

**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΡΗΤΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΠΟΛΥΜΕΣΩΝ**



Πτυχιακή εργασία Λογισμικό απεικόνισης προγράμματος μαθημάτων σε Flash.

Flash presentation of course syllabus.

Σπουδαστής : Παντελής Δρουμπάκης

Επιβλέπων Καθηγητής : Νικόλαος Βιδάκης

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	3
ΠΕΡΙΛΗΨΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	5
ABSTRACT	7
ΚΕΦΑΛΑΙΟ 1.....	8
1.1 ΣΤΟΧΟΣ	8
1.2 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	9
1.2.1 HTML.....	9
1.2.1.1 DOCTYPE	10
1.2.2 CSS.....	12
1.2.2.1 Εισαγωγή ορισμάτων CSS	12
1.2.2.2 Κλάσεις CSS	13
1.2.3.1 Ιεραρχία JavaScript	16
1.2.4 PHP.....	18
1.2.4.1 Ιστορία της PHP	18
1.2.4.2 Τι είναι η PHP	19
1.2.4.3 Πλεονεκτήματα της PHP	19
1.2.4.4 Που Χρησιμοποιείται ένα PHP script.....	21
1.2.4.5 Τι μπορεί να κάνει η PHP.....	22
1.2.4.6 Συναρτήσεις.....	22
1.2.5 XML	24
1.2.5.1 XML δήλωση	24
1.2.5.2 Σύνταξη της xml.....	25
1.2.5.3 Πλεονεκτήματα και προοπτικές	27
1.2.5.4 Μειονεκτήματα της XML.....	27
1.2.5.5 Προσχεδιασμένοι στόχοι της XML	28
1.2.5.6 Elements ή Στοιχεία	28
1.2.5.7 Attributes ή Ιδιότητες	29
1.2.5.8 Η διαφορά μεταξύ XML και HTML	29
1.2.6 Action script 3.....	30
1.2.6.1 Classes, objects , packages	30
1.2.6.2 Κλάση object().....	31
1.2.6.3 Προσδιοριστές πρόσβασης.....	32
1.3 ΤΕΧΝΟΛΟΓΙΑ.....	33
1.3.1 Εγκατάσταση Xampp	33
ΚΕΦΑΛΑΙΟ 2.....	36
2.1 ΔΙΕΠΑΦΗ ΛΟΓΙΣΜΙΚΟΥ	36
2.1.1 Αρχική σελίδα.....	38
2.1.2 Εισαγωγή δεδομένων.....	38
2.1.3 Επιβεβαίωση	41
2.1.4 Προβολή Δεδομένων.....	42
2.1.5 Προσθήκη , Διαγραφή , Αλλαγή μαθήματος	44
2.1.5.1 Προσθήκη.....	44
2.1.5.2 Διαγραφή.....	45
2.1.5.3 Αλλαγή.....	46
2.1.6 Προβολή δεδομένων στο Flash	46
ΚΕΦΑΛΑΙΟ 3.....	50
DATA MODEL.....	50
3.1 ΑΙΘΟΥΣΕΣ XML	51
3.2 ΜΑΘΗΜΑΤΑ XML	52
ΚΕΦΑΛΑΙΟ 4.....	54
ΚΩΔΙΚΑΣ.....	54
4.1 ΕΠΙΛΕΓΜΕΝΟΣ ΚΩΔΙΚΑΣ	55

ΚΕΦΑΛΑΙΟ 5 - ΕΠΙΛΟΓΟΣ	60
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	60
5.2 ΕΞΕΛΙΞΗ ΣΥΣΤΗΜΑΤΟΣ	60

Περίληψη Πτυχιακής Εργασίας.

Αυτή η πτυχιακή εργασία έχει ως στόχο την παρουσίαση των τεχνολογιών flash, οι οποίες αφορούν στην σχεδίαση και υλοποίηση λογισμικού διαχείρισης και παρουσίασης εκπαιδευτικού προγράμματος από το Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων του ΤΕΙ Ηρακλείου (Ε.Π.Π.).

Συγκεκριμένα:

- Δίνει δυνατότητα διαχείρισης των δεδομένων κάθε μαθήματος μέσω κατάλληλης φόρμας. Τα δεδομένα του μαθήματος θα αποθηκεύονται σε αρχείο XML. Δηλαδή, ο χρήστης θα μπορεί να μεταβάλει δεδομένα, όπως την αλλαγή της αίθουσας, της ομάδας, της ώρας και της μέρας διεξαγωγής του μαθήματος και θα μπορεί να διαγράψει κάποιο μάθημα.
- Το XML αρχείο θα ενημερώνει το πρόγραμμα με βάση τα στοιχεία των μαθημάτων της σχολής (εξάμηνο, τίτλο μαθήματος, θεωρία ή εργαστήριο, ομάδα -αν είναι πάνω από 2 ομάδες σε ένα μάθημα-, αίθουσα, όνομα καθηγητή).
- Δυνατότητα προβολής του προγράμματος μαθημάτων με την χρήση τεχνολογίας flash.
 - Ανάγνωση XML (αρχείο δεδομένων μαθήματος).
 - Δυναμική κατανομή και τοποθέτηση αντικειμένων σε καμβά εβδομαδιαίου προγράμματος.
 - Παρουσίαση των δεδομένων. Απεικόνιση των στοιχείων των μαθημάτων.
 - Ο καμβάς (flash) θα είναι δυναμικός. Θα μπορεί να ελέγχει αν την ίδια ημέρα και ώρα συμπέτουν δυο ή και παραπάνω μαθήματα και θα προσαρμόζεται ανάλογα με τις απαιτήσεις του προγράμματος μαθημάτων της σχολής.
- Δίνει την δυνατότητα οποιασδήποτε προβολής επιλέξουμε στο flash (καμβάς).

Ανάλογα με την επιλογή που θα πραγματοποιείται κάθε φορά, στα στοιχεία του καμβά (component ή combobox), το πρόγραμμα θα αλλάζει και θα προσαρμόζεται ανάλογα. Αυτές οι επιλογές είναι :

- Δυνατότητα επιλογής εξαμήνου προβολής.
 - Δυνατότητα προβολής συγκεκριμένου μαθήματος.
 - Δυνατότητα προβολής μαθημάτων συγκεκριμένου καθηγητή.
 - Δυνατότητα προβολής συγκεκριμένης αίθουσας.
- Δίνει το τελικό αποτέλεσμα σε ιστοχώρο (web site) κατασκευασμένο σε PHP.

Για την καλύτερη κατανόηση των παραπάνω, δίνεται συγκεκριμένο παράδειγμα από το πρόγραμμα μαθημάτων του τμήματος Ε.Π.Π.

Abstract

This final work aims at as the presentation of technologies flash, that concerns in the designing and concretisation of software of management and presentation of educational program from the Department of Applied Information technology and Multimedias of POLYTECHNIC COLLEGE Heraklion ([E].[P].[P]).

Concretely, the software:

- It gives possibility of management of given each course via suitable [formas]. The data of course will be stored in file XML. That is to say, the user might alter data, as the change of room, the team, the hour and the day of conduct of course and he might erase some course.
- The XML file will inform the program with base the elements of courses of faculty (half-year period, title of course, theory or laboratory, team - if she is above 2 teams in a course, room, name of professor).
- It gives possibility of projection of program of courses with the use of technology flash.
 - Reading XML (file of data of course).
 - Dynamic distribution and placement of objects in canvas of weekly program.
 - Presentation of data. Depiction of elements of courses.
 - The canvas (flash) will be dynamic. It might check [an] the same day and hour coincide two or even [parapano] courses and it will be adapted depending on the requirements of program of courses of faculty.
- It gives the possibility of any projection we select in flash (canvas).
 - Depending on the choice that will be realised each time, in the elements of canvas (component or combobox), the program will change and it will be adapted proportionally. These choices are:
 - Possibility of choice of half-year period of projection.
 - Possibility of projection of concrete course.
 - Possibility of projection of courses of concrete professor.
 - Possibility of projection of concrete room.
- It gives the final result in website (web site) manufactured in PHP.
- For the better comprehension more, is given concrete example from the program of courses of department [E].[P].[P].

ΚΕΦΑΛΑΙΟ 1

Σε αυτό το κεφάλαιο θα γίνει η περιγραφή του στόχου της πτυχιακής εργασίας, των γλωσσών προγραμματισμού και της τεχνολογίας που έχει χρησιμοποιηθεί για την υλοποίηση της.

1.1 Στόχος.

Στόχος της συγκεκριμένης πτυχιακής εργασίας είναι η ψηφιοποίηση του εβδομαδιαίου εκπαιδευτικού προγράμματος του τμήματος Ε.Π.Π., με την δυνατότητα δυναμικής επεξεργασίας των μαθημάτων και συνολικής προβολής αυτών σε flash.

Για την υλοποίηση της εφαρμογής, έχουν χρησιμοποιηθεί τα παρακάτω προγράμματα Adobe Dreamweaver CS3, Adobe Flash CS4, Photoshop CS3, Notepad++. Αυτά τα προγράμματα χρησιμοποιήθηκαν για την καλύτερη και ευκολότερη επεξεργασία των γλωσσών προγραμματισμού. Οι γλώσσες αυτές είναι οι HTML , PHP , CSS , Actionscript 3, και XML. Αναλυτικότερα :

1. HTML για την δημιουργία του web site και την ενσωμάτωση της PHP.
2. PHP για την δημιουργία και επεξεργασία του XML .
3. CSS για την σχεδίαση του web site.
4. Action script για την δημιουργία του flash και προβολής των μαθημάτων σε αυτό.

1.2 Γλώσσες προγραμματισμού.

1.2.1 HTML

Η HTML (HYPERTEXT MARKUP LANGUAGE), είναι μία γλώσσα η οποία «λέει» στον υπολογιστή πως θα πρέπει να εμφανίσει μία ιστοσελίδα. Τα αρχεία στα οποία γράφεται αυτή η γλώσσα είναι απλά αρχεία κειμένου (ASCII), τα οποία μπορούν να δημιουργηθούν ακόμη και με το Notepad (Σημειωματάριο) των Windows έχοντας απλώς την κατάληξη .htm ή .html. Περιλαμβάνουν μέσα τα λεγόμενα tags, τα οποία χρησιμοποιούνται σε κάποιο πρόγραμμα εμφάνισης ιστοσελίδων (π.χ. Firefox), για να δημιουργηθούν και να εμφανιστούν οι ιστοσελίδες.

Όλα τα tags της HTML, δηλώνονται ως εξής: < ονομασία του tag >. Όλα τα tags της HTML έχουν μία αρχή και ένα τέλος. Στο ενδιάμεσο παρεμβάλλεται το περιεχόμενο του tag. Η αρχή κάθε tag δηλώνεται με <tag> ενώ το τέλος με </tag>. Γενικά το περιεχόμενο μίας ιστοσελίδας είναι tags τα οποία περιέχουν άλλα tags κ.λ.π. Πάντα τα tags τα οποία δημιουργούνται (ανοίγουν), θα πρέπει αφού πληκτρολογηθεί το περιεχόμενό τους να κλείνουν (τελειώνουν </>). Υπάρχουν εξαιρέσεις, κάποια tags δεν κλείνουν ποτέ. Πρέπει λοιπόν όλα τα tags τα οποία ανοίγουν αφού πληκτρολογηθεί το περιεχόμενό τους να κλείνουν, εκτός βέβαια των tags τα οποία δεν πρέπει εξ' ορισμού να κλείσουν.

Παρακάτω περιγράφεται η σημασία των tags που πάντα πρέπει να υπάρχουν σε μια σελίδα HTML.

<html> TAG με ονομασία HTML. Στην κορυφή του κώδικα βρίσκεται πάντα το tag αυτό, το οποίο δηλώνει ότι ο κώδικας που ακολουθεί παρακάτω είναι κώδικας HTML.

<head> TAG με ονομασία HEAD. Ακριβώς μετά το tag html, βρίσκεται το tag head. Ανάμεσα στο άνοιγμα (< >) και στο κλείσιμο (</ >) του tag, αυτού, υπάρχει περιεχόμενο που αφορά την ιστοσελίδα όπως:

1. Ο τίτλος της
2. Η γλώσσα που υποστηρίζει η σελίδα (π.χ. ελληνικά)

3. Διάφορα keywords κ.α. Το περιεχόμενο αυτό, δεν πρέπει να γραφτεί σε κανένα άλλο μέρος του κώδικα.

</head> TAG με ονομασία HEAD. Σε αυτό το σημείο το tag head κλείνει.

<body> TAG με ονομασία BODY. Από αυτό το σημείο και έπειτα, γράφεται ο κώδικας που αφορά την εμφάνιση της ιστοσελίδας μας στον browser π.χ. τα κείμενα, οι εικόνες, οι φόρμες, κ.λ.π. θα εισάγονται πάντα με τη βοήθεια άλλων tags

</body> TAG με ονομασία BODY. Σε αυτό το σημείο το tag body, κλείνει και πάντα πριν κλείσει το tag HTML.

</html> TAG με ονομασία HTML. Πάντα στο τέλος της σελίδας το tag με ονομασία HTML κλείνει, δηλώνοντας ότι δεν υπάρχει πλέον άλλος κώδικας HTML.

1.2.1.1 DOCTYPE

Επίσης πρέπει να περιέχει κάποιο τύπο DOCTYPE . Το DOCTYPE (*document type declaration*) είναι πολύ βασικό στην κατασκευή ιστοσελίδων – websites. Η χρήση του είναι απαραίτητη όταν θέλουμε να πληρούμε τα παγκόσμια πρότυπα W3C και να πετυχαίνουμε μέγιστη συμβατότητα με τα προγράμματα πλοήγησης διαδικτύου “browsers”.

Υπάρχουν διάφοροι τύποι DOCTYPE αντίστοιχα για HTML – XHTML και το καθένα λειτουργεί ανάλογα με την χρήση του. Αναλυτικότερα:

1.2.1.1.1 HTML 4.01 Strict

Σύμφωνα με τα Web standards το HTML 4.01 Strict DOCTYPE είναι αυτό που πρέπει να χρησιμοποιούμε στον σχεδιασμό web εφαρμογών εάν θέλουμε να πετύχουμε 100% εγκυρότητα (valid). Ουσιαστικά είναι μια «μειωμένη» έκδοση της HTML 4.01 που δίνει έμφαση περισσότερο στην δομή και όχι στην παρουσίαση. Τα Deprecated elements και χαρακτηριστικά (attributes) όπως frames, και link targets δεν επιτρέπονται στην HTML 4 Strict. Το βασικό πλεονέκτημα είναι πως με τη χρήση του συγκεκριμένου DOCTYPE ενημερώνουμε τους browsers να κάνουν «render» την ιστοσελίδα με τον πιο αυστηρό και έγκυρο (valid) τρόπο πετυχαίνοντας μέγιστη συμβατότητα και εγκυρότητα. Το μειονέκτημα όμως είναι πως είναι αρκετά περιοριστικό και δεσμευτικό σε θέματα

παρουσίασης, δεν επιτρέπονται απλές βασικές λειτουργίες όπως το target=”_blank” και θέλει αρκετή εξοικείωση από την HTML 4.01 Transitional.

Συμπέρασμα:

Χρησιμοποιήστε το HTML 4.01 Strict DOCTYPE όταν θέλετε να σχεδιάσετε web εφαρμογές που φιλοξενούν δομημένη πληροφόρηση και θέλετε να πετύχετε μέγιστη εγκυρότητα – συμβατότητα σε όλες τις πλατφόρμες «θυσιάζοντας» όμως την γραφιστική λεπτομέρεια.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

1.2.1.1.2 Transitional

Το HTML 4.01 Transitional DOCTYPE πιθανώς να είναι το πιο διαδεδομένο DOCTYPE στο διαδίκτυο. Η αρχική του χρήση ήταν για την μετάβαση από παλαιότερες εκδόσεις HTML σε HTML 4.01. Τώρα πλέον χρησιμοποιείται κυρίως γιατί υποστηρίζει την χρήση παλαιών deprecated tags τα οποία δεν υποστηρίζονται στην Strict έκδοση. Ουσιαστικά προσφέρει περισσότερη ελευθερία στους web designers και προωθεί καλύτερα τον τρόπο παρουσίασης μιας ιστοσελίδας ενώ πληρεί τα παγκόσμια πρότυπα W3C. Χρησιμοποιώντας το συγκεκριμένο DOCTYPE ενημερώνετε τους browsers ότι χρησιμοποιείτε την τελευταία έκδοση HTML αλλά θα πρέπει να προσέξετε το cross browser compatibility (συμβατότητα με τα προγράμματα πλοήγησης διαδικτύου) ώστε να έχετε συμβατότητα με τους browsers.

Συμπέρασμα:

Χρησιμοποιήστε το HTML 4.01 Transitional DOCTYPE όταν θέλετε να σχεδιάσετε web εφαρμογές που απαιτούν πιο πολύπλοκο τρόπο παρουσίασης και θέλετε να έχετε ελευθερία κινήσεων. Δεν συνιστάται εάν επιθυμείτε συμβατότητα σε όλες τις πλατφόρμες όπως wap sites κτλ.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Περισσότερα <http://www.webdesignblog.gr/web-design-valid-use-doctypes/>

1.2.2 CSS

Τα CSS (Cascading Style Sheets), είναι ένα σύνολο από ορίσματα, διαφορετικά από αυτά που συναντάμε στην HTML, τα οποία αναπτύχθηκαν με στόχο την καλύτερη διαχείριση της εμφάνισης των ιστοσελίδων. Μπορούμε μέσα από τα CSS, να ορίσουμε γραμματοσειρές, χρώματα, στοίχιση, backgrounds, κ.λ.π.

Ουσιαστικά αποτελούν κλάσεις αντικειμένων τα οποία παρέχουν μεθόδους για υλοποίηση σε κάποια tags της HTML.

Για παράδειγμα μία κλάση CSS μπορεί να είναι η main , η οποία ορίζει ότι η γραμματοσειρά που θα χρησιμοποιηθεί είναι η Verdana, μεγέθους 11, bold και το κείμενο θα είναι στοιχισμένο στο κέντρο. Για να εφαρμόσουμε αυτή την κλάση και τις ιδιότητες – μεθόδους της μέσα στην HTML, χρησιμοποιούμε εντός όλων σχεδόν των tags της HTML που αφορούν σε εμφάνιση κειμένου, πινάκων, γραμμών ή στηλών πινάκων, links, φορμών κ.λ.π. την παράμετρο class="όνομα κλάσης CSS". Για παράδειγμα αν θέλουμε σε μία παράγραφο να εφαρμόσουμε τις ιδιότητες της κλάσης main που αναφέραμε παραπάνω πρέπει να γράψουμε:

```
<p class="main">Το κείμενό μας</p>
```

Τα ορίσματα των CSS, μπορούν να εισαχθούν ως κώδικας τόσο μέσα στην HTML, όσο και σε ξεχωριστό αρχείο με κατάληξη .css το οποίο στη συνέχεια αναφέρεται ως link στην HTML.

1.2.2.1 Εισαγωγή ορισμάτων CSS

Για να ορίσετε ότι σε μία σελίδα HTML, θα εισαχθεί κώδικας CSS πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

```
<style type="text/css">
```

```
<!--
```

```
Ακολουθεί ο κώδικας των CSS.
```

```
-->  
</style>
```

Στην περίπτωση που ο κώδικας των CSS, έχει γραφτεί σε ξεχωριστό αρχείο με κατάληξη .css τότε αντίστοιχα πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

```
<link href=" διαδρομή και όνομα αρχείου.css " rel="stylesheet" type="text/css">
```

(μπορούμε να έχουμε περισσότερα από ένα αρχεία css στην ίδια σελίδα , μεγαλύτερη προτεραιότητα έχει αυτό που γράφτηκε τελευταίο.)

1.2.2.2 Κλάσεις CSS

Παραπάνω αναφέραμε ότι τα CSS, αποτελούνται από κλάσεις και ιδιότητες αυτών των κλάσεων. Ένα παράδειγμα μπορεί να είναι το παρακάτω το οποίο έχει δύο κλάσεις τις body και header οι οποίες ορίζουν διαφορετικές ιδιότητες.

```
body {  
    background-color:#FFFFCC;  
}  
#header {  
    background-image:url(images/logo.jpg);  
    width: auto;  
    height: 130px;  
    margin:10;  
    padding: 10;  
}
```

Κοινώς για να ορίσετε μία κλάση CSS, εντός ενός αρχείου ή εντός της ίδιας της HTML σελίδας, όπως αναφέρθηκε παραπάνω πρέπει να εισάγετε τα εξής:

```
.όνομα κλάσης {  
ιδιότητες κλάσης
```

}

1.2.3 JavaScript

Η JavaScript είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που χρησιμοποιείται για να προσθέσει εφέ και διαλογικότητα (αλληλεπίδραση, διαδραστικότητα) στις ιστοσελίδες μας και είναι ανταγωνιστική της γλώσσας προγραμματισμού VBScript. Δημιουργήθηκε από την εταιρεία Netscape και το αρχικό της όνομα ήταν LiveScript. Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεσθεί αμέσως ή όταν λαμβάνει χώρα ένα συμβάν (event). Δεν γίνεται μεταγλώττιση (compilation) του κώδικα της JavaScript, αρκεί μόνο ο φυλλομετρητής (browser) να υποστηρίζει την JavaScript.

Ένα παράδειγμα από το πρόγραμμα μας είναι :

```
<form name="form" method="post" onSubmit="return check_if_id_is_empty()"
action="confirm.php">
```

Καλεί την συνάρτηση check_if_id_is_empty() η οποία βρίσκεται μέσα <head>

```
<head>
<title>new_data</title>
<script type="text/javascript">
function check_if_id_is_empty()
    {
        var arxi=document.form.ora_arxi.value;
        var telos=document.form.ora_telos.value;
        arxi++;
        for (var ii=0; ii<=1;ii++){
            arxi++;
            if (arxi == telos ){
                return true;
            }
        }
    }
}
```

```
        alert("lathos ora");
        return false;

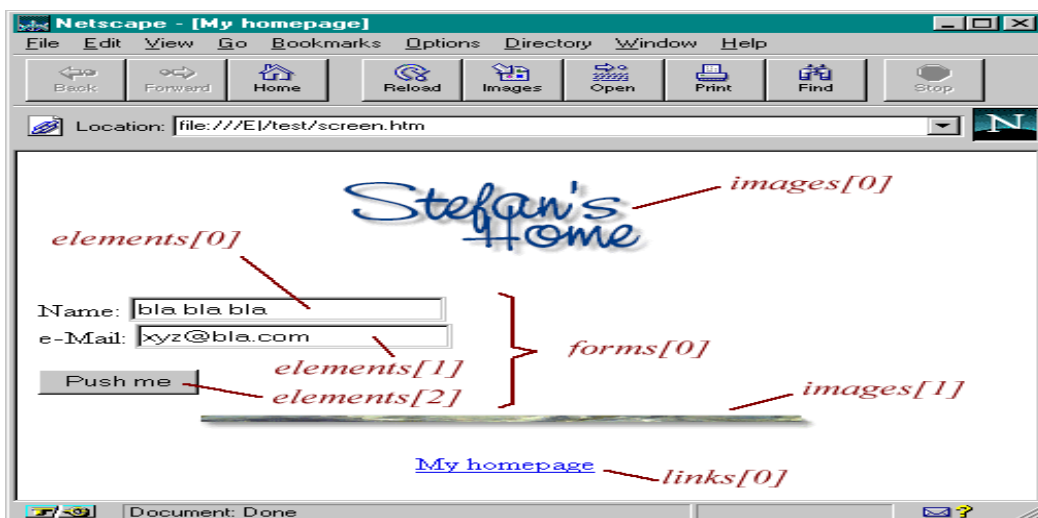
    }
</script>
```

Το script αυτό ελέγχει αν η ώρα έναρξης κάποιου μαθήματος είναι κατά 2 ή 3 ώρες μικρότερη από την ώρα λήξης. Αν το “return” επιστρέψει με τιμή “true”, τότε το πρόγραμμα συνεχίζει και προχωρά στην επόμενη σελίδα, διαφορετικά, παραμένει στην ίδια σελίδα.

1.2.3.1 Ιεραρχία JavaScript

Η JavaScript οργανώνει όλα τα στοιχεία μιας web σελίδας σε μια ιεραρχία. Κάθε στοιχείο της σελίδας βλέπεται σαν αντικείμενο. Κάθε αντικείμενο έχει τις δικές του ιδιότητες (properties) και μεθόδους (methods). Με τη βοήθεια του JavaScript μπορείτε εύκολα να διαχειριστείτε τα αντικείμενα .

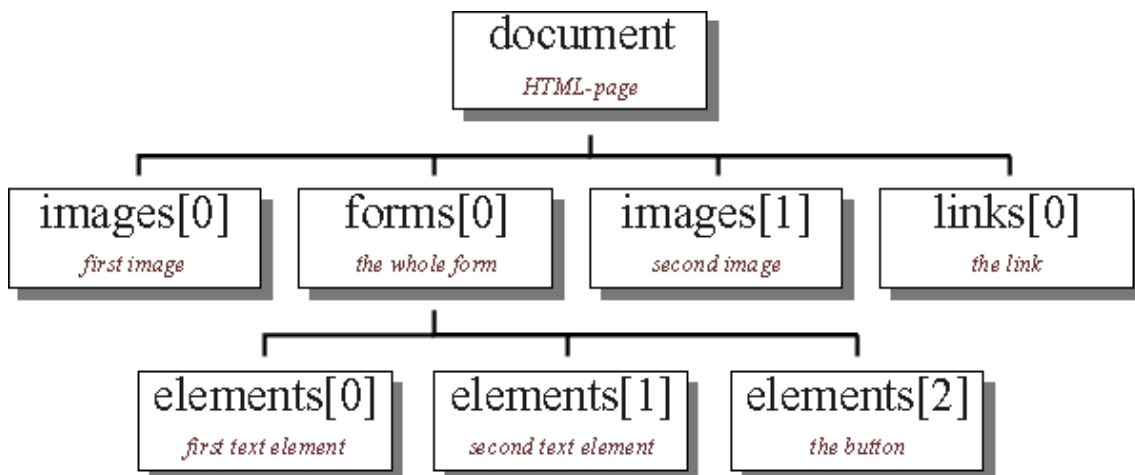
Παράδειγμα:



1.1 Ιεραρχία της JavaScript σε μια web σελίδα.

Έχουμε δύο εικόνες (images), ένα link και μία φόρμα (form) με δύο πεδία κειμένου και ένα κουμπί. Από πλευράς JavaScript, το παράθυρο του browser θεωρείται σαν ένα αντικείμενο, εν ονόματι window. Το αντικείμενο window περιέχει συγκεκριμένα στοιχεία, όπως η StatusBar. Μέσα στο παράθυρο, μπορεί να γίνει η ‘φόρτωση’ ενός HTML έγγραφου (ή άλλου τύπου αρχείου - προς το παρόν, θα περιοριστούμε σε HTML αρχεία). Αυτή η σελίδα είναι το αντικείμενο document. Αυτό σημαίνει ότι το αντικείμενο document αναπαριστά το HTML έγγραφο το οποίο ‘βλέπει’ ο χρήστης εκείνη τη στιγμή. Το document είναι ένα πολύ σημαντικό αντικείμενο στο JavaScript - θα χρησιμοποιείται συχνά. Οι ιδιότητες (properties) του document είναι για παράδειγμα το χρώμα του φόντου. Αλλά το πιο σημαντικό είναι ότι όλα τα αντικείμενα HTML είναι ιδιότητες του document. Αντικείμενα είναι για παράδειγμα τα links, ή οι φόρμες.

Η παρακάτω εικόνα παρουσιάζει την ιεραρχία του HTML εγγράφου του προηγούμενου παραδείγματος:



1.2.4 PHP

Η PHP είναι μια γλώσσα script από την πλευρά του διακομιστή, σχεδιασμένη ειδικά για το Web. Μέσα σε μια HTML σελίδα μπορεί να ενσωματωθεί ο PHP κώδικας, που θα εκτελείται κάθε φορά που κάποιος θα επισκέπτεται τη σελίδα. Ο PHP κώδικας μεταφράζεται στο Web διακομιστή και δημιουργεί HTML ή άλλη έξοδο που θα δει ο επισκέπτης.

1.2.4.1 Ιστορία της PHP

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

1.2.4.2 Τι είναι η PHP

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: HyperText preprocessor.

Είναι μια server-side (εκτελείται στον διακομιστή) scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα, η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), αντ' αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες.

1.2.4.3 Πλεονεκτήματα της PHP

Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι ο Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) και Allaire Cold Fusion.

Σε σύγκριση με αυτά τα προϊόντα, η PHP έχει πολλά πλεονεκτήματα όπως :

1. Υψηλή απόδοση
2. Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
3. Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
4. Χαμηλό κόστος
5. Ευκολία μάθησης και χρήσης
6. Διαθεσιμότητα του κώδικα προέλευσης

Ακολουθεί μια πιο λεπτομερή περιγραφή αυτών των πλεονεκτημάτων.

1. Απόδοση

Η PHP είναι πολύ αποτελεσματική. Με ένα φθηνό διακομιστή μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά. Οι δοκιμές που δημοσιεύθηκαν από την Zend Technologies (<http://www.zend.com>), δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

2. Ολοκλήρωση με Βάσεις δεδομένων

Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL, μπορεί κάποιος να συνδεθεί κατευθείαν με τις βάσεις δεδομένων PostgreSQL , mSQL , Oracle , dbm , filePro , Informix , InterBase , Sybase, μεταξύ άλλων. Χρησιμοποιώντας το Open Database Connectivity Standard (ODBC) μπορεί να συνδεθεί σε οποιαδήποτε βάση δεδομένων παρέχει ένα πρόγραμμα οδήγησης ODBC . Αυτό περιλαμβάνει και τα προϊόντα της Microsoft products.

3. Ενσωματωμένες Βιβλιοθήκες

Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web, έχει πολλές ενσωματωμένες βιβλιοθήκες που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web. Μπορεί κάποιος να δημιουργήσει εικόνες GIF δυναμικά, να συνδεθεί με άλλες υπηρεσίες δικτύων, να στείλει ηλεκτρονικό ταχυδρομείο, να δουλέψει με cookies και να δημιουργήσει PDF έγγραφα : όλα αυτά με λίγες γραμμές κώδικα.

4. Κόστος

Η PHP είναι δωρεάν. Μπορεί κάποιος να ‘κατεβάσει’ την τελευταία έκδοση από το <http://www.php.net> , χωρίς χρέωση.

1.2.4.4 Που Χρησιμοποιείται ένα PHP script.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- **Server-side scripting.** Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Απαιτούνται τρία πράγματα για να δουλέψει αυτό. Τον PHP μεταγλωττιστή (parser) (CGI ή server module), ένα webserver (εξυπηρετητή σελίδων) και ένα web browser ("φυλλομετρητή"). Πρέπει κάποιος να 'τρέξει' τον webserver, με μια συνδεδεμένη εγκατάσταση της PHP. Μπορεί να προσπελάσει τα αποτελέσματα του PHP προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.
- **Command line scripting.** Μπορεί κάποιος να φτιάξει ένα PHP script για να το 'τρέξει' χωρίς server ή browser. Απαιτείται μόνο ο PHP μεταγλωττιστής για να χρησιμοποιηθεί με αυτό τον τρόπο. 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 επεξεργασίες κειμένου.
- **Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών).** Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυρικές εφαρμογές, αλλά αν γνωρίζει την PHP πολύ καλά και θέλει να χρησιμοποιήσει κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές, μπορεί επίσης να χρησιμοποιήσει το PHP-GTK για αυτού του είδους τα προγράμματα. Έχει επίσης τη δυνατότητα να γράφει cross-platform εφαρμογές με αυτό τον τρόπο. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή. Αν κάποιος ενδιαφέρεται για το PHP-GTK, θα πρέπει να επισκεφθεί την δική του ιστοσελίδα.

Επίσης, η PHP μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP υποστηρίζει επίσης τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd, και πολλούς άλλους webserver. Για την πλειοψηφία των server η PHP έχει ένα module, για τους υπόλοιπους η PHP μπορεί να λειτουργήσει ως ένας CGI

επεξεργαστής. Έτσι με την PHP, υπάρχει η ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, μπορεί κάποιος ελεύθερα να χρησιμοποιήσει συναρτησιακό (procedural) ή αντικειμενοστραφή (object oriented) προγραμματισμό ή μια ανάμειξη τους. Αν και η παρούσα έκδοση δεν υποστηρίζει όλα τα πρότυπα χαρακτηριστικά, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR) είναι γραμμένες μόνο με αντικειμενοστραφή κώδικα.

Με την PHP δεν είναι κάποιος περιορισμένος να εξάγει HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας τα libswf και Ming) παράγονται αμέσως. Μπορεί επίσης να πραγματοποιηθεί εύκολα οποιοδήποτε κείμενο όπως XHTML και οποιοδήποτε άλλο XML αρχείο. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό σας περιεχόμενο.

Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη που έχει για ένα μεγάλο σύνολο βάσεων δεδομένων. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή.

1.2.4.5 Τι μπορεί να κάνει η PHP.

Οτιδήποτε. Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορεί να κάνει τα ίδια πράγματα με οποιοδήποτε άλλο CGI πρόγραμμα, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στείλει και να πάρει cookies.

1.2.4.6 Συναρτήσεις.

Στην PHP, για να «μεταφερθούν» οι τιμές που έχουν οι μεταβλητές μας από μία σελίδα σε μία άλλη, πρέπει να αποθηκευθούν προσωρινά. Αυτό γίνεται με 2 μεθόδους την GET και την POST.

1.2.4.6.1 GET

Όταν χρησιμοποιείται η μέθοδος GET τα δεδομένα είναι σε μορφή text και είναι ορατά από τον καθένα στο URL που καλεί το script. Επιπλέον η ποσότητα των δεδομένων που μπορούν να μεταδοθούν με τη χρήση της μεθόδου GET είναι περιορισμένη. Έτσι για να περάσουμε μια μεταβλητή που θα μας δίνει το όνομα κάποιου γίνεται ως εξής: (selida.php?name=Pantelis).

1.2.4.6.2 POST

Όταν χρησιμοποιείται η μέθοδος POST τα δεδομένα μεταδίδονται με ξεχωριστή σύνδεση στο script και δεν είναι ορατά στο URL που καλεί το script. Επιπλέον η ποσότητα των δεδομένων που μπορούν να μεταδοθούν με τη χρήση της μεθόδου POST. Τα δεδομένα στέλνονται κυρίως από φόρμες, όπου μέσα στο \$_POST[""] βάζουμε το όνομα (name) του πεδίου που μας στέλνει η φόρμα.

Οι πληροφορίες που στέλνονται από την φόρμα μας με την μέθοδο POST είναι «αόρατες», και δεν έχουν όρια στο μέγεθος των πληροφοριών που μπορούν να στείλουν.

1.2.4.6.3 Foreach

Η foreach χρησιμοποιείται για να κινούμαστε μέσα σε πίνακες. Για κάθε βρόχο που περνάμε, η αξία του τρέχοντος στοιχείου του πίνακα γίνεται ίση με το **\$value** και η θέση του πίνακα μετακινείται κατά ένα.

```
foreach (πίνακας as value)
{
    κωδίκας
}
```

1.2.5 XML

Η γλώσσα XML αναπτύχθηκε από μια Ομάδα Εργασίας της XML κάτω από την καλή κηδεμονία του διεθνούς οργανισμού World Wide Web Consortium (W3C) το 1996. Εδραιώθηκε από τον John Bosak της Sun Microsystems με την ενεργή συμμετοχή μιας XML Ομάδας Ειδικού Ενδιαφέροντος (που οργανώθηκε από τον οργανισμό W3C).

Η XML που είναι τα ακρωνύμια των λέξεων **Extensive Markup Language** αποτελεί μια περιγραφική γλώσσα δομών δεδομένων που καλούνται **XML documents** και μερικώς περιγράφει τη συμπεριφορά των προγραμμάτων που επεξεργάζεται. Η XML ουσιαστικά προέρχεται από την SGML και αποτελεί μια προσαρμοσμένη έκδοσή της με πρωτεύων σκοπό την εκτενή χρησιμοποίησή της σε κατανεμημένες εφαρμογές. Τα XML documents που αναφέρθηκαν παραπάνω αποτελούνται από μονάδες αποθήκευσης που καλούνται έννοιες (entities) οι οποίες περιέχουν data (parse ή Unparsed). Τα parsed data αποτελούν από characters που είναι είτε δεδομένα ή περιγραφικά στοιχεία (markup). Οι markup χαρακτήρες δεν είναι τίποτε άλλο από περιγραφές της διάταξης και λογικής δομής των δεδομένων που εμπεριέχει και ένα μηχανισμό ενσωμάτωσης περιορισμών (constraints).

1.2.5.1 XML δήλωση

Η πρώτη γραμμή ενός XML αρχείου

```
<?xml version="versionNumber"  
[encoding="encodingValue"]  
[standalone="yes | no"]?>
```

version e.g. "1.0"

encoding προαιρετική τιμή - προεπιλεγμένη UTF-8

standalone δηλώνει αν το XML αρχείο δεν εξαρτάται από άλλα XML αρχεία για να είναι έγκυρο ή όχι . Εξ' ορισμού η τιμή είναι yes (δηλ. ανεξάρτητο αρχείο)

1.2.5.2 Σύνταξη της xml.

Για να προχωρήσουμε στην περιγραφή της σύνταξη της γλώσσας πρέπει να δούμε τα δομικά στοιχεία της γλώσσας ώστε να παράγουμε ένα πιο ολοκληρωμένο συμπέρασμα για το τι δυνατότητες τελικά μας προσφέρει.

I) XML element: είναι το δομικό στοιχείο της και αποτελείται από ένα start tag (που περιγράφει ουσιαστικό τη σημασιολογία του δεδομένου το ίδιο το δεδομένο και ένα end tag που πλέον είναι απαραίτητο να υπάρχει.

```
<teacher_name>Ζαχαροπούλου Ιωάννα</teacher_name>
```

Σημειωτέον ότι τα tags είναι case sensitive. Επίσης ένα XML element μπορεί να περιέχει κάποια attributes της πληροφορίας

```
<teacher_name age =25>Ζαχαροπούλου Ιωάννα</teacher_name>
```

II)XML document: ένα XML document είναι ουσιαστικά ένα XML element που μπορεί και να εμπεριέχει φωλιασμένα (nested) άλλα XML elements.

```
<tomeis>  
  <tomeas>Πληροφορική</tomeas>  
  <tomeas>Επικοινωνίες</tomeas>  
</tomeis>
```

III) XML Data Island: Δεν είναι τίποτε άλλο από ένα XML document μέσα σε ένα αρχείο HTML. Η σύνταξή του είναι όπως ενός XML document μόνο που αυτό δηλώνεται και αναγνωρίζεται μέσα από την XML μέσω ενός XML id.

```
<XML ID="XMLid1">  
<customer>  
<name>Mark Hanson</name>  
<custID>81422</custID>
```

```
</customer>  
</XML>
```

IV)XML namespaces: για να αποφεύγεται η σύγκλιση με τη χρήση element ή attribute names με διπλή ερμηνεία χρησιμοποιούμε τα XML namespaces για να δηλώσουμε αποκλειστικά κάποιο XML tag.

```
<BOOKS>  
<bk:BOOK xmlns:bk="urn:BookLovers.org:BookInfo"  
xmlns:money="urn:Finance:Money">  
<bk:TITLE>A Suitable Boy</bk:TITLE>  
<bk:PRICE money:currency="US Dollar">22.95</bk:PRICE>  
</bk:BOOK>  
</BOOKS>
```

V) Document Type Definition(DTD): Είναι ένα set από κανόνες που περιγράφουν ποιά tags περιέχει το XML document , τη διάταξή τους ποιά έχουν attributes κλπ. Καθένας μπορεί να ορίσει διαφορετικά DTD (μην ξεχνάμε ότι η XML ουσιαστικά είναι ένα σετ κανόνων για να ορίσει στην ουσία ο καθένας τη δική του εκδοχή της γλώσσας). Η Microsoft έχει προτείνει μια τροποποίηση-εμπλουτισμό για το DTD το XML σχήμα το οποίο είναι πολύ πιθανό να εγκριθεί από το W3C.Προς το παρόν υποστηρίζει τόσο το XML schema όσο το και το DTD στον browser της. Ένα XML document θεωρείται *valid* αν συνοδεύεται και από το DTD ενώ καλείται *well formed* αν δεν συνοδεύεται από αυτό.

Πρέπει να δώσουμε προσοχή στα tags. Μερικοί ακόμα βασικοί κανόνες είναι:

- 1) <message> Welcome XML! Το κάθε tag πρέπει να κλείνει.
- 2) Τα tags είναι case sensitive π.χ.

```
<message> είναι διαφορετικό από το <Message>
```

- 3) Τα tags πρέπει να είναι εμφωλευμένα σωστά π.χ.

```
<myMessage> <message> Welcome XML </message></myMessage>
```

4) Άδεια tags

```
<Name/>
```

1.2.5.3 Πλεονεκτήματα και προοπτικές.

Οι βελτιώσεις που μπορεί να επιφέρει η XML είναι πράγματι εντυπωσιακές.

- Πιο αποτελεσματική αναζήτηση (search) λόγω της καλύτερης δόμησης της πληροφορίας θα μπορούμε να βρίσκουμε πιο εύκολα αυτό που θέλουμε π.χ web pages σχετικά με τον φοιτητή Δρουμπάκη και όχι για παράδειγμα το σχεδιαστή μόδας Δρουμπάκη. Αυτό πρακτικά δεν μπορούσε να γίνει παλιότερα, γιατί δεν είχαμε επαρκή δεδομένα για το database schema και η διαφοροποίηση στην οργάνωσή τους καθιστούσε κάτι τέτοιο ανέφικτο.
- Πιο αποτελεσματικές εφαρμογές αφού το layout πλέον είναι διαχωρίσιμο από τη δόμηση της πληροφορίας γεγονός που μας επιτρέπει να διαχειριστούμε τα δεδομένα πιο αποτελεσματικά.
- Μπορούμε να ανακτήσουμε εύκολα δεδομένα από πολλαπλές πηγές αλλά και διαφορετικές εφαρμογές και να τα χειριστούμε κατά το δοκούν.
- Διαχείριση και μετασχηματισμός δεδομένων ακόμα και σε επίπεδο client.
- Πολλαπλές εκδοχές και απεικονίσεις των δεδομένων, κάτι που δεν μπορούσε να γίνει φυσικά με την HTML.

1.2.5.4 Μειονεκτήματα της XML.

Ενώ η XML καλύπτει απόλυτα την ανταλλαγή δεδομένων από μια εφαρμογή σ' έναν browser ώστε να είναι κατανοητή από τον άνθρωπο, εν τούτοις δεν φαίνεται να προάγει την ανταλλαγή δεδομένων από μια εφαρμογή σε μια άλλη. Παρά το γεγονός ότι είναι θεωρητικά εφικτή η άμεση ανταλλαγή μεταξύ εφαρμογών, πρακτικά αυτές θα πρέπει να υποστηρίζουν την ίδια "διάλεκτο" XML ώστε και οι δύο εφαρμογές να κατανοούν τα δεδομένα. Δυστυχώς όμως η δημιουργία αυτών των διαλέκτων ούτε

ελέγχεται ούτε υπόκειται σε κάποιους κανόνες/standards, με αποτέλεσμα να οδηγεί στη δημιουργία πολλών διαφορετικών διαλέκτων για την υποστήριξη των ίδιων εφαρμογών.

1.2.5.5 Προσχεδιασμένοι στόχοι της XML.

Οι προσχεδιασμένοι στόχοι της XML είναι:

1. Η XML πρέπει να είναι εύχρηστη στο Internet.
2. Η XML πρέπει να υποστηρίζει μεγάλη ποικιλία από εφαρμογές.
3. Η XML πρέπει να είναι συμβατή με την SGML.
4. Θα είναι εύκολο να γράφονται προγράμματα που επεξεργάζονται XML έγγραφα.
5. Ο αριθμός των προαιρετικών χαρακτηριστικών στην XML θα είναι όσο το δυνατόν πιο μικρός, ιδανικό επίπεδο το μηδέν.
6. Τα XML έγγραφα θα πρέπει να είναι ευανάγνωστα.
7. Ο σχεδιασμός XML θα πρέπει να προετοιμάζεται γρήγορα.
8. Ο σχεδιασμός XML θα πρέπει να είναι τυπικός και περιεκτικός.
9. Τα XML έγγραφα θα πρέπει να δημιουργούνται εύκολα.
10. Η περιεκτικότητα στον XML συμβολισμό είναι μικρής σημασίας.

1.2.5.6 Elements ή Στοιχεία

Τα στοιχεία (elements) είναι ετικέτες, όπως και στην HTML, και περιέχουν τιμές. Επιπλέον τα elements είναι δομημένα σαν δένδρο. Σαν αποτέλεσμα, έχουμε τα στοιχεία οργανωμένα ιεραρχικά με ένα στοιχείο-πατέρα και στοιχεία-παιδιά. Τα στοιχεία-παιδιά μπορούν να περιέχουν και αυτά άλλα στοιχεία-παιδιά και ούτω καθεξής.

Τα στοιχεία έχουν κάποιους κανόνες.

- Όλα τα στοιχεία πρέπει να έχουν ετικέτα κλεισίματος αντίθετα με την HTML όπου υπάρχουν και ετικέτες που δε χρειάζονται κλείσιμο όπως για παράδειγμα η `
`.
- Οι ετικέτες των στοιχείων είναι case sensitive, δηλαδή υπάρχει διαχωρισμός μεταξύ κεφαλαίων και πεζών και τα ονόματά τους υπακούουν σε κανόνες ονοματολογίας.
- Τα στοιχεία πρέπει να είναι τοποθετημένα σωστά, αντίθετα με την HTML.

HTML : `<i>This text is bold and italic</i>`

XML : `<i>This text is bold and italic</i>`

- Τα έγγραφα της XML πρέπει να έχουν ακριβώς ένα αρχικό στοιχείο (root element).

1.2.5.7 Attributes ή Ιδιότητες

Οι ιδιότητες (attributes) μας βοηθούν να δώσουμε περισσότερο νόημα και να περιγράψουμε τα στοιχεία μας πιο αποτελεσματικά και με σαφήνεια. Όπως και τα στοιχεία έτσι και οι ιδιότητες έχουν κάποιους κανόνες :

Οι τιμές των ιδιοτήτων πρέπει να είναι εσωκλείονται σε " " ή σε ' '.

Τα ονόματα των ιδιοτήτων ακολουθούν τους ίδιους κανόνες με αυτά των ετικετών.

1.2.5.8 Η διαφορά μεταξύ XML και HTML.

Σε αντίθεση με την SGML ή την XML, η HTML είναι μια γλώσσα markup που περιέχει μια περιορισμένη ομάδα στοιχείων και χαρακτηριστικών. Η HTML έχει ένα περιορισμένο ρεπερτόριο "δομικών στοιχείων" όπως επικεφαλίδες, λίστες, links, κλπ. Αυτή η απλή φιλοσοφία σχεδίασης έδωσε τη δυνατότητα στην HTML να είναι εύκολα κατανοήσιμη και εφαρμόσιμη, οδηγώντας έτσι στη γρήγορη υιοθέτησή της. Η συνεχής όμως ανάπτυξη του Web και οι συνεχώς αυξανόμενες ανάγκες των χρηστών, κατέστησαν εμφανή πλέον τα "μειονεκτήματα" της HTML. Π.χ. είναι δύσκολο για τις μηχανές αναζήτησης να εκμεταλλευθούν τις πληροφορίες που τυχόν υπάρχουν σε μια ιστοσελίδα από τη στιγμή που ο τρόπος αναγραφής τους δεν είναι τυποποιημένος. Η XML μπορεί να επιλύσει αυτά τα προβλήματα και να παράσχει στο Web μια πολύ μεγαλύτερη ικανότητα για ηλεκτρονικό εμπόριο. Η XML βοηθά στην κωδικοποίηση πληροφοριών σε μια δομή με νόημα και σημειολογία μέσω ενός απλού "λεξιλογίου" που είναι εύκολα αναγνώσιμο από τον άνθρωπο και επεξεργάσιμο από τους Η/Υ.

1.2.6 Action script 3

1.2.6.1 Classes, objects , packages

Τα objects και τα classes είναι το ίδιο πράγμα βασικά, το object είναι η θεμέλιος λίθος του flash, το οποίο είναι ένα άδειο αντικείμενο που μπορείς να το διαπλάσεις όπως θέλεις, όλα τα αντικείμενα μέσα στο flash είναι objects (movieclips, bitmaps, sounds, code κ.λ.π.).

Οι classes είναι ένα βοήθημα έτσι ώστε να μπορεί κάποιος να δημιουργεί προσωπικά αντικείμενα πιο εύκολα και λογικά. Δηλαδή, ο χρήστης κατασκευάζει ένα custom αντικείμενο που περιέχει τις δικές του εντολές και μεταβλητές.

Για παράδειγμα, ζητείται η κατασκευή ενός αντικειμένου που αντιπροσωπεύει έναν ηρώα σε ένα game. Μια class θα μπορούσε να πραγματοποιηθεί κάπως έτσι (με πολύ χαλαρό κώδικα):

```
class hero {  
    function walk() {}  
    function shoot() {}  
    function die() {}  
}
```

Στη συνέχεια, μέσα στο κυρίως πρόγραμμα, κατασκευάζεται ένα αντικείμενο τύπου hero το οποίο μπορεί να χρησιμοποιηθεί όπως χρησιμοποιείται ένα movieclip.

π.χ.

```
var myHero = new hero()  
myHero.shoot()
```

Μια class θα πρέπει να βρίσκεται στη σκέψη του χρήστη ως ένα πραγματικό αντικείμενο, π.χ. ο ήρωας, και μετά θα πρέπει να ακολουθούν και όλες οι σχετικές

ιδιότητες, π.χ. τι κάνει ο ήρωας, και οι σχετικές πληροφορίες που θέλουμε να περιέχει, π.χ. πληροφορίες για το ύψος του ή για τις ικανότητές του, κλπ. Έτσι δομούμε ένα λογικό αντικείμενο που για εμάς αντιπροσωπεύει κάτι αληθινό, μαζί με όλες τις λειτουργίες που θέλουμε να εκτελεί αυτό το αντικείμενο.

Packages - Τα packages χρησιμοποιούνται για να ομαδοποιούν τις classes , και για να είναι πιο εύκολο να διαχειριστούμε πολλές classes.

π.χ. Μια class με μαθηματικές λειτουργίες ανήκει στο πακέτο myclasses.math .

```
package myclasses.math {  
    class myclass {}  
}
```

Τα αρχεία των classes πρέπει να ακολουθούν την δομή φακέλων που ορίζει ο χρήστης στο package, δηλαδή εάν είχε το προηγούμενο class πρέπει να είναι μέσα σε έναν φάκελο "myclasses" που περιέχει έναν φάκελο "math" που περιέχει ένα αρχείο myclass.as.

Εάν είχες ένα αρχείο fla και ένα class στο ίδιο επίπεδο αφήνεις το package κενό.

π.χ.

```
package {  
    class myclass {}  
}
```

1.2.6.2 Κλάση object().

Κλάση object() είναι μια κλάση ρίζα της ActionScript , αυτό σημαίνει ότι είναι η υψηλότερη κλάση στην ιεραρχία των κλάσεων. Χρησιμοποιώντας τις ιδιότητες και τις μεθόδους που είναι διαθέσιμες σε αυτή την κλάση, μπορεί κάποιος να επηρεάσει και να τροποποιήσει άλλες κλάσεις. Επίσης, είναι βολική για να δημιουργούνται στιγμιότυπα αντικειμένων που περιέχουν πληροφορίες για τον τρέχοντα χρήστη ή στιγμιότυπα που παρακολουθούν μέρη από σχετικά δεδομένα.

1.2.6.3 Προσδιοριστές πρόσβασης

Καθορίζουν τα δικαιώματα πρόσβασης στα αντικείμενα και τις μεθόδους της κλάσης.

public: ορατά από οποιασδήποτε κλάση

private: ορατά μόνο από την κλάση στην οποία ανήκουν

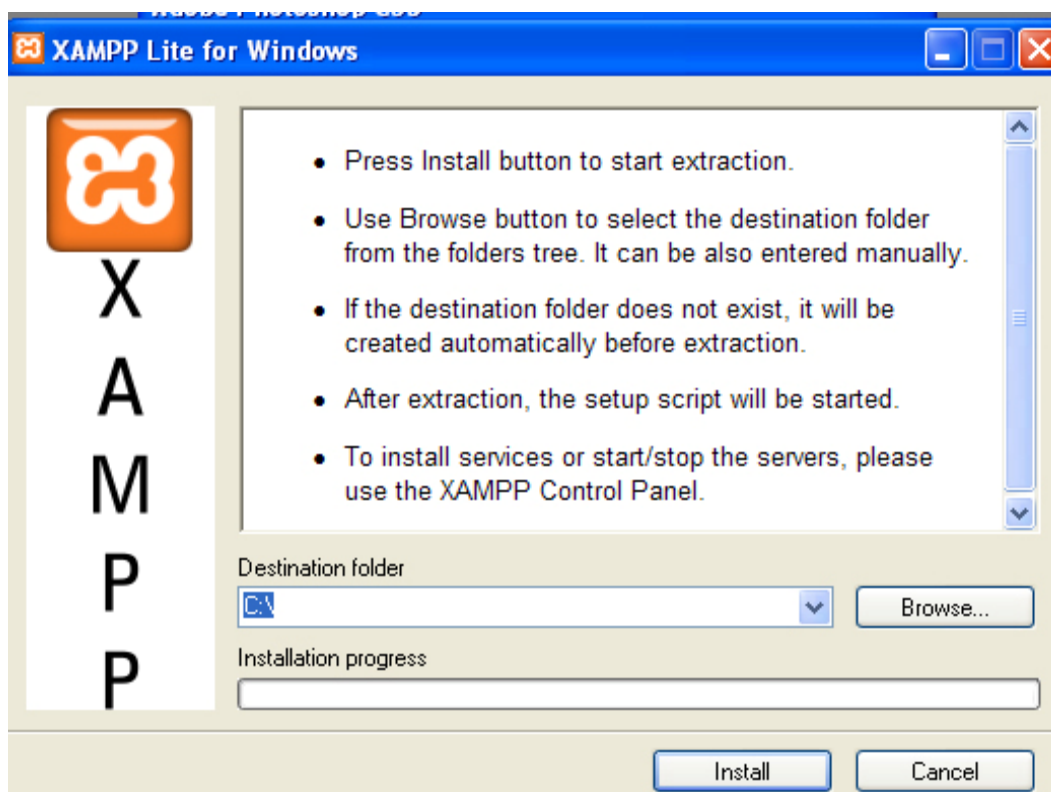
protected: ορατά από την κλάση στην οποία ανήκουν και από υποκλάσεις της

Χωρίς δήλωση: ορατά από κλάσεις που ανήκουν στο ίδιο πακέτο (package)

1.3 ΤΕΧΝΟΛΟΓΙΑ.

1.3.1 Εγκατάσταση Xampp

Έχω ‘κατεβάσει’ το αρχείο xamppLite.exe από τον ιστοχώρο Apache friends <http://www.apachefriends.org/en/xampp-windows.html> . Είναι η ελαφριά έκδοση και συνιστάται για ‘γρήγορη’ εργασία χρησιμοποιώντας μόνο PHP και MySQL (λείπουν μερικοί server και εργαλεία, όπως τα Mercury και FileZilla FTP). Ξεκινάμε το .exe αρχείο και επιλέγουμε μονοπάτι/κατάλογο για εγκατάσταση, κι αρχίζει το setup script. Τα βήματα φαίνονται παρακάτω με εικόνες.



1) Διαλέγουμε το μονοπάτι/κατάλογο για εγκατάσταση και πατάμε install.

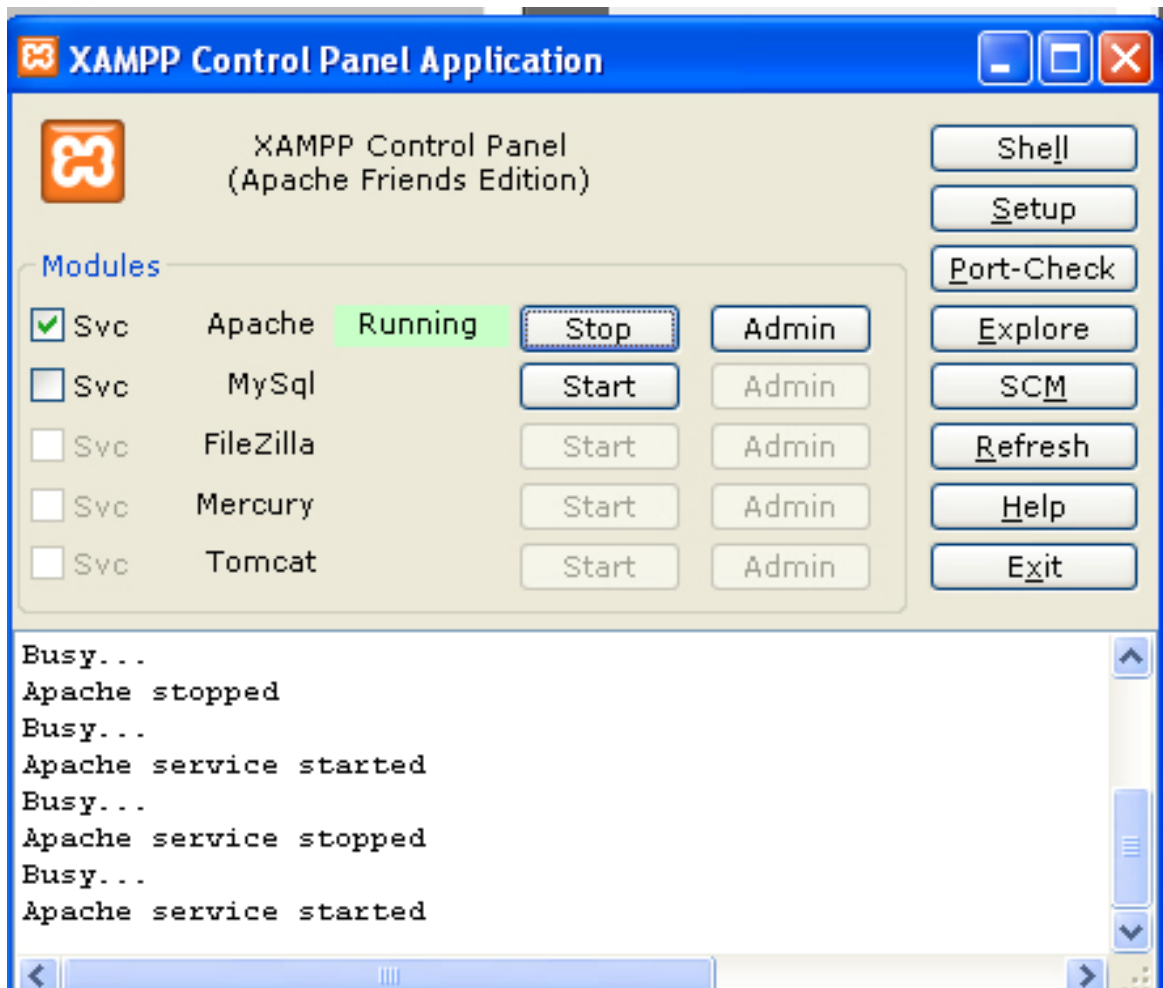
```
C:\WINDOWS\system32\cmd.exe
#####
# XAMPP Lite 1.7.3 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann <FreeBSD License>         #
#-----#
# Authors: Carsten Wiedmann <carsten_stt@tgm.de>           #
#           Kay Vogelgesang <kvo@apachefriends.org>         #
#####
Should I add shortcuts to the startmenu/desktop? <y/n>: y
```

2) Όταν ολοκληρωθεί η εγκατάσταση μας «ρωτάει» αν θέλουμε να φτιάξει συντόμευση στην επιφάνεια εργασίας αυτή η επιλογή είναι στην κρίση του κάθε χρήστη.

```
C:\WINDOWS\system32\cmd.exe
#####
# XAMPP Lite 1.7.3 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann <FreeBSD License>         #
#-----#
# Authors: Carsten Wiedmann <carsten_stt@tgm.de>           #
#           Kay Vogelgesang <kvo@apachefriends.org>         #
#####
1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: C:\xampplite)
3. disable HTTPS <SSL>
4. disable Server Side Includes <SSI>
5. enable IPv4 only (current: IPv4/6 <auto>)
x Exit
Please choose <1-5/x>: 1
```

Επιλέγουμε το 1 και πατάμε enter.

3) Ανοίγει το παράθυρο XAMPP CONTROL PANTEL APPLICATION πατάμε stop.



4) Μέσα στον φάκελο «πτυχιακή» υπάρχει ένας υπό φάκελος rhr κάνουμε copy/αντιγραφή το αρχείο rhr.ini που έχει μέσα. Μπαίνουμε στον φάκελο που είναι αποθηκευμένο ο xamp Lite .

5) Διαδρομή c:\xamppLite\rhr και κάνουμε paste/επικόλληση το αρχείο rhr στην ερώτηση 'αντικατάσταση' πατάμε ΝΑΙ.

6) Στο control panel πατάμε start (είχαμε πατήσει stop).

7) Περνούμε τον φάκελο «πτυχιακή» copy/αντιγραφή και τον κάνουμε paste/επικόλληση μέσα στο C:\xampplite\htdocs\.

8) Ανοίγουμε τον Web browser φυλλομετρητής και γράφουμε <http://localhost/delos/index.html>

Κεφάλαιο 2

2.1 Διεπάφη Λογισμικού

Σε αυτό το κεφάλαιο, θα γίνει μια αναλυτική παρουσίαση του τρόπου λειτουργίας της εφαρμογής. Θα παρουσιαστεί ο τρόπος επεξεργασίας των XML δυναμικών, αλλά και το πώς θα προβάλλονται τα δεδομένα.

Περίληπτικά, η εφαρμογή χωρίζεται σε δυο μέρη: αυτό του web site (ιστοχώρο) και αυτό του flash.

Στο πρώτο μέρος, ο χρήστης μπορεί να δημιουργήσει και να επεξεργαστεί ένα XML. Το XML αυτό περιέχει τα μαθήματα του τμήματος.

Πιο αναλυτικά, όπως θα δούμε παρακάτω, ο χρήστης θα μπορεί να δημιουργεί το XML (π.χ. 2009-10a.xml για το αέρινο εξάμηνο), και στην συνέχεια, θα μπορεί να προσθέτει τα μαθήματα μέσα σε αυτό. Επίσης, αν χρειαστεί, θα μπορεί να επεξεργαστεί τα μαθήματα που έχει ήδη προσθέσει, όπως το να τα διαγράψει, αλλά και το να κάνει τις παρακάτω αλλαγές:

- 1) αλλαγή ώρας διεξαγωγής του μαθήματος,
- 2) αλλαγή μέρας που γίνεται το μάθημα,
- 3) αλλαγή αίθουσας που θα γίνει το μάθημα και
- 4) αλλαγή ομάδας του μαθήματος.



Σχήμα: Η διεπαφή με διάγραμμα κλάσεων.

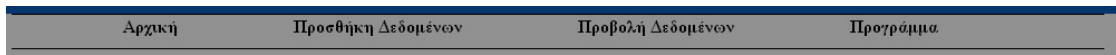
Στο δεύτερο κομμάτι ο χρήστης έχει την δυνατότητα να διαλέξει ποιο XML θέλει να δει, θα πρέπει να ανάλογα με τις επιλογές που θα κάνει στα combobox, θα μπορεί να δει τα μαθήματα (χωρίς να μπορεί να τα επεξεργαστεί), θα γίνεται δυναμική εναλλαγή των μαθημάτων στον καμβά.

Για την υλοποίηση της εφαρμογής, χρησιμοποιήθηκε η html που είναι ο βασικός κορμός για την προβολή της. Για την δημιουργία αλλά και για την επεξεργασία του XML χρησιμοποιήθηκε η php. Για το template (εμφάνιση ιστοσελίδας) χρησιμοποιήθηκε η CSS. Τέλος, για την δυναμική προβολή των μαθημάτων στο καμβά, χρησιμοποιήθηκε η ActionScript 3.0.

Στην συνέχεια, αναλύονται λεπτομερώς οι δυνατότητες της εφαρμογής.

Όλες οι σελίδες έχουν μία μπάρα με τα links (Εικόνα 1).

- αρχική,
- εισαγωγή δεδομένων,
- προσθήκη δεδομένων,
- πρόγραμμα.



Εικόνα 1

Με κάθε επιλογή που κάνουμε, βλέπουμε και την αντίστοιχη σελίδα.

2.1.1 Αρχική σελίδα.

Η αρχική σελίδα είναι η `index.html` , ένα όνομα απαραίτητο, σε περίπτωση που θέλουμε να ανεβάσουμε την ιστοσελίδα σε κάποιο server, ώστε να έχουμε ένα αρχείο `index.html`, ώστε να το πάρει ως αρχή. Το κείμενο της σελίδας αυτής αποτελεί την περίληψη της πτυχιακής.

2.1.2 Εισαγωγή δεδομένων.

Στο Link «εισαγωγή δεδομένων» έχουμε την δυνατότητα να δημιουργήσουμε το XML και να κάνουμε προσθήκη μαθημάτων σε αυτό. Η προσθήκη των μαθημάτων είναι μία πολύ εύκολη και απλή διαδικασία, το μόνο που χρειάζεται είναι να γνωρίζουμε τα στοιχεία του μαθήματος που θέλουμε να προσθέσουμε. Η εισαγωγή των πληροφοριών γίνεται στην παρακάτω φόρμα (Εικόνα 2).

A form for adding course data. It includes the following fields and controls:

- Έτος: dropdown menu
- Εξάμηνο: dropdown menu
- Μέρα: dropdown menu (selected: Δευτέρα)
- Επιλογή μαθήματος: dropdown menu (selected: —1 ΕΞΑΜΗΝΟ—, τομέας :)
- Κωδικός μαθήματος: text input
- Μαθήμα: text input
- Τομέας: text input
- Εργαστήριο: radio button
- Θεωρία: radio button
- Ομάδα: text input
- Αίθουσες εργαστηρίων: dropdown menu
- Αίθουσες θεωρίας: dropdown menu
- Όνομα καθηγήτη: dropdown menu
- Έναρξη μαθήματος: dropdown menu (selected: 8)
- Λήξη μαθήματος: dropdown menu (selected: 10)
- επιβεβαίωση: button
- καθαρισμός φόρμας: button

Εικόνα 2

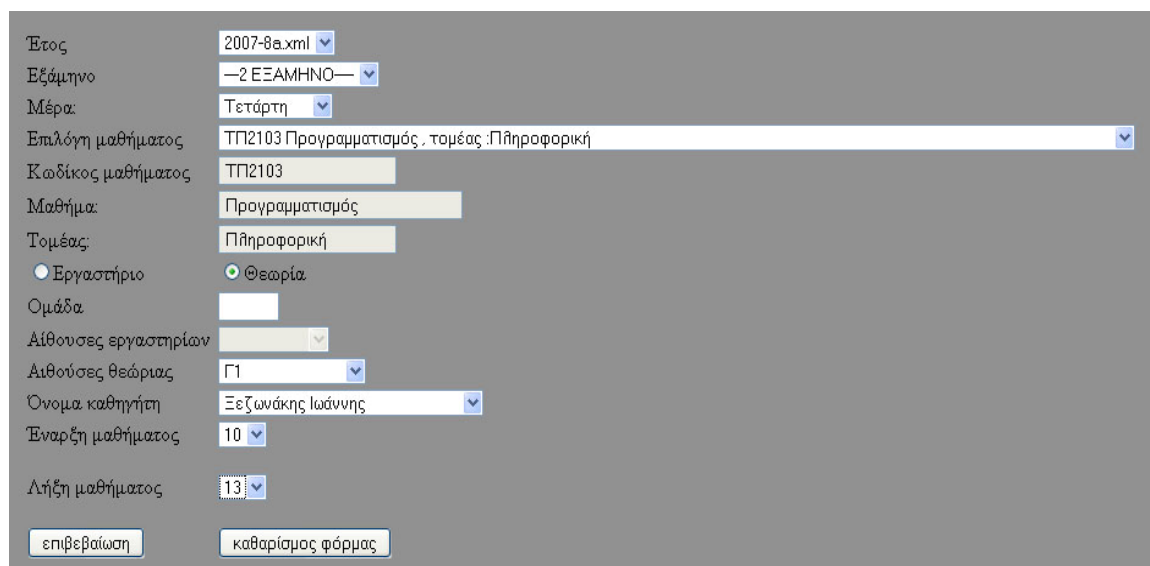
- «Έτος»: επιλέγουμε το εξάμηνο που θέλουμε να κάνουμε προσθήκη του μαθήματος. Το combobox που είναι δίπλα στο «έτος» ‘διαβάζει’ το αρχείο «\xml_info\etoi.xml». Στο αρχείο αυτό έχουν αποθηκευθεί τα εξάμηνα (αέρινα , χειμερινά) της σχολής· αυτά τα εξάμηνα μας εμφανίζονται όταν επιλέξουμε το combobox.
- «Εξάμηνο»: επιλέγουμε το εξάμηνο που γίνεται το μάθημα.
- «Μέρα»: επιλέγουμε την μέρα που γίνεται το μάθημα.
- «Επιλογή μαθήματος»: το combobox αυτό διαβάζει ένα αρχείο XML το οποίο περιέχει όλα τα μαθήματα του τμήματος.
- Όσον αφορά στα τρία πεδία «κωδικός μαθήματος», «μάθημα», « τομέας» και τα δύο radio button «εργαστήριο» και «θεωρία» δεν χρειάζεται να επιλέξει κάτι ο χρήστης, γιατί γεμίζουν δυναμικά, ανάλογα με την επιλογή που κάνουμε στο πεδίο «επιλογή μαθήματος».

ΠΡΟΣΟΧΗ! Αν ο χρήστης επιλέξει «εργαστήριο» ή «θεωρία» τότε θα εμφανιστεί ένα προειδοποιητικό παράθυρο, και θα πρέπει να επιλέξουμε ξανά μάθημα.

- «Ομάδα»: εδώ ο χρήστης έχει την δυνατότητα να βάλει την ομάδα του μαθήματος (αυτό γίνεται στην περίπτωση που έχουμε παραπάνω από ένα τμήμα).
- «Αίθουσα θεωρίας» και «αίθουσα εργαστηρίου»: αυτά τα δύο combobox πότε δεν θα είναι και τα δύο ενεργοποιημένα. Αυτό έχει διαμορφωθεί με αυτό τον τρόπο, ώστε το ένα να περιέχει τις αίθουσες που διεξάγονται οι θεωρίες και το άλλο τις αίθουσες που διεξάγονται τα εργαστηριακά μαθήματα.

- «Όνομα καθηγητή»: σε αυτή την περίπτωση, το combobox διαβάζει ένα XML που περιέχει τα ονόματα των καθηγητών που κάνουν μάθημα στην σχολή.
- «Έναρξη μαθήματος» και «Λήξη μαθήματος»: επιλέγουμε την ώρα που αρχίζει και τελειώνει το μάθημα. Εδώ έχω βάλει έλεγχο ώστε τα μαθήματα να είναι είτε 2ώρα είτε 3ώρα.

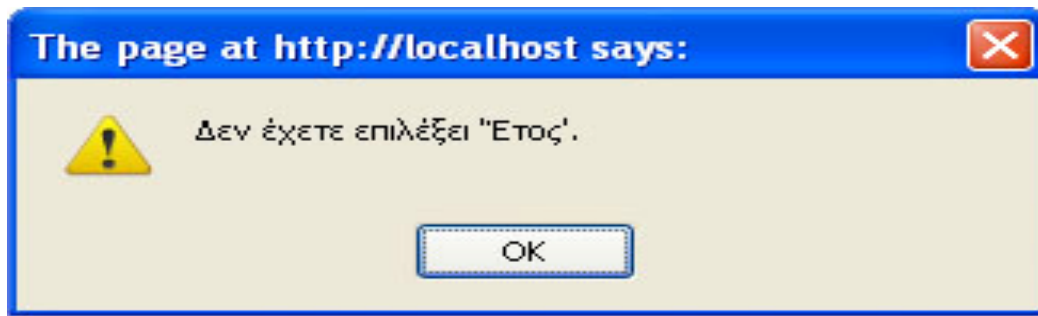
Στην παρακάτω εικόνα (Εικόνα 3) φαίνονται οι επιλογές που μόλις αναφέρθηκαν παραπάνω.



Έτος	2007-08.xml
Εξάμηνο	—2 ΕΞΑΜΗΝΟ—
Μέρα	Τετάρτη
Επιλογή μαθήματος	ΤΠ2103 Προγραμματισμός, τομέας :Πληροφορική
Κωδικός μαθήματος	ΤΠ2103
Μαθήμα:	Προγραμματισμός
Τομέας:	Πληροφορική
<input type="radio"/> Εργαστήριο	<input checked="" type="radio"/> Θεωρία
Ομάδα	
Αίθουσες εργαστηρίων	
Αιθούσες θεωρίας	Γ1
Όνομα καθηγητή	Ξεζωνάκης Ιωάννης
Έναρξη μαθήματος	10
Λήξη μαθήματος	13

Εικόνα 3

Όλα τα παραπάνω πεδία θα πρέπει να είναι συμπληρωμένα, όταν θα πατήσουμε το κουμπί «επιβεβαίωση» · αν έχουμε ξεχάσει να συμπληρώσουμε κάποιο πεδίο, τότε εμφανίζεται ένα μήνυμα με το λάθος που έχουμε κάνει.



Εικόνα 3

Όταν ολοκληρωθεί η εισαγωγή των πληροφοριών για το μάθημα, επιλέγουμε το κουμπί «επιβεβαίωση».

2.1.3 Επιβεβαίωση

Σε αυτή την ιστοσελίδα εμφανίζονται οι πληροφορίες του μαθήματος που έχουμε εισάγει στην προηγούμενη σελίδα (εισαγωγή δεδομένων).

Εδώ μπορεί ο χρήστης να δει αν έχει παραλείψει ή έχει κάνει λάθος τις πληροφορίες για το μάθημα, τότε μπορεί να πατήσει το κουμπί «Διόρθωση» και να επιστρέψει στην προηγούμενη σελίδα για να κάνει τις αλλαγές που χρειάζεται ή να πατήσει το κουμπί «Επιβεβαίωση» ώστε να δημιουργηθεί το μάθημα αυτό.

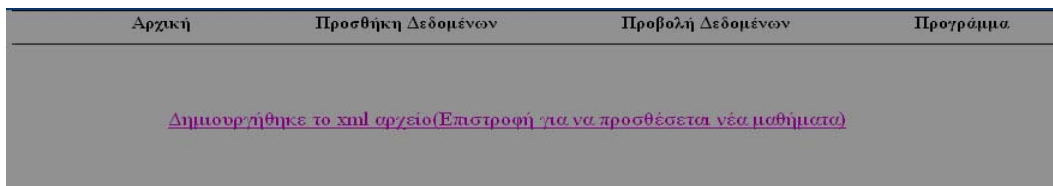
Έχετε επιλέξει το :	---2 ΕΞΑΜΗΝΟ----
Μέρα της βδομάδας :	Τετάρτη
Το μάθημα είναι :	Προγραμματισμός
Όνομα καθηγητή :	Ξεζωνάκης Ιωάννης
Έχετε επιλέξει την αθούσα :	Γ1
Ωρα έναρξης μαθήματος :	10
Ωρα λήξης μαθήματος :	13
Ομάδα :	
<input type="button" value="Επιβεβαίωση"/>	<input type="button" value="Διόρθωση"/>

Εικόνα 4

Όταν πατήσουμε το κουμπί «Επιβεβαίωση» εμφανίζεται μια νέα σελίδα που ενημερώνει τον χρήστη αν έγινε η αποθήκευση του μαθήματος.

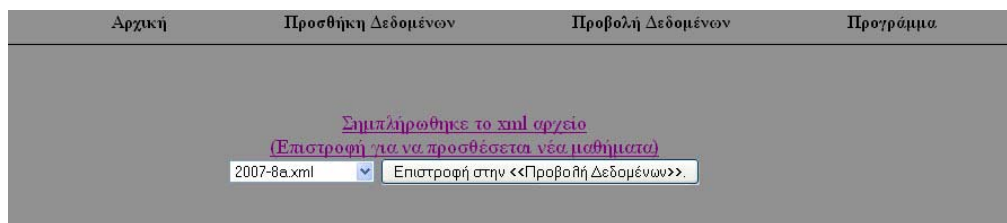
Εδώ εμφανίζονται τα παρακάτω μηνύματα:

A) Το μήνυμα «*Δημιουργήθηκε το XML αρχείο(Επιστροφή για να προσθέσετε νέα μαθήματα)*» εμφανίζεται αν δεν υπάρχει το XML.



Εικόνα 5

Αν υπάρχει ήδη XML, εμφανίζονται δύο επιλογές (Εικόνα 2.4): η πρώτη είναι «*Συμπληρώθηκε το XML αρχείο (Επιστροφή για να προσθέσετε νέα μαθήματα)*», ενώ η δεύτερη είναι «*Επιστροφή στην προβολή των δεδομένων.*» Αν επιλεγεί η δεύτερη, «*Επιστροφή στην προβολή των δεδομένων*», ο χρήστης θα μπορέσει να δει συγκεντρωτικά τα μαθήματα που έχει το συγκεκριμένο XML ή κάποιο άλλο, διαλέγοντάς το από την λίστα που εμφανίζεται.

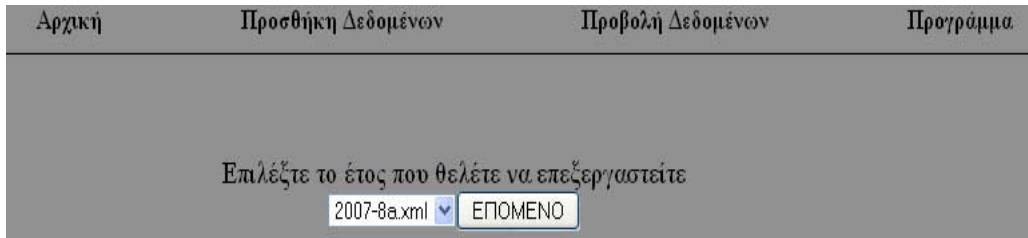


Εικόνα 6

2.1.4 Προβολή Δεδομένων.

Ένα πολύ σημαντικό μέρος της εργασίας αυτής είναι η ικανότητα του χρήστη να δει συγκεντρωτικά τα μαθήματα που έχει εισάγει, αλλά και να μπορεί να τα επεξεργαστεί. Αυτό μπορεί να γίνει στην σελίδα «*Προβολή Δεδομένων*».

Στην σελίδα «Προβολή Δεδομένων» δίνεται η δυνατότητα στον χρήστη να επιλέξει ποιο έτος θέλει να δει, από τα έτη που έχουμε δημιουργήσει και αποθηκεύσει στον φάκελο «xmls/». (Εικόνα 7)



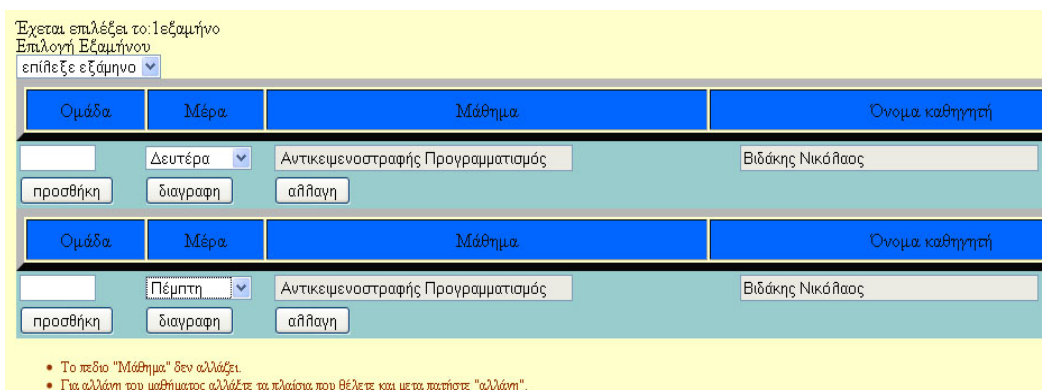
Εικόνα 7

Όταν επιλέξουμε ποιο έτος θέλουμε να δούμε πατάμε το κουμπί «επόμενο» , πρώτα εμφανίζονται τα μαθήματα που έχει το πρώτο εξάμηνο. Για να εμφανιστούν και τα υπόλοιπα εξάμηνα θα πρέπει να επιλεγεί και το αντίστοιχο εξάμηνο από το combobox.

Επίσης, σε αυτή την σελίδα ο χρήστης έχει την δυνατότητα :

- 1) Να κάνει προσθήκη ενός νέου μαθήματος με τις ίδιες πληροφορίες ενός υπάρχοντος μαθήματος.
- 2) Να διαγράψει κάποιο μάθημα.
- 3) Να αλλάξει κάποιο μάθημα.

Αυτά οι τρεις δυνατότητες αναλύονται με λεπτομέρειες παρακάτω (Εικόνα 3).



- Το πεδίο "Μάθημα" δεν αλλάζει.
- Για αλλαγή του μαθήματος αλλάξτε τα πλαίσια που θέλετε και μετά πατήστε "αίτηση".

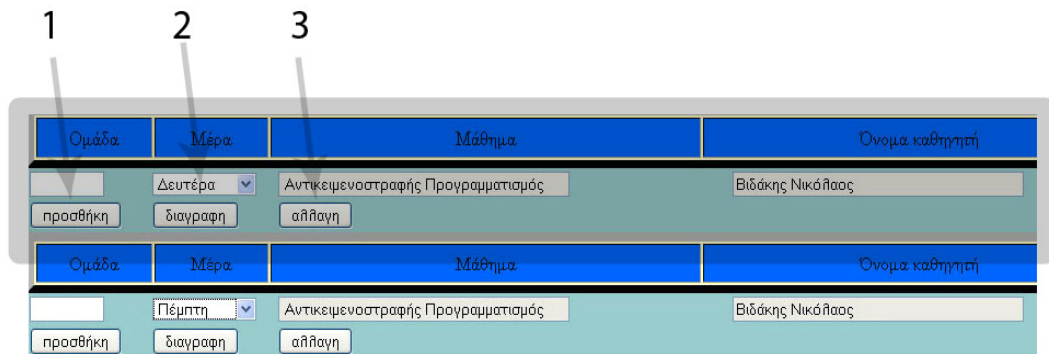
Εικόνα 3

2.1.5 Προσθήκη , Διαγραφή , Αλλαγή μαθήματος.

Όπως φαίνεται στη ‘εικόνα 4’ υπάρχουν 3 διαφορετικά κουμπιά ,

- 1) «προσθήκη» ,
- 2) «διαγραφή»
- 3) «αλλαγή μαθήματος».

Σε κάθε μάθημα υπάρχει μια ομάδα από αυτά τα κουμπιά, που μας επιτρέπει να επεξεργαζόμαστε το συγκεκριμένο μάθημα.



Εικόνα 4

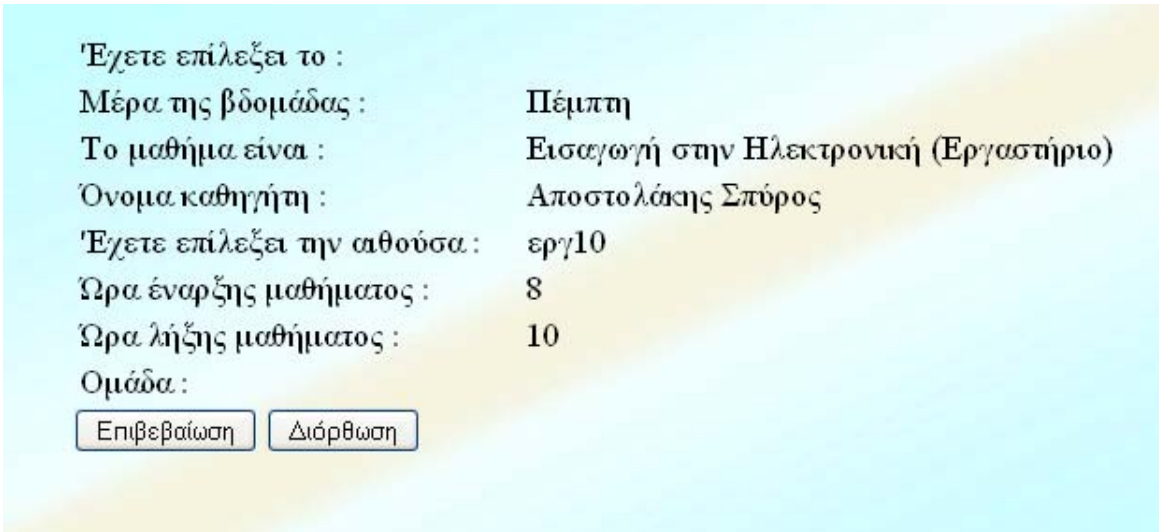
Τα τρία αυτά κουμπιά έχουν διαφορετικούς ελέγχους για την σωστή επεξεργασία του xml μας, ο κώδικας αυτός αναφέρεται στις επόμενες παραγράφους.

2.1.5 .1 Προσθήκη.

Εδώ, μπορεί ο χρήστης να κάνει προσθήκη ένα νέο μάθημα με ακριβώς τις ίδιες πληροφορίες με ένα υπάρχον μάθημα. Αυτό γίνεται πατώντας το κουμπί «προσθήκη» στο αντίστοιχο μάθημα που θέλει να προσθέσει.

Όταν επιλέξουμε το κουμπί «προσθήκη», ‘ανοίγει’ μία νέα σελίδα όπου προβάλλονται οι πληροφορίες του μαθήματος (Εικόνα 4). Αν είναι σωστές τότε πατάμε

«επιβεβαίωση», ενώ αντίθετα, αν θέλουμε να κάνουμε αλλαγές, πατάμε «διόρθωση» και επιστρέφουμε στην αρχική θέση.



Έχετε επιλέξει το :	
Μέρα της βδομάδας :	Πέμπτη
Το μαθήμα είναι :	Εισαγωγή στην Ηλεκτρονική (Εργαστήριο)
Όνομα καθηγητή :	Αποστολάκης Σπύρος
Έχετε επιλέξει την αθούσα :	εργ10
Ωρα έναρξης μαθήματος :	8
Ωρα λήξης μαθήματος :	10
Ομάδα :	

Εικόνα 5

ΠΡΟΣΟΧΗ! Αν θέλει ο χρήστης να κάνει αλλαγή τότε θα πρέπει να αλλάξει τις πληροφορίες του μαθήματος που θέλει και μετά να πατήσει το κουμπί «αλλαγή» και όχι «προσθήκη».

2.1.5 .2 Διαγραφή

Για να μπορέσει να διαγράψει ο χρήστης κάποιο μάθημα, το μόνο που πρέπει να κάνει είναι να πατήσει το κουμπί «διαγραφή» που βρίσκεται στο μάθημα που θέλει. Όταν πατηθεί το κουμπί «διαγραφή», εμφανίζεται ένα προειδοποιητικό μήνυμα για το αν θέλουμε να διαγράψουμε το μάθημα ή όχι. Αυτό προνοήθηκε για την περίπτωση που πατηθεί εσφαλμένα το κουμπί «διαγραφή».

Η εφαρμογή μας ενημερώνει ότι πραγματοποιήθηκε η διαγραφή του μαθήματος (Εικόνα 5).

Το μάθημα « Εισαγωγή στην Πληροφορική » με κωδικό « ΤΠ1003» που διεξάγεται στην αίθουσα «Γ4» και μέρα «Παρασκευή» διαγράφηκε επιτυχώς

-> 2009-10x.xml<-

Εικόνα 6

2.1.5.3 Αλλαγή.

Εδώ ο χρήστης μπορεί να αλλάξει τις πληροφορίες των μαθημάτων.

Για να αλλάξει τις πληροφορίες ενός μαθήματος, πρέπει να βάλει τα νέα δεδομένα που θέλει στις αντίστοιχες θέσεις και να πατήσει το κουμπί «προσθήκη».

Τα δεδομένα που μπορούμε να αλλάξουμε είναι :

1. Ομάδα.
2. Μέρα.
3. Αίθουσα.
4. Ώρα έναρξη και λήξη του μαθήματος.

2.1.6 Προβολή δεδομένων στο Flash.

Παραπάνω, αναφερθήκαμε στον τρόπο επεξεργασίας των μαθημάτων. Στη συνέχεια, στο δεύτερο μέρος, θα αναλυθούν οι λειτουργίες του flash.

Στο flash δεν έχει ο χρήστης την δυνατότητα να επεξεργαστεί τα μαθήματα. Η μόνη δυνατότητα που έχει, είναι να τα δει συνολικά ανάλογα τις επιλογές που θα κάνει από τα combobox. Οι επιλογές αυτές είναι:

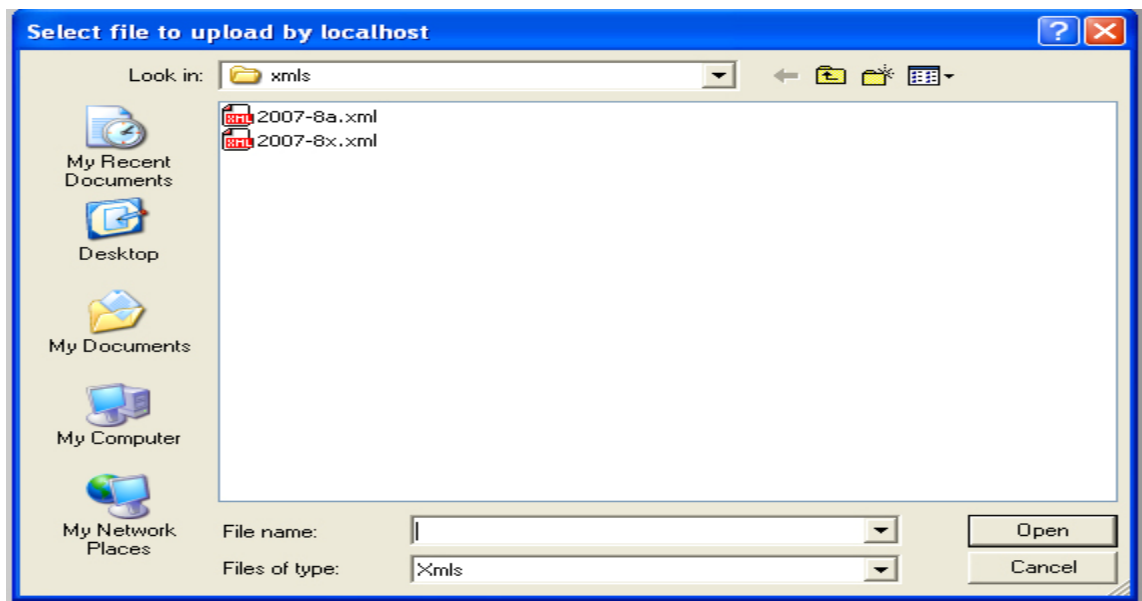
- 1) Ανά εξάμηνο.
- 2) Ανά αίθουσα θεωρίας ή εργαστήριου.
- 3) Ανά καθηγητή.
- 4) Ανά μάθημα.
- 5) Ανά τομέα.

Συνδυασμός αυτών μεταξύ τους **δεν** γίνεται.

Σε κάθε επιλογή που κάνουμε, αλλάζουν δυναμικά τα πλαίσια (τα κουτιά που περιέχουν τα μαθήματα). Ανάλογα με τον αριθμό των μαθημάτων που βρίσκονται την ίδια ώρα την ίδια , προσαρμόζεται αντίστοιχα και το μέγεθος στο πλαίσιο.

Σε κάθε εναλλαγή των combobox σβήνουν τα προηγούμενα πλαίσια.

Όταν γίνεται η εκκίνηση του flash, θα πρέπει πρώτα να γίνει η επιλογή του XML (έτους) που θέλουμε να δούμε. Για να γίνει αυτό, πρέπει να πατήσουμε το κουμπί «load», όπου και εμφανίζεται στην οθόνη το παράθυρο της Εικόνας 7. Επιλέγουμε το έτος που θέλουμε να δούμε και πατάμε «OK».



Εικόνα 7

Το επιλεγμένο έτος εμφανίζεται πάνω δεξιά στο καμβά μας, Εικόνα 8.

load	...5 ΕΞΑΜΗΝΟ...	ΑΙΘΟΥΣΕΣ ΘΕΩΡΙΑΣ	ΑΙΘΟΥΣΕΣ ΕΡΓΑΣΤΗΡΙΩΝ	ΚΑΘΗΓΗΤΕΣ	Επικοινωνίες
2009-10x.xml	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ
8:15 - 9:15	Ντελέκης Ιωάννης εργ3ΠΚ Δίκτυα Υπολογιστών (Εργαστήριο) ΟΜ1				
9:15 - 10:15				Τριαντ εργ.ΤΗΑ. Ψηφιακή Επεξεργασία Σήματος (Εργαστήριο) ΟΜ3	Πετρούλα εργ1ΠΚ Ψηφιακή Επεξεργασία Σήματος (Εργαστήριο)
10:15 - 11:15					
11:15 - 12:15					
12:15 - 13:15			Σιδέρης Ανόργυρος εργ1ΠΚ Δίκτυα Υπολογιστών (Εργαστήριο) ΟΜ4		
13:15 - 14:15		Καλογιαννίδης Μιχαήλ Γ6 Διδασκαλία της Πληροφορικής στην Εκπαίδευση			
14:15 - 15:15					
15:15 - 16:15		Παπαδουράκης Γεώργιος Γ6 Ψηφιακή Επεξεργασία Σήματος			
16:15 - 17:15					
17:15 - 18:15	Ανδριανίκη Γ1 Ψηφιακή Επεξεργασία Σήματος (Εργαστήριο)		Ανδριανίκη Γ1 Ανάπτυξη Ευρωπαϊκών Προϊόντων		
18:15 - 19:15	Φραγκιαδάς εργ1ΠΚ Δίκτυα Υπολογιστών (Εργαστήριο) ΟΜ2	Μανουσά εργ4ΠΚ Ψηφιακή Επεξεργασία Σήματος (Εργαστήριο) ΟΜ1	Ντελέκης Ιωάννης εργ3ΠΚ Δίκτυα Υπολογιστών (Εργαστήριο) ΟΜ3	Μανουσά εργ4ΠΚ Ψηφιακή Επεξεργασία Σήματος (Εργαστήριο) ΟΜ2	Φραγκιαδάς εργ3ΠΚ Ψηφιακή Επεξεργασία Σήματος (Εργαστήριο) ΟΜ5
19:15 - 20:15					Μελισσόπουλος Γ7 Δίκτυα Υπολογιστών
20:15 - 21:15					

Εικόνα 8.

Για να εμφανιστούν τα μαθήματα, θα πρέπει να επιλέξουμε ένα από τα combobox· αυτό καλεί μια συνάρτηση η οποία συγκρίνει την επιλογή που κάναμε με τα δεδομένα (μαθήματα) που έχουμε στο XML και εκτυπώνει τα αντίστοιχα μαθήματα (μπορούμε να κάνουμε μόνο μία επιλογή από ένα combobox και όχι συνδυασμός αυτών· π.χ. επιλεγούμε ένα εξάμηνο και όχι εξάμηνο και καθηγητή).

Όπως είναι φυσικό, υπάρχει περίπτωση να πραγματοποιούνται περισσότερα από ένα μαθήματα την ίδια ώρα. Για αυτό το λόγο, οι μέρες, οι ώρες και τα πλαίσια που περιέχουν τα μαθήματα, ‘μπαίνουν’ στο καμβά δυναμικά. Όλο αυτό θα πρέπει να διέπεται από μία σωστή ιεραρχική δομή.

Αρχικά, θα πρέπει να δημιουργηθεί ο καμβάς, ώστε να γίνει αργότερα η προσθήκη των μαθημάτων. Δηλαδή, οι μέρες από Δευτέρα ως Παρασκευή και οι ώρες από 8:00 μέχρι 21:00. Η κάθε ώρα της ημέρας έχει έναν ξεχωριστό μετρητή με αρχική τιμή 1 (δες παρακάτω) για να ξέρει το πρόγραμμα ότι υπάρχει το λιγότερο ένα μάθημα την συγκεκριμένη ώρα, αυτός ο μετρητής υπάρχει για να μην μένει αυτή η ώρα κενή με αποτέλεσμα να χαλάσει η δομή του καμβά μας. Επιλέγουμε από τα combobox τι θέλουμε να δούμε (μαθήματα ανά εξάμηνο/ καθηγητή κτλ). Συγκρίνουμε την επιλογή που δώσαμε

με τα δεδομένα που βρίσκονται μέσα στο combobox. Το πρόγραμμα ‘διαβάζει’ το XML ανά εξάμηνο μετά ανά μέρα και μετά ανά ώρα. Αποθηκεύει τα μαθήματα που ταιριάζουν σε ένα πίνακα. Όταν ολοκληρωθεί η ανάγνωση του XML από το πρόγραμμα, ελέγχει τα μαθήματα που έχουν αποθηκευτεί μέσα στον πίνακα. Ελέγχει τις μέρες και τις ώρες που γίνεται κάθε μάθημα. Αν κάποιο μάθημα συμπίπτει με κάποιο άλλο την ίδια ώρα και μέρα, τότε αυξάνει τον μετρητή που έχει αυτή η ώρα. Αφού ολοκληρώσουμε με όλες τις μέρες της βδομάδας και γνωρίζουμε πόσα μαθήματα είναι συνολικά σε κάθε ώρα και μέρα, ξεκινάμε την «εκτύπωση» των μαθημάτων στον καμβά μας. Ανάλογα πόσα μαθήματα είναι την ίδια ώρα αυξομειώνεται και το μέγεθος της κάθε στήλης / γραμμής.

Η κάθε ώρα έχει έναν συγκεκριμένο χώρο στον καμβά μας (π.χ. 2cm). Αν, δηλαδή, έχουμε 4 μαθήματα σε μία ώρα, τότε είναι 2cm δια 4, που είναι τα μαθήματα μας. Υπάρχει και η περίπτωση σε μία μέρα να υπάρχουν τέσσερα μαθήματα την ίδια ώρα και σε μία άλλη μέρα να υπάρχει μόνο ένα μάθημα αυτήν την ώρα, τότε η στήλη (μέρα) με τα περισσότερα μαθήματα γίνεται πιο φαρδιά και η στήλη με τα λιγότερα μαθήματα γίνεται πιο μικρή, έτσι ώστε να είναι ευδιάκριτα τα μαθήματα στον καμβά. Αυτή η διαδικασία έχει ως αποτέλεσμα την δυναμική μεταβολή του μεγέθους της κάθε στήλης και γραμμής, ανάλογα με το πόσα μαθήματα υπάρχουν σε μία συγκεκριμένη ώρα.

Κεφάλαιο 3

Data Model

Στην εργασία αυτή, για την αποθήκευση των δεδομένων, δεν χρησιμοποιήθηκαν βάσεις δεδομένων, αλλά XML. Ο λόγος που προτιμήθηκε η XML είναι η ευκολία του χρήστη να δημιουργήσει το αρχείο XML, να το γεμίσει με τις πληροφορίες που θέλει και να προσπελάσει τα δεδομένα που περιέχει. Επίσης, πολύ σημαντικό είναι να μπορούν να επικοινωνήσουν οι διαφορετικές εφαρμογές μεταξύ τους (πχ php και action Script).

Έχουν δημιουργηθεί συνολικά οκτώ XML, τα οποία περιέχουν πληροφορίες σχετικά με τα μαθήματα του τμήματος, όπως οι τίτλοι των μαθημάτων που γίνονται στο τμήμα, τις διαθέσιμες αίθουσες και εργαστήρια, τα ονόματα των καθηγητών, κ.τ.λ. Τα XML αυτά δεν συνδέονται μεταξύ τους. Λεπτομέρειες για δυο από αυτά τα XML δίνονται στις επόμενες παραγράφους.

Μερικοί βασικοί κανόνες που χρησιμοποιήθηκαν για την σωστή δομή των XML είναι :

- 1) Το έγγραφο πρέπει να περιέχει ένα μόνο στοιχείο ανωτάτου επιπέδου (το στοιχείο εγγράφου ή βασικό στοιχείο). Όλα τα υπόλοιπα στοιχεία πρέπει να είναι ένθετα σε αυτό.

```
<?xml version="1.0" encoding="utf-8" ?>
<aithouses>
    ....
    ....
</ aithouses>
```

- 2) Η ένθεση των στοιχείων πρέπει να είναι σωστή. Αυτό σημαίνει ότι αν ένα στοιχείο ξεκινάει μέσα σε κάποιο άλλο, πρέπει και να τελειώνει μέσα σε αυτό.

```
<?xml version="1.0" encoding="UTF-8" ?>
<mathimata>
```

```
<mathima>
  <id>ΤΠ1004</id>
  <lessons>Εισαγωγή στην Ηλεκτρονική</lessons>
  <tomeas>Επικοινωνίες</tomeas>
</mathima>
</mathimata>
```

- 3) Το όνομα του στοιχείου στην ετικέτα αρχής πρέπει να είναι ίδιο ακριβώς με εκείνο της ετικέτας τέλους.

```
<?xml version="1.0" encoding="UTF-8" ?>
<mathimata>
  ...
  ...
</mathimata>
```

3.1 Αίθουσες XML

Στη συνέχεια, θα αναφερθούμε στο XML με τις αίθουσες θεωρίας. Αυτό το XML περιέχει τις αίθουσες που γίνονται τα θεωρητικά μαθήματα. Το ίδιο XML όπως και όλα τα άλλα XML χρησιμοποιούνται και στο κομμάτι της rhp (προβολή και επεξεργασία) αλλά και στο κομμάτι του flash .

Όλα τα XML έχουν δημιουργηθεί με βάση τους παραπάνω κανόνες σύνταξης του XML. Όλα τα XML έχουν την ίδια δομή, το μόνο που αλλάζει στα XML που έχουν δημιουργηθεί, με σκοπό τη συγκεκριμένη πτυχιακή εργασία, είναι τα δεδομένα που περιέχουν.

Στην πρώτη γραμμή βλέπουμε την δήλωση του XML και την κωδικοποίηση που χρησιμοποιούμε. Για κωδικοποίηση χρησιμοποιώ την «utf-8»(για να υποστηρίζει τους ελληνικούς χαρακτήρες). Στις επόμενες γραμμές είναι τα δεδομένα που βρίσκονται μέσα

στα elements. Στην τρίτη σειρά προστέθηκαν ορισμένα σχόλια (βρίσκονται μέσα στα <!-- -->).

```
1) <?xml version="1.0" encoding="utf-8" ?>
2) <aithouses>
3)     <aithousa>Γ1</aithousa> <!-- αίθουσες  σχόλια-->
4)     <aithousa>Γ2</aithousa>
5)     ....
6)     ....
7)     ....
8)     <aithousa>ΦΧ</aithousa>
9)     <aithousa>ΑΜΘΙ. ΚΑΜΑΡΗ</aithousa>
10) </aithouses>
```

3.2 Μαθήματα XML

Το παρακάτω XML περιέχει τα μαθήματα που έχει το τμήμα, με βάση το εκάστοτε ετήσιο εκπαιδευτικό πρόγραμμα. Η ιεραρχία που διέπει το XML, παρουσιάζεται στην Εικόνα 9.



Εικόνα 9

Με αυτό τον τρόπο μπορούμε να διαβάσουμε το κάθε μάθημα την μέρα που γίνεται, στο αντίστοιχο εξάμηνο.

```

<?xml version="1.0" encoding="utf-8"?>
<bdomada>
  <Aexam>
    <deutera>
      <teacher>
        <imeresgr>Δευτέρα</imeresgr>
        <imereseng>deutera</imereseng>
        <id>ΤΠ3103</id>
        <name>Βαρδιάμπασης Δημήτρης</name>
        <omada>OM1</omada>
        <examino>1</examino>
        <mathima>Δομές Δεδομένων Αλγόριθμοι</mathima>
        <aithousa>εργ2ΠΚ </aithousa>
        <ora_arxi>14</ora_arxi>
        <ora_telos>16</ora_telos>
        <exam_xml>Aexam</exam_xml>
        <theo_erg>ergastirio</theo_erg>
        <tomeasName> Πληροφορική</tomeasName>
      </teacher>
      <teacher>
        ...
        ...
      </teacher>
    </deutera>
  </Aexam>
  ....
  ....
  ....
  <Eexam>
    <deutera>

```

```
<teacher>
  <imeresgr>Δευτέρα</imeresgr>
  <imereseng>deutera</imereseng>
  <id>ΤΠ3103</id>
  <name>Βαρδιάμπασης Δημήτρης</name>
  <omada>OM1</omada>
  <examino>5</examino>
  <mathima>ΔομέςΔεδομένων </mathima>
  <aithousa>εργ2ΠΚ </aithousa>
  <ora_arxi>14</ora_arxi>
  <ora_telos>16</ora_telos>
  <exam_xml>Eexam</exam_xml>
  <theo_erg>ergastirio</theo_erg>
  <tomeasName> Πληροφορική</tomeasName>
</teacher>
</deutera>
</Eexam>
</bdomada>
```

Κεφάλαιο 4

Κώδικας

4.1 Επιλεγμένος κωδικός.

Η αρχική σελίδα (new_data.php) περιέχει την φόρμα εισαγωγής. Σε αυτή την φόρμα μπορεί ο χρήστης να κάνει την προσθήκη των δεδομένων.

Για να γίνει η προσθήκη των δεδομένων με τον σωστό τρόπο, πρέπει πρώτα να τα ελέγχουμε για να μην έχουμε μελλοντικά προβλήματα. Παρακάτω, βλέπουμε μερικούς από αυτούς τους ελέγχους:

```
function check_is_empty()
{
    var etos=document.form.etos.value;
    var examino=document.form.examino.value;
    var mathima=document.form.mathima.value;
    var name=document.form.name.value;
    var arxi=document.form.ora_arxi.value;
    var telos=document.form.ora_telos.value;

    if(etos == " "){
        alert("Δεν έχετε επιλέξει «Έτος».");
        document.form.etos.focus();
        return false;
    }
    if(examino==" "){
        alert("Δεν έχετε επιλέξει «Εξάμηνο».");
        document.form.examino.focus();
        return false;
    }
    if(mathima==" "){
        alert("Δεν έχετε επιλέξει «Μάθημα».");
        document.form.select1.focus();
        return false;
    }
    if(name==" "){
```

```

alert("Δεν έχετε επιλέξει «Καθηγητή».");
document.form.name.focus();
return false;
}
arxi++;
for (var ii=0; ii<=1;ii++){
    arxi++;
    if (arxi == telos ){
        return true;
    }
}
alert("Έχετε βάλει λάθος ώρα, πρέπει να είναι 2-3 ώρα");
return false;

```

Σε αυτό το κομμάτι ελέγχουμε για κενά πεδία όπως το «έτος», «εξάμηνο», «μάθημα», «καθηγητή» επίσης κάνει έλεγχο για τις ώρες των μαθημάτων, αν δηλαδή τα μαθήματα είναι δύο ή τρία.

Όταν πατάμε το κουμπί «επιβεβαίωση», καλούμε την συνάρτηση `check_is_empty()`. Δεν έχουμε βάλει ορίσματα, καθώς τις τιμές τις περνάμε στις μεταβλητές με την `document.form.xxx.value;` (βλέπε *JavaScript*) και τις συγκρίνουμε με το κενό (" "), δηλαδή αν έχουμε εισάγει δεδομένα σε συγκεκριμένα πεδία για να δούμε αν είναι κενά τα πεδία. Αν τα πεδία είναι κενά τότε βγάζει και το αντίστοιχο σφάλμα (*Εικόνα 3*).

Με την εντολή `document.form.xxx.focus();` εστιάζει το λανθασμένο πεδίο.

Με τον παρακάτω κώδικα 'διαβάζουμε' το αρχείο «*etoi.xml*». Το αρχείο αυτό περιέχει τα εξάμηνα (*χειμερινό*, *εαρινό*) που έχει το τμήμα. Η μορφή του xml είναι:

```

<?xml version="1.0" encoding="utf-8"?>
< examina >
    < eksamino >2007-8a.xml</ eksamino >
    < eksamino >2007-8x.xml</ eksamino >

```



```
< eksamino >2008-9a.xml</ eksamino >
</examina>
```

Παρακάτω φαίνεται ο κώδικας που έχουμε στην ιστοσελίδα «προσθήκη δεδομένων».

```
<td>ετος</td>
<?php
$count=1;
    $doc = new DOMDocument();
    $doc->load('xml_info/etoi.xml');
    foreach($doc->childNodes as $setoi){
        if($setoi->nodeName=="#text")
            continue;
        $id_xml = $setoi->getElementsByTagName( "eksamino" );
        $total=$id_xml->length;
        for($i=0; $i<=$total; $i++){
            $data_id_xml = $id_xml->item($i)->nodeValue;
            $setoi_eksamina[$count]=$data_id_xml;
            $count++;
        }
    }
?><td width="83%"><select name="etos">
<?php
    for( $i=0; $i<=$count; $i++)
    {
        echo "<option value='$setoi_eksamina[$i]'"> $setoi_eksamina[$i]
</option>";
    }
}
?>
</select>
```

Ποιο αναλυτικά : δημιουργούμε μια μεταβλητή \$doc της κλάσης DOMDocument() και φορτώνουμε το XML. Με την εντολή foreach διαβάζουμε τα στοιχεία του πίνακα και τα αποθηκεύουμε στην μεταβλητή \$setoi. Με την συνάρτηση getElementsByTagName("eksamino"); μας επιστρέφει μία λίστα των στοιχείων «eksamino». Τα στοιχεία αυτά τα αποθηκεύουμε στην μεταβλητή \$id_xml. Με την εντολή \$total=\$id_xml->length; Παίρνουμε βλέπουμε πόσες έγγραφες έχουμε μέσα στο \$id_xml και το αποτέλεσμα αποθηκεύεται στο \$total. Το αποτέλεσμα αυτό το χρησιμοποιώ για το μέγεθος του combobox. Με την εντολή \$data_id_xml = \$id_xml->item(\$i)->nodeValue; Περνούμε ένα ένα τα \$id_xml και τα αποθηκεύουμε στο \$setoi_eksamina, για να μπορέσουμε να τα προβάλουμε αργότερα. Με τον ίδιο τρόπο γεμίζουμε και τα υπόλοιπα combobox που έχουμε στο πρόγραμμα μας.

Για να προβάλουμε τα μαθήματα στην σελίδα μας, θα πρέπει να επιλέξουμε το «Προβολή δεδομένων». Στη συνέχεια, επιλέγουμε ποιο εξάμηνο θέλουμε να δούμε,(πρώτα μας εμφανίζονται τα μαθήματα του πρώτου εξαμήνου). Αν θέλουμε να δούμε και τα υπόλοιπα μαθήματα τότε επιλέγουμε από το combobox το εξάμηνο της επιλογής μας.

Ο κώδικας της διαδικασίας επιλογής του εξαμήνου παρουσιάζεται παρακάτω.

```
?>Επιλογή Εξαμήνου<?
$eksaminoArray=array(' ',1 εξάμηνο',2 εξάμηνο',3 εξάμηνο',4 εξάμηνο',5
εξάμηνο',6 εξάμηνο',7 εξάμηνο','προαιρετικά ', 'υποχρεωτικά');
echo "<form name='eksaminoemfanisis' method='post' >";
echo "<select name='eksaminoemfanisis' onChange='changeExam(this.form)'>";
?><option> επίλεξε εξάμηνο</option><?
for ($eksamino = 1; $eksamino<=9; $eksamino++)
{
echo"<option value='$eksamino' > $eksaminoArray[$eksamino]</option>";
}
....
....
</form>
```

Εδώ βλέπουμε μια φόρμα που περιέχει το Combobox που με την σειρά του περιέχει τα εξάμηνα μας. Κάθε φορά που αλλάζουμε την επιλογή μας, καλείται η συνάρτηση «*changeExam(this.form)*», σαν όρισμα έχουμε βάλει όλη την φόρμα για να μπορούμε να την κάνουμε μετά submit().

```
function changeExam(fRefs){  
    fRefs.action="see_data.php";  
    fRefs.submit();  
}
```

Κεφάλαιο 5 - Επίλογος

5.1 Συμπεράσματα

Η δυναμική δημιουργία του προγράμματος θα βοηθάει τον φοιτητή να βρίσκει σε λιγότερο χρόνο τα μαθήματα που τον ενδιαφέρουν χρησιμοποιώντας τις δυνατές επιλογές που του δύνονται. Επίσης ο διαχειριστής του προγράμματος θα μπορεί εύκολα να αλλάζει τα μαθήματα και να τα προβάλλει.

Το κομμάτι της PHP θα βρίσκεται είτε σε κάποιο server (και στο διαδίκτυο) ώστε να έχει πρόσβαση οποιοδήποτε έχει τα δικαιώματα σε αυτόν ώστε να μπορεί να κάνει τις αλλαγές των μαθημάτων (γραμματεία ή καθηγητής) είτε να είναι τοπικά αποθηκευμένο σε έναν μόνο ηλεκτρονικό υπολογιστή(γραμματεία) για να μπορεί να γίνεται σωστός έλεγχος της σωστής διεξαγωγής των μαθημάτων.

Το δεύτερο κομμάτι αναφέρεται στην προβολή των μαθημάτων (flash). Αυτό πρέπει να βρίσκεται σε ένα server (web) ώστε να μπορούν να έχουν πρόσβαση οι φοιτητές.

5.2 Εξέλιξη Συστήματος

Είναι γνωστό ότι όλες οι σελίδες που βρίσκονται στο διαδίκτυο, αλλά και γενικότερα, όλα τα προγράμματα που υπάρχουν, μπορούν να δεχτούν βελτιώσεις. Έτσι και η παρούσα πτυχιακή εργασία μπορεί να εξελιχτεί.

Καταρχήν, μπορούμε να αναφερθούμε στη διαμόρφωση της αρχικής σελίδας που βλέπει ο επισκέπτης. Θα μπορούσε να γίνει πιο φιλική και ευχάριστη προς τους χρήστες, τοποθετώντας flash animation και αλλάζοντας το interface, βοηθώντας τον χρήστη στις επιλογές του.

Αν είχαμε όλα τα μαθήματα που γίνονται στο τμήμα δομημένα(αρχείο .xls) π.χ. 1εξάμηνο Μαθηματικά, Καθηγητές: Καραγιαννάκης Σμυρνάκης, θα μπορούσαμε να κάνουμε περισσότερους ελέγχους, αλλά και περιορισμούς στην σελίδα «εισαγωγή δεδομένων» για να μπορεί ο χρήστης, όταν διαλέγει το εξάμηνο που θέλει να του εμφανίζονται τα μαθήματα του συγκεκριμένου εξάμηνου και όταν επιλέξει ποίο μάθημα θέλει να του εμφανίζονται και οι καθηγητές που διδάσκουν το συγκεκριμένο μάθημα. Με

αυτόν τον τρόπο, θα ήταν πιο γρήγορη και εύκολη η επιλογή των μαθημάτων και των καθηγητών.

Επίσης θα μπορούσαμε να είχαμε δυο εφαρμογές flash (swf files). Το ένα (A) θα ήταν αυτό που ήδη έχουμε και το δεύτερο (B), κάποιο όπου θα είχε πρόσβαση μόνο ο διαχειριστής. Σε A θα μπορούσε ο διαχειριστής επιλέγοντας κάποιο μάθημα να το «διαγράψει» ή να το «μεταφέρει» σε κάποια άλλη ώρα και μέρα της βδομάδας και δυναμικά να άλλαζαν και το XML. Με αυτό τον τρόπο, θα ήταν πιο εύκολη η επεξεργασία (αλλαγή, διαγράφη) των μαθημάτων. Ωστόσο, για να γίνει αυτό στα πλαίσια μίας πτυχιακής εργασίας, είναι αν όχι ακατόρθωτο, πάντως πάρα πολύ δύσκολο. Έτσι προς το παρόν, έχουμε δημιουργήσει μόνο το πρώτο swf file.

Βιβλιογραφία

Ανάπτυξη web εφαρμογών με Php και MySQL Luke Welling Laura Thomson εκδόσεις Μ. Γκιούρδας.

Ο επίσημος οδηγός του flash 8 Jobe Maker/ Danny Patterson εκδόσεις Μ. Γκιούρδας.

Πλήρες εγχειρίδιο της JavaScript Martin Webb, Michel Plungjan εκδόσεις Μ. Γκιούρδας.

www.freestuff.gr

www.adobe.com