997

Δημοσιεύτηκε η HTML 4.0 από το W3C, ως Σύσταση. Παρείχε τρεις εκδοχές:

1. την Strict, στην οποία απαγορεύεται η χρήση ξεπερασμένων στοιχείων,
2. την Transitional, στην οποία επιτρέπονται τα ξεπερασμένα στοιχεία,
3. την Frameset, στην οποία επιτρέπονται μόνο στοιχεία frame.

Αρχικά η έκδοση είχε το κωδικό όνομα «Cougar, Η HTML 4.0 υιοθέτησε πολλά στοιχεία και ιδιότητες που προηγουμένως χρησιμοποιούνταν μόνο από συγκεκριμένους browser, αλλά ταυτόχρονα προσέβλεπε στην σταδιακή κατάργηση των δυνατοτήτων οπτικής διαμόρφωσης του Netscape, σημειώνοντάς τις ως ξεπερασμένες, και προτείνοντας τα CSS στη θέση τους. Η HTML 4 είναι μια εφαρμογή του SGML, σύμφωνη με το πρότυπο ISO 8879 - SGML.[19]

- Απρίλιος 1998

Επανεκδίδεται η HTML 4.0[20] με μικρές αλλαγές, χωρίς να αλλάξει ο αριθμός έκδοσης.

- Δεκέμβριος 1999

Δημοσιεύεται ως Σύσταση του W3C η HTML 4.01[21]. Προσέφερε τις ίδιες τρεις εκδοχές με την HTML 4.0. Τα πιο πρόσφατα λάθη και παραλείψεις (errata) δημοσιεύτηκαν στις 12 Μαΐου 2001.[22]

- Μάιος 2000

Με το ISO/IEC 15445:2000[23][24] προτυποποιείται η ISO HTML, βασισμένη στην HTML 4.01 Strict, ως διεθνές πρότυπο ISO/IEC. Αυτό το πρότυπο εμπίπτει στον τομέα του ISO/IEC JTC1/SC34 για τις γλώσσες περιγραφής και επεξεργασίας εγγράφων.[23]

Από τα μέσα του 2008, η HTML 4.01 και το ISO/IEC 15445:2000 είναι οι πιο πρόσφατες εκδόσεις της HTML. Παράλληλα αναπτύχθηκε η XHTML, βασισμένη στην XML, η οποία απασχόλησε την Ομάδα Εργασίας του W3C για την HTML από την αρχή μέχρι τα μέσα της δεκαετίας του 2000.

Χρονικό εκδόσεων προχειρών της HTML

Το λογότυπο της HTML 5

- Οκτώβριος 1991

Πρώτη δημοσίευση του HTML Tags,[5] ενός ανεπίσημου εγγράφου του CERN αναφέρει δώδεκα ετικέτες HTML.

- Ιούνιος 1992

Πρώτο ανεπίσημο πρόχειρο του HTML DTD,[25] με επτά[26][27][28] αναθεωρήσεις (15 Ιουλίου, 6 Αυγούστου, 18 Αυγούστου, 17 Νοεμβρίου, 19 Νοεμβρίου, 20 Νοεμβρίου, 22 Νοεμβρίου)

- Νοέμβριος 1992

Ανεπίσημο πρόχειρο HTML DTD 1.1 (το πρώτο με αριθμό έκδοσης, βασισμένο στις αναθεωρήσεις RCS, οι οποίες ξεκινούν με 1.1 αντί για 1.0)[28]

- Ιούνιος 1993

Δημοσιεύεται η Hypertext Markup Language από την Ομάδα Εργασίας IIR του IETF ως ένα πρόχειρο για το Διαδίκτυο (κάτι σαν εισήγηση για πρότυπο). Αντικαταστάθηκε από μια δεύτερη έκδοση ένα μήνα αργότερα, ακολουθούμενη από έξι ακόμα πρόχειρα δημοσιευμένα από το ίδιο το IETF τα οποία τελικά οδήγησα στην HTML 2.0 με το RFC1866.

- Νοέμβριος 1993

Δημοσιεύτηκε η HTML+ από το IETF ως Πρόχειρο-Διαδικτύου, και ως ανταγωνιστική πρόταση για το πρόχειρο της Hypertext Markup Language. Έληξε τον Μάιο του 1994.

- Απρίλιος 1995 (γραμμένη τον Μάρτιο του 1995)

Η HTML 3.0 προτάθηκε ως πρότυπο από το IETF, αλλά η πρόταση έληξε πέντε μήνες αργότερα χωρίς άλλη ενεργεια. Περιείχε πολλές από τις δυνατότητες που συμπεριλαμβάνονταν στην πρόταση του Raggett για την HTML+, όπως την υποστήριξη για πίνακες, τη ροή κειμένου γύρω από εικόνες και την προβολή πολύπλοκων μαθηματικών τύπων.

Το W3C ξεκίνησε την ανάπτυξη του δικού του Arena browser ως δοκιμαστική πλατφόρμα για την HTML 3 και για τα CSS αλλά η HTML 3.0 δεν πέτυχε, για διάφορους λόγους. Το πρόχειρο θεωρήθηκε υπερβολικά μακρύ, με 150 σελίδες, και ο ρυθμός ανάπτυξης του browser, καθώς και ο αριθμός των ενδιαφερόμερων μερών υπερέβαιναν τις δυνατότητες του IETF. Οι εταιρείες που διέθεταν browser, συμπεριλαμβανομένων των Microsoft και Netscape εκείνο τον καιρό, επέλεξαν να υλοποιήσουν διαφορετικά υποσύνολα των δυνατοτήτων του προχειρου της HTML 3, καθώς και να εισάγουν τις δικές του επεκτάσεις σ' αυτό. Οι επεκτάσεις αυτές έλεγχαν εμφανισιακά στοιχεία των εγγράφων, αντίθετα με την «πεποίθηση [της ακαδημαϊκής κοινότητας μηχανικών] ότι το χρώμα, το παρασκήνιο, το μέγεθος και ο τύπος της γραμματοσειράς ήταν οπωσδήποτε έξω από το στόχος μιας

γλώσσας της οποίας η μοναδική πρόθεση ήταν να καθορίσει πώς οργανώνεται ένα έγγραφο.» Ο Dave Raggett, συνεργάτης του W3C για πολλά χρόνια, σχολίασε ότι «Μέχρι ενός σημείου, η Microsoft έκτισε την επιχειρηματική της δραστηριότητα στον Ιστό επεκτείνοντας τις δυνατότητες της HTML.»

- Ιανουάριος 2008

Δημοσιεύεται η HTML5 ως ένα Πρόχειρο Εργασίας από το W3C.

Παρότι η σύνταξη μοιάζει αρκετά με το SGML, η HTML5 δεν προσπαθεί πλέον να αποτελεί εφαρμογή του SGML, και ορίζεται ως αυτόνομη, μαζί με την XHTML5 η οποία βασίζεται στην XML Εκδόσεις της XHTML

- Η XHTML είναι ξεχωριστή γλώσσα η οποία ως αναδιαμόρφωση της HTML 4.01 με χρήση της XML 1.0. Συνεχίζει να αναπτύσσεται:
- XHTML 1.0,[39] δημοσιεύτηκε στις 26 Ιανουαρίου 2000, ως Σύσταση του W3C, μετά αναθεωρήθηκε και επανεκδόθηκε στις 1 Αυγούστου 2002. Προσφέρει τις ίδιες τρεις εκδοχές όπως η HTML 4.0 και 4.01, αναδιαμορφωμένες ως XML, με μικρούς περιορισμούς.
- XHTML 1.1,[40] δημοσιεύτηκε στις 31 Μαΐου 2001, ως Σύσταση του W3C. Βασίζεται στην XHTML 1.0 Strict, αλλά περιέχει μικρές αλλαγές, μπορεί να παραμετροποιηθεί, μπορεί να αναμορφωθεί χρησιμοποιώντας Αρθρώματα της XHTML, τα οποία δημοσιεύτηκαν στις 10 Απριλίου 2001, ως Σύσταση του W3C.
- XHTML 2.0,[41][42] Δεν υπάρχει πρότυπο XHTML 2.0, είναι ένα πρόχειρο έγγραφο και θεωρείται ακόμα έργο σε εξέλιξη. Η XHTML 2.0 δεν είναι συμβατή με την XHTML 1.x και επομένως μπορεί πιο σωστά να χαρακτηριστεί ως μια νέα γλώσσα που είναι εμπνευσμένη από την XHTML παρά ως αναβάθμιση της υπάρχουσας XHTML 1.x.
- XHTML5, είναι αναβάθμιση της XHTML 1.x, και εξελίσσεται παράλληλα με την HTML5 στο ίδιο πρόχειρο.

HTML 5

Η HTML5 προορίζεται για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιότητα plug-in και πλούσιες διαδικτυακές εφαρμογές (RIA) όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX.

Ειδικότερα, η HTML5 προσθέτει πολλά νέα συντακτικά χαρακτηριστικά. Αυτά περιλαμβάνουν τα νέα στοιχεία <video>, <audio> και <canvas>, καθώς και την ενσωμάτωση του περιεχομένου Scalable Vector Graphics (SVG) (που αντικαθιστά το γενικό <object> tags) και του MathML για μαθηματικές φόρμουλες. Τα χαρακτηριστικά αυτά είναι σχεδιασμένα ώστε να είναι εύκολο να συμπεριληφθούν και να χειριστούν περιεχόμενο πολυμέσων και γραφικών στο διαδίκτυο, χωρίς να χρειάζεται να καταφύγουμε σε plugins και API κλειστού κώδικα.

2.1.4.6 CSS

Το CSS (Cascading Style Sheets) είναι μια απλή γλώσσα που μας βοηθάει να ορίσουμε με σαφήνεια και ιδιαίτερη ευελιξία τον τρόπο με τον οποίο θα εμφανίζονται τα διάφορα στοιχεία στην ιστοσελίδα μας.

Ποια πλεονεκτήματα έχει η χρήση CSS έναντι της μορφοποίησης μέσω HTML attributes?

- Πολύ μεγαλύτερη ευελιξία. Το CSS κατέστησε εφικτές μορφοποιήσεις οι οποίες ήταν αδύνατες ή πολύ δύσκολες με την κλασική HTML.
- Ευκολότερη συντήρηση των ιστοσελίδων. Η εμφάνιση ενός ολόκληρου site μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο CSS. Έτσι, κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μια μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων σε κάθε σελίδα που υπάρχει στο site.

- Μικρότερο μέγεθος αρχείου, δεδομένου ότι ο κάθε κανόνας μορφοποίησης γράφεται μόνο μια φορά και όχι σε κάθε σημείο που εφαρμόζεται.
- Καλύτερο SEO (Search engine optimization). Οι μηχανές αναζήτησης δεν «μπερδεύονται» ανάμεσα σε περιεχόμενο και τη μορφοποίηση του , αλλά έχουν πρόσβαση στο περιεχόμενο σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν (indexing).
- Γρηγορότερες σελίδες. Όταν χρησιμοποιούμε εξωτερικό αρχείο CSS , ο browser την πρώτη φορά που θα φορτώσει κάποια σελίδα του site μας το αποθηκεύει στην cache, οπότε δεν χρειάζεται να το κατεβάσει ξανά κάθε φορά που κατεβάζει ο χρήστης του κάποια άλλη σελίδα του site μας.

Πως εισάγουμε CSS στη σελίδα μας;

Μπορούμε να εισάγουμε CSS στη σελίδα μας με τρεις διαφορετικούς τρόπους , αναλόγως την περίπτωση. Οι τρόποι αυτοί, κατά σειρά φθίνουσα προτεραιότητας, είναι:

- Inline CSS : Αν επιθυμούμε να μορφοποιήσουμε ένα στοιχείο μόνο , και δεν πρόκειται να χρειαστούμε αυτό το είδος μορφοποίησης για τίποτε άλλο στο site, μπορούμε να γράψουμε «χύμα» (δηλ. χωρίς το κομμάτι του CSS selector και χωρίς αγκύλες) CSS properties μέσα στο attribute style, το οποίο το διαθέτει σχεδόν κάθε στοιχείο html. Παραδείγματος χάριν, για να κάνουμε μια συγκεκριμένη παράγραφο κόκκινη , μπορούμε να προσθέσουμε style="color:red;" στο <p> tag (ολοκληρωμένα: <p style="color:red;">). Αξίζει να σημειωθεί, ότι αν κάποιες από τις ιδιότητες που θα γράψουμε εντός της attribute style ενός στοιχείου «συγκρούονται» με κανόνες CSS που έχουν οριστεί για αυτό αλλού και το αφορούν , τότε ο browser θα επιλέξει να εφαρμόσει αυτά που γράψαμε εντός του attribute style , ως πιο συγκεκριμένα για το στοιχείο αυτό.
- CSS για μια συγκεκριμένη σελίδα : Πολλές φορές , μπορεί να θέλουμε να εφαρμόσουμε κάποιους κανόνες CSS μόνο για τη συγκεκριμένη σελίδα και όχι για όλο το site (ή να διαφοροποιήσουμε σε κάποιες ιδιότητες τους υπάρχοντες). Ένας τρόπος να το κάνουμε αυτό , για να μην δημιουργήσουμε ξεχωριστό αρχείο CSS είναι να εισάγουμε εντός των tags <head>...</head> της σελίδας τους κανόνες CSS μας μέσα σε <style> tags (με την html attribute type τους σε text/css). Παραδείγματος χάριν, για να κάνουμε το χρώμα γραμμμάτων μιας συγκεκριμένης σελίδας γκρι, θα γράφαμε στο head της:

```
<style type="text/css">
  body
  {
    color:gray;
  }
</style>
```

Αξίζει να σημειωθεί ότι συνήθως όταν ξεκινάμε την ανάπτυξη ενός site, είναι πιο βολικό να χρησιμοποιήσουμε αυτό τον τρόπο αρχικά, μιας και πρέπει να επεξεργαζόμαστε μόνο ένα αρχείο αντί για δύο. Όταν τελειώνουμε την πρώτη σελίδα, συνήθως μεταφέρουμε το CSS που έχουμε γράψει σε εξωτερικό αρχείο (βλ. παρακάτω) ώστε να μπορούμε να χρησιμοποιήσουμε τους ίδιους κανόνες CSS και στις άλλες σελίδες του site μας, χωρίς να πρέπει φυσικά να τους κάνουμε copy-paste σε κάθε σελίδα.

- Εξωτερικό αρχείο CSS: Η πιο «σωστή» χρήση του CSS και αυτή που θα έπρεπε να είναι η πρώτη λύση στην οποία θα καταφύγουμε , είναι η χρήση εξωτερικού αρχείου CSS. Για να τη χρησιμοποιήσουμε, γράφουμε τους κανόνες CSS μας σε ένα αρχείο με επέκταση css (πχ main.css) και στο head της κάθε σελίδας του site μας γράφουμε <link href="main.css" type="text/css" /> (αν το όνομα του αρχείου CSS είναι main.css, αλλιώς προφανώς γράφουμε

το σωστό όνομα αρχείου). Αξίζει να σημειωθεί ότι μπ ορούμε να έχουμε πολλά αρχεία css στην ίδια σελίδα, και μεγαλύτερη προτεραιότητα έχει πάντα αυτό που έχει γραφτεί τελευταία.

Βασικές ιδιότητες CSS (CSS properties)

Ιδιότητες που αφορούν μορφοποίηση κειμένου

- **color**
Αφορά το χρώμα του κειμένου, αλλά αν δεν οριστεί χρώμα περιγράμματος (μέσω της ιδιότητας border-color), ο browser χρησιμοποιεί αυτό που ορίστηκε στην ιδιότητα color. Τα χρώματα μπορούν να εισαχθούν είτε σε μορφή RGB (πχ color: rgb(255,128,30);), είτε σε μορφή hex (πχ color: #ff801e;) είτε με τη μορφή κάποιου keyword (πχ color:orange;).
- **font-size**
Αφορά το μέγεθος της γραμματοσειράς. Οι τιμές που δέχεται μπορούν να είναι εκφρασμένες σε ένα μεγάλο πλήθος μονάδων μεγέθους, από τις οποίες οι πιο ευρέως διαδεδομένες είναι τα pixels (πχ font-size: 12px;) και οι στιγμές (πχ font-size:10pt).
- **font-family**
Η ιδιότητα αυτή μας επιτρέπει να ορίσουμε ένα πλήθος γραμματοσειρών που θα χρησιμοποιηθούν για το κείμενο, κατά σειρά προτίμησης. Ουσιαστικά μέσω αυτής ορίζουμε τη γραμματοσειρά του κειμένου, απλά μας επιτρέπει να ορίσουμε και εναλλακτικές επιλογές, ώστε αν η γραμματοσειρά που ορίσαμε δεν υπάρχει στον υπολογιστή του χρήστη, να μην μας καταστρέψει ο browser την εμφάνιση της σελίδας επιλέγοντας όποια αυτός νομίζει, απλά την επόμενη επιλογή μας. Αν καμία από όσες γραμματοσειρές ορίσαμε δεν υπάρχει στον υπολογιστή του χρήστη, τότε ο browser επιλέγει κάποια που να ανήκει στη γενική οικογένεια γραμματοσειρών (generic font-family) που θα ορίσουμε στο τέλος με το κατάλληλο keyword. Τέτοια keywords είναι τα εξής:
 1. serif: Γραμματοσειρές με «πατούρες» όπως η Georgia.
 2. sans-serif: Γραμματοσειρές χωρίς «πατούρες» όπως η Trebuchet MS.
 3. monospace: Γραμματοσειρές όπου το κάθε γράμμα καταλαμβάνει ίσο πλάτος με τα υπόλοιπα, όπως η Courier New.
 4. cursive: Καλλιγραφικές γραμματοσειρές όπως η Monotype Corsiva
 5. fantasy: «Διακοσμητικές» γραμματοσειρές.Οι τελευταίες 2 γενικές οικογένειες γραμματοσειρών χρησιμοποιούνται πολύ σπάνια.
Παράδειγμα χρήσης της ιδιότητας: font-family: Calibri, Trebuchet MS, Verdana, sans-serif;
- **font-style**
Σε αντίθεση με αυτό που θα περίμενε κανείς από μια ιδιότητα με αυτό το όνομα, δηλαδή την εφαρμογή πολλών και διαφόρων «εφέ» στο κείμενο, στην πραγματικότητα αφορά μόνο την περίπτωση όπου το κείμενο θα είναι πλάγιο. Για άλλα εφέ, χρησιμοποιούνται άλλες ιδιότητες, οι οποίες θα εξεταστούν παρακάτω. Οι πιθανές τιμές της είναι normal, italic και oblique. Οι δύο τελευταίες συνήθως κάνουν το ίδιο πράγμα (και είναι προτιμότερη η χρήση της italic μιας και υποστηρίζεται και από Internet Explorer 3, ενώ η υποστήριξη για την oblique ξεκινά από Internet Explorer 4), ενώ είναι προφανής η λειτουργία της τιμής normal.
- **font-weight**
Αφορά το «βάρος» της γραμματοσειράς και στην πράξη χρησιμοποιείται για να ορίσει αν το κείμενο μας θα είναι έντονο ή όχι, μιας και οι περισσότερες γραμματοσειρές που χρησιμοποιούνται στο web διατίθενται μόνο σε δύο βάρη: Κανονικό και έντονο, σε αντίθεση με πιο εξειδικευμένες γραμματοσειρές που χρησιμοποιούνται από γραφίστες, οι οποίες πολλές φορές διατίθενται σε διάφορα βάρη. Οπότε οι τιμές που συνήθως χρησιμοποιούνται σε αυτή την ιδιότητα είναι οι normal και bold, η λειτουργία των οποίων είναι προφανής.
- **text-decoration**

Μας επιτρέπει να εφαρμόσουμε στο κείμενο μας διάφορα εφέ , συμπεριλαμβανόμενης και της υπογράμμισης. Οι τιμές που δέχεται είναι οι εξής:

1. none: Καμία διακόσμηση
2. underline: Υπογράμμιση
3. overline: Γραμμή πάνω από το κείμενο (ουσιαστικά το αντίθετο της υπογράμμισης)
4. line-through: Διαγράμμιση
5. blink: Κάνει το κείμενο να αναβοσβήνει . Δεν πρέπει να χρησιμοποιείται , παρά μόνο σε εξαιρετικά σπάνιες περιπτώσεις, μιας και η εμπειρία έχει δείξει ότι είναι εξαιρετικά κουραστικό για τον αναγνώστη και αφαιρεί επαγγελματικότητα από το design της ιστοσελίδας.

Οι παραπάνω τιμές μπορούν να χρησιμοποιηθούν και συνδυαστικά , όταν επιθυμούμε να εφαρμόσουμε πάνω από ένα τέτοιο εφέ στο κείμενο μας . Παραδείγματος χάριν text-decoration: underline overline;

- text-align

Μας επιτρέπει να καθορίσουμε τη στοίχιση του κειμένου μας . Όπως θα περίμενε κανείς, οι πιθανές τιμές είναι left, center, right και justify.

- font

Μας επιτρέπει να καθορίσουμε αρκετές ιδιότητες κειμένου με μια μόνο ιδιότητα (συγκεκριμένα τις font-size, font-family, font-weight και font-style από όσες αναφέρθηκαν παραπάνω, καθώς και κάποιες άλλες που δεν αναφέρθηκαν). Αν κάποια ιδιότητα δεν συμπεριληφθεί, ο browser θα «υποθέσει» την προεπιλεγμένη τιμή της.

Παραδείγματα:

1. font: 12px Calibri, Trebuchet MS, Verdana, sans-serif;
2. font: italic bold 12px Calibri, Trebuchet MS, Verdana, sans-serif;

Προσοχή: Αν δεν ορίσουμε κάποια επιμέρους ιδιότητα όταν χρησιμοποιούμε την συγκεντρωτική ιδιότητα font, ο browser «υποθέτει» την προεπιλεγμένη τιμή για αυτή, όχι ότι δεν ορίστηκε. Αυτό σημαίνει ότι αν γράψουμε για παράδειγμα

```
font-weight:bold;
```

```
font: 12px Calibri, Trebuchet MS, Verdana, sans-serif;
```

τότε το κείμενο μας δεν θα είναι έντονο, μιας και αφού δεν ορίσαμε στη font και την font-weight, τότε ο browser «υποθέτει» την προεπιλεγμένη τιμή για αυτή την ιδιότητα, η οποία είναι η normal, και μιας και η font βρίσκεται κάτω από την font-weight, έχει μεγαλύτερη προτεραιότητα, οπότε επικρατεί για τη font-weight η τιμή normal.

2.1.4.7 Bootstrap

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα (Ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Είναι το πιο δημοφιλές πρόγραμμα στο GitHub [1] και έχει χρησιμοποιηθεί από τη NASA και το MSNBC, μεταξύ άλλων.

Το Bootstrap αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα πλαίσιο για την ενθάρρυνση της συνέπειας στα εσωτερικά εργαλεία. Πριν το Bootstrap, διάφορες βιβλιοθήκες χρησιμοποιήθηκαν για την ανάπτυξη της διεπαφής, η οποία οδήγησε σε αντιφάσεις και υψηλή φορολογική επιβάρυνση συντήρησης. Σύμφωνα με τον Twitter developer Mark Otto, για την αντιμετώπιση αυτών των προκλήσεων : "... πήρα μαζί μία σούπερ μικρή ομάδα προγραμματιστών για να σχεδιάσει και να κατασκευάσει ένα νέο εργαλείο εσωτερικής και είδα μια ευκαιρία να κάνουμε κάτι περισσότερο. Μέσω αυτής της διαδικασίας, είδαμε τους εαυτούς μας να χτίζουν κάτι πολύ πιο σημαντικό από οποιοδήποτε άλλο εσωτερικό εργαλείο. Μήνες αργότερα, καταλήξαμε σε μια πρόωμη

έκδοση του Bootstrap ως έναν τρόπο για να καταγράψουμε και να μοιραστούμε κοινά πρότυπα σχεδιασμού και περιουσιακών στοιχείων εντός της εταιρείας ».Η πρώτη εγκατάσταση υπό πραγματικές συνθήκες συνέβη κατά τη διάρκεια της πρώτης Hackweek του Twitter ». Ο Mark Otto έδειξε σε κάποιους συναδέλφους πώς να επιταχύνουν την ανάπτυξη του έργου τους με τη βοήθεια της εργαλειοθήκης. Ως αποτέλεσμα, δεκάδες ομάδες έχουν μετακινηθεί στο πλαίσιο.Τον Αύγουστο του 2011 κυκλοφόρησε Twitter Bootstrap ως λογισμικό ανοιχτού κώδικα. Τον Φεβρουάριο του 2012, ήταν το πιο δημοφιλές έργο ανάπτυξης στο GitHub.

Το Bootstrap έχει σχετικά ελλιπή υποστήριξη για HTML5 και CSS , αλλά είναι συμβατό με όλους τους φυλλομετρητές (browsers). Βασικές πληροφορίες συμβατότητας των ιστοσελίδων ή εφαρμογές είναι διαθέσιμες για όλες τις συσκευές και τα προγράμματα περιήγησης. Υπάρχει μια έννοια της μερικής συμβατότητας που κάνει τα βασικά στοιχεία μιας ιστοσελίδας που διατίθενται για όλες τις συσκευές και τα προγράμματα περιήγησης. Για παράδειγμα, οι ιδιότητες πάνω στις οποίες θεσπίστηκε το CSS3 για στρογγυλεμένες γωνίες, κλίσεις και σκιές, χρησιμοποιούνται από το Bootstrap παρά την έλλειψη υποστήριξης από μεγάλα προγράμματα περιήγησης στο Web. Αυτά επεκτείνουν τη λειτουργικότητα του πακέτου εργαλείων, αλλά δεν απαιτούνται για τη χρήση του. Από την έκδοση 2.0 υποστηρίζει επίσης ανταποκρίσιμο σχεδιασμό (responsive design). Αυτό σημαίνει ότι η διάταξη των ιστοσελίδων προσαρμόζεται δυναμικά, λαμβάνοντας υπόψη τα χαρακτηριστικά της συσκευής που χρησιμοποιείται (PC, tablet, κινητό τηλέφωνο).

Το Bootstrap είναι ανοικτού κώδικα και είναι διαθέσιμο στο GitHub. Οι προγραμματιστές ενθαρρύνονται να συμμετέχουν στο έργο και να κάνουν τη δική τους συνεισφορά στην πλατφόρμα.

Το Bootstrap είναι σπονδυλωτό και αποτελείται ουσιαστικά από μια σειρά στυλ(stylsheets) που εφαρμόζουν τα διάφορα συστατικά του πακέτου εργαλείων. Ένα στυλ που ονομάζεται bootstrap.less περιλαμβάνει τα συστατικά stylesheets. Οι προγραμματιστές μπορούν να προσαρμόσουν το αρχείο Bootstrap, επιλέγοντας τα στοιχεία που θέλουν να χρησιμοποιήσουν στο έργο τους.

Προσαρμογές είναι δυνατές σε περιορισμένη έκταση μέσω ενός κεντρικού στυλ διαμόρφωσης. Η χρήση γλώσσας στυλ επιτρέπει τη χρήση για μεταβλητές, λειτουργίες και φορείς (operators), ένθετους επιλογείς, γνωστά και ως μείγματα mixin.

Από την έκδοση 2.0, η διαμόρφωση του Bootstrap έχει επίσης μία ειδική επιλογή "Προσαρμογή " στην τεκμηρίωση (documentation). Επιπλέον, ο σχεδιαστής του έργου επιλέγει σε μια φόρμα τα επιθυμητά συστατικά και τα προσαρμόζει, εάν είναι αναγκαίο, σε τιμές διαφόρων εναλλακτικών λύσεων για τις ανάγκες του. Στη συνέχεια δημιουργείται ένα πακέτο που περιλαμβάνει ήδη το προ-χτισμένο CSS στυλ.

Σύστημα πλέγματος(Grid System) και ανταποκρίσιμος σχεδιασμός(responsive design)

Το Bootstrap έρχεται σταθερό με 940 pixel πλάτος. Εναλλακτικά, ο προγραμματιστής μπορεί να χρησιμοποιήσει μία μεταβλητού πλάτους διάταξη. Και για τις δύο περιπτώσεις, η εργαλειοθήκη έχει τέσσερις παραλλαγές για χρήση των διαφόρων ψηφισμάτων και τύπους συσκευών κινητά τηλέφωνα, πορτρέτα και το τοπία, ταμπλέτες και υπολογιστές με χαμηλή και υψηλή ανάλυση. Κάθε παραλλαγή ρυθμίζει το πλάτος των στηλών.

Το Bootstrap παρέχει ένα σύνολο στυλ που παρέχουν βασικούς ορισμούς στυλ για όλα τα βασικά στοιχεία HTML. Αυτά παρέχουν ενιαία, σύγχρονη εμφάνιση για πίνακες, μορφοποίηση κειμένου, καθώς και στοιχεία μιας φόρμας.

Εκτός από τα βασικά HTML στοιχεία, το Bootstrap περιέχει και άλλα στοιχεία περιβάλλοντος που χρησιμοποιούνται συχνά. Αυτά περιλαμβάνουν κουμπιά με προηγμένα χαρακτηριστικά (π.χ. ομαδοποίηση κουμπιών ή drop -down επιλογή, οριζόντιες και κάθετες καρτέλες, πλοήγηση, σελιδοποίηση, κ.λπ.), ετικέτες, προηγμένες τυπογραφικές δυνατότητες, εικονίδια, προειδοποιητικά μηνύματα και μια γραμμή προόδου.

Το Bootstrap έρχεται με πολλά συστατικά JavaScript σε μια μορφή jQuery plugin. Παρέχουν πρόσθετη διεπαφή χρήστη με στοιχεία όπως παράθυρα διαλόγου, επεξηγήσεις, και καρουσέλ. Μπορούν επίσης να επεκτείνουν τη λειτουργικότητα ορισμένων υφιστάμενων στοιχείων της διασύνδεσης, όπως για παράδειγμα μια αυτόματη πλήρη λειτουργία για πεδία εισαγωγής. Στην έκδοση 2.0, υποστηρίζονται τα ακόλουθα JavaScript plugins: Modal, Αναπτυσσόμενο, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel και Typeahead.

Διατίθεται επίσης μια υλοποίηση του Bootstrap Twitter που χρησιμοποιεί το Dojo Toolkit. Λέγεται Dojo Bootstrap και είναι στα plugins του Bootstrap Twitter. Χρησιμοποιεί 100% κώδικα Dojo και έχει υποστήριξη για AMD (Asynchronous Module Definition).

Για να χρησιμοποιηθεί το Bootstrap σε μια σελίδα HTML, ο σχεδιαστής του έργου κάνει λήψη του στυλ CSS Bootstrap και περιλαμβάνει μια σύνδεση στο αρχείο HTML.

Αν ο προγραμματιστής θέλει να χρησιμοποιήσει τα στοιχεία JavaScript, θα πρέπει να αναφέρονται μαζί με τη βιβλιοθήκη jQuery στο HTML έγγραφο.

Το ακόλουθο παράδειγμα δείχνει πώς λειτουργεί αυτό. Ο κώδικας HTML ορίζει μια απλή μορφή αναζήτησης και μια λίστα των αποτελεσμάτων σε μορφή πίνακα. Η σελίδα αποτελείται από στοιχεία πληροφοριών HTML 5 και CSS σύμφωνα με την τεκμηρίωση του Bootstrap. Το σχήμα δείχνει την αναπαράσταση του εγγράφου στον Mozilla Firefox 10.

2.1.4.8 AJAX

Η τεχνολογία AJAX (Asynchronous Javascript and XML) αυτή τη στιγμή αποτελεί τη πιο σύγχρονη τεχνολογία στον προγραμματισμό στο internet, δίνοντας διαδραστικές δυνατότητες σε ένα δυναμικό site, μετατρέποντας το από ένα απλό site σε μια διαδικτυακή εφαρμογή. Ένας από τους κύριους εκφραστές αυτής της τεχνολογίας είναι και η jQuery.

Σίγουρα θα έχετε παρατηρήσει τα τελευταία χρόνια σε διάφορα sites την εντυπωσιακή εμφάνιση κειμένων, τα πρωτότυπα scrolling, τα δυναμικά ξεφυλλίσματα σε photogalleries. Αυτές οι υλοποιήσεις χρησιμοποιούν την τεχνολογία AJAX και κύριο χαρακτηριστικό τους δεν είναι οι εντυπωσιακές κινήσεις που κάνουν αλλά η αλλαγή τους χωρίς την επαναφόρτωση του site. Η AJAX δίνει τη δυνατότητα εμφάνισης νέων στοιχείων στο site, χωρίς τη φόρτωση νέας σελίδα. Μπορεί δηλαδή ο web developer να δημιουργήσει ένα site με μία μόνο σελίδα, στην οποία θα φορτώνονται διαφορετικά δεδομένα ανάλογα με τις επιλογές του χρήστη. Έτσι καταργεί τους ατελείωτους φακέλους με τα html αρχεία, στα οποία επαναλαμβάνεται το ίδιο κομμάτι κώδικα, βελτιώνοντας παράλληλα και την ασφάλεια του site καθώς καταργεί την αλλαγή του url στη μπάρα διευθύνσεων. Όπως δηλώνει και στο όνομα της, η κύρια γλώσσα με την οποία εφαρμόζεται η AJAX είναι η JavaScript. Συνεπώς κάποιος που χρησιμοποιεί τη JavaScript μπορεί να την χρησιμοποιήσει για να εφαρμόσει τη τεχνολογία AJAX. Πέρα όμως από την JavaScript, τα τελευταία χρόνια έχουν κάνει την εμφάνισή τους και κάποιες πρόσθετες βιβλιοθήκες οι οποίες δίνουν τη δυνατότητα στον προγραμματιστή να γράψει κώδικα σε JavaScript με ποιο εύκολο, σύντομο και κατανοητό τρόπο. Τέτοιες βιβλιοθήκες είναι οι:

- jQuery
- Dojo
- YUI
- MooTools
- Prototype

Η jQuery είναι μια βιβλιοθήκη η χρήση της οποίας μας δίνει τη δυνατότητα να γράψουμε με ποιο ευανάγνωστο τρόπο JavaScript κώδικα, δίνοντας μας παράλληλα τη δυνατότητα να δημιουργήσουμε και γραφικά.

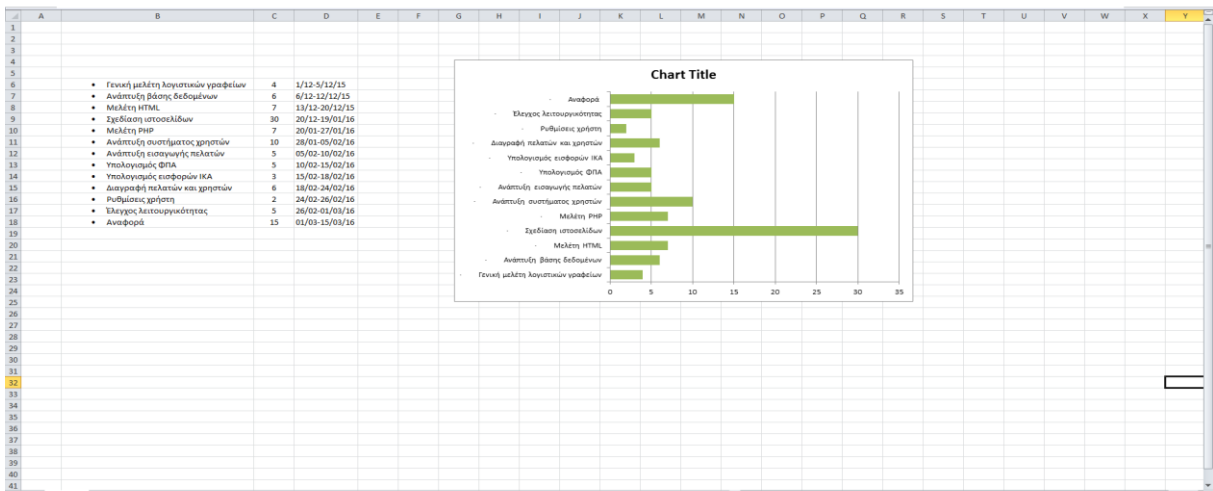
Η δημοφιλέστερη και η πιο εύχρηστη από αυτές είναι η jQuery. Το μόνο που χρειάζεται για την εγκατάστασή της, είναι το κατέβασμα του αρχείου της βιβλιοθήκης και η αποθήκευσή του στο φάκελο του site που θα χρησιμοποιηθεί η jQuery. Ένας άλλος εναλλακτικός τρόπος είναι η δήλωση του url του αρχείου της βιβλιοθήκης απευθείας μέσα στο κώδικα του site αποφεύγοντας έτσι το κατέβασμα της βιβλιοθήκης.

ΚΕΦΑΛΑΙΟ 3 | ΣΧΕΔΙΟ ΔΡΑΣΗΣ ΓΙΑ ΤΗΝ ΕΚΜΠΟΝΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

3.1 ΣΗΜΑΝΤΙΚΟΙ ΣΤΟΧΟΙ ΓΙΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ

• Γενική μελέτη λογιστικών γραφείων	4
• Ανάπτυξη βάσης δεδομένων	6
• Μελέτη HTML	7
• Σχεδίαση ιστοσελίδων	30
• Μελέτη PHP	7
• Ανάπτυξη συστήματος χρηστών	10
• Ανάπτυξη εισαγωγής πελατών	5
• Υπολογισμός ΦΠΑ	5
• Υπολογισμός εισφορών ΙΚΑ	3
• Διαγραφή πελατών και χρηστών	6
• Ρυθμίσεις χρήστη	2
• Έλεγχος λειτουργικότητας	10
• Αναφορά	20

3.2 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ



Εικόνα 3: Χρονοδιάγραμμα

ΚΕΦΑΛΑΙΟ 4 | ΚΥΡΙΟ ΜΕΡΟΣ ΠΤΥΧΙΑΚΗΣ

4.1 ΑΝΑΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Θέλουμε να σχεδιάσουμε και να υλοποιήσουμε ένα σύστημα μηχανογράφησης πελατών και εργασιών για να χρησιμοποιηθεί σε λογιστικό γραφείο.

Ένα λογιστικό γραφείο έχει πελάτες οι οποίοι θέλουν να εξυπηρετούνται για συγκεκριμένες εργασίες από τους υπαλλήλους του γραφείου. Πιο συγκεκριμένα οι πελάτες θέλουν να εξυπηρετούνται για τον υπολογισμό του ΦΠΑ που πρέπει να αποδώσουν στο κράτος και για τον υπολογισμό των εισφορών που πρέπει να πληρώσουν στο ΙΚΑ για το προσωπικό τους. Έτσι λοιπόν διακρίνουμε τις εξής πέντε περιπτώσεις:

- Ο πελάτης να απασχολεί το γραφείο για απόδοση ΦΠΑ και για υπολογισμό εισφορών στο ΙΚΑ
- Ο πελάτης να απασχολεί το γραφείο μόνο για απόδοση ΦΠΑ (εφόσον δεν έχει προσωπικό).
- Ο πελάτης να απασχολεί το γραφείο για όλα τα παραπάνω.

Οι υπάλληλοι του γραφείου μοιράζονται τις εργασίες ανάλογα με το είδος της εργασίας. Ένας συγκεκριμένος αριθμός υπαλλήλων εξυπηρετεί τους πελάτες για απόδοση ΦΠΑ. Ένας άλλος αριθμός υπαλλήλων εξυπηρετεί για υπολογισμό των εισφορών για το ΙΚΑ. Τέλος υπάρχουν και οι υπάλληλοι που εξυπηρετούν τους πελάτες για την συμπλήρωση και υποβολή των φορολογικών τους δηλώσεων.

Το σύστημα που θα υλοποιηθεί πρέπει να πληροί συγκεκριμένες προδιαγραφές τόσο για τη χρήση του από τους υπαλλήλους του λογιστικού γραφείου, όσο και για το διαχωρισμό των υπαλλήλων ανάλογα με την εργασία που εξυπηρετούν τους πελάτες. Οι υπάλληλοι που έχουν ως εργασία την απόδοση ΦΠΑ πρέπει να βλέπουν στοιχεία της σελίδας που αφορούν μόνο το ΦΠΑ και καμία άλλη εργασία. Οι υπάλληλοι που έχουν ως εργασία τον υπολογισμό εισφορών στο ΙΚΑ πρέπει να βλέπουν στοιχεία της σελίδας που αφορούν μόνο το ΙΚΑ. Επίσης οι υπάλληλοι δεν πρέπει να μπορούν να έχουν πρόσβαση στις οφειλές του πελάτη προς το γραφείο.

Ένας ειδικός τύπος χρήστη του συστήματος είναι ο διαχειριστής (admin). Αυτός ο τύπος χρήστη έχει πρόσβαση στις οφειλές του πελάτη. Επίσης είναι ο μόνος χρήστης που έχει δικαιώματα για τη δημιουργία νέου χρήστη συστήματος, για τη διαγραφή χρήστη από το σύστημα, για την εισαγωγή νέου πελάτη στο σύστημα και τέλος για τη διαγραφή πελάτη από το σύστημα. Επίσης θα πρέπει να έχει πρόσβαση σε όλους τους τύπους εργασιών, δηλαδή και στην απόδοση του ΦΠΑ, αλλά και στον υπολογισμό των εισφορών του πελάτη προς το ΙΚΑ.

Για την εισαγωγή των πελατών στο σύστημα θέλουμε να κρατάμε συγκεκριμένα στοιχεία. Αυτά είναι:

- Όνομα
- Επώνυμο
- Διεύθυνση
- Τηλέφωνα
- Χρέωση
- ΑΦΜ
- Αριθμό δελτίου αστυνομικής ταυτότητας
- Αριθμό υπαλλήλων που απασχολεί
- Έσοδα
- Έξοδα
- ΦΠΑ
- Εισφορές ΙΚΑ

Για την εισαγωγή των υπαλλήλων στο σύστημα ως χρήστες θα πρέπει να κρατάμε κάποια στοιχεία. Αυτά είναι:

- Όνομα

- Επώνυμο
- Διεύθυνση
- Όνομα χρήστη
- Κωδικός χρήστη
- Δικαιώματα χρήστη (user ή admin)
- Εργασία χρήστη (ΦΠΑ, ΙΚΑ ή admin για τον διαχειριστή)

Τέλος για κάθε εργασία που εξυπηρετείται ο κάθε πελάτης θέλουμε να υπάρχει κάποια χρέωση που θα προστίθεται στις συνολικές οφειλές του πελάτη κάθε φορά που θα εκτελείται η συγκεκριμένη εργασία. Ενδεικτικό κοστολόγιο είναι το εξής:

- 50€ για απόδοση ΦΠΑ
- 70€ για υπολογισμό εισφορών στο ΙΚΑ

4.1.2 ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ

Οι απαιτήσεις συστήματος, όπως αναγνωρίστηκαν και κατηγοριοποιήθηκαν μετά από κατάλληλη έρευνα είναι οι εξής κατηγορίες:

- Λειτουργικές απαιτήσεις
- Μη λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τι πρέπει να κάνει το σύστημα (π.χ. ως συναρτήσεις που λαμβάνουν είσοδο και δίδουν έξοδο). Έτσι λοιπόν ως λειτουργικές απαιτήσεις καθορίστηκαν οι εξής:

- Το σύστημα πρέπει να αποθηκεύει τα στοιχεία των πελατών.
- Το σύστημα πρέπει να αποθηκεύει τα στοιχεία των χρηστών.
- Το σύστημα πρέπει να υπολογίζει την απόδοση ΦΠΑ για κάθε πελάτη.
- Το σύστημα πρέπει να υπολογίζει τις εργοδοτικές εισφορές στο ΙΚΑ για κάθε πελάτη που έχει προσωπικό.
- Το σύστημα πρέπει να υποστηρίζει την διαγραφή πελατών.
- Το σύστημα πρέπει να υποστηρίζει τη διαγραφή χρηστών.
- Πρέπει να έχει την επιλογή για αλλαγή κωδικού του χρήστη.
- Πρέπει να διαθέτει διαφορετικό μενού για τον διαχειριστή συστήματος και για τους απλούς χρήστες.
- Πρέπει να κατηγοριοποιεί τους χρήστες ανάλογα με την εργασία που τους έχει ανατεθεί από το γραφείο.

Οι μη λειτουργικές απαιτήσεις περιγράφουν ιδιότητες του συστήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής:

1. Απόδοση
2. Χρηστικότητα
3. Ασφάλεια
4. Νομιμότητα
5. Ιδιωτικότητα

Με άλλα λόγια: περιγράφουν το πώς (ή το πόσο καλά) το σύστημα θα υποστηρίξει τις λειτουργικές απαιτήσεις. Μπορούμε να τις θεωρήσουμε ως «περιορισμούς» που περιορίζουν τους τρόπους με τους οποίους θα μπορούσαμε να πραγματώσουμε τις λειτουργικές απαιτήσεις. Οι μη λειτουργικές απαιτήσεις καταγράφηκαν ως εξής:

- Ο χρόνος απόκρισης του συστήματος δεν πρέπει να υπερβαίνει τα 3 δευτερόλεπτα
- Το σύστημα πρέπει να μπορεί να ενοποιηθεί με το υπάρχον
- Τα προσωπικά στοιχεία των πελατών πρέπει να προστατεύονται.
- Να αποθηκεύει τα στοιχεία των πελατών
- Το σύστημα πρέπει να λειτουργεί αδιάλειπτα (όλο το χρόνο)
- Το σύστημα πρέπει συμμορφώνεται με τα πρότυπα της βιομηχανίας
- Ο χώρος στο δίσκο για έναν πελάτη δεν πρέπει να υπερβαίνει τα 200 bytes
- Το σύστημα πρέπει είναι μπορεί να διεκπεραιώσει τουλάχιστον 100 δοσοληψίες ταυτόχρονα
- Η εκμάθηση του συστήματος δεν πρέπει να απαιτήσει πάνω από 4 ώρες εκπαίδευση
- Το σύστημα δεν πρέπει να επιτρέπει την εισαγωγή στοιχείων που δεν ικανοποιούν τον τύπο των αντίστοιχων πεδίων
- Το ποσοστό των συμβάντων που έχουν σαν αποτέλεσμα την πτώση (διακοπή ομαλής λειτουργίας) του συστήματος δεν πρέπει να υπερβαίνει το 2%
- Ο χρόνος επανεκκίνησης του συστήματος μετά από οποιαδήποτε διακοπή δεν πρέπει να υπερβαίνει τα 3 λεπτά
- Το σύστημα πρέπει να κάνει αυτόματη επανεκκίνηση μετά από πτώση
- Το σύστημα δεν πρέπει να καταρρέει πάνω από 2 φορές το χρόνο
- Ο μέσος χρόνος μεταξύ 2 καταρρέσεων πρέπει να είναι τουλάχιστον 4 μήνες
- Ο ποσοστό των γραμμών κώδικα που εξαρτώνται από την πλατφόρμα υλοποίησης δεν πρέπει να υπερβαίνει το 4%
- Κάθε χρήστης πρέπει να έχει όνομα εισόδου και κωδικό πρόσβασης
- Δυνατότητα χρήσης του συστήματος σε υπάρχον εφαρμογές ή άλλες μελλοντικές
- Το σύστημα πρέπει να κάνει ορθή χρήση πόρων
- Το σύστημα δεν θα πρέπει να κρατάει τα προσωπικά δεδομένα των πελατών πάνω από τρεις μέρες όπως ορίζει η συνθήκη της Segen.

4.2 ΣΧΕΔΙΑΣΜΟΣ ΥΛΟΠΟΙΗΣΗΣ

Στα αρχικά βήματα του σχεδιασμού του συστήματος πραγματοποιήθηκε η ανάλυση λειτουργικών και μη λειτουργικών απαιτήσεων με βάση τις ερωτήσεις που έγιναν σε λογιστικό γραφείο των Χανίων.

Η πρώτη εργασία που πραγματοποιήθηκε είναι ο σχεδιασμός και η υλοποίηση λογισμικού που θα αποθηκεύει τα δεδομένα που χρειαζόμαστε έτσι ώστε να ικανοποιηθούν οι απαιτήσεις του συστήματος. Χρησιμοποιήθηκε το σύστημα βάσεων δεδομένων MYSQL, το πιο δημοφιλές σύστημα διαχείρισης βάσεων δεδομένων που βασικά πλεονεκτήματα του είναι:

- Εύκολο στην εγκατάσταση και χρήση με έτοιμο γραφικό περιβάλλον χρήστη (χρησιμοποιήθηκε το σύστημα DataGrip της IntelliJ)
- Ενσωματώνει πλούσια λειτουργικότητα της SQL, λειτουργικότητα που απαιτείται από οποιοδήποτε σύστημα διαχείρισης βάσεων δεδομένων, είτε άμεσα, είτε έμμεσα.
- Ενσωματώνει αρκετές λειτουργίες ασφαλείας, μερικές αρκετά προχωρημένες.
- Μπορεί να χειριστεί τεράστιο όγκο δεδομένων και αν χρειαστεί είναι κλιμακούμενο.
- Παρέχει μεγάλη ταχύτητα στις αναζητήσεις και γενικότερα στις λειτουργίες που πρέπει να υποστηρίζει ένα σύστημα διαχείρισης βάσεων δεδομένων.

Αποφασίστηκε με βάση τις απαιτήσεις του συστήματος η βάση δεδομένων να αποτελείται από δύο οντότητες, τον πελάτη (customer) και τον υπάλληλο (employee), όπως φαίνεται και στο διάγραμμα οντοτήτων παρακάτω.

customer	employee
afm int(11)	idemployee int(11)
lastname varchar(45)	privileges varchar(45)
name varchar(45)	username varchar(45)
tayotita varchar(45)	password varchar(45)
xrewsi int(11)	task varchar(45)
adress varchar(45)	name varchar(20)
employees int(11)	lastname varchar(20)
exoda int(11)	adress varchar(20)
esoda int(11)	
ika int(11)	
fpa int(11)	
Phones varchar(25)	

Εικόνα 4: Πίνακες

Οι δύο οντότητες (customer, employee) μεταφράζονται στους πίνακες customer και employee. Με το πρόγραμμα Datagrip που διευκολύνει αρκετά τον χρήστη στο να σχεδιάσει και να υλοποιήσει τη βάση δεδομένων, προχώρησε η σχεδίαση αρκετά γρήγορα.

Κατόπιν με δοκιμαστικές επερωτήσεις προς τη βάση εισήχθησαν καταχωρήσεις στον πίνακα customer.

```
MySQL - logistik@localhost (logistik)
customer
afm INT(11)
lastname VARCHAR(45)
name VARCHAR(45)
tayotita VARCHAR(45)
xrewsi INT(11)
adress VARCHAR(45)
employees INT(11)
esoda INT(11)
ika INT(11)
fpa INT(11)
Phones VARCHAR(25)
INSERT INTO customer VALUES (123456,'customer1','customer','tayotita',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123457,'customer2','customer','tayotita2',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123458,'customer3','customer','tayotita3',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123459,'customer4','customer','tayotita2',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123450,'customer5','customer','tayotita3',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123451,'customer6','customer','tayotita4',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123452,'customer7','customer','tayotita5',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123453,'customer8','customer','tayotita6',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123454,'customer9','customer','tayotita7',0,'esoda',1,0,0,0,0,'2821034343');
INSERT INTO customer VALUES (123455,'customer8','customer','tayotita8',0,'esoda',1,0,0,0,0,'2821034343');
```

Εικόνα 5: Εισαγωγές στον πίνακα πελατών

afm	name	lastname	taytotita	xrewsi	address	employees	exoda	esoda	ika	fpa	Phones
1 123450	customer5	customer	tautotita3	0	odos	1	0	10	0	0	2821034343
2 123451	customer6	customer	tautotita4	0	odos	1	12	30	0	0	2821034343
3 123452	customer7	customer	tautotita5	0	odos	1	87	50	0	0	2821034343
4 123453	customer8	customer	tautotita6	0	odos	1	100	65	0	0	2821034343
5 123454	customer9	customer	tautotita7	0	odos	1	140	200	0	0	2821034343
6 123455	customer0	customer	tautotita8	0	odos	1	89	125	0	0	2821034343
7 123457	customer2	customer	tautotita	0	odos	1	8	15	0	0	2821034343
8 123458	customer3	customer	tautotita1	0	odos	1	666	25	0	0	2821034343
9 123459	customer4	customer	tautotita2	0	odos	1	565	465	0	0	2821034343

Εικόνα 6: Πίνακας πελατών

Ομοίως και για τον πίνακα employee

```

MySQL - logistik@localhost (logistika)
└─> customer
    └─> employee
        └─> idemployee INT(11)
            ├── privileges VARCHAR(45)
            ├── username VARCHAR(45)
            ├── password VARCHAR(45)
            ├── task VARCHAR(45)
            ├── name VARCHAR(20)
            ├── lastname VARCHAR(20)
            └── address VARCHAR(20)
INSERT INTO employee VALUES (1, 'user', 'username1', 'password', 'ika', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (2, 'user', 'username1', 'password', 'ika', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (3, 'user', 'username2', 'password', 'ika', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (4, 'user', 'username3', 'password', 'ika', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (5, 'user', 'username4', 'password', 'fpa', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (6, 'user', 'username5', 'password', 'fpa', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (7, 'user', 'username6', 'password', 'fpa', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (8, 'user', 'username7', 'password', 'fpa', 'name', 'lastname', 'address');
INSERT INTO employee VALUES (9, 'user', 'username8', 'password', 'fpa', 'name', 'lastname', 'address');
    
```

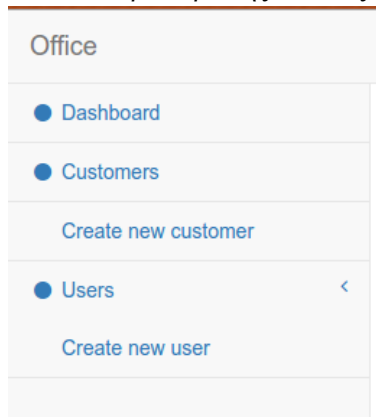
Εικόνα 7: Εισαγωγές στον πίνακα χρηστών

idemployee	privileges	username	password	task	name	lastname	address
1	admin	vagsmyrn	12345	admin	vag	sadr	ydramias
2	user	username1	password	name	name	lastname	address
3	user	username2	password	ika	name	lastname	address
4	user	username3	password	ika	name	lastname	address
5	user	username4	password	fpa	name	lastname	address
6	user	username5	password	fpa	name	lastname	address
7	user	username6	password	fpa	name	lastname	address
8	user	username7	password	fpa	name	lastname	address
9	11	username	password	ika	name	lastname	address
10	19	username8	password	fpa	name	lastname	address

Εικόνα 8: Πίνακας χρηστών

Στη συνέχεια ξεκίνησε η σχεδίαση του γραφικού περιβάλλοντος χρήστη (GUI). Για το σκοπό αυτό χρησιμοποιήθηκε η βιβλιοθήκη bootstrap που έχει δημιουργηθεί για σχεδίαση περιβάλλοντος διαδικτυακών εφαρμογών παρέχοντας στον προγραμματιστή έτοιμες συναρτήσεις javascript μέσω της βιβλιοθήκης jquery αλλά και έτοιμα css stylesheets για τη διαμόρφωση της κάθε ιστοσελίδας σε οποιαδήποτε στυλ επιθυμεί ο προγραμματιστής.

Πιο συγκεκριμένα χρησιμοποιήθηκαν κομμάτια από το template sb admin 2, το οποίο παρέχει components σχεδιασμένα για εφαρμογές διαχείρισης γραφείων. Καταρχήν, σχεδιάστηκε η μπάρα περιήγησης που βρίσκεται στα αριστερά της σελίδας.



Εικόνα 9: Sidebar

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ύστερα, η σχεδίαση του κομματιού που θα χρησιμοποιηθεί ως πίνακας για την προβολή των πελατών, αλλά και των χρηστών του συστήματος, δηλαδή των υπαλλήλων του γραφείου.

Insert last name to search for users

Name	Last Name	Address	Task	Edit
vag	sadr	ydramias	admin	Delete
name	lastname	adress	ika	Delete
name	lastname	adress	ika	Delete
name	lastname	adress	ika	Delete
name	lastname	adress	fpa	Delete
name	lastname	adress	fpa	Delete
name	lastname	adress	fpa	Delete
name	lastname	adress	fpa	Delete
name	lastname	adress	ika	Delete
name	lastname	adress	fpa	Delete

Εικόνα 10: Καρτέλα χρηστών


Insert AFM or last name to search for customers

Atm	Name	Last Name	Id	Address	Phones	Employees	Debt to us	Delete customer	Edit
123450	customer5	customer	tautoita3	odos		1	0	Delete	Edit fpa Edit ika
123451	customer6	customer	tautoita4	odos		1	0	Delete	Edit fpa Edit ika
123452	customer7	customer	tautoita5	odos		1	0	Delete	Edit fpa Edit ika
123453	customer8	customer	tautoita6	odos		1	0	Delete	Edit fpa Edit ika
123454	customer9	customer	tautoita7	odos		1	0	Delete	Edit fpa Edit ika
123455	customer0	customer	tautoita8	odos		1	0	Delete	Edit fpa Edit ika
123457	customer2	customer	tautoita	odos		1	0	Delete	Edit fpa Edit ika
123458	customer3	customer	tautoita1	odos		1	0	Delete	Edit fpa Edit ika
123459	customer4	customer	tautoita2	odos		1	0	Delete	Edit fpa Edit ika

Εικόνα 11: Καρτέλα πελατών

Στη συνέχεια σχεδιάστηκαν τα banners που χρησιμοποιήθηκαν στην αρχική σελίδα της εφαρμογής και τα οποία υποδεικνύουν στους χρήστες πόσοι πελάτες είναι εγγεγραμμένοι στο σύστημα και πόσες και ποιες εργασίες πρέπει να περατωθούν αναλόγως πάντα του τι εργασία έχει ανατεθεί σε κάθε υπάλληλο.

Dashboard

3

You have Customers!

[View Details](#)

Tasks

3

FPA

[View Details](#)

3

IKA

[View Details](#)

Εικόνα 12: Dashboard

Αποφασίστηκε να χρησιμοποιηθεί η scripting γλώσσα php για την ανάπτυξη του backend συστήματος, κυρίως επειδή εμφανίζει τα εξής πλεονεκτήματα:

- Είναι ανοικτού κώδικα, και είναι εντελώς δωρεάν.
- Μπορεί πολύ εύκολα να ενσωματωθεί σε κώδικα HTML.
- Μπορεί ελεύθερα να τρέξει σε οποιοδήποτε λειτουργικό σύστημα / server.
- Τρέχει πολύ γρήγορα στο διαδίκτυο και μπορεί να ενσωματώσει πολύ εύκολα ajax.
- Συνεργάζεται πολύ καλά με Apache / MYSQL.
- Υπάρχει πολύ βοηθητικό υλικό και κώδικας στο διαδίκτυο.

Πρώτη δουλειά ήταν να δημιουργηθεί ένα σύστημα που θα επιτρέπει στους χρήστες να πραγματοποιούν είσοδο στο σύστημα χρησιμοποιώντας το όνομα χρήστη και τον κωδικό τους. Έτσι χρησιμοποιήθηκε η global μεταβλητή session που παρέχει η PHP και στην οποία αφού ελέγξουμε αν υπάρχει ο χρήστης στη βάση δεδομένων και αν είναι σωστός ο κωδικός που έδωσε αποθηκεύουμε κάποια στοιχεία του που θα χρησιμοποιηθούν αργότερα, όπως τα δικαιώματα που έχει, η εργασία που του έχει ανατεθεί και το όνομα χρήστη του. Απο εκεί και πέρα μπορούμε να έχουμε πρόσβαση στα στοιχεία αυτά μέσω του πίνακα \$_SESSION[] σε οποιοδήποτε αρχείο τα χρειαστούμε.

Η επόμενη φάση αποτελούταν από την σύνδεση ουσιαστικά της βάσης δεδομένων με το γραφικό περιβάλλον. Έπρεπε να δημιουργηθεί ένα σύστημα που θα ελέγχει τους χρήστες κατά την είσοδό τους, αλλά και δεν θα επιτρέπει την ποσβαση σε σελίδες του συστήματος αν δεν έχει προηγηθεί είσοδος από τον χρήστη.

Αυτό υλοποιήθηκε στο αρχείο **index.php** όπου ουσιαστικά καταχωρεί ο χρήστης το όνομα χρήστη του και τον κωδικό του σε μία φόρμα και κατόπιν τα στοιχεία αυτά αποστέλλονται στο αρχείο **login.php**.

```
echo
<div class="container">
<div class="row">
<div class="col-md-4 col-md-offset-4">
<div class="login-panel panel panel-default">
<div class="panel-heading">
<h3 class="panel-title">Please Sign In</h3>
</div>
<div class="panel-body">
<form role="form" name="loginform" action="login.php" method="post" onsubmit="return check()"
id="login-form">
<fieldset>
<div class="form-group">
<input class="form-control" placeholder="Username" name="username" type="
"text" autofocus>
</div>
<div class="form-group">
<input class="form-control" placeholder="Password" name="password" type="
"password" value="">
</div>
<!-- Change this to a button or input when using this as a form -->
<input name="Submit" type="submit" value="Login" id="button" />
</form>
</fieldset>
</div>
</div>
</div>
</div>
</div>
```

Εικόνα 13: index.php

Στο αρχείο **login.php** παίρνουμε τα στοιχεία του χρήστη χρησιμοποιώντας τη μέθοδο POST που μας παρέχει η PHP και αφού συνδεθούμε με τη βάση δεδομένων πραγματοποιούμε επερώτηση στη βάση για να δούμε πρώτα αν υπάρχει ο χρήστης και κατόπιν αν ο κωδικός που έδωσε είναι σωστός. Εφόσον πληρούνται αυτές οι προϋποθέσεις ξανακάνουμε επερώτηση στη βάση για να ανασύρουμε τα δικαιώματα του χρήστη στο σύστημα αλλά και την εργασία που του έχει ανατεθεί. Αποθηκεύουμε τα στοιχεία στον global πίνακα \$_SESSION για να τα έχουμε διαθέσιμα απο εδώ και πέρα σε όλα τα αρχεία χωρίς να ξανακάνουμε επερώτηση στη βάση.

```
$username = $_POST['username'];
$password = $_POST['password'];

$mysqli = new mysqli('localhost', 'root', '12345', 'logistikos');
$username = $mysqli->real_escape_string($username);

$query = "SELECT password
          FROM employee
          WHERE username = '$username'";

$result = mysqli_query($mysqli, $query);

if($result->num_rows == 0) // User not found. So, redirect to login_form again.
{
    die("Username not found");
}
$passcheck = mysqli_fetch_array($result);

if($password != $passcheck['0']) // Incorrect password. So, redirect to login_form again.
{
    die ("Wrong Password! <a href=\"index.php\">Go Back!</a>");
}
else{ // Redirect to home page after successful login.
    $query = "SELECT privileges, task FROM employee WHERE username = '$username'";
    $result = mysqli_query($mysqli, $query);
    $forsession = mysqli_fetch_array($result);

    $_SESSION['sess_username'] = $username;
    $_SESSION['sess_privileges'] = $forsession['0'];
    $_SESSION['sess_task'] = $forsession['1'];
}
```

Εικόνα 14: login.php

Μετά το σύστημα μας οδηγεί στην αρχική σελίδα που είναι το αρχείο *main.php* και από εκεί έχουμε διάφορες επιλογές για την περιήγησή μας στο σύστημα. Για την εισαγωγή νέου πελάτη στο σύστημα πατάμε την επιλογή Create new customer που μας οδηγεί στο αρχείο *new_customer.php*. Εισάγουμε τα στοιχεία του πελάτη στη φόρμα που μας παρουσιάζεται και πατάμε το κουμπί **Submit**. Η εισαγωγή του πελάτη πραγματοποιείται εσωτερικά χωρίς να φαίνεται στο χρήστη χρησιμοποιώντας τη φόρμα και τη μέθοδο POST.

```
<div class="panel panel-default">
  <div class="panel-heading">
    Complete all boxes
  </div>
  <div class="panel-body">
    <div class="row">
      <div class="col-lg-6">
        <form role="form" action="insert_customer.php" method="post">
          <div class="form-group">
            <label>Afm</label>
            <input class="form-control" name="afm">
            <label>Name</label>
            <input class="form-control" name="name">
            <label>Last name</label>
            <input class="form-control" name="last_name">
            <label>Id</label>
            <input class="form-control" name="id">
            <label>Debt</label>
            <input class="form-control" name="debt">
            <label>Address</label>
            <input class="form-control" name="adress">
            <label>Phones</label>
            <input class="form-control" name="phones">
            <label>No Of Employees</label>
            <input class="form-control" name="employees">
          </div>
          <button type="submit" class="btn btn-default">Submit</button>
        </form>
      </div>
    </div>
  </div>
</div>
<!-- /.row (nested) -->
</div>
<!-- /.panel-body -->
</div>
```

Εικόνα 15: new_customer.php

Τα στοιχεία αποστέλλονται στο αρχείο *insert_customer.php* και από εκεί πραγματοποιούμε σύνδεση με τη βάση δεδομένων και κατόπιν με την εντολή INSERT INTO της MYSQL εισάγουμε τα στοιχεία του πελάτη στη βάση.

```
session_start();
if (!isset($_SESSION['sess_username'])) {
    header("Location: unauthorized_error.php");
}

if(isset($_SESSION['sess_username']) && isset($_SESSION['sess_privileges']))
{
    $usercheck = $_SESSION['sess_username'];
    $privcheck = $_SESSION['sess_privileges'];
}
else {
    $privcheck=1;
    $usercheck=0;
}
require_once('dbcon.php');
$sql="INSERT INTO customer (afm, name, lastname, taytotita, xrewsi, adress, phones, employees)
VALUES
('$POST[afm]', '$POST[name]', '$POST[last_name]', '$POST[id]', '$POST[debt]', '$POST[address]', '$POST[phones]', '$POST[employees]')";
if (!mysqli_query($con,$sql))
die("Error: ". mysqli_error($con));
```

Εικόνα 16: insert_customer.php

Για να έχουμε πρόσβαση στους πελάτες πηγαίνουμε στην επιλογή Customers στη μπάρα περιήγησης που μας οδηγεί στο αρχείο *customers.php*. Εκεί μπορούμε να δούμε τους πελάτες του γραφείου χρησιμοποιώντας ασύγχρονη μετάδοση δεδομένων (AJAX) μέσω του *prototype.js* και χρησιμοποιώντας τη λειτουργία *Ajax.Updater* η οποία δέχεται ένα αρχείο ως είσοδο από όπου θα λάβει τα δεδομένα των πελατών που σε αυτή την περίπτωση είναι το αρχείο *get_customers.php*.

```
<div id="page-wrapper">
  <div class="row">
    <div id="content">
      <?php
        echo "
        <div class='row'>
          <div class='col-lg-6'>
            <h4 class='page-header'>Insert AFM or last name to search for customers</h4>
          </div>
        </div>;

        echo "
        <form class='col-lg-6' id='searchform' method='post' onsubmit='return false;'>
          <input autocomplete='off' id='searchbox ex3' name='searchv' onkeyup='sendRequest()' onclick='sendRequest()' type=
            'text' class='col-xs-4 form-control'>
        </form>
        <br><br><br><br>
        <div id='show_results'></div>

        <script src='prototype.js' type='text/javascript'></script>
        <script>
          function sendRequest() {
            new Ajax.Updater('show_results', 'get_customers.php', { method: 'post', parameters: $('searchform').serialize() });
          }
        </script>;
      <?>
    </div>
  </div>
</div>
<!-- /#page-wrapper -->
```

Εικόνα 17: customers.php

Το αρχείο *get_customers.php* κάνει μία επερώτηση στη βάση για να πάρει όλους τους πελάτες και κατόπιν τους εκτυπώνει σε έναν πίνακα βάζοντας τα κατάλληλα στοιχεία αναλόγως τα δικαιώματα του χρήστη και την εργασία που του έχει ανατεθεί. Αυτός ο πίνακας ουσιαστικά στέλνεται πίσω στο αρχείο *customers.php*.

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

```
if(empty($searchv)) {
    $sql = "SELECT * FROM customer";
} else {
    //If there is text in the search field, this code is executed every time the input changes.
    $sql = "SELECT * FROM customer WHERE afm LIKE '%$searchv%' OR lastname LIKE '%$searchv%'";
}
$result = mysqli_query($con, $sql);
echo "<table id='results' class='table table-striped table-bordered table-hover'>
<tr>
<th>Afm</th>
<th>Name</th>
<th>Last Name</th>
<th>Id</th>
<th>Address</th>
<th>Phones</th>
<th>Employees</th>
";
if ($privcheck == 'admin'){
    echo "<th>Debt to us</th>";
    echo "<th>Delete customer</th>";
}
echo "
<th colspan='2'>Edit</th>
</tr>";
while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['afm'] . "</td>";
    echo "<td>" . $row['name'] . "</td>";
    echo "<td>" . $row['lastname'] . "</td>";
    echo "<td>" . $row['taytotita'] . "</td>";
    echo "<td>" . $row['adress'] . "</td>";
    echo "<td>" . $row['phones'] . "</td>";
    echo "<td>" . $row['employees'] . "</td>";
    if ($privcheck == 'admin'){
        echo "<td>" . $row['xrewsi'] . "</td>";
        echo "<td><a href='remove_customer.php?takeafm=" . $row['afm'] . "'>Delete</a></td>";
        echo "<td><a href='fpa.php?takeafm=" . $row['afm'] . "'>Edit fpa</a></td>";
        echo "<td><a href='ika.php?takeafm=" . $row['afm'] . "'>Edit ika</a></td>";
    } elseif ($privcheck == 'user') {
        if ($taskcheck == 'fpa') {
            echo "<td><a href='fpa.php?takeafm=" . $row['afm'] . "'>Edit fpa</a></td>";
        } else{
            echo "<td><a href='ika.php?takeafm=" . $row['afm'] . "'>Edit ika</a></td>";
        }
    }
    echo "</tr>";
}
echo "</table>";
```

Εικόνα 18: get_customers.php

Η ίδια διαδικασία υλοποίησης χρησιμοποιήθηκε και για την επιλογή που έχει ο admin για να έχει πρόσβαση στους χρήστες του συστήματος. Πηγαίνοντας στο αρχείο **users.php** από την επιλογή Users στη μπάρα περιήγησης.

```
<div id="page-wrapper">
<div class="row">
<div id="content">
<?php
echo "
<div class="row">
<div class="col-lg-6">
<h4 class="page-header">Insert last name to search for users</h4>
</div>
</div>";

echo "
<form class="col-lg-6" id="searchform" method="post" onsubmit="return false;">
<input autocomplete="off" id="searchbox" name="searchv" onkeyup="sendRequest()" onclick="sendRequest()" type="text" class="col-xs-4 form-control">
</form>
<br><br><br><br>
<div id="show_results"></div>

<script src="prototype.js" type="text/javascript"></script>
<script>
function sendRequest() {
    new Ajax.Updater('show_results', 'get_users.php', { method: 'post', parameters: $('searchform').serialize() });
}
</script>";
?>
</div>
</div>
<!-- /#page-wrapper -->
```

Εικόνα 19: users.php

Χρησιμοποιούμε πάλι ασύγχρονη μετάδοση δεδομένων για να εμφανίσουμε τους χρήστες στον διαχειριστή συστήματος. Τα δεδομένα τα λαμβάνουμε από το αρχείο **get_users.php**.

```
<?php
include ("dbcon.php");
session_start();
if (!$_SESSION["sess_username"]) {
    header("Location: unauthorized_error.php");
}
$privcheck = $_SESSION['sess_privileges'];
$searchv = $_POST['searchv'];
if(empty($searchv)) {
    $sql = "SELECT * FROM employee";
} else {
    //If there is text in the search field, this code is executed every time the input changes.
    $sql = "SELECT username, name, lastname, address, task FROM employee WHERE lastname LIKE '%$searchv%'";
}
$result = mysqli_query($con, $sql);
echo "<table id='results' class='table table-striped table-bordered table-hover'>
<tr>
<th>Name</th>
<th>Last Name</th>
<th>Address</th>
<th>Task</th>
";
echo "
<th colspan='2'>Edit</th>
</tr>";
while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['name'] . "</td>";
    echo "<td>" . $row['lastname'] . "</td>";
    echo "<td>" . $row['address'] . "</td>";
    echo "<td>" . $row['task'] . "</td>";
    if ($privcheck == 'admin'){
        echo "<td><a href='remove_user.php?username=" . $row['username'] . "'>Delete</a></td>";
    }
    echo "</tr>";
}
echo "</table>";
?>
```

Εικόνα 20: get_users.php

Για να εισάγουμε νέους χρήστες στο σύστημα που είναι επιλογή μόνο για τον διαχειριστή συστήματος πηγαίνουμε στην μπάρα περιήγησης στην επιλογή Create new user και το αρχείο *new_user.php*. Εκεί θα μας εμφανίσει μία φόρμα όπου συμπληρώνουμε τα στοιχεία του χρήστη και πατάμε το αντίστοιχο κουμπί submit.

```
<div class="panel panel-default">
<div class="panel-heading">
    Complete all boxes
</div>
<div class="panel-body">
<div class="row">
<div class="col-lg-6">
<form role="form" action="insert_user.php" method="post">
<div class="form-group">
<label>Name</label>
<input class="form-control" name="name">
<label>Last Name</label>
<input class="form-control" name="lastname">
<label>Address</label>
<input class="form-control" name="address">
<label>Username</label>
<input class="form-control" name="username">
<label>Password</label>
<input class="form-control" type="password" name="password">
<label>Privileges</label>
<input class="form-control" name="privileges">
<label>Task</label>
<input class="form-control" name="task">
</div>
<button type="submit" class="btn btn-default">Submit</button>
</form>
</div>
</div>
<!-- /.row (nested) -->
</div>
<!-- /.panel-body -->
</div>
```

Εικόνα 21: new_user.php

Τα δεδομένα αποστέλλονται όπως και στην περίπτωση που εισάγουμε νέο πελάτη στο σύστημα στο αρχείο *insert_user.php* μέσω της μεθόδου POST και με μία εντολή INSERT INTO της MYSQL εισάγουμε τα στοιχεία του χρήστη στη βάση δεδομένων.


```
<?php
session_start();
if (!$_SESSION["sess_username"]) {
    header("Location: unauthorized_error.php");
}
if(isset($_SESSION['sess_username']) && isset($_SESSION['sess_privileges']))
{
    $usercheck = $_SESSION['sess_username'];
    $privcheck = $_SESSION['sess_privileges'];
}
else {
    $privcheck=1;
    $usercheck=0;
}
require_once('dbcon.php');
$sql="INSERT INTO employee (name, lastname, adress, username, password, privileges, task)
VALUES
('$_POST[name]', '$_POST[lastname]', '$_POST[adress]', '$_POST[username]', '$_POST[password]', '$_POST[privileges]', '$_POST[task]')";
if (!mysqli_query($con,$sql))
{
    die('Error: ' . mysqli_error($con));
}
```

Εικόνα 22: insert_user.php

Επίσης δίνεται η δυνατότητα στον διαχειριστή συστήματος να διαγράψει πελάτες και χρήστες από τη βάση δεδομένων. Πηγαίνοντας στην καρτέλα Customers βλέπουμε ότι υπάρχει η επιλογή για διαγραφή πελάτη.

Afm	Name	Last Name	Id	Address	Phones	Employees	Debt to us	Delete customer	Edit
123450	customer5	customer	tautoitita3	odos		1	0	Delete	Edit fpa Edit ika
123451	customer6	customer	tautoitita4	odos		1	0	Delete	Edit fpa Edit ika

Εικόνα 23: Καρτέλα πελατών

Αυτό γίνεται μέσω του αρχείου *remove_customer.php* όπου με την εντολή DELETE της MYSQL αφαιρείται ο πελάτης από το σύστημα. Το αρχείο παίρνει τα δεδομένα του πελάτη για να τον διαγράψει με τη μέθοδο GET.

```
<?php
session_start();
if (!$_SESSION["sess_username"]) {
    header("Location: unauthorized_error.php");
}
if(isset($_SESSION['sess_username']) && isset($_SESSION['sess_privileges']))
{
    $usercheck = $_SESSION['sess_username'];
    $privcheck = $_SESSION['sess_privileges'];
}
else {
    $privcheck=1;
    $usercheck=0;
}
require_once('dbcon.php');

$afm = $_GET["takeafm"];
echo $id;

$sql="DELETE FROM customer WHERE afm = '$afm'";

if (!mysqli_query($con,$sql))
{
    die('Error: ' . mysqli_error($con));
}
```

Εικόνα 24: remove_dustomer.php

Ομοίως και για τη διαγραφή χρηστών, πάλι μόνο από τον διαχειριστή συστήματος.

```
session_start();
if (!$SESSION["sess_username"]) {
    header("Location: unauthorized_error.php");
}

if(isset($_SESSION['sess_username']) && isset($_SESSION['sess_privileges']))
{
    $usercheck = $_SESSION['sess_username'];
    $privcheck = $_SESSION['sess_privileges'];
}
else {
    $privcheck=1;
    $usercheck=0;
}
require_once('dbcon.php');

$username = $_GET["username"];
echo $id;

$sql="DELETE FROM employee WHERE username = '$username'";

if (!mysqli_query($con,$sql))
{
    die('Error: ' . mysqli_error($con));
}
```

Εικόνα 25: remove_user.php

Η μπάρα περιήγησης, ένα πολύ σημαντικό κομμάτι του συστήματος έχει υλοποιηθεί στο αρχείο *menu.php*, έτσι ώστε να μην εμφανίζει τις επιλογές για προβολή χρηστών, εισαγωγή χρήστη και εισαγωγή πελάτη, παρά μόνο στον διαχειριστή συστήματος.

```
<?php
if ($privcheck === 0) {
    echo " ";
}
else {
    echo '<li>
        <a href="main.php"><i class="fa fa-circle fa-fw"></i> Dashboard</a>
    </li>
    <li>
        <a href="customers.php"><i class="fa fa-circle fa-fw"></i> Customers</a>
    </li>';
    if ($privcheck == "admin")
    {
        echo '<li>
            <ul class="nav nav-second-level collapse in">
                <li>
                    <a href="new_customer.php">Create new customer</a>
                </li>
            </ul>
            <!-- /.nav-second-level -->
        </li>';
    }

    if ($privcheck == "admin")
    {
        echo '<li>
            <a href="users.php"><i class="fa fa-circle fa-fw"></i> Users<span class="fa arrow"></span></a>
            <ul class="nav nav-second-level collapse in">
                <li>
                    <a href="new_user.php">Create new user</a>
                </li>
            </ul>
            <!-- /.nav-second-level -->
        </li>';
    }
}
?>
```

Εικόνα 26: menu.php

Για την υλοποίηση τώρα της εξυπηρέτησης των πελατών, δηλαδή τον υπολογισμό του ΦΠΑ και τον υπολογισμό των εισφορών στο ΙΚΑ, δημιουργήθηκαν δύο νέα αρχεία, το *fpa.php* και το *ika.php*. Εκεί παίρνουμε το ΑΦΜ του πελάτη με τη μέθοδο GET και τυπώνουμε αναλυτικά τα στοιχεία του, όπως έσοδα, έξοδα και το τρέχον ΦΠΑ. Πληκτρολογούμε τα καινούρια έξοδα, τα καινούρια έσοδα και πατώντας το κουμπί SUBMIT οδηγούμαστε στο αρχείο *calc_fpa.php* όπου πραγματοποιείται ο υπολογισμός και ενημερώνεται κατάλληλα η βάση δεδομένων.

```
<?php
session_start();
if (!$SESSION["sess_username"]) {
    header("Location: unauthorized_error.php");
}
if(isset($SESSION['sess_username']) && isset($SESSION['sess_privileges']))
{
    $usercheck = $SESSION['sess_username'];
    $privcheck = $SESSION['sess_privileges'];
}
else {
    $privcheck=1;
    $usercheck=0;
}
require_once('dbcon.php');
$afm = $SESSION['afm'];
$doldexp = $SESSION['expenses'];
$doldinc = $SESSION['income'];
$doldfpa = $SESSION['fpa'];
$doldxrewsi = $SESSION['xrewsi'];
$expenses = $_POST['expenses'];
$income = $_POST['income'];

$doldexp = $doldexp + $expenses;
$doldinc = $doldinc + $income;
$fpa = $doldinc - $doldexp;
//$oldfpa = $oldfpa + $fpa;
$doldxrewsi = $doldxrewsi + 50;

$sql = "UPDATE customer
SET exoda = ".$doldexp.", esoda = ".$doldinc.", fpa = ".$fpa.", xrewsi = ".$doldxrewsi."
WHERE afm=".$afm."";

if (!mysqli_query($con,$sql))
{
    die('3Error: ' . mysqli_error($con));
}
```

Εικόνα 27: calc_fpa.php

Για τον υπολογισμό των εισφορών στο ΙΚΑ οδηγούμαστε στο αρχείο *calc_ika.php* όπου με τους κατάλληλους υπολογισμούς βρίσκουμε τις εισφορές του πελάτη και τις καταχωρούμε στη βάση δεδομένων.

```
<?php
session_start();
if (!$SESSION["sess_username"]) {
    header("Location: unauthorized_error.php");
}
if(isset($SESSION['sess_username']) && isset($SESSION['sess_privileges']))
{
    $usercheck = $SESSION['sess_username'];
    $privcheck = $SESSION['sess_privileges'];
}
else {
    $privcheck=1;
    $usercheck=0;
}
require_once('dbcon.php');
$afm = $SESSION['afm'];
$doldempl = $SESSION['employees'];
$doldxrewsi = $SESSION['xrewsi'];
$newemployees = $_POST['employees'];

$doldempl = $doldempl + $newemployees;
$ika = (586 * 0.25) * $doldempl;
$doldxrewsi = $doldxrewsi + 70;

$sql = "UPDATE customer
SET ika = ".$ika.", employees = ".$doldempl.", xrewsi = ".$doldxrewsi."
WHERE afm=".$afm."";

if (!mysqli_query($con,$sql))
{
    die('3Error: ' . mysqli_error($con));
}
```

Εικόνα 28: calc_ika.php

Τέλος για την αλλαγή κωδικού πρόσβασης του χρήστη πρέπει να πάμε πάνω δεξιά σε οποιαδήποτε σελίδα στις ρυθμίσεις, που μας οδηγεί στο αρχείο *user_settings.php*. Εκεί, σε μία φόρμα πληκτρολογούμε τον παλιό κωδικό, τον καινούριο κωδικό και την επιβεβαίωση του καινούριου κωδικού. Με τη μέθοδο POST στέλνουμε τα δεδομένα στο αρχείο *change_password.php* όπου πραγματοποιείται η ενημέρωση της βάσης δεδομένων για την αποθήκευση του καινούριου κωδικού πρόσβασης που έχει επιλέξει ο χρήστης.

```
<?php
session_start();
if (!$SESSION["sess_username"]) {
    header("Location: unauthorized_error.php");
}
if(isset($SESSION['sess_username']) && isset($SESSION['sess_privileges']))
{
    $usercheck = $SESSION['sess_username'];
    $privcheck = $SESSION['sess_privileges'];
}
else {
    $privcheck=1;
    $usercheck=0;
}
require_once('dbcon.php');

$oldpass = $_POST['oldpass'];
$newpass = $_POST['newpass'];
$confirmpass = $_POST['confirmpass'];

$sql = "UPDATE employee
SET password='$newpass'
WHERE username='$usercheck'";

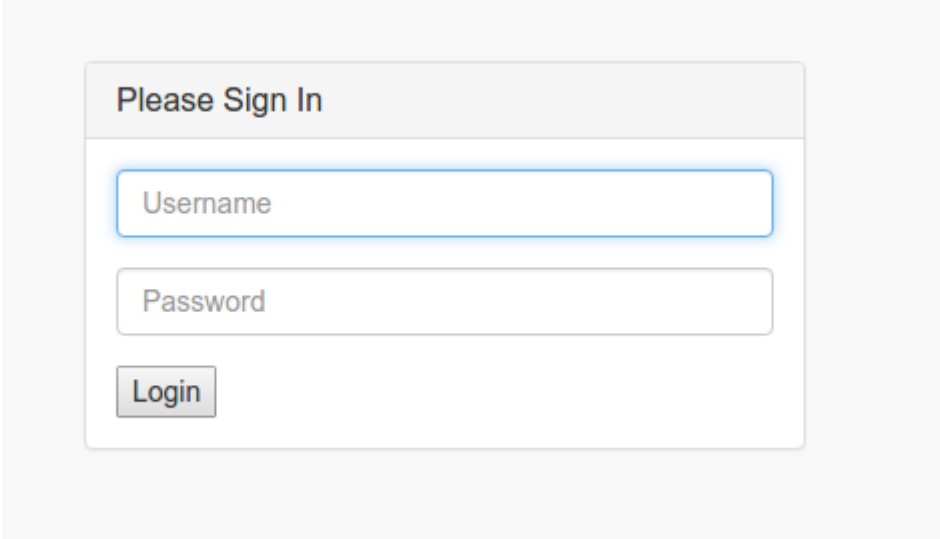
if (!mysqli_query($con,$sql))
{
    die('3Error: ' . mysqli_error($con));
}
```

Εικόνα 29: change_password.php

4.3 ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ

4.3.1 ΚΑΝΟΝΤΑΣ SIGN IN

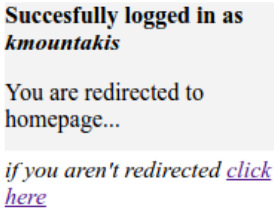
Έστω λοιπόν ότι θέλουμε να συνδεθούμε στο σύστημά μας. Συμπληρώνουμε την φόρμα που βλέπουμε παρακάτω και πατάμε Login .



The image shows a login form with a light gray background. At the top, there is a header box with the text "Please Sign In". Below this, there are two input fields: the first is labeled "Username" and the second is labeled "Password". Below the password field is a button labeled "Login".

Εικόνα 30: Login

Αμέσως μετά , θα εμφανιστεί μήνυμα για την επιτυχή σύνδεσή μας στο σύστημα και το μήνυμα ανακατεύθυνσης στην αρχική σελίδα. Τέλος , αν δεν πραγματοποιηθεί η ανακατεύθυνση αυτόματα , τότε μπορούμε να επιλέξουμε [click here](#) για να γίνει η ανακατεύθυνση .

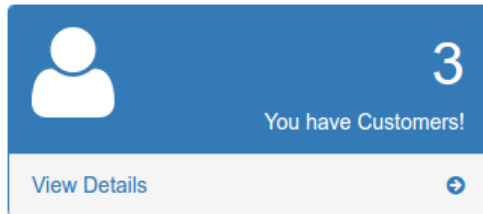


The image shows a success message in a light gray box. The text reads: "Successfully logged in as kmountakis", "You are redirected to homepage...", and "if you aren't redirected [click here](#)".

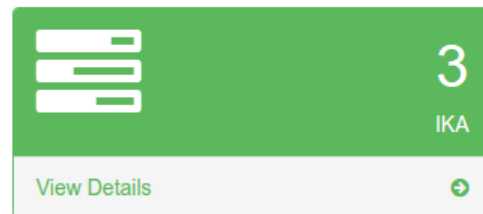
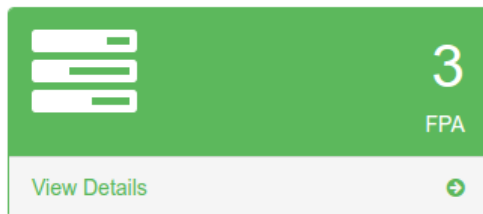
Εικόνα 31: Redirect 1

4.3.2 DASHBOARD

Dashboard



Tasks



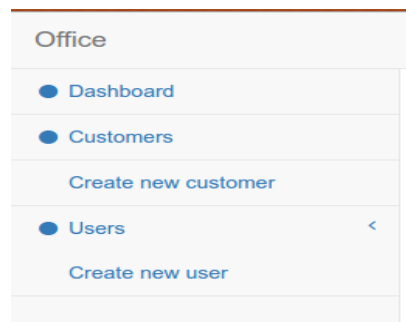
Εικόνα 32: Dashboard

4.3.3 SIDEBAR

Επίσης, στα αριστερά παρατηρούμε την μπάρα περιήγησης συστήματος η οποία αποτελείται από τις παρακάτω επιλογές:

Έστω ότι έχουμε συνδεθεί ως administrator (διαχειριστής) στο σύστημα.

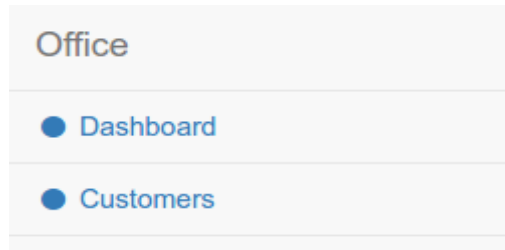
- Την επιλογή **Dashboard**, η οποία οδηγεί στην αρχική σελίδα «ακαριαία» από που και αν βρισκόμαστε στο σύστημα.
- Την επιλογή **Customers**, η οποία μας δίνει την δυνατότητα (ως administrator) να προσθέσουμε πελάτες στο σύστημα, όπως επίσης και να διαχειριστούμε τις εργασίες των πελατών.
- Την επιλογή **Users**, η οποία μας δίνει την δυνατότητα (ως administrator) να προσθέσουμε χρήστες στο σύστημα.



Εικόνα 33: Admin Sidebar

Έστω ότι έχουμε συνδεθεί ως user (χρήστης) στο σύστημα.

- Την επιλογή **Dashboard** , η οποία οδηγεί στην αρχική σελίδα «ακαριαία» από που και αν βρισκόμαστε στο σύστημα.
- Την επιλογή **Customers** , η οποία μας δίνει την δυνατότητα (ως user) να διαχειριστούμε τις εργασίες των πελατών.

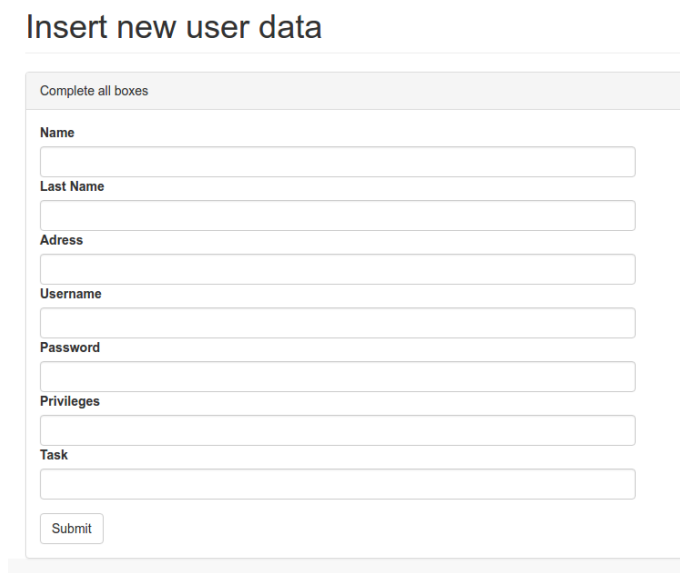


Εικόνα 34: User Sidebar

4.3.4 ΠΡΟΣΘΗΚΗ ΝΕΟΥ ΧΡΗΣΤΗ

Έστω λοιπόν ότι έχουμε συνδεθεί στο σύστημα ως administrator και θέλουμε να πραγματοποιήσουμε προσθήκη νέου χρήστη. Πηγαίνουμε στην φόρμα που βλέπουμε παρακάτω πατώντας στην μπάρα περιήγησης στον σύνδεσμο admin – user. Το μόνο που έχουμε να κάνουμε πλέον είναι να προσθέσουμε τα στοιχεία και να πατήσουμε Submit για να ολοκληρωθεί η διαδικασία. Η φόρμα αποτελείται από τα εξής πεδία:

- **Name:** Εισαγωγή ονόματος.
- **Last Name:** Εισαγωγή επιθέτου.
- **Address:** Εισαγωγή διεύθυνσης.
- **Username:** Εισαγωγή ονόματος χρήστη.
- **Password:** Εισαγωγή κωδικού πρόσβασης.
- **Privileges:** Δικαιώματα χρήστη. Θα ορίζεται ποιες λειτουργίες θα μπορεί να πραγματοποιήσει αυτό ο χρήστης.
- **Task:** Η εργασία που έχει ανατεθεί στον χρήστη.

The image shows a web form titled "Insert new user data". At the top, there is a grey bar with the text "Complete all boxes". Below this, there are seven input fields, each with a label above it: "Name", "Last Name", "Address", "Username", "Password", "Privileges", and "Task". At the bottom left of the form is a "Submit" button.

Εικόνα 35: Εισαγωγή νέου χρήστη

Όταν ολοκληρωθεί η διαδικασία , υπάρχει ένα μήνυμα επαλήθευσης εισαγωγής χρήστη που ουσιαστικά επιβεβαιώνει την προσθήκη χρήστη. Τέλος , αν δεν πραγματοποιηθεί η ανακατεύθυνση αυτόματα , τότε μπορούμε να επιλέξουμε [click here](#) για να γίνει η ανακατεύθυνση .

User inserted succesfully!
You are redirected to homepage...
if you aren't redirected [click here](#)

Εικόνα 36: Redirect 2

4.3.5 ΠΡΟΒΟΛΗ ΧΡΗΣΤΩΝ

Επίσης , μπορούμε να κάνουμε αναζήτηση χρήστη χρησιμοποιώντας το επώνυμό του , όπως φαίνεται παρακάτω.

Insert last name to search for users

Name	Last Name	Adress	Task	Edit
kiriakos	mountakis	hania	admin	Delete
name	lastname	adress	ika	Delete

Εικόνα 37: Καρτέλα χρηστών

Τέλος, υπάρχει η δυνατότητα διαγραφής κάποιου χρήστη . Αυτή η λειτουργία γίνεται μόνο από τον administrator. Αν δεν πραγματοποιηθεί η ανακατεύθυνση αυτόματα , τότε μπορούμε να επιλέξουμε [click here](#) για να γίνει η ανακατεύθυνση .

User removed succesfully!

You are redirected to
homepage...

if you aren't redirected [click
here](#)

Εικόνα 38: Redirect 3

4.3.6 ΠΡΟΣΘΗΚΗ ΠΕΛΑΤΩΝ

Έστω λοιπόν ότι έχουμε συνδεθεί στο σύστημα ως administrator και θέλουμε να πραγματοποιήσουμε προσθήκη πελάτη. Πηγαίνουμε στην φόρμα που βλέπουμε παρακάτω πατώντας στην μπάρα περιήγησης στον σύνδεσμο admin – customer. Το μόνο που έχουμε να κάνουμε πλέον, είναι να προσθέσουμε τα στοιχεία και να πατήσουμε Submit για να ολοκληρωθεί η διαδικασία. Η φόρμα αποτελείται από τα εξής πεδία:

- **Afm:** Εισαγωγή ΑΦΜ.
- **Name:** Εισαγωγή Ονόματος
- **Last name:** Εισαγωγή επιθέτου.
- **Id:** Αριθμός ταυτότητας.
- **Debt:** Χρέωση.
- **Address:** Εισαγωγή Διεύθυνσης.
- **Phones:** Εισαγωγή τηλεφώνων επικοινωνίας.
- **No of Employees:** Αριθμός Υπαλλήλων.

Insert new customer data

Complete all boxes

Afm

Name

Last name

Id

Debt

Adress

Phones

No Of Employees

Submit

Εικόνα 39: Εισαγωγή πελατών

4.3.7 ΔΙΑΧΕΙΡΙΣΗ ΚΑΡΤΕΛΑΣ ΠΕΛΑΤΗ

Σε αυτό το σημείο πραγματοποιείται διαχείριση της καρτέλας πελάτη. Ως administrator , μπορούμε να βλέπουμε παραπάνω πληροφορίες απ'ότι ένας χρήστης. Για παράδειγμα , βλέπουμε όλες τις εργασίες που χρειάζεται ο πελάτης . Εάν είχαμε συνδεθεί ως χρήστης , τότε θα βλέπαμε μόνο τις εργασίες που έχουμε ως αρμοδιότητα να εκτελέσουμε για τον συγκεκριμένο πελάτη.

Επίσης , ως administrator μπορούμε να βλέπουμε και την πληροφορία οφειλής ή μη, του πελάτη προς το γραφείο.

Το μόνο που χρειάζεται είναι να πληκτρολογήσουμε το ΑΦΜ του πελάτη ή ,το επώνυμό του. Όσο πληκτρολογούμε , τόσο πιο συγκεκριμένες γίνονται οι φράσεις αναζήτησης και έτσι μειώνονται οι επιλογές μας.

Insert AFM or last name to search for customers

Afm	Name	Last Name	Id	Address	Phones	Employees	Debt to us	Delete customer	Edit
123450	customer5	customer	tautoita3	odos		1	0	Delete	Edit fpa Edit ika
123451	customer6	customer	tautoita4	odos		1	0	Delete	Edit fpa Edit ika
123452	customer7	customer	tautoita5	odos		1	0	Delete	Edit fpa Edit ika
123453	customer8	customer	tautoita6	odos		1	0	Delete	Edit fpa Edit ika
123454	customer9	customer	tautoita7	odos		1	0	Delete	Edit fpa Edit ika
123455	customer0	customer	tautoita8	odos		1	0	Delete	Edit fpa Edit ika
123457	customer2	customer	tautoita	odos		1	0	Delete	Edit fpa Edit ika
123458	customer3	customer	tautoita1	odos		1	0	Delete	Edit fpa Edit ika
123459	customer4	customer	tautoita2	odos		1	0	Delete	Edit fpa Edit ika

Εικόνα 40: Καρτέλα πελάτη

Τέλος, υπάρχει η δυνατότητα διαγραφής κάποιου πελάτη . Αυτή η λειτουργία γίνεται μόνο από τον administrator. Αν δεν πραγματοποιηθεί η ανακατεύθυνση αυτόματα , τότε μπορούμε να επιλέξουμε [click here](#) για να γίνει η ανακατεύθυνση στην αρχική σελίδα .

**Customer removed
succesfully!**

You are redirected to
homepage...

if you aren't redirected [click here](#)

Εικόνα 41: Redirect 4

4.3.8 ΥΠΟΛΟΓΙΣΜΟΣ ΦΠΑ

Εδώ βλέπουμε ένα σύντομο υπολογισμό του ΦΠΑ του πελάτη. (Η κίνηση αυτή πραγματοποιείται είτε ήμαστε συνδεδεμένοι στο σύστημα ως administrator είτε ως user). Το μόνο που έχουμε να κάνουμε είναι να δώσουμε τους αριθμούς εσόδων και εξόδων (income και expenses αντίστοιχα) και στην συνέχεια να πατήσουμε το submit.

Calculate Fpa for customer

Afm	Name	Last Name	Address	Expenses	Income	Current fpa
123456789	customer	customer	street	0		

Calculate new fpa

Expenses

Income

Εικόνα 42: Υπολογισμός ΦΠΑ

4.3.9 ΥΠΟΛΟΓΙΣΜΟΣ ΙΚΑ

Αν ο πελάτης απασχολεί προσωπικό , τότε μπορούμε να υπολογίσουμε σύντομα το ΙΚΑ του συμπληρώνοντας απλά τον αριθμό των υπαλλήλων που έχει. Επίσης, υπάρχει η δυνατότητα να προσθέσουμε νέους υπαλλήλους στην επιχείρηση. (Η κίνηση αυτή πραγματοποιείται είτε ήμαστε συνδεδεμένοι στο σύστημα ως administrator είτε ως user). Στην συνέχεια πατάμε το submit.

Calculate Ika for customer

Afm	Name	Last Name	Address	Employees	Ika
123456789	customer	customer	street	1	

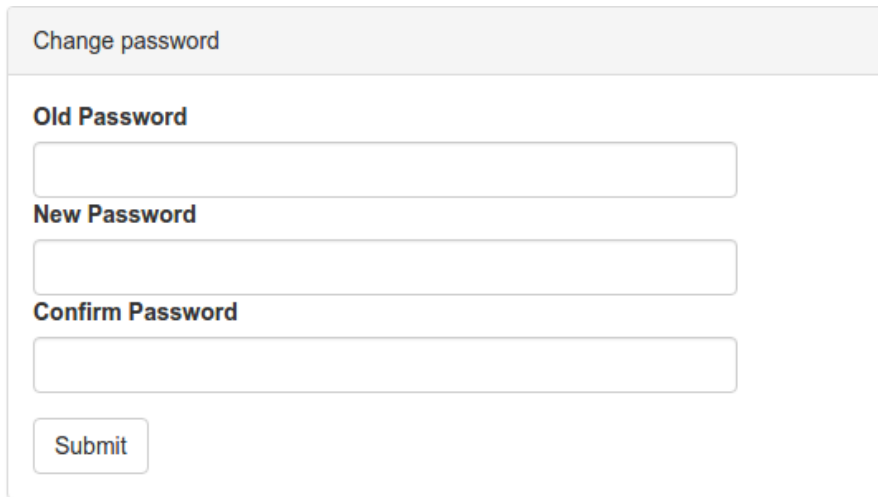
Calculate new Ika

New Employees

Εικόνα 43: Υπολογισμός ΙΚΑ

4.3.10 ΡΥΘΜΙΣΕΙΣ ΧΡΗΣΤΗ

Για κάθε χρήστη που είναι εγγεγραμμένος στο σύστημα , δίνεται η δυνατότητα αλλαγής password ,μέσω της συμπλήρωσης φόρμας. Η διαδικασία φαίνεται παρακάτω.



Change password

Old Password

New Password

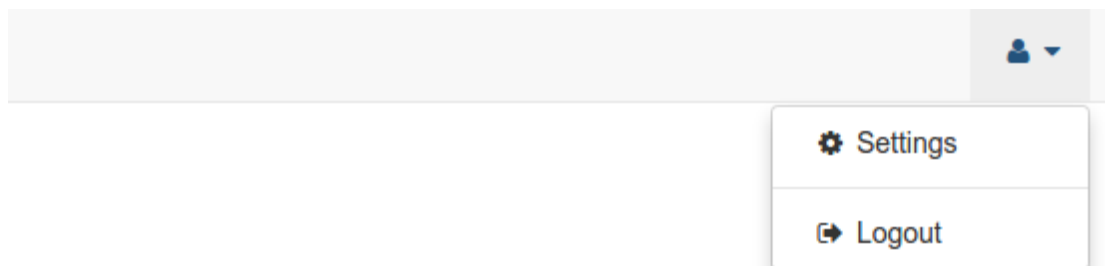
Confirm Password

Submit

Εικόνα 44: Αλλαγή κωδικού χρήστη

4.3.11 LOGOUT

Κοιτάζοντας πάνω δεξιά στο σύστημα βλέπουμε τις επιλογές settings (που την χρησιμοποιήσαμε παραπάνω) και Logout. Επιλέγοντας Logout πραγματοποιείται αποσύνδεση του χρήστη και γίνεται ανακατεύθυνση στην φόρμα εισαγωγής χρήστη στο σύστημα.



Εικόνα 45: Αποσύνδεση χρήστη

Ακολουθεί το μήνυμα επιβεβαίωσης και ανακατεύθυνσης. Αν δεν πραγματοποιηθεί η ανακατεύθυνση αυτόματα , τότε μπορούμε να επιλέξουμε [click here](#) για να γίνει η ανακατεύθυνση στην αρχική σελίδα

Successfully logged out
You are redirected to
homepage...
if you aren't redirected [click here](#)

Εικόνα 46: Redirect 5

ΚΕΦΑΛΑΙΟ 5 | ΑΠΟΤΕΛΕΣΜΑΤΑ

5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Το σύστημα που αναπτύχθηκε για την πτυχιακή εργασία, διευκολύνει αρκετά ένα λογιστικό γραφείο στο να διεκπεραιώνει τις εργασίες προς τους πελάτες του, προσθέτοντας ταχύτητα αλλά και εργονομία στον υπολογισμό της απόδοσης ΦΠΑ και τον υπολογισμό των εισφορών του εργοδότη προς το ΙΚΑ. Προσφέρει εικόνα όλων των πελατών γρήγορα και εύκολα στους υπαλλήλους του γραφείου αλλά μπορεί να παρέχει και πιο λεπτομερή στοιχεία με το απλό πάτημα ενός πλήκτρου ποντικιού.

Διευκολύνει πάρα πολύ τον ιδιοκτήτη του λογιστικού γραφείου προσφέροντάς του εικόνα όλων των πελατών με στοιχεία που δεν μπορούν να δουν οι υπάλληλοι, όπως οι οφειλές του πελάτη προς το γραφείο, αλλά επίσης δίνει και μία εικόνα των χρηστών του συστήματος, δηλαδή των υπαλλήλων. Παρέχει ευκολία στον ιδιοκτήτη στο να εισάγει πελάτες και να τους διαγράψει, αλλά και στο να εισάγει νέους χρήστες και να τους διαγράψει.

Προσφέρει οφέλη και στους πελάτες καθώς εξυπηρετούνται πολύ πιο γρήγορα, αλλά και εκμηδενίζει την πιθανότητα λάθους στην εξυπηρέτησή τους

5.2 ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ

Όσον αφορά τις μελλοντικές επεκτάσεις του συστήματος, υπάρχει σκοπός να προστεθούν αρκετές λειτουργίες, οι οποίες όταν υλοποιηθούν θα δώσουν στα χέρια των υπαλλήλων μία λύση για όλες τις εργασίες που θα πρέπει να φέρουν εις πέρας. Κάποιες από αυτές είναι:

- Προσθήκη συστήματος αυτόματης ενημέρωσης του πελάτη, όταν διεκπεραιωθεί κάποια εργασία. Η ενημέρωση αυτή σχεδιάζεται να πραγματοποιείται με δύο τρόπους:
 - Με αποστολή e-mail.
 - Με αποστολή SMS.
- Προσθήκη λειτουργίας πλήρους συμπλήρωσης της φορολογικής δήλωσης του πελάτη.
- Λειτουργία συλλογής αποδείξεων.
- Λειτουργία πρόσθετης δήλωσης Ε9.
- Σύστημα απογραφής για τη επιχείρηση του πελάτη.

Με όλες αυτές τις λειτουργίες πιστεύουμε πως το σύστημα θα μπορέσει να υποστηρίξει πλήρως τις ανάγκες ενός λογιστικού γραφείου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <http://computer.howstuffworks.com/web-server4.htm>,
- [2] https://en.wikipedia.org/wiki/Apache_HTTP_Server,
- [3] https://el.wikipedia.org/wiki/Apache_HTTP_εξυπηρέτησης,
- [4] <https://en.wikipedia.org/wiki/MySQL>,
- [5] <https://el.wikipedia.org/wiki/MySQL>,
- [6] <https://en.wikipedia.org/wiki/PHP>,
- [7] <https://el.wikipedia.org/wiki/PHP>,
- [8] [https://en.wikipedia.org/wiki/LAMP_\(software_bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle)),
- [9] http://www.w3schools.com/php/php_intro.asp,
- [10] <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>,
- [11] http://www.tutorialspoint.com/php/php_and_mysql.htm,
- [12] <https://en.wikipedia.org/wiki/MySQLi>,
- [13] http://www.w3schools.com/php/php_ref_mysql_i.asp,

ΠΑΡΑΡΤΗΜΑ 1: ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός συστήματος μηχανογράφησης που θα χρησιμοποιηθεί σε ένα λογιστικό γραφείο. Στόχος είναι η ταχύτερη διεκπεραίωση των εργασιών του λογιστικού γραφείου, κάτι που συνεπάγεται την ταχύτερη εξυπηρέτηση των πελατών του γραφείου.

Σχεδιάστηκε και υλοποιήθηκε μία βάση δεδομένων, η οποία είναι γενικά ένα πολύ χρήσιμο εργαλείο για μηχανογράφηση γραφείων, κυρίως λόγω της ταχύτητας που προσφέρει στην πρόσβαση των δεδομένων, αλλά και της ασφάλειας που παρέχει.

Η ανάπτυξη του γραφικού περιβάλλοντος χρήστη πραγματοποιήθηκε με τη χρήση της HTML5. Η υλοποίηση της σύνδεσης της βάσης δεδομένων με το γραφικό περιβάλλον χρήστη υλοποιήθηκε στην scripting γλώσσα PHP. Πρόκειται για μία ευρύτατα χρησιμοποιούμενη λύση, επειδή παρέχει πολύ καλή λειτουργικότητα αλλά και ταχύτητα στο τελικό μας σύστημα, ενώ ταυτόχρονα μας προσφέρει ασφάλεια για την αλληλεπίδραση μας με τη βάση δεδομένων.

Για τη βάση δεδομένων χρησιμοποιήθηκε η MYSQL. Πλεονεκτήματα της είναι:

- Είναι εύκολη στη χρήση. Ενώ απαιτείται μία βασική γνώση SQL, και οι περισσότερες σχεσιακές βάσεις δεδομένων το απαιτούν, μόνο με κάποιες απλές εντολές SQL, μπορούμε να αλληλοεπιδράσουμε με μια βάση δεδομένων MYSQL.
- Είναι ασφαλής. Μπορούμε να δώσουμε δικαιώματα μόνο σε όποιον χρήστη επιθυμούμε εμείς. Οι κωδικοί είναι κρυπτογραφημένοι.
- Είναι δωρεάν.
- Είναι γρήγορη.
- Είναι κλιμακούμενη. Η MYSQL μπορεί να διαχειριστεί οποιαδήποτε ποσότητα δεδομένων, ακόμη και μέχρι και επιπλέον από 50 εκατομμύρια σειρές.
- Χειρίζεται εξαιρετικά τη μνήμη.
- Μπορεί να τρέξει στα περισσότερα λειτουργικά συστήματα.

Στο κεφάλαιο 1 της πτυχιακής παρουσιάζεται μία σύνοψη της εργασίας.

Στο κεφάλαιο 2 παρουσιάζονται οι μέθοδοι που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος μηχανογράφησης. Επίσης παρατίθενται οι θεωρίες που μελετήθηκαν για τις απαιτούμενες πληροφορίες που οδήγησαν στην περάτωση της εργασίας. Τέλος παρουσιάζονται τα μοντέλα λογισμικού που χρησιμοποιήθηκαν και πιο συγκεκριμένα η στοιβιά LAMP, η HTML, το BOOTSTRAP και το AJAX.

Στο κεφάλαιο 3 παρατίθεται το χρονοδιάγραμμα που ακολουθήθηκε κατά τη διάρκεια ανάπτυξης του συστήματος, καθώς και οι σημαντικότεροι στόχοι που επιτεύχθηκαν.

Στο κεφάλαιο 4 παρουσιάζεται το κύριο μέρος της πτυχιακής. Πραγματοποιείται ανάλυση του προβλήματος σε όλες τις πτυχές του. Παρατίθενται οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος. Παρουσιάζεται η διαδικασία που ακολουθήθηκε για την υλοποίηση του συστήματος αναλυτικά. Περιληπτικά αποτελείται από τα εξής βήματα:

- Σχεδίαση της βάσης δεδομένων.
- Δοκιμαστική λειτουργία της βάσης δεδομένων.
- Σχεδίαση του γραφικού περιβάλλοντος χρήστη.
- Ανάπτυξη συστήματος χρηστών με χρήση της PHP.
- Υλοποίηση λειτουργίας υπολογισμού της απόδοσης ΦΠΑ των πελατών.
- Υλοποίηση λειτουργίας υπολογισμού των εισφορών στο ΙΚΑ των πελατών.
- Υλοποίηση εισαγωγής και διαγραφής πελατών στη βάση δεδομένων.
- Υλοποίηση εισαγωγής και διαγραφής χρηστών στη βάση δεδομένων.
- Ανάπτυξη διαφορετικού γραφικού περιβάλλοντος χρήστη για τον διαχειριστή συστήματος.

Τέλος στο κεφάλαιο 4 παρουσιάζεται ένα πλήρες εγχειρίδιο χρήσης του συστήματος, με εικονογράφηση.

Στο κεφάλαιο 5 παρουσιάζονται τα συμπεράσματα που βγήκαν από την ανάπτυξη του συστήματος μηχανογράφησης για λογιστικό γραφείο. Επίσης παρατίθενται πιθανές μελλοντικές λειτουργίες και επεκτάσεις του συστήματος, όπως η εγκατάσταση της λειτουργίας ενημέρωσης των πελατών μέσω ηλεκτρονικού ταχυδρομείου και μέσω SMS.

Στο παράρτημα 1 δίνεται μία περίληψη της πτυχιακής και στο παράρτημα 2 δίνονται οι διαφάνειες από την παρουσίαση της πτυχιακής