

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Γραφικής Διεπαφής Εισαγωγής και Απεικόνισης Δεδομένων Διαχείρισης Έργων
Όνοματεπώνυμο Φοιτητή	Μαυρογιαννόπουλος Παναγιώτης
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΣΠ/ 09055
Επιβλέπων	Δημήτριος Αποστόλου, Επίκουρος Καθηγητής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ABSTRACT

The concept of project management is to continuously improve initiatives. Since the industrial revolution, improvements in product and service delivery, to the early 1950s, when the project management applications were running on huge computers (mainframes), huge strides of progress have been made in the way data and the image of the project is presented to the specialized software manager.

Although tools and methodologies continue to evolve, the basics of project management remain the same relating to people, processes and technologies that work in a dynamic group to make things better, faster and more efficiently.

The main objective of this thesis is the design and implementation of a web application organization and presentation projects using php language and database mysql.

This application was designed to help a project manager to develop a plan, assigning resources to tasks, tracking progress , managing the budget and analyzing the workload.

ΕΙΣΑΓΩΓΗ

Η διοίκηση και διαχείριση έργου είναι ένα σύνολο συνεχόμενων πρωτοβουλιών βελτίωσης. Από τη βιομηχανική επανάσταση και τις βελτιώσεις στη παράδοση προϊόντων και υπηρεσιών έως και τις αρχές της δεκαετίας του 1950 που η διαχείριση έργου έτρεχε σε τεράστιους υπολογιστές (mainframes) καθώς και την εξέλιξη μέχρι σήμερα που το εξειδικευμένο λογισμικό παρέχει στον manager έμμεσα την εικόνα του έργου, έχουν γίνει τεράστια βήματα σε πολύ σύντομο χρονικό διάστημα.

Παρόλο που τα εργαλεία και οι μέθοδοι έχουν ίσως αλλάξει και συνεχίζουν να εξελίσσονται , τα βασικά στοιχεία της διαχείρισης έργου παραμένουν ίδια. Αφορούν ανθρώπους, διαδικασίες και τεχνολογίες που συνεργάζονται σε ένα δυναμικό σύνολο για να κάνουν τα πράγματα καλύτερα, πιο γρήγορα και πιο αποτελεσματικά.

Ο βασικός στόχος της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μίας διαδικτυακής εφαρμογής οργάνωσης και παρουσίασης έργων με χρήση γλώσσας php και βάσης δεδομένων mysql.

Αυτή η εφαρμογή σχεδιάστηκε για να βοηθήσει το διαχειριστή ενός έργου στην ανάπτυξη ενός σχεδίου, την ανάθεση πόρων σε εργασίες, την παρακολούθηση της προόδου, τη διαχείριση του προϋπολογισμού, καθώς και την ανάλυση του φόρτου εργασίας.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Πίνακας Περιεχομένων

ABSTRACT	4
ΕΙΣΑΓΩΓΗ	4
ΚΕΦΑΛΑΙΟ 1 ΕΡΓΟ	8
1.1 ΤΙ ΕΙΝΑΙ ΕΡΓΟ	8
1.2 ΕΡΓΑ ΠΛΗΡΟΦΟΡΙΚΗΣ	8
1.3 ΟΡΓΑΝΩΤΙΚΕΣ ΔΟΜΕΣ	9
ΚΕΦΑΛΑΙΟ 2 ΙΕΡΑΡΧΙΚΑ ΔΕΔΟΜΕΝΑ.....	12
2.1 ΜΟΝΤΕΛΟ ΛΙΣΤΑΣ ΓΕΙΤΩΝΙΑΣΗΣ(ADJACENCY LIST MODEL)	12
2.2 ΜΟΝΤΕΛΟ ΕΜΦΩΛΕΥΜΕΝΟΥ ΣΥΝΟΛΟΥ(NESTED SET MODEL)	14
2.3 ΣΥΓΚΡΙΣΗ ΤΩΝ 2 ΜΟΝΤΕΛΩΝ	16
ΚΕΦΑΛΑΙΟ 3 ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ	17
3.1 APACHE WEB SERVER	17
3.2 MYSQL	18
3.3 PHP	18
3.3.1 Ιστορική Αναδρομή	18
3.3.2 Η Λειτουργικότητα της Php	19
3.3.3 Τα πλεονεκτήματα της Php	21
3.4 GD LIBRARY(ΒΙΒΛΙΟΘΗΚΗ ΓΡΑΦΙΚΩΝ)	21
3.5 HTML	21
3.6 CSS	22
3.7 ΣΧΗΜΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	22
3.7.1 ΣΧΗΜΑ ΚΑΙ ΣΤΙΓΜΙΟΤΥΠΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	22
3.7.2 ΣΧΗΜΑ ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΕ.....	23
3.8 ΔΟΜΗ ΙΣΤΟΣΕΛΙΔΑΣ	24
ΚΕΦΑΛΑΙΟ 4 Η ΕΦΑΡΜΟΓΗ.....	26
4.1 ΕΙΣΑΓΩΓΗ ΕΡΓΩΝ.....	26
4.2 ΕΠΕΞΕΡΓΑΣΙΑ ΕΡΓΩΝ.....	28
4.3 ΔΙΑΓΡΑΦΗ ΕΡΓΩΝ	30
4.4 ΕΜΦΑΝΗΣΗ ΓΡΑΦΗΜΑΤΟΣ ΑΝΑ ΕΡΓΟ	31

4.5 Περιοχή Διαχείρισης	36
ΚΕΦΑΛΑΙΟ 5 ΠΑΡΑΡΤΗΜΑ	40
5.1 ΠΕΡΙΓΡΑΦΗ ΤΜΗΜΑΤΩΝ ΤΟΥ ΚΩΔΙΚΑ.....	40
ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΒΕΛΤΙΩΣΕΙΣ	68
ΒΙΒΛΙΟΓΡΑΦΙΑ	69

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΚΕΦΑΛΑΙΟ 1 ΕΡΓΟ

1.1 ΤΙ ΕΙΝΑΙ ΕΡΓΟ

Ένα έργο έχει πολύ γενική έννοια, αναπαριστά οποιαδήποτε προσχεδιασμένη εργασία που περιλαμβάνει έναν αριθμό ανθρώπων και δραστηριοτήτων, υπό την προϋπόθεση ότι πρόκειται για προσωρινή προσπάθεια, η οποία λαμβάνει χώρα προκειμένου να παραχθεί, δημιουργηθεί ένα μοναδικό προϊόν ή υπηρεσία

Για παράδειγμα η κατασκευή κτιρίων και δρόμων, Σχεδιασμός νέου προϊόντος, Πολιτική καμπάνια κ.α.

Το κάθε έργο έχει καθορισμένη έναρξη και λήξη. Μπορεί να αποτελείται από περισσότερα από ένα έργα και έχει «ανοιχτό» τέλος. Παρότι η λήξη του καθορίζεται από την αρχή, διάφοροι παράγοντες μπορούν να προκαλέσουν την παράταση του.

Τα έργα είναι προσωρινά, προσωρινό σημαίνει ότι κάθε έργο έχει προσδιορισμένη αρχή και τέλος. Το έργο τελειώνει όταν έχουν επιτευχθεί οι στόχοι του έργου ή καθίσταται ξεκάθαρο ότι οι στόχοι δεν πρόκειται ή δεν μπορούν να επιτευχθούν. Προσωρινό βέβαια δεν σημαίνει μικρής χρονικής διάρκειας, αντιθέτως πολλά έργα μπορούν να διαρκέσουν για αρκετά χρόνια. Παρότι η διάρκεια των έργων είναι πεπερασμένη τα έργα δεν μπορούν να οριστούν σαν προσπάθειες που εκτελούνται σε συνεχή βάση.

Ένα βασικό χαρακτηριστικό του έργου είναι η μοναδικότητα του, το αποτέλεσμα ενός έργου είναι μοναδικό ακόμα και εάν μπορεί να ενταχθεί σε μια ευρύτερη κατηγορία έργων π.χ. Κατασκευάζεται μεγάλος αριθμός κτιρίων γραφείων καθένα όμως είναι μοναδικό, άλλος ιδιοκτήτης & κατασκευαστής, άλλη αρχιτεκτονική, άλλη τοποθεσία, άλλο μέγεθος κλπ.

1.2 ΕΡΓΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Τα έργα πληροφορικής μπορεί να διαφέρουν σε μέγεθος, πολυπλοκότητα, τελικά προϊόντα, περιοχές εφαρμογής, πόρους, κλπ. Παράλληλα, τα μέλη της ομάδας έργων πληροφορικής μπορεί να διαφέρουν σημαντικά σε δεξιότητες και προηγούμενη εργασιακή εμπειρία.

Τα έργα πληροφορικής χρησιμοποιούν μεγάλο εύρος τεχνολογιών και απαιτούν ανάλογη τεχνογνωσία.

1.3 ΟΡΓΑΝΩΤΙΚΕΣ ΔΟΜΕΣ

Υπάρχουν 4 ειδών οργανωτικές δομές

- Οργανωτική δομή ανά λειτουργία
- Οργανωτική δομή ανά προϊόν/έργο
- Οργανωτική δομή πίνακα
 - Οργανωτική δομή αδύναμου πίνακα (ή πίνακα ανά λειτουργία)
 - Οργανωτική δομή ισορροπημένου πίνακα
 - Οργανωτική δομή δυνατού πίνακα (ή πίνακα ανά προϊόν/έργο)
- Μεικτές οργανωτικές δομές

Οργανωτική δομή ανά λειτουργία:

Η Διεύθυνση Πληροφορικής χωρίζεται σε τμήματα με λειτουργική εξειδίκευση π.χ. στην Ανάλυση Απαιτήσεων, Ανάπτυξη Λογισμικού, Έλεγχος Λογισμικού κλπ.

Οι δραστηριότητες που σχετίζονται με κάποια λειτουργία γίνονται στο πλαίσιο του αντίστοιχου τμήματος.

Τη διαχείριση (πόρων, προϋπολογισμού) αναλαμβάνουν οι υπεύθυνοι λειτουργιών (functional managers).

Τα πλεονεκτήματα αυτού του τρόπου δομής είναι:

- Ευελιξία στη χρήση Ανθρώπινων Πόρων
- Βελτιστοποίηση Διαχείρισης Γνώσης, με ομαδοποίηση του εξειδικευμένου προσωπικού σε διαφορετικές θεματικές περιοχές
- Επιτρέπεται η διατήρηση της συνέχειας στις διαφορετικές λειτουργίες
- Η επικοινωνία είναι «κάθετη» και αρκετά εύκολη
- Ο καθορισμός προτεραιοτήτων είναι εύκολος, αλλά εξαρτάται από τους υπεύθυνους των λειτουργικών περιοχών

Τα μειονεκτήματα είναι:

- Η ευθύνη για ένα έργο δεν είναι ατομική (συνήθως δημιουργούνται επιτροπές)
- Υπάρχει δυσκολία στην ανάθεση υπευθυνοτήτων (εφόσον δεν υπάρχει έμφαση προς και προγραμματισμός για τις δραστηριότητες των έργων)
- Ο συντονισμός είναι δύσκολος και χρονοβόρος
- Οι αποφάσεις συνήθως ευνοούν τις λειτουργίες που θεωρούνται πλέον σημαντικές
- Οι ιδέες κατευθύνονται σε λειτουργικά προβλήματα και όχι στην επίλυση προβλημάτων των έργων
- Δεν υπάρχει εστιακό σημείο προς τους πελάτες
- Η ανταπόκριση προς τις ανάγκες των πελατών είναι αργή
- Τα κίνητρα για την υλοποίηση των έργων είναι αρκετά μειωμένα

Οργανωτική δομή ανά έργο

Σε αυτή τη δομή απαιτείται η συγκέντρωση όλων των λειτουργικών ομάδων που χρειάζονται για ένα προϊόν ή έργο πληροφορικής στην ίδια ιεραρχία.

Ο υπεύθυνος έργου έχει πλήρη αρμοδιότητα για οργάνωση, στελέχωση αξιολόγηση και έλεγχο σε ολόκληρο τον κύκλο ζωής του έργου.

Τα έργα βρίσκονται ιεραρχικά κάτω από τη διεύθυνση η οποία ασκεί κεντρικό έλεγχο συντονισμό και προγραμματισμό όλων των έργων της διεύθυνσης.

Τα πλεονεκτήματα αυτής της δομής είναι

- Πλήρης εξουσιοδότηση για ολόκληρο τον κύκλο ζωής του έργου
- Σαφείς σχέσεις αναφοράς και υπευθυνότητας
- Ταχύτητα στη λήψη αποφάσεων
- Γρήγορος εντοπισμός και διόρθωση αστοχιών στο έργο
- Το προσωπικό επικεντρώνεται στην υλοποίηση των έργων
- Ανάπτυξη εστιακών σημείων για το εξωτερικό της επιχείρησης (π.χ. πελάτες)
- Ευελιξία στην επίτευξη των στόχων (χρονικών, κόστους και πόρων)
- Η διοίκηση διαθέτει περισσότερο χρόνο για συντονιστικές εργασίες

Τα μειονεκτήματα είναι:

- Μεγάλο κόστος για την επιχείρηση (λόγω του πολλαπλασιασμού των λειτουργικών ομάδων)
- Πιθανότητα μη σωστής κατανομής του προσωπικού
- Εξάρτηση από μεμονωμένα άτομα
- Οι αλλαγές προσώπων έχουν επιπτώσεις στην πρόοδο των έργων
- Έλλειψη τεχνικού συντονισμού μεταξύ διαφορετικών έργων
- Έλλειψη τεχνικής βελτίωσης λόγω επικέντρωσης στην υλοποίηση έργων

Οργανωτική δομή πίνακα

Στη Προσπάθεια συνδυασμού των πλεονεκτημάτων των δύο προηγούμενων οργανωτικών δομών βασίστηκε η οργανωτική δομή πίνακα η οποία και είναι ιδανική για τις εταιρείες πληροφορικής.

Οι βασικοί Κανόνες της δομής πίνακα απαιτούν ότι όλα τα στελέχη της διεύθυνσης συμμετέχουν στην εκπόνηση έργων. Οι υπεύθυνοι έργων και οι υπεύθυνοι των λειτουργικών μονάδων συμμετέχουν στην κατανομή του ανθρωποχρόνου και στην κατανομή των εργασιών και επιπλέον

,οι ομάδες εκπόνησης έργων λειτουργούν σαν αυτόνομες μονάδες κατά την διάρκεια της υλοποίησης των έργων.

Στην υλοποίηση έργων υπάρχουν τρεις μορφές οργανωτικής δομής πίνακα

- Αδύναμος (weak/functional matrix)
- Ισορροπημένος (balanced matrix)
- Δυνατός (strong/project matrix)

Στην οργανωτική Δομή Αδύναμου Πίνακα ο υπεύθυνος έργου επιβλέπει το έργο οριζόντια στις διάφορες λειτουργικές μονάδες και έχει περιορισμένο ρόλο, αυτόν του συντονισμού των λειτουργικών μερών που εμπλέκονται στο έργο .Η συγκεκριμένη δομή είναι κατάλληλη για μικρό οργανισμό, με μικρό αριθμό έργων, όπου η διοίκηση μπορεί να έχει συνολική εικόνα των έργων και μπορεί να αφιερώσει χρόνο στο συντονισμό τους.

Στην οργανωτική Δομή Ισορροπημένου Πίνακα ο υπεύθυνος έργου και οι υπεύθυνοι λειτουργιών διαχειρίζονται από κοινού το έργο.

Τέλος, η διοίκηση δεν αποτελεί το σημείο εστίασης και συντονισμού των έργων ,αντίθετα, ο υπεύθυνος έργου επιβλέπει το έργο και είναι αποκλειστικά υπεύθυνος για την ολοκλήρωσή του. Η ανάμειξη των υπεύθυνων λειτουργιών περιορίζεται στην ανάθεση προσωπικού και στην παροχή εξειδικευμένων συμβουλών.

Τα πλεονεκτήματα της οργανωτικής δομής είναι:

- Ο υπεύθυνος έργου διατηρεί όλο τον έλεγχο των πόρων του έργου και έχει την αρμοδιότητα κατανομής των πόρων
- Ταχύτητα στην λήψη αποφάσεων και στην ανάγκη διοικητικών αλλαγών
- Οι λειτουργικές μονάδες έχουν ρόλο υποστήριξης της εκπόνησης των έργων
- Ελαχιστοποίηση του κόστους των έργων με χρήση "κοινών" πόρων
- Εύκολη ανάπτυξη τεχνικής υποδομής και τεχνογνωσίας
- Ελαχιστοποίηση της έντασης διαμαχειών και συγκρούσεων
- Εξισορρόπηση κόστους, χρόνου και απόδοσης στην υλοποίηση των έργων

Τα μειονεκτήματα είναι:

- Πολυδιάστατη ροή πληροφοριών
- Πολυδιάστατη ροή εργασίας
- Διπλό σύστημα αναφορών
- Συνεχής διαφοροποίηση προτεραιοτήτων
- Διαφοροποίηση στόχων διοίκησης και έργων
- Δυσκολίες στον συνολικό έλεγχο
- Αύξηση των αναγκών για τον διοικητικό μηχανισμό

- Κάθε έργο λειτουργεί αυτόνομα, με κίνδυνο διπλασιασμού των δραστηριοτήτων
- Διαφοροποίηση στόχων μεταξύ λειτουργικών μονάδων και διοίκησης έργων
- Η διαχείριση διαμαχειών και συγκρούσεων είναι μία συνεχής διαδικασία
- Ασάφεια ρόλων, υπευθυνοτήτων, εξουσιών και αρμοδιοτήτων

Η σύνθετη οργανωτική δομή είναι αποτέλεσμα ύπαρξης στον ίδιο οργανισμό διαφορετικών οργανωτικών δομών τόσο ανά λειτουργία όσο και ανά έργο.

Για την επιλογή της κατάλληλης οργανωτικής δομής θα πρέπει να λάβουμε υπόψη ότι προτιμάμε οργανωτική δομή ανά λειτουργία όταν τα έργα εστιάζονται σε εφαρμογές νέων τεχνολογιών και απαιτούν μεγάλα κεφάλαια επένδυσης. Επιλέγουμε οργανωτική δομή ανά έργο όταν η διεύθυνση συμμετέχει σε πολλά παρόμοια έργα ή αναπτύσσεται ένα καινούριο προϊόν. Τέλος, οργανωτική δομή πίνακα θα επιλέξουμε όταν πολλά έργα πρέπει να μοιράζονται ειδικούς σε κάποια τεχνολογία και το έργο απαιτεί τη συμμετοχή διαφόρων λειτουργικών περιοχών, αλλά όχι ειδικούς με πλήρη απασχόληση στο έργο.

ΚΕΦΑΛΑΙΟ 2 ΙΕΡΑΡΧΙΚΑ ΔΕΔΟΜΕΝΑ

Η ιεραρχική δομή των δεδομένων είναι μια από τις πιο κοινές δομές δεδομένων στον προγραμματισμό. Αντιπροσωπεύει ένα σύνολο δεδομένων όπου κάθε στοιχείο (εκτός της ρίζας) έχει ένα κόμβο γονέα και ένα ή περισσότερα παιδιά κόμβους. Προγραμματιστές Web χρησιμοποιούν ιεραρχικά δεδομένα για μια σειρά εφαρμογών, συμπεριλαμβανομένων των συστημάτων διαχείρισης περιεχομένου, θέματα του φόρουμ, λίστες, και στο ηλεκτρονικό εμπόριο για κατηγοριοποίηση του προϊόντος.

Τα δύο πιο γνωστά ιεραρχικά μοντέλα είναι το ΜΟΝΤΕΛΟ ΛΙΣΤΑΣ ΓΕΙΤΝΙΑΣΗΣ (ADJACENCY LIST MODEL) και το ΜΟΝΤΕΛΟ ΕΜΦΩΛΕΥΜΕΝΟΥ ΣΥΝΟΛΟΥ (NESTED SET MODEL)

2.1 ΜΟΝΤΕΛΟ ΛΙΣΤΑΣ ΓΕΙΤΝΙΑΣΗΣ(ADJACENCY LIST MODEL)

Το μοντέλο λίστας γειτνίασης είναι ένα ιεραρχικό μοντέλο δεδομένων, όπου κάθε κόμβος έχει ένα δείκτη προς τον γονέα του (στοιχείο ρίζας, έχει ένα δείκτη NULL). Ο πίνακας SQL θα μοιάζει με αυτό:

```
1 CREATE TABLE category (  
2     category_id INT(10) AUTO_INCREMENT PRIMARY KEY,  
3     category_name VARCHAR(50) NOT NULL,  
4     parent_id INT(10) DEFAULT NULL  
5 );
```

Το parent_id είναι στην πραγματικότητα μια κατηγορία id της μητρικής κατηγορίας. Αυτή η μέθοδος είναι αρκετά απλή. Είναι εύκολο να δούμε σχέσεις γονιού-παιδιού. Επιπλέον, η διαδρομή δένδρου μπορεί να παραχθεί χρησιμοποιώντας μία απλή PHP function όπως φαίνεται και στο παρακάτω παράδειγμα

```
<?php  
function get_path($category_id)  
1 {  
2     // look up the parent of this node  
3     $result = mysql_query("SELECT c1.parent_id,c2.category_name AS  
4     parent_name FROM category AS c1  
5     LEFT JOIN category AS c2 ON c1.parent_id=c2.category_id  
6     WHERE c1.category_id='$category_id' ");  
7     $row = mysql_fetch_array($result);  
8  
9  
10    // save the path in this array  
11    $path = array();  
12  
13    //continue if this node is not the root node  
14    if ($row['parent_id']!=NULL)  
15    {  
16        // the last part of the path to node  
17  
18        end($path);  
19        $last_key = key($path);  
20        $key = $last_key==0 ? 0 : $last_key+1;  
21  
22        $path[$key]['category_id'] = $row['parent_id'];  
23        $path[$key]['category_name'] = $row['parent_name'];  
24  
25        $path = array_merge(get_path($row['parent_id']), $path);  
26    }  
27  
28    return $path;  
29 }  
30 ?>
```

Για να εμφανίσουμε τη διαδρομή στην οθόνη εκτελούμε την παρακάτω εντολή:

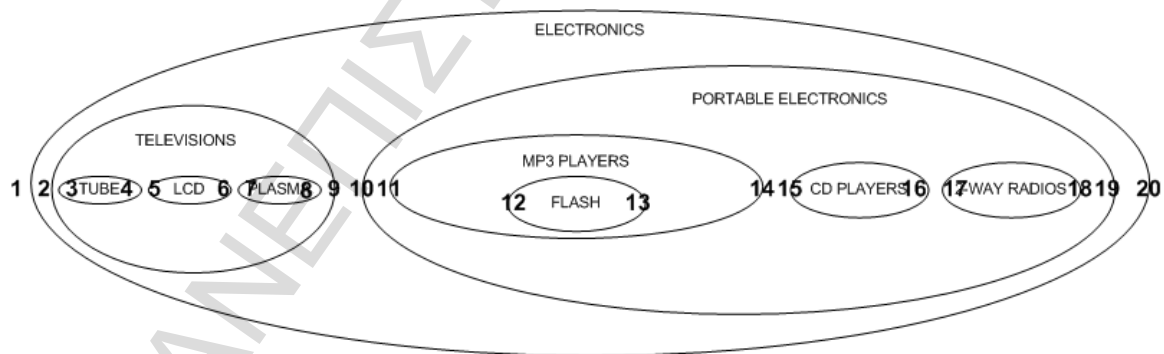
```
1 <?php
2 for ($i=count($path)-1;$i==0;$i--)
3 {
4     echo $path[$i]['category_name']. '>';
5 }
6 ?>
```

Ωστόσο, το μοντέλο λίστας γειννίαςσης έχει τα μειονεκτήματά του. Πρώτον, είναι δύσκολο εφαρμοστεί χρησιμοποιώντας μόνο ερωτήματα στη βάση δεδομένων. Απαιτούνται SQL ερωτήματα για να βρούμε σε ποιο επίπεδο βρίσκεται ο κόμβος. Επιπλέον, για κάθε επίπεδο του δέντρου υλοποιείται και ένα join ερώτημα, πράγμα που σημαίνει ότι για κάθε επίπεδο του δέντρου μειώνετε η απόδοση της βάσης δεδομένων.

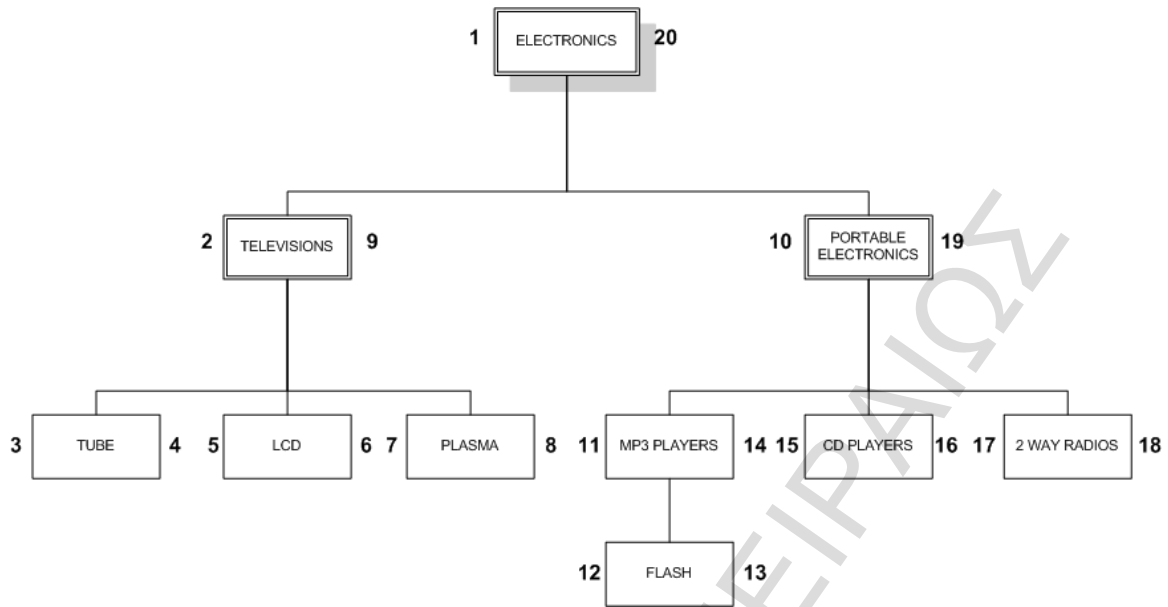
Η Διαγραφή κόμβων θα μπορούσε επίσης να προκαλέσει προβλήματα. Αν η διαγραμμαμένη κόμβος έχει παιδιά, θα να μείνουν “ορφανά” δηλαδή δεν θα έχουν κανένα γονικό στοιχείο που να τα συνδέουν και θα εξαφανιστούν από το δέντρο.

2.2 ΜΟΝΤΕΛΟ ΕΜΦΩΛΕΥΜΕΝΟΥ ΣΥΝΟΛΟΥ(NESTED SET MODEL)

Το ένθετο σύνολο μοντέλο, αποκαλούμενο επίσης modified preorder tree traversal, είναι μια άλλη προσέγγιση για το χειρισμό των ιεραρχικά δεδομένων. Αντί για σχέσεις γονιού-παιδιού, η ιεραρχία αναπαρίσταται ως ένα σύνολο με εμφωλευμένους κύκλους και την ανάθεση αριστερά και δεξιά αριθμητικών τιμών σε κάθε κόμβο όπως φαίνεται και στη παρακάτω εικόνα.



Το οποίο αναπαριστάται σε δενδρική μορφή ως:



Η διαδικασία του προσδιορισμού τιμών αριστερά και δεξιά γίνεται από αριστερά προς τα δεξιά, βάζοντας πρώτα τιμή στα αριστερά, και μετά προχωράμε εκχωρώντας τιμές στα παιδιά του κόμβου.

Δημιουργούμε τον πίνακα στη βάση:

```

1 CREATE TABLE category (
2     category_id INT(10) AUTO_INCREMENT PRIMARY KEY,
3     category_name VARCHAR(50) NOT NULL,
4     lft INT(10) NOT NULL,
5     rgt INT(10) NOT NULL
6 );
  
```

Για να εμφανίσουμε όλα τα δεδομένα χρησιμοποιούμε τη παρακάτω εντολή

```

1 SELECT * FROM category WHERE lft BETWEEN 1 AND 20 ORDER BY lft ASC
  
```

Το ίδιο ερώτημα χρησιμοποιείται για να επιλέξουμε ένα υποέργο περιορίζοντάς το στις τιμές αριστερά και δεξιά που το περιέχουν.

```
1 SELECT * FROM category WHERE lft BETWEEN 2 AND 7 ORDER BY lft ASC
```

Τοποθέτηση ενός κόμβου στην ιεραρχία απαιτεί την ενημέρωση του αριστερού και του δεξιού τιμές των κόμβων σε όλο το δέντρο.

```
1 UPDATE category SET rgt=rgt+2 WHERE rgt>6
2 UPDATE category SET lft=lft+2 WHERE lft>6
```

Τώρα δημιουργήσαμε αρκετό χώρο για τον νέο κόμβο, οπότε:

```
INSERT INTO category (category_name,lft,rgt) VALUES ('Sneakers', '7', '8')
```

Η διαγραφή ενός κόμβου στο μοντέλο εμφωλευμένου συνόλου είναι λίγο πιο περίπλοκη από ότι στο μοντέλο λίστας γεινίασης. Η διαδικασία είναι λίγο διαφορετική ανάλογα με το αν διαγράφουμε ένα κόμβο με παιδιά ή όχι. Αν ο κόμβος προς διαγραφή έχει από δεξιά άλλους κόμβους, μειώνουμε όλες τις αριστερές και δεξιές τιμές που είναι μεγαλύτερες από ότι η αριστερή και η δεξιά τιμή του κόμβου κατά δύο(2) και στη συνέχεια διαγράφουμε τον κόμβο αυτό.

```
1 UPDATE category SET lft=lft-2 WHERE lft>5
2 UPDATE category SET rgt=rgt-2 WHERE rgt>6
3 DELETE FROM category WHERE lft='5' AND rgt='6'
```

Αν και πιο περίπλοκο και πιο δύσκολο στη κατανόηση, η μέθοδος του εμφωλευμένου συνόλου έχει πολλά πλεονεκτήματα. Δε στηρίζεται πολύ στην αναδρομή, επίσης, τα ερωτήματα της βάσης δεδομένων είναι απλούστερα και τα περισσότερα από αυτά μπορούν να εκτελεστούν με ένα μόνο ερώτημα. Αυτά τα χαρακτηριστικά μπορούν να αυξήσουν σημαντικά την απόδοση της εφαρμογής και ακόμα μπορεί να χειριστεί πολύπλοκα ιεραρχικές δομές με αποδεκτή ταχύτητα.

Από την άλλη πλευρά, η ενημέρωση της ιεραρχίας (προσθήκη ή διαγραφή κόμβων) είναι πιο πολύπλοκη και μπορεί να είναι πιο αργή.

2.3 ΣΥΓΚΡΙΣΗ ΤΩΝ 2 ΜΟΝΤΕΛΩΝ

Η επιλογή του κατάλληλου μοντέλου για την εκάστοτε περίπτωση εξαρτάται από το αν χρειαζόμαστε ένα πιο ευέλικτο μοντέλο που μπορεί εύκολα να ενημερωθεί. Σε αυτή την περίπτωση, κατάλληλο είναι το μοντέλο λίστας γεινίασης. Διαφορετικά, αν έχουμε ένα

πολύπλοκο δέντρο που δεν χρειάζεται συχνή ενημέρωση των κόμβων, επιλέγουμε το μοντέλο εμφωλευμένου συνόλου.

Στη συγκεκριμένη εργασία που δεν έχουμε όριο στο πόσο πολύπλοκο και πόσο μεγάλο θα είναι το έργο χρησιμοποιήσαμε το μοντέλο εμφωλευμένου συνόλου.

ΚΕΦΑΛΑΙΟ 3 ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

3.1 APACHE WEB SERVER

Ο Apache web server είναι ένας server του παγκοσμίου ιστού. Ο Apache δημιουργήθηκε από τον Robert McCool, ως ένα project του National Center for Supercomputing Application (NCSA) με όνομα HTTPd (HTTP daemon). Το 1994 ο McCool αποχώρησε από το NCSA και έτσι το NCSA

HTTPd έμεινε εγκαταλελειμμένο. Το 1995 το ανέλαβε το ίδρυμα λογισμικού Apache (Apache Software Foundation) , το οποίο διατηρεί την εποπτεία του μέχρι και σήμερα.

Σήμερα ο Apache είναι ένα πρόγραμμα ανοιχτού κώδικα, δηλαδή διατίθεται δωρεάν και μπορούν να γίνουν ελεύθερα προσθήκες και τροποποιήσεις στον κώδικα του. Το ίδιο το πρόγραμμα δεν έχει και τόσες πολλές δυνατότητες, όμως πάνω του μπορούν να προσαρμοστούν διαφορετικές προσθήκες προγραμμάτων(modules), οι οποίες του παρέχουν πολλές λειτουργίες πχ. Mod_access, mod_digest, mod_auth κλπ. Επιπλέον, βασικό πλεονέκτημά του είναι ότι μπορεί να εγκατασταθεί σε διαφορετικά λειτουργικά και να υποστηρίξει διάσημες γλώσσες προγραμματισμού όπως Php, Perl, Python κλπ.

3.2 MYSQL

Η Mysqι δημιουργήθηκε από την σουηδική εταιρία Mysqι AE η οποία σήμερα ανήκει στην Oracle. Είναι το πιο διαδεδομένο ανοιχτού κώδικα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων (RDBMS), το οποίο χρησιμοποιεί την SQL, για την προσθήκη, πρόσβαση, και την επεξεργασία δεδομένων.

Λόγω του ότι είναι ανοιχτού κώδικα, μπορεί ο κάθε χρήστης να εκτελεί, αλλά ακόμα και να τροποποιεί την εφαρμογή. Ωστόσο υπάρχουν κάποιες περιπτώσεις όπου ο χρήστης πρέπει να πάρει άδεια από την MySQL έτσι ώστε να υλοποιήσει συγκεκριμένες ενέργειες.

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από συστηματικά οργανωμένα σχετιζόμενα δεδομένα , είτε πρόκειται για κείμενο, αριθμούς, ή δυαδικά αρχεία. Αυτό που πετυχαίνουμε μέσω ενός ΣΔΣΒΔ είναι η αποθήκευση, χρήση, συντήρηση, οργάνωση του μεγάλου όγκου πληροφοριών που είναι αποθηκευμένα στην βάση δεδομένων. Το ΣΔΣΒΔ είναι ουσιαστικά ένα εργαλείο αλληλεπίδρασης με την βάση δεδομένων.

Το λογισμικό της MySQL αποτελείται από αρκετά μέρη, στα οποία περιλαμβάνεται ο διακομιστής MySQL (mysqld, ο οποίος εκτελεί και διαχειρίζεται τις βάσεις δεδομένων), ο πελάτης MySQL (mysql, ο οποίος παρέχει μια διασύνδεση για το διακομιστή), και τα πολλά βοηθητικά προγράμματα για συντήρηση και άλλους σκοπούς.

Η γλώσσες μέσω των οποίων μπορεί να γίνει η διασύνδεση είναι η Php, Perl, Java, C#, C, Ruby, C++ και άλλες.

Η MySQL έχει δημιουργηθεί σε C και C++, και λειτουργεί εξίσου καλά σε αρκετά διαφορετικά λειτουργικά συστήματα. Για την MySQL είναι γνωστό ότι χειρίζεται βάσεις δεδομένων τόσο μεγάλες που μπορούν να περιέχουν 60000 πίνακες με περισσότερες από πέντε δισεκατομμύρια εγγραφές. Η Mysqι μπορεί αν λειτουργήσει με πίνακες τόσο μεγάλους μέχρι και οκτώ εκατομμύρια terabyte σε μερικά λειτουργικά συστήματα, ενώ σε άλλα 4 GB.

3.3 PHP

3.3.1 Ιστορική Αναδρομή

Η πολύ δημοφιλής γλώσσα συγγραφής σεναρίων (scripting language) Php, όπως την γνωρίζουμε σήμερα, γράφτηκε στην αρχική της μορφή από τον φοιτητή Rasmus Lerdorf . Ο λόγος που τον οδήγησε σε αυτή του την επινόηση ήταν η επιθυμία του να παρακολουθεί με κάποιο τρόπο τους χρήστες που διάβαζαν online το βιογραφικό του σημείωμα.

Αρχικά, η πρώτη εκείνη έκδοση της Php, μια συσκευασία (wrapper) των perl/CGI εμφάνιζε προβλήματα καθυστέρησης στην προσπέλαση. Για να λύσει αυτό το πρόβλημα ο Rasmus ξαναέγραψε την επικάλυψη της Perl σε C.

Αυτό το πρόγραμμα επικάλυψης του Rasmus είχε μεγάλη απήχηση στα άτομα που βρισκόταν στον ίδιο web server μαζί του, με αποτέλεσμα να του ζητάνε να το χρησιμοποιήσουν και αυτοί.

Σταδιακά η ζήτηση για το συγκεκριμένο πρόγραμμα συσκευασίας του Rasmus αυξανόταν, έτσι ο Rasmus αποφάσισε να φτιάξει μια πλήρη διανομή, μαζί με τεκμηρίωση, και ένα Αρχείο Συχνών Ερωτήσεων (FAQ). Η πρώτη αυτή διανομή ονομάστηκε Personal Home Page Tools (εργαλεία προσωπικής αρχικής σελίδας), που αργότερα μετονομάστηκε σε Personal Home Page Construction Kit (πακέτο κατασκευής προσωπικής σελίδας).

Παράλληλα με τις δραστηριότητες που αναφέρθηκαν παραπάνω, ο Rasmus εργαζόταν σε ένα εργαλείο για εύκολη ενσωμάτωση ερωτημάτων SQL μέσα σε μία ιστοσελίδα. Ουσιαστικά αυτό το εργαλείο ήταν άλλο ένα πρόγραμμα συσκευασίας CGI που ανέλυε ερωτήματα SQL και διευκόλυνε την δημιουργία φορμών και πινάκων με βάση τα ερωτήματα. Το όνομα του εργαλείου αυτού ήταν FI (form interpreter-διερμηνέας φορμών)

Η συνένωση αυτών των δύο είχε σαν αποτέλεσμα την Php/FI 2.0. Η Php/FI ήταν μια εύκολη γλώσσα προγραμματισμού και όχι ένα απλό πρόγραμμα επικάλυψης του CGI. Η γλώσσα αυτή επέτρεπε σε προγραμματιστές να ενσωματώσουν απλά script κατευθείαν μέσα σε μια ιστοσελίδα.

Η Php/FI άρχισε να χρησιμοποιείται ευρέως από τους προγραμματιστές, αυτό έκανε των Andi Gutmans και τον Zeev Suraski να την αναδιαμορφώσουν με αποτέλεσμα να φτιάξουν την Php3.

Η Php 3 ξεπέρασε τα προβλήματα που αντιμετώπιζε Php/FI. Πιο συγκεκριμένα είχε μεγαλύτερη απόδοση, σαφέστερη σύνταξη, υποστήριξη αντικειμενοστραφών χαρακτηριστικών, το βρόχο for και πολλές άλλες βελτιώσεις.

Αν και ιδιαίτερα δημοφιλής και αρκετά γρήγορη η Php3 ξαναγράφτηκε από τους Andi και Zeev, για να δημιουργηθεί η ακόμα πιο βελτιωμένη Php 4. Παράλληλα με αυτήν φτιάχτηκε και το framework Zend (από τα αρχικά των ονομάτων τους). Μέσω αυτού του framework η Php 4 προσφέρει πολλά νέα χαρακτηριστικά, και επεκτείνει την λειτουργικότητα της Php 4, η οποία πλέον μπορεί να καλύψει τις ανάγκες περίπλοκων εφαρμογών στο παγκόσμιο ιστό.

3.3.2 Η Λειτουργικότητα της Php

Η Php επικεντρώνεται κυρίως στο server-side scripting. Ουσιαστικά μπορεί να κάνει ότι και οποιοδήποτε άλλο CGI πρόγραμμα δηλαδή να προσπελάσει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, να στείλει και να πάρει cookies. Εκτός από αυτές τις βασικές λειτουργίες όμως η Php μπορεί να κάνει πολλά περισσότερα.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα Php script.

- **Server - side scripting.** Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την Php. Χρειάζονται τρία πράγματα για να λειτουργήσει η Php με αυτόν τον τρόπο. Τον Php μεταγλωττιστή (CGI ή server module), ένα web server (εξυπηρετητή σελίδων) και ένα web browser (φυλλομετρητή).

- **Command line scripting.** Σε αυτή την μορφή υπάρχει δυνατότητα δημιουργίας ενός Php script το οποίο τρέχει χωρίς server ή browser. Χρειάζεται μόνο ένας Php μεταγλωττιστής. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται συχνά με τη χρήση της cron (σε *nix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου.

- **Εγγραφή client-side GUI εφαρμογών** (Γραφικά περιβάλλοντα χρηστών). Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυριακές εφαρμογές.

Μια κύρια αιτία για την μεγάλη διάδοση της PHP είναι η συμβατότητα. Ότι δηλαδή μπορεί να χρησιμοποιηθεί στα πιο γνωστά λειτουργικά συστήματα, να υποστηρίζει πολλούς servers, αλλά και η υποστήριξη που έχει σε ένα μεγάλο σύνολο βάσεων δεδομένων.

Τα λειτουργικά συστήματα στα οποία μπορεί να χρησιμοποιηθεί η PHP είναι τα εξής: Linux, πολλές εκδοχές του Unix (HP-UX, Solaris, και OpenBSD), Microsoft Windows, Mac OS X, RISC OS κ.α.

Όσον αφορά τους Server που υποστηρίζει η PHP έχουμε και εδώ έναν καθόλου ευκαταφρόνητο αριθμό. Συγκεκριμένα οι Servers που υποστηρίζονται από την PHP είναι: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, Oreilly Website Pro server, Caudium, Hitami, OmniHTTPd, και πολλούς άλλους.

Επίσης η Php είναι μέχρι στιγμής συμβατή και με τις εξής βάσεις δεδομένων: Adabas D, dBase, Empress, FilePro (read-only), Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase, mSQL, Direct MS-SQL, MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL, Solid, Sybase, Velocis, Unix dbm.

Επιπρόσθετα υπάρχει μια αφαιρετική επέκταση DBX βάσεων δεδομένων (DBX database abstraction extension) μέσω της οποίας μπορούμε να χρησιμοποιούμε με την ίδια διασύνδεση (transparently) οποιαδήποτε βάση δεδομένων υποστηρίζεται από αυτή την επέκταση. Εκτός αυτού η PHP υποστηρίζει το ODBC, το Open Database Connectivity standard (Ανοιχτό πρότυπο Σύνδεσης Βάσεων δεδομένων) και έτσι μπορούμε να συνδεθούμε σε οποιαδήποτε βάση δεδομένων που υποστηρίζει αυτό το πρότυπο.

Ως εκ τούτου, χρησιμοποιώντας την PHP, υπάρχει μεγάλη ποικιλία στην επιλογή λειτουργικού συστήματος, webserver αλλά και βάσης δεδομένων. Άλλη μία επιλογή που υπάρχει κατά την χρήση της Php είναι ο προγραμματισμός σε διαδικασιακή (procedural) ή αντικειμενοστραφή (object oriented) μορφή.

Με την Php μπορούμε επίσης να εξάγουμε εικόνες, αρχεία PDF, ταινίες Flash, αλλά και να εξάγουμε κείμενο όπως XHTML και οποιοδήποτε άλλο XML αρχείο. Τα αρχεία αυτά δημιουργούνται αυτόματα και αποθηκεύονται παίζοντας το ρόλο μια server side cache.

Η PHP έχει επίσης υποστήριξη για επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και άλλα. Μπορούμε επίσης να ανοίξουμε υποδοχές δικτύου (network sockets) και να αλληλεπιδράσουμε με οποιοδήποτε άλλο πρωτόκολλο. Η PHP έχει ακόμη υποστήριξη για την περίπλοκη ανταλλαγή δεδομένων WDDX μεταξύ σχεδόν όλων των Web programming γλωσσών. Ακόμα, η PHP υποστηρίζει δημιουργία instantiation αντικειμένων Java και τα χρησιμοποιεί διάφανα σαν αντικείμενα PHP. Μπορεί επίσης να χρησιμοποιηθεί η CORBA(Common Object Request Broker Architecture) επέκταση για να προσπελαθούν απομακρυσμένα (remote) αντικείμενα.

Η PHP έχει εξαιρετικά χρήσιμα χαρακτηριστικά επεξεργασίας κειμένων, από την POSIX επέκταση ή τις Perl regular expressions μέχρι XML parsing αρχείων. Για τη μεταγλώττιση και την πρόσβαση αρχείων XML, υποστηρίζει τα πρότυπα SAX και DOM. Μπορούμε επίσης να χρησιμοποιήσουμε την XSLT επέκταση μας για να μετατρέψουμε τα XML αρχεία σε άλλες μορφές.

Στον τομέα ηλεκτρονικού εμπορίου, έχουμε τις συναρτήσεις: Cybercash payment, CyberMUT, VeriSign Payflow Pro και C CVS, οι οποίες είναι χρήσιμες για τα online προγράμματα πληρωμής.

Τέλος, υπάρχουν πολλές άλλες ενδιαφέρουσες επεκτάσεις, τις mhoGoSearch search engine συναρτήσεις, πολλά εργαλεία συμπίεσης (gzip, bz2), μετατροπές ημερολογίου και μεταφράσεις.

Οι δυνατότητες της Php είναι πάρα πολλές, παραπάνω αναφέρθηκαν οι πιο σημαντικές από αυτές.

3.3.3 Τα πλεονεκτήματα της Php

Ο λόγος που η Php είναι ίσως η πιο διαδεδομένη ανάμεσα στις scripting γλώσσες είναι τα πλεονεκτήματα που έχει έναντι στις υπόλοιπες γλώσσες ίδιου τύπου τα οποία αναφέρονται παρακάτω:

- Υψηλή απόδοση
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
- Χαμηλό κόστος
- Ευκολία μάθησης και χρήσης
- Μεταφερσιμότητα
- Διαθεσιμότητα του κώδικα προέλευσης.

3.4 GD LIBRARY(ΒΙΒΛΙΟΘΗΚΗ ΓΡΑΦΙΚΩΝ)

Η βιβλιοθήκη GD Graphics είναι μια βιβλιοθήκη γραφικών λογισμικού από τον Thomas Boutell για το δυναμικό χειρισμό εικόνων . Η GD είναι γραμμένη σε ANSI C , αλλά έχει διεπαφές για πολλές άλλες γλώσσες προγραμματισμού. Μπορεί να δημιουργήσει αρχεία GIF , JPEG, PNGs και WBMPs .

Η GD μπορεί να δημιουργήσει τις εικόνες που αποτελούνται από γραμμές, τόξα , κείμενο, άλλες εικόνες , και πολλά χρώματα . Η έκδοση 2.0 προσθέτει υποστήριξη για τις εικόνες truecolor , κανάλια άλφα , resampling (για την ομαλή προσαρμογή των εικόνων truecolor) , και πολλά άλλα χαρακτηριστικά .

Η GD υποστηρίζει πολλές γλώσσες προγραμματισμού , συμπεριλαμβανομένων C , PHP , Perl , Python , OCaml , Tcl , Lua , Pascal , το GNU Octave , REXX , Ruby and Go.

Η GD χρησιμοποιείται εκτενώς με την PHP. Άλλωστε μια τροποποιημένη έκδοση που υποστηρίζουν πρόσθετα χαρακτηριστικά περιλαμβάνεται από την PHP 4.3 .

3.5 HTML

Η HTML (ακρωνύμιο του αγγλικού HyperText Markup Language, - Γλώσσα Σήμανσης Υπερκειμένου), είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες. Ουσιαστικά πρόκειται για μια γλώσσα, μέσω της οποίας παρέχεται η δυνατότητα μορφοποίησης και εμφάνισης των στοιχείων μιας ιστοσελίδας.

Η ιστορία της HTML ξεκινά στα τέλη της δεκαετίας του '80 από τον φυσικό Τιμ μπέρνερς Λι, ο οποίος πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο με βάση τα υπερκείμενα. Έτσι έφτιαξε την προδιαγραφή της HTML. Αυτή η αρχική επινόηση του Λι, εξελίχθηκε συν τω χρόνω, και το 1995

δημοσιεύτηκε η επίσημη έκδοση HTML 2.0. Από τότε και μέχρι σήμερα έχουν κάνει την εμφάνιση τους αρκετές εκδόσεις της HTML, οι οποίες έχουν ολοένα και περισσότερα πρόσθετα χαρακτηριστικά με πιο εξελιγμένη έκδοση την HTML 5, η οποία δημοσιεύτηκε το 2008.

Μέσω της HTML μπορούμε να κατασκευάσουμε στατικά website. Ωστόσο ακόμα και τα δυναμικά, εκτός από τις υπόλοιπες τεχνολογίες χρησιμοποιούν σχεδόν πάντα και HTML. Η HTML υποστηρίζει την εισαγωγή εικόνων, συνδέσμων, πινάκων και άλλων αντικειμένων στις ιστοσελίδες. Παράλληλα, μπορεί να χρησιμοποιηθεί για την εμφάνιση διαδραστικών φορμών, όπως και να καθορίσει δομικά στοιχεία για το κείμενο, τις κεφαλίδες, παραγράφους, λίστες συνδέσμων κ.α.

Η HTML αποτελείται από ετικέτες (tags) τα οποία έχουν την εξής γενική μορφή: `<tag attribute1="value1" attribute2="value2">content</tag>`. Σε αυτή την περίπτωση το όνομα της ετικέτας τερματισμού ξεκινά με μια κάθετο «/». Υπάρχει μια ακόμα γενική κατηγορία τα επονομαζόμενα άδεια στοιχεία τα οποία έχουν την εξής μορφή `<tag attribute1="value1" attribute2="value2" >` και δεν έχουν καθόλου περιεχόμενο.

3.6 CSS

Τα CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) ή (αλληλουχία φύλλων στυλ), είναι ένα εργαλείο για περαιτέρω μορφοποίηση των στοιχείων μιας ιστοσελίδας, η οποία έχει γραφεί σε μια γλώσσα σήμανσης (HTML, XHTML).

Μέσω των CSS μπορούμε να αλλάξουμε την εμφάνιση, αλλά και την διάταξη των ιστοσελίδων μας. Πιο συγκεκριμένα τα CSS είναι μια γλώσσα διακόσμησης, η οποία δίνει την μεγαλύτερες δυνατότητες από την HTML πάνω στις γραμματοσειρές (fonts), χρώματα (colours), περιθώρια (margins), εικόνες παρασκήνιου (background images) και πολλά άλλα. Επίσης μπορούμε να ορίσουμε την τοποθέτηση των στοιχείων το browser.

Αρκετά σημαντική είναι όμως και η δυνατότητα ομαδοποίησης κάποιων σημείων του εγγράφου (δημιουργία κλάσης), στα οποία θα εφαρμόσουμε μια συγκεκριμένη διακόσμηση. Επιπλέον έχουμε την δυνατότητα για την ταυτόχρονη μορφοποίηση πολλών εγγράφων, συνδέοντάς τα με ένα φύλο στυλ (style sheet).

Με την πάροδο του χρόνου, παρέχονται περισσότερες δυνατότητες διακόσμησης μέσω των καινούργιων εκδόσεων CSS, με αποτέλεσμα πλέον η ύπαρξη CSS να θεωρείται απαραίτητη για την δημιουργία ενός καλοσχεδιασμένου site.

3.7 ΣHEMA ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

3.7.1 ΣHEMA ΚΑΙ ΣΤΙΓΜΙΟΤΥΠΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Οι βάσεις δεδομένων μεταβάλλονται με την πάροδο του χρόνου καθώς η πληροφορία προστίθεται, αφαιρείται ή τροποποιείται.

Το σύνολο της πληροφορίας το οποίο βρίσκεται αποθηκευμένο σε μια ΒΔ σε μια συγκεκριμένη χρονική στιγμή λέγεται στιγμιότυπο (instance or snapshot) της ΒΔ.

Το σχήμα (schema) της ΒΔ είναι το σύνολο των εννοιών που περιγράφουν τη δομή της και είναι – σχεδόν πάντα – σταθερό.

Αναλογία : record types <-> schema, variables <-> instances

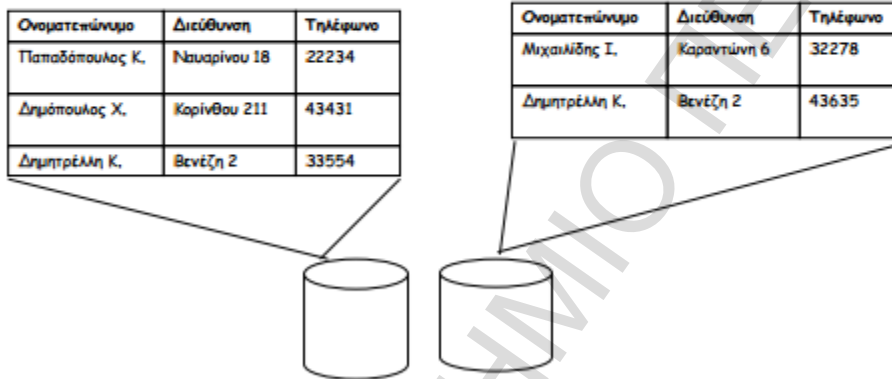
Μια ΒΔ έχει ένα (ή περισσότερα) σχήματα σε κάθε επίπεδο αφαίρεσης:

- Εσωτερικό σχήμα
- Εννοιολογικό σχήμα
- Εξωτερικό σχήμα (σχήματα όψεων)

Παράδειγμα

Όνοματεπώνυμο	Διεύθυνση	Τηλέφωνο
---------------	-----------	----------

Σχήμα βάσης δεδομένων (database schema)



Στιγμιότυπα βάσης δεδομένων (database instances)

3.7.2 ΣHEMA ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΕ

Πίνακας project

Project_id	Project_name	lft	rgt	Date_from	Date_to	days	depth
------------	--------------	-----	-----	-----------	---------	------	-------

Project_id: Το μοναδικό id που έχει κάθε νέα εγγραφή.

Project_name: Το όνομα του έργου ή υποέργου.

Lft: Ο αριθμός που ορίζει το αριστερό σημείο της θέσης του έργου ή υποέργου.

Rgt: : Ο αριθμός που ορίζει το δεξί σημείο της θέσης του έργου ή υποέργου.

Date_from: Η ημερομηνία εκκίνησης του έργου ή υποέργου.

Date_to: Η ημερομηνία λήξης του έργου ή υποέργου.

Days: Η διάρκεια του έργου ή υποέργου σε ημέρες.

Depth: Ένα αριθμητικό στοιχείο που μας βοηθάει να βρούμε τη θέση του έργου ή υποέργου στη δενδρική μορφή.

Πίνακας “admin”

Table_id	Alert_days	New_column
----------	------------	------------

Table_id: Το μοναδικό id που έχει κάθε νέα εγγραφή.

Alert_days: Πόσες ημέρες πριν την ολοκλήρωση του έργου να μας ενημερώνει.

New_column: Όλες οι καινούριες στήλες που έχει προσθέσει ο χρήστης.

3.8 ΔΟΜΗ ΙΣΤΟΣΕΛΙΔΑΣ

Ως δομή Web site ορίζουμε τον τρόπο με τον οποίο είναι συνδεδεμένες (παραπέμπουν η μία στην άλλη) οι Web σελίδες που το απαρτίζουν.

Η δομή αυτή μπορεί να είναι **ιεραρχική** (δηλ. όπως ένα δενδροειδές διάγραμμα με την αρχή του στην κορυφή), **γραμμική** (όπως αυτή των σελίδων ενός βιβλίου όπου έχουμε μια ακολουθία με αρχή την πρώτη σελίδα και τέλος την τελευταία), **μικτή** (που αποτελεί ένα συνδυασμό των δύο προηγούμενων) και **ακανόνιστη** (χωρίς κεντρική δομή και εμφανή αρχή και τέλος).

Η δομή της ιστοσελίδας που αναπτύχθηκε είναι η παρακάτω.

- Αρχική σελίδα
 - Εμφάνιση πληροφοριών ανά έργο
 - Προσθήκη διαγραφή έργων
 - Προσθήκη νέου έργου
 - Διαγραφή έργου
 - Επεξεργασία έργου
 - Δενδρική εμφάνιση έργων
 - Περισσότερες πληροφορίες ανα έργο
 - Προηδοποίηση σύντομης ολοκλήρωσης έργου
 - Περιοχή διαχείρισης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΚΕΦΑΛΑΙΟ 4 Η ΕΦΑΡΜΟΓΗ

4.1 ΕΙΣΑΓΩΓΗ ΕΡΓΩΝ

Επιλέγουμε από το μενού Add/Delete Projects και εμφανίζονται πίνακες με όλα τα έργα και τα υποέργα τους. Έχουμε 3 επιλογές Project, Delete και edit.

Για τη προσθήκη έργου πάμε στο πεδίο Project και επιλέγουμε το επιθυμητό έργο όπου θέλουμε να του προσθέσουμε ένα υποέργο.



ΠΑΝΕΠΙΣΤΗΜΙΟΝ

Menu

- Home
- Add/Delete Project
- Tree View
- Graph

Project	Delete	Edit
New Project 1 52		edit

Project	Delete	Edit
Taiwan Project 2 43	delete	edit
GAMING CONSUMABLES 3 10	delete	edit
Approve Samples 4 5	delete	edit
Produce Samples 6 7	delete	edit
8 9 Prepare and Approve Consumables	delete	edit
Games DATA MIGRATION 11 26	delete	edit
Perform Switchover 12 13	delete	edit
Perform Data Migration 14 19	delete	edit
Perform COMS Update 15 16	delete	edit
Perform Backup Restore 17 18	delete	edit
Review Data Migration Plan 20 25	delete	edit
22 Review and Confirm Roles 21	delete	edit
Review and Confirm Data 23 24	delete	edit

Στη συνέχεια εισάγουμε το όνομα έργου, την ημερομηνία εκκίνησης και την ημερομηνία τέλους.



Εφόσον τα στοιχεία εισαχθούν στη βάση με επιτυχία το μήνυμα που παίρνουμε είναι το παρακάτω:

Data has been inserted succesfully

[Return](#)

Σε περίπτωση που οι ημερομηνίες που έχουμε συμπληρώσει για το υποέργο που φτιάχνουμε είναι έξω από το εύρος ημερομηνιών του έργου τότε εμφανίζεται το ακόλουθο μήνυμα:

Dates has to be between 2013-11-06 and 2013-11-21

[Return](#)

4.2 ΕΠΕΞΕΡΓΑΣΙΑ ΕΡΓΩΝ

Εάν επιθυμούμε αλλαγή του ονόματος ή αλλαγή της αρχής ή του τέλους του έργου επιλέγουμε edit.



ΠΑΝΕΠΙΣΤΗΜΙΟΝ

Menu

- Home
- Add/Delete Project
- Tree View
- Graph

Project	Delete	Edit
New Project 1 52		edit

Project	Delete	Edit
Taiwan Project 2 43	delete	edit
GAMING CONSUMABLES 3 10	delete	edit
Approve Samples 4 5	delete	edit
Produce Samples 6 7	delete	edit
8 9 Prepare and Approve Consumables	delete	edit
Games DATA MIGRATION 11 26	delete	edit
Perform Switchover 12 13	delete	edit
Perform Data Migration 14 19	delete	edit
Perform COMS Update 15 16	delete	edit
Perform Backup Restore 17 18	delete	edit
Review Data Migration Plan 20 25	delete	edit
22 Review and Confirm Roles 21	delete	edit
Review and Confirm Data 23 24	delete	edit
DATA CENTER IT AND TELECOMS 27 38	delete	edit
Establish Primary Data Center 28 35	delete	edit
30 30 Deliver and Check Data Center	delete	edit

Εμφανίζονται στη συνέχεια τα δεδομένα που είχαμε, και έχουμε τη δυνατότητα να τα αλλάξουμε.



GAMING CONSUMABLES2013-12-252013-12-30

Name:

Start Date:

End Date:

[Return](#)

Και αφού πατήσουμε το πλήκτρο SEND εμφανίζεται το μήνυμα που βεβαιώνει ότι τα δεδομένα μας έγιναν update με επιτυχία.

Data has been updated succesfully

[Return](#)

4.3 ΔΙΑΓΡΑΦΗ ΕΡΓΩΝ

Στη περίπτωση που επιθυμούμε τη διαγραφή ενός έργου ,επιλέγουμε DELETE που βρίσκεται δίπλα από το έργο προς διαγραφή.



Menu

- Home
- Add/Delete Project
- Tree View
- Graph

Project	Delete	Edit
New Project 1 52		edit

Project	Delete	Edit
Taiwan Project 2 43	delete	edit
GAMING CONSUMABLES 3 10	delete	edit
Approve Samples 4 5	delete	edit
Produce Samples 6 7	delete	edit
Prepare and Approve Consumables	delete	edit
8 9		
Games DATA MIGRATION 11 26	delete	edit
Perform Switchover 12 13	delete	edit
Perform Data Migration 14 19	delete	edit
Perform COMS Update 15 16	delete	edit
Perform Backup Restore 17 18	delete	edit
Review Data Migration Plan 20 25	delete	edit
22		
Review and Confirm Roles 21	delete	edit
Review and Confirm Data 23 24	delete	edit
DATA CENTER IT AND TELECOMS 27 38	delete	edit
Establish Primary Data Center 28 35	delete	edit
29 30		
Deliver and Check Data Center	delete	edit

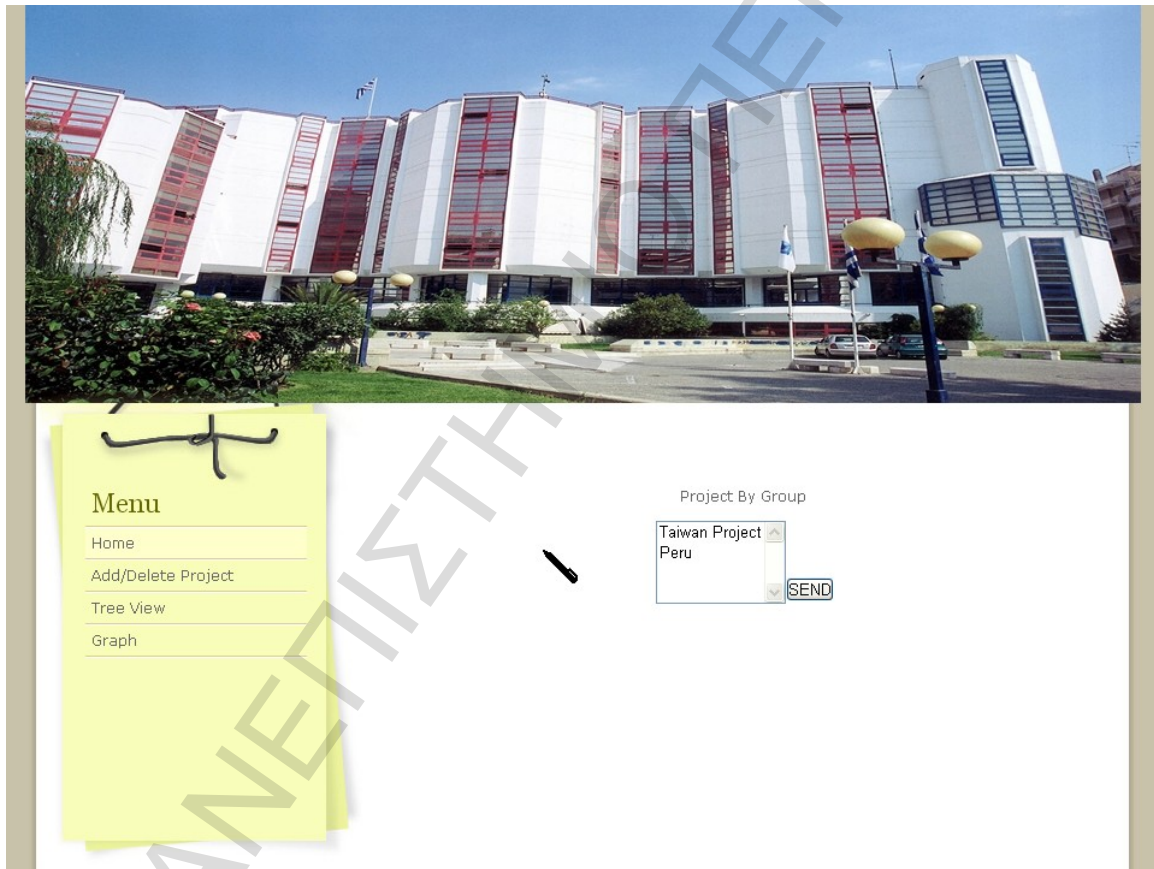
Εάν η διαγραφή είναι επιτυχής τότε εμφανίζεται το παρακάτω μήνυμα:

Data has been deleted succesfully

[Return](#)

4.4 ΕΜΦΑΝΗΣΗ ΓΡΑΦΗΜΑΤΟΣ ΑΝΑ ΕΡΓΟ

Για να δούμε δεντρική μορφή για ένα έργο πηγαίνουμε στην αρχική οθόνη (HOME) και επιλέγουμε ένα από τα έργα που εμφανίζονται στη φόρμα όπως φαίνεται παρακάτω.



Για να δούμε όλα τα έργα και περισσότερες πληροφορίες για αυτά επιλέγουμε “tree view” και στη συνέχεια επιλέγουμε “Details”



Menu

- Home
- Add/Delete Project
- Tree View
- Alert
- Admin

Project	From Date	To Date	More Details
MALAYSIA	2013-12-06	2014-01-08	Details

Project	From Date	To Date	More Details	
Taiwan Project	2013-12-01	2013-12-30	Details	
GAMING CONSUMABLES	2013-12-25	2013-12-30	Details	
Approve Samples	2013-12-30	2013-12-30	Details	
Produce Samples	2013-12-27	2013-12-29	Details	
Prepare and Approve Consumables	2013-12-25	2013-12-27	Details	
Games DATA MIGRATION	2013-12-15	2013-12-25	Details	
Perform Switchover	2013-12-22	2013-12-25	Details	
Migration	Perform Data	2013-12-18	2013-12-22	Details
Update	Perform COMS	2013-12-20	2013-12-22	Details
Restore	Perform Backup	2013-12-18	2013-12-20	Details
	Review Data	2013-	2013-	Details

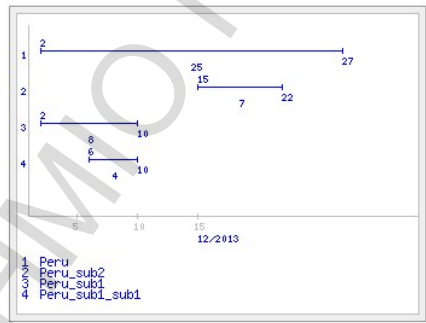
Αφου επιλέξουμε το έργο, εμφανίζεται στην οθόνη η δενδρική μορφή του έργου και το διάγραμμα Gantt για αυτό το έργο.



Menu

- Home
- Add/Delete Project
- Tree View
- Graph

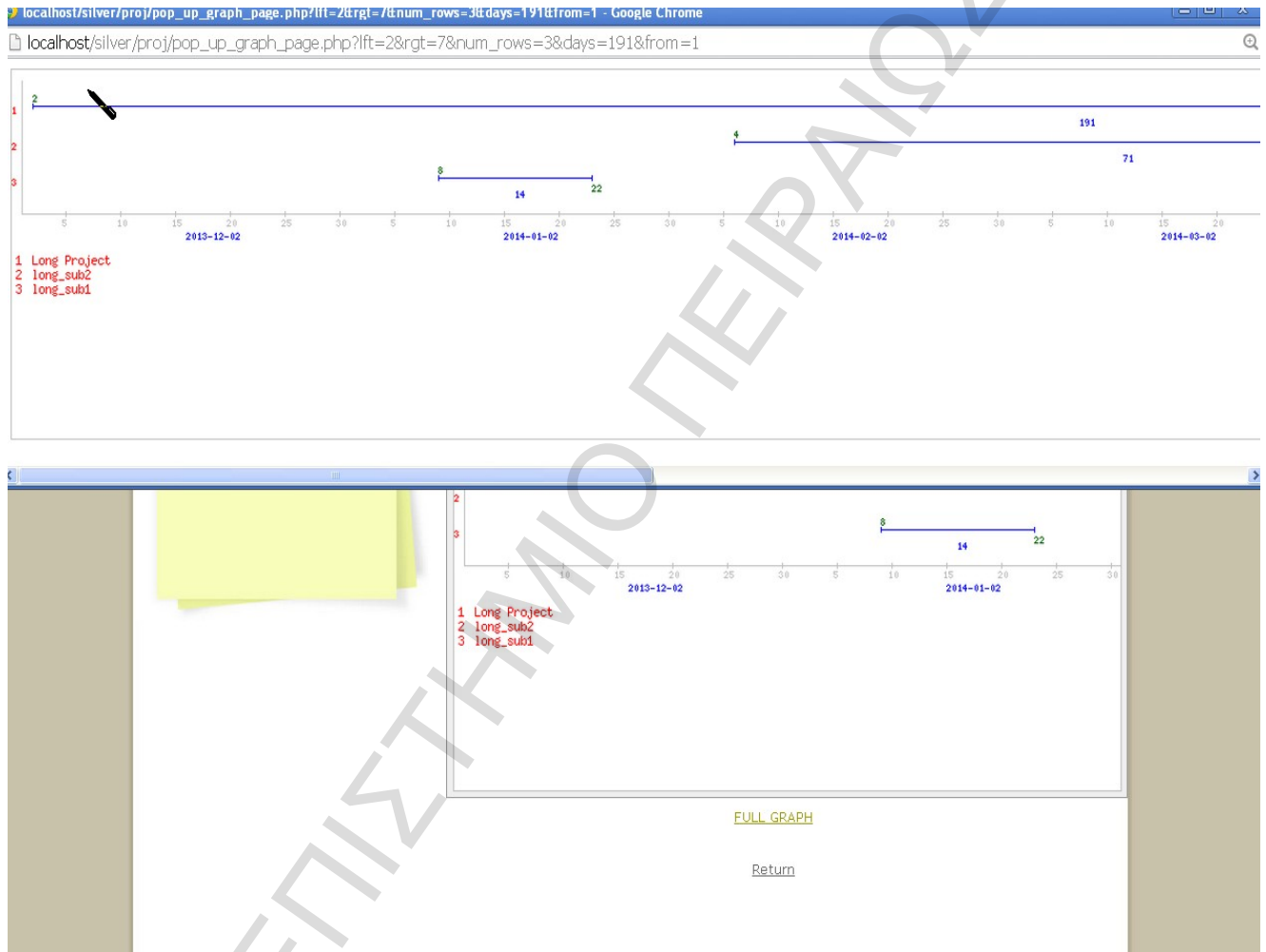
Project	From Date	To Date
Peru	2013-12-02	2013-12-27
Peru_sub2	2013-12-15	2013-12-22
Peru_sub1	2013-12-02	2013-12-10
Peru_sub1_sub1	2013-12-06	2013-12-10



[View](#)

[Return](#)

Στη περίπτωση που το έργο έχει μέγεθος μεγαλύτερο των 60 ημερών και προκειμένου να μην ξεπεράσουμε τα όρια της σελίδας έχουμε τη δυνατότητα να επιλέξουμε το link «Full Graph» που μας εμφανίζει ολόκληρο το γράφημα σε ξεχωριστή σελίδα όπως φαίνεται παρακάτω:



Υπάρχει ένα πεδίο στο μενού όπου ονομάζεται Alert και σκοπός του είναι να μας ενημερώνει για το όποια έργα ολοκληρώνονται σε λιγότερες από όσες μέρες θέλουμε από την τρέχουσα ημερομηνία.



Menu

- Home
- Add/Delete Project
- Tree View
- Alert
- Admin

Project	Main	From Date	To Date
Design of Data Center IT	Taiwan Project	2013-12-04	2013-12-06

[Return](#)

4.5 Περιοχή Διαχείρισης

Το πόσες μέρες πριν τη λήξη του έργου θα μας ενημερώνει η σελίδα alert εξαρτάται από το πόσες θα επιλέξουμε εμείς στο πεδίο “Days to Alert Before the end Of The Project” ,το οποίο βρίσκεται στο πεδίο μενού Admin.

Έχουμε τη δυνατότητα να επιλέξουμε από 0(καθόλου ενημέρωση) έως 999 ημέρες.

Ακόμα, το σύστημα υποστηρίζει να εισάγουμε επιπλέον στήλες. Αυτό σημαίνει ότι μπορούμε να προσθέσουμε πεδία με επιπλέον στοιχεία όπου θα εμφανίζονται σε κάθε έργο ή υποέργο που χρειαζόμαστε σύμφωνα με της δικές μας ανάγκες.

Για το σκοπό αυτό υπάρχει το πεδίο “add new column” στο οποίο δίνουμε το όνομα της στήλης που θέλουμε να προσθέσουμε.

Τέλος στο πεδίο Admin μπορούμε να διαγράψουμε μία στήλη που τη δημιουργήσαμε κατά λάθος ή δε χρειαζόμαστε πια. Αυτό βέβαια είναι επίφοβο μιας και με το που διαγραφεί η στήλη ό,τι δεδομένα υπήρχαν πάνω σε έργα σε αυτή τη στήλη ,θα διαγραφούν.



The screenshot displays a web application interface. On the left, there is a yellow sticky note with a hand-drawn sketch of a menu structure. The main content area is divided into three sections:

- Days To Alert Before The End Of The Project:** A text input field containing the number '3' and a 'SEND' button.
- Add new column:** A text input field and an 'ADD' button.
- Select Column To Delete:** A dropdown menu with 'Budget' selected and a 'DELETE' button.

A large, semi-transparent watermark reading 'ΠΑΝΕΠΙΣΤΗ' is overlaid diagonally across the interface.

Μόλις εισάγουμε το όνομα της στήλης που θέλουμε να προσθέσουμε και με τη προϋπόθεση ότι τα στοιχεία εισήχθησαν με επιτυχία εμφανίζεται το παρακάτω μήνυμα:

New column ' Budget ' has been inserted succesfully

[Return To Admin](#)

[Return To Main](#)

Μόλις εισάγουμε τον αριθμό των ημερών που επιθυμούμε εμφανίζεται το παρακάτω μήνυμα

Alert days has been changed to 3

[Return To Admin](#)

[Return To Main](#)

Μόλις επιλέξουμε το όνομα της στήλης που θέλουμε να διαγράψουμε και με τη προϋπόθεση ότι τα στοιχεία διαγράφησαν με επιτυχία εμφανίζεται το παρακάτω μήνυμα:

Column ' Budget ' has been deleted succesfully

[Return To Admin](#)

[Return To Main](#)

Όπως φαίνεται και στην παρακάτω εικόνα το νέο πεδίο που προσθέσαμε εμφανίζεται στην επιλογή "edit project" και μπορούμε να εισάγουμε στοιχεία.



Με την εισαγωγή μίας νέας στήλης και δεδομένων σε αυτή έχουμε τη δυνατότητα να τα δούμε επιλέγοντας “tree view” και στη συνέχεια “Details” όπως φαίνεται και στη παρακάτω εικόνα:

The screenshot displays a web application interface. At the top, a browser window shows the URL `panoscore.no-ip.biz/proj/details.php?lft=2&rft=3`. The browser content shows a table with the following data:

Project	From Date	To Date	Budget	new
MALAYSIA	2013-12-06	2014-01-08	2000	

Below the browser window, a sidebar on the left contains a menu with the following items:

- Add/Delete Project
- Tree View
- Alert
- Admin

The main content area shows a table with the following data:

Project	From Date	To Date	More Details
Taiwan Project	2013-12-01	2013-12-30	Details
GAMING CONSUMABLES	2013-12-25	2013-12-30	Details
Approve Samples	2013-12-30	2013-12-30	Details
Produce Samples	2013-12-27	2013-12-29	Details
Prepare and Approve Consumables	2013-12-25	2013-12-27	Details
Games DATA MIGRATION	2013-12-15	2013-12-25	Details
Perform Switchover	2013-12-22	2013-12-25	Details
Migration Perform Data	2013-12-18	2013-12-22	Details

ΚΕΦΑΛΑΙΟ 5 ΠΑΡΑΡΤΗΜΑ

5.1 ΠΕΡΙΓΡΑΦΗ ΤΜΗΜΑΤΩΝ ΤΟΥ ΚΩΔΙΚΑ

Η σελίδα `tree_view_by_project`

Σε αυτή τη σελίδα εμφανίζουμε το έργο που επέλεξε ο χρήστης και τα υποέργα του σε δενδρική μορφή, καθώς επίσης και το αντίστοιχο διάγραμμα Gantt.

Εδώ ζητάμε από τη σελίδα να φορτώσει τη σελίδα header η οποία περιέχει τα στοιχεία του template που χρησιμοποιούμε καθώς επίσης και τα στατικά κομμάτια της σελίδας στο πάνω μέρος της σελίδας.

```
<?php require("header.php"); ?>
```

Με τη παρακάτω εντολή ζητάμε να φορτώσει η σελίδα το όνομα χρήστη και το κωδικό που χρησιμοποιούμε για να συνδεθούμε στη βάση δεδομένων.

Σε περίπτωση δηλαδή που επιθυμούμε να αλλάξουμε αυτά τα στοιχεία δε χρειάζεται να τα αλλάξουμε σε όλες της σελίδες.

```
<?php require("mysql_conn.php"); ?>
```

```
<div id="content">
```

```
<center>
```

Δημιουργούμε τους τίτλους του πίνακα στον οποίο θα εισαχθούν τα στοιχεία

```
<table cellspacing="0" cellpadding="0">
<tr>
<th scope="col">Project</th>
<th scope="col">From Date</th>
<th scope="col">To Date</th>
</tr>
```

Σε αυτό το SQL ερώτημα παίρνουμε τα όρια του project για το οποίο θέλουμε να βρούμε όλα τα υποέργα του.

Στο αμέσως επόμενο SQL ερώτημα παίρνουμε όλα τα στοιχεία του έργου και των υποέργων που επέλεξε ο χρήστης.

```
<?php
mysql_select_db("proj",$conn);
$sql2 = "SELECT * FROM project WHERE project_name='$_POST[project_name]'
ORDER BY lft ASC";
```


Σε αυτό το κομμάτι του κώδικα εμφανίζεται η εικόνα με το διάγραμμα Gantt. Επίσης μπορούμε να επιλέξουμε το λίνκ που μας εμφανίζει το διάγραμμα σε νέα σελίδα χωρίς να έχει όριο μεγέθους εικόνας.

```
        echo "</td></tr>";
    }
    echo "</table>";
    echo " <img
src=\"barchart_per_project.php?project_id=$project_id&lft=$lft&rgt=$rgt&num_rows=$num_rows
&days=$days&from=0\" />";
?>

<br>

    <a href="javascript:void(0);"
NAME="Detailed Window" title=" Details "
onClick=window.open("pop_up_graph_page.php?lft=<?php echo "$lft";?>&rgt=<?php echo
"$rgt"; ?>&num_rows=<?php echo "$num_rows"; ?>&days=<?php echo "$days";
?>&from=1", "Ratting", "width=470,height=190,0,status=1,");>FULL GRAPH</a>

</center>
<br><br>
<center><a href = index.php>Return</a></center>
</div>

<?php require("footer.php"); ?>
```

Η σελίδα registered projects

Στη παρακάτω σελίδα παρουσιάζονται όλα τα έργα σε δενδρική μορφή και δίνουμε τη δυνατότητα στο χρήστη να προσθέσει ένα έργο, να διαγράψει ένα έργο ή να επεξεργαστεί ένα έργο αλλάζοντας οποιοδήποτε απο τα πεδία επιθυμεί.

```
<?php require("header.php"); ?>
<?php require("mysql_conn.php"); ?>
```

```
<div id="content">
```

<center>

```
<table cellpadding="0" cellspacing="0">
  <tr>
    <th scope="col">Project</th>
    <th scope="col">Delete</th>
    <th scope="col">Edit</th>
  </tr>
```

<?php

```
mysql_select_db("proj",$conn);

mysql_select_db("proj",$conn);

$sql2 = "SELECT rgt FROM project WHERE project_id=1";
$result2 = mysql_query($sql2,$conn) or die(mysql_error());

while($newArray = mysql_fetch_array($result2))
{
  $max_cycle=$newArray['rgt'];
}

$sql = "SELECT * FROM project WHERE lft BETWEEN 1 AND $max_cycle ORDER BY
lft ASC";
$result = mysql_query($sql,$conn) or die(mysql_error());

while($newArray = mysql_fetch_array($result))
{
  $project_id=$newArray['project_id'];
  $project_name=$newArray['project_name'];
  $lft=$newArray['lft'];
  $rgt=$newArray['rgt'];
```


Η σελίδα barchar_per_project

Αυτή η σελίδα έχει σαν σκοπό να δημιουργήσει μία εικόνα με το γράφημα Gantt. Αυτό επιτυγχάνεται με την GD η οποία είναι ενσωματωμένη στην έκδοση της php που χρησιμοποιούμε.

```
<?php
require("mysql_conn.php");
$num_rows=$_GET['num_rows'];
$max_days=$_GET['days'];

// Define .PNG image
header("Content-type: image/png");

if($_GET['from']==0)
$imgWidth=310+30*10;

else if($_GET['from']==1)
$imgWidth=310+$max_days*10;

$imgHeight=190+$num_rows*40;
// Create image and define colors
$image=imagecreate($imgWidth, $imgHeight);
$colorWhite=imagecolorallocate($image, 255, 255, 255);
$colorGrey=imagecolorallocate($image, 192, 192, 192);
$colorBlue=imagecolorallocate($image, 0, 0, 255);
$colorRed=imagecolorallocate($image, 255, 0, 0);
$colorGreen=imagecolorallocate($image, 0, 100, 0);

// Create border around image
imageline($image, 0, 0, 0, $imgHeight, $colorGrey);
imageline($image, 0, 0, $imgWidth, 0, $colorGrey);
imageline($image, $imgWidth-1, 0, $imgWidth-1, $imgHeight-1, $colorGrey);
imageline($image, 0, $imgHeight-1, $imgWidth, $imgHeight-1, $colorGrey);
```

```
$y_line=1;
$line_num=1;
```

```
mysql_select_db("proj",$conn);
```

```
$sql2 = "
SELECT
project_name,
lft,
rgt,
date_from,
date_to,
days,
EXTRACT(YEAR FROM date_from) AS year_from,
EXTRACT(MONTH FROM date_from) AS month_from,
EXTRACT(DAY FROM date_from) AS day_from,
EXTRACT(YEAR FROM date_to) AS year_to,
EXTRACT(MONTH FROM date_to) AS month_to,
EXTRACT(DAY FROM date_to) AS day_to
FROM project WHERE lft>=$_GET[lft] AND rgt<=$_GET[rgt] ORDER BY lft ASC
";
```

```
$result2 = mysql_query($sql2,$conn) or die(mysql_error());
```

```
while($newArray = mysql_fetch_array($result2))
{
```

```
    $project_name=$newArray['project_name'];
    $lft=$newArray['lft'];
    $rgt=$newArray['rgt'];
```

```
    $year_from=$newArray['year_from'];
    $month_from=$newArray['month_from'];
    $day_from=$newArray['day_from'];
```

```
    $year_to=$newArray['year_to'];
    $month_to=$newArray['month_to'];
    $day_to=$newArray['day_to'];
```

```
    $date_from=$newArray['date_from'];
    $date_to=$newArray['date_to'];
    $days=$newArray['days'];
```

```

$date_from_inDays=strtotime($date_from)/3600/24;
$date_to_inDays=strtotime($date_to)/3600/24;

if($line_num==1)
{
$first_date_from=$newArray['date_from'];
$first_day_from=$newArray['day_from'];

$first_date_from_inDays=strtotime($date_from)/3600/24;
$first_date_to_inDays=strtotime($date_to)/3600/24;
$first_day_from_inDays=$day_from;
}

$y_line=$y_line+30;

// Add values to the graph
//$months_array=array(31,28,31,30,31,30,31,31,30,31,30,31);

// Create grid
/*for ($i=1; $i<11; $i++){
imageline($image, $i*25, 0, $i*25, 250, $colorGrey);
imageline($image, 0, $i*25, 250, $i*25, $colorGrey);
}*/

// Create line graph
//for ($i=0; $i<10; $i++)
{
//start point
imageline($image, ($date_from_inDays-$first_date_from_inDays+$first_day_from)*10, $y_line-2,
($date_from_inDays-$first_date_from_inDays+$first_day_from)*10, $y_line+2, $colorBlue);
//end point
imageline($image, ($date_from_inDays-$first_date_from_inDays+$first_day_from)*10+$days*10,
$y_line-2, ($date_from_inDays-$first_date_from_inDays+$first_day_from)*10+$days*10,
$y_line+2, $colorBlue);
//length
imageline($image, ($date_from_inDays-$first_date_from_inDays+$first_day_from)*10, $y_line,
($date_from_inDays-$first_date_from_inDays+$first_day_from)*10+$days*10, $y_line,
$colorBlue);

```

```

//Number of project
ImageString ($image, 1, 1, $y_line, $line_num, $colorRed);

//start date
ImageString ($image, 1, ($date_from_inDays-$first_date_from_inDays+$first_day_from)*10,
$y_line-10, $day_from, $colorGreen);
//end date
ImageString ($image, 1, (($date_from_inDays-
$first_date_from_inDays+$first_day_from)*10+$days*10), $y_line+5, $day_to, $colorGreen);
//duration
ImageString ($image, 1, ($date_from_inDays-
$first_date_from_inDays+$first_day_from)*10+($days*10)/2, $y_line+10, $days, $colorBlue);
}

        $project_names_array[$line_num] = $project_name;

        $line_num++;
    }//End Of While

//create axis x
imageline($image, 10, $y_line+30, $imgWidth, $y_line+30, $colorGrey);

//create axis y
imageline($image, 10, 10, 10, $y_line+30, $colorGrey);

//The first date below axis x
$d = $first_date_from;
ImageString ($image, 1, 160, $y_line+45, $d, $colorBlue);

//days
$k=1;
$minus=0;
for($i=1;$i<$max_days+20;$i++)
{
    $numbers_axis_x = $i-$minus;
    if($numbers_axis_x%30==0)
    {
        $minus=30*$k;
        $k++;
    }

//date below axis x except first month
$d = date('Y-m-d', strtotime($d. ' + 1 months'));
ImageString ($image, 1, $i*10+150, $y_line+45, $d, $colorBlue);
}

```



```
//dates below axis x
imageline($image, $i*50, $y_line+28, $i*50, $y_line+32, $colorGrey);
if($i%5==0)
ImageString ($image, 1, $i*10-3, $y_line+34, $numbers_axis_x, $colorGrey);
//ImageString ($image, 1, $i*50-3, 135, $day_from*10, $colorGrey);
}
```

```
//here i find the date difference in days
//$date_from_days=strtotime($date_from)/3600/24;
//$date_to_days=strtotime($date_to)/3600/24;
```

```
for($j=1;$j<$line_num;$j++)
{
//Name of projects bellow
ImageString ($image, 2, 5, $y_line+50+$j*12, $j, $colorRed);
ImageString ($image, 2, 20, $y_line+50+$j*12, $project_names_array[$j],
$colorRed);
}
```

```
// Output graph and clear image from memory
imagepng($image);
imagedestroy($image);
?>
```

Η σελίδα alert

Αυτή η σελίδα έχει ως σκοπό να μας ενημερώσει για το εάν κάποιο έργο πλησιάζει στο τέλος του. Ελέγχει την πρώτη σειρά του πίνακα "admin" όπου είναι αποθηκευμένες οι μέρες που επιθυμούμε να ενημερωνόμαστε πριν το τέλος του έργου.

```
<?php require("header.php"); ?>
<?php require("mysql_conn.php"); ?>

<div id="content">

<center>

<?php
mysql_select_db("proj",$conn);

$sql2 = "SELECT alert_days FROM admin WHERE table_id=1";
$result2 = mysql_query($sql2,$conn) or die(mysql_error());

    while($newArray = mysql_fetch_array($result2))
    {
        $alert_days=$newArray['alert_days'];
    }

    $current_date = date("Y-m-d");
    $current_date_plus = $current_date;
    for($i=1;$i<=$alert_days;$i++)
    $current_date_plus = date('Y-m-d', strtotime($current_date_plus. '+ 1 days'));

    $sql = "SELECT * FROM project WHERE date_to BETWEEN '$current_date' AND
    '$current_date_plus'";
```

```

$result = mysql_query($sql,$conn) or die(mysql_error());

echo "</table><br>";
echo "<table cellspacing=\"0\" cellpadding=\"0\">";
echo " <tr> ";
echo "   <th scope=\"col\">Project</th> ";
echo "     <th scope=\"col\">Main</th> ";
echo "   <th scope=\"col\">From Date</th> ";
echo "   <th scope=\"col\">To Date</th> ";
echo " </tr> ";

while($newArray = mysql_fetch_array($result))
{
$project_name=$newArray['project_name'];
$lft=$newArray['lft'];
$date_from=$newArray['date_from'];
$date_to=$newArray['date_to'];
$depth=$newArray['depth'];

if($depth>1)
{
$sql = "SELECT project_name FROM project WHERE lft < $lft AND depth=1 ORDER
BY lft ASC";
$result = mysql_query($sql,$conn) or die(mysql_error());
while($newArray = mysql_fetch_array($result))
{
$main_project_name=$newArray['project_name'];
}
}

echo "<tr><td>";

echo "$project_name";

echo "</td><td>";
if($depth>1)
echo "$main_project_name";
echo "</td><td>";

```

```
echo "$date_from";  
echo "</td><td>";  
echo "$date_to";
```

```
echo "</td></tr>";  
}  
?>
```

```
</table>  
</center>  
<br>  
<center><a href = index.php>Return</a></center>  
</div>
```

```
<?php require("footer.php"); ?>
```

Η σελίδα admin

Η σελίδα του διαχειριστή μας δίνει τη δυνατότητα να αλλάξουμε τις ημέρες ειδοποίησης και ακόμα να προσθέσουμε μία στήλη στις πληροφορίες του εκάστοτε έργου ή να τη διαγράψουμε.

```
<?php require("header.php"); ?>  
<?php require("mysql_conn.php"); ?>
```

```
<center>
```

Days To Alert Before The End Of The Project


```
<form name="form" method="POST" action="admin_submit.php" onSubmit="return formValidation(this)">
```

```
<?php
```

```
mysql_select_db("proj",$conn);
```

```
$sql2 = "SELECT alert_days FROM admin WHERE table_id=1";
```

```
$result2 = mysql_query($sql2,$conn) or die(mysql_error());
```

```
while($newArray = mysql_fetch_array($result2))
```

```
{
```

```
  $alert_days=$newArray['alert_days'];
```

```
}
```

```
//Here i print the groups from SQL
```

```
echo "<input type='text' name='alert_days' value = $alert_days size=1 maxlength=3
```

```
/>";
```

```
?>
```

```
<br>
```

```
<input type="submit" value = "SEND" />
```

```
</form>
```

```
<br><br><br>
```

```
Add new column
```

```
<br><br>
```

```
<form name="form1" method="POST" action="admin_enter_column.php" onSubmit="return formValidation(this)">
```

```
<input type="text" name="column_name" size=10 maxlength=30 />
```

```
<br>
```

```
<input type="submit" value = "ADD" />
</form>
```

```
<br><br><br>
Select Column To Delete
<br><br>
```

```
<form name="form2" method="POST" action="delete_column.php" onSubmit="return
formValidation(this)">
```

```
<?php
    mysql_select_db("proj",$conn);

    $sql = "SELECT new_column FROM admin WHERE table_id>1";

    $result = mysql_query($sql,$conn) or die(mysql_error());

    //Here i print the groups from SQL
    echo "<select name = \"new_column\" size=\"3\">";
    //echo "<option> </option>";

    while($newArray = mysql_fetch_array($result))
    {
    $new_column=$newArray['new_column'];

    echo "<option value=\"".$new_column.\">$new_column</option>";
    }
    ?>

<input type="submit" value = "DELETE" />
</form>
```

```
</center>
```

```
<?php require("footer.php"); ?>
```

Η σελίδα `Insert_into_database_project`

Η σελίδα αυτή παίρνει τα στοιχεία του νέου έργου που θέλουμε να προσθέσουμε και τα εισάγει μέσα στη βάση δεδομένων εμφανίζοντας το ανάλογο μήνυμα επιτυχούς ή αποτυχημένης εισαγωγής των δεδομένων.

```
<?php require("header.php"); ?>
<?php require("mysql_conn.php"); ?>

<?php
mysql_select_db("proj",$conn);

$sql = "SELECT * FROM project WHERE lft = $_POST[lft] ORDER BY lft ASC";
$result = mysql_query($sql,$conn) or die(mysql_error());

    while($newArray = mysql_fetch_array($result))
    {
        $date_from_father_display=($newArray['date_from']);
        $date_to_father_display=($newArray['date_to']);

        $date_from_father=strtotime($newArray['date_from']);
        $date_to_father=strtotime($newArray['date_to']);

        $date_from_child=striotime($_POST['date_from']);
        $date_to_child=striotime($_POST['date_to']);
    }

    if(($date_from_child >= $date_from_father && $date_to_child <= $date_to_father) ||
    ($_POST['depth']==0))
    {
        $sql1 = "UPDATE project SET lft=lft+2 WHERE lft > $_POST[lft]";
        $result1 = mysql_query($sql1,$conn) or die(mysql_error());

        $sql2 = "UPDATE project SET rgt=rgt+2 WHERE rgt > $_POST[lft]";
```

```

$result = mysql_query($sql2,$conn) or die(mysql_error());

$lft=$_POST['lft']+1;
$rgt=$lft+1;
$depth=$_POST['depth']+1;

//here i find the date difference in days
$date_from=strtotime($_POST['date_from']);
$date_to=strtotime($_POST['date_to']);
$seconds_diff = $date_to - $date_from;
$days_diff=$seconds_diff/3600/24;

$sql = "insert into project(project_id,project_name,lft,rgt,date_from,date_to,days,depth)
values('NULL',$_POST[project_name],$lft,$rgt,$_POST[date_from'],$_POST[date_to'],$days
_diff,$depth)";

$result = mysql_query($sql,$conn) or die(mysql_error());

echo "<center> Data has been inserted succesfully </center>";
}

else
echo "<center> Dates has to be between $date_from_father_display and
$date_to_father_display</center>";

?>

<br>
<center><a href = index.php>Return</a></center>

<br >
<br >
<?php
mysql_close($conn);
?>

<?php require("footer.php"); ?>

```

Η σελίδα delete_project

Διαγράφουμε το έργο που επέλεξε ο χρήστης, επιστρέφοντας το ανάλογο μήνυμα επιτυχούς ή αποτυχημένης διαγραφής του έργου.

Η σελίδα edit_project

Συνδεόμαστε στη βάση και φορτώνουμε τα δεδομένα που έχουν ήδη προστεθεί από τον χρήστη. Ύστερα, έχει τη δυνατότητα να αντικαταστήσει τα παλιά δεδομένα με οτιδήποτε επιθυμεί.

```
<?php require("header.php"); ?>
<?php require("mysql_conn.php"); ?>

<center>
<table border="0" width="60%" cellpadding="15">
<tr>
<td>
<form name="form" method="POST" action="edit_project_submit.php" onSubmit="return
formValidation(this)">

<table border="0" width="50%">
<tr>

<?php
mysql_select_db("proj",$conn);

        $number_of_created_column=0;

        $sql2 = "SELECT new_column FROM admin WHERE table_id > 1";
        $result2 = mysql_query($sql2,$conn) or die(mysql_error());

        while($newArray = mysql_fetch_array($result2))
        {
            $number_of_created_column++;
            $new_column[$number_of_created_column]=$newArray['new_column'];
        }

        $sql = "SELECT * FROM project WHERE lft=$_GET[lft] AND rgt=$_GET[rgt]";

        $result = mysql_query($sql,$conn) or die(mysql_error());

        while($newArray = mysql_fetch_array($result))
        {
            $project_name=$newArray['project_name'];
            $date_from=$newArray['date_from'];
            $date_to=$newArray['date_to'];
```

```
for($i=1;$i<=$number_of_created_column;$i++)
$data[$i]=$newArray[$new_column[$i]];

}??>
```

```
<td>Name: </td>
<td><input type="text" name="project_name" value = "<?php echo "$project_name"; ?>" size=30
maxlength=40></td>
</tr>
```

```
<tr>
<td>Start Date</td>
```

```
<td>
<!-- calendar attaches to existing form element -->
<input type="text" name="date_from" value = "<?php echo "$date_from"; ?>" size=26
maxlength=10 />
```

```
<script language="JavaScript">
new tcal ({
// form name
'formname': 'form',
// input name
'controlname': 'date_from'
});
```

```
</script>
```

```
</td>
```

```
<tr>
<td>End Date</td>
```

```
<td>
<!-- calendar attaches to existing form element -->
<input type="text" name="date_to" value = "<?php echo "$date_to"; ?>" size=26
maxlength=10 />
```

```
<script language="JavaScript">
new tcal ({
// form name
'formname': 'form',
// input name
'controlname': 'date_to'
});
```

```
</script>
```

```
</td>
```

```

<?php

    for($i=1;$i<=$number_of_created_column;$i++)
    {
        echo "<tr><td>$new_column[$i]</td>";
        echo "<td><input type='text' name='\$i' value = '\$data[$i]' size=30
maxlength=100 /></td></tr> ";
    }

?>

</tr>

<input type="hidden" name="number_of_created_column" value= <?php echo
"$number_of_created_column" ?> >
<?php if($number_of_created_column>0)
{?>
<input type='hidden' name='new_column' value="<?php echo
htmlentities(serialize($new_column)); ?>" />
<?php }?>

<input type="hidden" name="lft" value= <?php echo "$_GET[lft]" ?> >
<input type="hidden" name="rgt" value= <?php echo "$_GET[rgt]" ?> >
<input type="hidden" name="depth" value= <?php echo "$_GET[depth]" ?> >

<tr><td></td><td><center><input type="submit" value = "SEND" /></center></td></tr>
</form>

</table>
</center>

<br>
<center><a href = index.php>Return</a></center>

<br >
<br >
<?php
mysql_close($conn);
?>

<?php require("footer.php"); ?>

```

Η σελίδα edit_project_submit

Σε αυτό το σημείο παίρνουμε τα στοιχεία από τη προηγούμενη οθόνη και τα αποθηκεύουμε στη βάση αντικαθιστώντας τα παλιά δεδομένα.

```
<?php require("header.php"); ?>
<?php require("mysql_conn.php"); ?>

<?php
$number_of_created_column = $_POST['number_of_created_column'];
if($number_of_created_column>0)
$new_column = unserialize($_POST['new_column']);

for($i=1;$i<=$number_of_created_column;$i++)
{
$data[$i] = $_POST[$i];
}

mysql_select_db("proj",$conn);

//here i find the date difference in days
$date_from=strtotime($_POST['date_from']);
$date_to=strtotime($_POST['date_to']);
$seconds_diff = $date_to - $date_from;
$days_diff=$seconds_diff/3600/24;

$sql = "
UPDATE project
SET project_name='$_POST[project_name]', date_from='$_POST[date_from]',
date_to='$_POST[date_to]' , days='$days_diff'
WHERE rgt='$_POST[rgt]' AND lft='$_POST[lft]'
";
$result = mysql_query($sql,$conn) or die(mysql_error());

for($i=1;$i<=$number_of_created_column;$i++)
{
$sql = "
UPDATE project
SET $new_column[$i]='$data[$i]'
WHERE rgt='$_POST[rgt]' AND lft='$_POST[lft]'
";
$result2 = mysql_query($sql,$conn) or die(mysql_error());
}
```

```
echo "<center> Data has been updated succesfully </center>";  
?>
```

```
<br>  
<center><a href = index.php>Return</a></center>
```

```
<br >  
<br >  
<?php  
mysql_close($conn);  
?>
```

```
<?php require("footer.php"); ?>
```

Η σελίδα details

Αυτή η σελίδα μας εμφανίζει ό,τι πληροφορία έχουμε για το έργο που επιλέξαμε. Βλέπουμε και τυχόν νέες στήλες που έχουμε προσθέσει.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>Diplomatiki</title>  
<script language="JavaScript" src="calendar/calendar_db.js"></script>  
<link rel="stylesheet" href="calendar/calendar.css" />  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<link href="style.css" rel="stylesheet" type="text/css" />  
</head>  
<body>
```

```
<?php require("mysql_conn.php"); ?>
```

```
<div id="content">
```

```
<center>
```

```

<?php
    mysql_select_db("proj",$conn);

    echo "<table cellspacing=\"0\" cellpadding=\"0\">";
    echo " <tr> ";
    echo " <th scope=\"col\">Project</th> ";
    echo " <th scope=\"col\">From Date</th> ";
    echo " <th scope=\"col\">To Date</th> ";

    $number_of_created_column=0;

    $sql2 = "SELECT new_column FROM admin WHERE table_id > 1";
    $result2 = mysql_query($sql2,$conn) or die(mysql_error());

    while($newArray = mysql_fetch_array($result2))
    {
        $number_of_created_column++;
        $new_column[$number_of_created_column]=$newArray['new_column'];
        echo " <th scope=\"col\">$new_column[$number_of_created_column]</th> ";
    }

    echo " </tr> ";

    $sql = "SELECT * FROM project WHERE lft = $_GET[lft] AND rgt = $_GET[rgt]";

    $result = mysql_query($sql,$conn) or die(mysql_error());

```



```
}
```

```
    echo "</tr>";
```

```
?>
```

```
</table>  
</center>
```

```
</div>  
</html>
```

Η σελίδα create_project

Όταν γίνει η επιλογή για πρόσθεση νέου έργου ή υποέργου μεταβαίνουμε σε αυτή τη σελίδα όπου συμπληρώνουμε τα απαραίτητα πεδία (ημερομηνία αρχής ,τέλους και όνομα έργου).

```
<?php require("header.php"); ?>  
<?php require("mysql_conn.php"); ?>
```

```
<center>
```

```
<table border="0" width="60%" cellpadding="15">
<tr>
<td>
<form name="form" method="POST" action="insert_into_database_project.php"
onSubmit="return formValidation(this)">
```

```
<table border="0" width="50%">
<tr>
<td>Name: </td>
<td><input type="text" name="project_name" size=30 maxlength=40></td>
</tr>
```

```
<tr>
<td>Start Date</td>
```

```
<td>
<!-- calendar attaches to existing form element -->
<input type="text" name="date_from" size=26 maxlength=10 />
```

```
<script language="JavaScript">
new tcal ({
// form name
'formname': 'form',
// input name
'controlname': 'date_from'
});
</script>
```

```
</td>
```

```
<tr>
<td>End Date</td>
```

```
<td>
<!-- calendar attaches to existing form element -->
<input type="text" name="date_to" size=26 maxlength=10 />
```

```
<script language="JavaScript">
new tcal ({
// form name
'formname': 'form',
// input name
```

```
        'controlname': 'date_to'
    });

</script>

</td>

</tr>

<input type="hidden" name="lft" value= <?php echo "$_GET[lft]" ?> >
<input type="hidden" name="rgt" value= <?php echo "$_GET[rgt]" ?> >
<input type="hidden" name="depth" value= <?php echo "$_GET[depth]" ?> >

<tr><td></td><td><center><input type="submit" value = "SEND" /></center></td></tr>
</form>

</table>

<br><br><br><br>

<center><a href = index.php>Return</a></center>

<br><br><br>

<?php require("footer.php"); ?>
```

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΒΕΛΤΙΩΣΕΙΣ

Η ανάπτυξη του «project management» και οι συνεχώς αυξανόμενες απαιτήσεις έχουν συντελέσει στην ολοένα μεγαλύτερη ανάγκη για ανάπτυξη εργαλείων. Σκοπός του είναι οργάνωση των έργων να γίνεται με τέτοιο τρόπο ώστε να αποτραπεί τυχόν επικάλυψη υποέργων και για αποφυγή της μη ενημέρωσης σε περίπτωση μη τήρησης των χρονικών ορίων κ.α.

Αυτή η εργασία αποτελεί τον κορμό ενός συστήματος οργάνωσης έργων όπου με κάποιες προσθήκες θα μπορούσε να γίνει ένα πολύ αποτελεσματικό διαδικτυακό εργαλείο.

Μελλοντικά, θα μπορούσε το σύστημα να επεκταθεί έτσι ώστε να υποστηρίξει χρήστες οι οποίοι ανάλογα με τα δικαιώματα που θα τους δοθούν θα μπορούν να έχουν πρόσβαση μόνο στα έργα που έχουν δημιουργήσει οι ίδιοι ή και άλλοι. Μία ακόμη σημαντική προσθήκη θα ήταν η ενσωμάτωση ενός συστήματος διαχείρισης περιεχομένου με περισσότερες δυνατότητες όπως το Joomla.

Τέλος, θα μπορούσε να υποστηρίξει εισαγωγή δεδομένων από άλλες αντίστοιχες εφαρμογές όπως είναι για παράδειγμα το Microsoft Project.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Παρουσίαση: Διοίκηση ανθρώπινου δυναμικού της διεύθυνσης πληροφορικής Στελέχωση έργων πληροφορικής Οργανωτικές δομές. Δημήτρης Αποστόλου Επ. Καθηγητής
2. <http://mikehillyer.com/articles/managing-hierarchical-data-in-mysql/>
3. <http://mikehillyer.com/articles/managing-hierarchical-data-in-mysql/>
4. <http://php.net/manual/en/book.image.php>
5. <http://www.dreamincode.net/forums/topic/54393-a-beginners-look-at-the-gd-library-part-23/>
6. <http://php.net/manual/en/book.image.php>
7. http://ifestos.teilar.gr/index.php?option=com_docman&task=doc_view&gid=238
8. <http://www.w3schools.com/>

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑΣ