



**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**



**Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και  
επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ  
ΠΕΙΡΑΙΑ Τ.Τ.**

**Γεωργάς Σ. Ευστάθιος  
Κογιανός Α. Δημήτριος**

**Εισηγητής: Δρ Παναγιώτης Γιαννακόπουλος, Καθηγητής**

**ΑΘΗΝΑ  
ΜΑΙΟΣ 2017**

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.**

**Γεωργάς Σ. Ευστάθιος  
Α.Μ. 40459  
Κογιανός Α. Δημήτριος  
Α.Μ. 42633**

**Εισηγητής:**

**Δρ Παναγιώτης Γιαννακόπουλος, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ζάχαρης Νικόλαος, Καθηγητής  
Νικολόπουλος Δημήτριος, Καθηγητής**

**Ημερομηνία εξέτασης 26/5/2017**

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Οι κάτωθι υπογεγραμμένοι ΓΕΩΡΓΑΣ ΕΥΣΤΑΘΙΟΣ και ΚΟΓΙΑΝΟΣ ΔΗΜΗΤΡΙΟΣ, του ΣΑΒΒΑ και ΑΛΕΞΑΝΔΡΟΥ αντίστοιχα, με αριθμό μητρώου 40459 και 42633 αντίστοιχα, φοιτητές του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της δημιουργίας μιας ιστοσελίδας. Την προσπάθειά μας αυτή υποστήριξε ο επιβλέπων καθηγητής μας, τον οποίο θα θέλαμε να ευχαριστήσουμε.

Η πτυχιακή εργασία αυτή, αποτελεί καρπό χρόνου, κόπης και συνεργασίας (εμάς) των δημιουργών της, Γεωργά Ευστάθιου και Δημήτριου Κογιανού που από κοινού δουλέψαμε σκληρά αναμειγνύοντας τις γνώσεις μας για την εκπόνηση της.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.



## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη ενός διαδικτυακού τύπου (ιστοσελίδας). Σε αυτό το πεδίο έχουν αναπτυχθεί ιστορικά αρκετές ιστοσελίδες, καθώς το διαδίκτυο αποτελεί ένα μεγάλο μέρος της καθημερινής μας ζωής, το οποίο διαρκώς αναπτύσσεται έχοντας λάβει τεράστιες διαστάσεις και θα μπορούσαμε να πούμε με έναν πιο αόριστο όρο, με την έκταση και διαρκή ανάπτυξη του αποτελεί ένα ζωντανό οργανισμό.

Η πτυχιακή εργασία αυτή προορίζεται για χρήση εντός του ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ. από το γραφείο του προγράμματος Erasmus+ υπεύθυνο για τη διεκπεραίωση αιτήσεων που αφορούν το συγκεκριμένο πρόγραμμα.

Σκοπός της πτυχιακής εργασίας είναι να βοηθήσει στην εξυπηρέτηση των φοιτητών που είναι άμεσα ενδιαφερόμενοι για την εισαγωγή τους στο πρόγραμμα Erasmus+ με σκοπό να εκπληρώσουν μέρος της φοίτησης που δικαιούνται σε κάποιο πανεπιστήμιο εξωτερικού εντός της Ευρώπης.

Ο τρόπος με τον οποίο αυτή η πτυχιακή εργασία βοηθά στο να γίνει αυτό, είναι μέσω της ανάπτυξης μιας ιστοσελίδας (πλατφόρμας) στην οποία οι ενδιαφερόμενοι φοιτητές, καταθέτουν όλα τα έγγραφα που απαιτούνται για την έγκρισή τους για το πρόγραμμα Erasmus+ και κατόπιν έλεγχου της ορθότητας τους από τους υπεύθυνους του γραφείου, ενημερώνονται για την τελική υποβολή τους μέσω της φυσικής τους παρουσίας στο χώρο του γραφείου Erasmus+ για εξακρίβωση και την τελική έγκριση/επιλογή τους για το πρόγραμμα από τους υπόλοιπους υποψήφιους.

Με αυτό τον τρόπο εξασφαλίζεται η ορθότητα και η ταχύτητα της διαδικασίας, αποσυμφορίζοντας τόσο τους φοιτητές όσο και τους εργαζόμενους του γραφείου από περίσσιο φόρτο εργασίας ή τυχών λάθη και λανθασμένη συνεννόηση που μπορεί να προκύψουν δημιουργώντας ένα πιο ευχάριστο, εύχρηστο και αποδοτικό τρόπο διεξαγωγής της διαδικασίας υποβολής και έγκρισης των εγγράφων.

## **ABSTRACT**

This diploma thesis deals with the development of a website. In this field, several websites have historically been developed, as the internet is a major part of our everyday life, which is constantly developing, having gained enormous dimensions, and it can be said it with a more indefinite term, with the vast expansion and growth of the internet, it can be described as a living organism.

This dissertation is intended for use within the University of Piraeus from the Erasmus+ office, responsible for handling the applications for this program.

The purpose of this diploma thesis is to help students who are directly interested in entering the Erasmus+ program in order to fulfill part of the studies they are entitled to at an external university within Europe.

The way this diploma thesis helps to achieve this goal is through the development of a website (platform) in which interested students submit all the documents required for their Erasmus+ approval, and a check of their correctness by those responsible from the office occurs in order for the students to be informed of their final submission through their physical presence at the Erasmus+ office for verification and final approval / selection for the program from the rest of the candidates.

This ensures the correctness and speed of the process, decongesting both students and office workers from unnecessary workloads, or any mistakes that may arise, creating a more enjoyable, manageable, and cost-effective way of submitting and having the documents approved.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Διαδίκτυο και Λογισμικό Ηλεκτρονικών Υπολογιστών  
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ιστοσελίδα, διαδίκτυο, πρόγραμμα, μεταβλητές, Erasmus+

Γεωργάς Σ. Ευστάθιος  
Κογιανός Α. Δημήτριος

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. ΕΙΣΑΓΩΓΗ.....</b>	<b>15</b>
1.1 Περιγραφή του αντικείμενου της πτυχιακής εργασίας.....	15
1.2 Λίγα λόγια για την πλατφόρμα Erasmus+ και τη δομή της Πτυχιακής Εργασίας.....	15
1.3 Εισαγωγή στο διαδίκτιο.....	16
1.4 Η εξέλιξη του διαδικτυου.....	17
1.5 World Wide Web (WWW).....	18
<b>2. ΠΡΟΓΡΑΜΜΑ ERASMUS PLUS.....</b>	<b>21</b>
2.1 Γενικές πληροφορίες σχετικά με το πρόγραμμα Erasmus+.....	21
2.2 Γενικός στόχος.....	22
2.3 Αναγνώριση και επικύρωση των δεξιοτήτων και των επαγγελματικών προσόντων.....	23
2.4 Ποιος μπορεί να συμμετέχει στο πρόγραμμα Erasmus+.....	24
<b>3. PHP.....</b>	<b>25</b>
3.1 Βασικές γνώσεις PHP.....	25
3.2 Βασικά στοιχεία της PHP.....	26
3.3 Πλεονεκτήματα.....	26
3.4 Βασικές αρχές λειτουργίας.....	27
<b>4. MySQL.....</b>	<b>29</b>
4.1 Βασικές γνώσεις MySQL.....	29
<b>5. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.....</b>	<b>31</b>
5.1 Τι είναι βάσεις δεδομενων.....	31
5.2 Ορισμός βάσεων δεδομένων.....	31
5.3 Χρήση της βάσης δεδομενων.....	31
5.4 Πρότυπα βάσεων δεδομενων.....	32
5.5 Βασικές έννοιες στις βάσεις δεδομενων.....	33
5.6 Πρότυπα δεδομενων.....	33
5.7 Ανεξαρτησία δεδομενων.....	33
5.8 Πλεονεκτήματα των βάσεων δεδομενων.....	34
5.9 Μειονεκτήματα των βάσεων δεδομενων.....	34

5.10	Αρχιτεκτονική και σχεδίαση - αρχές σωστής σχεδίασης.....	35
5.11	Αρχιτεκτονικές συστημάτων διαχείρισης βάσεων δεδομένων - βαθμίδες και αρχιτεκτονικές.....	36
5.12	Αρχιτεκτονική μιας βαθμίδας (1-tier).....	37
5.13	Αρχιτεκτονική δύο βαθμίδων (2-tier).....	38
5.14	Αρχιτεκτονική πολλαπλών βαθμίδων (n-tier).....	39
5.15	Αρχιτεκτονική διαδικτύου.....	40
5.16	Σύγκριση αρχιτεκτονικών 2-βαθμίδων και 3-βαθμίδων.....	41
5.17	Σχεδίαση βάσεων δεδομένων - στάδια ζωής βάσεων δεδομένων.....	42
5.18	Επίπεδα σχεδίασης βάσεων δεδομένων.....	43
5.19	Μοντέλα σχεδίασης εννοιολογικού - λογικού επιπέδου.....	43
5.20	Επίπεδο Μοντέλο (Flat Model).....	43
5.21	Ιεραρχικό Μοντέλο (Hierarchical Model).....	44
5.22	Μοντέλο Δικτύου (Network Model).....	45
5.23	Σχεσιακό Μοντέλο (Relational Model).....	46
5.24	Αντικειμενοστραφές μοντέλο (Object Oriented Model).....	46
5.25	Λειτουργίες και γλωσσες.....	46
5.26	Βάση δεδομένων πλατφόρμας Erasmus+ Office ΑΕΙ Πειραιά Τ.Τ...	47
<b>6.</b>	<b>ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΕΝΝΟΙΕΣ BACKEND FRONTEND.....</b>	<b>51</b>
6.1	Frontend.....	51
6.2	Backend.....	52
<b>7.</b>	<b>JSON.....</b>	<b>55</b>
7.1	Τι είναι το JSON.....	55
<b>8.</b>	<b>WEB SERVICES.....</b>	<b>59</b>
8.1	Ορισμός των web services.....	59
8.2	Η δύναμη των web services.....	60
8.3	Αρχιτεκτονική των web services ( SOA ).....	61
8.4	Τεχνολογίες των web services.....	62
<b>9.</b>	<b>SOAP – SIMPLE OBJECT ACCESS PROTOCOL.....</b>	<b>65</b>
9.1	Τι είναι το SOAP.....	65
9.2	Εισαγωγή στις βασικές εννοιες.....	65
9.3	Μήνυμα SOAP.....	66
9.4	Μοντέλο επεξεργασίας SOAP.....	67

9.5 Μοντέλο ανταλλαγής μηνυμάτων.....	68
9.6 Απομακρυσμένες κλήσεις διαδικασιών.....	68
9.7 HTTP Binding.....	70
<b>10. WSDL - WEB SERVICES DESCRIPTION LANGUAGE.....</b>	<b>71</b>
10.1 Τι είναι το WSDL.....	71
10.2 Εισαγωγή στις βασικές εννοιες.....	71
10.3 Τεχνική περιγραφή της WSDL.....	72
<b>11. UDDI - UNIVERSAL DESCRIPTION DISCOVERY AND INTEGRATION.....</b>	<b>75</b>
11.1 Τι είναι το UDDI.....	75
11.2 Εισαγωγή στις βασικές εννοιες.....	75
11.3 Ορισμός δομών δεδομένων.....	77
<b>12. HTML.....</b>	<b>79</b>
12.1 Τι είναι η HTML.....	79
12.2 Προέλευση.....	80
12.3 Πρώτες προδιαγραφές.....	80
12.4 Σήμανση.....	81
12.5 Στοιχεία.....	82
12.6 Η δομική σήμανση περιγράφει τον σκοπό του κειμένου.....	82
12.7 Η σήμανση οπτικής μορφοποίησης και κειμένου.....	82
12.8 Η σήμανση υπερκειμένου και σύνδεση εγγράφων.....	83
12.9 Ιδιότητες.....	83
12.10 Αναφορές οντοτήτων χαρακτήρων.....	84
12.11 Τύποι δεδομένων.....	85
12.12 Δήλωση τύπου εγγράφου.....	85
12.13 Σημασιολογική.....	86
12.14 Μεταφορά της HTML.....	87
12.15 HTTP.....	87
12.16 HTML e-mail.....	87
12.17 Ονόματα αρχείων.....	88
12.18 Δημιουργία με προγράμματα WYSIWYG.....	88
12.19 HTML 5.....	88
12.20 Εισαγωγή στην HTML και τα CSS.....	89

12.21 Βασική δομή μιας ιστοσελίδας γραμμένης με HTML.....	89
12.22 Επεκτείνοντας τη βασική δομή μιας ιστοσελίδας HTML.....	90
<b>13. CSS.....</b>	<b>93</b>
13.1 Τι είναι η CSS.....	93
13.2 Η αλληλουχία εφαρμογής των φύλλων στυλ.....	93
13.3 Εισαγωγή στη CSS.....	94
<b>14. JAVASCRIPT.....</b>	<b>95</b>
14.1 Τι είναι Javascript.....	95
14.2 Λίγα λόγια για τη Javascript.....	96
14.3 Ιστορία της Javascript.....	97
14.4 Μοντέλο εκτέλεσης.....	97
14.5 Javascript και Java.....	97
<b>15. ANGULARJS.....</b>	<b>99</b>
15.1 Τι είναι η Angular.....	99
<b>16. PHALCON FRAMEWORK.....</b>	<b>101</b>
16.1 Τι είναι το Phalcon Framework.....	101
16.2 Πλεονεκτήματα του framework.....	101
16.3 Βασικά Χαρακτηριστικά.....	101
16.4 Data & Storage.....	102
16.5 Απόψεις και frontend.....	102
<b>17. ΚΡΥΠΤΟΓΡΑΦΗΣΗ ΚΑΙ ΑΣΦΑΛΕΙΑ ΙΣΤΟΣΕΛΙΔΑΣ ERASMUS+.....</b>	<b>105</b>
17.1 Τι είναι το SHA1.....	105
<b>18. ΠΛΑΤΦΟΡΜΑ ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ. ERASMUS+ OFFICE.....</b>	<b>107</b>
18.1 Περιήγηση στο Site και εγγραφή - Οδηγός Χρήσης.....	107
18.2 Δημιουργία Λογαριασμού Χρήστη (Registration).....	108
18.3 Εισαγωγή στο Site Erasmus+.....	110
18.4 Διαδικασία Εγγραφής – Υποβολής Εγγραφών.....	111
18.5 Μέρος Πρωτο.....	111
18.6 Μέρος Δευτερο.....	114
18.7 Διαχειριστικό Erasmus+ Office Site.....	115
<b>19. ΠΑΡΑΡΤΗΜΑ Α'.....</b>	<b>121</b>
<b>20. ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>123</b>

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

**BAB** Binary Alpha Blocks

**BAP** Binary Alpha Planes

**BMA** Block Matching Algorithm

**TCP** Transmission Control Protocol

**WWW** World Wide Web

**ISP** Internet Service Providers

**IP** Internet Protocol

**ΒΔ** Βάση/Βάσεις Δεδομένων

**ΣΔΒΔ** Σύστημα Διαχείρισης Βάσης Δεδομένων

**NSF** National Science Foundation

**JSON** JavaScript Object Notation

**TM** Technical Model

**SUDO** Super User Doer

**PAD** Presentation Application Data

**SHA1** Secure Hash Algorithm 1

**HTML** HyperText Markup Language

**IETF** Internet Engineering Task Force

**DTD** Document Type Definition

**CSS** Cascading Style Sheets

**JS** JavaScript

**SOAP** Simple Object Access Protocol

**SOA** Service Oriented Architecture

**RPC** Remote Procedure Call

**WSDL** Web Services Description Language

**CLI** Command Line Interface

## **ΚΕΦΑΛΑΙΟ 1**

### **ΕΙΣΑΓΩΓΗ**

#### **1.1 Περιγραφή του αντικείμενου της πτυχιακής εργασίας**

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας πλατφόρμας / ιστοσελίδας για το γραφείο του Erasmus+ του ΑΕΙ Πειραιά Τ.Τ. που ως σκοπό έχει την αποτελεσματικότερη και γρηγορότερη εγγραφή ενδιαφερόμενων φοιτητών στο ευρωπαϊκό πρόγραμμα Erasmus+. Με τη χρήση πλέον του διαδικτύου ως βασικό εργαλείο, οι φοιτητές μέσω αυτής της πλατφόρμας μπορούν πλέον να έχουν τη λιγότερη δυνατή επαφή με τους υπαλλήλους του γραφείου Erasmus+ καθώς το μεγαλύτερο αν όχι ολόκληρο το μέρος της διαδικασίας αυτοματοποιείται και οι αιτήσεις γίνονται ηλεκτρονικά για την εκδήλωση του ενδιαφέροντος για το πρόγραμμα. Οι ενδιαφερόμενοι φοιτητές μέσω αυτής της πλατφόρμας καλούνται να υποβάλλουν σε ηλεκτρονική μορφή όλα τα έγγραφα που απαιτούνται για τη συμμετοχή τους στο πρόγραμμα. Η εξακρίβωση και η έγκριση των εγγράφων θα γίνεται από τους υπαλλήλους του γραφείου Erasmus+ και η γενικότερη αλληλεπίδραση με τους ενδιαφερόμενους φοιτητές. Αφού το γραφείο επιλέξει μέσα από τα δυο στάδια υποβολής εγγράφων (γίνεται εκτενέστερη περιγραφή στο τελευταίο κεφάλαιο της ακριβούς διαδικασίας) που απαιτούνται να υποβληθεί ο ενδιαφερόμενος και γίνει δεκτός για το πρόγραμμα, απαιτείται κατόπιν ειδοποίησης του γραφείου Erasmus+ η φυσική παρουσία του στο χώρο (του γραφείου), ώστε να γίνει και η τελική πιστοποίηση των εγγράφων του που πλέον είναι σε επίπεδο υλικής μορφής.

Με τον τρόπο αυτό, επιτυγχάνεται η βέλτιστη συνεργασία του ιδρύματος με τον ενδιαφερόμενο φοιτητή για την διεξαγωγή αυτής της διαδικασίας. Εξασφαλίζεται, καλύτερη συνεννόηση, αποσυμφόρηση του γραφείου Erasmus+ από εξέχουσα κίνηση φοιτητών, ταχύτητα και αποτελεσματικότητα στο πέρας της διαδικασίας. Εκτός αυτών, πρόκειται για μια τελείως αυτοματοποιημένη διαδικασία που διευκολύνει τόσο το φοιτητή όσο και το γραφείο και βοηθά στη πραγματοποίηση λιγότερων σφαλμάτων όπως επίσης και στη δημιουργία μιας ψηφιακής βάσης δεδομένων όπου όλοι οι ενδιαφερόμενοι φοιτητές για όλο το ΑΕΙ Πειραιά Τ.Τ. θα είναι καταχωρημένοι εκεί.

#### **1.2 Λίγα λόγια για την πλατφόρμα Erasmus+ και τη δομή της Πτυχιακής Εργασίας**

Στην πτυχιακή εργασία αυτή κάθε κεφάλαιο περιλαμβάνει και από ένα από τα διάφορα ήδη τεχνολογιών λογισμικού (software) από προγραμματιστικής πλευράς που έχουν χρησιμοποιηθεί για τη δημιουργία αυτής της πλατφόρμας / ιστοσελίδας. Ο τρόπος δομής της πτυχιακής εργασίας αυτής ξεκινά από το Backend της ιστοσελίδας που χρησιμοποιήθηκε και το πώς χρησιμοποιήθηκε φτάνοντας ως το Frontend. Αφού γίνει κατανοητό αυτό το μέρος στο τέλος της εργασίας δίνεται ένας αναλυτικός οδηγός ο οποίος επεξηγεί εικονογραφημένα και



βήμα προς βήμα τη χρήση της ιστοσελίδας.

Σκοπός είναι με αυτό τον τρόπο η πτυχιακή εργασία αυτή να έχει Αρχή-Μέση-Τέλος, για να είναι όσο το δυνατόν πιο κατανοητή σε κάποιον που είναι σχετικός με τον προγραμματισμό, τους υπολογιστές και τη δημιουργία ιστοσελίδων, αλλά και σε έναν αρχάριο ο οποίος δεν γνωρίζει και είτε έχει ή όχι τις κατάλληλες βάσεις για κάτι τέτοιο είτε αποσκοπεί να ξεκινήσει να τις αποκτάει κάνοντας μια εισαγωγή από εδώ.

Για τις περισσότερες από τις τεχνολογίες λογισμικού (software) που χρησιμοποιήθηκαν για τη δημιουργία αυτής της πλατφόρμας γίνεται εκτενής αναφορά και όσο το δυνατόν αναλυτικότερη καθώς υπήρχαν πολλά δεδομένα που έπρεπε να συμπεριληφθούν και ορισμένες φορές ήταν τόσα, οπου για λόγους όγκου δεν συμπεριλήφθηκαν όλα. Σε ελάχιστες περιπτώσεις τα δεδομένα που καταφέραμε να συλλέξουμε για ορισμένες τεχνολογίες λογισμικού ήταν λίγα, αλλά τουλάχιστον προσπαθήσαμε να γίνουμε όσο το δυνατόν πιο κατανοητοί.

### 1.3 Εισαγωγή στο διαδίκτυο

Με την εμφάνιση οποιουδήποτε νέου μέσου επικοινωνίας, η κοινωνία και οι δομές της αναμφισβήτητα επηρεάζονται. Η επίδραση αυτή πηγάζει κυρίως από την τεχνολογία του νέου μέσου. Είναι πλέον εμφανές πως το διαδίκτυο, ως ένα από τα πλέον πιο σημαντικά μέσα επικοινωνίας στον σύγχρονο κόσμο έχει διεισδύσει σε κάθε γωνιά της υφηλίου, δημιουργώντας εικονικούς "χώρους" καθώς και εικονικές "κοινότητες", όπου παύουν να υφίστανται οι γεωγραφικές, κοινωνικές και πολιτιστικές διαχωριστικές γραμμές που υπάρχουν στον πραγματικό κόσμο και που τα παραδοσιακά μέσα επικοινωνίας αδυνατούν να ξεπεράσουν εύκολα.

Με την δημιουργία του διαδικτύου, η επικοινωνία έγινε άμεση και αμφίδρομη. Οι αποστάσεις εκμηδενίστηκαν και δόθηκε η δυνατότητα σε κάθε χρήστη ηλεκτρονικού υπολογιστή συνδεδεμένου στο Διαδίκτυο, να πληροφορηθεί αλλά και να πληροφορήσει ανταλλάσσοντας απόψεις μέσω ενός συμμετοχικού διαύλου επικοινωνίας. Οι χρήστες αποκτούν ολοένα και περισσότερο την ιδιότητα του παγκόσμιου πολίτη. Είναι εμφανείς λοιπόν οι λόγοι για τους οποίους το διαδίκτυο θεωρείται ένα από τα πιο δημοκρατικά σύγχρονα μέσα μαζικής επικοινωνίας καθώς δίνει στον μέσο άνθρωπο τη δυνατότητα πρόσβασης σε μεγάλο όγκο πληροφοριών συγκεντρωμένων σε ένα "χώρο" και την δυνατότητα της προσωπικής επιλογής των πληροφοριών αυτών.

Ωστόσο δε γίνεται να μην αναφερθούμε και στα μειονεκτήματα που προκύπτουν μέσω της ελευθερίας της έκφρασης και της άμεσης πρόσβασης σε ένα μεγάλο όγκο πληροφοριών που υποστηρίζει το Διαδίκτυο. Η παραβίαση πνευματικών δικαιωμάτων, η πορνογραφία, η ταυτοπροσωπία και η προσφορά παρανόμων προϊόντων είναι μερικά μόνο από τα αποτελέσματα της κακής χρήσης των δυνατοτήτων που παρέχει το Διαδίκτυο. Το τεχνικό και νόμιμο framework της λειτουργίας του Internet θα πρέπει να αντιμετωπίσει την αναταραχή της κακής αυτής χρήσης μέσα από την δικιά του διαδικασία εξέλιξης.

#### 1.4 Η εξέλιξη του διαδικτύου

Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, με τους όρους Διαδίκτυο (με κεφαλαίο το αρχικό γράμμα) περιγράφεται το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και των υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Η τεχνολογία του είναι κυρίως βασισμένη στην διασύνδεση επιμέρους δικτύων ανά τον κόσμο και πολυάριθμα τεχνολογικά πρωτόκολλα. Ωστόσο είναι προφανές πως για να φτάσει στην σημερινή του δομή ακολούθησε την δική του διαδρομή εξέλιξης.

Το σημερινό Διαδίκτυο ( Internet ) αποτελεί εξέλιξη του ARPANET, ενός δικτύου που άρχισε να αναπτύσσεται πειραματικά στα τέλη της δεκαετίας του 60 στις ΗΠΑ. Εγκαταστάθηκε και λειτούργησε για πρώτη φορά το 1969 με 4 κόμβους μέσω των οποίων συνδέονται 4 μίνι υπολογιστές. Μέχρι το 1972 οι συνδεδεμένοι στο ARPANET υπολογιστές είχαν φτάσει τους 23, οπότε και εφαρμόζεται για πρώτη φορά το σύστημα διαχείρισης ηλεκτρονικού ταχυδρομείου ( e-mail ). Παράλληλα δημιουργήθηκαν και άλλα δίκτυα, που χρησιμοποιούσαν διαφορετικές μεθόδους και τεχνικές (όπως το x.25 και το UUCP) τα οποία συνδέονταν με το ARPANET. Το πρωτόκολλο που χρησιμοποιούσε το ARPANET ήταν το NCP ( Network Control Protocol ), το οποίο όμως είχε το μειονέκτημα ότι λειτουργούσε μόνο με συγκεκριμένους τύπους υπολογιστών. Έτσι δημιουργήθηκε η ανάγκη στις αρχές του 1970 για ένα πρωτόκολλο που θα ένωνε όλα τα δίκτυα που είχαν δημιουργηθεί μέχρι τότε.

Το 1973 ξεκινά ένα νέο ερευνητικό πρόγραμμα που ονομάζεται Interneting Project (Πρόγραμμα Διαδικτύωσης). Από την έρευνα γεννιέται μια νέα τεχνική, το Internet Protocol (IP) (Πρωτόκολλο Διαδικτύωσης), από την οποία θα πάρει αργότερα το όνομά του το Internet. Διαφορετικά δίκτυα που χρησιμοποιούν το κοινό πρωτόκολλο IP μπορούν να συνδέονται και να αποτελούν ένα διαδίκτυο. Επίσης σχεδιάζεται μια άλλη τεχνική για τον έλεγχο της μετάδοσης των δεδομένων, το Transmission Control Protocol (TCP) (Πρωτόκολλο Ελέγχου Μετάδοσης). Ορίζονται προδιαγραφές για τη μεταφορά αρχείων μεταξύ υπολογιστών ( FTP ) και για το ηλεκτρονικό ταχυδρομείο ( E-mail ). Σταδιακά συνδέονται με το ARPANET ιδρύματα από άλλες χώρες, με πρώτα το University College of London (Αγγλία) και το Royal Radar Establishment (Νορβηγία).

Το 1983, το πρωτόκολλο TCP/IP (δηλ. ο συνδυασμός των TCP και IP) αναγνωρίζεται ως πρότυπο από το Υπουργείο Άμυνας των ΗΠΑ. Η έκδοση του λειτουργικού συστήματος Berkeley UNIX το οποίο περιλαμβάνει το TCP/IP συντελεί στη γρήγορη εξάπλωση της δικτύωσης των υπολογιστών. Το 1985 το National Science Foundation (NSF) έκανε ένα σημαντικό βήμα στην ανάπτυξη του Διαδικτύου. Δημιουργεί την πρώτη διαδικτυακή πανεπιστημιακή ραχοκοκαλιά (backbone), το NSFNet, χρησιμοποιώντας το πρωτόκολλο TCP/IP. Στα τέλη της δεκαετίας του '80, όλο και περισσότερες χώρες συνδέονται στο NSFNET (Καναδάς, Γαλλία, Σουηδία, Αυστραλία, Γερμανία, Ιταλία, κ.α.). Χιλιάδες πανεπιστήμια και οργανισμοί δημιουργούν τα δικά τους δίκτυα και τα συνδέουν πάνω στο παγκόσμιο αυτό δίκτυο το οποίο αρχίζει να γίνεται γνωστό σαν INTERNET και να εξαπλώνεται με τρομερούς ρυθμούς σε ολόκληρο τον κόσμο. Το 1990 το ARPANET πλέον καταργείται.

Το παγκόσμιο δίκτυο για όλους είναι πλέον γεγονός. Όλο και περισσότερες

χώρες συνδέονται στο NSFNet, μεταξύ των οποίων και η Ελλάδα το 1990. Το 1993 το εργαστήριο CERN στην Ελβετία παρουσιάζει το World Wide Web (WWW) - ( Παγκόσμιο Ιστό ) που αναπτύχθηκε από τον Tim Berners-Lee. Πρόκειται για ένα σύστημα διασύνδεσης πληροφοριών σε μορφή πολυμέσων ( multimedia ) που βρίσκονται αποθηκευμένες σε χιλιάδες υπολογιστές του Internet σε ολόκληρο τον κόσμο και παρουσίασης τους σε ηλεκτρονικές σελίδες, στις οποίες μπορεί να περιηγηθεί κανείς χρησιμοποιώντας το ποντίκι. Το γραφικό αυτό περιβάλλον έκανε την εξερεύνηση του Internet προσιτή στον απλό χρήστη. Παράλληλα, εμφανίζονται στο Internet διάφορα εμπορικά δίκτυα που ανήκουν σε εταιρείες παροχής υπηρεσιών Internet ( Internet Service Providers – ISP ) και προσφέρουν πρόσβαση στο Internet για όλους. Η ανακάλυψη του WWW σε συνδυασμό με την ευκολία απόκτησης πρόσβασης στο Internet προσέλκυσε έναν μεγάλο αριθμό καινούργιων χρηστών και έφερε την “έκρηξη” που παρακολουθήσαμε τα τελευταία χρόνια.

### 1.5 World Wide Web (WWW)

Η απλούστερη περιγραφή που θα μπορούσε να δώσει κανείς για το Internet, η οποία συνοψίζει ταυτόχρονα και την ουσία της ύπαρξης και της λειτουργίας του, είναι ότι αυτό αποτελεί έναν “χώρο” όπου αποθηκεύεται η πληροφορία. Από την ίδια τη φύση της όμως η πληροφορία δεν μπορεί να είναι στατική, διότι τότε χάνει την έννοιά της. Επεκτείνοντας λοιπόν την αρχική περιγραφή, ένας πιο ακριβής ορισμός θα έλεγε ότι το Internet είναι ένας χώρος αποθήκευσης αλλά ταυτόχρονα και ένα μέσο μετάδοσης, επεξεργασίας και μετεξέλιξης της πληροφορίας.

Κατά τη διάρκεια της εξέλιξης του δικτύου, εμφανίστηκαν και δοκιμάστηκαν πολλά εργαλεία, μέθοδοι και τρόποι αποθήκευσης και μετάδοσης της πληροφορίας, αρκετοί από τους οποίους πέτυχαν και κατάφεραν τελικά να καθιερωθούν. Ωστόσο το αποφασιστικό βήμα για την ολοκλήρωση (integration) της αναζήτησης της πληροφορίας στο Internet έγινε με την εισαγωγή μίας νέας υπηρεσίας που ονομάζεται World Wide Web ( WWW ) ή αλλιώς WEB. Ο Παγκοσμίως Ευρύς Ιστός, με απλά λόγια το Web, είναι ένα δίκτυο υπολογιστών βασισμένο στο Internet, που επιτρέπει στους χρήστες ενός υπολογιστή να έχουν πρόσβαση σε πληροφορίες που είναι αποθηκευμένες σε κάποιο άλλο υπολογιστή μέσω αυτού του παγκόσμιου δικτύου.

Πιο συγκεκριμένα, έστω ότι έχουμε πολλούς υπολογιστές διασυνδεδεμένους μέσω καλωδιώσεων σε ένα είδος δικτύου ευρείας περιοχής WAN (Wide Area Network), που είναι ουσιαστικά η ίδια ανάπτυξη που διακρίνουμε σε όλο το Internet . Κάθε ένας από αυτούς τους υπολογιστές καλείται εξυπηρετητής ή διανομέας Web (Web server) και περιέχει ένα σύνολο από δεδομένα υπερμέσων, τα λεγόμενα Web Documents, τα οποία δεικτοδοτούνται μεταξύ τους μέσω των διευθύνσεων των υπολογιστών στους οποίους περιέχονται συν των σχετικών τους εσωτερικών διευθύνσεων εντός των υπολογιστών αυτών. Πρόκειται, στην ουσία για ένα αλληλεπιδραστικό multimedia περιβάλλον που προσφέρει τη δυνατότητα στον χρήστη να δημιουργήσει εντυπωσιακά ηλεκτρονικά έγγραφα και να τα “δημοσιεύσει” στον Web, καθιστώντας τα έτσι προσβάσιμα για οποιονδήποτε ενδιαφερόμενο.

Ο Web είναι δομημένος σε μία αρχιτεκτονική client / server. Σε μία απλουστευμένη προσέγγιση, ο server όπως είδαμε περιέχει τα λεγόμενα Web Documents, μέσα στα οποία βρίσκονται οι διάφοροι σύνδεσμοι και “διοχετεύει” τις πληροφορίες προς τον client, χειριζόμενος ταυτόχρονα και τις αιτήσεις για επικοινωνία και διασύνδεση με άλλους απομακρυσμένους υπολογιστές (hosts). Στο Web ο client αποκαλείται Web Browser. Ένας Web browser (φυλλομετρητής ιστοσελίδων, πλοηγός Web, πρόγραμμα περιήγησης Web ή περιηγητής Ιστού) είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει και να αλληλεπιδρά με όλα τα Web Documents, που είναι συνήθως αναρτημένα σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το πρωτόκολλο που χρησιμοποιείται από τον Web για την εκτέλεση όλων αυτών των διεργασιών είναι το HTTP (Πρωτόκολλο Μεταφοράς Υπερκειμένου), ενώ η HTML ((Γλώσσα Επισήμανσης Υπερκειμένου) αποτελεί την γλώσσα κατασκευής των Web Documents, της πληροφορίας που ρέει στο Web.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 2

### ΠΡΟΓΡΑΜΜΑ ERASMUS PLUS

#### 2.1 Γενικές πληροφορίες σχετικά με το πρόγραμμα Erasmus+

Το Erasmus+ αποτελεί το πρόγραμμα της ΕΕ που αφορά τους τομείς της εκπαίδευσης, της κατάρτισης, της νεολαίας και του αθλητισμού για την περίοδο 2014-2020. Η εκπαίδευση, η κατάρτιση, η νεολαία και ο αθλητισμός μπορούν να συνεισφέρουν σημαντικά στην αντιμετώπιση των κοινωνικοοικονομικών αλλαγών, των κύριων προκλήσεων, στις οποίες καλείται να ανταποκριθεί η Ευρώπη ως το τέλος της δεκαετίας, καθώς και να υποστηρίξουν την εφαρμογή της Ευρωπαϊκής Πολιτικής Ατζέντας για την ανάπτυξη, την απασχόληση, την ισότητα και την κοινωνική ένταξη.

Η καταπολέμηση των αυξανόμενων επιπέδων ανεργίας –ιδίως μεταξύ των νέων– αποτελεί πλέον ένα από τα πιο επιτακτικά καθήκοντα των ευρωπαϊκών κυβερνήσεων. Πάρα πολλοί νέοι εγκαταλείπουν το σχολείο πρόωρα, αντιμετωπίζοντας υψηλό κίνδυνο ανεργίας και κοινωνικής περιθωριοποίησης. Ο ίδιος κίνδυνος απειλεί πολλούς ενήλικες με χαμηλό επίπεδο δεξιοτήτων. Οι τεχνολογίες αλλάζουν τον τρόπο λειτουργίας της κοινωνίας, και είναι αναγκαίο να διασφαλιστεί η βέλτιστη χρήση τους. Οι επιχειρήσεις στην ΕΕ πρέπει να γίνουν περισσότερο ανταγωνιστικές μέσω του ταλέντου και της καινοτομίας.

Η Ευρώπη χρειάζεται πιο συνεκτικές και χωρίς αποκλεισμούς κοινωνίες που επιτρέπουν στους πολίτες τους να διαδραματίζουν έναν ενεργό ρόλο στη δημοκρατική ζωή. Η εκπαίδευση και η εργασία στον τομέα της νεολαίας αποτελούν κλειδί για την προώθηση των κοινών ευρωπαϊκών αξιών της κοινωνικής ένταξης, της ενίσχυσης της διαπολιτισμικής κατανόησης και της αίσθησης του ανήκειν σε μία κοινότητα, καθώς και για την πρόληψη βίαιων ακραίων τάσεων. Το Πρόγραμμα Erasmus+ είναι ένα σημαντικό μέσο για την προώθηση της ένταξης των ατόμων που μειονεκτικό υπόβαθρο, συμπεριλαμβανομένων των νεοαφιχθέντων μεταναστών.

Μια άλλη πρόκληση αφορά την ανάπτυξη κοινωνικού κεφαλαίου μεταξύ των νέων, την ενδυνάμωση και την ανάπτυξη της δυνατότητας των νέων να συμμετέχουν ενεργά στην κοινωνία, σύμφωνα με τις διατάξεις της Συνθήκης της Λισαβόνας, όπου προβλέπεται ο στόχος της ΕΕ «να ενθαρρύνει τη συμμετοχή των νέων στον δημοκρατικό βίο της Ευρώπης». Η αντιμετώπιση του ζητήματος αυτού μπορεί επίσης να επιδιωχθεί μέσω δραστηριοτήτων μη τυπικής μάθησης που στοχεύουν στην ενίσχυση των δεξιοτήτων και των ικανοτήτων των νέων καθώς και της ενεργού συμμετοχής τους στα κοινά. Επιπλέον, είναι ανάγκη να δοθούν ευκαιρίες κατάρτισης και συνεργασίας στις οργανώσεις και στα άτομα που δραστηριοποιούνται στον τομέα της νεολαίας, προκειμένου να αναπτυχθεί ο επαγγελματισμός τους και η ευρωπαϊκή διάσταση της εργασίας στον εν λόγω τομέα.

Η καλή και αποδοτική λειτουργία των συστημάτων εκπαίδευσης και κατάρτισης και των πολιτικών για τη νεολαία παρέχουν στα άτομα τις δεξιότητες που απαιτεί η αγορά εργασίας και η οικονομία, επιτρέποντάς τους ταυτόχρονα να

διαδραματίσουν ενεργό ρόλο στην κοινωνία και να επιτύχουν προσωπική ολοκλήρωση. Οι μεταρρυθμίσεις στους τομείς της εκπαίδευσης, της κατάρτισης και της νεολαίας μπορούν να συντελέσουν στην εκπλήρωση αυτών των στόχων, έχοντας ως βάση ένα κοινό όραμα των υπευθύνων για τη χάραξη πολιτικής και των ενδιαφερόμενων μερών, ορθά στοιχεία και συνεργασία σε διάφορους τομείς και επίπεδα.

Το πρόγραμμα Erasmus+ έχει σχεδιαστεί με σκοπό να στηρίξει τις προσπάθειες των Χωρών του Προγράμματος για τη δια βίου αξιοποίηση των δυνατοτήτων του ταλέντου και των κοινωνικών εφοδίων της Ευρώπης μέσω της ενοποίησης της παρεχόμενης στήριξης προς την τυπική, τη μη τυπική και την άτυπη μάθηση σε ολόκληρο το φάσμα της εκπαίδευσης, της κατάρτισης και της ενασχόλησης με τη νεολαία. Το Πρόγραμμα ενισχύει επίσης τις δυνατότητες συνεργασίας και κινητικότητας με τις Χώρες Εταίρους, ιδίως στους τομείς της ανώτατης/τριτοβάθμιας εκπαίδευσης και της νεολαίας.

Σύμφωνα με ένα από τα νέα στοιχεία που εισήχθησαν στη Συνθήκη της Λισαβόνας, το Erasmus+ υποστηρίζει επίσης δραστηριότητες που αποσκοπούν στην ανάπτυξη της ευρωπαϊκής διάστασης στον αθλητισμό, προωθώντας τη συνεργασία μεταξύ των φορέων που είναι υπεύθυνοι για τον αθλητισμό. Το πρόγραμμα προωθεί επίσης τη δημιουργία και ανάπτυξη ευρωπαϊκών δικτύων παρέχοντας ευκαιρίες συνεργασίας μεταξύ των ενδιαφερόμενων μερών, καθώς και την ανταλλαγή και μετάδοση γνώσεων και τεχνογνωσίας σε διάφορους τομείς που συνδέονται με τον αθλητισμό και τη σωματική άσκηση. Ειδικότερα, αυτή η ενισχυμένη συνεργασία θα επιδράσει θετικά στην ανάπτυξη του δυναμικού του ανθρώπινου κεφαλαίου της Ευρώπης συμβάλλοντας στη μείωση του κοινωνικού και οικονομικού κόστους που συνεπάγεται η έλλειψη σωματικής άσκησης.

Το Πρόγραμμα υποστηρίζει δράσεις, συνεργασίες και εργαλεία που συνάδουν με τους στόχους της στρατηγικής «Ευρώπη 2020» και των εμβληματικών πρωτοβουλιών της, όπως οι πρωτοβουλίες «Νεολαία σε κίνηση» και «Ατζέντα για νέες δεξιότητες και θέσεις εργασίας». Το πρόγραμμα συμβάλλει επίσης, στην επίτευξη των στόχων του στρατηγικού πλαισίου για την Ευρωπαϊκή συνεργασία στην εκπαίδευση και την κατάρτιση, και της στρατηγικής της ΕΕ για τη νεολαία, μέσω των Ανοικτών Μεθόδων Συντονισμού.

Αυτή η επένδυση σε γνώσεις, δεξιότητες και ικανότητες θα ωφελήσει άτομα, ιδρύματα, οργανισμούς και την κοινωνία στο σύνολό της, συμβάλλοντας στην ανάπτυξη και διασφαλίζοντας δικαιοσύνη, ευημερία και κοινωνική ένταξη στην Ευρώπη και πέραν αυτής.

## 2.2 Γενικός στόχος

**Το πρόγραμμα Erasmus+ συμβάλλει στην επίτευξη:**

- των στόχων της στρατηγικής «Ευρώπη 2020», συμπεριλαμβανομένου του πρωταρχικού στόχου της που αφορά την εκπαίδευση.
- των στόχων του στρατηγικού πλαισίου για την ευρωπαϊκή συνεργασία στον τομέα της εκπαίδευσης και της κατάρτισης («ΕΚ 2020»), συμπεριλαμβανομένων των αντίστοιχων σημείων αναφοράς.

- της βιώσιμης ανάπτυξης των χωρών εταίρων στον τομέα της ανώτατης/τριτοβάθμιας εκπαίδευσης.
- των συνολικών στόχων του ανανεωμένου πλαισίου ευρωπαϊκής συνεργασίας στον τομέα της νεολαίας (2010- 2018).
- του στόχου της ανάπτυξης ευρωπαϊκής διάστασης στον αθλητισμό, και ιδίως στον μαζικό αθλητισμό, σύμφωνα με το πρόγραμμα εργασίας της ΕΕ για τον αθλητισμό.
- της προαγωγής των ευρωπαϊκών αξιών σύμφωνα με το άρθρο 2 της Συνθήκης για την Ευρωπαϊκή Ένωση.

### **2.3 Αναγνώριση και επικύρωση των δεξιοτήτων και των επαγγελματικών προσόντων**

Το πρόγραμμα Erasmus+ στηρίζει τα μέσα της ΕΕ που εξασφαλίζουν διαφάνεια και αναγνώριση των δεξιοτήτων και των προσόντων – ιδίως το Europass, το Youthpass, το ευρωπαϊκό πλαίσιο επαγγελματικών προσόντων (ΕΠΕΠ-EQF), το ευρωπαϊκό σύστημα μεταφοράς διδακτικών μονάδων (ECTS), το ευρωπαϊκό σύστημα ακαδημαϊκών μονάδων για την επαγγελματική εκπαίδευση και κατάρτιση (ECVET), το ευρωπαϊκό πλαίσιο αναφοράς για τη διασφάλιση της ποιότητας στην επαγγελματική εκπαίδευση και κατάρτιση (EQAVET), το ευρωπαϊκό μητρώο οργανισμών διασφάλισης της ποιότητας (EQAR), την ευρωπαϊκή ένωση για τη διασφάλιση της ποιότητας στην ανώτατη/τριτοβάθμια εκπαίδευση (ENQA), καθώς και τα πανευρωπαϊκά δίκτυα στον τομέα της εκπαίδευσης και κατάρτισης που στηρίζουν τα εν λόγω μέσα, ιδίως τα Εθνικά Κέντρα Ακαδημαϊκής Αναγνώρισης (NARIC), τα δίκτυα Euroguidance, τα Εθνικά Κέντρα Europass και τα εθνικά σημεία συντονισμού του ευρωπαϊκού πλαισίου επαγγελματικών προσόντων (ΕΠΕΠ).

Κοινός σκοπός αυτών των μέσων είναι να διασφαλίσουν την ευκολότερη αναγνώριση και την καλύτερη κατανόηση των δεξιοτήτων και των επαγγελματικών προσόντων, εντός και εκτός εθνικών συνόρων, σε όλα τα υποσυστήματα εκπαίδευσης και κατάρτισης καθώς και στην αγορά εργασίας, ανεξαρτήτως του αν έχουν αποκτηθεί μέσω επίσημης εκπαίδευσης και κατάρτισης ή άλλων εκπαιδευτικών εμπειριών (πχ. εργασιακή εμπειρία, εθελοντισμός, διαδικτυακή εκπαίδευση). Τα μέσα αυτά αποσκοπούν επίσης να διασφαλίσουν ότι οι πολιτικές για την εκπαίδευση, την κατάρτιση και τη νεολαία θα συμβάλουν περαιτέρω στην επίτευξη τόσο των στόχων της στρατηγικής «Ευρώπη 2020» για βιώσιμη και χωρίς αποκλεισμούς ανάπτυξη όσο και τους πρωταρχικούς της στόχους για την εκπαίδευση και την απασχόληση μέσω της πιο επιτυχούς ένταξης στην αγορά εργασίας και της κινητικότητας.

Για την εκπλήρωση των στόχων αυτών, τα διαθέσιμα μέσα θα πρέπει να είναι σε θέση να ανταποκρίνονται σε νέα φαινόμενα, όπως η διεθνοποίηση της εκπαίδευσης και η αυξανόμενη χρήση της ψηφιακής μάθησης, καθώς και να υποστηρίζουν τη δημιουργία ευέλικτων δυνατοτήτων μάθησης ανάλογα με τις ανάγκες και τους στόχους των εκπαιδευομένων. Τα μέσα αυτά ενδέχεται να χρειαστεί να εξελιχθούν επίσης στο μέλλον, οδηγώντας σε ενίσχυση της συνοχής



και απλοποίηση, που θα επιτρέπει οι εκπαιδευόμενοι και οι εργαζόμενοι να κινούνται ελεύθερα για επαγγελματικούς σκοπούς ή για περαιτέρω μάθηση.

## **2.4 Ποιος μπορεί να συμμετέχει στο πρόγραμμα Erasmus+**

Φυσικά πρόσωπα: φοιτητές, σπουδαστές, ασκούμενοι, μαθητευόμενοι, μαθητές, ενήλικοι εκπαιδευόμενοι, νέοι, εθελοντές, καθηγητές, εκπαιδευτικοί, εκπαιδευτές, εργαζόμενοι στον τομέα της νεολαίας ή σε οργανισμούς που δραστηριοποιούνται στους τομείς της εκπαίδευσης, της κατάρτισης και της νεολαίας συνιστούν τον βασικό πληθυσμό, στον οποίο απευθύνεται το πρόγραμμα. Ωστόσο, η επαφή των εν λόγω φυσικών προσώπων με το πρόγραμμα γίνεται μέσω οργανισμών, Ιδρυμάτων Ανώτατης/Τριτοβάθμιας Εκπαίδευσης, φορέων ή ομάδων που διοργανώνουν τέτοιου είδους δραστηριότητες. Συνεπώς, οι προϋποθέσεις πρόσβασης στο πρόγραμμα συνδέονται με τις ακόλουθες δύο κατηγορίες φορέων: τους «συμμετέχοντες» (φυσικά πρόσωπα που συμμετέχουν στο πρόγραμμα) και τους «συμμετέχοντες οργανισμούς» (συμπεριλαμβανομένων ομάδων τουλάχιστον τεσσάρων νέων ατόμων, που δραστηριοποιούνται στον τομέα της εργασίας με νέους, αλλά όχι απαραίτητα στο πλαίσιο οργανώσεων νεολαίας, γνωστών και ως άτυπων ομάδων νέων). Οι προϋποθέσεις συμμετοχής τόσο για τα φυσικά πρόσωπα (συμμετέχοντες) όσο και για τους συμμετέχοντες οργανισμούς εξαρτώνται από τη χώρα στην οποία είναι εγκατεστημένοι.

## ΚΕΦΑΛΑΙΟ 3

### PHP



#### 3.1 Βασικές γνώσεις PHP

Η γλώσσα PHP (είναι ανοικτό-ελεύθερο λογισμικό) μπορεί να εγκατασταθεί σχεδόν σε όλα τα λειτουργικά συστήματα όπως Windows, Linux, Mac OS X, Risc OS κλπ αλλά και υποστηρίζεται και από τα περισσότερους εξυπηρετητές ιστοσελίδων όπως ο Apache ή ο IIS. Η php μπορεί να λειτουργήσει είτε ως εγκατεστημένη μονάδα (module) στον εξυπηρετητή ιστοσελίδων είτε μέσω ενός επεξεργαστή CGI σεναρίων. Η PHP μπορεί να χρησιμοποιηθεί για εκτέλεση σεναρίων (scripts) από την πλευρά του απομακρυσμένου εξυπηρετητή ιστοσελίδων όπως γίνεται και με τα σεναρία CGI. Επίσης η php μπορεί να χρησιμοποιηθεί για είσοδο/έξοδο δεδομένων από τον χρήστη ή για την δυναμική δημιουργία σελίδων. Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script.

- Σενάρια PHP σε ένα απομακρυσμένο εξυπηρετητή ιστοσελίδων: Αυτή είναι η κύρια χρήση της γλώσσας PHP. Η γλώσσα PHP βρίσκεται εγκατεστημένη είτε ως module στο εξυπηρετητή ιστοσελίδων είτε εκτελείται μέσω ενός CGI σεναρίου και χρησιμοποιείται δια μέσου ενός φυλλομετρητή από τον υπολογιστή του πελάτη-χρήστη.
- Χρήση της PHP σε επίπεδο γραμμής εντολών (command line): Ένα σενάριο PHP μπορεί να εκτελεστεί μέσω του διερμηνέα PHP τοπικά στον υπολογιστή χωρίς να χρειάζεται να μεσολαβήσει ένας εξυπηρετητής ιστοσελίδων.
- Εφαρμογές με την γλώσσα PHP: Αν και δεν συνηθίζεται, μπορούν να φτιαχτούν προγράμματα με γραφικό περιβάλλον (π.χ. χρησιμοποιώντας το PHP-GTK) που να τρέχουν κατευθείαν στον υπολογιστή πελάτη τα οποία είναι ανεξάρτητα πλατφόρμας.

Το συντακτικό της PHP είναι βασισμένο στην σύνταξη της γλώσσας C, Java και Perl και είναι εύκολη στην εκμάθηση.

### 3.2 Βασικά στοιχεία της PHP

- Το πρόγραμμα ξεκινά και τελειώνει με `<?php` και `<?>`.
- Η εντολή **echo** τυπώνει ένα μήνυμα στην έξοδο (οθόνη).
- Η **\$name** είναι η μεταβλητή **name**. Στην γλώσσα php οι μεταβλητές ξεκινάνε με τον χαρακτήρα **\$**.
- Η εντολή `trim(fgets(STDIN))` διαβάσει από την καθορισμένη είσοδο (STDIN) μια συμβολοσειρά (string) με το όνομά μας το οποίο αποθηκεύεται στην μεταβλητή `name`.
- Ο τελεστής `.` ενώνει δύο ή περισσότερες συμβολοσειρές σε μια μεγαλύτερη. Η συμβολοσειρά "Γεια σου " συνενώνεται με την συμβολοσειρά που περιέχει η μεταβλητή `name` μαζί με την συμβολοσειρά `!\n`;
- Το `\n` είναι ο χαρακτήρας αλλαγής γραμμής (newline).
- Το `;` ερωτηματικό δηλώνει τέλος της εντολής.
- Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού ανοικτού λογισμικού ειδικά σχεδιασμένη για δημιουργία σελίδων Web με δυναμικό περιεχόμενο.
- Ο κώδικας PHP ενσωματώνεται μέσα σε HTML σελίδες και εκτελείται από τον Server (που είναι αποθηκευμένες) κάθε φορά που ο χρήστης επισκέπτεται τις σελίδες αυτές.
- Η εκτέλεση ενός κώδικα PHP έχει ως αποτέλεσμα τη παραγωγή κώδικα HTML ο οποίος στέλνεται τελικά στο φυλλομετρητή του χρήστη.

### 3.3 Πλεονεκτήματα

- Υψηλή απόδοση.
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, κ.ά.).
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες.
- Χαμηλό κόστος.
- Ευκολία μάθησης και χρήσης.
- Μεταφερσιμότητα.
- Διαθεσιμότητα του πηγαίου κώδικα.
- Αρκετά frameworks με καλό documentation

### 3.4 Βασικές αρχές λειτουργίας

- Η PHP τρέχει σε ένα διακομιστή.
- Ο Web browser ενός χρήστη κάνει μία http:// αίτηση για μια συγκεκριμένη php Web σελίδα.
- Ο Web διακομιστής λαμβάνει την αίτηση για την σελίδα php, ανακαλεί το αρχείο και το περνά στην μηχανή php για επεξεργασία.
- Η μηχανή php αρχίζει την ανάλυση του php κώδικα επικοινωνώντας αν υπάρχει ανάγκη με τον διακομιστή της βάσης δεδομένων.
- Η μηχανή php σταματά την εκτέλεση του προγράμματος επιστρέφοντας στον Web διακομιστή την τελική HTML σελίδα.
- Ο Web διακομιστής περνά την html σελίδα ξανά στο Web browser όπου ο χρήστης μπορεί να διαβάσει το αποτέλεσμα της php σελίδας.



Εικόνα 3.1: Γραφική απεικόνιση αλληλεπίδρασης PHP

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 4

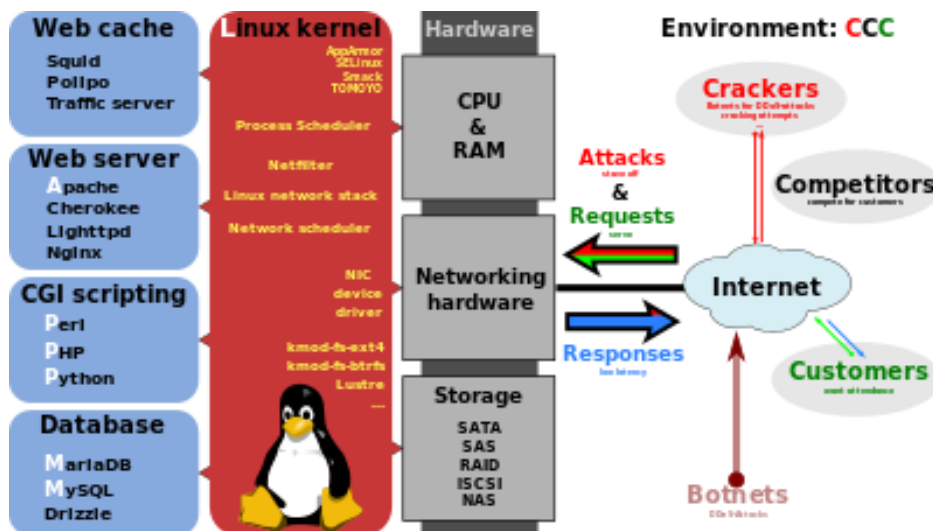
### MySQL

#### 4.1 Βασικές γνώσεις MySQL



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

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.



Εικόνα 4.1: Σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων

Η MySQL είναι αυτό που αποκαλούμε DBMS. (Database Management System – Σύστημα Διαχείρισης Βάσης Δεδομένων)

Από το όνομα μπορούμε να συμπεράνουμε ότι ένα DBMS είναι ένα πρόγραμμα που διαχειρίζεται την δημιουργία, συντήρηση και χρήση μια βάσης δεδομένων.

Τι είναι όμως μια βάση δεδομένων;

Όπως μας υποδεικνύει το όνομα της, μια βάση δεδομένων είναι μια οργανωμένη συλλογή δεδομένων. Με άλλα λόγια, συναφή δεδομένα που είναι αποθηκευμένα μαζί.

Για παράδειγμα, μια βίντεο λέσχη, μια βιβλιοθήκη και ένα κατάστημα θα μπορούσαν να χρησιμοποιούν βάσεις δεδομένων για τις ενοικιάσεις τους, τις πωλήσεις τους, τους πελάτες τους, την αποθήκη τους κτλ..

Αλλά πώς οργανώνονται τα δεδομένα μέσα στην βάση δεδομένων;

Οργανώνονται με την χρησιμοποίηση πινάκων.

Οι πίνακες είναι μικρότερες συλλογές συναφών δεδομένων.

Ας πάρουμε το παράδειγμα της βίντεο λέσχης. Έστω μια βάση δεδομένων με το όνομα video club.

Μέσα σε αυτή πρέπει να αποθηκευτούν πολλά διαφορετικά δεδομένα για τα μέλη, τις ταινίες, τις κατηγορίες των ταινιών, τις ενοικιάσεις κτλ..

Θα χρησιμοποιήσουμε πίνακες. Ένα πίνακα για τα μέλη, άλλο για τις ταινίες, άλλο για τις κατηγορίες των ταινιών κτλ.

Κάθε πίνακας θα περιέχει συγκεκριμένες πληροφορίες.

Ένας πίνακας αποτελείται από στήλες. Κάθε στήλη μπορεί να περιέχει μόνο ένα κομμάτι από τα δεδομένα.

Για παράδειγμα στον πίνακα members(μέλη) της βάσης

δεδομένων videoclub θα είχαμε κάποιες από τις παρακάτω στήλες:

first\_name (όνομα), last\_name (επίθετο), birth\_date (ημερομηνία γέννησης),

registration\_date (ημερομηνία εγγραφής), address (διεύθυνση), city (πόλη),

post\_code (ταχυδρομικός κώδικας) κτλ.

## **ΚΕΦΑΛΑΙΟ 5**

### **ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

#### **5.1 Τι είναι βάσεις δεδομένων**

Μια βάση δεδομένων, με απλά λόγια, είναι ένα σύνολο από πληροφορίες που έχουν οργανωθεί ώστε η χρήση αυτών των πληροφοριών να είναι γρήγορη και αποτελεσματική. Οι βάσεις δεδομένων βρίσκονται παντού: στην εργασία, στο σπίτι, στα σχολεία ακόμα και σε τόπους διασκέδασης.

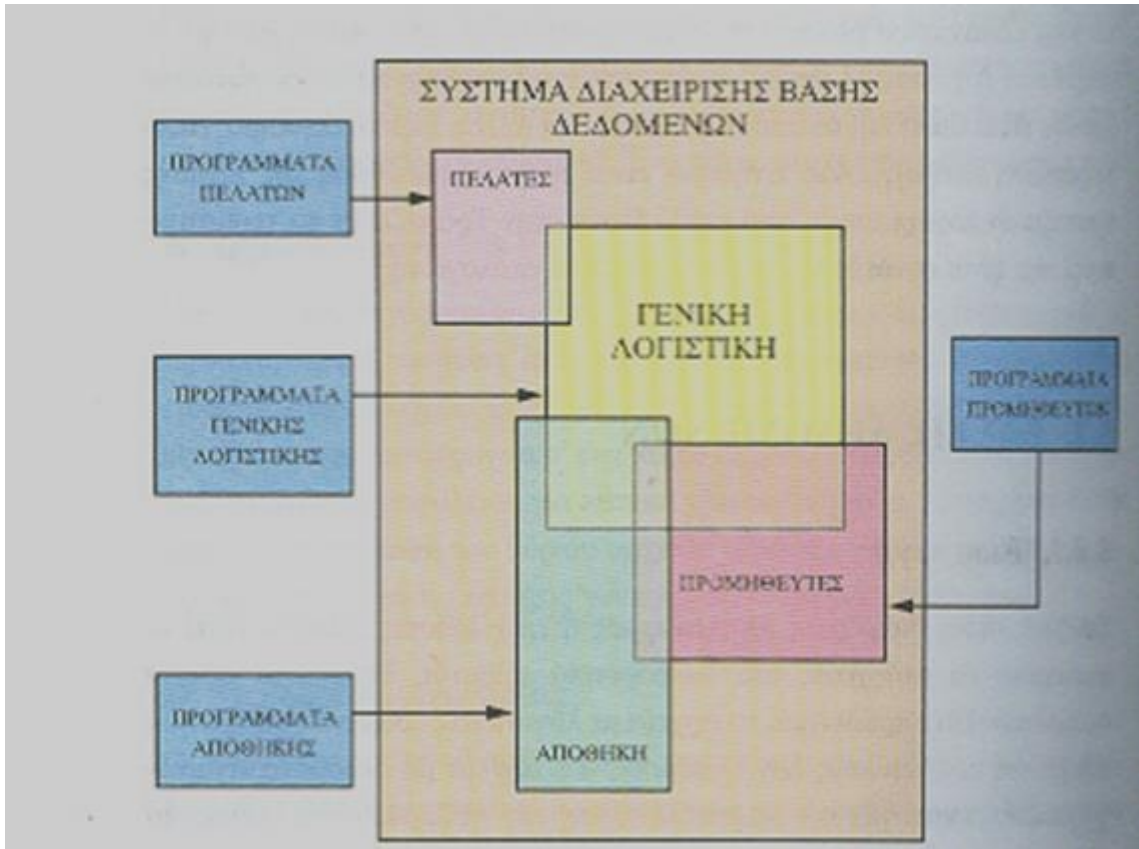
#### **5.2 Ορισμός βάσεων δεδομένων**

Μία Βάση Δεδομένων είναι μία οργανωμένη συλλογή από συσχετιζόμενα δεδομένα που χρησιμοποιούνται από όλες τις εφαρμογές του Οργανισμού ή της Επιχείρησης.

#### **5.3 Χρήση της βάσης δεδομένων**

Η χρησιμοποίηση Βάσεων Δεδομένων (Databases) εξασφαλίζει ένα γενικό τρόπο αποθήκευσης των δεδομένων. Τα δεδομένα είναι ολοκληρωμένα (integrated) και καταμερισμένα (shared) έτσι ώστε, αφενός μεν τα πλεονάζοντα (redundant) δεδομένα να αποθηκεύονται όσο το δυνατόν λιγότερες φορές, αφετέρου δε να είναι προσπελάσιμα, από διάφορους χρήστες, για ποικίλες εφαρμογές. Βασικά πλεονεκτήματα από την οργάνωση των δεδομένων της Βάσης με τον τρόπο αυτό είναι ότι μειώνεται ο χώρος αποθήκευσης στο δίσκο καθώς ο χρόνος για ενημέρωση των δεδομένων.





Εικόνα 5.1: Σύστημα Διαχείρισης Βάσης Δεδομένων

#### 5.4 Πρότυπα βάσεων δεδομένων

Η Βάση Δεδομένων οργανώνει τα δεδομένα σύμφωνα με κάποιο πρότυπο που χαρακτηρίζει τη δομή της.

Οι περισσότερο γνωστές δομές των Βάσεων Δεδομένων είναι:

- Η ιεραρχική (Hierarchical)
- Η δικτυωτή (Network)
- Η σχεσιακή (Relational)

Κάθε δομή έχει διαφορετικό τρόπο οργάνωσης και χρήσης των δεδομένων. Οι ιεραρχικές και δικτυωτές δομές χρησιμοποιούνται κυρίως σε μεγάλους και σε μεσαίου μεγέθους υπολογιστές. Η σχεσιακή δομή, που είναι και η νεότερη, είναι δυνατόν να χρησιμοποιηθεί σε κάθε τύπο υπολογιστή.

Η σχεσιακή δομή έχει πολλά πλεονεκτήματα σε σχέση με την ιεραρχική και τη δικτυωτή, όπως:

- Παρέχει διαφάνεια και σαφήνεια. Διευκολύνει το χρήστη στην κατανόηση και στην υλοποίηση.
- Είναι εύκολη η μετατροπή κάθε άλλης δομής σε σχεσιακή.
- Είναι εύκολη η αναπαραγωγή και υλοποίηση νέων σχέσεων για ειδικές εφαρμογές.
- Η αναζήτηση δεδομένων είναι πολύ ταχύτερη από κάποια άλλη δομή που απαιτεί μία σειρά από δείκτες.
- Η σχεσιακή δομή μπορεί να τροποποιηθεί πολύ ευκολότερα από την ιεραρχική ή τη δικτυωτή και αυτό προσδίνει ευελιξία στο σύστημα.

### **5.5 Βασικές Έννοιες στις Βάσεις Δεδομένων**

Δύο θεμελιώδεις έννοιες των Βάσεων Δεδομένων αναφέρονται στα πρότυπα δεδομένων και στην ανεξαρτησία δεδομένων:

- Πρότυπα Δεδομένων
- Ανεξαρτησία Δεδομένων

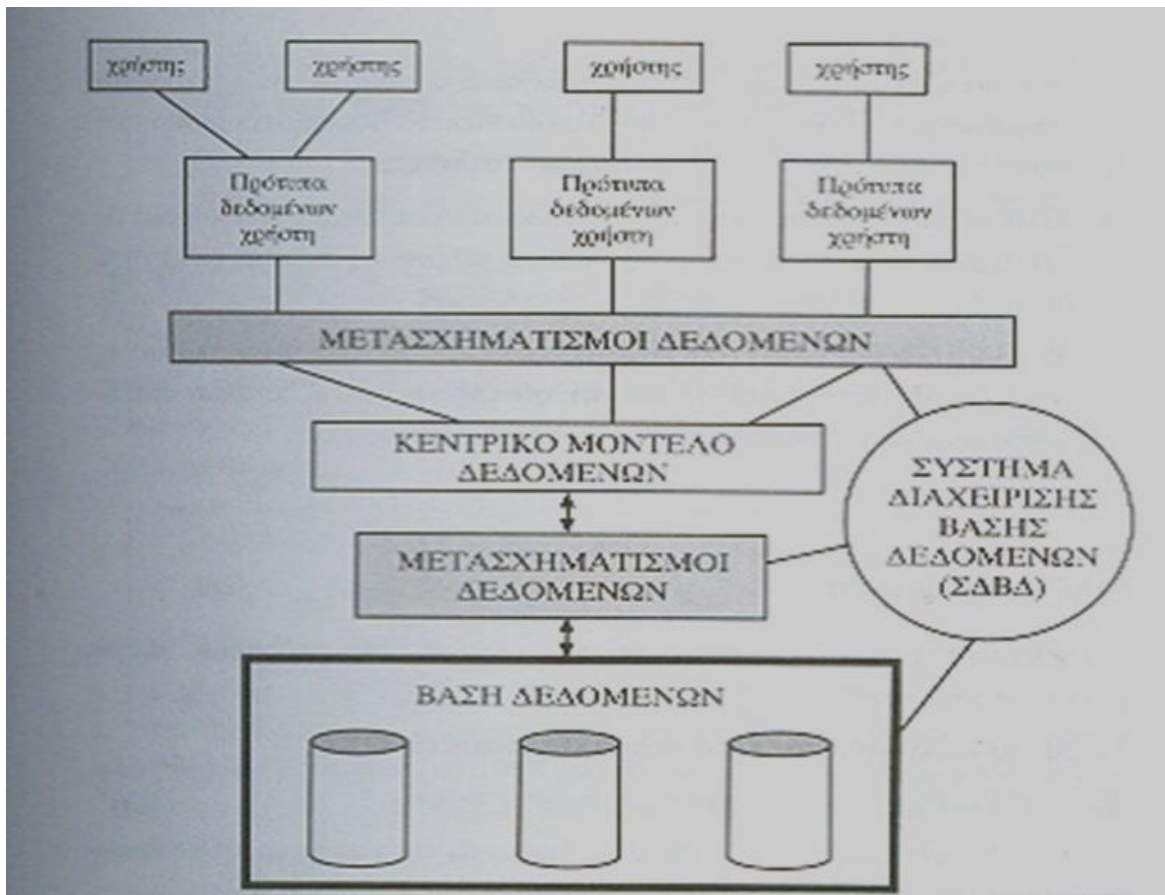
### **5.6 Πρότυπα Δεδομένων**

Με το πρότυπο δεδομένων κάθε μεμονωμένος χρήστης έχει τη δυνατότητα να λαμβάνει τα δεδομένα εκείνα που χρειάζεται σε κάθε περίπτωση. Κάθε χρήστης έχει το δικό του πρότυπο δεδομένων το οποίο μπορεί να παίρνει δεδομένα από πολλά φυσικά αρχεία.

### **5.7 Ανεξαρτησία Δεδομένων**

Είναι απαραίτητο τα προγράμματα εφαρμογών να είναι ανεξάρτητα από τη φυσική οργάνωση των δεδομένων, διότι:

- Σπάνια τα δεδομένα αυτά παραμένουν αμετάβλητα.
- Τα δεδομένα μεταβάλλονται και επομένως ενημερώνονται συνεχώς.
- Ακόμη νέα δεδομένα προστίθενται και έτσι ο συνολικός όγκος τους συνεχώς αυξάνεται.
- Τέλος προκύπτουν νέες απαιτήσεις δομής αποθήκευσης και μεθόδων προσπέλασης δεδομένων. Το αντίθετο θα απαιτούσε διορθώσεις σ' όλα τα προγράμματα εφαρμογών, με αποτέλεσμα να έχουμε καθυστερήσεις λάθη και να αυξάνεται το κόστος επεξεργασίας.



Εικόνα 5.2: Σύστημα Διαχείρισης Βάσης Δεδομένων

### 5.8 Πλεονεκτήματα των Βάσεων Δεδομένων

- Επεξεργασία των δεδομένων κατά ένα ολοκληρωμένο και ενιαίο τρόπο.
- Προσπέλαση στα δεδομένα από διάφορους χρήστες και εφαρμογές.
- Εύκολη ενημέρωση και αποφυγή επανάληψης των δεδομένων.
- - Ευελιξία στην αξιοποίηση της πληροφορίας αφού τα δεδομένα είναι συγκεντρωμένα και προσπελάσιμα από τους χρήστες.

### 5.9 Μειονεκτήματα των Βάσεων Δεδομένων

- Δαπανηρή προμήθεια, οργάνωση και η συντήρηση.
- Απαιτήσεις σε υλικό, όπως πρόσθετη μνήμη, πρόσθετες περιφερειακές μονάδες και, πολλές φορές, ανάγκη προμήθειας νέου ισχυρότερου υπολογιστή. Επίσης απαιτεί στελέχωση με εξειδικευμένο προσωπικό.
- Κίνδυνος πρόσβασης από μη εξουσιοδοτημένα άτομα.
- Υπερβολική ή μη ελεγχόμενη πληροφόρηση.

## 5.10 Αρχιτεκτονική και Σχεδίαση - Αρχές Σωστής Σχεδίασης

Για να λειτουργεί μια βάση δεδομένων σωστά και αποδοτικά πρέπει, τόσο η ίδια, όσο και το Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) που τη φιλοξενεί να είναι σωστά σχεδιασμένα. Αυτό ανάγεται σε μια σειρά από απαιτήσεις :

- Η βάση δεδομένων να μην περιλαμβάνει **περιττά δεδομένα (redundant data)**. Τα ίδια δεδομένα δεν θα πρέπει να καταχωρούνται στη βάση δύο φορές. Εάν συμβαίνει κάτι τέτοιο, τότε ανακύπτουν δυο σοβαρά προβλήματα. Το πρώτο και προφανές πρόβλημα, είναι ότι σπαταλούμε άσκοπα αποθηκευτικό χώρο. Το δεύτερο και σοβαρότερο πρόβλημα, είναι ότι υπάρχει ο κίνδυνος δημιουργίας **ασυνεπών δεδομένων (inconsistent data)**. Πράγματι, εάν κρατάμε δύο φορές την ίδια πληροφορία και η πληροφορία αυτή σε κάποια χρονική στιγμή υποστεί κάποιο είδος επεξεργασίας – για παράδειγμα την τροποποιήσουμε ή τη διαγράψουμε – τότε η επεξεργασία αυτή θα πρέπει να εφαρμοσθεί και στις δύο καταχωρήσεις που αφορούν το ίδιο δεδομένο, διότι διαφορετικά, η βάση θα περιέχει δεδομένα που δεν είναι συνεπή. Για το λόγο αυτό, ένας από τους πρώτους ελέγχους που πραγματοποιούμε στη βάση αμέσως μετά το σχεδιασμό της, είναι ο έλεγχος παρουσίας επαναλαμβανόμενων πεδίων, και η απομάκρυνσή τους, εφόσον υπάρχουν.
- Η βάση δεδομένων πρέπει να είναι σχεδιασμένη έτσι ώστε τα δεδομένα που περιλαμβάνει να ανακτώνται εύκολα και γρήγορα. Μια κακή σχεδίαση της δομής της βάσης, οδηγεί στη δημιουργία μιας βάσης η οποία είναι εξαιρετικά δυσκίνητη και αναποτελεσματική. Η σωστή σχεδίαση μιας βάσης δεδομένων, επιτυγχάνεται με εφαρμογή επί της δομής της βάσης, μιας τεχνικής, η οποία ονομάζεται **κανονικοποίηση (normalization)**. Η περιγραφή αυτής της τεχνικής παρουσιάζεται διεξοδικά σε παρακάτω.
- Το ΣΔΒΔ να χαρακτηρίζεται από **σύστημα ασφάλειας (security system)** που να απαγορεύει την πρόσβαση στα δεδομένα μη εξουσιοδοτημένων ατόμων. Αυτό ισχύει κυρίως όπως θα δούμε, σε μεγάλες βάσεις δεδομένων με πολλούς χρήστες, και η τεχνική που συνήθως εφαρμόζεται, συνίσταται στον καθορισμό **ομάδων χρηστών (user groups)** με διαφορετικά δικαιώματα πρόσβασης στον καθένα από αυτούς. Ο κάθε χρήστης λαμβάνει ένα κωδικό πρόσβασης (password) και τα καθήκοντα που μπορεί να επιτελέσει είναι εντελώς συγκεκριμένα και καθορισμένα εκ των προτέρων.
- Το ΣΔΒΔ να παρέχει **έλεγχο σε ταυτόχρονες προσπελάσεις (concurrency control)** πάνω στα ίδια δεδομένα.
- Το ΣΔΒΔ να διαθέτει σύστημα δημιουργίας **αντιγράφων ασφαλείας (backups)** των δεδομένων που είναι καταχωρημένα σε αυτή. Η ταυτόχρονη αποθήκευση των δεδομένων σε περισσότερους από ένα δίσκους, είναι μια εργασία επιβεβλημένη, προκειμένου να είναι δυνατή η ανάκτησή τους σε περιπτώσεις κατάρρευσης της βάσης για οποιοδήποτε λόγο.

Ειδικότερα, μια καλά σχεδιασμένη βάση δεδομένων σε ένα λειτουργικό και αξιόπιστο ΣΔΒΔ διακρίνεται από τα εξής χαρακτηριστικά:

- Είναι απλή και κατανοητή ως προς τη δομή της.
- Διασφαλίζεται η ακεραιότητα δεδομένων (data integrity).
- Αντεπεξέρχεται σε αναμενόμενα (planned) αλλά και σε μη αναμενόμενα (ad hoc) ερωτήματα (queries).
- Ανταποκρίνεται στο σκοπό για τον οποίο δημιουργείται και να έχει βιωσιμότητα.
- Ελαχιστοποιείται η εξάρτησή της από τις εφαρμογές που θα την προσπελαίνουν.

### **5.11 Αρχιτεκτονικές Συστημάτων Διαχείρισης Βάσεων Δεδομένων - Βαθμίδες και Αρχιτεκτονικές**

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων είναι εφαρμογές λογισμικού που λειτουργούν συνήθως μέσω ενός δικτύου ή του διαδικτύου. Οι κοινοί τρόποι σχεδιασμού του λογισμικού σε ό,τι αφορά τις επιμέρους μονάδες (modules) και τη μεταξύ τους επικοινωνία χαρακτηρίζονται ως αρχιτεκτονικά μοντέλα ή αρχιτεκτονικές.

Στη συνέχεια θα εξετάσουμε τις βασικές αρχιτεκτονικές των σύγχρονων ΣΔΒΔ που κατηγοριοποιούνται ανάλογα με τις βαθμίδες (tiers) που εμπλέκονται. Ο όρος των βαθμίδων ανάγεται στη λογική αρχιτεκτονική P-A-D από τα αρχικά των λέξεων Παρουσίαση-Εφαρμογή-Δεδομένα (Presentation-Application-Data), και βασίζεται στην αλληλεπίδραση των τριών αυτών λογικών διαδικασιών:

- Παρουσίαση –Presentation : γνωστή και ως διεπαφή χρήστη (user interface)
- Εφαρμογή – Application : γνωστή και ως επεξεργασία ή λογική εφαρμογής
- Δεδομένα – Data : γνωστή και ως διαχείριση

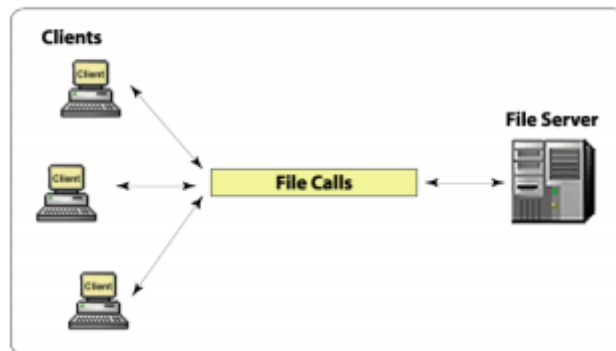
Η ανάπτυξη μεγάλων εφαρμογών επικεντρωμένων στην ανταλλαγή δεδομένων μέσω δικτύου οδήγησε στο περιβάλλον Πελάτη/Εξυπηρετητή (Client/Server) στη δεκαετία του 1980. Σε αυτού του περιβάλλοντος τις αρχιτεκτονικές υπάρχει μια κεντρική εφαρμογή που λέγεται Εξυπηρετητής (Server) και με την οποία συνδέονται και ανταλλάσσουν πληροφορίες άλλες εφαρμογές που βρίσκονται στους υπολογιστές των χρηστών και χαρακτηρίζονται ως Πελάτες (Clients). Η ανταλλαγή της πληροφορίας γίνεται μέσω συναλλαγών (transactions), δηλαδή μέσω καλά ορισμένων αιτημάτων (requests) και αποκρίσεων (responses). Το περιβάλλον Πελάτη/Εξυπηρετητή αφορά στη φυσική αρχιτεκτονική και εμφανίζει επιμέρους αρχιτεκτονικές. Όταν το μεγαλύτερο μέρος της επεξεργασίας (λογική της εφαρμογής) γίνεται στον Εξυπηρετητή και ο Πελάτης αναλαμβάνει κυρίως την παρουσίαση αποτελεσμάτων της διαδικασίας, τότε έχουμε την περίπτωση Αδύνατου Πελάτη (Thin Client) – Παχύ Εξυπηρετητή (Fat Server), ενώ όταν ο Πελάτης αναλαμβάνει το μεγαλύτερο μέρος της επεξεργασίας μιλάμε για Παχύ Πελάτη (Fat Client) – Αδύνατο Εξυπηρετητή (Thin Server). Στην ειδική περίπτωση

του Διαδικτύου, όπου αποκλειστικά και μόνο το επίπεδο της παρουσίασης είναι στο φυλλομετρητή (browser), ενώ το σύνολο της επεξεργασίας είναι στον Εξυπηρετητή, μιλάμε για Υπέρ-Παχύ Εξυπηρετητή (Super-Fat Server).

### 5.12 Αρχιτεκτονική μιας βαθμίδας (1-tier)

Η απλούστερη αρχιτεκτονική είναι η αρχιτεκτονική μιας βαθμίδας (1-tier) στην οποία ο υπολογιστής που φιλοξενεί την εφαρμογή είναι ο ίδιος με το οποίο ο χρήστης επικοινωνεί με την εφαρμογή. Τυπική περίπτωση είναι η ανάπτυξη μιας βάσης στο ΣΔΒΔ MS Access, όπου αποθηκεύει τα δεδομένα της στον τοπικό δίσκο ενός προσωπικού υπολογιστή και ταυτόχρονα παρέχει τη δυνατότητα προσπέλασης των δεδομένων μέσω διεπαφής που αποθηκεύεται στον ίδιο υπολογιστή.

Μια επέκταση αυτής της αρχιτεκτονικής είναι όταν ένας υπολογιστής – εξυπηρετητής αρχείων (file server) φιλοξενεί τα αρχεία αποθήκευσης του ΣΔΒΔ (αλλά όχι κατ' ανάγκη το ίδιο) και βρίσκεται σε τοπικό δίκτυο με άλλους υπολογιστές. Οι άλλοι υπολογιστές έχουν δυνατότητα ΑΜΕΣΗΣ προσπέλασης των αρχείων του με τη μορφή διαμοιρασμού (filesharing) και ΟΛΟΙ έχουν εγκατεστημένη την εφαρμογή του ΣΔΒΔ. Με αυτόν τον τρόπο διαφορετικοί υπολογιστές έχουν πρόσβαση στην ίδια βάση δεδομένων, αλλά με αρχιτεκτονική ίδια με αυτήν του ενός υπολογιστή.



Εικόνα 5.3: Μοντέλο Client/Server 1-tier

#### Πλεονεκτήματα:

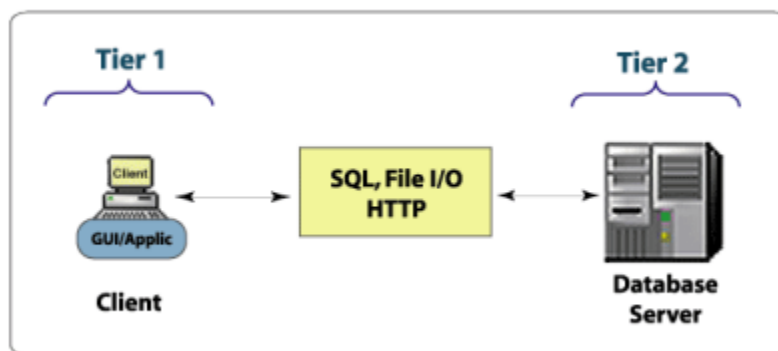
- Διαμοιρασμός δεδομένων σε πολλαπλούς χρήστες γίνεται με απλό τρόπο αν και οι δυνατότητες είναι περιορισμένες.
- Το κόστος αποθήκευσης είναι χαμηλό και μοιράζεται στους διασυνδεδεμένους χρήστες.
- Επειδή η ίδια εφαρμογή ΣΔΒΔ εγκαθίσταται πολλές φορές, ενδεχομένως επιτυγχάνεται χαμηλότερη τιμή αγοράς ανά εγκατάσταση (site license).

### Μειονεκτήματα:

- Μικρή δυνατότητα κλιμάκωσης.
- Αδυναμία ταυτόχρονης χρήσης.
- Προβλήματα ασφάλειας

### 5.13 Αρχιτεκτονική δύο βαθμίδων (2-tier)

Πρόκειται για την περίπτωση που ένα κεντρικό σύστημα, πολλές φορές με αποκλειστική αρμοδιότητα (server), παρέχει τη λειτουργικότητα για πολλά συστήματα χρηστών (clients). Το ΣΔΒΔ είναι στον εξυπηρετητή, με τον οποίο επικοινωνούν οι εφαρμογές –πελάτες.



Εικόνα 5.4: Μοντέλο Client/Server 2-tier

### Πλεονεκτήματα

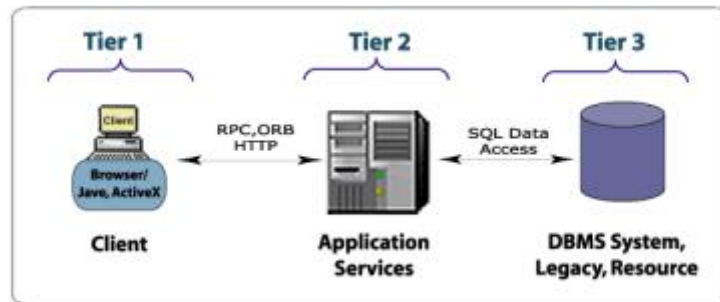
- Κλιμάκωση, οριζόντια (αύξηση πελατών) και κάθετη (αύξηση εξυπηρετητών).
- Αποτελεσματική χρήση των υπολογιστικών πόρων με καλό επίπεδο ασφάλειας.
- Εύκολη ενσωμάτωση νέας τεχνολογίας.
- Μαζί με τον client μπορεί να υπάρχουν και άλλες εφαρμογές.

### Μειονεκτήματα

- Η client εφαρμογή εγκαθίσταται σε πολλά συστήματα, πράγμα που προκαλεί δυσκολία αλλαγής της (π.χ. αναβάθμισης) και, για το λόγο αυτό, το κόστος συντήρησης αυξάνεται σημαντικά.

### 5.14 Αρχιτεκτονική πολλαπλών βαθμίδων (n-tier)

Η λειτουργικότητα της εφαρμογής του ΣΔΒΔ χωρίζεται σε διαφορετικά επίπεδα που διασυνδέονται λογικά και μοιράζονται πληροφορίες. Κάθε ενδιάμεσο επίπεδο αποτελεί ένα μοντέλο client/server. Η πιο συνήθης υλοποίηση είναι αυτή των τριών βαθμίδων (3-tier), 20 Εισαγωγή στις Σχισιακές Βάσεις Δεδομένων για Οικονομολόγους όπου υπάρχει σαφής φυσικός διαχωρισμός της επεξεργασίας από τη βάση δεδομένων και την παρουσίαση.



Εικόνα 5.5: Μοντέλο Client/Server 3-tier

#### Πλεονεκτήματα

- Αυξημένη ασφάλεια, αφού ο χρήστης δεν προσπελαύνει άμεσα τη βάση δεδομένων.
- Αυξημένη ευελιξία και έλεγχος λόγω των ενδιάμεσων επιπέδων και, επομένως, καλύτερη απόδοση.
- Ελαχιστοποιημένη διαχείριση σε σχέση με το μοντέλο client/server.

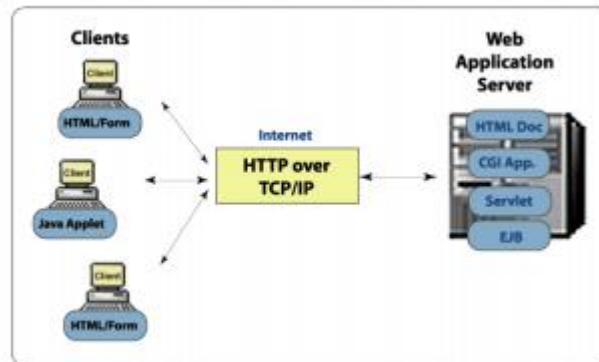
#### Μειονεκτήματα

- Πολυπλοκότητα της σχεδίασης.
- Αυξημένο αρχικό κόστος υλοποίησης.



## 5.15 Αρχιτεκτονική διαδικτύου

Στην αρχιτεκτονική πολλαπλών βαθμίδων εντάσσεται και η αρχιτεκτονική διαδικτύου. Σε αυτήν την περίπτωση το τελευταίο μέρος προς τον Πελάτη υλοποιείται μέσω του πρωτοκόλλου http και η μόνη απαίτηση από αυτόν είναι η εφαρμογή φυλλομετρητή (περίπτωση Υπέρ-Παχύ Εξυπηρετητή).



Εικόνα 5.6: Μοντέλο Client/Server Διαδικτύου

### Πλεονεκτήματα

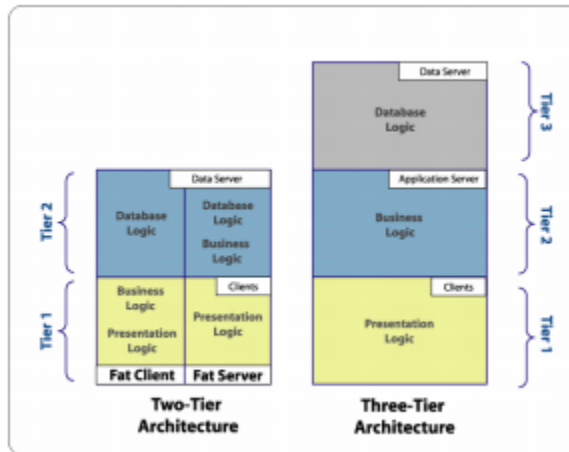
- Αυξημένη ασφάλεια, αφού ο χρήστης δεν προσπελαίνει άμεσα τη βάση δεδομένων.
- Αυξημένη ευελιξία και έλεγχος και, επομένως, καλύτερη απόδοση.
- Ελαχιστοποιημένη διαχείριση σε σχέση με το μοντέλο client/server
- Χαμηλές απαιτήσεις για την πλευρά του χρήστη

### Μειονεκτήματα

- Αυξημένοι κίνδυνοι που αντιμετωπίζονται μέσω ισχυρών πολιτικών ασφάλειας (firewalls, λογισμικό ανίχνευσης εισβολών, κ.ά.)

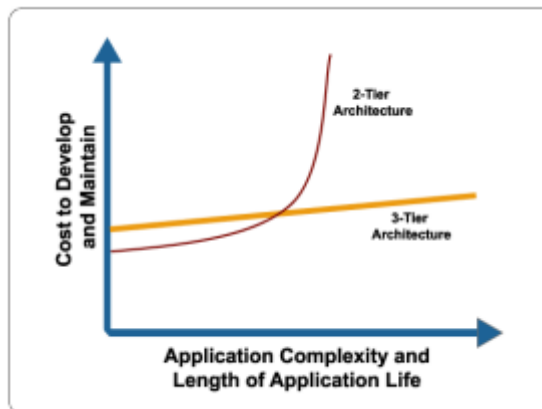
### 5.16 Σύγκριση αρχιτεκτονικών 2-βαθμίδων και 3-βαθμίδων

Η βασική διαφορά μεταξύ των 2 και των 3 βαθμίδων είναι ο τρόπος που υλοποιείται το μέρος της Εταιρικής Λογικής (Business Logic), δηλαδή η λογική της εφαρμογής κατά P-A-D. Έτσι, στην αρχιτεκτονική 2 βαθμίδων η Εταιρική Λογική αναπτύσσεται και στις δύο βαθμίδες, και στις δύο πλευρές του μοντέλου Πελάτη-Εξυπηρετητή. Στο παρακάτω σχήμα δίδεται παραστατικά η διαφορά μεταξύ των αρχιτεκτονικών 2-tier και 3-tier:



Εικόνα 5.7: Σύγκριση αρχιτεκτονικών 2-tier και 3-tier

Συγκρίνοντας τις δύο αρχιτεκτονικές ως προς το κόστος ανάπτυξης και συντήρησης, τα συστήματα 2 βαθμίδων μπορεί να κλιμακώσουν υπερβολικά το κόστος καθώς αυξάνει η πολυπλοκότητα των εφαρμογών δηλ. η επεξεργασία και ο χρόνος ζωής της εφαρμογής.



Εικόνα 5.8: Κλιμάκωση κόστους αρχιτεκτονικών 2-tier και 3-tier

### 5.17 Σχεδίαση Βάσεων Δεδομένων - Στάδια Ζωής Βάσεων Δεδομένων

Στον κύκλο ζωής μιας βάσης δεδομένων μπορούμε να διακρίνουμε τρία στάδια. Το πρώτο Στάδιο προηγείται χρονικά και πρέπει να ολοκληρωθεί για να προχωρήσουμε στα επόμενα, ενώ για τα Στάδια 2 και 3 υπάρχει χρονική αλληλοεπικάλυψη. Παρακάτω αναλύονται λεπτομερώς τα τρία στάδια

- Στάδιο 1: Σχεδίαση και υλοποίηση της βάσης: Στο στάδιο αυτό αναλύονται οι απαιτήσεις που έχουμε για τη βάση δεδομένων, πραγματοποιείται ο σχεδιασμός και υλοποιείται η βάση δεδομένων χρησιμοποιώντας τα κατάλληλο λογισμικό και εξειδικευμένο προσωπικό. Το στάδιο αυτό είναι κρίσιμο γιατί δεσμεύει την εξέλιξη της βάσης και πραγματοποιείται άπαξ. Λάθη κατά τη σχεδίαση είναι πολύ πιθανόν να οδηγήσουν σε αναποτελεσματική διαχείριση των δεδομένων, κάτι που μακροπρόθεσμα θα κοστίσει, τόσο χρόνο, όσο και χρήμα.
- Στάδιο 2: Καταχώρηση των δεδομένων στη βάση (data entry): Η καταχώρηση δεδομένων δεν απαιτεί την απασχόληση εξειδικευμένου προσωπικού. Γίνεται είτε αυτοματοποιημένα, είτε με συμπλήρωση φορμών σε ειδικές εφαρμογές διεπαφής με το χρήστη (user interfaces – UI).
- Στάδιο 3: Διαχείριση της βάσης δεδομένων και επεξεργασία των δεδομένων: Η διαχείριση των δεδομένων της βάσης πραγματοποιείται ανά πάσα στιγμή μετά την υλοποίησή της και εφαρμόζεται στα δεδομένα εκείνα που έχουν ήδη καταχωρηθεί. Η κυριότερη μορφή διαχείρισης των δεδομένων αφορά στην προβολή δεδομένων που πληρούν συγκεκριμένα κριτήρια αναζήτησης. Τα κριτήρια αυτά μπορεί, είτε να έχουν τεθεί εξ αρχής σε μια εφαρμογή που προσπελαύνει τη βάση δεδομένων (3- tier) και επιστρέφει το αποτέλεσμα με τυποποιημένη μορφοποίηση σε μορφή αναφοράς, είτε να τίθεται κατά τη διάρκεια εκτέλεσης της εφαρμογής. Άλλη μορφή διαχείρισης, εξ ίσου σημαντική, είναι η αλλαγή των δεδομένων (τροποποίηση / ενημέρωση / διαγραφή).

Το ενδιαφέρον μας εστιάζεται στο πρώτο Στάδιο και ιδιαίτερα στη σχεδίαση των βάσεων δεδομένων.

## 5.18 Επίπεδα Σχεδίασης Βάσεων Δεδομένων

Η σχεδίαση μιας βάσης δεδομένων μπορεί να χωριστεί γενικά σε τρία διακριτά εννοιολογικά επίπεδα: το φυσικό επίπεδο, το λογικό επίπεδο, και το επίπεδο όψης.

Στο **φυσικό επίπεδο (physical level)**, σχεδιάζεται και αναλύεται ο τρόπος με τον οποίο τα δεδομένα της βάσης θα αποθηκεύονται στο ΣΔΒΔ. Στο επίπεδο αυτό, καθορίζονται, μεταξύ άλλων, οι τύποι δεδομένων που θα χρησιμοποιηθούν.

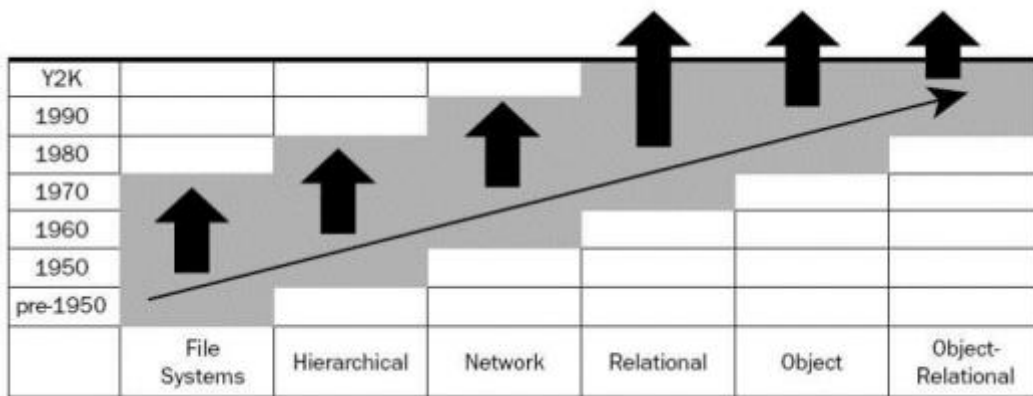
Στο **εννοιολογικό ή λογικό επίπεδο (conceptual / logical level)**, καθορίζεται ο τρόπος που τα δεδομένα και οι σχέσεις που υφίστανται ανάμεσά τους θα μοντελοποιηθούν στη βάση του συστήματος.

Τέλος, στο επίπεδο όψης (view level), καθορίζεται ο τρόπος που ένα μέρος των δεδομένων θα προβάλλεται στο χρήστη σύμφωνα με καθορισμένα κριτήρια που εκείνος θέτει με άμεσο ή έμμεσο τρόπο.

Εμείς επικεντρωνόμαστε στο λογικό επίπεδο γιατί εκεί γίνεται κυρίως η μοντελοποίηση της βάσης δεδομένων.

## 5.19 Μοντέλα Σχεδίασης Εννοιολογικού - Λογικού Επιπέδου

Τα μοντέλα σχεδίασης εννοιολογικού-λογικού επιπέδου δεν αφορούν απλά στους τρόπους οργάνωσης δεδομένων, αλλά προχωρούν στο να ορίσουν και τις πράξεις που τελούνται στα δεδομένα. Έτσι, για το σχεσιακό μοντέλο, για παράδειγμα, υφίστανται οι τελεστές της επιλογής (select), της προβολής (project), της σύζευξης (join) και της διαίρεσης (division). Η ιστορική εξέλιξη των μοντέλων φαίνεται στην ακόλουθη εικόνα:



Εικόνα 5.9: Ιστορική Εξέλιξη των Μοντέλων ΒΔ

## 5.20 Επίπεδο Μοντέλο (Flat Model)

Το επίπεδο μοντέλο ή μοντέλο **πίνακα (table)** ή μοντέλο **επίπεδου αρχείου (flat file)** είναι ο παλαιότερος τρόπος οργάνωσης δεδομένων. Πρόκειται για συλλογή ακολουθιών δεδομένων, διαχωρισμένα με ειδικούς χαρακτήρες (π.χ. κενό, tab, κόμμα, ερωτηματικό) που μπορούν να διερευνηθούν σειριακά για τον εντοπισμό τους. Η σειριακή αναζήτηση δεδομένων είναι αναποτελεσματική, αλλά

μπορεί να είναι χρήσιμη για μικρούς καταλόγους και για απλές δομές δεδομένων. Σε αυτό το μοντέλο δεν υπάρχει η δυνατότητα καθορισμού του τύπου των δεδομένων, καθώς η οργάνωση του κάθε αρχείου είναι ελεύθερη, χωρίς έλεγχο κανόνων από κάποιο σύστημα. Η διαχείριση γίνεται μέσω προγραμμάτων που διαβάζουν και γράφουν στα αρχεία των δεδομένων.

#### **Προβλήματα :**

- ταυτόχρονη εγγραφή από περισσότερες από μία εφαρμογές.
- απώλεια δεδομένων λόγω κατάρρευσης μιας εφαρμογής κατά τη διάρκεια προσπέλασης του αρχείου.

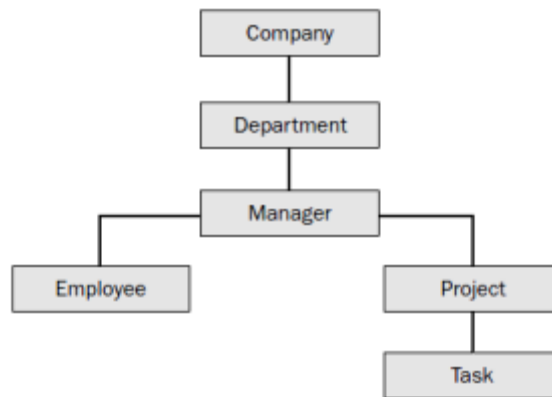
### **5.21 Ιεραρχικό Μοντέλο (Hierarchical Model)**

Στο ιεραρχικό μοντέλο τα δεδομένα οργανώνονται σε δομές που ακολουθούν τη λογική του ανεστραμμένου δένδρου. Τις δομές αυτές μπορούμε να τις θεωρήσουμε πίνακες εγγραφών (record tables), οι οποίοι χαρακτηρίζονται από δείκτες (pointers). Οι σχέσεις μεταξύ των πινάκων είναι του τύπου πρόγονος-απόγονος (parent-child), επειδή δημιουργούνται σχέσεις πολλαπλότητας «ένα προς πολλά». Αυτό καθίσταται δυνατό μέσω των δεικτών που φέρουν οι πίνακες. Ο πρώτος πίνακας-πρόγονος καλείται ρίζα (root), και έχει έναν ή περισσότερους πίνακες – απογόνους. Κάθε πίνακας, μπορεί να έχει έναν ή περισσότερους απογόνους, αλλά **μόνο έναν πρόγονο**. Η αναζήτηση είναι **μονοσήμαντη** μέσω των «κλαδιών», ξεκινώντας από τη ρίζα. Στο ιεραρχικό μοντέλο η ακεραιότητα αναφοράς των δεδομένων είναι εγγενής και δεν χρειάζεται ιδιαίτερος μηχανισμός για να επιτευχθεί.

Ένα τυπικό παράδειγμα οργάνωσης σύμφωνα με την ιεραρχικό μοντέλο είναι το πρότυπο καταλόγου Lightweight Directory Access Protocol – LDAP ή το πρότυπο eXtended Markup Language - XML.

Τα σοβαρότερα μειονεκτήματα είναι :

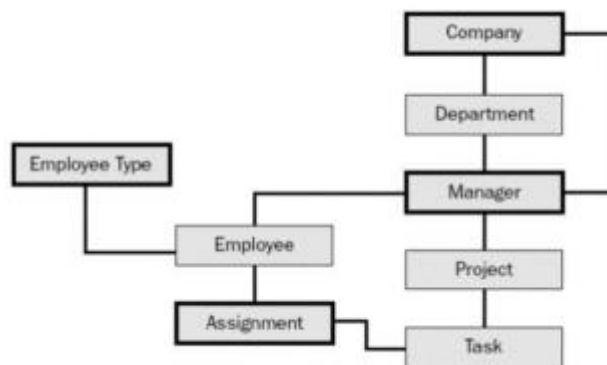
- η ανάγκη εκκίνησης της αναζήτησης από τη ρίζα.
- η μεγάλη δυσκολία αλλαγής της δομής του «δέντρου», η οποία επηρεάζει, όχι μόνο την ίδια τη βάση δεδομένων, αλλά και τις εφαρμογές που τη χειρίζονται.
- λόγω της στενής σχέσης των πινάκων, όταν διαγράφεται ένας πρόγονος, πρέπει να διαγραφούν και οι απόγονοί του.



Εικόνα 5.10: Οργάνωση δεδομένων με το Ιεραρχικό Μοντέλο

### 5.22 Μοντέλο Δικτύου (Network Model)

Το μοντέλο δικτύου οργανώνει τα δεδομένα ακολουθώντας δύο βασικές έννοιες : τις εγγραφές (records) και τα σύνολα (sets). Οι εγγραφές περιέχουν πεδία ενώ τα σύνολα (που δεν σχετίζονται με τη μαθηματική έννοια του συνόλου) ορίζουν σχέσεις πολλαπλότητας «ένα προς πολλά» μεταξύ των εγγραφών. Έτσι, Οι έγγραφες συνδέονται μεταξύ τους δημιουργώντας ένα δίκτυο συνδέσεων, και ουσιαστικά συνδυάζονται μεμονωμένες ιεραρχικές βάσεις δεδομένων σε μια μεγαλύτερη και με αυτόν τον τρόπο το μοντέλο δικτύου αποτελεί επέκταση του ιεραρχικού μοντέλου γιατί επιτρέπει σε κάθε απόγονο να έχει πολλούς προγόνους, δηλ. επιτρέπει τη δημιουργία σχέσεων πολλαπλότητας «πολλά προς πολλά». Η πλοήγηση δεν απαιτεί την έναρξη από κάποια ρίζα, **αλλά μπορεί να αρχίσει από οποιαδήποτε πίνακα.**



Εικόνα 5.11: Οργάνωση δεδομένων με το Μοντέλο Δικτύου

### 5.23 Σχεσιακό Μοντέλο (Relational Model)

Το σχεσιακό μοντέλο βασίζεται στην Κατηγορηματική Λογική (Predicate Logic) και τη Θεωρία Συνόλων (Set Theory). Τα βασικά στοιχεία του μοντέλου είναι οι οντότητες (entities), οι σχέσεις ή συσχετίσεις (relations), τα χαρακτηριστικά (attributes) και τα πεδία τιμών τους (domains). Στο σχεσιακό μοντέλο οι οντότητες και οι συσχετίσεις αποδίδονται με πίνακες, και σε αντίθεση με το ιεραρχικό και το μοντέλο δικτύου, η έννοια της ιεραρχίας και δεν αποτελεί το κυρίαρχο χαρακτηριστικό.

### 5.24 Αντικειμενοστραφές μοντέλο (Object Oriented Model)

Το μοντέλο αυτό προέκυψε από την ανάγκη άμεσης πρόσβασης στο μοντέλο της βάσης δεδομένων από σύγχρονες τεχνικές αντικειμενοστραφούς προγραμματισμού (object oriented programming). Αυτό έχει σαν αποτέλεσμα την άμεση προσπέλαση των δεδομένων από της εφαρμογές και τις καθιστά **αποδοτικότερες σε νέους τύπους δεδομένων** όπως ήχου και εικόνας (σταθερής ή video). Οι αντικειμενοστραφείς ΒΔ προσπελαύνονται άμεσα από γλώσσες όπως η Java και η C++, πράγμα που τις κάνει 10 έως 1000 φορές ταχύτερες από τις αντίστοιχες σχεσιακές.

Αν και σχεδιαστικά οι αντικειμενοστραφείς ΒΔ ξεπερνούν πολλές αδυναμίες των σχεσιακών, πρακτικά παρουσιάζουν το μειονέκτημα ότι η διείσδυσή τους στην αγορά είναι μικρή σε σύγκριση με τις σχεσιακές βάσεις. Αυτό σημαίνει λιγότερους εξειδικευμένους προγραμματιστές και μη διασφάλισης εκτέλεσης των εφαρμογών σε διαφορετικές πλατφόρμες λόγω μη επαρκούς τυποποίησης (standardization).

Τέλος, υφίσταται και ένα υβριδικό μοντέλο Συσχέτισης Αντικειμένων (Object Related Model) που προτάθηκε από την Oracle στο τέλος της δεκαετίας του 1990. Το μοντέλο αυτό ενσωματώνει αντικειμενοστραφή χαρακτηριστικά σε μια σχεσιακή βάση δεδομένων, επιτρέποντας τη δημιουργία νέων τύπων δεδομένων, αν και η βασική αρχιτεκτονική παραμένει σχεσιακή.

### 5.25 Λειτουργίες και Γλώσσες

Ο καθορισμός και η διαχείριση της δομής της βάσης πραγματοποιούνται με τη βοήθεια ειδικών γλωσσών προγραμματισμού. Οι γλώσσες αυτές, χαρακτηρίζονται ως γλώσσες τέταρτης γενεάς (4th Generation Languages, 4GL), διότι σε αντίθεση με τις συμβατικές γλώσσες προγραμματισμού δεν υλοποιούν πολύπλοκες λειτουργίες χαμηλού επιπέδου, αλλά το μόνο που κάνουν είναι να ανακτούν δεδομένα από τη βάση του συστήματος, χωρίς ο χρήστης να γνωρίζει πως λαμβάνει χώρα αυτή η διαδικασία. Εκτός από την ανάκτηση των δεδομένων, οι γλώσσες αυτές προσφέρουν και πιο προχωρημένες λειτουργίες, όπως είναι η δημιουργία νέων δομών δεδομένων μέσα στη βάση, η διαγραφή και η τροποποίηση αυτών, και ο ορισμός συσχετίσεων μεταξύ των στοιχείων της βάσης. Οι λειτουργίες αυτές μπορούν να διαχωριστούν σε δύο κατηγορίες: σ' αυτές που τροποποιούν τη **δομή της βάσης**, και σ' εκείνες που τροποποιούν το **περιεχόμενο της βάσης**, δηλαδή τα δεδομένα που αποθηκεύονται σε αυτή. Αυτό

σημαίνει ότι υφίστανται δύο επίπεδα διαχείρισης, και γι' αυτό χρησιμοποιούνται διαφορετικές γλώσσες προγραμματισμού: μια γλώσσα που είναι υπεύθυνη για τον ορισμό και την τροποποίηση της δομής της βάσης και η οποία ονομάζεται γλώσσα ορισμού δεδομένων (Data Definition Language, DDL), και μια γλώσσα που είναι υπεύθυνη για τη διαχείριση των δεδομένων της βάσης, και η οποία ονομάζεται γλώσσα διαχείρισης δεδομένων (Data Manipulation Language, DML).

## 5.26 Βάση δεδομένων πλατφόρμας Erasmus+ Office ΑΕΙ Πειραιά Τ.Τ.

Τώρα που επεξηγήσαμε αναλυτικά όλες τις τεχνολογίες αναφορικά με τη δομή των βάσεων δεδομένων, θα αναλύσουμε τη δομή της βάσης δεδομένων που χρησιμοποιήθηκε για την παρούσα πτυχιακή εργασία και τη δομή της πλατφόρμας της ιστοσελίδας του γραφείου του Erasmus+ του ΑΕΙ Πειραιά Τ.Τ. Όπως και προηγουμένως για να γίνουμε πιο επεξηγηματικοί, η παρουσίαση θα γίνει με τη βοήθεια εικόνων.

Οι **γλώσσες προγραμματισμού** που χρησιμοποιήθηκαν για τη βάση δεδομένων είναι **PHP 7.0** και **MySQL 5.0**. Το framework το οποίο χρησιμοποιήσαμε για την PHP είναι το **Phalcon Framework** και όλα τα **queries** προέρχονται από την MySQL. Για όλα αυτά γίνεται εκτενής περιγραφή και ανάλυση παρακάτω.

Για τη βάση υπάρχει μόνο ένα Login, και αυτό είναι για να συνδεθούμε ως **root user**, δηλαδή ως user με δικαιώματα διαχειριστή (Administrator Privileges).

Η βάση μας αποτελείται από **3 πίνακες (Tables)**.

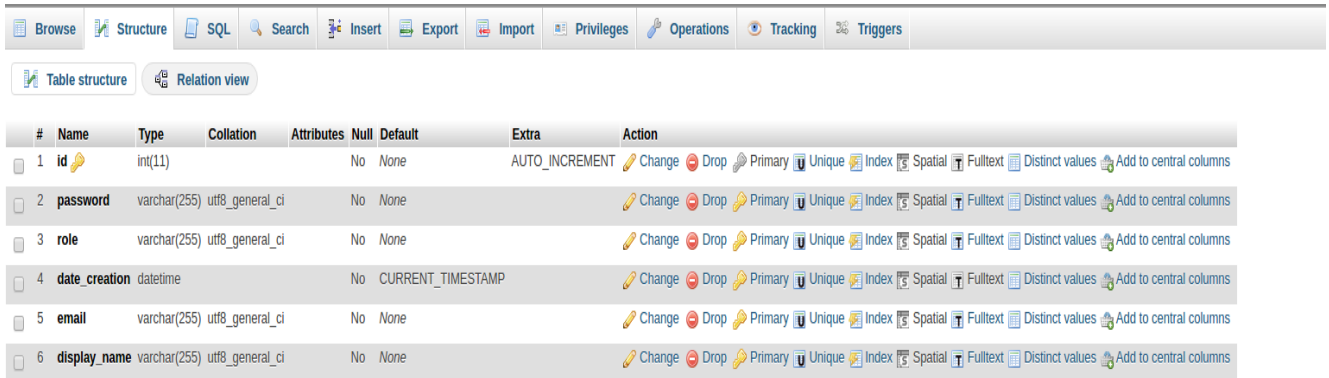
- Ο **πρώτος** πίνακας είναι ο πίνακας **users**. Σε αυτόν τον πίνακα αποθηκεύονται τα **username** και **password** καθώς και τα **logs** και ο **ρόλος** αυτού που έκανε τη σύνδεση, δηλαδή είτε ο χρήστης έχει εισέρθει στην πλατφόρμα ως απλός user (φοιτητής) είτε ως admin (διαχειριστής).
- Ο **δεύτερος** πίνακας είναι ο πίνακας **student / data**. Σε αυτόν τον πίνακα αποθηκεύονται όλα τα στοιχεία των **students** δηλαδή των φοιτητών εκτός από τα credentials δηλαδή τα διαπιστευτήρια (username και password) και συνδέονται με τα κλειδιά των **users (id)**.
- Ο  **τρίτος** πίνακας είναι ο πίνακας **application form**. Σε αυτό τον πίνακα υπάρχουν όλα τα **έγγραφα** τα οποία πρέπει να ανεβάσουν οι φοιτητές, ο οποίος έχει το **id** του κάθε φοιτητή και το **path** (μονοπάτι) από το κάθε έγγραφο.

**Είναι σημαντικό να αναφέρουμε ότι οι πίνακες 2 και 3 συνδέονται με τον 1 μέσω του κλειδιού student\_id.**



Στις εικόνες που ακολουθούν παρακάτω βλέπουμε αναλυτικά πως ακριβώς έχει δομηθεί η βάση μας και οι 3 πίνακες που προαναφέραμε.

1. Αυτός είναι ο πρώτος από τους τρεις πίνακες που περιέχει τα **username**, **password**, **roles** και **logs**.

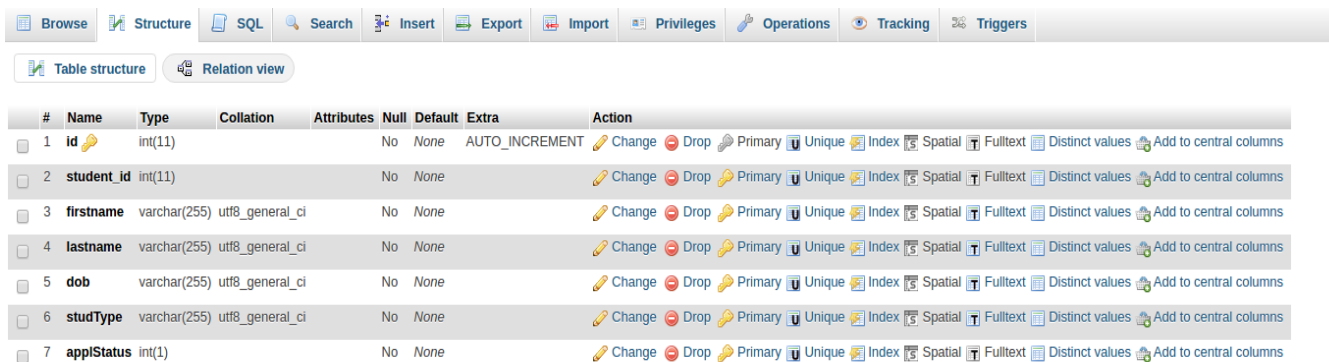


The screenshot shows the MySQL Table structure for the 'Users' table. The table has 6 columns: id, password, role, date\_creation, email, and display\_name. The 'id' column is an integer with an auto-increment attribute. The 'password', 'role', 'email', and 'display\_name' columns are all VARCHAR(255) with utf8\_general\_ci collation. The 'date\_creation' column is a DATETIME with a default value of CURRENT\_TIMESTAMP. Each column has a set of actions available, including Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Distinct values, and Add to central columns.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
2	password	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
3	role	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
4	date_creation	datetime			No	CURRENT_TIMESTAMP		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
5	email	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
6	display_name	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns

Εικόνα 5.12: Πίνακας Users

2. Αυτός είναι ο δεύτερος από τους τρεις πίνακες που περιέχει όλα τα στοιχεία των **students**.

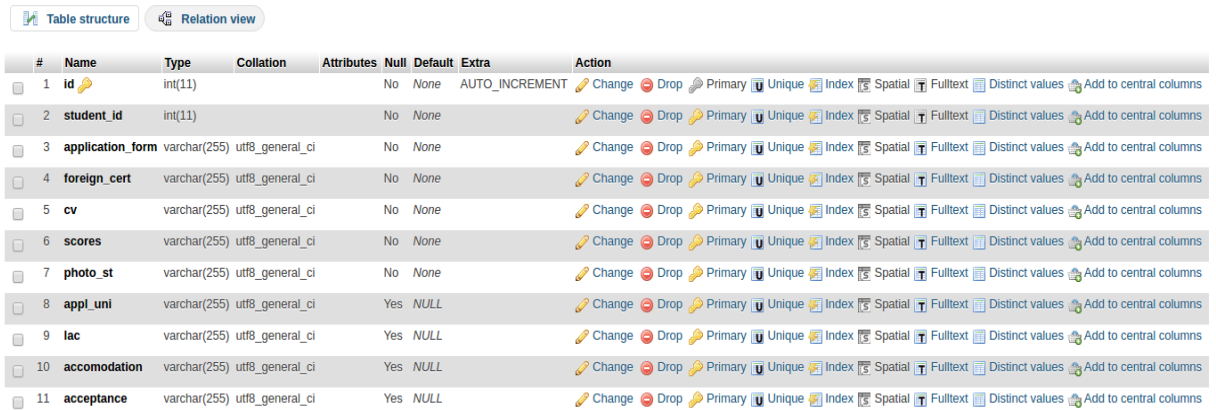


The screenshot shows the MySQL Table structure for the 'student / data' table. The table has 7 columns: id, student\_id, firstname, lastname, dob, studType, and applStatus. The 'id' column is an integer with an auto-increment attribute. The 'student\_id' column is an integer. The 'firstname', 'lastname', 'dob', and 'studType' columns are all VARCHAR(255) with utf8\_general\_ci collation. The 'applStatus' column is an integer. Each column has a set of actions available, including Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Distinct values, and Add to central columns.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
2	student_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
3	firstname	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
4	lastname	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
5	dob	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
6	studType	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
7	applStatus	int(1)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns

Εικόνα 5.13: Πίνακας student / data

3. Αυτός είναι ο τρίτος από τους τρεις πίνακες που περιέχει όλα τα **έγγραφα** και το **id** κάθε φοιτητή και το **path** από το κάθε έγγραφο.



The screenshot shows a database table structure view for a table named 'Application Form'. The table has 11 columns. The first column, 'id', is an integer (int(11)) and is the primary key. The other columns are 'student\_id', 'application\_form', 'foreign\_cert', 'cv', 'scores', 'photo\_st', 'appl\_uni', 'lac', 'accomodation', and 'acceptance', all of which are varchar(255) with utf8\_general\_ci collation. The 'application\_form' column is also the primary key. The 'id' column has an 'AUTO\_INCREMENT' extra attribute. The 'Action' column lists various database actions for each column, such as Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Distinct values, and Add to central columns.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
2	student_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
3	application_form	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
4	foreign_cert	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
5	cv	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
6	scores	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
7	photo_st	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
8	appl_uni	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
9	lac	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
10	accomodation	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns
11	acceptance	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values Add to central columns

Εικόνα 5.14: Πίνακας Application Form

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 6

### ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΕΝΝΟΙΕΣ BACKEND FRONTEND

#### 6.1 Frontend

Όταν συζητάμε για το frontend, μιλάμε για το τμήμα του ιστού που μπορείτε να δείτε και να αλληλεπιδράσετε. Το frontend συνήθως αποτελείται από δύο μέρη: το σχεδιασμό και την ανάπτυξη frontend.

Στο παρελθόν όταν κάποιος συζητούσαν για την ανάπτυξη συνήθως αναφέρονταν στο backend, αλλά τα τελευταία χρόνια έχει υπάρξει μια πραγματική ανάγκη για διαφοροποίηση μεταξύ των σχεδιαστών που εργάστηκαν αυστηρά στο Photoshop και εκείνων που γράφουν κώδικα HTML και CSS. Τα πράγματα μπερδεύονται μάλιστα ακόμη περισσότερο, όταν οι σχεδιαστές διέσχισαν τις γραμμές του σχεδιασμού και άρχισαν να εργάζονται με την Javascript και την jQuery.

Έτσι τώρα, όταν συζητάμε για τον όρο “σχεδιασμός ιστοσελίδων”, μιλάμε για εκείνους που εργάζονται με το Photoshop και το Fireworks, καθώς και εκείνους που χρησιμοποιούν κώδικα HTML, CSS, JavaScript, jQuery κτλ.

Ό, τι βλέπετε, στο διαδίκτυο είναι ένας συνδυασμός HTML, CSS, JavaScript, τα οποία ελέγχονται από το πρόγραμμα περιήγησης (browser) του υπολογιστή σας. Αυτά περιλαμβάνουν πράγματα όπως γραμματοσειρές, drop-down μενού, κουμπιά, transitions, sliders, φόρμες επικοινωνίας, κλπ.

Για να γίνουν όλα αυτά πραγματικότητα και να αποθηκεύουν οι πληροφορίες που έχετε βάλει στα στοιχεία frontend, χρειαζόμαστε την τεχνολογία που θα τα κάνει εφικτά ... το backend ...

Οι χρήστες Frontend έχουν παραπάνω δικαιώματα σε σχέση με τους επισκέπτες, τα οποία μπορεί να περιλαμβάνουν τη δυνατότητα δημιουργίας και έκδοσης περιεχομένου στην ιστοσελίδα. Συνήθως αναφερόμαστε σε αυτούς τους χρήστες ως παρόχους περιεχομένου, καθώς πρωταρχικός τους στόχος είναι να παρέχουν περιεχόμενο στην ιστοσελίδα, και όχι να διαχειρίζονται την ιστοθέση ή να τροποποιούν τη δομή της. Αυτοί οι πάροχοι περιεχομένου μπορούν να παρέχουν περιεχόμενο μέσω της διεπιφάνειας web (web interface), χρησιμοποιώντας ένα ενσωματωμένο εργαλείο επεξεργασίας (editor) WYSIWG (What You See Is What You Get), χωρίς να χρειάζεται να έχουν γνώσεις κωδικοποίησης σε HTML.

Στην κατηγορία των χρηστών Front-end υπάρχουν τέσσερα διακριτά επίπεδα, τα οποία καθορίζει ο διαχειριστής για κάθε χρήστη. Τα επίπεδα αυτά είναι: Εγγεγραμμένος (Registered), Συγγραφέας (Author), Συντάκτης (Editor), και εκδότης (Publisher).

- **Εγγεγραμμένος (Registered):** Δεν έχει τη δυνατότητα να δημιουργεί, να επεξεργάζεται, ή να εκδίδει περιεχόμενα στην ιστοσελίδα. Μπορεί να υποβάλλει νέα Web Links για έκδοση και μπορεί να έχει πρόσβαση σε περιορισμένο περιεχόμενο το οποίο δεν είναι διαθέσιμο σε επισκέπτες.
- **Συγγραφέας (Author):** Μπορεί να δημιουργεί δικό του περιεχόμενο, να καθορίζει ως ένα βαθμό τον τρόπο εμφάνισης του περιεχομένου, και να ορίζει την ημερομηνία έκδοσης του περιεχομένου.
- **Συντάκτης (Editor):** Έχει τα ίδια δικαιώματα με ένα Συγγραφέα και μπορεί επίσης να επεξεργάζεται πέρα από το δικό του περιεχόμενο και το περιεχόμενο οποιουδήποτε άλλου Συγγραφέα.
- **Εκδότης (Publisher):** Μπορεί να επιτελέσει όλα τα καθήκοντα ενός Συγγραφέα και ενός Εκδότη, έχοντας παράλληλα τη δυνατότητα να εκδίδει περιεχόμενο.

## 6.2 Backend

Το backend αποτελείται συνήθως από τρία μέρη: ένα διακομιστή (server), μια εφαρμογή (application), και την βάση δεδομένων (database). Αν πχ κλείσετε μια πτήση ή αγοράσετε εισιτήρια συναυλιών, ανοίγετε συνήθως μια ιστοσελίδα και αλληλεπιδράτε με το frontend.

Μόλις καταχωρήσετε τις πληροφορίες, η εφαρμογή τις αποθηκεύει σε μια βάση δεδομένων που έχει δημιουργηθεί σε ένα διακομιστή.

Για λόγους ευκολίας, απλά σκεφτείτε την βάση δεδομένων ως ένα τεράστιο υπολογιστικό φύλλο του Excel στον υπολογιστή σας, απλά στην συγκεκριμένη περίπτωση ο server που αποθηκεύονται τα δεδομένα μπορεί να είναι στην Αμερική, την Γερμανία κτλ.

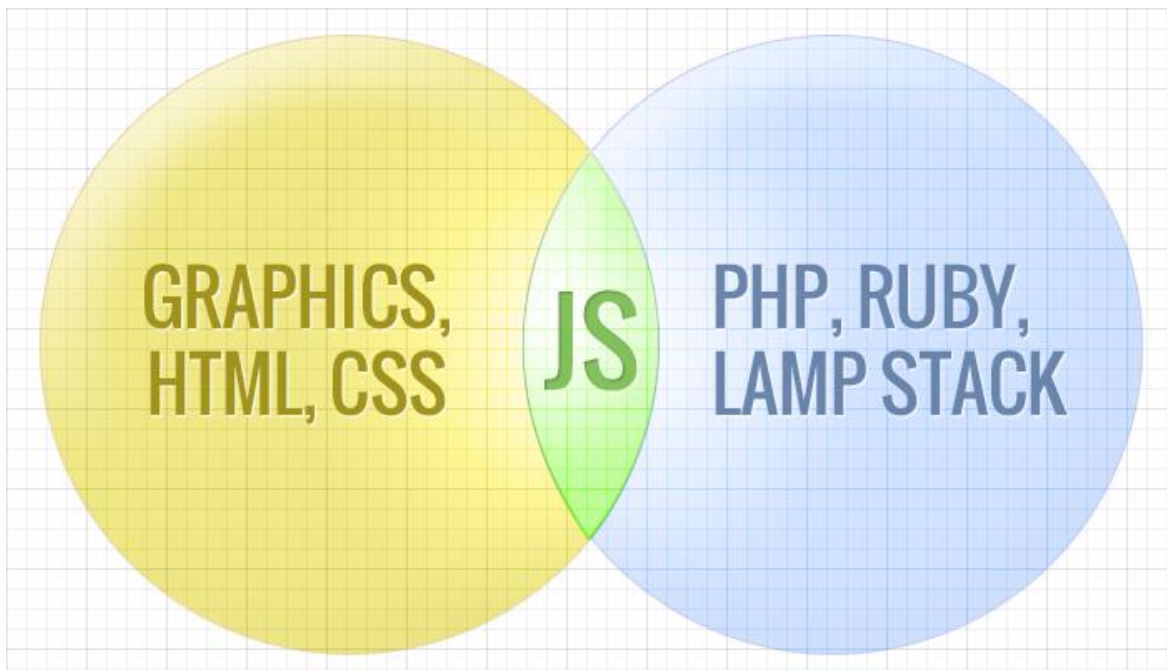
Όλες αυτές οι πληροφορίες παραμένουν στο διακομιστή, έτσι ώστε όταν συνδεθείτε ξανά στην εφαρμογή για πχ την εκτύπωση των εισιτηρίων σας, όλες οι πληροφορίες να είναι ακόμα στο λογαριασμό σας.

Το πρόσωπο που χτίζει όλα αυτά, έτσι ώστε να εργαστούν από κοινού αυτές οι τεχνολογίες, είναι ο backend developer του έργου. Οι Backend τεχνολογίες συνήθως αποτελούνται από γλώσσες όπως η PHP, η Ruby, η Python, κλπ.

Για να γίνουν ακόμα ευκολότερες στη χρήση, ενισχύονται από frameworks όπως, το Ruby on Rails, το Cake PHP, το Code Igniter κτλ κάνοντας με αυτό τον τρόπο την ανάπτυξη πιο γρήγορη και πιο εύκολη.

Οι χρήστες Back-end αναφέρονται συνήθως ως οι διαχειριστές της ιστοσελίδας, αλλά έχουν ταυτόχρονα και δικαίωμα στη διεπιφάνεια Front-end της ιστοσελίδας. Όπως και με τους χρήστες Front-end, έτσι και οι χρήστες Back-end έχουν διαφορετικά προνόμια.

- **Manager:** Έχει πρόσβαση σε όλα τα εργαλεία διαχείρισης περιεχομένου που βρίσκονται στον πίνακα διαχείρισης (administration panel) της ιστοσελίδας, αλλά δεν έχει τη δυνατότητα να αλλάξει πρότυπα (templates), να τροποποιήσει τη δομή των σελίδων, ή να προσθέσει και διαγράψει επεκτάσεις. Δεν έχει επίσης τη δυνατότητα να προσθέσει χρήστες ή να τροποποιήσει υπάρχοντα προφίλ χρηστών.
- **Διαχειριστής (Administrator):** Μπορεί να προσθέτει και να διαγράφει επεκτάσεις στην ιστοθέση, να αλλάζει πρότυπα ή να τροποποιεί τη δομή των σελίδων, ενώ μπορεί επίσης να τροποποιεί υπάρχοντα προφίλ χρηστών που βρίσκονται στην ίδια βαθμίδα με αυτόν ή σε κατώτερη.
- **Υπερδιαχειριστής (Super Administrator):** Έχει απεριόριστη πρόσβαση να επιτελεί όλες τις διαχειριστικές λειτουργίες μέσα στην ιστοσελίδα. Μόνο οι υπερδιαχειριστές έχουν τη δυνατότητα να δημιουργούν νέους χρήστες με δικαιώματα υπερδιαχειριστή ή να δίνουν δικαιώματα υπερδιαχειριστή σε υπάρχοντες χρήστες.



Εικόνα 6.1: Σχεδιάγραμμα αλληλεπίδρασης JSON

## **Συμπέρασμα**

Ελπίζουμε τα παραπάνω να βοήθησαν αρκετούς από εσάς να καταλάβετε για τι πράγμα μιλάμε όταν αναφερόμαστε στο frontend και το backend, καθώς και να γνωρίζετε πότε μιλάμε για σχεδιασμό και πότε για ανάπτυξη.

Οι γραμμές μεταξύ του σχεδιασμού και της ανάπτυξης γίνονται όλο και πιο θολές σε καθημερινή βάση, αλλά ουσιαστικά είναι τελείως διαφορετικό κομμάτι.

Για να καταφέρουμε να συνεργαστούμε σε μεγάλα projects είναι πολύ σημαντικό να είμαστε σαφείς σχετικά με το μέρος του “προϊόντος” για το οποίο μιλάμε.

Ποιες είναι οι δικές σας σκέψεις σχετικά με το σχεδιασμό και την ανάπτυξη, για το frontend σε σχέση με το backend;

## ΚΕΦΑΛΑΙΟ 7

### JSON

#### 7.1 Τι είναι το JSON

Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να το διαβάσουν και γράψουν. Είναι εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate). Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999. Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

Το JSON είναι χτισμένο σε δύο δομές:

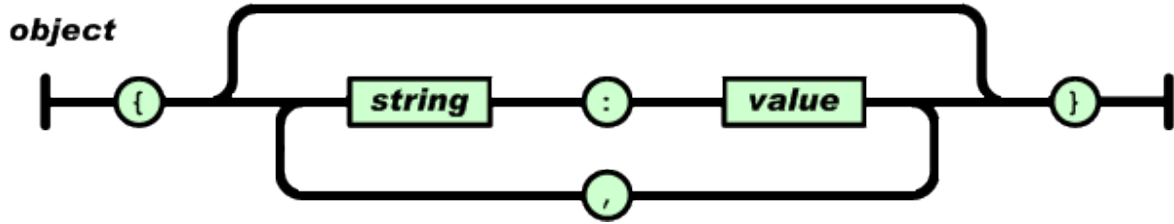
- Μια συλλογή από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, ή associative πίνακα.
- Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία.

Αυτά είναι τα universal data structures. Ουσιαστικά όλες οι μοντέρνες γλώσσες προγραμματισμού τα υποστηρίζουν με τον έναν ή τον άλλον τρόπο. Λογικό είναι πως ένα πρότυπο δεδομένων το οποίο είναι εύκολα μεταβαλλόμενο με γλώσσες προγραμματισμού οι οποίες επίσης είναι βασισμένες σε αυτές τις δομές.

Στο JSON, παίρνουν αυτές τις μορφές:

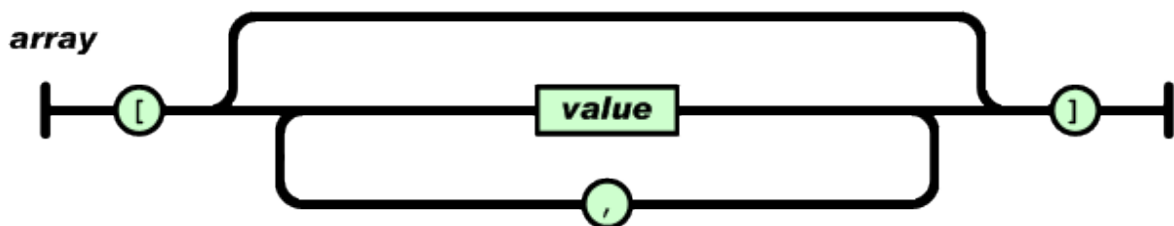
1. Ένα αντικείμενο (object) είναι ένα άτακτο σύνολο από ζευγάρια ονομάτων/τιμών. Ένα αντικείμενο (object) ξεκινάει με { (αριστερό άγκιστρο) και τελειώνει με } (δεξιό άγκιστρο). Κάθε όνομα ακολουθείται από : (άνω-κάτω τελεία) και τα ζευγάρια ονόματος/τιμής χωρίζονται από , (κόμμα).





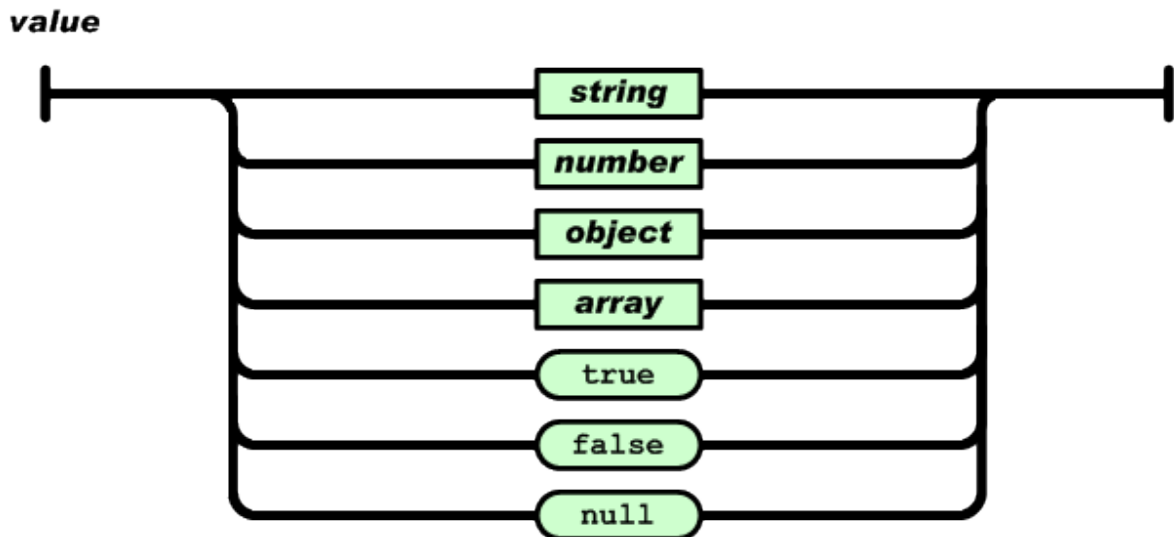
Εικόνα 7.1: Αντικείμενο Object

2. Ένας πίνακας (array) είναι μια συλλογή από τιμές σε σειρά. Ένας πίνακας (array) ξεκινάει με [ (αριστερή αγγύλη) και τελειώνει με ] (δεξιά αγγύλη). Οι τιμές χωρίζονται με , (κόμμα).



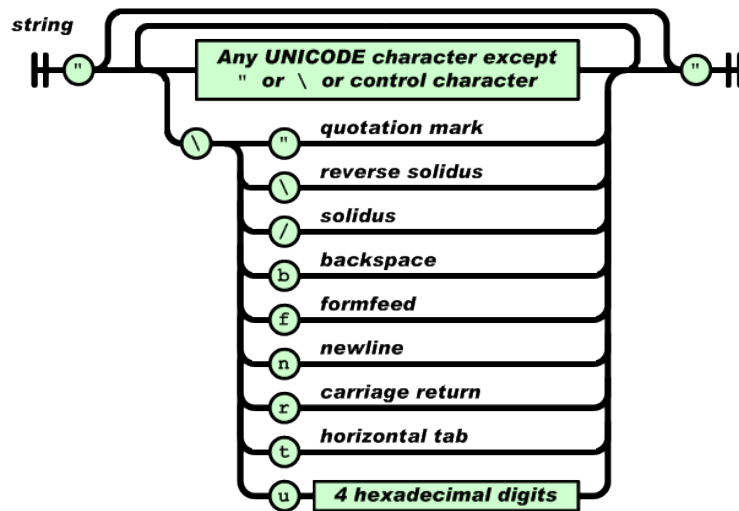
Εικόνα 7.2: Πίνακας Array

3. Μία τιμή μπορεί να είναι string μέσα σε διπλά quotes, ή αριθμός (number), ή **true** ή **false** ή **null**, ή αντικείμενο (object) ή πίνακας (array). Αυτές οι τιμές μπορεί να είναι και ανακατεμένες.



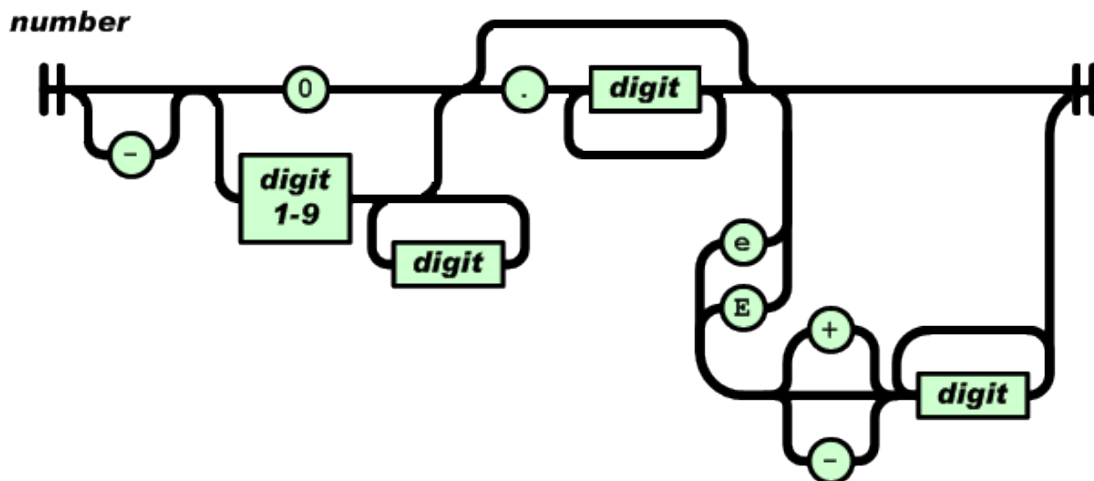
Εικόνα 7.3: Είδη τιμών

4. Ένα string είναι μια συλλογή από μηδέν ή περισσότερους Unicode χαρακτήρες, μέσα σε διπλά quotes, χρησιμοποιώντας αντίστροφους κάθετους \ (backslash) για escapes. Ένας χαρακτήρας αντιπροσωπεύεται ως ένας μονός χαρακτήρας string. Ένα string μοιάζει πολύ σαν ένα C ή Java string.



Εικόνα 7.4: Είδη χαρακτήρων

5. Ένας αριθμός (number) μοιάζει πάρα πολύ με ένα C ή Java αριθμό (number), με την διαφορά πως τα οκταδικά και δεκαεξαδικά συστήματα δεν χρησιμοποιούνται.



Εικόνα 7.5: Είδη αριθμών

Τα κενά (whitespace) μπορούν να εισαχθούν ανάμεσα σε οποιοδήποτε ζευγάρι tokens. Με εξαίρεση μερικών λεπτομερειών κωδικοποίησης (encoding), αυτό περιγράφει γενικότερα την γλώσσα (προγραμματισμού).

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 8

### WEB SERVICES

#### 8.1 Ορισμός των web services

Τα web services εμφανίστηκαν το 2001 όταν άρχισε να γίνεται λόγος για αυτο-περιγραφόμενες και αυτο-προσδιοριζόμενες εφαρμογές οι οποίες μπορούν να ανακαλυφθούν και να προσπελαστούν μέσω διαδικτύου από άλλες εφαρμογές. Οι υπηρεσίες WEB κατάφεραν όχι μόνο να παραμείνουν στο προσκήνιο των εξελίξεων στον χώρο της τεχνολογίας όλη τη δεκαετία, αλλά να αποτελέσουν ένα από τα πιο σημαντικά κεφάλαια της υψηλής τεχνολογίας του Διαδικτύου.

Web service είναι οποιοδήποτε τμήμα λογισμικού ή συστατικό εφαρμογής που είναι διαθέσιμο μέσω διαδικτύου και το οποίο μπορεί να ολοκληρώνει εργασίες και να λύνει προβλήματα. Ωστόσο αν θέλουμε να προσδιορίσουμε πιο σωστά την έννοια των web services, θα μπορούσαμε να χρησιμοποιήσουμε τον ορισμό που έρχεται από την IBM:

- Τα web services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα web service αποτελεί μια διεπαφή λογισμικού ( software interface ) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μια λειτουργία προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από web services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή web services. Με απλά λόγια μια Web service μπορούμε να την αντιληφθούμε σαν ένα λογισμικό που γνωρίζει με ποιόν τρόπο πρέπει να 'μιλήσει' σε διαφορετικού τύπου λογισμικά μέσω δικτύου. Πιο συγκεκριμένα μια υπηρεσία Web πραγματοποιεί τις ακόλουθες λειτουργίες.
- Παρέχει την δυνατότητα σε άλλες εφαρμογές να κατανοήσουν την λειτουργικότητά της, εκθέτοντας και περιγράφοντας την παρουσία της σε αυτές με τη χρήση ενός πρότυπου δικτυακού πρωτοκόλλου.
- Μπορεί να γίνει εύκολα αντιληπτή από άλλες εφαρμογές μέσω ενός καταλόγου στον οποίο βρίσκετε καταχωρημένα.
- Μπορεί να προβληθεί ή να χρησιμοποιηθεί από την εφαρμογή που την χρειάζεται μέσω απλών πρωτοκόλλων.

Στα κεφάλαια που θα ακολουθήσουν θα γίνει μια εκτενής αναφορά στις παραπάνω λειτουργίες, ενώ ταυτόχρονα θα δοθεί ιδιαίτερη έμφαση στα πρωτόκολλα που συναντάμε και χωρίς την παρουσία των οποίων δεν θα μπορούσαμε σήμερα να μιλάμε για την ύπαρξη των web services. Η δυνατότητα που δίνεται σε οποιαδήποτε σύγχρονη επιχείρηση να παρέχει on line συναλλαγές με τους προμηθευτές της αλλά και με τους πελάτες της μέσω της συγκεκριμένης υπηρεσίας, είναι μόλις μια από τις πολλές εφαρμογές της υπηρεσίας Web. Δεν είναι τυχαία άλλωστε η ραγδαία εξάπλωση της παρά τις αρνητικές πιθανότητες

των αρχών της δεκαετίας του 2000, όπου πολλές άλλες προσανατολισμένες εφαρμογές της τεχνολογίας προς το Διαδίκτυο, εξαφανίστηκαν σχεδόν όσο γρήγορα εμφανίστηκαν.

## 8.2 Η δύναμη των web services

Οι web services χαρακτηρίζονται από τη ανεξάρτητη και ουδέτερη φύση τους. Σε αντίθεση με τις παλαιότερες εφαρμογές λογισμικού που βασίζονταν σε κλειστά πρότυπα, οι υπηρεσίες Web βασίζονται σε ευρέως αποδεκτά πρότυπα του κλάδου. Με την χρήση μιας γλώσσας ανοιχτού προτύπου, την XML, σαν το μόνο πρότυπο στα web services, το πρόβλημα της διαλειτουργικότητας από το οποίο έπασχαν όλες οι προηγούμενες τεχνολογίες έπαψε να υφίσταται. Η XML ερμηνεύεται από οποιαδήποτε γλώσσα προγραμματισμού, καθιστώντας έτσι τις υπηρεσίες Web ουδέτερες και τις υπηρεσίες που τις υιοθετούν να μην χρειάζεται να ανησυχούν για θέματα συμβατότητας μεταξύ λειτουργικών συστημάτων.

Ταυτόχρονα, η ευκολία επικοινωνίας μεταξύ συμβατών και ασυμβατών συστημάτων μεταξύ τους εξαιτίας της XML, μας οδήγησε σε εφαρμογές με χαλαρή συνδεσιμότητα, ευελιξία και προσαρμοστικότητα. Σε τεχνικό επίπεδο τα web services χειρίζονται τα δεδομένα με πολύ πιο εύκολο τρόπο από παλαιότερες τεχνολογίες καθώς λειτουργούν σε ένα πιο αφηρημένο επίπεδο. Επιπλέον οι δυναμικές συνεργασίες μεταξύ συστημάτων και εταιρών είναι πλέον γεγονός. Οι περισσότερες τεχνολογίες χρησιμοποιούσαν μη πρότυπες πόρτες, με αποτέλεσμα η συνεργασία μεταξύ εταιρειών να απαιτεί άνοιγμα οπών στα τείχη προστασίας, κάτι το οποίο ήταν μη αποδεκτό από τους ανθρώπους της πληροφορικής. Η δυνατότητα ωστόσο των web services να χρησιμοποιούν πέρα άλλων και το πρωτόκολλο μεταφοράς HTTP, μετέτρεψε αυτό το πρόβλημα σε παρελθόν. Τα περισσότερα τείχη προστασίας επιτρέπουν την πρόσβαση του HTTP μέσω συγκεκριμένης θύρας και έτσι οδηγηθήκαμε σε ευκολότερες συνεργασίες μεταξύ εταιρειών.

Η απλότητα υποδομής είναι ένας ακόμα λόγος που καθιστά τις web services τόσο σημαντικές και ευρέως διαδεδομένες. Τα web services λειτουργούν με πολύ πιο απλά πρωτόκολλα επικοινωνίας (SOAP), τα πρωτόκολλα HTTP-TCP/IP, καθώς όπως προαναφέρθηκε χρησιμοποιούν την γλώσσα ανοιχτού προτύπου XML. Αυτά τα πρότυπα με την απλότητα τους οδηγούν σε ταχύτερη ανάπτυξη εφαρμογών με πολύ πιο χαμηλά κόστη. Η πλειονότητα των εταιρειών διέθεταν στην ουσία ήδη την δικτυακή υποδομή αλλά και την απαραίτητη γνώση, ώστε να μπορέσουν να υποστηρίξουν άμεσα την καινούργια αυτή μετάβαση στον 'κόσμο' των web services. Η προτυποποίηση, η ανοιχτή αρχιτεκτονική, η ευκολία επικοινωνίας συμβατών και ασυμβατών συστημάτων μεταξύ τους καθώς και η μεγάλη εξοικονόμηση κόστους λόγω των παραπάνω, καθιστούν τις web services ικανές να βοηθήσουν στην βελτίωση της αποτελεσματικότητας επιτρέποντας στις επιχειρήσεις και όχι μόνο να ρυθμίζουν τις διαδικασίες μεταξύ τους και των πελατών τους.

### 8.3 Αρχιτεκτονική των web services ( SOA )

Κάθε υπηρεσία για να μπορεί να λειτουργήσει αποδοτικά, αλλά ταυτόχρονα και να εξελιχθεί πρέπει να βασίζεται σε κάποια συγκεκριμένη δομή. Να ακολουθεί ένα συγκεκριμένο μοντέλο πάνω στο οποίο θα αναπτύσσονται όλες οι λειτουργίες και τα πρωτόκολλα της αντίστοιχης υπηρεσίας. Το μοντέλο που χρησιμοποιείται στις μέρες μας για τις Υπηρεσίες Διαδικτύου είναι το λεγόμενο SOA-Service Oriented Architecture. Αντιπροσωπεύει μια αρχιτεκτονική προσανατολισμένη σε υπηρεσίες. Σε αυτό το μοντέλο συναντάμε ξεχωριστές δομικές μονάδες, οι οποίες μπορούν να διανεμηθούν πάνω από ένα δίκτυο, να συνδυαστούν και να ξαναχρησιμοποιηθούν έτσι ώστε να δημιουργήσουν επιχειρησιακές εφαρμογές. Στη συνέχεια γίνεται μια ανάλυση στις δομικές μονάδες της συγκεκριμένης αρχιτεκτονικής καθώς και στις λειτουργίες που εκτελεί κάθε δομική μονάδα είτε ανεξάρτητα είτε σε συνδυασμό με κάποια άλλη.

#### • Service Provider ( Πάροχος Υπηρεσίας ):

Αποτελεί την οντότητα, η οποία συντάσσει την αντίστοιχη υπηρεσία στο Διαδίκτυο. Είναι υπεύθυνος για την επιλογή των εφαρμογών που θα παρέχονται στους χρήστες της συγκεκριμένης web service, το επίπεδο ασφάλειας και προσβασιμότητας καθώς και τον τρόπο κοστολόγησης. Η πιο σημαντική αρμοδιότητα του Παρόχου Υπηρεσίας ωστόσο είναι να δημιουργήσει τις κατάλληλες προϋποθέσεις ώστε η συγκεκριμένη υπηρεσία να είναι ορατή από όλους τους χρήστες του Διαδικτύου. Για την συγκεκριμένη αρμοδιότητα απαιτείται η συνεργασία του με μια άλλη δομική μονάδα, την Service Registry ( Υπηρεσία Καταλόγου ). Ο Πάροχος υπηρεσίας περιγράφει τη υπηρεσία του ενώ παρέχει και κάποιες πληροφορίες πρόσβασης σε αυτή, στον κατάλογο υπηρεσίας. Ταυτόχρονα κατηγοριοποιεί την υπηρεσία μέσα στον κατάλογο και ορίζει τι είδους συμφωνίες απαιτούνται για να μπορεί κανείς να προσπελάσει την συγκεκριμένη λειτουργία.

#### • Service Registry ( Υπηρεσία Καταλόγου ):

Η υπηρεσία καταλόγου αποτελεί μια σύγχρονη <Βιβλιοθήκη>. Αρμοδιότητα της είναι η καταχώρηση μαζί με τις αντίστοιχες περιγραφές όλων των υπάρχοντων υπηρεσιών Διαδικτύου. Ο υπεύθυνος του κάθε καταλόγου αποφασίζει για το φάσμα χρηστών στους οποίους θα είναι ορατός ο κατάλογος, τη ποσότητα των καταχωρήσεων που θα υποστηρίζει καθώς και το επίπεδο ασφαλείας που θα παρέχει. Υπάρχουν δημόσιοι αλλά και ιδιωτικοί καταλόγοι, μερικοί καλύπτουν ένα ευρύ φάσμα υπηρεσιών ενώ άλλοι ειδικεύονται σε καταχωρήσεις συγκεκριμένων κλάδων. Η συγκεκριμένη μονάδα αποτελεί στην ουσία τον συνδετικό κρίκο ανάμεσα στον Πάροχο και στον Χρήστη Υπηρεσίας. Εδώ δημοσιοποιείται η Περιγραφή της εκάστοτε Υπηρεσίας από τον Πάροχο, η οποία ταυτόχρονα αποτελεί το αποτέλεσμα της αναζήτησης του χρήστη στον κατάλογο.

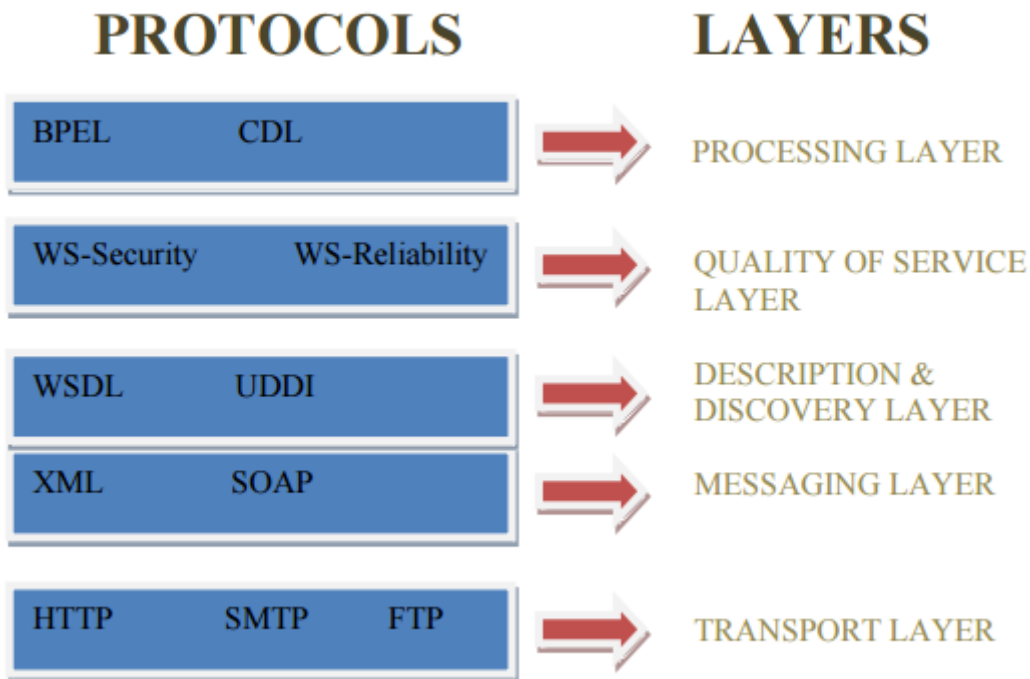
• **Service Requestor ( Αιτούμενος Υπηρεσίας ):**

Αποτελεί τον αιτούμενο της κάθε υπηρεσίας. Αρχικά αναζητά καταχωρήσεις μέσα από έναν αριθμό καταλόγων αντλώντας την περιγραφή της κάθε υπηρεσίας και καταλήγει στον εντοπισμό της ζητούμενης καταχώρησης καθώς και της διεύθυνσης της. Εδώ λήγει και ο ρόλος της ενδιάμεσης δομικής μονάδας. Ο χρήστης συνδέεται με τον πάροχο και καλεί το web service που εκείνος δημοσιεύει.

Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι οι αρμοδιότητες των παραπάνω δομικών μονάδων δεν είναι καθορισμένες. Ο χρήστης ανά πάσα στιγμή μπορεί να γίνει πάροχος και αντίστροφα. Σε αυτό το σημείο λοιπόν μπορούμε να πούμε ότι έχουμε καλύψει τις βασικές λειτουργίες που εκτελεί κάθε δομική μονάδα ξεχωριστά και σε συνεργασία. Ωστόσο μιλάμε για υπολογιστικές εφαρμογές. Για να πραγματοποιηθούν όλες οι παραπάνω λειτουργίες χρησιμοποιούνται συγκεκριμένα πρότυπα, τα οποία είναι ανεξάρτητα από λειτουργικές πλατφόρμες και γλώσσες προγραμματισμού. Στο κεφάλαιο που ακολουθεί γίνεται μια αναφορά στα συγκεκριμένα πρότυπα.

#### **8.4 Τεχνολογίες των web services**

Η αρχιτεκτονική που αναλύθηκε πιο πάνω ακολουθεί με απλά λόγια το μοντέλο δημοσίευση, εύρεση και σύνδεση. Ο πάροχος της υπηρεσίας δημοσιεύει την υπηρεσία στον κατάλογο υπηρεσιών. Ο χρήστης ψάχνει μέσα από τον κατάλογο την υπηρεσία η οποία καλύπτει τις ανάγκες του και στην συνέχεια συνδέεται με αυτήν για να την εκτελέσει. Είναι προφανές λοιπόν ότι οι δομικές μονάδες της αρχιτεκτονικής SOA πρέπει να είναι σε θέση να βρίσκουν η μια την άλλη, να ανακαλύπτουν πληροφορίες που τις επιτρέπουν να αλληλοσυνδεθούν, να καθορίζουν ποιες είναι οι αναμενόμενες μορφές αλληλοσύνδεσης και να διαπραγματεύονται ποιότητες υπηρεσίας όπως η ασφάλεια και η αξιόπιστη επικοινωνία. Για να πραγματοποιηθούν όλες οι παραπάνω λειτουργίες οι Web Services δεν χρησιμοποιούν συγκεκριμένη τεχνολογία, αλλά ένα σύνολο από από καθιερωμένα πρωτόκολλα επικοινωνίας πολλών επιπέδων, που περιλαμβάνουν το HTTP, την XML, το SOAP, το UDDI και το WSDL. Σχηματικά μπορούμε να το φανταστούμε σαν μια κατανομή επιπέδων, όπου η πληροφορία περνάει από το ένα στρώμα στο άλλο, αποκρύπτοντας τις πολλές λεπτομέρειες που δε χρειάζονται στα άλλα στρώματα.



Εικόνα 8.1: Στοιβά Πρωτοκόλλων των Web Services

Πιο αναλυτικά έχουμε τα εξής επίπεδα:

- **TRANSPORT LAYER ( ΕΠΙΠΕΔΟ ΜΕΤΑΦΟΡΑΣ )**

Σε αυτό το επίπεδο συναντάμε την υπηρεσία μεταφοράς, υπεύθυνη για την μεταφορά των διάφορων δεδομένων από τον πάροχο στον χρήστη και αντίστροφα. Εδώ περιλαμβάνονται κάποια από τα πιο διαδεδομένα πρωτόκολλα μεταφοράς όπως το HTTP, το SMTP καθώς και το FTP. Τα συγκεκριμένα πρωτόκολλα τα συναντάμε όχι μόνο στις Υπηρεσίες Web αλλά και σε πολλές άλλες υπηρεσίες που διατίθενται μέσω Internet καθώς αποτελούν κάποια από τα πιο εύχρηστα πρωτόκολλα μεταφοράς χρησιμοποιώντας πόρτες μη περιορισμένες από firewalls.

- **MESSAGING LAYER ( ΕΠΙΠΕΔΟ ΜΗΝΥΜΑΤΩΝ )**

Για να είναι δυνατή η μεταφορά δεδομένων σε ένα συγκεκριμένο υπολογιστικό περιβάλλον πρέπει να χρησιμοποιούνται συγκεκριμένα πρωτόκολλα καθορισμού του τρόπου μορφοποίησης και αναπαράστασης των δεδομένων ώστε στη συνέχεια να μπορούν να μεταβούν στο επίπεδο μεταφοράς. Στο συγκεκριμένο επίπεδο συναντάμε την υπηρεσία μηνυμάτων, η οποία βασισμένη σε συγκεκριμένα πρωτόκολλα, παρέχει το υπόβαθρο στις Υπηρεσίες Web για μεταφορά πληροφορίας. Πιο συγκεκριμένα συναντάμε την γλώσσα σήμανσης XML, καθώς και το SOAP – Πρωτόκολλο Απλής Προσπέλασης Αντικειμένου. Η XML αποτελεί μια περιγραφική γλώσσα με καλά καθορισμένη σύνταξη και σημασιολογία παρέχοντας μια πρότυπη και κοινή δομή για την διανομή των μηνυμάτων.



Το SOAP από την πλευρά του αποτελεί το πλαίσιο της επικοινωνίας μεταξύ μιας εφαρμογής-προμηθευτή web service και μιας εφαρμογής πελάτη σύμφωνα με τη γλώσσα XML.

- **DESCRIPTION & DISCOVERY LAYER (ΕΠΙΠΕΔΟ ΠΕΡΙΓΡΑΦΗΣ & ΑΝΑΚΑΛΥΨΗΣ)**

Για να έχουν ισχύ οι Υπηρεσίες Web θα πρέπει ο κάθε χρήστης να είναι σε θέση να αναζητάει από ένα κατάλογο όλες τις υπάρχουσες web services και με κάποια κριτήρια να συνδέεται με τον πάροχο της Υπηρεσίας που τον ενδιαφέρει. Από το στάδιο αυτό και έπειτα αναλαμβάνουν τα προαναφερθέντα επίπεδα. Τα πρωτόκολλα που συναντάμε σε αυτό το επίπεδο αφορούν την πραγματοποίηση του Service Registry). Δυο βασικά πρωτόκολλα συναντάμε εδώ, τη περιγραφική γλώσσα WSDL και το UDDI. Η WSDL ( Web Services Description Language ) χρησιμοποιώντας ένα σύνολο ετικετών ( tags ) περιγράφει ένα web service, τις υπηρεσίες που παρέχονται, την διεύθυνση του κλπ. Το UDDI από την πλευρά του αποτελεί τον <Χρυσό Οδηγό> των web services. Όλες οι εφαρμογές που παρέχουν web services παρατίθενται σε έναν κατάλογο από παρόχους υπηρεσιών χρησιμοποιώντας το UDDI.

- **QUALITY OF SERVICE LAYER ( ΕΠΙΠΕΔΟ ΠΟΙΟΤΗΤΑΣ )**

Βασικά στοιχεία που πρέπει να εξασφαλίζονται σε κάθε Υπηρεσία Web πρέπει να είναι η διαθεσιμότητα, η αξιοπιστία και η ασφάλεια. Τα πρότυπα που συναντάμε σε αυτό το επίπεδο είναι το WS-Security και το WS-Reliability υπεύθυνα για την εξασφάλιση του απαραίτητου QoS σε μια Web Service.

- **PROCESSES LAYER ( ΕΠΙΠΕΔΟ ΔΙΕΡΓΑΣΙΩΝ )**

Πρόκειται για το επίπεδο που είναι υπεύθυνο για το συνδυασμό ήδη υλοποιημένων υπηρεσιών ώστε να προκύψουν νέες σύνθετες υπηρεσίες. Επιτρέπει στους χρήστες να περιγράφουν αλλά και να υλοποιούν δραστηριότητες μέσω web services ορίζοντας τον τρόπο με τον οποίο θα συνεργαστούν για να πραγματοποιήσουν διάφορες λειτουργίες. Στο συγκεκριμένο επίπεδο συναντάμε τα εξής δυο πρωτόκολλα. Το BPEL (Business Process Execution Language ), το οποίο συνθέτει τις νέες υπηρεσίες χρησιμοποιώντας τις ήδη υπάρχουσες. Στην ουσία αποτελεί μια γλώσσα <ενορχήστρωσης> που καθορίζει τον ρόλο της κάθε υπηρεσίας στην νέα σύνθεση. Το CDL (Choreography Description Language) από την άλλη πλευρά αποτελεί μια choreography γλώσσα ,η οποία καθορίζει την επικοινωνία ανάμεσα στις υπηρεσίες.

## ΚΕΦΑΛΑΙΟ 9

### SOAP – SIMPLE OBJECT ACCESS PROTOCOL

#### 9.1 Τι είναι το SOAP

Στα τέλη της δεκαετίας του 90, μεγάλες εταιρείες πληροφορικής όπως η Microsoft, άρχισαν να διερευνούν κατά πόσο ο κατακευματισμένος υπολογισμός μπορεί να βασιστεί στην XML. Ο σκοπός της έρευνας ήταν να γίνει εφικτή η επικοινωνία μεταξύ των εφαρμογών μέσω κλήσεων απομακρυσμένων διαδικασιών ( Remote Procedure Calls-RPCs ), χρησιμοποιώντας απλά πρωτόκολλα δικτύου, όπως το HTTP. Το 1999 έκανε την εμφάνιση του το Πρωτόκολλο Απλής Προσπέλασης Αντικειμένου ( Simple Object Access Protocol, SOAP ), ένας RPC μηχανισμός βασισμένος σε XML.

Το SOAP αποτελεί μια προδιαγραφή πρωτοκόλλου, που ορίζει με σαφήνεια την ανταλλαγή δομημένων πληροφοριών σε ένα κατακευματισμένο υπολογιστικό περιβάλλον ενώ αποτελείται από διάφορα μέρη. Το πρώτο μέρος είναι ο φάκελος, που χρησιμοποιείται για την περιγραφή του περιεχομένου ενός μηνύματος και μερικών ενδείξεων σχετικά με το πώς να το επεξεργαστεί. Το δεύτερο μέρος αποτελείται από τους κανόνες κωδικοποίησης των δομημένων δεδομένων. Το τελευταίο μέρος είναι μια σύμβαση για την σωστή αναπαράσταση κλήσεων διαδικασίας και απαντήσεων σε αυτές. Ακολουθεί μια λεπτομερής ανάλυση του SOAP.

#### 9.2 Εισαγωγή στις βασικές έννοιες

Η εμφάνιση του SOAP το 1999 τράβηξε την προσοχή του οργανισμού W3C και ύστερα από μια διαδικασία αλλαγών, βελτιώσεων και απαραίτητων τροποποιήσεων, το 2003 το πρωτόκολλο SOAP με την έκδοση 1.2 έγινε η προτεινόμενη προδιαγραφή-πρωτόκολλο για τις Υπηρεσίες Web. Σύμφωνα με μια σύσταση του W3C, ένας ορισμός που αντιπροσωπεύει την προδιαγραφή του SOAP είναι ο εξής:

"Το SOAP είναι ένα ελαφρύ πρωτόκολλο προορισμένο για την ανταλλαγή δομημένων πληροφοριών σε ένα αποκεντρωμένο, διανεμημένο περιβάλλον. Χρησιμοποιεί τεχνολογίες XML για να καθορίσει ένα επεκτάσιμο πλαίσιο παρέχοντας μια δομή μηνυμάτων η οποία μπορεί να ανταλλαχτεί πάνω από ποικίλα δικτυακά πρωτόκολλα. Το πλαίσιο έχει σχεδιαστεί να είναι ανεξάρτητο από οποιοδήποτε προγραμματιστικό μοντέλο και σημασιολογία υλοποίησης.

Στον πυρήνα του το SOAP χρησιμοποιείται κατά την ανταλλαγή μηνυμάτων μεταξύ των web services και είναι ένα ελαφρύ πλαίσιο ανταλλαγής μηνυμάτων και δομημένων δεδομένων σύμφωνα με τη γλώσσα XML, με την οποία περιγράφεται το περιεχόμενο των μηνυμάτων που μεταφέρει. Ακριβώς λόγω του ότι το SOAP αποτελεί ένα XML σχήμα, υποστηρίζεται από την πλειοψηφία των υφιστάμενων και νέων τεχνικών περιβαλλόντων και πλατφόρμων και μπορεί να μεταφερθεί από ποικίλα πρωτόκολλα μεταφοράς. Ο πιο διαδεδομένος τρόπος ανταλλαγής SOAP

μηνυμάτων ωστόσο είναι μέσω του HTTP, το οποίο χρησιμοποιείται ευρέως στο Διαδίκτυο καθώς χρησιμοποιείται από τους browsers για την πλοήγηση στον παγκόσμιο ιστό. Τα δεδομένα μεταφέρονται ακριβώς όπως οι ιστοσελίδες και δεν εμποδίζονται από τείχη ασφαλείας ( firewalls ) που ενδέχεται να διαθέτουν οι υπολογιστές.

Το SOAP αποτελεί ένα σύνολο συμβάσεων που όχι μόνο καθορίζουν και περιγράφουν ένα μήνυμα αλλά ορίζουν και τις αλληλεπιδράσεις των κόμβων SOAP, οι οποίοι επεξεργάζονται τα μηνύματα κατά μήκος της διαδρομής τους. Ο κύριος στόχος του είναι η παροχή ενός ενιαίου τρόπου για το πακετάρισμα των δεδομένων των μηνυμάτων και ο καθορισμός κανόνων για τη κωδικοποίηση και αποκωδικοποίηση των δεδομένων κατά την μεταφορά. Τα βασικά χαρακτηριστικά του πρωτόκολλου SOAP είναι τα εξής :

**1. ΑΠΛΟΤΗΤΑ:** Εξαιτίας αυτού του χαρακτηριστικού του, το κόστος αλλά και η πολυπλοκότητα υλοποίησης του πρωτόκολλου SOAP μειώνονται αισθητά.

**2. ΕΥΕΛΙΞΙΑ:** Το συγκεκριμένο πρωτόκολλο κάνει χρήση προτύπων πρωτοκόλλων όπως το HTTP ως μέσα μεταφοράς. Συνεπώς είναι αρκετά ευέλικτο καθώς μπορεί να χρησιμοποιηθεί στο Διαδίκτυο χωρίς συμβιβασμούς στην ασφάλεια της υποδομής μιας επιχείρησης.

**3. ΑΝΕΞΑΡΤΗΣΙΑ:** Είναι ανεξάρτητο από πλατφόρμα και γλώσσα προγραμματισμού οπότε μπορεί να χρησιμοποιηθεί για την επικοινωνία μεταξύ εφαρμογών γραμμένων σε διαφορετικές γλώσσες προγραμματισμού και για διαφορετικές πλατφόρμες.

**4. ΕΠΕΚΤΑΣΙΜΟ:** Αποτελεί το πιο σημαντικό του χαρακτηριστικό καθώς έτσι μπορούν να βασιστούν επάνω του πολλές αναπτυσσόμενες τεχνολογίες υπηρεσιών παγκοσμίου ιστού προσφέροντας αξιοπιστία και ασφάλεια.

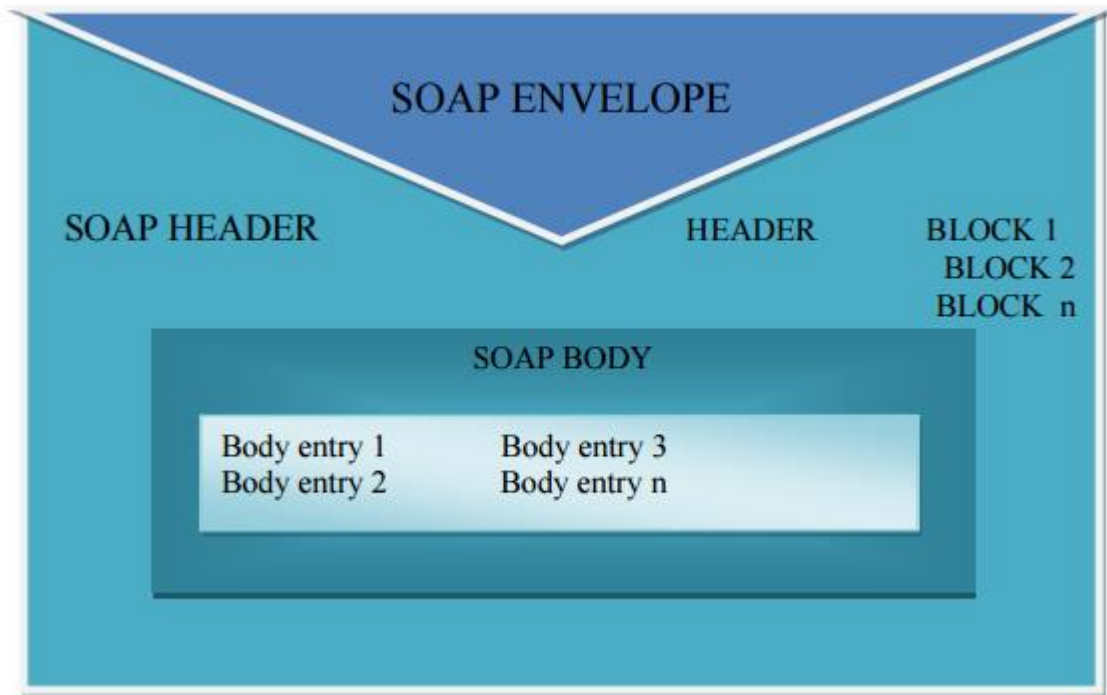
Τα παραπάνω χαρακτηριστικά είναι μόλις κάποια από τα βασικά πλεονεκτήματα του πρωτοκόλλου SOAP, το οποίο όπως έχει προαναφερθεί καθορίζει σαφώς το πλαίσιο ανταλλαγής μηνυμάτων μεταξύ των web services.

### 9.3 Μήνυμα SOAP

Το πρωτόκολλο SOAP αποτελεί ένα συνηθισμένο έγγραφο XML, το οποίο χρησιμοποιείται για την ανταλλαγή πληροφορίας μέσω δομής μηνυμάτων. Η δομή ενός μηνύματος SOAP, το οποίο αποτελεί και αυτό ένα έγγραφο XML, είναι πολύ οικεία καθώς θα μπορούσε να συμβολιστεί με την απλή δομή των γραμμάτων που ταχυδρομούνται στην πραγματική ζωή και έχει ως εξής:

• **ENVELOPE ( ΦΑΚΕΛΟΣ ):** Αποτελεί ουσιαστικά το XML στοιχείο-ρίζα ενός μηνύματος SOAP μέσω του οποίου αναγνωρίζεται το XML έγγραφο ως μήνυμα SOAP. Σε αυτό το στοιχείο περιέχονται όλα τα άλλα στοιχεία XML του μηνύματος SOAP.

- **HEADER ( ΕΠΙΚΕΦΑΛΙΔΑ )**: Το συγκεκριμένο στοιχείο είναι πολύ σημαντικό για λόγους που θα παρουσιάσουμε στην συνέχεια και περικλείει ένα επιπρόσθετο και προαιρετικό είδος πληροφορίας όπως η ασφάλεια και οι συναλλαγές.
- **BODY ( ΣΩΜΑ )**: Αποτελεί υποχρεωτικό στοιχείο του μηνύματος και περιέχει τα πραγματικά ανταλλασσόμενα δεδομένα. Στην ουσία αυτό το στοιχείο αποτελεί τον σκοπό του μηνύματος.



Εικόνα 9.1: Δομή ενός μηνύματος SOAP

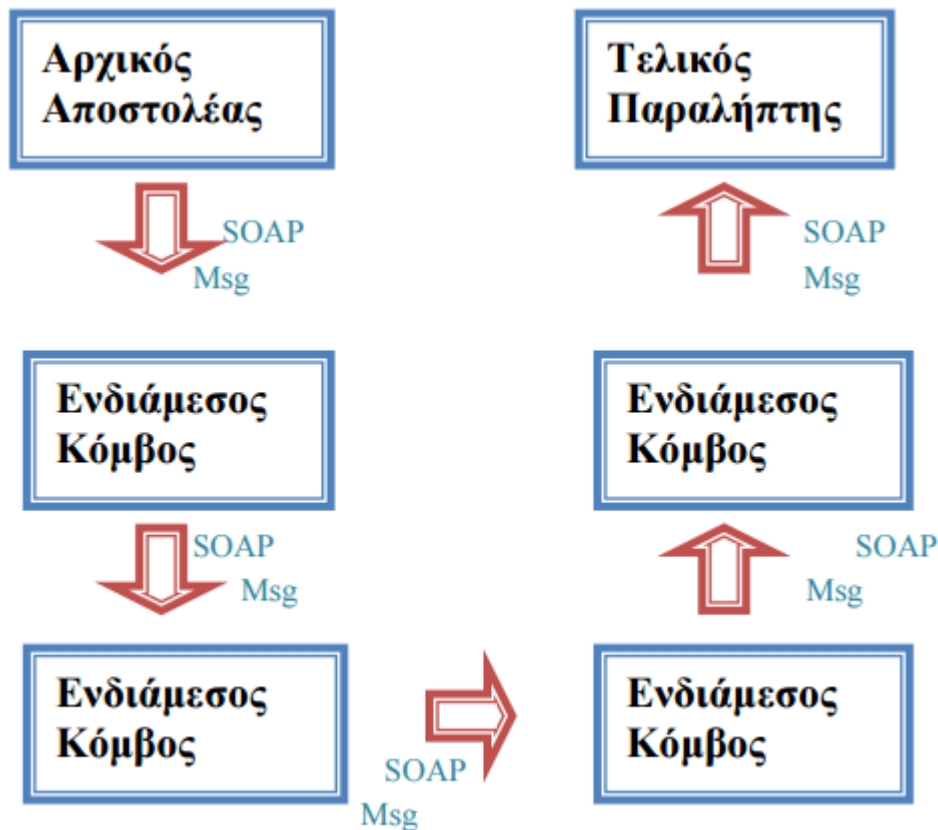
#### 9.4 Μοντέλο επεξεργασίας SOAP

Το SOAP είναι βασικά ένα μοντέλο μονόδρομης επικοινωνίας, το οποίο εγγυάται ότι ένα μήνυμα μεταφέρεται από τον αποστολέα στον παραλήπτη, ενδεχομένως περιλαμβάνοντας ενδιάμεσους σταθμούς ( κόμβους ) που μπορούν να επεξεργαστούν μέρος του μηνύματος ή και να το μεταβάλουν.

Το μοντέλο επεξεργασίας SOAP περιγράφει αυτό το κατανεμημένο μοντέλο επεξεργασίας, τους συμμετέχοντες, τους κόμβους SOAP και πως ένας δέκτης SOAP επεξεργάζεται ένα μήνυμα SOAP. Πιο συγκεκριμένα ένας κόμβος αναπαριστά τη λογική που είναι υπεύθυνη για τη μεταβίβαση, λήψη, επεξεργασία ή και αναμετάδοση ενός μηνύματος SOAP.

Σε γενικές γραμμές οι κόμβοι SOAP έχουν τη δυνατότητα να στέλνουν και να λαμβάνουν μηνύματα SOAP. Κάθε κόμβος ανάλογα με την λειτουργία που εκτελεί σε σχέση με το μήνυμα SOAP έχει και την ανάλογη ιδιότητα. Αναλυτικότερα ένας κόμβος λέγεται αποστολέας SOAP όταν μεταδίδει το μήνυμα, παραλήπτης SOAP όταν λαμβάνει το μήνυμα, ενώ ενδιάμεσος SOAP όταν ταυτόχρονα λαμβάνει και στην συνέχεια μεταδίδει το μήνυμα SOAP. Ο αρχικός κατασκευαστής του

μηνύματος ονομάζεται Αρχικός Αποστολέας ενώ ο τελικός αποδέκτης λέγεται Τελικός Παραλήπτης. Στο σχήμα που ακολουθεί παρουσιάζεται μια αναπαράσταση της διαδρομής που ακολουθεί κάθε μήνυμα SOAP ώστε να καταλήξει από τον αποστολέα στο παραλήπτη.



Εικόνα 9.2: Διαδρομή Μηνύματος SOAP

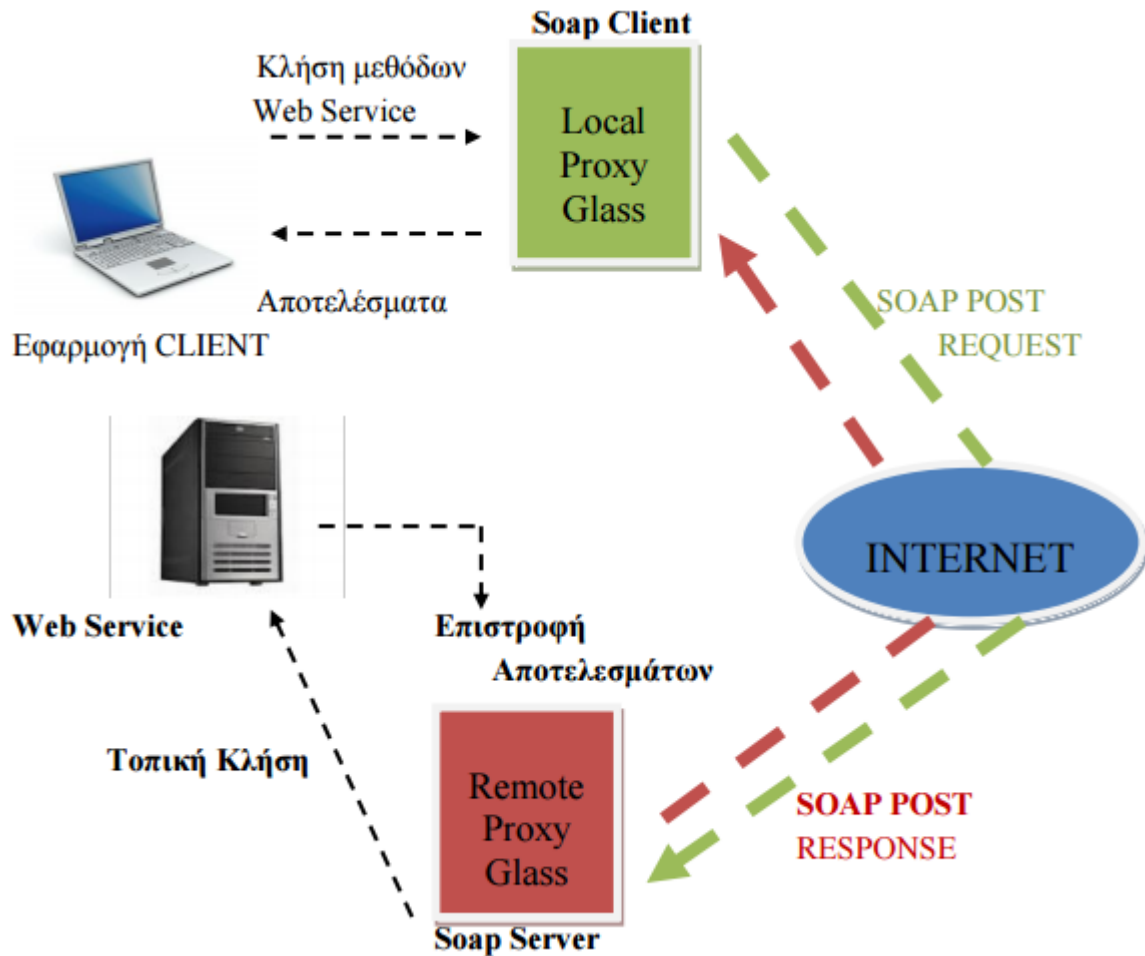
## 9.5 Μοντέλο ανταλλαγής μηνυμάτων

Το SOAP είναι ένα απλό πλαίσιο μηνυμάτων για ανταλλαγή μηνυμάτων σε μορφή XML μεταξύ ενός αρχικού αποστολέα και ενός τελικού αποδέκτη. Τα πιο ενδιαφέροντα σενάρια περιλαμβάνουν πολλαπλές ανταλλαγές μηνυμάτων μεταξύ των δυο αυτών κόμβων. Η πιο απλή μορφή είναι αυτής της αίτησης-απάντησης.

## 9.6 Απομακρυσμένες Κλήσεις Διαδικασιών

Κάθε μήνυμα SOAP ταυτόχρονα με όλες τις παραπάνω λειτουργίες του, υλοποιεί κλήσεις απομακρυσμένης διαδικασίας RPC( Remote Procedure Call), κάτι το οποίο αποτελούσε και ένα από τους στόχους κατά τον σχεδιασμό του. Πιο συγκεκριμένα για την κλήση μιας μεθόδου ενός απομακρυσμένου αντικειμένου

μιας Web Service έχει καθοριστεί μια συγκεκριμένη αναπαράσταση μηνύματος SOAP, η οποία περιλαμβάνει στο σώμα του το όνομα της μεθόδου, τις επιτρεπόμενες παραμέτρους και το URL της. Το μοντέλο που ακολουθείται είναι κυρίως της μορφής αίτησης – απάντησης ενώ η όλη διαδικασία πραγματοποιείται μεταξύ του πελάτη και του διακομιστή. Ακολουθεί ένα σχήμα το οποίο θα μας βοηθήσει να κατανοήσουμε αυτή την αμφίδρομη επικοινωνία.



Εικόνα 9.3: Κλήση των Web Services

Στο συγκεκριμένο παράδειγμα ο client εκτελεί κλήση με το μοντέλο request – response. Παρατηρούμε ότι εδώ ο client εκτελεί αίτηση για συγκεκριμένη web service στον διακομιστή μεσολάβησης και αυτός με τη σειρά του αναλαμβάνει να προσκομίσει σε αυτόν τις σελίδες που ζητάει.

## 9.7 HTTP Binding

Λόγω της ευρείας χρήσης του στο Internet αλλά και της απλότητας του, το HTTP έχει χρησιμοποιηθεί σε μεγάλο βαθμό για την μεταφορά μηνυμάτων SOAP. Ο συσχετισμός του SOAP με το HTTP πρωτόκολλο έχει τυποποιηθεί από τον W3C και σύμφωνα με το πρότυπο είναι δυνατή η υποστήριξη του μοντέλου αμφίδρομης επικοινωνίας για ανταλλαγή μηνυμάτων (request-response) μέσω του συσχετισμού του SOAP με τη μέθοδο HTTP POST. Σε αυτή την περίπτωση εφόσον ο Αρχικός Αποστολέας αναγνωρίσει τον εξυπηρετητή μέσω ενός URI, συνδέεται σε αυτόν χρησιμοποιώντας το TCP/IP. Στην συνέχεια στέλνει το αρχικό μήνυμα SOAP από τον κόμβο του σε μία HTTP POST ΑΙΤΗΣΗ (request) ενώ το απαντητικό μήνυμα αποστέλλεται από τον Τελικό Παραλήπτη στην HTTP POST ΑΠΑΝΤΗΣΗ (response). Το HTTP συνδέει το μήνυμα αίτησης με το μήνυμα απάντησης.

## ΚΕΦΑΛΑΙΟ 10

### WSDL - WEB SERVICES DESCRIPTION LANGUAGE

#### 10.1 Τι είναι το WSDL

Στο προηγούμενο κεφάλαιο υπήρξε μια πλήρης αναφορά στο πρωτόκολλο SOAP. Είδαμε ότι το συγκεκριμένο πρωτόκολλο είναι βασισμένο στην XML και ότι μέσω αυτού του πρωτοκόλλου μπορούν να γίνουν αιτήσεις για απομακρυσμένες κλήσεις διαδικασιών από κάποιον client σε κάποιον server. Από την μεριά του πελάτη όμως υπήρχαν δυσκολίες στο καθορισμό της σύνταξης του μηνύματος SOAP, καθώς ο πελάτης δε μπορούσε να ξέρει τι ακριβώς μήνυμα να στείλει. Αν και το πρωτόκολλο SOAP καθορίζει κάποιους κανόνες και ένα συγκεκριμένο μορφότυπο για τα μηνύματα, ωστόσο δεν διέθετε κανέναν μηχανισμό στον πελάτη ώστε να είναι σε θέση να γνωρίζει αρκετές λεπτομέρειες προτού στείλει το μήνυμα, όπως που ακριβώς να το στείλει, ποιες μεθόδους πρέπει να καλέσει από την υπηρεσία ή ποια πρωτόκολλα επικοινωνίας υποστηρίζονται από τον παροχέα της υπηρεσίας.

Είναι εμφανές λοιπόν ότι πελάτης θα πρέπει να είναι σε θέση να γνωρίζει όλες τις λεπτομέρειες για την κλήση μιας υπηρεσίας διαδικτύου, συνεπώς να μπορεί να έχει στην κατοχή του μια πλήρη περιγραφή αυτής. Γι' αυτό το λόγο δημιουργήθηκε ένας καθορισμένος μηχανισμός, η γλώσσα περιγραφής Υπηρεσιών Παγκοσμίου Ιστού WSDL ( Web Services Description Language), με σκοπό την παροχή πληροφοριών σχετικά με όλα αυτά που χρειάζεται να γνωρίζει ο πελάτης ώστε να καλέσει μια υπηρεσία. Ακολουθεί μια πλήρης αναφορά

#### 10.2 Εισαγωγή στις βασικές έννοιες

Σύμφωνα με μια σύσταση του W3C, ένας ορισμός που αντιπροσωπεύει την προδιαγραφή του SOAP είναι ο εξής:

"Η WSDL είναι ένα σχήμα XML για την περιγραφή δικτυακών υπηρεσιών σαν ένα σύνολο από τελικά σημεία που λειτουργούν σε μηνύματα τα οποία περιέχουν πληροφορία είτε προσανατολισμένη στα έγγραφα είτε προσανατολισμένη στις διαδικασίες.

Οι λειτουργίες και τα μηνύματα περιγράφονται περιληπτικά, και τότε δένονται σε ένα συγκεκριμένο πρωτόκολλο δικτύων και μορφή μηνυμάτων για να καθορίζουν ένα τελικό σημείο. Πολλά σχετικά τελικά σημεία συνδυάζονται σε υπηρεσίες.

Η WSDL είναι επεκτάσιμη στο να επιτρέπει την περιγραφή τελικών σημείων και των μηνυμάτων τους άσχετα από τη μορφή των μηνυμάτων και των πρωτοκόλλων δικτύων που χρησιμοποιούνται για την επικοινωνία. Παρόλα αυτά, αυτή τη στιγμή στην προδιαγραφή της WSDL οι μόνες συνδέσεις που περιλαμβάνονται περιγράφουν πως μπορούμε να χρησιμοποιήσουμε την WSDL σε συνδυασμό με το SOAP 1.1, το HTTP GET/POST και το MIME"

Με απλά λόγια η WSDL μας βοηθάει να περιγράψουμε ένα σύνολο από μηνύματα και τον τρόπο με τον οποίο αυτά τα μηνύματα ανταλλάσσονται. Η



WSDL βασίζεται στην XML για την περιγραφή των δημόσιων διεπαφών ( public interfaces ) μιας υπηρεσίας παγκοσμίου ιστού. Η δημόσια διεπαφή μπορεί να περιλαμβάνει πληροφορία σχετική με:

- Όλες τις διαθέσιμες δημόσιες δραστηριότητες της υπηρεσίας παγκοσμίου ιστού.
- Τα πρωτόκολλα XML μηνυμάτων που υποστηρίζουν Web Service.
- Το είδος δεδομένων
- Τη δέσμευση για το συγκεκριμένο πρωτόκολλο μεταφοράς που πρόκειται να χρησιμοποιηθεί.
- Τη διεύθυνση για τον εντοπισμό της συγκεκριμένης υπηρεσίας παγκοσμίου ιστού.

Η WSDL είναι ανεξάρτητη ως προς την πλατφόρμα και την γλώσσα προγραμματισμού. Αυτό το γεγονός την κάνει κατάλληλη για να περιγράψει διεπαφές web services, οι οποίες είναι προσβάσιμες από μια μεγάλη ποικιλία πλατφόρμων και γλωσσών προγραμματισμού. Επιπρόσθετα εκτός του ότι περιγράφει τα περιεχόμενα των μηνυμάτων, ορίζει που είναι διαθέσιμη μια υπηρεσία και ποια πρωτόκολλα επικοινωνίας χρησιμοποιούνται για να επικοινωνήσουμε με αυτή την υπηρεσία. Επομένως κάθε αρχείο WSDL ορίζει όλα όσα χρειάζονται για να γράψουμε ένα πρόγραμμα το οποίο να λειτουργεί με ένα Web Service.

Συνοπτικά η WSDL περιγράφει τις ακόλουθες τρεις σημαντικές ιδιότητες κάθε υπηρεσίας web.

- Τι κάνει η Υπηρεσία:  
Μέθοδοι της  
Παράμετροι της  
Αποτελέσματα
- Πως γίνεται η πρόσβαση στην Υπηρεσία:  
Πρωτόκολλο  
Είδος κωδικοποίησης
- Που βρίσκεται η Υπηρεσία:  
Δικτυακή Διεύθυνση

### 10.3 Τεχνική περιγραφή της WSDL

Η γραμματική και η σύνταξη που περιγράφουν την υπηρεσία παγκοσμίου ιστού, καθορίζονται από την WSDL σαν μια συλλογή από επικοινωνιακά άκρα. Ένα έγγραφο WSDL καθορίζει τις υπηρεσίες Διαδικτύου ( web services ) ως ένα σύνολο από τελικά σημεία ή θύρες Υπηρεσίας( ports ). Η περιγραφή της κάθε θύρας Υπηρεσίας αποτελείται από τρία μέρη:

- **Τύπος Πόρτας Υπηρεσιών (portType)**, που αναφέρεται στις Λειτουργίες (operations) που προσφέρονται από τον συγκεκριμένο τύπο πόρτας. Κάθε λειτουργία περιγράφεται ως ανταλλαγή μηνυμάτων η οποία μπορεί να γίνει

μονόδρομα ή αμφίδρομα ή σύμφωνα με άλλα μοντέλα. Για κάθε μήνυμα που ανταλλάσσεται πρέπει να γίνει γνωστός ο τύπος των δεδομένων που μεταφέρει.

- **Συσχετισμός (binding)**, που αναφέρεται στο συσχετισμό ενός τύπου πόρτας υπηρεσιών σε συγκεκριμένο πρωτόκολλο επικοινωνίας και σε συγκεκριμένο τρόπο αναπαράστασης των δεδομένων που ανταλλάσσονται. Το πιο σύνηθες πρωτόκολλο επικοινωνίας με το οποίο συσχετίζονται οι τύποι πόρτας που ορίζονται στην WSDL είναι το πρωτόκολλο SOAP.

- **Διεύθυνση (address)**, που αναφέρεται στην ηλεκτρονική διεύθυνση από την οποία οι Λειτουργίες της συγκεκριμένης πόρτας υπηρεσιών είναι διαθέσιμες.

Μια περιγραφή σε WSDL αποτελεί ένα κείμενο XML το οποίο σέβεται τον ορισμό ενός συγκεκριμένου XML Schema (WSDL Schema definition). Σύμφωνα με αυτό ένα κείμενο XML έχει ένα στοιχείο-ρίζα το `wSDL:definitions`, όπου το 'wSDL' αντιστοιχεί στον ονοματοχώρο (namespace) `http://schemas.xmlsoap.org/wSDL/`. Το στοιχείο `wSDL:definitions` περιέχει έναν αριθμό στοιχείων τα οποία μπορούν να είναι τα εξής:

#### < TYPES >

Το συγκεκριμένο στοιχείο ορίζει τους τύπους δεδομένων που ανταλλάσσονται μέσω των μηνυμάτων όταν χρησιμοποιούνται οι διάφορες λειτουργίες που προσφέρει μία υπηρεσία ιστού μέσω των θυρών υπηρεσιών που παρουσιάζει.

#### < MESSAGE >

Το συγκεκριμένο στοιχείο ορίζει τη μορφή των μηνυμάτων που ανταλλάσσονται κατά την χρήση των λειτουργιών των θυρών υπηρεσίας.

#### < PORTTYPE >

Το συγκεκριμένο στοιχείο ορίζει όλες τις λειτουργίες που υποστηρίζονται από μια ή περισσότερες θύρες Υπηρεσίας. Ουσιαστικά μέσω αυτού του στοιχείου η κάθε Υπηρεσία Web ορίζεται σαν το σύνολο των λειτουργιών που υποστηρίζονται και των μηνυμάτων που ανταλλάσσονται.

#### < OPERATION >

Το συγκεκριμένο στοιχείο ορίζει όλες τις λειτουργίες που υποστηρίζει μια συγκεκριμένη Υπηρεσία.

#### < BINDING >

Κάθε τέτοιο στοιχείο ορίζει το πως κάθε ανταλλασσόμενο μήνυμα στα πλαίσια κάθε λειτουργίας μπορεί να συσχετιστεί με ένα πρωτόκολλο ανταλλαγής μηνυμάτων για ένα συγκεκριμένο τύπο θυρών Υπηρεσίας.

#### < SERVICE >

Το συγκεκριμένο στοιχείο αποτελεί μια συλλογή από σχετικά τελικά σημεία ή θύρες Υπηρεσίας ( ports ). Στην ουσία το συγκεκριμένο στοιχείο χρησιμοποιείται για να περικλείσει μέσα του το σύνολο των πληροφοριών που χρησιμοποιούνται για να περιγράψουν κάθε υπηρεσία παγκοσμίου Ιστού.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 11

# UDDI - UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION

### 11.1 Τι είναι το UDDI

Στο προηγούμενο κεφάλαιο αναφερθήκαμε στην WSDL, την γλώσσα μέσω της οποίας ο κάθε πάροχος μιας Υπηρεσίας Διαδικτύου περιγράφει την υπηρεσία του έτσι ώστε να γνωρίζει ο κάθε πιθανός πελάτης τι κάνει αυτή η Υπηρεσία, πώς να εγκαθιδρύσει επικοινωνιακή σύνδεση με τον πάροχο της καθώς και τον τρόπο με τον οποίο πρέπει να την καλέσει. Το ερώτημα είναι που δημοσιεύει ο κάθε πάροχος την περιγραφή της Υπηρεσίας του έτσι ώστε να είναι προσβάσιμη σε όλους τους πιθανούς πελάτες.

Όπως έχει ήδη αναφερθεί σε προηγούμενο κεφάλαιο στην SOA αρχιτεκτονική υπάρχει ο ρόλος της Υπηρεσίας Καταλόγου. Ο πελάτης προκειμένου να ανακαλύψει την Υπηρεσία που τον ενδιαφέρει, ψάχνει πρώτα στον κατάλογο αυτό και είναι εδώ το σημείο όπου ο κάθε πάροχος μιας Υπηρεσίας δημοσιεύει την περιγραφή της Υπηρεσίας του. Η επικρατούσα τεχνολογία καταλόγου στον κόσμο των web services είναι το UDDI ( Universal Description, Discovery and Integration ), για το οποίο ακολουθεί μια σχετικά λεπτομερής περιγραφή.

### 11.2 Εισαγωγή στις βασικές έννοιες

Η αρχιτεκτονική SOA υποστηρίζει τόσο την δυναμική σύνδεση με Υπηρεσίες Web όσο και την δυνατότητα επαναχρησιμοποίησης τους μέσω της σύνθεσης υπάρχοντων υπηρεσιών σε νέες. Συνεπώς η δημιουργία του Προτύπου Καταλόγου UDDI το 2000 αποσκοπούσε στην διευκόλυνση ανακάλυψης των Υπηρεσιών Διαδικτύου τόσο κατά τον χρόνο σχεδίασης όσο και δυναμικά κατά τον χρόνο εκτέλεσης τους.

Το UDDI παρέχει ένα παγκόσμιο αρχείο καταγραφής Υπηρεσιών Διαδικτύου για να διευκολύνει τις επιχειρήσεις στη γρήγορη, εύκολη και δυναμική εύρεση και συναλλαγή με άλλες επιχειρήσεις. Ουσιαστικά για πρώτη φορά λειτούργησε στις 2 Μαΐου 2001 και αποτελείται στην πραγματικότητα από 2 πανομοιότυπους καταλόγους, που διατηρούνται αυτή τη στιγμή από δύο εταιρείες (IBM και Microsoft), οι οποίες αποτελούν και τους διαχειριστές του UDDI.

Σύμφωνα με μια σύσταση του OASIS, ο σκοπός για το UDDI( Universal Description, Discovery and Integration ) είναι ο εξής:

"Τα web services έχουν νόημα μόνο όταν δυνητικοί χρήστες μπορούν να βρουν πληροφορίες ικανές ώστε να επιτρέψουν την εκτέλεση τους. Το UDDI ( Universal Description, Discovery and Integration ) εστιάζει στον καθορισμό ενός συνόλου από υπηρεσίες που θα υποστηρίξουν την περιγραφή και την ανακάλυψη :

- Των εταιρειών, των οργανισμών και άλλων παρόχων web services.
- Των web services που είναι διαθέσιμες

- Των τεχνικών διεπαφών οι οποίες μπορούν να χρησιμοποιηθούν ώστε να έχει κάποιος χρήστης πρόσβαση σε αυτές τις υπηρεσίες.

Βασισμένο σε ένα κοινό σύνολο από βιομηχανικά πρότυπα συμπεριλαμβανομένων των HTTP, XML, XML Schema και SOAP παρέχει μια δυσλειτουργική, θεμελιώδη υποδομή για ένα περιβάλλον λογισμικού προσανατολισμένο στις υπηρεσίες τόσο για δημόσια διαθέσιμες υπηρεσίες όσο και για υπηρεσίες που εκτίθενται μόνο εσωτερικά ενός οργανισμού. "

Με πολύ απλά λόγια το UDDI αποτελεί τον Χρυσό Οδηγό των web services. Ορίζει τον τρόπο καταχώρησης των Web Services στο μητρώο και είναι και αυτό βασισμένο στην γλώσσα XML. Στόχος του είναι η παροχή αναγκαίας υποδομής για την περιγραφή και αναζήτηση Web Services. Έτσι, προσφέρεται πρόσβαση σε έναν κατάλογο που περιλαμβάνει τις επιχειρήσεις, τις Υπηρεσίες Ιστού που αυτές προσφέρουν και τον τρόπο με τον οποίο οι υπηρεσίες αυτές θα μπορούσαν να χρησιμοποιηθούν.

Συγκεκριμένα το UDDI απευθύνεται σε δυο ειδών πελάτες. Τις επιχειρήσεις( Πάροχοι Υπηρεσίας ) που μπορούν να δημοσιεύσουν μια πλήρη περιγραφή της υπηρεσίας που παρέχουν και τους πελάτες ( Αιτουμένους Υπηρεσίας ) που επιθυμούν να χρησιμοποιήσουν συγκεκριμένες Υπηρεσίες συνδεδεμένοι προγραμματιστικά με αυτές. Εξαιτίας του Καταλόγου UDDI όλες οι Υπηρεσίες έχουν το δικαίωμα να καταχωρούνται πλέον σε ένα Μητρώο Υπηρεσιών ( UDDI Service Register ) και από εκεί είναι προσπελάσιμες τόσο από τον Πάροχο Υπηρεσίας ( Service Provider ) όσο και από τον Πελάτη (Service Requestor ). Είναι σαφές λοιπόν ότι υπάρχει μια αναγκαία αλληλεπίδραση μεταξύ του Παρόχου Υπηρεσίας, του Αιτούμενου και του Μητρώου Υπηρεσιών. Αυτή η αλληλεπίδραση πραγματοποιείται μέσω των εξής τριών βασικών λειτουργιών:

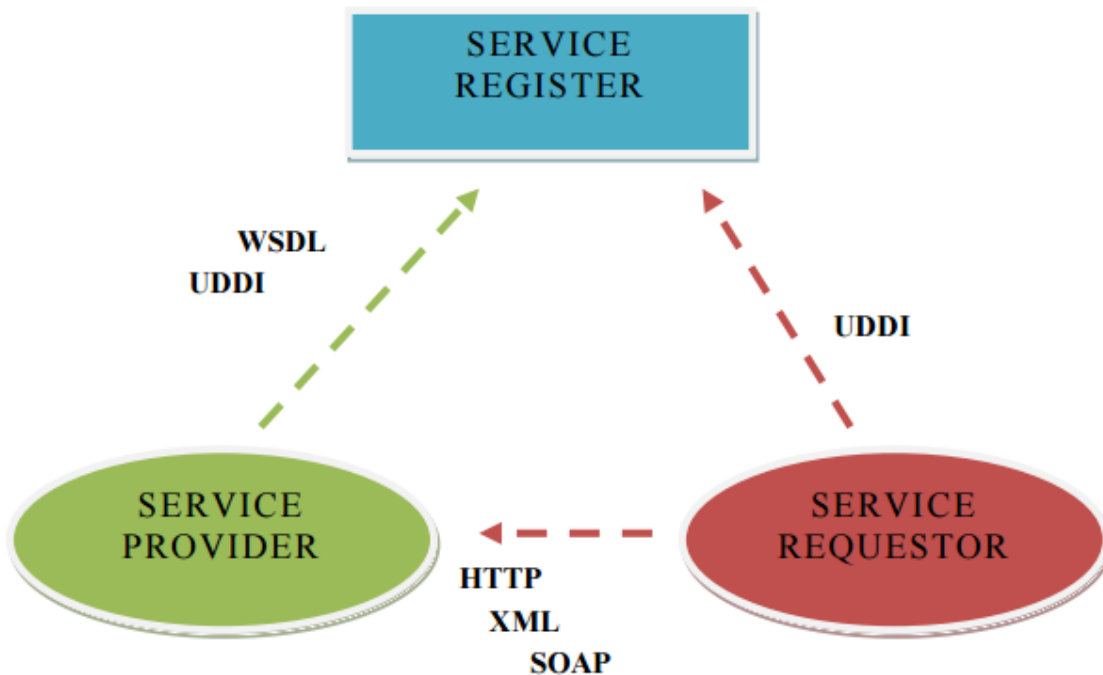
- PUBLISH:Δημοσίευση περιγραφής Υπηρεσιών
- FIND:Εύρεση περιγραφών Υπηρεσιών
- BIND:Επίκληση Υπηρεσιών

Ακολουθεί μια εικονική αναπαράσταση του τρόπου αλληλεπίδρασης των τριών υπηρεσιών.

Ο πάροχος Υπηρεσίας μέσω του WSDL και του UDDI δημοσιοποιεί ( publish ) την περιγραφή της υπηρεσίας του στον UDDI κατάλογο Υπηρεσίας.

Ο Αιτούμενος αναζητάει ( find ) μια Υπηρεσία μέσω του καταλόγου UDDI.

Ο Αιτούμενος Υπηρεσίας εφόσον ανακαλύψει μέσω του καταλόγου την Υπηρεσία που τον ενδιαφέρει συνδέεται-αλληλεπιιδρά ( bind ) με τον αντίστοιχο Πάροχο μέσω των πρωτοκόλλων HTTP, SOAP και XML.



Εικόνα 11.1: Αλληλεπίδραση Υπηρεσιών

### 11.3 Ορισμός δομών δεδομένων

Όλες οι εταιρείες και οι οργανισμοί που επιθυμούν να δημιουργήσουν μια Υπηρεσία Διαδικτύου, έχουν το δικαίωμα εφόσον το επιθυμούν, να δημοσιοποιούν δημόσιες πληροφορίες γι' αυτούς και πληροφορίες για τις Υπηρεσίες Διαδικτύου που παρέχουν μέσω του προτύπου Καταλόγου UDDI. Η πληροφορία στο UDDI ταξινομείται σύμφωνα με τον τύπο των εταιρειών (πάροχοι Υπηρεσιών Διαδικτύου) και την γεωγραφική τους θέση. Όταν κάποια επιχείρηση ψάξει να βρει κάποια άλλη επιχείρηση βάση των παραπάνω κριτηρίων, το UDDI μέσω ενός δείκτη στο WSDL αρχείο περιγράφει τις Υπηρεσίες Διαδικτύου που παρέχει η συγκεκριμένη επιχείρηση-Πάροχος.

Μπορούμε να ταξινομήσουμε τις παρεχόμενες Υπηρεσίες του UDDI, με βάση την δομή ενός τηλεφωνικού καταλόγου οπότε θα είχαμε

- Λευκές Σελίδες που περιέχουν γενικές πληροφορίες επικοινωνίας με τον πάροχο.
- Κίτρινες Σελίδες που περιέχουν πληροφορίες ταξινόμησης επιχειρήσεων και οργανισμών ανάλογα με την λειτουργία τους για τους τύπους και την τοποθεσία των υπηρεσιών.
- Πράσινες Σελίδες που περιέχουν τεχνικές πληροφορίες πρόσβασης στις προσφερόμενες υπηρεσίες του συγκεκριμένου παρόχου.

Κάθε καταχώρηση ενός παρόχου σε ένα Μητρώο Υπηρεσιών (UDDI Service Register), ουσιαστικά είναι ένα XML έγγραφο που αποτελείται κυρίως από τους εξής τέσσερις τύπους δεδομένων:

### **businessEntity**

Ο συγκεκριμένος τύπος περιγράφει οποιαδήποτε πάροχο προσφέρει Υπηρεσίες Ιστού και έχει επιλέξει να τις καταχωρήσει στον Μητρώο Υπηρεσιών ( UDDI Service Register ) και παρέχει λεπτομέρειες για το πως μπορεί να επικοινωνήσει κάποιος μαζί του, κλπ.

### **businessService**

Ο συγκεκριμένος τύπος δεδομένων περιγράφει το σύνολο των Υπηρεσιών Διαδικτύου που παρέχονται από έναν συγκεκριμένο οργανισμό ή επιχείρηση( Πάροχος Υπηρεσίας ), ο οποίος περιγράφεται με το τύπο businessEntity.

### **bindingTemplate**

Ο συγκεκριμένος τύπος δεδομένων περιγράφει όλες τις τεχνικές πληροφορίες υλοποίησης μιας Υπηρεσίας Διαδικτύου.

### **tModel**

Ο συγκεκριμένος τύπος εμπεριέχει επαναχρησιμοποιήσιμες έννοιες που προσφέρουν το τεχνικό μοντέλο (technical model) περιγραφής προδιαγραφών των Υπηρεσιών όπως ένας τύπος υπηρεσίας ιστού, ένα πρωτόκολλο για επικοινωνία με υπηρεσίες ιστού. Έτσι, στοιχεία tModel μπορούν να αναπαριστούν πρωτόκολλα μεταφοράς δεδομένων όπως HTTP ή SMTP, φόρμα ταχυδρομικών διευθύνσεων κ.α.

## ΚΕΦΑΛΑΙΟ 12

### HTML

#### 12.1 Τι είναι η HTML

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

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

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

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

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



## 12.2 Προέλευση

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN. Στις προσωπικές του σημειώσεις από το 1990, ο Μπέρνερς Λι αριθμεί «μερικές από τις πολλές χρήσεις του υπερκειμένου», και αναφέρει πρώτα από όλες μια εγκυκλοπαίδεια.

## 12.3 Πρώτες προδιαγραφές

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

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

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993, η οποία περιλάμβανε και έναν Ορισμό τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το ανταγωνιστικό πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη

υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.

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

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

## 12.4 Σήμανση

Η σήμανση HTML αποτελείται από μερικά βασικά συστατικά, συμπεριλαμβανομένων των στοιχείων (και των ιδιοτήτων τους), τους βασισμένους σε χαρακτήρες τύπους δεδομένων, τις αναφορές χαρακτήρων και τις αναφορές οντοτήτων. Ένα ξεχωριστό σημαντικό συστατικό είναι η δήλωση τύπου εγγράφου (document type declaration), η οποία ορίζει στον browser τον τρόπο εμφάνισης της σελίδας.

Στην HTML, το πρόγραμμα Hello world, ένα συνηθισμένο πρόγραμμα υπολογιστή που χρησιμεύει για τη σύγκριση γλωσσών προγραμματισμού, γλωσσών σεναρίων και γλωσσών σήμανσης, φτιάχνεται με 9 γραμμές κώδικα, παρότι οι νέες γραμμές είναι προαιρετικές στην HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

**Εικόνα 12.1:** Δείγμα προγράμματος “Hello World” γραμμένο σε HTML

Το κείμενο ανάμεσα στο `<html>` και το `</html>` περιγράφει την ιστοσελίδα, και το κείμενο μεταξύ του `<body>` και του `</body>` είναι το ορατό μέρος της. Το σημασμένο κείμενο `<title>Hello HTML</title>` καθορίζει τον τίτλο που θα εμφανίζεται στην μπάρα τίτλου του browser.)

Το Document Type Declaration στον πιο πάνω κώδικα είναι για την HTML5. Αν δεν συμπεριλαμβάνεται η δήλωση `<!doctype html>`, τότε μερικοί browser θα καταφύγουν στην λειτουργία quirks για την εμφάνιση της σελίδας.

## 12.5 Στοιχεία

Τα έγγραφα HTML αποτελούνται από στοιχεία HTML τα οποία στην πιο γενική μορφή τους έχουν τρία συστατικά: ένα ζεύγος από ετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», μερικές ιδιότητες μέσα στην ετικέτα εκκίνησης, και τέλος το κείμενο ή το γραφικό περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από» και «μικρότερο από», δηλαδή `<` και `>`.

Επομένως, η γενική μορφή ενός στοιχείου HTML είναι: `<ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">περιεχόμενο</ετικέτα>`. Μερικά στοιχεία HTML περιγράφονται ως άδεια στοιχεία, έχουν τη μορφή `<ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">`, και δεν έχουν καθόλου περιεχόμενο. Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια κάθετο `</>`, η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση.

## 12.6 Η δομική σήμανση περιγράφει τον σκοπό του κειμένου

Για παράδειγμα, το `<h2>Golf</h2>` σημαίνει ότι η λέξη «Golf» θα είναι μια επικεφαλίδα δεύτερου επιπέδου. Η δομική σήμανση δεν εμπεριέχει συγκεκριμένο τρόπο εμφάνισης, αλλά οι περισσότεροι φυλλομετρητές έχουν δικό τους προκαθορισμένο τρόπο για τη μορφοποίηση όλων των στοιχείων. Το περιεχόμενο μπορεί να μορφοποιηθεί οπτικά με τη χρήση των CSS.

## 12.7 Η σήμανση οπτικής μορφοποίησης και κείμενο

Για παράδειγμα το `<b>έντονο κείμενο</b>` υποδεικνύει ότι οι συσκευές που θα εμφανίσουν το κείμενο θα το κάνουν έντονο, αλλά δεν λέει τι θα κάνουν με το κείμενο οι συσκευές που δεν μπορούν να το εμφανίσουν, όπως για παράδειγμα οι συσκευές φωνητικής ανάγνωσης σελίδων. Τόσο στην περίπτωση του `<b>έντονου</b>` όσο και του `<i>πλάγιου</i>`, υπάρχουν άλλα στοιχεία τα οποία μπορεί οπτικά να εμφανίζονται ίδια, αλλά είναι πιο σημαντικά, όπως

το **σημαντικό κείμενο** και το *κείμενο με έμφαση* αντίστοιχα. Είναι ευκολότερο να ερμηνεύσει ένας φωνητικός φυλλομετρητής τα τελευταία δύο στοιχεία. Ωστόσο, δεν είναι ισότιμα με τα αντίστοιχα στοιχεία οπτικής μορφοποίησης. Για παράδειγμα ένα πρόγραμμα φωνητικής ανάγνωσης της σελίδας δεν θα πρέπει να προφέρει με έμφαση τον τίτλο ενός βιβλίου, αλλά όταν το ίδιο περιεχόμενο εμφανίζεται σε μια οθόνη τότε ο τίτλος θα είναι με πλάγια γράμματα. Τα περισσότερα στοιχεία οπτικής μορφοποίησης έχουν θεωρηθεί ξεπερασμένα μετά την προδιαγραφή HTML 4.0, και έχουν αντικατασταθεί από τα CSS.

## 12.8 Η σήμανση υπερκειμένου και σύνδεση εγγράφων

Το στοιχείο anchor δημιουργεί έναν υπερσύνδεσμο στο έγγραφο, και η ιδιότητα href ορίζει τον στόχο του συνδέσμου. Για παράδειγμα, η σήμανση HTML `<a href="http://el.wikipedia.org">Βικιπαίδεια</a>` εμφανίζει το κείμενο «Βικιπαίδεια» ως υπερσύνδεσμο. Για την εμφάνιση μιας εικόνας ως συνδέσμου, μπορεί να εισαχθεί ένα στοιχείο «img» ως περιεχόμενο του στοιχείου «a». Όπως και το «br», το «img» είναι άδειο στοιχείο. Έχει ιδιότητες αλλά δεν έχει περιεχόμενο, ούτε ετικέτα τερματισμού: `<a href="http://example.org"></a>`.

## 12.9 Ιδιότητες

Οι περισσότερες ιδιότητες των στοιχείων είναι ζεύγη ονομάτων και τιμών, τα οποία διαχωρίζονται με ένα «=» και γράφονται μέσα στην ετικέτα εκκίνησης ενός στοιχείου, μετά το όνομα του στοιχείου. Η τιμή μπορεί να περικλείεται σε μονά ή διπλά εισαγωγικά, παρότι τιμές που αποτελούνται από συγκεκριμένους χαρακτήρες μπορούν να γράφονται χωρίς εισαγωγικά στην HTML, αλλά όχι στην XHTML. Το να μένουν οι τιμές των ιδιοτήτων χωρίς εισαγωγικά θεωρείται ανασφαλές. Εκτός από τις ιδιότητες που γράφονται ως ζεύγη ονομάτων και τιμών, υπάρχουν και μερικές οι οποίες επηρεάζουν το στοιχείο απλά με την παρουσία τους μέσα στην ετικέτα εκκίνησης, όπως η ιδιότητα ismap του στοιχείου img.

Υπάρχουν και μερικές κοινές ιδιότητες οι οποίες εμφανίζονται σε πολλά στοιχεία:

- Η ιδιότητα id παρέχει ένα αναγνωριστικό για ένα στοιχείο το οποίο είναι μοναδικό σε ολόκληρο το έγγραφο. Χρησιμοποιείται για να ταυτοποιεί το στοιχείο ώστε τα CSS να μπορούν να αλλάξουν τον τρόπο που αυτό εμφανίζεται, καθώς και τα σενάρια μπορούν να αλλάξουν, να μετακινήσουν ή να διαγράψουν τα περιεχόμενα ή την εμφάνισή του. Ακόμα, αν ένα id προστεθεί στο URL μιας σελίδας, παρέχει ένα μοναδικό αναγνωριστικό για ένα τμήμα της σελίδας, για παράδειγμα: `http://el.wikipedia.org/wiki/HTML#Ιδιότητες`

- Η ιδιότητα `class` παρέχει τη δυνατότητα ταξινόμησης παρόμοιων αντικειμένων στην ίδια κλάση. Μπορεί να χρησιμοποιηθεί για να αποδώσει κάποια σημασία στο στοιχείο, ή για σκοπούς εμφάνισης. Για παράδειγμα, ένα έγγραφο HTML μπορεί να χρησιμοποιεί την επισήμανση `class="notation"` σε μερικά στοιχεία για να ξεχωρίσει από το υπόλοιπο κείμενο του εγγράφου. Κατά την εμφάνιση του εγγράφου, αυτά τα στοιχεία μπορεί -για παράδειγμα- να εμφανίζονται όλα μαζί στο τέλος της σελίδας ως υποσημειώσεις, άσχετα με την θέση που εμφανίζονται μέσα στον κώδικα. Επίσης οι ιδιότητες `class` χρησιμοποιούνται σημασιολογικά στα `microformat`. Ένα στοιχείο μπορεί να έχει πολλαπλές κλάσεις, για παράδειγμα το `class="notation important"` βάζει το στοιχείο τόσο στην κλάση «notation» όσο και στην «important».
- Η ιδιότητα `style` εφαρμόζει στυλ εμφάνισης σε συγκεκριμένα στοιχεία. Θεωρείται καλύτερη τακτική να χρησιμοποιούνται οι ιδιότητες `id` ή `class` ώστε να επιλέγεται το στοιχείο μέσα σε ένα CSS, αλλά μερικές φορές μπορεί να είναι πιο απλό να ανατεθούν `style` κατευθείαν στο στοιχείο.
- Η ιδιότητα `title` προσθέτει μια εξήγηση στο στοιχείο στο οποίο εφαρμόζεται. Στους περισσότερους φυλλομετρητές αυτή η ιδιότητα εμφανίζεται ως αναδυόμενο παράθυρο βοήθειας.
- Η ιδιότητα `lang` ταυτοποιεί την φυσική γλώσσα των περιεχομένων του στοιχείου, η οποία μπορεί να είναι διαφορετική από το υπόλοιπο έγγραφο.

## 12.10 Αναφορές οντοτήτων χαρακτήρων

Από την έκδοση 4.0 και μετά, στην HTML ορίζεται ένα σύνολο από 252 αναφορές οντοτήτων χαρακτήρων και ένα σύνολο από 1.114.050 αναφορές οντοτήτων αριθμών. Και τα δύο σύνολα επιτρέπουν τη γραφή μοναδικών χαρακτήρων ως σήμανση, αντί χρησιμοποιώντας τους ίδιους τους χαρακτήρες. Ένας χαρακτήρας και η αντίστοιχη σήμανση γι' αυτόν θεωρούνται ισοδύναμες οντότητες, και εμφανίζονται ίδια.

Αυτή η δυνατότητα διαφυγής χαρακτήρων με αυτόν τον τρόπο επιτρέπει την ερμηνεία χαρακτήρων όπως `<` και `&` (όταν γράφονται ως `&lt;` και `&amp;`, αντίστοιχα) ως κείμενο και όχι ως σήμανση. Για παράδειγμα, ο χαρακτήρας `<` κανονικά σημαίνει την έναρξη μιας ετικέτας, και το `&` σημαίνει την έναρξη μιας αναφοράς οντότητας χαρακτήρα ή αριθμού. Η γραφή τους ως `&amp;` ή `&#x26;` ή `&#38;` επιτρέπει στο `&` να συμπεριλαμβάνεται μέσα στο περιεχόμενο ενός στοιχείου ή μέσα στην τιμή μιας ιδιότητας. Επίσης, ο χαρακτήρας διπλών εισαγωγικών (`"`), ο οποίος χρησιμοποιείται για να περιβάλλει την τιμή μιας ιδιότητας, πρέπει να γράφεται ως `&quot;` ή `&#x22;` ή `&#34;`; όταν εμφανίζεται μέσα στην ίδια την τιμή της ιδιότητας. Ομοίως, ο χαρακτήρας μονών εισαγωγικών (`'`), πρέπει να γράφεται ως `&#x27;` ή `&#39;`; (αλλά όχι ως `&apos;`, με εξαίρεση την γραφή σε XHTML όταν εμφανίζεται μέσα στην ίδια την τιμή της ιδιότητας. Αν ένας δημιουργός εγγράφου HTML παραβλέψει να γράψει αυτά τα

σύμβολα ως οντότητες χαρακτήρων ή αριθμών, τότε μερικοί browser μπορούν να είναι αρκετά επιεικείς και προσπαθούν να χρησιμοποιήσουν το παραπλήσιο περιεχόμενο για να μαντέψουν την πρόθεση του χρήστη. Το αποτέλεσμα δεν παύει να είναι άκυρη σήμανση, η οποία κάνει το έγγραφο λιγότερο προσιτό σε άλλους browser οι οποίοι μπορεί να προσπαθήσουν να αναλύσουν το έγγραφο για άλλους σκοπούς, όπως για παράδειγμα για την καταχώρηση του περιεχομένου του σε μηχανές αναζήτησης.

Η διαφυγή διευκολύνει επίσης τη χρήση χαρακτήρων που είτε είναι δύσκολο να δακτυλογραφηθούν, είτε δεν συμπεριλαμβάνονται καν στην κωδικοποίηση χαρακτήρων του εγγράφου. Για παράδειγμα, ο τονισμένος χαρακτήρας e (δηλαδή το é), ο οποίος τυπικά συναντάται σε μερικές ευρωπαϊκές γλώσσες, μπορεί να γραφεί σε οποιοδήποτε έγγραφο HTML ως η οντότητα &acute; ή ως η αριθμητική αναφορά &#233; ή &#xE9;, χρησιμοποιώντας χαρακτήρες οι οποίοι είναι διαθέσιμοι σε όλα τα πληκτρολόγια και υποστηρίζονται σε όλες τις κωδικοποιήσεις χαρακτήρων. Οι κωδικοποιήσεις Unicode, όπως η UTF-8, είναι συμβατές με όλους τους σύγχρονους browser και επιτρέπουν την πρόσβαση σε σχεδόν όλους τους χαρακτήρες όλων των συστημάτων γραφής του κόσμου.

### 12.11 Τύποι δεδομένων

Στην HTML ορίζονται μερικοί τύποι δεδομένων για το περιεχόμενο των στοιχείων, όπως σενάρια εντολών ή stylesheet, και μια πληθώρα τύπων για τις τιμές των ιδιοτήτων, συμπεριλαμβανομένων των ID, των name, των URI, διαφόρων αριθμών και μονάδων μήκους, γλωσσών, τύπων αρχείων πολυμέσων, χρωμάτων, κωδικοποιήσεων χαρακτήρων, ημερομηνιών κλπ.

### 12.12 Δήλωση τύπου εγγράφου

Τα έγγραφα HTML πρέπει να αρχίζουν με μια Δήλωση τύπου εγγράφου (Document Type Declaration, ανεπίσημα λέγεται και «doctype»). Αυτή η δήλωση βοηθά τους browser να καταλάβουν πώς πρέπει να διαβάσουν το περιεχόμενο του εγγράφου και πώς να το παρουσιάσουν μετά, και ιδιαίτερα όταν χρησιμοποιείται το quirks mode.

Ο αρχικός σκοπός του doctype ήταν να επιτρέψει την ανάλυση και επιβεβαίωση των εγγράφων HTML από εργαλεία SGML τα οποία ήταν βασισμένα στο Document Type Definition (DTD). Το DTD στο οποίο αναφέρεται το DOCTYPE περιέχει μια γραμματική σε γλώσσα μηχανής, η οποία καθορίζει τι επιτρέπεται και τι απαγορεύεται να υπάρχει μέσα στο έγγραφο. Οι browser, από την άλλη, δεν υλοποιούν την HTML ως εφαρμογή της SGML και συνεπώς δεν διαβάζουν το DTD. Στην HTML5 δεν ορίζεται κανένα DTD, λόγω εγγενών τεχνολογικών περιορισμών. Έτσι το doctype <!doctype html> δεν αναφέρεται σε κανένα.

Αυτή η δήλωση αναφέρεται στο DTD για την εκδοχή «strict» της HTML 4.01. Τα προγράμματα επιβεβαίωσης κώδικα τα οποία βασίζονται σε SGML, διαβάζουν το DTD με σκοπό να ερμηνεύσουν κατάλληλα το έγγραφο και να επαληθεύσουν την

ορθότητά του. Στους σύγχρονους browser, ένα έγκυρο doctype ενεργοποιεί τη λειτουργία κατά τα πρότυπα, αντί για το quirks mode.

Επιπλέον, στην HTML 4.01 παρέχονται και τα DTD Transitional και Frameset.

### 12.13 Σημασιολογική

Η Σημασιολογική HTML είναι ένας τρόπος γραφής της HTML με έμφαση στο νόημα της σημασμένης πληροφορίας παρά στην εμφάνισή της. Η HTML συμπεριλαμβάνει σημασιολογικά στοιχεία από την σύλληψή της, συμπεριλαμβάνει όμως και στοιχεία αποκλειστικά εμφανισιακά, όπως τις ετικέτες <font>, <i> και <center>. Υπάρχουν επίσης και τα σημασιολογικά ουδέτερα span και div. Από τα τέλη της δεκαετίας του 1990, όταν τα CSS άρχισαν να δουλεύουν στους περισσότερους browser, οι σχεδιαστές ιστοσελίδων ενθαρρύνονταν να αποφεύγουν τη χρήση των εμφανισιακών ετικετών της HTML και να προτιμούν να διαχωρίζουν το περιεχόμενο από την παρουσίαση.

Σε μια συζήτηση για τον Σημασιολογικό ιστό, ο Τιμ Μπέρνερς Λι και άλλοι έδωσαν παραδείγματα τρόπων με τους οποίους έξυπνα λογισμικά πελάτη ιστού θα μπορούν μια μέρα να περιηγούνται στον Ιστό και να βρίσκουν, να φιλτράρουν και να συσχετίζουν τα προηγούμενως άσχετα και αδημοσίευτα δεδομένα, προς όφελος των ανθρώπων χρηστών. Δέκα χρόνια μετά, δεν είναι ακόμα κοινή η χρήση τέτοιων προγραμμάτων, αλλά μερικές από τις ιδέες του Web 2.0, των mashups και των ιστοσελίδων σύγκρισης τιμών προϊόντων μπορεί να πλησιάζουν στο σκεπτικό εκείνο. Η βασική διαφορά μεταξύ αυτών των εφαρμογών και των έξυπνων λογισμικών του Μπέρνερς Λι, είναι ότι η τρέχουσες μέθοδοι συγκέντρωσης πληροφοριών συνήθως σχεδιάζεται από προγραμματιστές ιστού, οι οποίοι ήδη ξέρουν τις τοποθεσίες ιστού και τα API των δεδομένων που θέλουν να συνενώσουν, να συγκρίνουν και να συνδυάσουν.

Ένας σημαντικός τύπος λογισμικοί που περιηγείται αυτόματα το διαδίκτυο και διαβάζει ιστοσελίδες, χωρίς προηγούμενη γνώση του περιεχομένου τους, είναι και τα προγράμματα των μηχανών αναζήτησης. Αυτά τα λογισμικά εξαρτούνται από την σημασιολογική σαφήνεια των ιστοσελίδων που βρίσκουν, καθώς χρησιμοποιούν διάφορες τεχνικές και αλγόριθμους για να διαβάζουν και να ταξινομούν εκατομμύρια σελίδων καθημερινά, και να παρέχουν στους επισκέπτες τους τη δυνατότητα αναζήτησης, χωρίς την οποία ο Παγκόσμιος Ιστός θα είχε μόνο ένα μικρό κλάσμα της χρησιμότητάς του.

Οι σημασιολογικές δομές που ήδη υπάρχουν στην HTML, είναι σημαντικό να εφαρμόζονται καθολικά, για να βοηθούν το περιεχόμενο του δημοσιευμένου κειμένου να γίνεται καλύτερα αντιληπτό. Με αυτό τον τρόπο, τα λογισμικά των μηχανών αναζήτησης, αλλά και άλλα λογισμικά που φτιάχνουν mashup ή άλλα υβρίδια από το περιεχόμενο των ιστότοπων θα μπορούν καλύτερα να εκτιμούν τη σημασία του κειμένου που βρίσκουν στα έγγραφα HTML. Προς το σκοπό αυτό, οι ετικέτες της HTML που εξυπηρετούσαν μόνο εμφανισιακό σκοπό θεωρούνται ξεπερασμένες στην XHTML, και απαγορεύονται στην HTML5.

Τέλος, η σημασιολογικά γραμμένη HTML βελτιώνει την προσιτότητα των εγγράφων στον ιστό. Για παράδειγμα, στους browser για χρήστες με προβλήματα

όρασης ή ακοής, είναι πιο εύκολη η σωστή απόδοση της δομής του εγγράφου με τρόπο άλλο από τον οπτικό, όταν αυτό είναι σωστά σημασιολογικά σημασμένο.

### 12.14 Μεταφορά της HTML

Τα αρχεία HTML, όπως συμβαίνει και με τους υπόλοιπους τύπους αρχείων του υπολογιστή, μπορούν να μεταφερθούν με πολλούς τρόπους. Ωστόσο, λόγω της λειτουργίας της HTML, οι δύο πιο συνηθισμένοι τρόποι είναι μέσω HTTP από έναν εξυπηρετητή, ή μέσω E-mail.

### 12.15 HTTP

Ο Παγκόσμιος Ιστός αποτελείται κυρίως από αρχεία HTML τα οποία μεταφέρονται από εξυπηρετητές προς browsers χρησιμοποιώντας το πρωτόκολλο HTTP. Ωστόσο, μια που το ίδιο πρωτόκολλο μπορεί επιπλέον να μεταφέρει εικόνες, ήχο και άλλο περιεχόμενο, χρειάζεται ένας τρόπος αναγνώρισης του τύπου του περιεχομένου που μεταφέρεται. Έτσι, μαζί με το ίδιο το αρχείο, μεταφέρονται και μερικές επιπλέον πληροφορίες, ή μεταδιδόμενα, μεταξύ αυτών και ο τύπος MIME (για παράδειγμα text/html ή application/xhtml+xml) καθώς και η κωδικοποίηση χαρακτήρων που χρησιμοποιείται.

Στους σύγχρονους browser, ο τύπος MIME που συνοδεύει το αρχείο HTML μπορεί να επηρεάζει τον τρόπο που αυτό εμφανίζεται. Για παράδειγμα, ένα αρχείο το οποίο συνοδεύεται από έναν τύπο XHTML MIME, αναμένεται να είναι γραμμένο σε γλώσσα σωστά διατυπωμένη, κατά τα πρότυπα της XML. Αν δεν είναι, τότε τα σφάλματα στον κώδικα μπορεί να αποτρέψουν την ορθή απεικόνιση του αρχείου. Επειδή η XHTML 1.0 και η XHTML 1.1 είναι πάντα συμβατές με την XML, το W3C υποστηρίζει ότι δεν παίζει ρόλο ποιος τύπος MIME θα χρησιμοποιηθεί.

### 12.16 HTML e-mail

Αρκετά προγράμματα ηλεκτρονικής αλληλογραφίας ενσωματώνουν δυνατότητες της HTML, ώστε να επιτρέπουν στους χρήστες να χρησιμοποιούν την σήμανση της HTML στα μηνύματά τους, για να στέλνουν κείμενο με χρώμα, με μορφοποίηση, ή και με ενσωματωμένες εικόνες και διαγράμματα. Ωστόσο, οι μέθοδοι αυτές προκαλούν μερικά προβλήματα, τα οποία προκαλούνται αφ' ενός από την έλλειψη κάποιου προτύπου για την περίληψη HTML σε E-mail (με αποτέλεσμα ο κάθε προγραμματιστής να ενεργεί κατά βούληση), και αφ' ετέρου από τη δυνατότητα εκμετάλλευσης αυτών των δυνατοτήτων από κακόβουλους χρήστες.



### 12.17 Ονόματα αρχείων

Ο πιο κοινός τύπος αρχείου για έγγραφα HTML είναι .html, όμως έχει επιβιώσει και η συντόμευση .htm, από μερικά παλαιότερα λειτουργικά συστήματα που δεν αναγνώριζαν επεκτάσεις αρχείων με περισσότερα από τρία γράμματα. Επιπλέον, στα Microsoft Windows χρησιμοποιείται ο τύπος .hta (από το HTML Application) ο οποίος δείχνει ότι το αρχείο συμπεριλαμβάνει HTML μαζί με κάποια δυναμικά στοιχεία που το κάνουν να εκτελείται ως εφαρμογή.

### 12.18 Δημιουργία με προγράμματα WYSIWYG

Ένα θέμα αντιπαράθεσης στις κοινότητες των δημιουργών περιεχομένου για τον Ιστό είναι η χρήση προγραμμάτων WYSIWYG (What You See Is What You Get), δηλαδή προγραμμάτων στα οποία ο χρήστης σχεδιάζει οπτικά τη σελίδα όπως θα φαίνεται μετά στον browser του επισκέπτη. Αυτό κάνει τη γνώση της HTML προαιρετική για την κατασκευή μιας ιστοσελίδας. Αυτό το μοντέλο δημιουργίας με προγράμματα WYSIWYG έχει γνωρίσει έντονες κριτικές, κυρίως λόγω της κακής ποιότητας του κώδικα που δημιουργείται αυτόματα.

Παρόλα αυτά, οι επεξεργαστές ιστοσελίδων WYSIWYG είναι δημοφιλείς λόγω της ευκολίας που παρέχουν.

### 12.19 HTML 5

Η HTML5 είναι μια υπό ανάπτυξη γλώσσα σήμανσης για τον Παγκόσμιο Ιστό που όταν ετοιμαστεί θα είναι η επόμενη μεγάλη έκδοση της HTML (Γλώσσα Υπερκειμένου, HyperText Markup Language). Η ομάδα Web Hypertext Application Technology Working Group (WHATWG) άρχισε δουλειά σε αυτή την έκδοση τον Ιούνιο του 2004 με το όνομα Web Applications 1.0. Το Φεβρουάριο του 2010 το πρότυπο ήταν ακόμη σε κατάσταση "Last Call" στο WHATWG.

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

Οι ιδέες πίσω από την HTML5 εμφανίστηκαν αρχικά το 2004 από την ομάδα WHATWG. Η HTML5 εμπεριέχει το πρότυπο Web Forms 2.0 που είναι επίσης της WHATWG.

Το πρότυπο HTML5 υιοθετήθηκε ως αρχικό βήμα για τις εργασίες της νέας ομάδας εργασίας HTML του W3C το 2007. Αυτή η ομάδα εργασίας δημοσίευσε το Πρώτο Δημόσιο Working Draft του προτύπου στις 22 Ιανουαρίου 2008. Το πρότυπο είναι ακόμη υπό ανάπτυξη, και αναμένεται να παραμείνει έτσι για πολλά χρόνια, παρόλο που μέρη της HTML5 θα τελειώσουν και θα υποστηριχτούν από περιηγητές πριν το όλο πρότυπο φτάσει στη τελική κατάσταση Recommendation.

Οι συντάκτες της HTML5 είναι ο Ίαν Χίκσον της εταιρίας Google και ο Ντέιβ Χιάτ της εταιρίας Apple.

## 12.20 Εισαγωγή στην HTML και τα CSS

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. Internet Explorer), για να δημιουργηθούν και να εμφανιστούν οι ιστοσελίδες.

## 12.21 Βασική δομή μιας ιστοσελίδας γραμμένης με HTML

Ο παρακάτω κώδικας αποτελεί τη βασική δομή μιας HTML, σελίδας. Οι 6 γραμμές κώδικα που υπάρχουν παρακάτω απαρτίζουν οποιαδήποτε σελίδα HTML. Κάθε γραμμή αυτού του κώδικα αποτελεί και ένα tag.

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 ενώ το τέλος με . 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: Ο τίτλος της Η γλώσσα που υποστηρίζει η σελίδα (π.χ. ελληνικά) 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 Σητεια το tag head κλείνει.
- <body> TAG με ονομασία BODY. Από αυτό το Σητεια 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, που θα αναφερθούν παρακάτω, μετά και κάτω από αυτό το tag.
- </body> TAG με ονομασία BODY. Σε αυτό το Σητεια 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.

## 12.22 Επεκτείνοντας τη βασική δομή μιας ιστοσελίδας HTML

Ο παρακάτω κώδικας αφορά το βασικό περιεχόμενο του tag HEAD.

```
<head>
```

```
<title>Τίτλος Σελίδας</title>
```

Το tag `<title>` μας δίνει τη δυνατότητα να δώσουμε ένα τίτλο στην ιστοσελίδα μας. Ανάμεσα στο άνοιγμα και στο κλείσιμο του tag, πληκτρολογούμε αυτό που θέλουμε να είναι ο τίτλος της ιστοσελίδας μας.

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7">
```

Το tag `<meta>` αποτελεί εξαίρεση σε ότι έχει αναφερθεί σχετικά με τα tags της HTML. Το TAG `<meta>` δεν κλείνει. Δεν υπάρχει δηλαδή το `</meta>`. Το tag αυτό χρησιμοποιείται με κάποιες παραμέτρους (π.χ. `<meta name="description" content="Η σελίδα αυτή είναι μία σελίδα που...">`) που συνήθως διαβάζουν οι μηχανές αναζήτησης. Η χρήση του που αφορά καθαρά τον κώδικα HTML που αναφέρεται παραπάνω, σχετίζεται με τη δήλωση της γλώσσας που θα χρησιμοποιήσει η ιστοσελίδα μας.

Οι παράμετροι του tag `<meta>` που αφορούν στη χρήση της γλώσσας φαίνονται παραπάνω. Η δήλωση της γλώσσας γίνεται στο κομμάτι που αναφέρεται ως `charset`, το οποίο παίρνει ως τιμές ονόματα κωδικοσελίδων που αναπαριστούν τη γλώσσα για τους υπολογιστές. Οι τιμές του πεδίου `charset` που μας αφορούν και χρησιμοποιούνται επί το πλείστον είναι `iso-8859-1` για τις λατινογενείς γλώσσες και `iso-8859-7`, για την ελληνική γλώσσα.

```
</head>
```

Ο παρακάτω κώδικας αφορά τις βασικές παραμέτρους του tag BODY. Σημειώνεται πως καμία από τις παραμέτρους που αναφέρονται παρακάτω, δεν είναι υποχρεωτικό να υπάρχουν σε κάποια σελίδα.

```
<body bgcolor="#009900" background="back.jpg" text="#003399"
link="#993300" vlink="#009999" leftmargin="0" topmargin="0"
marginwidth="0" marginheight="0">
```

Το tag `body`, παίρνει κάποιες παραμέτρους. Παρακάτω αναφέρεται η χρήση της καθεμιάς.

Παράμετρος `bgcolor`. Η παράμετρος αυτή παίρνει ως τιμή το χρώμα που θα είναι το φόντο της σελίδας μας. Π.χ. `bgcolor="#009900"` θα φέρει ως φόντο της σελίδας μας ένα πράσινο σκούρο χρώμα. Το χρώμα δηλώνεται σε hexadecimal code όπως φαίνεται και παραπάνω και αυτό για να διατηρείται ομοιομορφία όταν η σελίδα πρόκειται να εμφανιστεί σε διαφορετικούς υπολογιστές με διαφορετικούς

browsers κ.λ.π. Εναλλακτικά μπορούν να χρησιμοποιηθούν ονομασίες χρωμάτων όπως “red” , “green” κ.λ.π.

Παράμετρος **background** . Η παράμετρος αυτή παίρνει ως τιμή μια εικόνα που θα χρησιμοποιηθεί ως φόντο της σελίδας μας. Στο παραπάνω παράδειγμα ως φόντο της σελίδας μας θα εμφανιστεί η εικόνα `back.jpg` . Η χρήση των παραμέτρων `bgcolor` και `background` αλληλοαναιρείται. Αν δηλαδή δηλωθούν και οι δύο παράμετροι, τότε θα υλοποιηθεί μόνο η παράμετρος `background`. Σημειώνεται ότι η τιμή της παραμέτρου `background`, ορίζει και την διαδρομή στο σκληρό δίσκο στον οποίο βρίσκεται η εικόνα. Στο παραπάνω παράδειγμα εννοείται πως η εικόνα βρίσκεται στον ίδιο φάκελο με το αρχείο `html` που δημιουργούμε. Αν η εικόνα βρισκόταν σε κάποιο φάκελο τότε θα έπρεπε να οριστεί και το όνομα του φακέλου στον οποίο βρίσκεται η εικόνα (π.χ. αν η εικόνα βρισκόταν στο φάκελο `images` τότε θα έπρεπε να γράψουμε `background="images/back.jpg"`

Γενικά όλα τα αρχεία ιστοσελίδων βρίσκονται σε κάποιο κύριο φάκελο π.χ. `web` . Ο φάκελος αυτός μπορεί να περιέχει υποφακέλους. Όταν καλούμε αρχεία (εικόνες, αρχεία πολυμέσων , ...) μέσα από κάποια παράμετρο ενός `tag`, τότε θα πρέπει να ορίζονται οι απόλυτες διαδρομές των αρχείων αυτών. Υπάρχουν οι εξής 3 περιπτώσεις.

- Σε ένα αρχείο `html` πρέπει να γίνει αναφορά σε μία εικόνα `img.gif` που βρίσκεται σε ένα φάκελο `imgs`. Το αρχείο `html` βρίσκεται μέσα στο φάκελο `imgs`. Η αναφορά θα γίνει ως `“img.gif”`.
- Σε ένα αρχείο `html` πρέπει να γίνει αναφορά σε μία εικόνα `img.gif` που βρίσκεται σε ένα φάκελο `imgs`. Το αρχείο `html` βρίσκεται έξω από το φάκελο `imgs`. Η αναφορά θα γίνει ως `“imgs/img.gif”`.
- Σε ένα αρχείο `html` που βρίσκεται σε ένα φάκελο `imgs`, πρέπει να γίνει αναφορά σε μία εικόνα `img.gif`. Το αρχείο `img.gif` βρίσκεται εκτός του φακέλου `imgs`. Η αναφορά θα γίνει ως `“/img.gif”`.

Παράμετρος **text**. Η παράμετρος αυτή παίρνει ως τιμή το χρώμα με το οποίο θα εμφανίζεται το κείμενο της σελίδας μας.

Παράμετρος **link**. Η παράμετρος αυτή παίρνει ως τιμή το χρώμα με το οποίο θα εμφανίζονται τα `links` της σελίδας μας.

Παράμετρος **vlink**. Η παράμετρος αυτή παίρνει ως τιμή το χρώμα με το οποίο θα εμφανίζονται τα `links` της σελίδας μας τα οποία κάποιος χρήστης τα έχει επισκεφτεί (τα έχει επιλέξει).

Παράμετρος **leftmargin & marginwidth**. Οι παράμετροι αυτοί παίρνουν ως τιμές των αριθμό των `pixels` τα οποία θα απέχει η αρχή της σελίδας μας από το αριστερό περιθώριο του `browser`. Η πρώτη παράμετρος αφορά την υλοποίηση για τον `Internet Explorer`, η δεύτερη για το `Netscape Navigator`.

Παράμετρος **topmargin & marginheight**. Οι παράμετροι αυτοί παίρνουν ως τιμές των αριθμό των `pixels` τα οποία θα απέχει η αρχή της σελίδας μας από το πάνω περιθώριο του `browser`. Η πρώτη παράμετρος αφορά την υλοποίηση για τον `Internet Explorer`, η δεύτερη για το `Netscape Navigator`.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 13

### CSS

#### 13.1 Τι είναι η CSS

Η CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) ή ( αλληλουχία φύλλων στυλ ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

#### 13.2 Η αλληλουχία εφαρμογής των φύλλων στυλ

Για ένα έγγραφο πχ xhtml θα υπάρχουν παραπάνω από ένα φύλλα στυλ τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου. Το Φύλλο στυλ που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από :

- το συγγραφέα μιας ιστοσελίδας
- το χρήστη του πλοηγού
- τον ίδιο τον πλοηγό, αν έχει το δικό του προκαθορισμένο φύλλο στυλ .

Συνεπώς για ένα xhtml στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες. Το πρότυπο css για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει μια αλληλουχία-σειρά στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί πχ η δήλωση που είναι πρώτη στη σειρά.

Ο αλγόριθμος δημιουργίας αυτής της σειράς-αλληλουχίας είναι ο ακόλουθος:

1. Βρες όλες τις δηλώσεις που εφαρμόζονται στο στοιχείο που μας ενδιαφέρει. Οι δηλώσεις εφαρμόζονται στο στοιχείο αν ο επιλογέας του το επιλέξει (ταιριάζει με αυτό).
2. Ταξινόμησε με βάση τη σημασία (κανονική ή σημαντική) και προέλευση ( συγγραφέας , χρήστη ή πλοηγός χρήστη). Με αύξουσα σειρά προτεραιότητας:
  1. Δηλώσεις πλοηγού χρήστη
  2. Κανονικές δηλώσεις χρήστη

3. Κανονικές δηλώσεις συγγραφέα
  4. Σημαντικές δηλώσεις συγγραφέα
  5. Σημαντικές δηλώσεις χρήστη
3. Ταξινόμησε τις δηλώσεις ίδιας σημασίας και προέλευσης με κριτήριο την εξειδίκευση του επιλογέα: οι πιο εξειδικευμένοι επιλογείς υπερισχύουν των πιο γενικών. Τα ψευδο-στοιχεία και οι ψευδο-κλάσεις λογαριάζονται σαν κανονικά στοιχεία και κλάσεις αντίστοιχα.
  4. Τέλος ταξινόμησε ανάλογα με τη σειρά καθορισμού: αν δύο δηλώσεις έχουν το ίδιο βάρος , προέλευση και εξειδίκευση , αυτή που προσδιορίστηκε τελευταία επικρατεί. Οι δηλώσεις σε εισαγόμενα φύλλα στυλ θεωρούνται ότι δηλώνονται πριν από τις δηλώσεις στο ίδιο το φύλλο στυλ .

Αφού λοιπόν προκύψει μια σειρά-αλληλουχία κανόνων εμφάνισης που αφορούν το ίδιο στοιχείο θα επιλεγεί προς εφαρμογή (για την αποφυγή συγκρούσεων ) η δήλωση που θα είναι τελευταία στην σειρά που αναλύθηκε πιο πάνω.

### 13.3 Εισαγωγή στη 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.

## ΚΕΦΑΛΑΙΟ 14

# JAVASCRIPT

### 14.1 Τι είναι Javascript

Javascript είναι μία γλώσσα scripting (ουσιαστικά προγραμματισμού) με την οποία κάνουμε διάφορα πράγματα σε μία ιστοσελίδα.

Αυτή η γλώσσα είναι client side που σημαίνει ότι εκτελείται και παράγεται το αποτέλεσμα στον browser του client - του επισκέπτη δηλαδή.

Δεν έχει σχέση με την Java.

Δεν είναι κατάλληλη στο να φτιάξουμε εξ' ολοκλήρου μία δυναμική ιστοσελίδα με αυτήν, απλά λειτουργεί "συμπληρωματικά".

Επιτρέπει δηλαδή στον επισκέπτη να κάνει κάποια πράγματα δυναμικά σε μία ιστοσελίδα, σε μικρό βαθμό και άμεσα, χωρίς να επικοινωνήσει το σύστημα με τον server για να εκτελέσει το αίτημά του. Πολλές φορές συνεργάζεται με την HTML. Με απλά λόγια, πχ μπορεί να φτιάξουμε ένα script σε javascript το οποίο να κάνει το εξής:

Έστω σε μία ιστοσελίδα έχουμε μία φόρμα. Θέλουμε όμως να μην επιτρέπουμε στο πεδίο με το email να γράψει πάνω από 50 χαρακτήρες.

Αντίστοιχα θέλουμε πριν αποσταλεί η φόρμα, να ελέγξει αν το email που πληκτρολόγησε ο χρήστης είναι σωστό, δηλαδή της μορφής πχ kati@kati.com.

Έτσι μόλις πατήσουμε το κουμπί αποστολή, η javascript εφαρμογή θα ελέγξει αν τα πληκτρολόγησε ο χρήστης σωστά και είτε θα τον αφήσει να συνεχίσει, είτε όχι, βγάζοντάς του πχ ένα μήνυμα: το email που πληκτρολογήσατε δεν είναι σωστό.

### 14.2 Λίγα λόγια για τη Javascript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.



Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript.

### 14.3 Ιστορία της Javascript

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξη της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995. LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3.

Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java. JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (και μεταξύ άλλων λόγων). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό

ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side).

#### 14.4 Μοντέλο εκτέλεσης

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες.

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

Παρά την ευρεία χρήση της Javascript για συγγραφή προγραμμάτων σε περιβάλλον φυλλομετρητή, από την αρχή χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή, από την ίδια τη Netscape στο προϊόν LiveWire, με μικρή επιτυχία. Η χρήση της Javascript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

#### 14.5 Javascript και Java

Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης "Java" στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστραφείς γλώσσες. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι "Javascript" και όχι 'Java script' σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 15

### ANGULARJS



#### 15.1 Τι είναι η Angular

Μια δυναμική νέα “γλώσσα προγραμματισμού” για την javascript είναι η angular. Η angular είναι μια γλώσσα που δρα στον browser του υπολογιστή δηλαδή μια client-side γλώσσα.

Για την ακρίβεια και για να είμαστε πιο σωστοί δεν είναι γλώσσα αλλά ένα mvc( η ακόμα καλύτερα ένα mwn(model view whatever) ) Framework το οποίο σου δίνει απεριόριστες δυνατότητες.

Η angular είναι ένα framework που βασίζεται στην javascript και δημιουργεί ένα “αντικείμενο” σε όλη την σελίδα που θα υποδείξουμε. Αυτό το αντικείμενο που δημιουργείτε μπορούμε να το αλλάξουμε μέσα από τους controllers να το μορφοποιήσουμε με τα services προσθέτοντας data και γενικότερα να το κάνουμε ότι θέλουμε.

Η κυρίως μεταβλητή της angular είναι η \$scope και εκεί που γίνεται όλη η “μαγεία” είναι ο controller.

Το πιο σημαντικό προτέρημα της angular είναι η προσαρμοστικότητα της και η ευκολία εκμάθησης της.

Το testing με την angular είναι εύκολο και πιο συγκεκριμένα με το yeoman που είναι το κατάλληλο εργαλείο. Αν και δεν είναι σωστό αυτό που θα πούμε το yeoman είναι κάτι σαν την sass για την css μπορείς να την χρησιμοποιήσεις από κατασκευή απλών ιστοσελίδων μέχρι σε εφαρμογή για εξυχρονισμό παλαιών προγραμμάτων. Για παράδειγμα μπορείς να χρησιμοποιήσεις την angular για να εξυχρονίσεις μια εφαρμογή που γράφτηκε απλά με php η asp πολύ απλά εάν δημιουργήσεις ένα json object και μετά να το μεταλλάξεις με την angular.

Η εξήγηση για την μεγάλη χρήση της AngularJS είναι απλή. Η AngularJS χρησιμοποιεί JavaScript, απλή απλούστατη Javascript. Δεν χρησιμοποιεί ούτε ES6/ES2015 ούτε TypeScript. Δεν βασίζεται σε πολύπλοκα εργαλεία ανάπτυξης και ανάλυσης. Δεν υπάρχει κανένα μυστήριο κρυμμένο από πίσω της. Είναι πολύ καλά δομημένη και δοκιμασμένη στο να λύνει προβλήματα. Προσφέρει σταθερότητα, εξαιρετικό οικοσύστημα ερωτήσεων/απαντήσεων, λύσεων, modules,

tutorials και stack overflow answers.

Angular ονομάζουμε την έκδοση 2 και τις επόμενες που είναι οι αντικαταστάτες της AngularJS (Angular 1). Η Angular λοιπόν, προσφέρει ένα εξαιρετικά μοντέρνο περιβάλλον ανάπτυξης με εξαιρετικές επιδόσεις, καλύτερη αρχιτεκτονική και πιο σύγχρονες λύσεις. Αυτό φαίνεται από την αρχή με το Command Line Interface (CLI) που μας βοηθάει στις διαδικασίες ανάπτυξης νέων projects μέσω boilerplates αλλά και κατά τη διάρκεια της εξέλιξης ενός project. Είναι ένα framework που τα περιλαμβάνει όλα.

Είναι επίσης το framework που πραγματικά φέρνει την TypeScript σε εξέχουσα θέση. Η TypeScript έχει ένα ανερχόμενο πρότυπο που επιτρέπει στους developers να δουλεύουν σε μία κανονική strict γλώσσα, η οποία είναι πολύ πιο εύκολο να αναλυθεί και να απασφαλιστοποιηθεί.

Επιπλέον, η Angular είναι το πρώτο framework που πραγματικά κάνει implement τα Observables, τα οποία είναι ένα βήμα πιο πάνω από τα Promises. Είναι εξαιρετικά δυνατά για “streams” και live updated σε πραγματικό χρόνο, δεδομένα.

Για μεγάλες εφαρμογές που θα χρειαστούν αρκετή λειτουργικότητα όπως routers, state management, συνδέσεις με API's κ.λ.π, είναι καλύτερο να έχεις ένα περιεκτικό framework παρά να συνδυάζεις αμφίβολης ποιότητας modules από διαφορετικούς προγραμματιστές..

## ΚΕΦΑΛΑΙΟ 16

### PHALCON FRAMEWORK



#### 16.1 Τι είναι το Phalcon Framework

Ένα πλαίσιο PHP πλήρης στοίβας, που παρέχεται σαν επέκταση C. Η καινοτόμα αρχιτεκτονική του Phalcon, το κάνει το γρηγορότερο πλαίσιο PHP που κατασκευάστηκε ποτέ!

#### 16.2 Πλεονεκτήματα του framework

- Οι επεκτάσεις Zephir/C φορτώνονται με το PHP μια φορά μόνο, κατά τη διαδικασία εκκίνησης τού δαίμονα διακομιστή web.
- Κλάσεις και οι λειτουργίες που παρέχονται από την επέκταση είναι έτοιμες προς χρήση για οποιαδήποτε εφαρμογή.
- Ο κώδικας έχει μεταγλωττιστεί και δεν ερμηνεύεται, επειδή είναι ήδη μεταγλωττισμένο σε μια συγκεκριμένη πλατφόρμα και επεξεργαστή.
- Χάρη στη χαμηλού επιπέδου αρχιτεκτονική και βελτιστοποιήσεις το Phalcon παρέχει τη χαμηλότερη επιβάρυνση για εφαρμογές που βασίζονται στο MVC.

#### 16.3 Βασικά Χαρακτηριστικά

- Χαμηλή επιβάρυνση - Χαμηλή κατανάλωση μνήμης και CPU σε σύγκριση με παραδοσιακές στοίβες.
- MVC & HMVC - Λειτουργικές μονάδες, συστατικά, μοντέλα, απόψεις και ελεγκτές.
- Dependency Injection - Dependency Injection και Τοποθεσία υπηρεσιών και είναι το ίδιο ένα δοχείο για αυτούς.
- Rest - Στην περίπτωση αυτή, μπορείτε να χρησιμοποιήσετε είτε μια εφαρμογή micro ή πλήρης στοίβας για να πετύχετε τον στόχο σας. Επιπλέον, ένα ισχυρό σετ βοηθών HTTP.
- Autoloader - Παρέχει μηχανισμό αυτόματης φόρτωσης τάξεων PHP ακλουθώντας το PSR-4.
- Δρομολογητής - Το Phalcon\MvcRouter παρέχει προηγμένες δυνατότητες δρομολόγησης.

## 16.4 Data & Storage

- ORM - Αντικείμενο Σχεσιακής Χαρτογράφησης. Ένα ισχυρό ORM παρέχεται από το Phalcon που σας επιτρέπει να χειριστείτε αρχεία βάσεων δεδομένων ως κλάσεις και αντικείμενα. MySQL, PostgreSQL και SQLite υποστηρίζονται αυτόματα από το Phalcon.
- PHQL - Η ισχυρή και ασφαλή Phalcon Query Language, PHQL. Το PHQL είναι μια object-oriented διάλεκτος SQL υψηλού επιπέδου, που σας επιτρέπει να γράψετε ερωτήματα χρησιμοποιώντας μια τυποποιημένη γλώσσα SQL. Το PHQL υλοποιείται ως ένα πρόγραμμα ανάλυσης (γραμμένο σε C) που μεταφράζεται στη σύνταξη του στόχου RDBMS. Για να επιτύχετε την υψηλότερη δυνατή απόδοση, το Phalcon παρέχει ένα πρόγραμμα ανάλυσης που χρησιμοποιεί την ίδια τεχνολογία όπως το SQLite. Η τεχνολογία αυτή παρέχει ένα μικρό πρόγραμμα ανάλυσης στη μνήμη που ξοδεύει πολύ λίγη μνήμη και προσφέρει ασφάλεια για νήματα επεξεργασίας.
- ODM για Mongo - Αντικείμενο Αντιστοίχισης Εγγράφων για MongoDB. Στην ικανότητά καταγραφής πινάκων σε σχεσιακές βάσεις δεδομένων, το Phalcon μπορεί να αντιστοιχίσετε έγγραφα σε μια βάση δεδομένων του MongoDB. Το ODM προσφέρει λειτουργίες CRUD, εκδηλώσεις, επικυρώσεις μεταξύ άλλων υπηρεσιών.
- Cache - Βελτιώστε την απόδοση της εφαρμογής σας με τις πολλές υπηρεσίες αποθήκευσης προσωρινών δεδομένων που παρέχει το Phalcon. Το στοιχείο cache επιτρέπει την ταχύτερη πρόσβαση σε συχνά χρησιμοποιούμενα ή ήδη επεξεργασμένα δεδομένα. Υποστηρίζει πολλά backends όπως Redis, Memcached, Mongo, Files, Arc και παραπάνω.

## 16.5 Απόψεις και frontend

- Κινητήρας Πρότυπων - Απόψεις είναι τα στοιχεία που αντιπροσωπεύουν το περιβάλλον εργασίας του χρήστη της εφαρμογής σας. Απόψεις είναι συχνά αρχεία HTML με ενσωματωμένο PHP κώδικα που εκτελούν εργασίες οι οποίες σχετίζονται αποκλειστικά με την παρουσίαση των δεδομένων. Οι απόψεις χειρίζονται την εργασία της παροχής των δεδομένων για το πρόγραμμα περιήγησης ή άλλο εργαλείο που χρησιμοποιείται για να κάνουν αιτήσεις από την εφαρμογή σας.
- Μηχανή Προτύπων (Volt) - Ένα πρότυπο κινητήρα απόψεων, εμπνευσμένο από τη Jinja αλλά χτισμένο σε C για το PHP. Το Volt είναι μια εξαιρετικά γρήγορη και φιλική για τους σχεδιαστές templating γλώσσα γραμμένο σε Zephir/C για PHP. Σας παρέχει ένα σύνολο των βοηθών να γράψετε απόψεις εύκολα. Το Volt είναι ενσωματωμένη με άλλα στοιχεία του Phalcon, και μπορείτε να το χρησιμοποιήσετε ως ένα αυτόνομο στοιχείο στις εφαρμογές σας.
- Δημιουργία Φορμών - Δημιουργήστε εύκολα φόρμες HTML. Κάθε στοιχείο με τη φόρμα μπορεί να αποδοθεί όπως απαιτείται από τον προγραμματιστή. Εσωτερικά, το Phalcon\Tag χρησιμοποιείται για να

- παράγει το σωστό HTML για κάθε στοιχείο και μπορείτε να περάσετε τα επιπλέον χαρακτηριστικά HTML ως τη δεύτερη παράμετρο του render():
- Κρυπτογράφηση - Κρυπτογράφηση/αποκρυπτογράφηση σημαντικών δεδομένων για να τα κρατήσετε ασφαλή από μη εξουσιοδοτημένα τρίτα μέρη. Το Phalcon παρέχει κρυπτογράφηση μέσω του στοιχείου Phalcon\Crypt. Αυτή η τάξη προσφέρει απλά object-oriented περιτυλίγματα για την openssl βιβλιοθήκη κρυπτογράφησης του PHP.
  - Υπηρεσίες Ουράς - διαδικασίες στο παρασκήνιο. Δραστηριότητες όπως η επεξεργασία βίντεο, αλλαγή μεγέθους των εικόνων ή αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου δεν είναι κατάλληλες για να εκτελεστούν σε απευθείας σύνδεση ή σε πραγματικό χρόνο, επειδή επιβραδύνουν το χρόνο φόρτωσης των σελίδων και συνεπώς να επηρεάσουν σοβαρά την εμπειρία του χρήστη. Η καλύτερη λύση εδώ είναι η εφαρμογή θέσεις εργασίας στο φόντο. Η ιστοσελίδα τοποθετεί εργασίες σε μια ουρά και που θα υποβληθούν για επεξεργασία ξεχωριστά από την κύρια ιστοσελίδα.



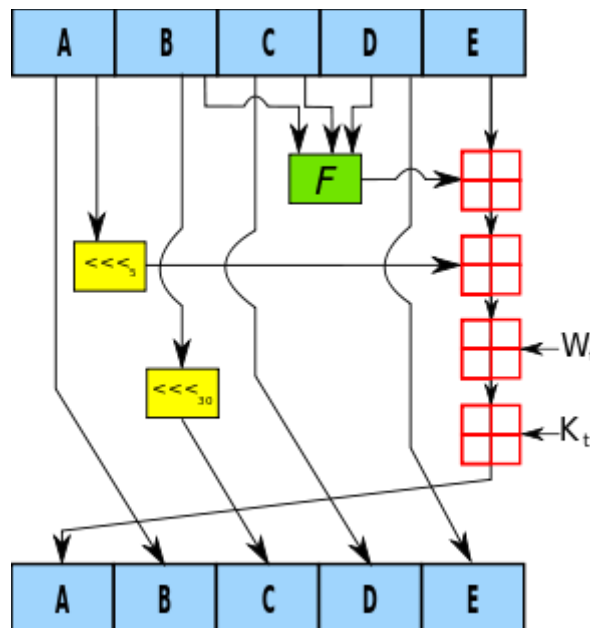
Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΚΕΦΑΛΑΙΟ 17

### ΚΡΥΠΤΟΓΡΑΦΗΣΗ ΚΑΙ ΑΣΦΑΛΕΙΑ ΙΣΤΟΣΕΛΙΔΑΣ ERASMUS+

#### 17.1 Τι είναι το SHA1

Το SHA1 (Secure Hash Algorithm 1) είναι cryptographic hash function (κρυπτογραφική συνάρτηση κατακερματισμού στα Ελληνικά). Μια μαθηματική συνάρτηση δηλαδή, ή αλλιώς ένας αλγόριθμος. Η λειτουργία του είναι αντίστοιχη με άλλες τέτοιες συναρτήσεις: του δίνουμε κάποια δεδομένα και μας επιστρέφει έναν ακέραιο αριθμό σταθερού μεγέθους. Ο αριθμός αυτός εκφράζεται συνήθως με το δεκαεξαδικό σύστημα, έχει μήκος 40 ψηφία ενώ το μέγεθός του μπορεί να ξεκινάει από τα 32bits και να ξεπερνάει τα 256. Αναφέρεται δε ως «hash sum» ή πιο απλά «hash». Τα δεδομένα που θα του δώσουμε μπορούν να είναι οτιδήποτε, από μια απλή φράση ή ένα αρχείο κειμένου μέχρι εικόνες και βίντεο. Ενδεικτικά, το SHA1 hash της φράσης «Μαθαίνουμε για το SHA1», χωρίς τα εισαγωγικά βεβαίως, είναι 81761f2fe4807e11248b90f0a150e74f39cdc5.



Εικόνα 17.1: Γραφική απεικόνιση λειτουργίας ασφάλειας SHA1

Όπως και αρκετά άλλα πράγματα στο ίδιο πεδίο, το SHA1 είναι δημιούργημα της γνωστής και μη εξαιρετέας N.S.A. Εμφανίστηκε το 1993, χρησιμοποιείται στο χώρο της κρυπτογραφίας και αυτό που κάνει βασικά είναι να ταυτοποιεί δεδομένα. Αυτό επιτυγχάνεται γιατί ο αριθμός που μας δίνει το SHA1 ως αποτέλεσμα είναι απόλυτα συνδεδεμένος με τα δεδομένα που θα του εισάγουμε κι έτσι είναι μοναδικός για κάθε εισαγωγή. Λέγοντας «κρυπτογραφία» όμως, δεν εννοούμε μόνο μυστικές υπηρεσίες και παιχνίδια κατασκοπείας αλλά και απλά, καθημερινά θέματα που είναι μεν σημαντικά, μας απασχολούν δε ελάχιστα· τα SSL

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

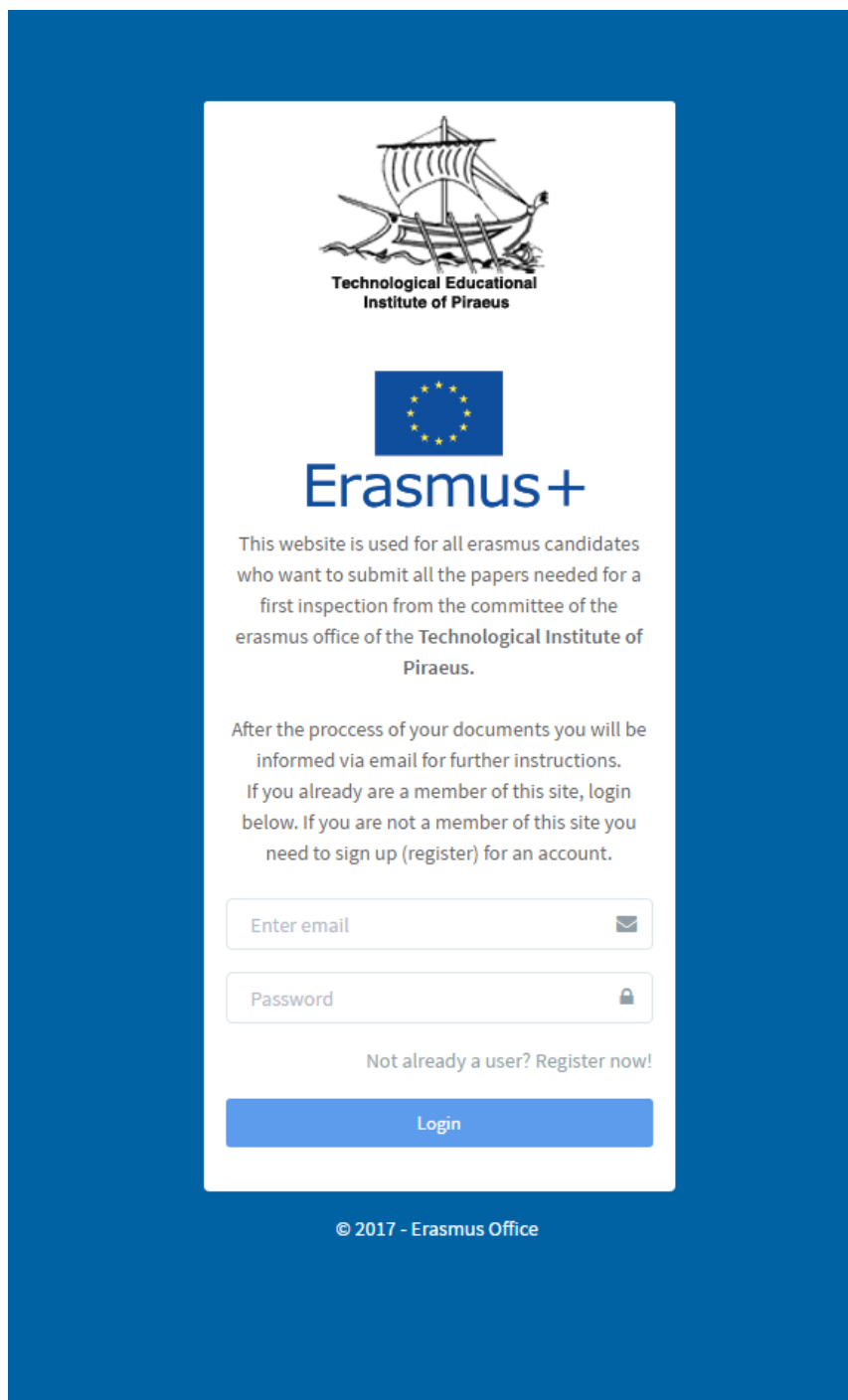
πιστοποιητικά που χρησιμοποιούν όλοι οι browsers είναι ένα τέτοιο παράδειγμα.

Για την ασφάλεια της πλατφόρμας για το γραφείο Erasmus+ του ΑΕΙ Πειραιά Τ.Τ. αυτό το είδος κρυπτογράφησης έχει χρησιμοποιηθεί.

## ΚΕΦΑΛΑΙΟ 18

### ΠΛΑΤΦΟΡΜΑ ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ. ERASMUS+ OFFICE

#### 18.1 Περιήγηση στο Site και εγγραφή - Οδηγός Χρήσης



Εικόνα 18.1: Αρχική σελίδα Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

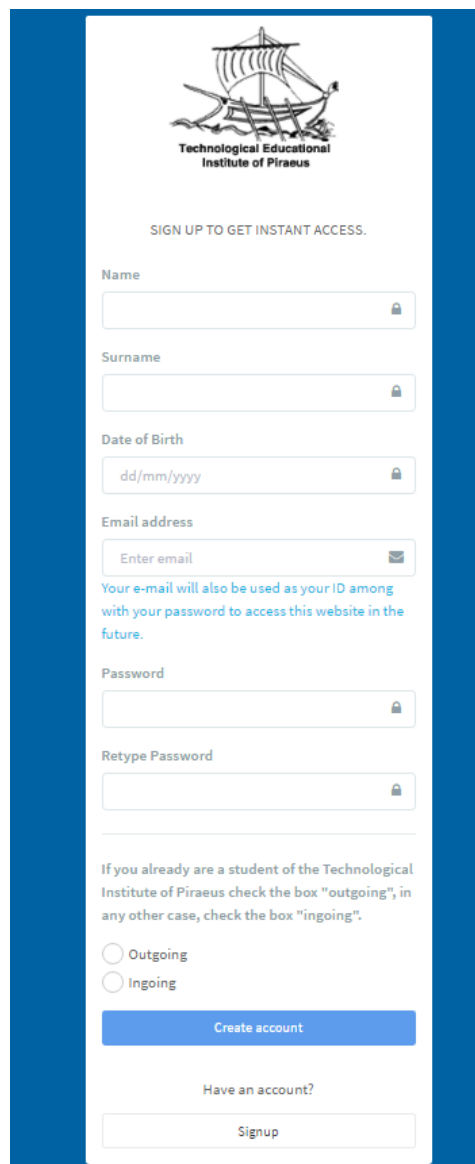
Όπως παρατηρούμε από την προηγούμενη εικόνα, εισερχόμενοι στην ιστοσελίδα, εμφανίζεται η αρχική σελίδα η οποία ζητά από τους ενδιαφερόμενους χρήστες να εγγραφούν στον ιστότοπο φτιάχνοντας ένα λογαριασμό. Σε περίπτωση προηγούμενης εισόδου με ήδη υπάρχων λογαριασμό, οι χρήστες μπορούν να εισέρθουν βάζοντας τα στοιχεία τους στα κενά.

## 18.2 Δημιουργία Λογαριασμού Χρήστη (Registration)

Για να ξεκινήσει η διαδικασία εγγραφής (registration) με σκοπό τη δημιουργία λογαριασμού οι χρήστες καλούνται να πατήσουν το:

**Not already a user? Register now!**

Το κουμπί (button) αυτό βρίσκεται ακριβώς πάνω από το κουμπί (button) με το γαλάζιο πλαίσιο “Login”.



The image shows a registration form for the Technological Educational Institute of Piraeus. At the top, there is a logo of a ship and the text "Technological Educational Institute of Piraeus". Below the logo, it says "SIGN UP TO GET INSTANT ACCESS." The form contains the following fields and options:

- Name: A text input field with a lock icon on the right.
- Surname: A text input field with a lock icon on the right.
- Date of Birth: A date input field with the placeholder "dd/mm/yyyy" and a lock icon on the right.
- Email address: A text input field with the placeholder "Enter email" and an envelope icon on the right. Below this field, there is a note: "Your e-mail will also be used as your ID among with your password to access this website in the future."
- Password: A text input field with a lock icon on the right.
- Retype Password: A text input field with a lock icon on the right.
- Below the password fields, there is a note: "If you already are a student of the Technological Institute of Piraeus check the box 'outgoing', in any other case, check the box 'ingoing'."
- Two radio buttons: "Outgoing" and "Ingoing".
- A blue button labeled "Create account".
- Below the button, there is a link: "Have an account?"
- A button labeled "Signup" with a blue border.

**Εικόνα 18.2:** Σελίδα δημιουργίας λογαριασμού χρήστη Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

Σε αυτό το σημείο πλέον ξεκινά η εγγραφή με σκοπό τη δημιουργία λογαριασμού για το site.

Συμπληρώνουμε τα στοιχεία στα κενά όπως αναγράφονται με Αγγλικούς (λατινικούς χαρακτήρες) ή με ελληνικούς χαρακτήρες.

**Name** (Όνομα)

**Surname** (Επώνυμο)

**Date of Birth** (Ημερομηνία Γέννησης – ημέρα/μήνας/χρονιά)

**Email Address** (Διεύθυνση Ηλεκτρονικού Ταχυδρομείου)

**Password** (Κωδικός για την εισαγωγή στην Ιστοσελίδα Erasmus+)

**Retype Password** (Βάζουμε ξανά τον ίδιο κωδικό πρόσβασης Password για επιβεβαίωση)

Προτού πατήσουμε το κουμπί (button) **Create Account**, για την τελική δημιουργία του λογαριασμού, επιλέγουμε μια από τις επιλογές **Outgoing** ή **Ingoing**.

#### **Outgoing**

Αφορά τους φοιτητές εκείνους οι οποίοι φοιτούν στο ΑΕΙ Πειραιά Τ.Τ. και ενδιαφέρονται να επισκεφτούν μια χώρα της Ευρωπαϊκής Ένωσης για τη διεξαγωγή των σπουδών τους στο πρόγραμμα Erasmus+.

#### **Ingoing**

Αφορά τους φοιτητές εκείνους οι οποίοι έρχονται από το εξωτερικό (χώρες τις Ευρωπαϊκής Ένωσης) με σκοπό για τη διεξαγωγή των σπουδών τους στο πρόγραμμα Erasmus+ εντός του ΑΕΙ Πειραιά Τ.Τ.

Πατώντας το κουμπί (button) **Create account** αυτόματα στέλνεται στο mail του χρήστη ένα email επιβεβαίωσης της εγγραφής.

### 18.3 Εισαγωγή στο Site Erasmus+

Έχοντας υλοποιηθεί η διαδικασία της εγγραφής, ο χρήστης μπορεί να εισέρθει στην ιστοσελίδα βάζοντας στο πρώτο πλαίσιο το **Email** του και ως **Password** τον **κωδικό** που επέλεξε στη φόρμα εγγραφής του προηγουμένως. Κατόπιν πατάει το κουμπί **Login**.



Technological Educational Institute of Piraeus

Erasmus+

This website is used for all erasmus candidates who want to submit all the papers needed for a first inspection from the committee of the erasmus office of the Technological Institute of Piraeus.

After the process of your documents you will be informed via email for further instructions.

If you already are a member of this site, login below. If you are not a member of this site you need to sign up (register) for an account.

Enter email

Password

Not already a user? Register now!

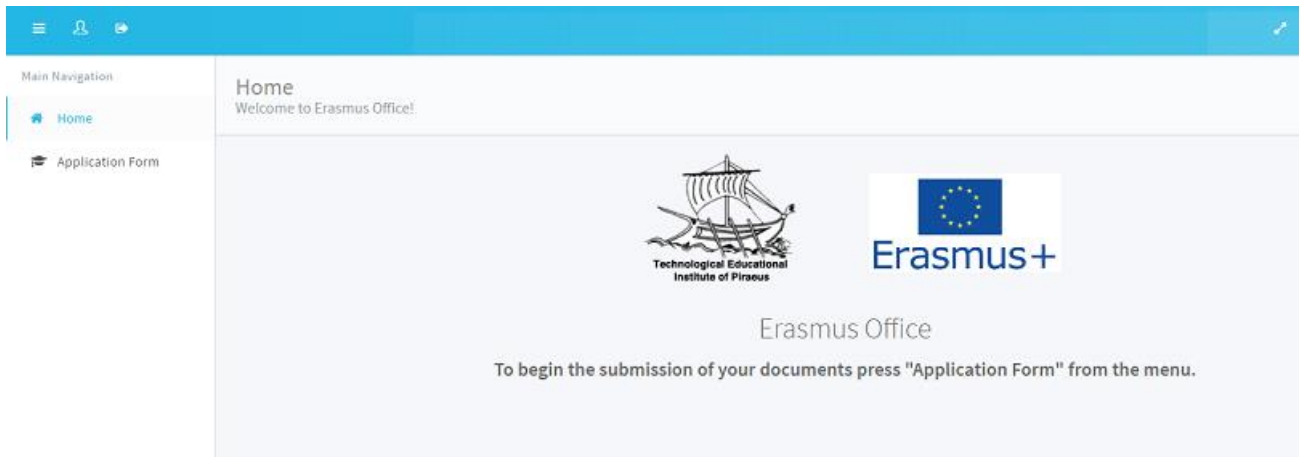
Login

© 2017 - Erasmus Office

Εικόνα 18.3: Τρόπος σύνδεσης Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

Πατώντας **Login** θα βρεθούμε στη σελίδα **Home** που στη δεξιά πλευρά περιλαμβάνει και το μενού όπως φαίνεται στην παρακάτω εικόνα.



**Εικόνα 18.4:** Βασική σελίδα Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

## 18.4 Διαδικασία Εγγραφής – Υποβολής Εγγράφων

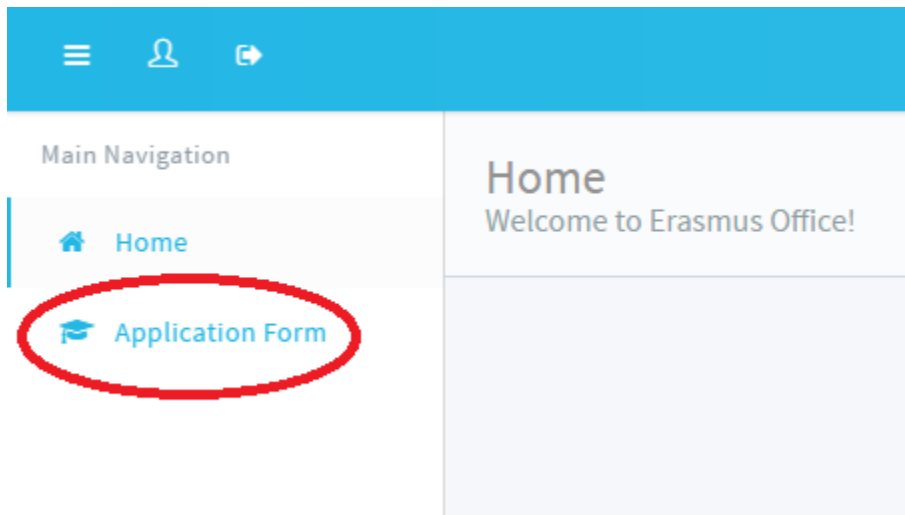
Η υποβολή εγγράφων για το πρόγραμμα Erasmus χωρίζεται σε δυο μέρη. Αυτό συμβαίνει διότι ο φοιτητής αρχικά πρέπει να υποβάλει 4 έγγραφα (στα οποία αναφερόμαστε εκτενέστερα στην επόμενη σελίδα) από τον συνολικό αριθμό εγγράφων που απαιτούνται για να γίνει η αξιολόγηση του ως προς το αν μπορεί ως υποψήφιος να πληρεί της προϋποθέσεις του προγράμματος. Στην περίπτωση που επιλεγεί από την επιτροπή του γραφείου, καλείται από εκείνη κατόπιν ενημέρωσης της στον φοιτητή να υποβάλει και τα υπόλοιπα έγγραφα που απαιτούνται για την ολοκλήρωση της διαδικασίας υποβολής εγγράφων. Τέλος, όπως θα δούμε και παρακάτω, ο φοιτητής καλείται με τη φυσική του παρουσία στο γραφείο του προγράμματος Erasmus με σκοπό την τελική επαλήθευση των εγγράφων του οπού και η διαδικασία τερματίζεται.

## 18.5 Μέρος Πρώτο

Όπως αναγράφεται και στο κέντρο της συγκεκριμένης σελίδας (**To begin the submission of your documents press "Application Form" from the menu**), για να ξεκινήσει ο ενδιαφερόμενος φοιτητής την υποβολή των εγγράφων που απαιτούνται για την επιλογή του από το πρόγραμμα του Erasmus+, επιλεγεί από το μενού στα δεξιά το **Application Form**.

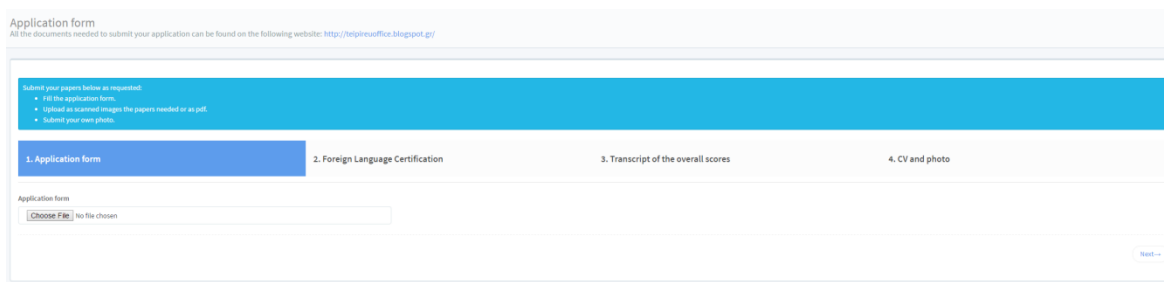


Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.



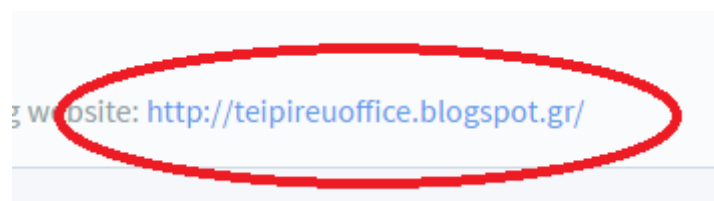
Εικόνα 18.5: Κουμπιά εκκίνησης διαδικασίας υποβολής εγγράφων

Έπειτα, στη νέα σελίδα που εμφανίζεται, ο φοιτητής πρέπει να υποβάλει συμπληρωμένα όλα τα έγγραφα που αρχικά απαιτεί το γραφείο Erasmus+ του ΑΕΙ Πειραιά Τ.Τ.



Εικόνα 18.6: Έγγραφα για υποβολή

Τα έγγραφα τα οποία απαιτούνται για συμπλήρωση όπως ήδη αναγράφεται και στη σελίδα υποβολής τους στη σελίδα αυτή, μπορούν να τα κατεβάσουν και να τα συμπληρώσουν οι ενδιαφερόμενοι φοιτητές από το παρακάτω **Link**:  
<http://teipireuoffice.blogspot.gr/>



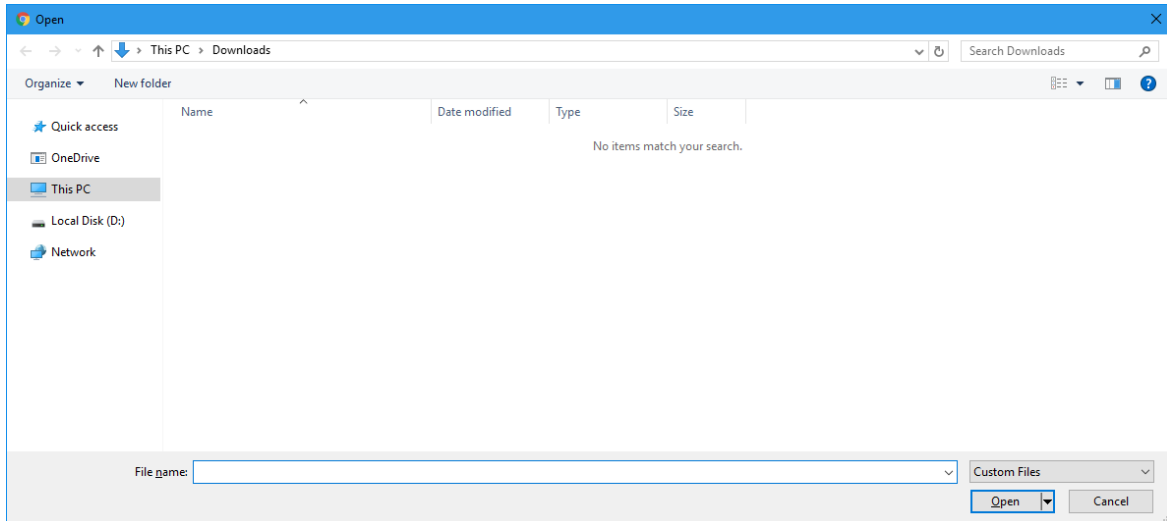
Εικόνα 18.7: Link ανεύρεσης εγγράφων προς υποβολή

Τα έγγραφα που αρχικά πρέπει να ανεβάσουν οι φοιτητές αφού τα συμπληρώσουν είναι:

- Application form** (Φόρμα αίτησης για πρόγραμμα Erasmus+)
- Foreign Language Certification** (Πιστοποιητικό ξένης γλώσσας)
- Transcript of the overall scores** (Αντίγραφο αναλυτικής βαθμολογίας)
- CV and photo** (Βιογραφικό και φωτογραφία)

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

Για να εκχωρήσουμε το κάθε αρχείο πατάμε **Choose File** και επιλέγουμε το αρχείο που θέλουμε να ανεβάσουμε από το παράθυρο **Windows Explorer** που θα ανοίξει επιλέγοντας **Open**. Πατώντας Next και επαναλαμβάνουμε τη διαδικασία για το επόμενο αρχείο. Αφού όλα τα αρχεία τοποθετηθούν, πατάμε το κουμπί **Save** και τα αρχεία ανεβαίνουν και τα λαμβάνει το γραφείο του Erasmus+.



**Εικόνα 18.8:** Παράθυρο διαλόγου για ανέβασμα αρχείου προς υποβολή

Όλα τα έγγραφα εκτός του Application Form πρέπει να είναι σε μορφή **PDF**, όπως επίσης και η φωτογραφία πρέπει να οποιοδήποτε είδος **αρχείου εικόνας**.

Με την ολοκλήρωση της διαδικασίας ο φοιτητής ενημερώνεται αυτομάτως με το παρακάτω μήνυμα της ιστοσελίδας:

**After 1-2 months period an email will be send to you to verify your acceptance to the Erasmus. Once you receive this email login to this website again to submit any further documents needed.**

Στα ελληνικά:

**Μετά από 1-2 μήνες, θα σας αποσταλεί ένα μήνυμα ηλεκτρονικού ταχυδρομείου για την επιβεβαίωση της αποδοχής σας στο Erasmus. Μόλις λάβετε αυτό το μήνυμα ηλεκτρονικού ταχυδρομείου, συνδεθείτε ξανά σε αυτόν τον ιστότοπο, για να υποβάλετε τα περαιτέρω απαραίτητα έγγραφα.**

Στους φοιτητές που θα επιλεγούν από το γραφείο του Erasmus+ του ΑΕΙ Πειραιά Τ.Τ. για το πρόγραμμα, αφού γίνει η έγκριση των εγγράφων τους στέλνεται ένα **email** το οποίο αναγράφει:

**You have been accepted to the erasmus program to submit your final documents. Please login to the website to finalise the process.**

Στα ελληνικά:

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

**Έχετε γίνει δεκτός/ή στο πρόγραμμα erasmus για να υποβάλετε τα τελικά σας έγγραφα. Παρακαλούμε συνδεθείτε στον ιστότοπο για να ολοκληρώσετε τη διαδικασία.**

Αφού συμβεί αυτό ο φοιτητής πρέπει να εισέρθει ξανά στην ιστοσελίδα και να καταχωρήσει με τον ίδιο τρόπο που εξηγήσαμε προηγουμένως τα υπόλοιπα έγγραφα που απαιτούνται για να τελειοποιήσει τη διαδικασία εισαγωγής του στο πρόγραμμα του Erasmus+.

## 18.6 Μέρος Δεύτερο

Για την ολοκλήρωση της εγγραφής στο πρόγραμμα Erasmus ο φοιτητής καλείται να ξανασυνδεθεί στην ιστοσελίδα ώστε να συμπληρώσει και να ανεβάσει τα υπολειπόμενα 4 έγγραφα που απαιτούνται για την τελική διεξαγωγή της διαδικασίας.

**Εικόνα 18.9:** Υπόλοιπα έγγραφα για υποβολή

Τα έγγραφα που πρέπει να ανεβάσουν τώρα οι φοιτητές αφού τα συμπληρώσουν είναι:

**Application to the university** (Αίτηση στο πανεπιστήμιο)

**LAC** (Learning Agreement - Συμφωνητικό μάθησης)

**Accommodation Form** (Φόρμα Καταλύματος/Διαμονής)

**Acceptance Confirmation** (Επιβεβαίωση Όρων Αποδοχής)

Ακολουθούμενοι την ίδια διαδικασία με προηγουμένως. (Για να εκχωρήσουμε το κάθε αρχείο πατάμε **Choose File** και επιλέγουμε το αρχείο που θέλουμε να ανεβάσουμε από το παράθυρο **Windows Explorer** που θα ανοίξει επιλέγοντας **Open**. Πατώντας Next και επαναλαμβάνουμε τη διαδικασία για το επόμενο αρχείο. Αφού όλα τα αρχεία τοποθετηθούν, πατάμε το κουμπί **Save** και τα αρχεία ανεβαίνουν και τα λαμβάνει το γραφείο του Erasmus+.)

Με το πέρας της διαδικασίας αυτής ο φοιτητής ενημερώνεται άμεσα με το ακόλουθο μήνυμα:

**You have completed all the steps successfully! An email will be send to you for further instructions.**

Στα ελληνικά:

**Έχετε ολοκληρώσει όλα τα βήματα με επιτυχία! Θα σας αποσταλεί ένα μήνυμα ηλεκτρονικού ταχυδρομείου για περαιτέρω οδηγίες.**

Το μήνυμα ηλεκτρονικού ταχυδρομείου αναγράφει το εξής:

**Thank you for your submitted papers. We request the week following to attend to the erasmus office of Technological Institute of Piraeus to submit all your papers for the last verification needed.**

Στα ελληνικά:

**Σας ευχαριστούμε για την υποβολή των εγγράφων σας. Ζητούμε την επόμενη εβδομάδα να παρευρεθείτε στο γραφείο Erasmus του ΑΕΙ Πειραιά Τ.Τ για την τελική υποβολή των έγγραφων σας και την επαλήθευση/ταυτοποίηση που απαιτείται.**

Όπως αναφέρεται και στο **mail** προηγουμένως, απαιτείται η φυσική παρουσία των φοιτητών για την ταυτοποίηση και εξακρίβωση των (υλικής μορφής) εγγράφων τους. Τέλος, με αυτό το βήμα, η διαδικασία εγγραφής – υποβολής εγγραφών λαμβάνει τέλος.

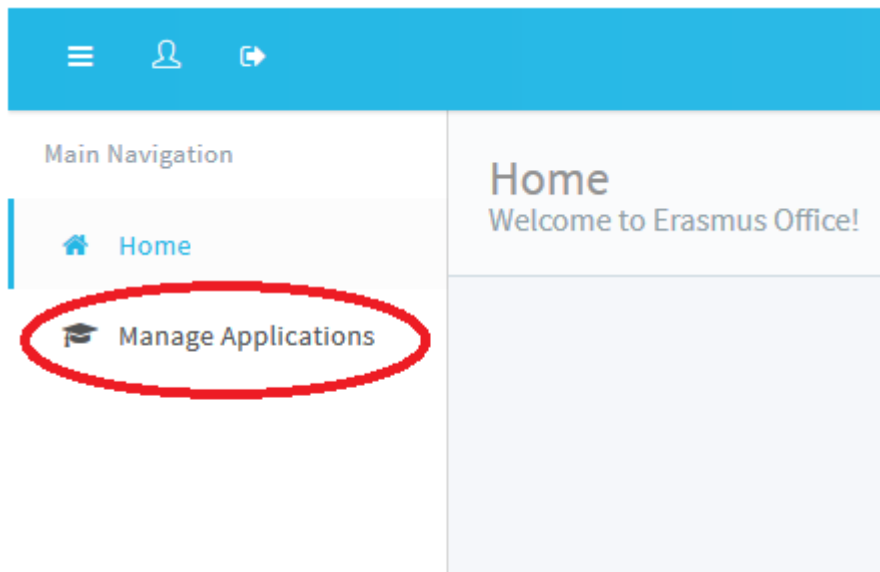
### **18.7 Διαχειριστικό Erasmus+ Office Site**

Για την χρήση της ιστοσελίδας από πλευράς διαχειριστή (Administrator) καλείται ο εκάστοτε διαχειριστής να εισέρθει στην ιστοσελίδα μέσω των διαπιστευτηρίων (ID/Mail και Password) διαχειριστή τα οποία δεν αναγράφονται εδώ για λόγους ασφαλείας. Τα διαπιστευτήρια αυτά τα γνωρίζουν αυτοί που έχουν οριστεί ως υπεύθυνοι από το γραφείο Erasmus για τη λειτουργία του ιστότοπου και για την επιλογή των φοιτητών που πληρούν τις προϋποθέσεις συμμετοχής και εγγραφής στο πρόγραμμα Erasmus.

Αρχικά, για την σύνδεση στην ιστοσελίδα ως διαχειριστής τοποθετούμε στο πλαίσιο mail και password τα διαπιστευτήρια του διαχειριστή.

**\*\*\*ΠΡΟΣΟΧΗ\*\*\* Έχει οριστεί μόνο ένας λογαριασμός (account) διαχειριστή και μπορεί να συνδεθεί για μια δεδομένη χρονική στιγμή από ένα υπολογιστή μόνο και όχι από περισσότερους συγχρόνως!**

Εισερχόμενοι στην ιστοσελίδα ως διαχειριστές πλέον, εμφανίζεται στο μενού δεξιά το παρακάτω όπως βλέπουμε και από την εικόνα που ακολουθεί:



**Εικόνα 18.10:** Κουμπί για τον έλεγχο αιτήσεων στο διαχειριστικό Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

Πατώντας **Manage Applications** μπορούμε να δούμε όλες τις τρέχουσες αιτήσεις ενδιαφερόμενων φοιτητών για το πρόγραμμα του Erasmus καθώς και αυτές που έχουν ήδη εγκριθεί. Για τις αιτήσεις που έχουν εγκριθεί κρατούνται ως ιστορικό.

Στην παρακάτω εικόνα απεικονίζεται το μενού χωρίς καθόλου αιτήσεις.



The image shows two tables stacked vertically. The top table is titled 'Successfull applications' and has two columns: 'Description' and 'Approve'. The bottom table is titled 'Approve Applicants' and also has two columns: 'Description' and 'Approve'. Both tables are currently empty.

**Εικόνα 1.1:** Καρτέλες αιτήσεων στο διαχειριστικό Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

Στην καρτέλα **Approve Applicants** καταχωρούνται και εμφανίζονται όσοι έχουν ολοκληρώσει το πρώτο μέρος της διαδικασίας εγγραφής τους και περιμένουν την έγκριση του γραφείου Erasmus για να περάσουν στο δεύτερο μέρος για να τελειοποιήσουν την εγγραφή τους. Στην καρτέλα **Successful Applicants**,

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

καταχωρούνται όσοι φοιτητές έχουν λάβει έγκριση από το γραφείο του Erasmus και έχουν γίνει πλέον δεκτοί στο πρόγραμμα.  
Ακολουθεί παράδειγμα στην εικόνα παρακάτω:

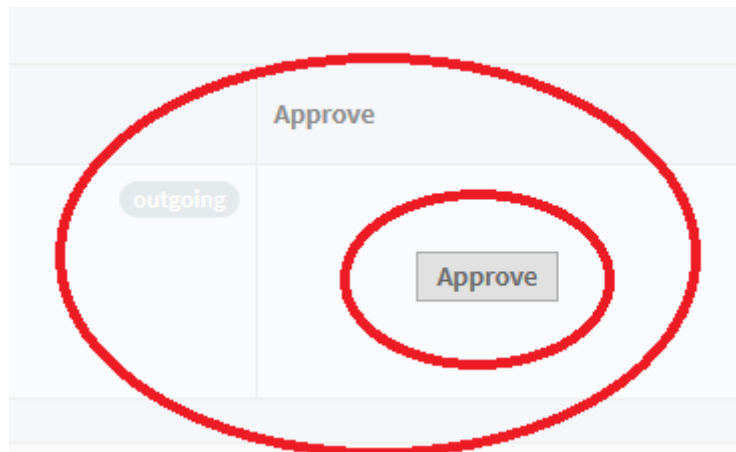
Successful applications	
Description	Approve
 <b>Sta Geo</b> Date of Birth: 28/4/1992 Category: outgoing <a href="#">Application Form</a> , <a href="#">Foreign Language Certification</a> , <a href="#">CV</a> , <a href="#">Transcript of the overall scores</a> , <a href="#">Photo</a> , <a href="#">Application Form to the University</a> , <a href="#">LAC</a> , <a href="#">Accommodation Form</a> , <a href="#">Acceptance Confirmation</a>	<input type="button" value="outgoing"/>
 <b>Nikos Koukos</b> Date of Birth: 28/4/1992 Category: outgoing <a href="#">Application Form</a> , <a href="#">Foreign Language Certification</a> , <a href="#">CV</a> , <a href="#">Transcript of the overall scores</a> , <a href="#">Photo</a> , <a href="#">Application Form to the University</a> , <a href="#">LAC</a> , <a href="#">Accommodation Form</a> , <a href="#">Acceptance Confirmation</a>	<input type="button" value="outgoing"/>

Approve Applicants	
Description	Approve
<b>Nikos Korobos</b> Date of Birth: 13/2/1990 Category: outgoing <a href="#">Application Form</a> , <a href="#">Foreign Language Certification</a> , <a href="#">CV</a> , <a href="#">Transcript of the overall scores</a> , <a href="#">Photo</a>	<input type="button" value="outgoing"/> <input type="button" value="Approve"/>

**Εικόνα 18.11:** Παράδειγμα με ενεργές αιτήσεις και στις δυο καρτέλες στο διαχειριστικό Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

Για να περάσει ένας υποβαλλόμενος από την καρτέλα **Approve Applicants** στην καρτέλα **Successful Applicants**, αρκεί να πατηθεί το κουμπί **Approve**.



**Εικόνα 18.12:** Κουμπί για την έγκριση φοιτητή στο διαχειριστικό Erasmus Office ΑΕΙ Πειραιά Τ.Τ.

Με το πάτημα αυτό, όπως αναφέραμε και προηγουμένως από την πλευρά του αιτούμενου φοιτητή, στέλνεται το αυτοματοποιημένο **mail**:

**You have been accepted to the erasmus program to submit your final documents. Please login to the website to finalise the process.**

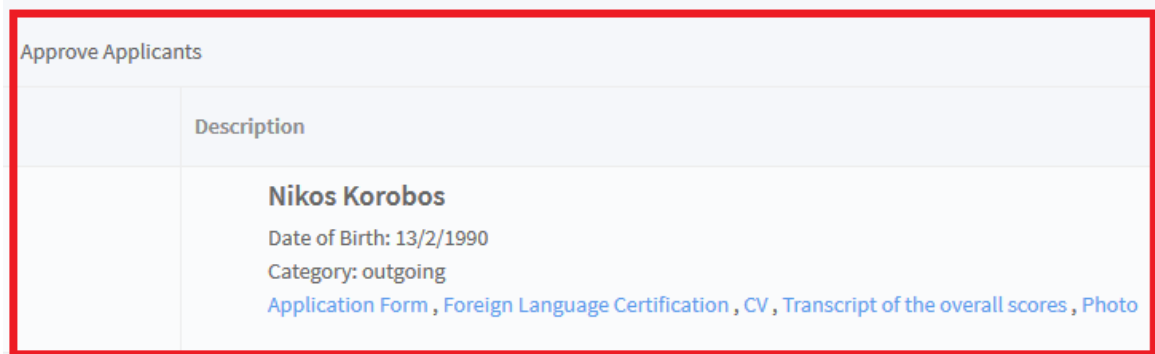
Όπως προαναφέραμε συμπληρώνοντας και ανεβάζοντας και τα υπόλοιπα έγγραφα του, ο φοιτητής ενημερώνεται πάλι από το αυτοματοποιημένο **mail**:

Γεωργάς Σ. Ευστάθιος  
Κογιανός Α. Δημήτριος

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

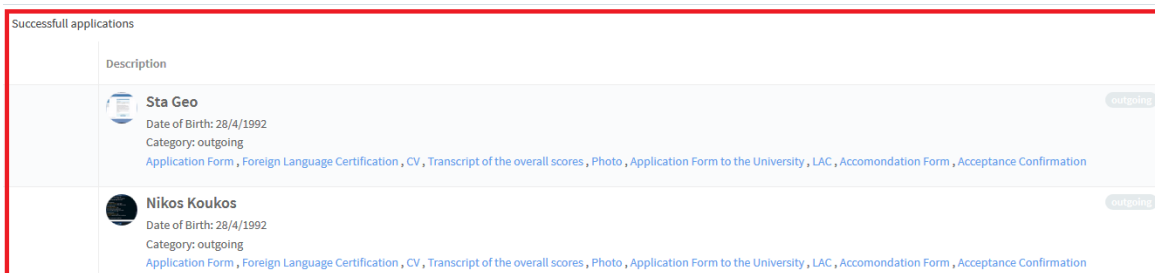
**Thank you for your submitted papers. We request the week following to attend to the erasmus office of Technological Institute of Piraeus to submit all your papers for the last verification needed.**



Και αφού συμβούν τα παραπάνω, ο φοιτητής από την καρτέλα **Approve Applicants** όπως φαίνεται στην εικόνα, καταχωρείται μαζί με τους υπόλοιπους επιτυχόντες στην καρτέλα **Successful Applicants** μετά το πάτημα του **Approve**.



	Description
	<b>Nikos Korobos</b> Date of Birth: 13/2/1990 Category: outgoing <a href="#">Application Form</a> , <a href="#">Foreign Language Certification</a> , <a href="#">CV</a> , <a href="#">Transcript of the overall scores</a> , <a href="#">Photo</a>

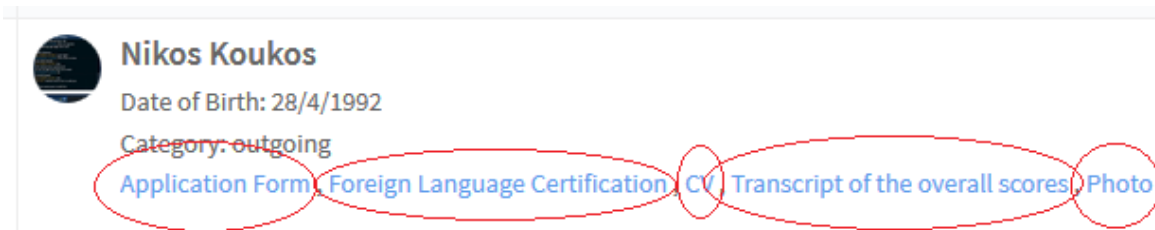
**Εικόνα 18.13:** Παράδειγμα φοιτητή προς έγκριση με το πρώτο σκέλος των απαιτούμενων αρχείων




	Description
	 <b>Sta Geo</b> Date of Birth: 28/4/1992 Category: outgoing <a href="#">Application Form</a> , <a href="#">Foreign Language Certification</a> , <a href="#">CV</a> , <a href="#">Transcript of the overall scores</a> , <a href="#">Photo</a> , <a href="#">Application Form to the University</a> , <a href="#">LAC</a> , <a href="#">Accommodation Form</a> , <a href="#">Acceptance Confirmation</a>
	 <b>Nikos Koukos</b> Date of Birth: 28/4/1992 Category: outgoing <a href="#">Application Form</a> , <a href="#">Foreign Language Certification</a> , <a href="#">CV</a> , <a href="#">Transcript of the overall scores</a> , <a href="#">Photo</a> , <a href="#">Application Form to the University</a> , <a href="#">LAC</a> , <a href="#">Accommodation Form</a> , <a href="#">Acceptance Confirmation</a>

**Εικόνα 18.14:** Παράδειγμα φοιτητών που έχουν λάβει ήδη έγκριση με όλο το πλήθος των αρχείων τους

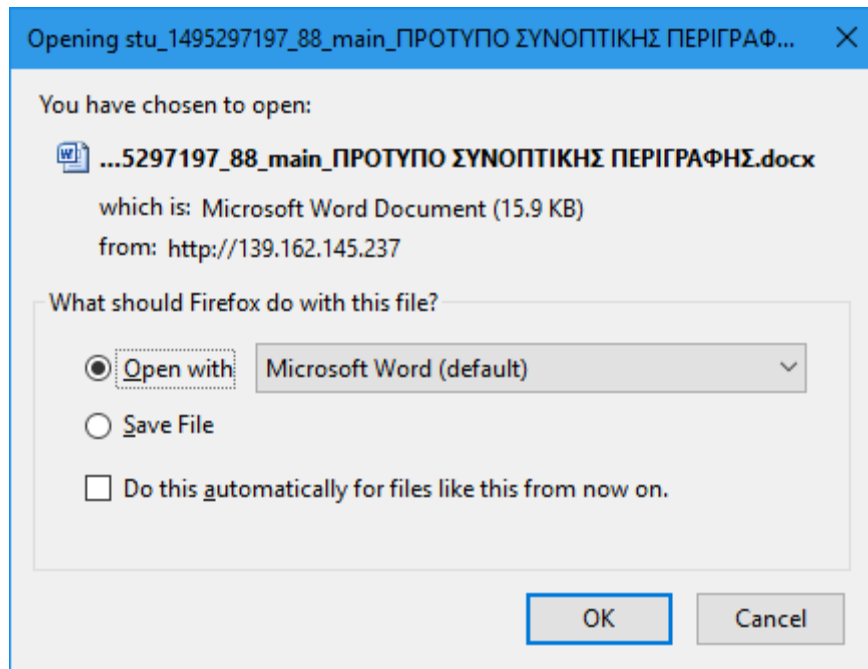
Τα αρχεία των αιτούμενων φοιτητών μπορεί να τα κατεβάσει ο διαχειριστής για να διαβάσει απλά πατώντας επάνω τους. Αυτό φυσικά ισχύει για όλα τα έγγραφα που υποβάλλονται.



 <b>Nikos Koukos</b> Date of Birth: 28/4/1992 Category: outgoing <a href="#">Application Form</a> , <a href="#">Foreign Language Certification</a> , <a href="#">CV</a> , <a href="#">Transcript of the overall scores</a> , <a href="#">Photo</a>
--

**Εικόνα 18.15:** Πλήθος αρχείων αιτούμενου φοιτητή που για λήψη, έλεγχο και έγκριση από το διαχειριστή

Πατώντας πάνω σε ένα έγγραφο ανοίγει το παράθυρο διαλόγου όπως φαίνεται παρακάτω για επιβεβαίωση της λήψης.



**Εικόνα 18.16:** Παράθυρο διαλόγου λήψης αρχείου των Windows

Αφού γίνει η λήψη του αρχείου, ο διαχειριστής μπορεί να το ανοίξει και να ελέγξει όπως επιθυμεί ο ίδιος.

Στο σημείο αυτό, η διαδικασία λήψης και αξιολόγησης των εγγράφων τελειώνει και από πλευράς διαχειριστή καθώς η σελίδα δεν περιέχει αλλά λειτουργικά στοιχεία για τα οποία να μπορεί να γίνει αναφορά και ανάλυση. Άλλωστε ο σκοπός δημιουργίας της σελίδας αυτής είναι η συμβολή της στην επιτάχυνση μιας διαδικασίας συλλογής, εξακρίβωσης και επικύρωσης εγγράφων για την εισαγωγή στο πρόγραμμα του Erasmus+. Με αυτό τον τρόπο επιτυγχάνονται πιο οργανωμένα αποτελέσματα για την διεξαγωγή μιας διαδικασίας δημιουργώντας στην ουσία με ηλεκτρονική πλατφόρμα αλληλεπίδρασης και αποθήκευσης δεδομένων.



Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΠΑΡΑΡΤΗΜΑ Α'

Στο παράρτημα αυτό παρατίθεται μικρό μέρος του κώδικα ανάπτυξης της παρούσας εφαρμογής. Ο λόγος που δεν έχει συμπεριληφθεί μεγαλύτερο μέρος κώδικα είναι επειδή θεωρήθηκε περιττό, μιας και ολόκληρος ο κώδικας που χρησιμοποιήθηκε για τη δημιουργία της πλατφόρμας για το γραφείο Erasmus+ του ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ. υπάρχει ατόφιος σε ξεχωριστό αρχείο που έχει παραδοθεί μαζί με την πτυχιακή εργασία. Το μεγαλύτερο μέρος του κώδικα περιέχει και σχόλια για να είναι επεξηγηματικός σε άτομα που ενδεχομένως στο μέλλον να χρειαστεί να τροποποιήσουν και να βελτιώσουν τον κώδικα αν υπάρχει κάποιο πρόβλημα (bug) ή εάν χρειαστεί κάποιο επιπλέον χαρακτηριστικό για προσθήκη από το γραφείο του Erasmus+ του ΑΕΙ Πειραιά Τ.Τ.

Το μέρος κώδικα που ακολουθεί παρακάτω είναι στο γραμμένο στο Phalcon Framework.

```
use Phalcon\Mvc\Router;

$routeur = new Router();

/////*****Common*****/////
$routeur->addPost("/login", array(
    'controller' => 'Users',
    'action' => 'login',
));
$routeur->addPost("/register", array(
    'controller' => 'Users',
    'action' => 'register',
));
// *****Schools*****//
$routeur->addPost("/schools", array(
    'controller' => 'Users',
    'action' => 'uploaddocs',
));
$routeur->addPost("/schools2", array(
    'controller' => 'Users',
    'action' => 'uploaddocs2',
));
//getStudentsSuccess
$routeur->addPost("/approve", array(
    'controller' => 'Users',
    'action' => 'approve',
));
$routeur->addGet("/getStudents", array(
    'controller' => 'Users',
    'action' => 'getStudents',
));
$routeur->addGet("/getStudentsSuccess", array(
    'controller' => 'Users',
    'action' => 'getStudentsSuccess',
));
return $router;
```

Μέρος κώδικα από Phalcon Framework που περιγράφει κομμάτι του routing της εφαρμογής.

Ανάπτυξη διαδικτυακής πλατφόρμας υποβολής και επεξεργασίας εγγράφων προγράμματος Erasmus+ φοιτητών ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### ΑΝΑΦΟΡΕΣ

1. Tsalgaidou, T. Pilioura, "An Overview of Standards and Related
2. Technology in Web Services". International Journal of Distributed and
3. Parallel Databases, Special Issue on E-Services, 12(2), Sep 2002, 135-162.
4. Graham et al., Building Web Services with Java: Making Sense of XML,
5. SOAP, WSDL and UDDI, Sams Publishing 2001.
6. G. Alonso, F. Casati, H. Kuno, V. Machiraju, Web Services Concepts,
7. Architectures and Applications, Springer-Verlag 2004
8. Werner Vogels, "Web Services are not Distributed Objects", IEEE Internet
9. Computing, Nov-Dec 2003.
10. HTML 4 «Conformance: requirements and recommendations».
11. Μπέρνερς Λι, Τιμ (Μάρτιος 1989, Μάιος 1990). «Information Management: A Proposal». CERN. Μπέρνερς Λι, Τιμ (1990). «Design Issues». World Wide Web Consortium. Μπέρνερς Λι, Τιμ (1990). «Design Issues, Uses». World Wide Web Consortium. «Tags used in HTML». World Wide Web Consortium. 3 Νοεμβρίου 1992. «First mention of HTML Tags on the www-talk mailing list». World Wide Web Consortium. 20 Οκτωβρίου 1991. «Index of elements in HTML 4». World Wide Web Consortium. 24 Δεκεμβρίου 1999. Μπέρνερς Λι, Τιμ. «"Hypertext Markup Language (HTML)" Internet-Draft». Μπέρνερς Λι, Τιμ (9 Δεκεμβρίου 1991). «Re: SGML/HTML docs, X Browser». Ρέιμοντ, Έρικ. «IETF and the RFC Standards Process» (στα Αγγλικά). The Art of Unix Programming. «In IETF tradition, standards have to arise from experience with a working prototype implementation — but once they become standards, code that does not conform to them is considered broken and mercilessly scrapped. ...Internet-Drafts are not specifications; software implementers and vendors are specifically barred from claiming compliance with them as if they were specifications. Internet-Drafts are focal points for discussion, usually in a working group... Once an Internet-Draft has been published with an RFC number, it is a specification to which implementers may claim conformance. It is expected that the authors of the RFC and the community at large will begin correcting the specification with field experience.»
12. «HTML+ Internet-Draft — Abstract». «Browser writers are experimenting with extensions to HTML and it is now appropriate to draw these ideas together into a revised document format. The new format is designed to allow a gradual roll over from HTML, adding features like tables, captioned figures and fill-out forms for querying remote databases or mailing questionnaires.»
13. Berners-Lee, Tim; Connolly, Dan (Νοέμβριος 1995). «RFC 1866: Hypertext Markup Language - 2.0 - Acknowledgments». Internet Engineering Task

- Force. «Since 1993, a wide variety of Internet participants have contributed to the evolution of HTML, which has included the addition of in-line images introduced by the NCSA Mosaic software for WWW. Dave Raggett played an important role in deriving the forms material from the HTML+ specification. Dan Connolly and Karen Olson Muldrow rewrote the HTML Specification in 1994. The document was then edited by the HTML working group as a whole, with updates being made by Eric Schieler, Mike Knezovich and Eric W. Sink at Spyglass, Inc. Finally, Roy Fielding restructured the entire draft into its current form.»
14. Berners-Lee, Tim; Connolly, Dan (November 1995). «RFC 1866: Hypertext Markup Language - 2.0 - Introduction». Internet Engineering Task Force. «This document thus defines an HTML 2.0 (to distinguish it from the previous informal specifications). Future (generally upwardly compatible) versions of HTML with new features will be released with higher version numbers.»
  15. Raggett, Dave (1998). Raggett on HTML 4. «HTML 3.2 Reference Specification». World Wide Web Consortium. 4 Ιανουαρίου 1997. «IETF HTML WG». «Note: This working group is closed»
  16. «HTML 4.0 Specification». World Wide Web Consortium. 18 Δεκεμβρίου 1997. Arnoud Engelfriet. «Introduction to Wilbur». Web Design Group. «HTML 4 - 4 Conformance: requirements and recommendations».
  17. «HTML 4.0 Specification». World Wide Web Consortium. 24 Απριλίου 1998. «HTML 4.01 Specification». World Wide Web Consortium. 24 Δεκεμβρίου 1999. «HTML 4.01 Errata». 12 Μαΐου 2001. «ISO/IEC 15445:2000 - Information technology -- Document description and processing languages -- HyperText Markup Language (HTML)». ISO. 2000. «Information technology — Document description and processing languages — HyperText Markup Language (HTML)». Connolly, Daniel (6 Ιουνίου 1992). «MIME as a hypertext architecture». CERN. Connolly, Daniel (15 Ιουλίου 1992). «HTML DTD enclosed». CERN. Connolly, Daniel (18 August 1992). «document type declaration subset for HyperText Markup Language as defined by the World Wide Web project». CERN. Connolly, Daniel (24 November 1992). «Document Type Definition for the HyperText Markup Language as used by the World Wide Web application». CERN. Ενότητα «Revision History»
  18. Μπέρνερς Λι, Τιμ (Ιούνιος 1993). «Hypertext Markup Language (HTML) Internet Draft version 1.1». IETF IIR Working Group. Μπέρνερς Λι, Τιμ (Ιούνιος 1993). «Hypertext Markup Language (HTML) Internet Draft version 1.2». IETF IIR Working Group. Μπέρνερς Λι, Τιμ (28 Νοεμβρίου 1994). «HyperText Markup Language Specification - 2.0 INTERNET DRAFT». IETF. «HTML 3.0 Draft (Expired!) Materials». World Wide Web Consortium. 12 Δεκεμβρίου 1995. «HyperText Markup Language Specification Version 3.0». Raggett, Dave (28 Μαρτίου 1995). «HyperText Markup Language Specification Version 3.0». HTML 3.0 Internet Draft Expires in six months. World Wide Web Consortium. Bowers, Neil. Weblint: Just Another Perl Hack.

19. Lie, Håkon Wium. Bert, Bos (Απρίλιος 1997). Cascading style sheets: designing for the Web. Addison Wesley Longman, σελ. 263. «HTML5». World Wide Web Consortium. 10 Ιουνίου 2008. «HTML5, one vocabulary, two serializations». «XHTML 1.0: The Extensible HyperText Markup Language (Second Edition)». World Wide Web Consortium. 26 Ιανουαρίου 2000. «XHTML 1.1 - Module-based XHTML — Second Edition». World Wide Web Consortium. 16 Φεβρουαρίου 2007. «XHTML 2.0». World Wide Web Consortium. 26 Ιουλίου 2006. «XHTML 2 Working Group Expected to Stop Work End of 2009, W3C to Increase Resources on HTML5». World Wide Web Consortium. 17 Ιουλίου 2009. «HTML5». World Wide Web Consortium. 24 Οκτωβρίου 2008. «Activating Browser Modes with Doctype».
20. «On SGML and HTML». World Wide Web Consortium. «XHTML 1.0 - Differences with HTML 4». World Wide Web Consortium. Korpela, Jukka (6 Ιουλίου 1998). «Why attribute values should always be quoted in HTML». «Objects, Images, and Applets in HTML documents». World Wide Web Consortium. 24 Δεκεμβρίου 1999 «H56: Using the dir attribute on an inline element to resolve problems with nested directional runs». Techniques for WCAG 2.0. World Wide Web Consortium. «The Named Character Reference &apos;». World Wide Web Consortium. 26 Ιανουαρίου 2000.
21. «The Unicode Standard: A Technical Introduction». Μπέρνερς Λι, Τιμ. Φισκέτι, Μαρκ (2000). Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor. San Francisco: Harper. ISBN 978-0-06-251587-X.
22. Raggett, Dave (2002). «Adding a touch of style». World Wide Web Consortium. Το άρθρο αυτό σημειώνει ότι η εμφανισιακή σήμανση HTML μπορεί να είναι ακόμα χρήσιμη για browser «πριν από τον Netscape 4.0 και τον Internet Explorer 4.0». Δείτε και τον Κατάλογο των Περιηγητών ιστού για τους browser που δημιουργήθηκαν πριν από το 1997.
23. Tim Berners-Lee, James Hendler και Ora Lassila (2001). «The Semantic Web». Scientific American. Nigel Shadbolt, Wendy Hall and Tim Berners-Lee (2006). «The Semantic Web Revisited». IEEE Intelligent Systems. «XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)». World Wide Web Consortium. 2000, revised 2002. «XHTML Documents which follow the guidelines set forth in Appendix C, "HTML Compatibility Guidelines" may be labeled with the Internet Media Type "text/html" [RFC2854], as they are compatible with most HTML browsers. Those documents, and any other document conforming to this specification, may also be labeled with the Internet Media Type "application/xhtml+xml" as defined in [RFC3236].»
24. «RFC 2119: Key words for use in RFCs to Indicate Requirement Levels». Harvard University. 1997. «3. SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.»

25. «XHTML 1.1 - Module-based XHTML — Second Edition». World Wide Web Consortium. 2007. «XHTML 1.1 documents SHOULD be labeled with the Internet Media Type text/html as defined in [RFC2854] or application/xhtml+xml as defined in [RFC3236].»
26. Sauer, C.: WYSIWIKI - Questioning WYSIWYG in the Internet Age. In: Wikimania (2006)
27. Spiesser, J., Kitchen, L.: Optimization of html automatically generated by WYSIWYG programs. In: 13th International Conference on World Wide Web, pp. 355--364. WWW '04. ACM, New York, NY (New York, NY, U.S., May 17–20, 2004)
28. «[whatwg] WHAT open mailing list announcement». Lists.whatwg.org. «[whatwg] HTML5 at Last Call (at the WHATWG)». Lists.whatwg.org. «HTML 5: A vocabulary and associated APIs for HTML and XHTML.». W3C. «When will HTML5 be finished?». WHATWG. WHATWG Wiki.
29. Leonard Richardson & Sam Ruby, RESTful Web Services
30. Chatterjee - Webber, Developing Enterprise Web Services, Prentice Hall PTR , N.Jersey 2004
31. "Relationship to the World Wide Web and REST Architectures". Web Services Architecture. W3C. Retrieved 2011-04-22.
32. Newcomer E., Lomow G (2005) " Understanding SOA with Web Services": Addison – Wesley Professional
33. Mathew Macdonald. Microsoft.NET Distributed Applications: Integrating XML Web Services and .NET Remoting, 2001
34. Gustavo Alonso, Fabio Casati, Harumi Kuno and Vijay Machiraju. Web Services Concepts, Architectures and Applications, Springer, 2004
35. "What's in a name? History of the Erasmus Programme". Archived from the original on 2013-04-04.
36. Ec.europa.eu, Table: Erasmus student mobility (number of outgoing students): 1987/88-2006/07
37. <http://ec.europa.eu/citizens-initiative/public/initiatives/ongoing/details/2012/000001>
38. Simona Pronckuté (November 1, 2013). "European Citizens Initiatives – one year of challenges". EuropeanPublicAffairs.eu. Retrieved August 20, 2014.
39. "European Commission - PRESS RELEASES - Press release - Green light for Erasmus+: More than 4 million to get EU grants for skills and employability". Retrieved 30 December 2014.
40. Regulation (EU) No 1288/2013 of the European Parliament and of the Council of 11 December 2013 establishing 'Erasmus+': the Union programme for education, training, youth and sport and repealing Decisions No 1719/2006/EC, No 1720/2006/EC and No 1298/2008/EC Text with EEA relevance
41. Procedure file of COD 2011/0371
42. "Table 8: Actual Number of Erasmus Students by Country of Home institution 1987/88-2006/07".
43. "Press release--Another record-breaking year for Erasmus".
44. "Swiss students out of Erasmus program starting in 2015". Retrieved 30 December 2014.

45. "Student rooms and accommodation, internships and erasmus guides.". Retrieved 30 December 2014.
46. "Erasmus 24\_7 Official Website".
47. Coffey, Mark (2012). Erasmus, Here I Come!.
48. This social aspect is fostered also by some commercial companies. Companies that help stimulating mobility are for example <http://www.housinganywhere.com/>, on which students can rent out rooms to each other; <http://www.linguistadores.com/> that helps students to learn new languages.
49. Bennhold, Katrin (26 April 2005). "Quietly sprouting: A European identity". International Herald Tribune. Retrieved 21 November 2006.

## **ΧΡΗΣΙΜΟΙ ΙΣΤΟΤΟΠΟΙ**

### **ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΤΙΚΑ ΜΕ ΥΠΗΡΕΣΙΕΣ ΙΣΤΟΥ**

1. <http://www.webservicesarchitect.com/>
2. <http://www.w3schools.com/webservices/default.asp>
3. <http://wiki.eeng.dcu.ie/ee557/g3/296-EE.html>
4. <https://www.wikipedia.org/>
5. <https://el.wikipedia.org/>
6. <https://el.wikipedia.org/wiki/HTML5>
7. <https://el.wikipedia.org/wiki/HTML>
8. <https://el.wikipedia.org/wiki/CSS>
9. <https://el.wikipedia.org/wiki/JavaScript>
10. <http://www.w3.org/TR/wsarch/#relwwwrest>
11. <http://www.w3.org/XML>
12. <http://www.w3.org/TR/ws-arch/>
13. <http://www.w3.org/TR/xmlschema-0/>
14. <http://www.w3.org/TR/soap/>
15. <http://www.w3.org/TR/wsdl/>
16. <http://www.w3.org/TR/wsdl20/>
17. <http://www.oasis-open.org/>
18. <http://www.xml.org/>
19. <http://www.uddi.org/>
20. <http://www.w3schools.com/webservices/default.asp>
21. <http://www.w3schools.com/dtd/default.asp>
22. <http://www.w3schools.com/soap/default.asp>
23. <http://www.w3schools.com/wsdl/default.asp>
24. <http://www.w3schools.com/xml/default.asp>
25. <http://www.roseindia.net/webservices/webservices.shtml>
26. <http://www.it.uom.gr/project/soap/Downloads/WebservicesTheory.pdf>
27. <http://www.hitmill.com/webservices/index.html>
28. [http://en.wikipedia.org/wiki/Web\\_application](http://en.wikipedia.org/wiki/Web_application)
29. [http://en.wikipedia.org/wiki/Web\\_services](http://en.wikipedia.org/wiki/Web_services)



30. <http://en.wikipedia.org/wiki/XML>
31. <http://en.wikipedia.org/wiki/HTTP>
32. <http://www.w3.org/DesignIssues/WebServices.html>
33. [www.digitalnews.gr/2037/net-framework](http://www.digitalnews.gr/2037/net-framework)
34. <http://www.w3schools.com/webservices/ws-example.asp>
35. <http://www.onjava.com/lpt/a/1025>
36. SOAP - <http://www.w3.org/TR/soap12-part0/>
37. WSDL 2.0 - <http://www.w3.org/TR/wsdl20-primer>
38. WSDL 1.1 - <http://www.w3.org/TR/wsdl>
39. UDDI - <http://uddi.xml.org/>

## HTML / HTML 5

1. <http://www.it.uom.gr/project/html2/tut0.html>
2. <http://www.eeei.gr/odhgos/htmlfaq.htm>

## CSS

1. Flanagan, 2006, σ. 1
2. «ECMAScript Language Overview» (PDF). 4. «ECMAScript Language Specification».
3. Douglas Crockford on Functional JavaScript >2:49< blinkx (flv)
4. The Little JavaScripter shows the relationship with Scheme in more detail.
5. Πρότυπο ECMA-262
6. Krill, Paul (2008-06-23). «JavaScript creator ponders past, future». InfoWorld. Hamilton, Naomi (2008-06-31). «The A-Z of Programming Languages: JavaScript». computerworld.com.au.
7. Press release announcing JavaScript, "Netscape and Sun announce Javascript", PR Newswire, December 4, 1995
8. «TechVision: Innovators of the Net: Brendan Eich and JavaScript». Web.archive.org. Αρχειοθετήθηκε από το πρωτότυπο στις 2008-02-08. Brendan Eich (3 April 2008). «Popularity». ECMAScript 3rd Edition specification
9. «JavaScript: The World's Most Misunderstood Programming Language». Crockford.com. Kris Kowal (1 December 2009). «CommonJS effort sets JavaScript on path for world domination». Ars Technica. Condé Nast Publications.