



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

### Ομογενοποίηση και Μεταφόρτωση Δεδομένων Ηλεκτρονικών Περιοδικών



Του φοιτητή  
Κουτούλα Απόστολου  
Αρ. Μητρώου: 06/3012

Επιβλέπων καθηγητής  
Δημήτρης Α. Δέρβος

Θεσσαλονίκη 2012

# Περίληψη

Στον Σύνδεσμο Ελληνικών Ακαδημαϊκών Βιβλιοθηκών (Σ.Ε.Α.Β.) με το πέρασμα των χρόνων έχει δημιουργηθεί ένας πελώριος όγκος δεδομένων σε ψηφιακή μορφή από επιστημονικά περιοδικά στα οποία είναι συνδρομητής, που στέλνονται σε αυτόν από εκδότες. Ο όγκος αυτός συνεχίζει καθημερινά να αυξάνεται.

Έτσι θα πρέπει όλος αυτός ο όγκος να αντιγραφεί σε δίσκους ώστε να γίνει λεπτομερής καταγραφή του τι υπάρχει σε ψηφιακή μορφή και να δημιουργηθεί ή να χρησιμοποιηθεί ένα λογισμικό που να διαθέτει σε χρήστες του Σ.Ε.Α.Β. τα παραπάνω δεδομένα του.

Τα παραπάνω τελικά τεύχη περιοδικών ή τα άρθρα είναι κυρίως αποθηκευμένα σε μορφή PDF.

Με τον προτεινόμενο τρόπο , δηλαδή με τη φόρτωση σε δίσκους θα μπορεί να γίνεται έλεγχος σε όλο το ψηφιακό υλικό που στέλνουν κάθε φορά οι εκδότες και θα εντοπίζονται τυχόν λάθη, ελλείψεις, ελαττώματα, καθώς και στα ήδη υπάρχον δεδομένα.

# Ευχαριστίες

*Θα ήθελα να ευχαριστήσω τον επιβλέποντα της παρούσας πτυχιακής εργασίας καθηγητή κ. Δημήτρη Δέρβο για την υποστήριξη, την υπομονή και την εμπιστοσύνη που μου έδειξε.*

*Ένα μεγάλο ευχαριστώ στον κ. Λεωνίδα Πισπιρίγκα, για την επιμονή του να μου δείξει και να με κατευθύνει σε συγκεκριμένους στόχους.*

*Ακόμη, θα ήθελα να ευχαριστήσω την οικογένεια μου που στέκεται στο πλευρό μου όλα αυτά τα χρόνια και υποστηρίζει με αγάπη την κάθε επιλογή μου.*

## Περιεχόμενα

Κεφάλαιο 1. Περιγραφή του προβλήματος.....	7
1.1. Το πρόβλημα προς επίλυση .....	7
1.2. Εκδότες ηλεκτρονικών περιοδικών.....	8
1.3. Θέματα προς ανάλυση .....	8
Κεφάλαιο 2. Η μορφή των δεδομένων.....	9
2.1. Elsevier.....	9
2.2. Springer.....	11
2.3. Kluwer.....	15
2.4. IEEE .....	16
Κεφάλαιο 3. Μεθοδολογία Υλοποίησης.....	19
3.1 Εργαλεία Ανάπτυξης.....	19
3.2 Apache HTTP Server .....	20
3.2.1 Το ξεκίνημα και η εξέλιξη του Apache .....	21
3.2.2 Η προέλευση του ονόματός του.....	21
3.2.3 Γιατί Apache HTTP;.....	21
3.3 PHP .....	22
3.3.1 Η ιστορία της PHP .....	23
3.3.2 Γιατί PHP; .....	23
3.4 SQL.....	24
3.4.1 Ιστορία.....	24
3.4.2 Γιατί MySQL; .....	25
3.5 XML .....	26
3.5.1 Προέλευση και στόχοι (Origin and goals).....	28
3.5.2 Εισαγωγή στο XML DOM .....	29
3.6 JavaScript.....	31
3.7 AJAX.....	32
Κεφάλαιο 4. ΧΑΜΡΡ .....	33
4.1 Τι είναι ΧΑΜΡΡ.....	33
4.1.1 Δυνατότητες και απαιτήσεις.....	33
4.1.2 Χρήση .....	34
4.2 Εγκατάσταση Λογισμικού ΧΑΜΡΡ .....	35
Κεφάλαιο 5. Εισαγωγή στις Βάσεις Δεδομένων .....	46
5.1 Τα Δεδομένα και οι Πληροφορίες .....	46

5.2 Βάσεις Δεδομένων .....	47
5.3 Κατηγορίες χρηστών.....	48
5.4 Γλώσσες Βάσεων Δεδομένων.....	50
5.5 Μοντέλο Οντοτήτων Συσχετίσεων.....	51
5.6 Σχεσιακό Μοντέλο Δεδομένων.....	52
Κεφάλαιο 6. Σχεδίαση της Βάσης Δεδομένων.....	53
6.1 Σχεδίαση διαγράμματος ER.....	53
6.2 Σχεδίαση Σχεσιακού Σχήματος Βάσης Δεδομένων.....	54
6.2.1 Αναλυτική Περιγραφή Πινάκων Βάσης Δεδομένων .....	55
6.2.2 Δημιουργία της βάσης δεδομένων με χρήση της MySQL.....	57
Κεφάλαιο 7. Ανάλυση, επεξεργασία και αποθήκευση των δεδομένων .....	60
7.1. ELSEVIER.....	60
7.2. SPRINGER.....	62
7.3. KLUWER.....	63
7.4. IEEE .....	65
Κεφάλαιο 8. Scripts .....	67
8.1 1 <sup>ο</sup> script: Εισαγωγή εγγραφών και έλεγχος για διπλότυπες εγγραφές .....	67
8.2 2ο Script: Έλεγχος για τον ακριβή αριθμό των δεδομένων.....	71
8.2.1. Journals.....	71
8.2.2. VOLUMES .....	73
8.2.3. ISSUES .....	74
8.3 3ο Script: Έλεγχος εγγραφών .....	76
8.4 4ο Script: Τυχαίος έλεγχος εγγραφών.....	77
8.5 5ο Script: Διαγραφή των πιο πρόσφατα εισαγμένων εγγραφών .....	78
8.6 6ο Script: Διαγραφή όλων των εγγραφών.....	79
8.7 7ο Script: Χρήση AJAX.....	80
Κεφάλαιο 9. Μελέτη και σύγκριση CMS.....	82
9.1 Τι είναι ένα CMS.....	82
9.1.1 Από τι αποτελείται.....	82
9.1.2 Τα χαρακτηριστικά ενός CMS.....	82
9.1.3 Το περιεχόμενο.....	83
9.1.4 Χρήση των CMS .....	84
9.1.5 Οι στόχοι ενός Συστήματος Διαχείρισης Διαδικτυακού .....	85
Περιεχομένου (WCMS).....	85

9.1.6 Πλεονεκτήματα Συστημάτων Διαχείρισης Περιεχομένου .....	88
9.2 Σύγκριση.....	90
9.2.1 Πλεονεκτήματα και μειονεκτήματα διαφόρων CMS.....	90
Κεφάλαιο 10. Συμπεράσματα και Μελλοντικές Προοπτικές .....	93
Βιβλιογραφία.....	94

# Κεφάλαιο 1. Περιγραφή του προβλήματος

Στο κεφάλαιο αυτό περιγράφουμε το πρόβλημα που έχουμε να αντιμετωπίσουμε και αναφέρουμε τι πρέπει να κάνουμε για την επίλυσή του. Παρουσιάζουμε τους εκδότες των ηλεκτρονικών περιοδικών από τα οποία θα αντλήσουμε τις πληροφορίες μας και αναφέρουμε τι άλλο θα πρέπει να πραγματοποιήσουμε στο πλαίσιο της πτυχιακής.

## 1.1. Το πρόβλημα προς επίλυση

Στα 10 χρόνια λειτουργίας του Σ.Ε.Α.Β. έχει συσσωρευτεί και συνεχίζει να συσσωρεύεται ένας τεράστιος όγκος/αρχείο επιστημονικού περιεχομένου σε ψηφιακή μορφή που αφορά στα αρχεία των τευχών των περιοδικών των συνδρομών του Σ.Ε.Α.Β. σε ψηφιακή μορφή, που παραδίδονται από τους εκδότες, για φύλαξη και μελλοντική χρήση αν και εφόσον απαιτηθεί για οποιοδήποτε λόγο. Πρόκειται ήδη για έναν τεράστιο, αλλά όχι επακριβώς προσδιορισμένο, όγκο περιεχομένου, ο οποίος συνεχίζει να αυξάνεται με ακαθόριστα μεγάλους ρυθμούς.

Η συσσώρευση αυτή του περιεχομένου οδηγεί στην ανάγκη αντιμετώπισης δύο βασικών ζητημάτων:

1. την αντιγραφή των πάσης φύσεως CD, DVD, hard disk, magnetic tapes που έχουν σταλεί και αποστέλλονται ετησίως από τους εκδότες σε δίσκους έτσι ώστε να γίνει δυνατή η καταγραφή του περιεχομένου (ποιοί τίτλοι περιοδικών, έτη, τεύχη) περιλαμβάνονται στα παραπάνω μέσα ΚΑΙ
2. τη δημιουργία ή χρησιμοποίηση υπάρχοντος λογισμικού για τη διάθεση του παραπάνω περιεχομένου στους χρήστες του Σ.Ε.Α.Β., στην περίπτωση που δεν υπάρξει ανανέωση της συνδρομής με κάποιον από τους εκδότες.

Αυτή τη στιγμή ο Σ.Ε.Α.Β. έχει backfiles (αντίγραφα) από έναν σημαντικό αριθμό εκδοτών, οι οποίοι τα προμηθεύουν με διαφορετικές δομές και πρακτικές αποθήκευσης και περιγραφής των δεδομένων. Τα backfiles είναι αρχεία αποθηκευμένα σε διάφορα μέσα και μορφές.

Τα μέσα αποθήκευσης είναι: CD-ROM, DVD-ROM, Εξωτερικοί σκληροί δίσκοι, DLT Tapes, FTPserver (download απευθείας από τον διακομιστή του εκδότη)

Τα τελικά τεύχη περιοδικών ή τα άρθρα (fulltext αρχεία) είναι αποθηκευμένα κυρίως σε μορφή PDF (ορισμένοι τίτλοι έχουν fulltext σε μορφή ascii text).

Μέχρι τώρα ο ΣΕΑΒ συγκεντρώνει το ψηφιακό υλικό που στέλνουν οι εκδότες σε διαφορετικές μορφές (CD, tapes, disks etc) χωρίς να είναι σε θέση να ελέγχει αν όντως περιέχονται όλοι οι τίτλοι των επιστημονικών περιοδικών που καλύπτονται από τις συμβάσεις και όλα τα έτη με αποτέλεσμα να μην είναι σε θέση και να ζητήσει να αποσταλούν τυχόν ελλείψεις ή τυχόν ελλατωματικά cds, tapes κτλ.

Πέραν από την καταγραφή των ήδη υπαρχόντων αρχείων που καλύπτουν διάστημα μεγαλύτερο της δεκαετίας (γιατί έχουν αγοραστεί και backfiles) θα δοθεί η δυνατότητα ο έλεγχος αυτός να γίνεται συστηματικά στο μέλλον κάθε φορά που παραλαμβάνεται ψηφιακό υλικό από τους εκδότες. Επιπλέον εφόσον τα αρχεία θα είναι φορτωμένα σε δίσκους θα γίνεται περιστασιακά έλεγχος για την ακεραιότητα των δεδομένων και των μέσων αποθήκευσης, καθώς επίσης θα είναι εύκολη η μεταφορά σε νεώτερης γενιάς τεχνικό εξοπλισμό όταν παραστεί η ανάγκη.

Τέλος, θα μελετήσουμε και θα συγκρίνουμε περιβάλλοντα Content Management System (CMS) για την δημοσιοποίηση πληροφοριών.

## **1.2. Εκδότες ηλεκτρονικών περιοδικών**

Έχουμε στην διάθεσή μας ηλεκτρονικά περιοδικά από τους εξής εκδότες:

- Elsevier
- Springer
- Kluwer
- IEEE

Από αυτούς θα συλλέξουμε τα δεδομένα που μας ενδιαφέρουν, θα τα ομαδοποιήσουμε και θα τα προσθέσουμε στην βάση δεδομένων.

## **1.3. Θέματα προς ανάλυση**

Θα μελετήσουμε ορισμένα περιβάλλοντα CMS, θα καταγράψουμε τα χαρακτηριστικά τους και θα τα συγκρίνουμε για να δούμε ποιο είναι το ιδανικότερο για την δημοσιοποίηση των πληροφοριών μας.



## Κεφάλαιο 2. Η μορφή των δεδομένων

Σε αυτό το κεφάλαιο θα δούμε τη μορφή των δεδομένων των περιοδικών κάθε εκδότη, και θα καταγράψουμε τι πληροφορίες μας δίνονται για τα περιοδικά, παρέχοντας παράλληλα και εικόνες.

### 2.1. Elsevier

Ο Elsevier δίνει τα metadata (δεδομένα για τα δεδομένα) σε αρχεία με την κατάληξη .toc. Παρακάτω παραθέτω ένα στιγμιότυπο με την μορφή που παρουσιάζονται όλες οι πληροφορίες.

```
_t0 YYTLO41A
_vn 4.1
_pd 20041112
_st ITM
_t1 YYTLO41A 01406736
_cd LANCAO
_jn The Lancet
_cr Copyright (c) 2004 Elsevier
_t2 YYTLO41A 01406736 18250103
_ps [S350]
_vl 4
_is 104-107
_pr 353-384+nil+nil+nil+nil+nil+nil+nil+nil+nil+nil+nil+nil
_dt 18250924
_np 44
_pn 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371
    372 373 374 375 376 377 378 379 380 381 382 383 384 nil nil nil nil nil nil
    nil nil nil nil nil nil
_t3 YYTLO41A 01406736 18250103 02943370
_ps [S350]
_ii S0140-6736(02)94337-0
_ty FLA
_li EN
_ti FROM LECTURES ON THE PRINCIPLES AND PRACTICE OF PHYSIC,
_au Armstrong
_pg 353-358
_mf [SGML ART 4.3.1 HEAD-AND-TAIL] main
_mf [PDF 1.2 WRAPPED OPTIMIZED] main
_mf [Raw ASCII] main
```

Εικόνα 2.1.1 - Elsevier

Στην εικόνα 2.1.1 μας ενδιαφέρουν τα εξής δεδομένα:

<b>Μορφή</b>	<b>Περιγραφή</b>
<b>_t1</b>	Το issn του περιοδικού
<b>_jn</b>	Ο τίτλος του περιοδικού
<b>_vl</b>	Το νούμερο του τόμου
<b>_is</b>	Το νούμερο του τεύχους
<b>_pr</b>	οι σελίδες του τεύχους
<b>_dt</b>	Η ημερομηνία έκδοσης του τεύχους
<b>_ii</b>	Το DOI του άρθρου
<b>_ti</b>	ο τίτλος του άρθρου
<b>_au</b>	Ο/Η συγγραφέας (author)
<b>_ab</b>	Η περίληψη του άρθρου
<b>_kw</b>	Οι λέξεις κλειδιά (keywords) του άρθρου
<b>_pg</b>	οι σελίδες του άρθρου

Εδώ βλέπουμε ότι η ημερομηνία έχει την μορφή YYYYMMDD

## 2.2. Springer

Ο Springer δίνει τα metadata σε αρχεία .xml και .xml.Meta. Παρακάτω παραθέτω στιγμιότυπα που δείχνουν πως παρουσιάζονται όλες οι πληροφορίες.

```
<PublisherInfo>
  <PublisherName>Birkhäuser Verlag</PublisherName>
  <PublisherLocation>Basel</PublisherLocation>
</PublisherInfo>

<Journal>
  <JournalInfo JournalProductType="Legacy" NumberingStyle="Unnumbered">
    <JournalID>00010</JournalID>
    <JournalPrintISSN>0001-9054</JournalPrintISSN>
    <JournalSPIN/>
    <JournalTitle>Aequationes Mathematicae</JournalTitle>
    <JournalSubTitle/>
  </JournalInfo>
</Journal>
```

Εικόνα 2.2.1- Springer

Εδώ βλέπουμε:

Μορφή	Περιγραφή
<PublisherName>	Το όνομα του εκδότη
<JournalPrintISSN>	Το ISSN του περιοδικού
<JournalTitle>	Ο τίτλος του περιοδικού

```

<VolumeInfo VolumeType="Regular">
  <VolumeIDStart>55</VolumeIDStart>
  <VolumeIDEnd>55</VolumeIDEnd>
  <VolumeIssueCount/>
</VolumeInfo>

<Issue IssueType="Regular">
  <IssueInfo TocLevels="0">
    <IssueIDStart>1</IssueIDStart>
    <IssueIDEnd>2</IssueIDEnd>
    <IssueHistory>
      <PrintDate>
        <Year>1998</Year>
        <Month>02</Month>

```

Εικόνα 2.2.2 - Springer

Εδώ βλέπουμε:

Μορφή	Περιγραφή
<VolumeIDStart>	Το νούμερο του volume
<IssueIDStart>	Το νούμερο(α) του τεύχους

```

<CoverDate>
  <Year>1998</Year>
  <Month>02</Month>
</CoverDate>

```

Εικόνα 2.2.3-Springer

Εδώ βλέπουμε:

Μορφή	Περιγραφή
<Year>	Ο χρόνος της έκδοσης του τεύχους
<Month>	Ο μήνας έκδοσης του τεύχους

```

<ArticleID>2278</ArticleID>

<ArticleDOI/>

<ArticleTitle Language="En">Asymptotic analysis of a class of functional equations</Arti

<ArticleSubTitle Language="En"/>

<ArticleCategory/>

<ArticleFirstPage>91</ArticleFirstPage>

<ArticleLastPage>105</ArticleLastPage>

```

Εικόνα 2.2.4-Springer

Εδώ βλέπουμε:

Μορφή	Περιγραφή
<ArticleTitle>	Ο τίτλος του άρθρου
<ArticleFirstPage>	Η πρώτη σελίδα του άρθρου
<ArticleLastPage>	Η τελευταία σελίδα του άρθρου

```

<AuthorName DisplayOrder="Western">
  <GivenName>F.</GivenName>
  <FamilyName>Vogl</FamilyName>
</AuthorName>
</Author>

<Affiliation ID="A1">
  <OrgName>Dept. of Mathematics and Computer Science, Ben Gurion University,

```

Εικόνα 2.2.5 - Springer

Εδώ βλέπουμε:

Μορφή	Περιγραφή
<GivenName>	Το όνομα του συγγραφέα
<FamilyName>	Το επώνυμο του συγγραφέα
<OrgName>	Το όνομα του οργανισμού

```

<Abstract ID="Abs1" Language="En">
  <Heading>Summary.</Heading>
  <Para> The aim of this paper is to solve the q-difference equation <Emphasis Typ
    <Subscript>1</Subscript> (<Emphasis Type="Italic">z</Emphasis>) = <Emphasis T

```

Εικόνα 2.2.6 - Springer

Εδώ βλέπουμε:

Μορφή	Περιγραφή
<Abstract>	Η περίληψη του άρθρου
<Keyword>	Οι λέξεις κλειδιά (keywords) του άρθρου

## 2.3. Kluwer

Ο Kluwer δίνει τα metadata σε αρχεία με την κατάληξη .oa3. Παρακάτω παραθέτω ένα στιγμιότυπο με την μορφή που παρουσιάζονται όλες οι πληροφορίες.

```

<pnm>Kluwer Academic/Plenum Publishers</pnm>
<loc>New York, U.S.A.</loc>
<jtl>African Archaeological Review</jtl>
<jcode>AARR</jcode>
<issn>0263-0338</issn>
<issueinfo>
<binding vid = "14"
      iid = "4">
<coverdate>199712</coverdate>
<artinfo>
<artty type = "RESEARCH-ARTICLE">
<categ>
<pages ppct = "26"
      ppf = "231"
      ppl = "256">
<crn>Plenum Publishing Corporation</crn>
<atl>Evidence for Interaction from Recent Hunter-Gatherer Sites in the Caledon Valley</atl>
<au>
<fnms>Carolyn</fnms>
<snm>Thorp</snm>
<aff>Museum of Human Sciences, P.O. Box CY 33, Causeway Harare, Zimbabwe</aff>
<abs><p><it>Evidence for contact between hunter-gatherers and agropastoralists from excavate
<kwds>
<kwd>southern Africa

```

Εικόνα 2.3.1-Kluwer

Εδώ μας ενδιαφέρουν τα εξής δεδομένα:

Μορφή	Περιγραφή
<jtl>	Ο τίτλος του περιοδικού
<issn>	Το ISSN του περιοδικού
vid	Το νούμερο του τόμου
iid	Το νούμερο του τεύχους
<coverdate>	Η ημερομηνία έκδοσης του τεύχους (μορφή YYYYMM)
ppf	Η πρώτη σελίδα του άρθρου στο τεύχος
ppl	Η τελευταία σελίδα του άρθρου στο τεύχος
<atl>	ο τίτλος του άρθρου
<fnms>	Το όνομα του συγγραφέα
<snm>	Το επίθετο του συγγραφέα
<aff>	Ο οργανισμός
<abs>	Η περίληψη του άρθρου
<kwds>	Οι λέξεις κλειδιά (keywords) του άρθρου

## 2.4. IEEE

Η IEEE δίνει τα metadata σε αρχεία .xml. Παρακάτω παραθέτω στιγμιότυπα που δείχνουν πως παρουσιάζονται όλες οι πληροφορίες.

```
<publication>
  <title><![CDATA[IEEE Transactions on Industrial Informatics]]></title>

  <titleabbrev><![CDATA[IEEE Trans. Ind. Inf.]]></titleabbrev>
```

Εικόνα 2.4.1-IEEE

Εδώ μας ενδιαφέρουν τα εξής δεδομένα:

Μορφή	Περιγραφή
<title>	Ο τίτλος του περιοδικού

```
<issn mediatype="Paper">1551-3203</issn>

<issn mediatype="Online">0000-0000</issn>

<copyright>
  <year>2006</year>

  <holder>IEEE</holder>
</copyright>

<publisher>
  <publishername>IEEE</publishername>
```

Εικόνα 2.4.2-IEEE

Μορφή	Περιγραφή
<issn>	Το issn του περιοδικού
<year>	Ο χρόνος της έκδοσης του περιοδικού
<publishername>	Το όνομα του εκδότη



```

<volume>
<volumeinfo>
  <year>2006</year>
  <volumenum>2</volumenum>
  <idamsid>0b0000648011bcd5</idamsid>
  <volumestatus>Active</volumestatus>
  <mediatype>Electronic</mediatype>
  <numissues>3</numissues>
  <issue>
    <issuenum>1</issuenum>
    <issuepart></issuepart>
    <uri></uri>
    <amsid>33541</amsid>
    <amscreatedate></amscreatedate>
    <issuestatus>Complete</issuestatus>
    <numpages></numpages>
    <shelfnum></shelfnum>
    <totalarticles>13</totalarticles>

```

Εικόνα 2.4.3-IEEE

Μορφή	Περιγραφή
<volumenum>	Το νούμερο του τόμου

```

<article>
<title><![CDATA[IEEE Transactions on Industrial Informatics
<articleinfo>
  <articleseqnum>13</articleseqnum>
  <articledoi>10.1109/TII.2006.870916</articledoi>
  <idamsid>0b0000648011bcfb</idamsid>
  <mediatype></mediatype>
  <articletype typecode="99"></articletype>
  <articlestatus>Active</articlestatus>
  <contenttype>info-author</contenttype>
  <holdstatus>Publish</holdstatus>
  <issuenum>1</issuenum>

```

Εικόνα 2.4.4-IEEE

Μορφή	Περιγραφή
<title>	Ο τίτλος του άρθρου
<articledoi>	το DOI του άρθρου
<issuenum>	ο αριθμός του τεύχους

```

<abstract abstracttype="Regular"><![CDATA[When sensory and actuation devices in a contr
<abstract abstracttype="Long"></abstract>
<abstract abstracttype="Violation"></abstract>
<abstract abstracttype="Standard"></abstract>
<authorgroup>
  <author>
    <normname>Feng-Li Lian</normname>
    <nonnormname>Feng-Li Lian</nonnormname>
    <authorrefid>1</authorrefid>
    <honorific></honorific>
    <lineage></lineage>
    <firstname></firstname>
    <othername></othername>
    <surname>Feng-Li Lian</surname>
    <affiliation><![CDATA[Dept. of Electr. Eng., Nat. Taiwan Univ., Taipei, Taiwan]]></

```

Εικόνα 2.4.5-IEEE

Μορφή	Περιγραφή
<abstract>	Η περίληψη του άρθρου
<normname>	Το πλήρες όνομα του συγγραφέα
<affiliation>	Ο οργανισμός στον οποίο δουλεύει ο συγγραφέας

```

<keywordset keywordtype="Inspec">
  <keyword>
    <keywordterm><![CDATA[telecommunication computing]]></keywordterm>
  </keyword>
  <keyword>
    <keywordterm><![CDATA[multi-agent systems]]></keywordterm>
  </keyword>

```

Εικόνα 2.4.6-IEEE

Μορφή	Περιγραφή
<keyword>	Το keyword του άρθρου

# Κεφάλαιο 3. Μεθοδολογία Υλοποίησης

Η ενσωμάτωση της Πληροφορικής και πιο συγκεκριμένα, του διαδικτύου (Internet) και του Παγκόσμιου Ιστού (World Wide Web), σε διάφορες πτυχές της καθημερινότητάς μας, την έχουν καταστήσει απαραίτητη για την σύγχρονη κοινωνία. Κάθε άνθρωπος, αυτή τη στιγμή, έχει πρόσβαση σε μια πληθώρα δεδομένων που μεταδίδονται μέσω του διαδικτύου από όλα τα σημεία του πλανήτη.

Η χρήση του Web μας δίνει τη δυνατότητα να ανακτήσουμε αυτά τα δεδομένα, να τα αποθηκεύσουμε και να τα επεξεργαστούμε. Πλέον μπορούμε να ενημερωνόμαστε, να πραγματοποιούμε αγορές, να ψυχαγωγούμαστε και να εκπαιδευόμαστε και όλα αυτά από το σπίτι μας με τη χρήση ενός υπολογιστή, μιας σύνδεσης Internet και μιας εφαρμογής για περιήγηση στον Παγκόσμιο Ιστό. Έτσι καθημερινά πολλοί άνθρωποι εκτελούν εργασίες με τη χρήση του Web, γλιτώνοντας κυρίως χρόνο και περιορίζοντας τις πιθανότητες σφάλματος.

## 3.1 Εργαλεία Ανάπτυξης

Για την της εφαρμογής μας είναι απαραίτητο να χρησιμοποιηθούν τα παρακάτω στοιχεία :

Ένα υπολογιστικό σύστημα με λειτουργικό σύστημα υποστήριξης εξυπηρετητή (server) :

- MS Windows
- Linux
- Mac OS X

Ένας εξυπηρετητής παγκόσμιου ιστού (web server) :

- IIS
- Apache

που να υποστηρίζει δυναμικές σελίδες, όπως :

- Active Server Pages (asp)
- Hypertext Preprocessor (php)
- Perl
- Java Server Pages (jsp)

Ένα Σχεσιακό Σύστημα διαχείρισης Βάσεων δεδομένων (RDBMS) :

- Apache Derby
- HSQLDB
- IBM DB2
- Microsoft SQL Server
- MSQL
- PostgreSQL
- SQLite

Για την υλοποίηση και ολοκλήρωση της εφαρμογής επιλέξαμε τα ακόλουθα στοιχεία και προσαρμόσαμε των κώδικά και τη λειτουργικότητα βάσει των ακολούθων στοιχείων:

- Υπολογιστικό σύστημα με λειτουργικό σύστημα Microsoft Windows 7 Ultimate.
- Εξυπηρετητής παγκόσμιου ιστού (web server) Apache 2.2.21, με την υποστήριξη δυναμικών σελίδων PHP 5.3.8
- Σχεσιακό Σύστημα διαχείρισης Βάσεων δεδομένων (RDBMS) MySQL 5.0.8

## **3.2 Apache HTTP Server**

Πρόκειται για έναν εξυπηρετητή (server) του παγκόσμιου ιστού (web). Με τον όρο server το μυαλό μας πηγαίνει ίσως σε ηλεκτρονικούς υπολογιστές που φιλοξενούν ιστοσελίδες και όχι άδικα. Ο όρος αυτός χρησιμοποιείται και για το μηχάνημα εξυπηρετητή (hardware) αλλά και για το πρόγραμμα (software). Στη συνέχεια θα ασχοληθούμε μόνο με το πρόγραμμα και συγκεκριμένα με τον Apache.

Ο Apache εγκαθίσταται σε έναν υπολογιστή ο οποίος μπορεί να χρησιμοποιεί διάφορα λειτουργικά συστήματα όπως Linux, Unix, Microsoft Windows, GNU, FreeBSD, Solaris, Novell NetWare, Mac OS X, OS/2, TPF.

Ο ρόλος του Apache είναι να αναμένει αιτήσεις από διάφορα προγράμματα – χρήστες (clients) όπως είναι ο φυλλομετρητής (browser) ενός χρήστη και στη συνέχεια, να εξυπηρετεί αυτές τις αιτήσεις "εξυπηρετώντας" τις σελίδες που ζητούν είτε απευθείας μέσω μιας ηλεκτρονικής διεύθυνσης (URL), είτε μέσω ενός συνδέσμου (link). Ο τρόπος με τον οποίο ο Apache εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνος με τα πρότυπα που ορίζει το πρωτόκολλο HTTP (Hypertext Transfer Protocol).

### 3.2.1 Το ξεκίνημα και η εξέλιξη του Apache

Η περίοδος έναρξης της δημιουργίας του προγράμματος χρονολογείται στις αρχές του 1990, όταν άρχισε να αναπτύσσεται από τον Robert McCool, ως ένα project του National Center for Supercomputing Applications (NCSA) με το όνομα HTTPd (HTTP daemon). Το 1994 ο Robert McCool αποχώρησε από το NCSA με αποτέλεσμα το NCSA HTTPd να μείνει σχεδόν εγκαταλειμμένο, πέρα από κάποιες διορθώσεις (patches) που ανέπτυσαν και διένειμαν εκτός από τον McCool και άλλοι προγραμματιστές. Το 1995 ανέλαβε το πρόγραμμα το Ίδρυμα Λογισμικού Apache (Apache Software Foundation), το οποίο διατηρεί την εποπτεία του έως και σήμερα.

Ο Apache HTTP αναπτύσσεται από την "Κοινότητα Ανοιχτού Λογισμικού" και η εποπτεία, υποστήριξη, και διάθεση του προγράμματος γίνεται από το Apache Software Foundation. Το πρόγραμμα είναι ανοιχτού κώδικα (open source), κάτι που σημαίνει ότι σύμφωνα με την άδεια χρήσης του (license), διατίθεται δωρεάν και μπορούν να γίνουν ελεύθερα από το χρήστη προσθήκες και τροποποιήσεις στον κώδικά του.

### 3.2.2 Η προέλευση του ονόματός του

Υπάρχουν δύο εκδοχές σχετικά με την προέλευση του ονόματός του. Η πρώτη εντοπίζεται στα πρώτα χρόνια της δημιουργίας του, τότε που ως NCSA HTTPd έπρεπε να αναπτύσσονται συνεχώς διορθώσεις (patches) για να ενσωματωθούν στον αρχικό του κώδικα με αποτέλεσμα να του δοθεί το όνομα a patchy server. Η δεύτερη εκδοχή σύμφωνα με το ίδρυμα Apache, αναφέρει ότι το όνομα αυτό δόθηκε προς τιμήν των ιθαγενών Ινδιάνων της Αμερικής και συμβολίζει το μαχητικό πνεύμα και την αντοχή.

### 3.2.3 Γιατί Apache HTTP;

Ο Apache διαθέτει ποικιλία χαρακτηριστικών και μπορεί να υποστηρίξει μια μεγάλη γκάμα εφαρμογών με τις οποίες και συνεργάζεται. Οι δυνατότητες του προγράμματος αυτού καθαυτού και τα χαρακτηριστικά του δεν είναι και τόσο πολλά. Ένα από τα βασικότερα χαρακτηριστικά του όμως, το οποίο και του δίνει μεγάλες δυνατότητες, είναι ότι μπορεί να προσαρμόσει επάνω του πολλές προσθήκες προγραμμάτων (modules), τα οποία με τη σειρά τους παρέχουν διαφορετικές λειτουργίες.

Μερικά από τα πιο γνωστά modules του Apache HTTP είναι τα modules πιστοποίησης, όπως για παράδειγμα τα mod\_access, mod\_auth, mod\_digest. Παρέχει επίσης SSL σε TLS μέσω των (mod\_ssl) και proxy module (mod\_proxy), πραγματοποιεί ανακατευθύνσεις διευθύνσεων (URL rewrites) μέσω του mod\_rewrite, καταγραφές συνδέσεων μέσω του mod\_log\_config, συμπίεση αρχείων μέσω του mod\_gzip και πολλά άλλα modules τα οποία διατίθενται είτε από το Apache Software Foundation, είτε από τρίτες εταιρίες λογισμικού.

Ένα άλλο χαρακτηριστικό - δυνατότητα του Apache HTTP, είναι ότι μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα. Ο Apache HTTP υποστηρίζει επίσης, αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως MySQL, PHP, Perl και Python. Αυτά είναι μερικά από τα χαρακτηριστικά και τις λειτουργίες του που κάνουν τον Apache τον πιο δημοφιλή Web Server από το 1996 έως τις μέρες μας.

Περισσότερο από το 50% των ιστοχώρων του παγκόσμιου ιστού χρησιμοποιεί τον Apache ως εξυπηρετητή. Το υπόλοιπο ποσοστό καλύπτουν αντίστοιχα προγράμματα, όπως το Microsoft Internet Information Services (IIS), ο Sun Java System Web Server, ο Zeus Web Server. Τον Μάιο του 2011 το μερίδιο του ήταν το 63% από όλους τους ιστοτόπους.

### **3.3 PHP**

Η PHP, της οποίας τα αρχικά αντιπροσωπεύουν το "PHP: Hypertext Preprocessor" είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το web και μπορεί να ενσωματωθεί στην HTML. Η σύνταξή της παίρνει στοιχεία των C, Java, και Perl και είναι εύκολη στη μάθηση. Ο κύριος στόχος της γλώσσας είναι να επιτρέπει σε web developers να γράφουν δυναμικά παραγόμενες σελίδες (webpages) γρήγορα, αλλά μπορεί κανείς να κάνει πολύ περισσότερα με την PHP.

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από έναν συμβατό διακομιστή του Παγκόσμιου Ιστού (όπως Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (όπως \*.php, \*.php4, \*.phtml). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δε θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server.

Επίσης, ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux και Microsoft Windows, υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP.

### 3.3.1 Η ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα rhp.cgi, για προσωπική χρήση. Το script αυτό είχε σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα, αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter.

Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι, η PHP έφθασε στην έκδοση 3.0 η οποία θυμίζει περισσότερο τη σημερινή μορφή της.

Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

### 3.3.2 Γιατί PHP;

Σε σύγκριση με τους ανταγωνιστές της, η PHP έχει πολλά πλεονεκτήματα, όπως :

**Υψηλή απόδοση** : Η PHP είναι πολύ αποτελεσματική. Με ένα φθινό διακομιστή μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά. Οι δόκιμες που δημοσιεύθηκαν από τη Zend technologies, δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

**Διασύνδεση με πολλά διαφορετικά συστήματα βάσεων δεδομένων** : Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων.

**Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες** : Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web, έχει πολλές ενσωματωμένες βιβλιοθήκες, που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web. Είναι δυνατόν να δημιουργηθούν εικόνες GIF δυναμικά, να γίνει σύνδεση με άλλες υπηρεσίες δικτύων, να σταλεί ηλεκτρονικό ταχυδρομείο, να γίνει εργασία με cookies και να δημιουργηθούν PDF έγγραφα, όλα αυτά με λίγες γραμμές κώδικα.

**Χαμηλό κόστος** : Η PHP είναι δωρεάν. Η τελευταία της έκδοση διανέμεται από το <http://www.php.net>, χωρίς χρέωση.

**Ευκολία μάθησης και χρήσης :** Η σύνταξη της PHP βασίζεται σε άλλες γλώσσες προγραμματισμού, βασικά στην C και στην Perl. Οι γνώστες C ή Perl, ή μιας γλώσσας παρόμοιας με την C, όπως την C++ ή την java, αρχίζουν να προγραμματίζουν με την PHP σχεδόν αμέσως.

**Μεταφερσιμότητα :** Η PHP είναι διαθέσιμη για πολλά λειτουργικά συστήματα. Μπορεί να γραφτεί κώδικας PHP για δωρεάν συστήματα τύπου Unix, όπως στο Linux και στο FreeBSD, για εμπορικές εκδόσεις του Unix, όπως το Solaris και το IRIX, ή για διαφορετικές εκδόσεις των Microsoft Windows. Ο κώδικας συνήθως δουλεύει χωρίς αλλαγές στα συστήματα που τρέχουν την PHP.

**Διαθεσιμότητα του κώδικα προέλευσης :** Υπάρχει πρόσβαση στον κώδικα προέλευσης της PHP. Αντίθετα με εμπορικά, κλειστά προϊόντα, αν υπάρχει κάτι που πρέπει να αλλαχθεί ή να προστεθεί στην γλώσσα, μπορεί να γίνει. Δεν χρειάζεται να αναμένεται ο κατασκευαστής να εμφανίσει διορθώσεις. Δεν υπάρχει ανησυχία για το αν ο κατασκευαστής σταματήσει να υπάρχει ή αν θα σταματήσει να υποστηρίζει το προϊόν.

## 3.4 SQL

Η SQL (Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS, Relational Database Management System) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα.

Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα.

Η SQL ήταν μία από τις πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του το 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων.

### 3.4.1 Ιστορία

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

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



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

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

### 3.4.2 Γιατί MySQL;

Σε σύγκριση με τους ανταγωνιστές της, η MySQL έχει πολλά πλεονεκτήματα, όπως :

**Απόδοση** : Η MySQL είναι χωρίς αμφιβολία γρήγορη. Πολλές δοκιμές έδειξαν ότι η MySQL είναι πολύ πιο γρήγορη από τον ανταγωνισμό.

**Χαμηλό κόστος** : Η MySQL είναι διαθέσιμη δωρεάν, με άδεια ανοικτού κώδικα (open source), ή με χαμηλό κόστος για εμπορική άδεια αν απαιτείται από την εφαρμογή.

**Ευκολία χρήσης** : Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Οι χρήστες άλλων συστημάτων μπορούν εύκολα να προσαρμοστούν. Επίσης η MySQL είναι ευκολότερη από παρόμοια προϊόντα.

**Μεταφερσιμότητα** : Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα UNIX, όπως επίσης και στα Microsoft Windows.

**Διαθεσιμότητα του κώδικα προέλευσης** : Όπως και με την PHP, είναι εύκολο να τροποποιηθεί ο κώδικας προέλευσης της MySQL.

### 3.5 XML

Σε ένα κόσμο όπου οι πληροφορίες παρέχονται μέσω του παγκόσμιου διαδικτύου, τα έγγραφα πρέπει να είναι εύκολα προσβάσιμα, μεταφέρσιμα και ευέλικτα. Πρέπει επίσης να είναι ανεξάρτητα οποιουδήποτε συστήματος και περιεχομένου. Οι γενικευμένες γλώσσες έχουν τέτοια χαρακτηριστικά, παρέχοντας στα έγγραφα αυτά μια δυνατότητα η οποία δεν υπάρχει σε άλλες γλώσσες περιγραφής εγγράφων. Η HTML είναι προβληματική και περιοριστική γλώσσα. Η XML έλυσε πολλά από τα προβλήματα που αντιμετώπισαν οι σχεδιαστές του web και είναι υπεύθυνη για την XHTML, μια ανασχεδιασμένη HTML. Θα χρησιμοποιείται για πολλά χρόνια επειδή προσφέρει αποτελεσματικές και δυναμικές πολυμεσικές λύσεις.

Η XML σχεδιάστηκε να ικανοποιήσει πολλές ανάγκες δίνοντας στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στο στυλ και τη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Η XML προσφέρει στους σχεδιαστές της HTML τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα. Δεν αναφέρεται μονάχα στους σχεδιαστές του web αλλά σε οποιονδήποτε ασχολείται με εκδόσεις.

Στην πραγματικότητα, η XML είναι markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Markup γλώσσα είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο. σχεδόν όλα τα έγγραφα έχουν την ίδια δομή.

Η XML είναι κάτι περισσότερο από markup language είναι metalanguage, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες markup γλώσσες. Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων η XML αναπαριστά τη συναφή έννοια των δεδομένων. Στην HTML τα tags είναι προκαθορισμένα ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις.

Τα XML έγγραφα δεν είναι πολύπλοκα αλλά απλά και πολύ αποτελεσματικά. Το διδακτικό υλικό της well-formed XML αναλύει τη δημιουργία των XML εγγράφων, η οποία είναι κατά κάποιο τρόπο ίδια με την HTML καθώς επιτρέπει τη μη δομημένη δημιουργία εγγράφου. Η valid XML είναι πιο σύνθετη. Απαιτεί την ύπαρξη ενός Document Type Definition πριν να γραφεί το έγγραφο αλλά παρέχει μια γενική δομή με βάση την οποία τη δημιουργούμε.

Η γλώσσα προγραμματισμού XML περιγράφει μια κατηγορία πληροφοριών (data objects) που καλούνται XML έγγραφα (documents) καθώς επίσης περιγράφει τμηματικά τη συμπεριφορά των προγραμμάτων που τα επεξεργάζονται.

Τα XML έγγραφα αποτελούνται από μονάδες αποθήκευσης που καλούνται *entities* (οντότητες), οι οποίες περιέχουν πληροφορίες αναλυμένες ή μη. Οι αναλυμένες πληροφορίες αποτελούνται από *χαρακτήρες* (*characters*) οι οποίοι συνθέτουν *character data* και άλλοι οι οποίοι συνθέτουν *markup*. Η μορφή markup κωδικοποιεί την περιγραφή της τελικής αποθήκευσης του εγγράφου καθώς και τη λογική δομή.

Ένα λογισμικό μοντέλο που καλείται **επεξεργαστής XML** χρησιμοποιείται να διαβάσει XML έγγραφα και παρέχει πρόσβαση στο περιεχόμενο και τη δομή τους. Υποτίθεται ότι ο επεξεργαστής XML λειτουργεί εκ μέρους ενός άλλου μοντέλου που καλείται **application (εφαρμογή)**. Αυτή η προδιαγραφή περιγράφει την απαιτούμενη συμπεριφορά του επεξεργαστή και συγκεκριμένα πως θα πρέπει να διαβάσει τα XML δεδομένα και ποιες πληροφορίες πρέπει να παρέχει στην εφαρμογή.

### 3.5.1 Προέλευση και στόχοι (Origin and goals)

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

Οι προσχεδιασμένοι στόχοι της XML είναι:

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

## 3.5.2 Εισαγωγή στο XML DOM

Το Μοντέλο Αντικειμένου Εγγράφου της XML (*XML Document Object Model, DOM*) είναι μια προγραμματιστική διεπαφή (programming interface) για τα XML έγγραφα. Ορίζει τον τρόπο που μπορούμε να προσπελάσουμε και να χειριστούμε ένα XML έγγραφο.

Σαν μια προδιαγραφή (specification) του W3C, ο αντικειμενικός στόχος του XML DOM είναι να παρέχει μια στάνταρτ προγραμματιστική διεπαφή σε μια ευρεία ποικιλία εφαρμογών. Το XML DOM είναι σχεδιασμένο για να μπορεί να χρησιμοποιηθεί με οποιαδήποτε γλώσσα προγραμματισμού και με οποιοδήποτε λειτουργικό σύστημα.

Με το XML DOM μπορούμε να δημιουργήσουμε ένα XML έγγραφο, να πλοηγηθούμε στη δομή του και να προσθέσουμε, τροποποιήσουμε ή διαγράψουμε τα στοιχεία του.

### 3.5.2.1 To Node Interface

Όπως θα δούμε και παρακάτω, ένα πρόγραμμα που αποκαλείται XML parser μπορεί να χρησιμοποιηθεί για να φορτώσουμε ένα XML έγγραφο στη μνήμη του υπολογιστή μας. Αφού φορτωθεί το έγγραφο, τα στοιχεία του μπορούν να ανακτηθούν και να διαχειριστούν με πρόσβαση στο DOM. Το DOM παριστάνει μια δενδροειδή άποψη (tree view) του XML εγγράφου. Το *documentElement* βρίσκεται στην κορυφή (top-level) του δένδρου και αυτό το στοιχείο περιέχει ένα ή περισσότερα *childNodes* που παριστάνουν τα κλαδιά του δένδρου.

Το Node Interface Model χρησιμοποιείται για να έχουμε πρόσβαση στα ανεξάρτητα στοιχεία του δένδρου. Για παράδειγμα, η ιδιότητα *childNodes* του *documentElement* μπορεί να προσπελαστεί με μια δομή (construct) *for/each* για να μπορέσουμε να απαριθμήσουμε τον κάθε ανεξάρτητο κόμβο (node). Το XML parser της Microsoft υποστηρίζει όλες τις συναρτήσεις που είναι απαραίτητες για να μπορέσουμε να διασχίσουμε το δένδρο των κόμβων (node tree), να έχουμε πρόσβαση στους κόμβους και τις τιμές των χαρακτηριστικών τους (attribute values), να εισάγουμε και να διαγράψουμε κόμβους και να μετατρέψουμε το node tree ξανά σε XML.

Οι περισσότερες από τις συναρτήσεις parser που παρουσιάζονται εδώ προέρχονται από τη σύσταση (recommendation) XML DOM του W3C, εκτός από τις συναρτήσεις load και error του document. Πρέπει να έχουμε υπόψη μας ότι το επίσημο DOM δεν περιλαμβάνει σπάνια συναρτήσεις για το φόρτωμα των XML εγγράφων. Ο επόμενος πίνακας παρουσιάζει τους πιο συχνά χρησιμοποιούμενους τύπους κόμβων που υποστηρίζονται από το Microsoft XML parser :

<b>Είδος Κόμβου</b>	<b>Παράδειγμα</b>
Document type	<!DOCTYPE food SYSTEM "food.dtd">
Εντολή επεξεργασίας	<?xml version="1.0"?>
Στοιχείο (Element)	<drink type="beer"> Carlsberg </drink>
Χαρακτηριστικό (Attribute)	type="beer"
Κείμενο (Text)	Carlsberg

### 3.5.2.1 Αναλύοντας (Parsing) το DOM

Το XML parser της Microsoft είναι ένα συστατικό COM που δίνεται μαζί με τον Microsoft Internet Explorer 5.0 και είναι διαθέσιμο σε scripts που βρίσκονται μέσα σε HTML έγγραφα και ASP αρχεία. Για να διαβάσουμε, δημιουργήσουμε και διαχειριστούμε ένα XML έγγραφο, χρειαζόμαστε έναν XML parser. Το Microsoft XMLDOM parser περιλαμβάνει ένα μοντέλο προγραμματισμού το οποίο :

- Υποστηρίζει PHP, JavaScript, VBScript, Perl, VB, Java, C++ κ.ά.
- Υποστηρίζει W3C XML και XML DOM.
- Υποστηρίζει DTD και validation.

## 3.6 JavaScript

### Τι είναι JavaScript?

- Η javascript είναι μια scripting γλώσσα η οποία έχει σχεδιαστεί και χρησιμοποιείται για να εισάγουμε την έννοια της διαδραστικότητας στις html σελίδες.
- Είναι μια ερμηνευτική γλώσσα, δηλαδή το script εκτελείτε χωρίς να έχει περάσει από την διαδικασία της σύνταξης.
- Η javascript και η java είναι δύο εντελώς διαφορετικές γλώσσες.

### Τι μπορούμε να κάνουμε με την javascript?

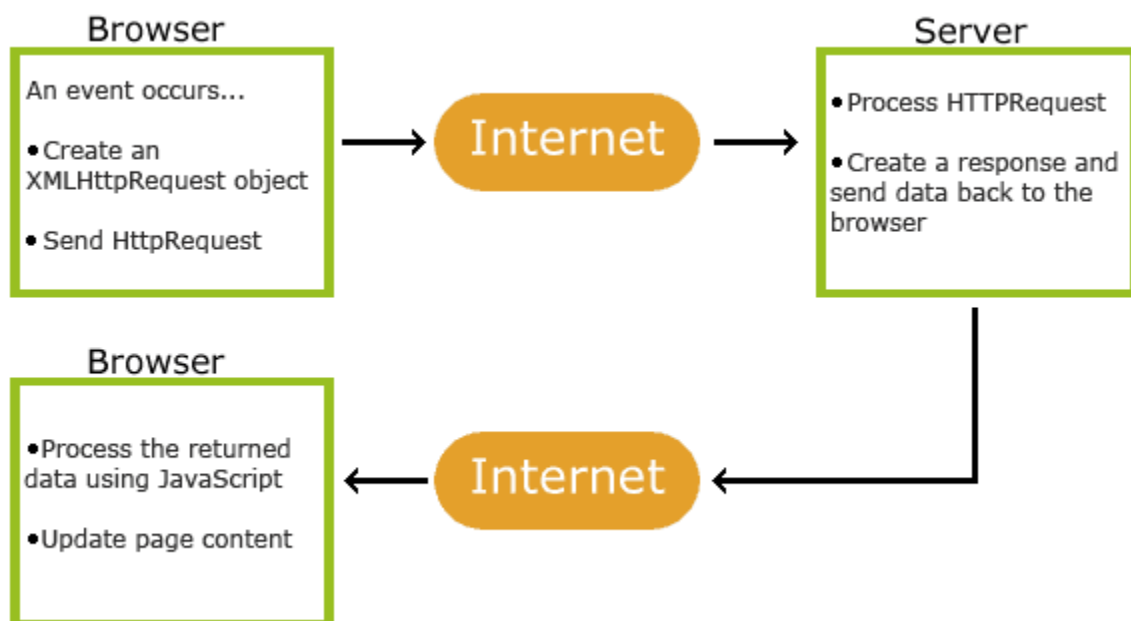
- Με την javascript μπορούμε να εκτελέσουμε κάποια πράγματα όταν συμβαίνει ένα γεγονός, για παράδειγμα όταν ο χρήστης κάνει κλικ σε ένα html στοιχείο, να εκτελείται κάποιο script και λαμβάνουμε τα αντίστοιχα αποτελέσματα.
- Η javascript μπορεί να διαβάσει και να αλλάξει τα περιεχόμενα ενός html στοιχείου.
- Μπορούμε να την χρησιμοποιήσουμε για να επικυρώσουμε δεδομένα μιας φόρμας(validate) προτού να υποβληθούν στον server (γλυτώνουμε και τον server από επιπλέον έλεγχο).
- Με την βοήθεια της javascript μπορούμε να εντοπίσουμε τον browser του επισκέπτη και ανάλογα με τον browser να φορτώσουμε την αντίστοιχη σελίδα που είναι φτιαγμένη για αυτόν τον browser.
- Μπορούμε να δημιουργήσουμε cookies (να αποθηκεύουμε και να λαμβάνουμε πληροφορίες στον υπολογιστή του επισκέπτη).

### 3.7 AJAX

Η AJAX δεν είναι γλώσσα προγραμματισμού όπως η JavaScript, αλλά ούτε και γλώσσα χαρακτηρισμού κειμένου όπως η HTML. Δεν θεωρείται καν γλώσσα. AJAX σημαίνει Asynchronous Javascript And XML και είναι η τεχνική με την οποία μπορούμε να δημιουργήσουμε πολύ πιο γρήγορες και δυναμικές ιστοσελίδες περιορίζοντας τον όγκο δεδομένων που ανταλλάσσει ο server με τον browser του επισκέπτη. Η ιδιαίτερη τεχνική αυτή επιτρέπει την ανανέωση περιεχομένων μιας ιστοσελίδας χωρίς αυτή να ανανεωθεί ολόκληρη.

Η υλοποίηση της τεχνικής AJAX γίνεται με τον συνδυασμό του αντικειμένου (object) XMLHttpRequest (πραγματοποιεί την ασύγχρονη επικοινωνία με τον server), την Javascript/DOM (αλληλεπιδρά με τα δεδομένα και τα εμφανίζει), την CSS (μορφοποιεί τα προς εμφάνιση δεδομένα) και την XML (συχνά χρησιμοποιείται για την μεταφορά δεδομένων).

Το παρακάτω σχεδιάγραμμα είναι από την σελίδα [w3schools.com](http://w3schools.com)





# Κεφάλαιο 4. XAMPP

## 4.1 Τι είναι XAMPP

Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξάρτητου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση ιστοσελίδων MySQL και ένα διερμηνέα για σενάρια γραμμένα σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- X (αναφέρεται στο "cross-platform" που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- Apache HTTP εξυπηρετητής
- MySQL
- PHP
- Perl

Το XAMPP είναι ένα ελεύθερο λογισμικό το οποίο περιέχει ένα εξυπηρετητή ιστοσελίδων το οποίο μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με την τεχνολογίες όπως PHP, JSP και Servlets.

### 4.1.1 Δυνατότητες και απαιτήσεις

Το XAMPP προϋποθέτει μόνο τα λογισμικά συμπίεσης αρχείων zip, tar, 7z ή exe κατά την διάρκεια της εγκατάστασης. Το XAMPP έχει δυνατότητα αναβάθμισης σε νέες εκδόσεις του εξυπηρετητή ιστοσελίδων http Apache, της βάσης δεδομένων MySQL, της γλώσσας PHP και Perl. Το XAMPP συμπεριλαμβάνει επίσης τα πακέτα OpenSSL και το phpMyAdmin.

### 4.1.2 Χρήση

Επίσημα οι σχεδιαστές του XAMPP προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Για να είναι δυνατή η χρήση του, πολλές σημαντικές λειτουργίες ασφάλειας έχουν απενεργοποιηθεί.

Στην πράξη το XAMPP ορισμένες φορές χρησιμοποιείται και για την φιλοξενία ιστοσελίδων. Υπάρχει ειδικό εργαλείο το οποίο περιέχεται στο XAMPP για την προστασία με κωδικό των σημαντικών μερών. Το XAMPP υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων τύπου MySQL και SQLite.

Όταν το XAMPP εγκατασταθεί στον τοπικό υπολογιστή διαχειρίζεται τον localhost ως ένα απομακρυσμένο κόμβο, ο οποίος συνδέεται με το πρωτόκολλο μεταφοράς αρχείων FTP. Η σύνδεση στον localhost μέσω του FTP μπορεί να γίνει με το όνομα χρήστη "newuser" και το κωδικό "wampp".

Για την βάση δεδομένων MySQL ο υπάρχει ο χρήστης "root" χωρίς κωδικό πρόσβασης.

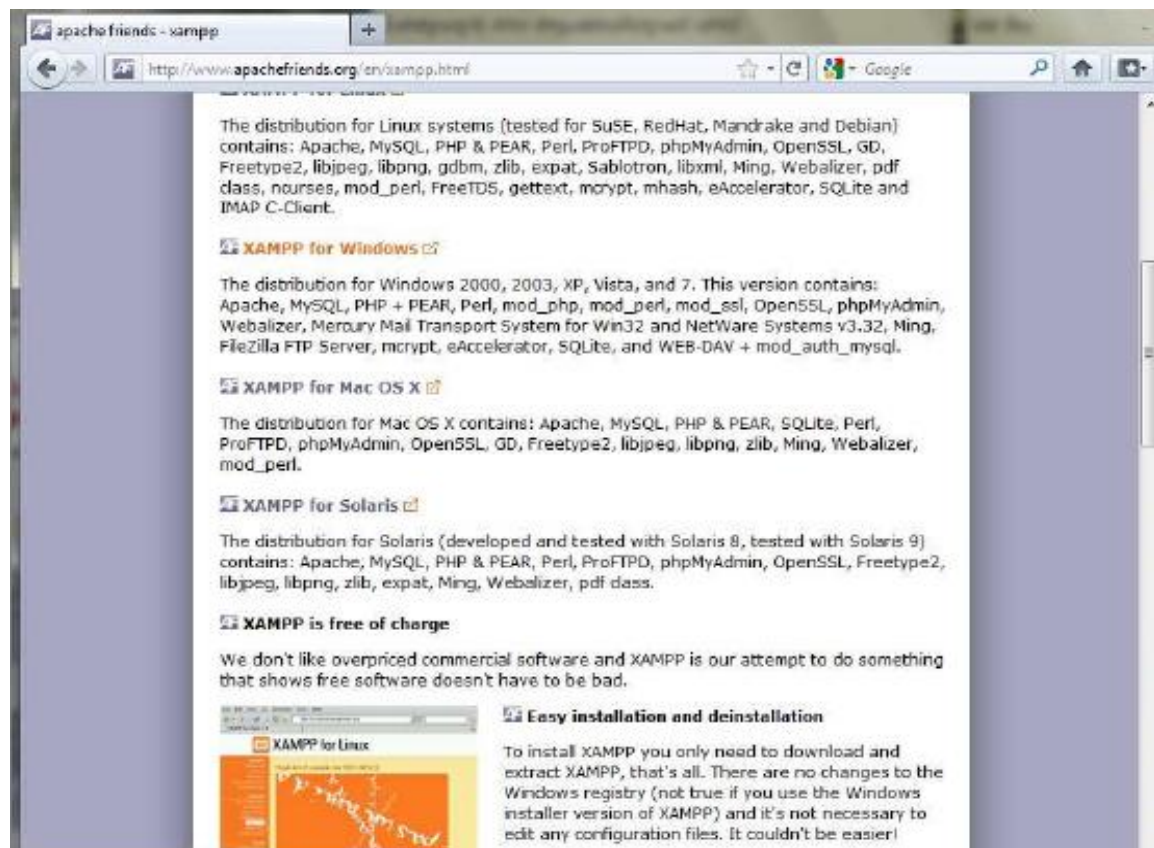
## 4.2 Εγκατάσταση Λογισμικού XAMPP

Για να κατεβάσουμε και να εγκαταστήσουμε το XAMPP στον υπολογιστή μας κατευθυνόμαστε στην επίσημη ιστοσελίδα

<http://www.apachefriends.org/en/index.html>

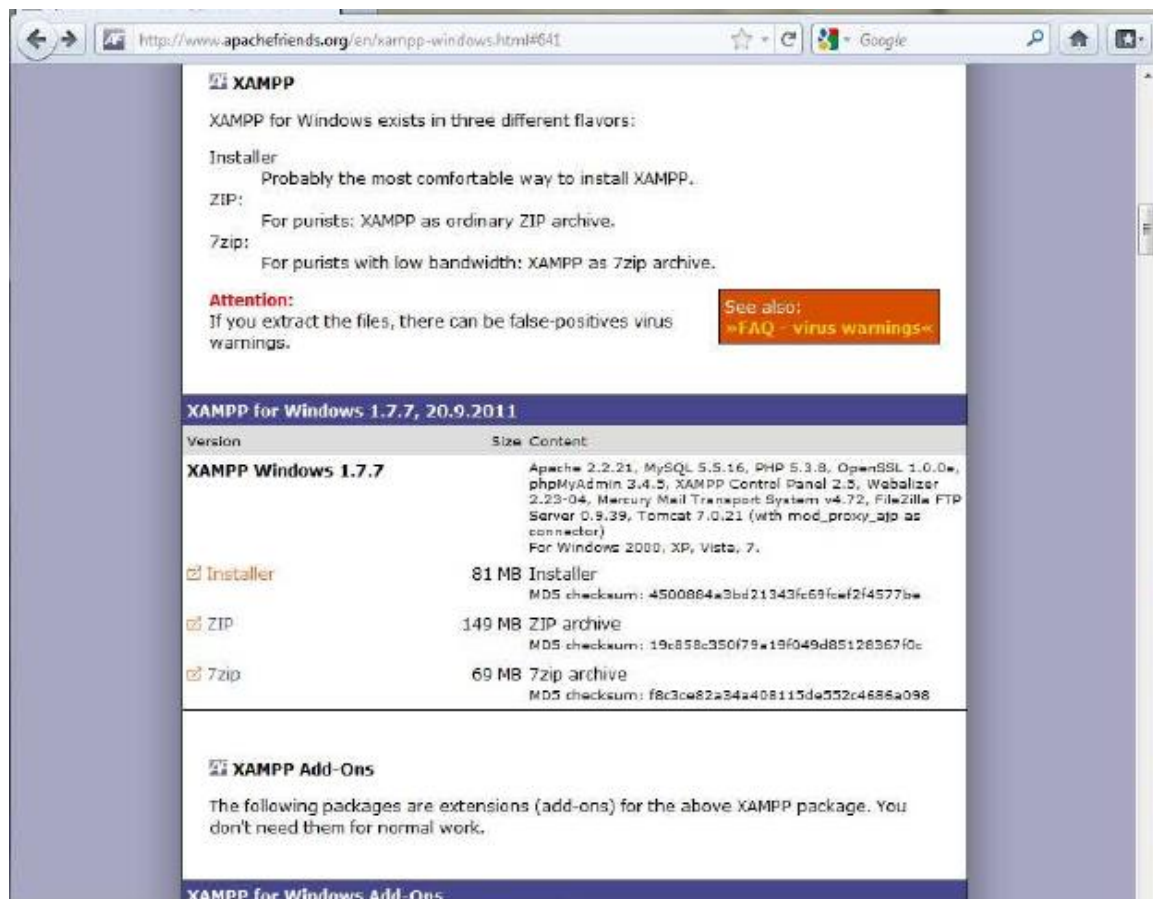


Εικόνα 4.2.1. Αρχική σελίδα [www.apachefriends.org](http://www.apachefriends.org)

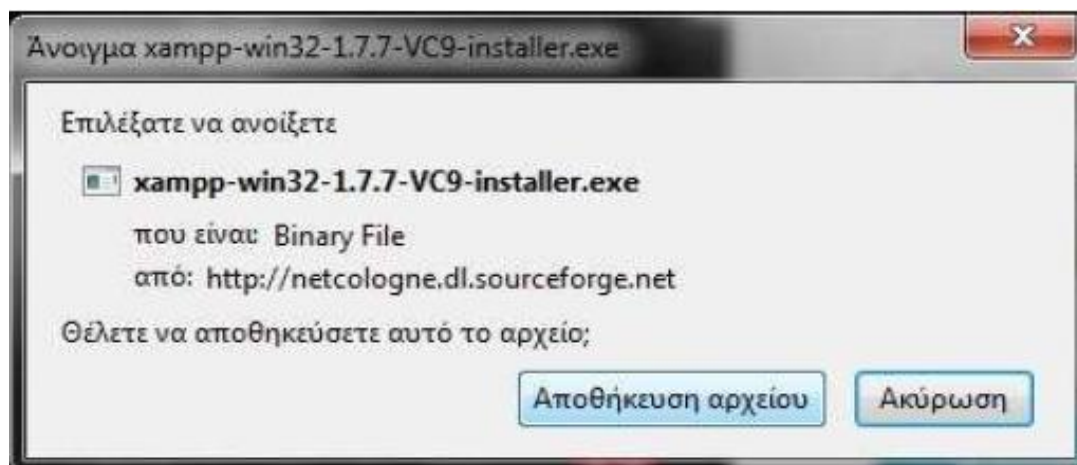


Εικόνα 4.2.2 Σελίδα επιλογής λειτουργικών συστημάτων για XAMPP

Στην συνέχεια επιλέγουμε την έκδοση «XAMPP for Windows»



Εικόνα 4.2.3. Επιλογή Installer δηλαδή το αρχείο .exe



Εικόνα 4.2.4. Παράθυρο λήψης αρχείου εγκατάστασης XAMPP

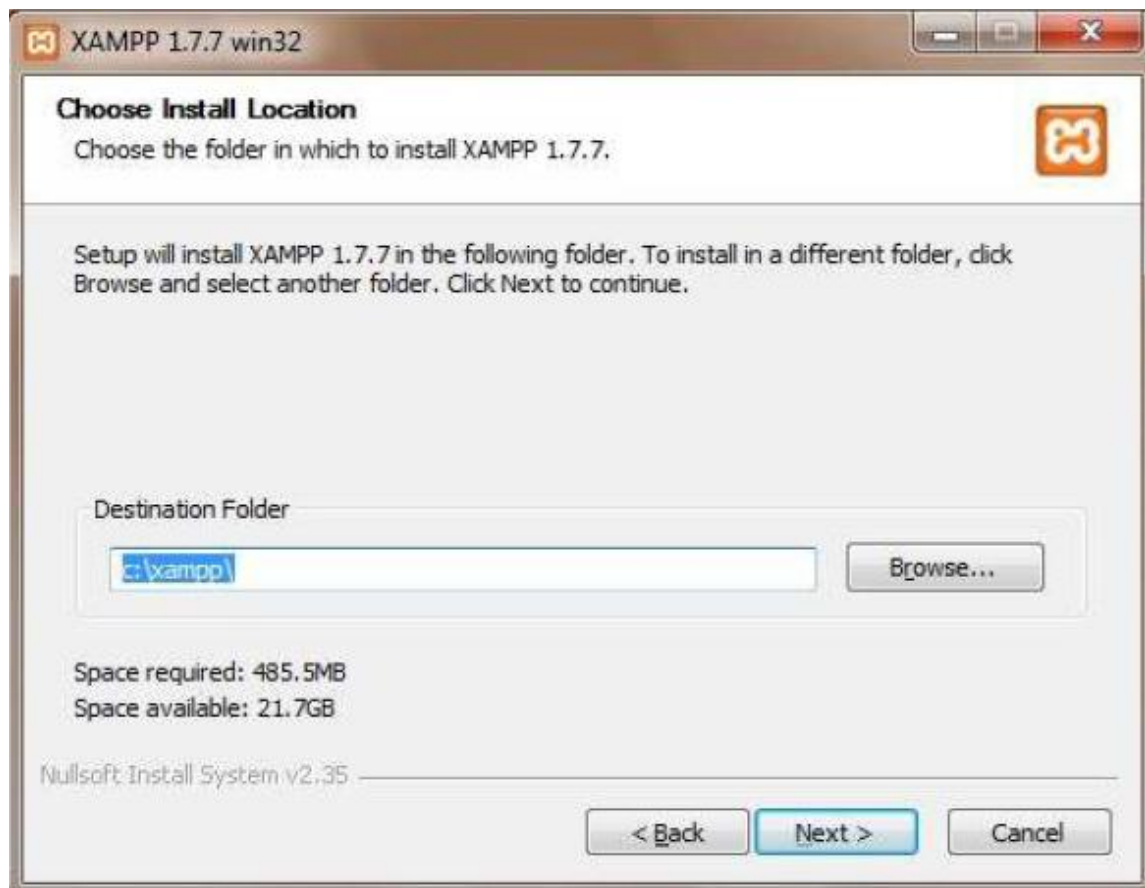
Επιλέγουμε «Αποθήκευση αρχείου»

Ξεκινάμε την εγκατάσταση:



Εικόνα 4.2.5. Εκκίνηση εγκατάστασης XAMPP

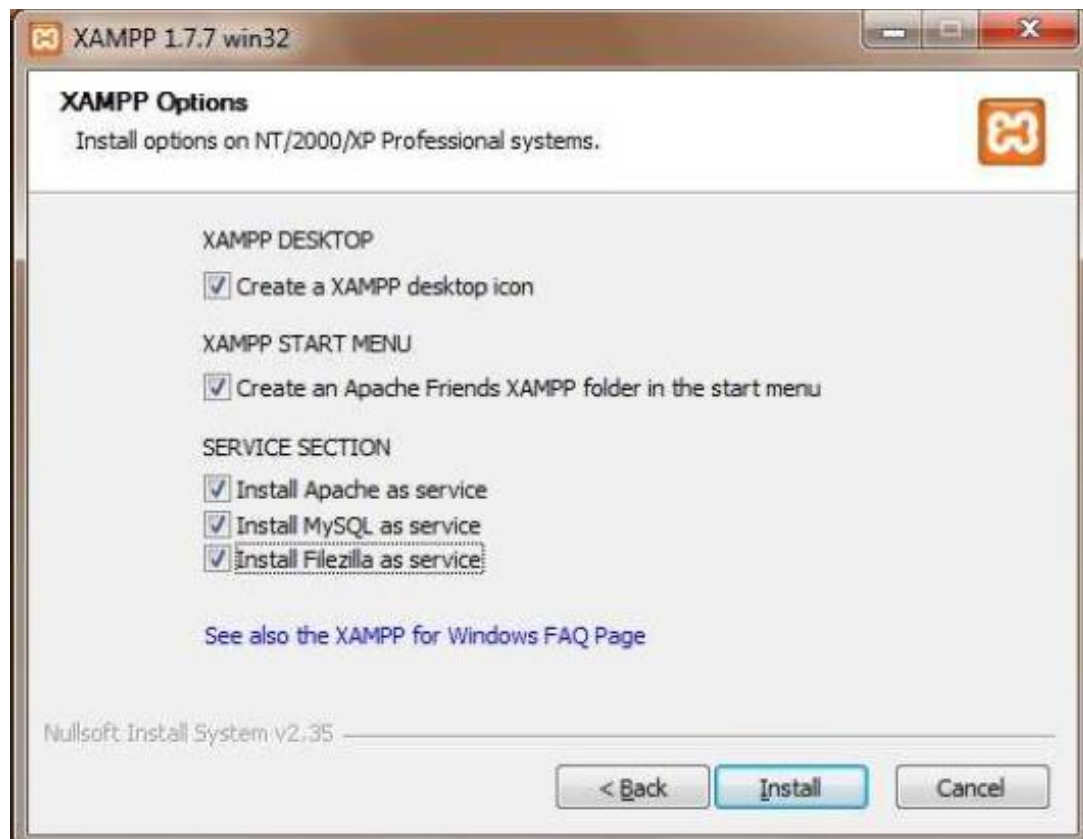
Μόλις ανοίξει το παράθυρο της εγκατάστασης επιλέγουμε «**Next**» για συνέχεια



Εικόνα 4.2.6. Επιλογή φακέλου εγκατάστασης στο δίσκο

Επιλέγουμε σε πιο μέρος του δίσκου θα το εγκαταστήσουμε και συνεχίζουμε την εγκατάσταση επιλέγοντας «**Next**»



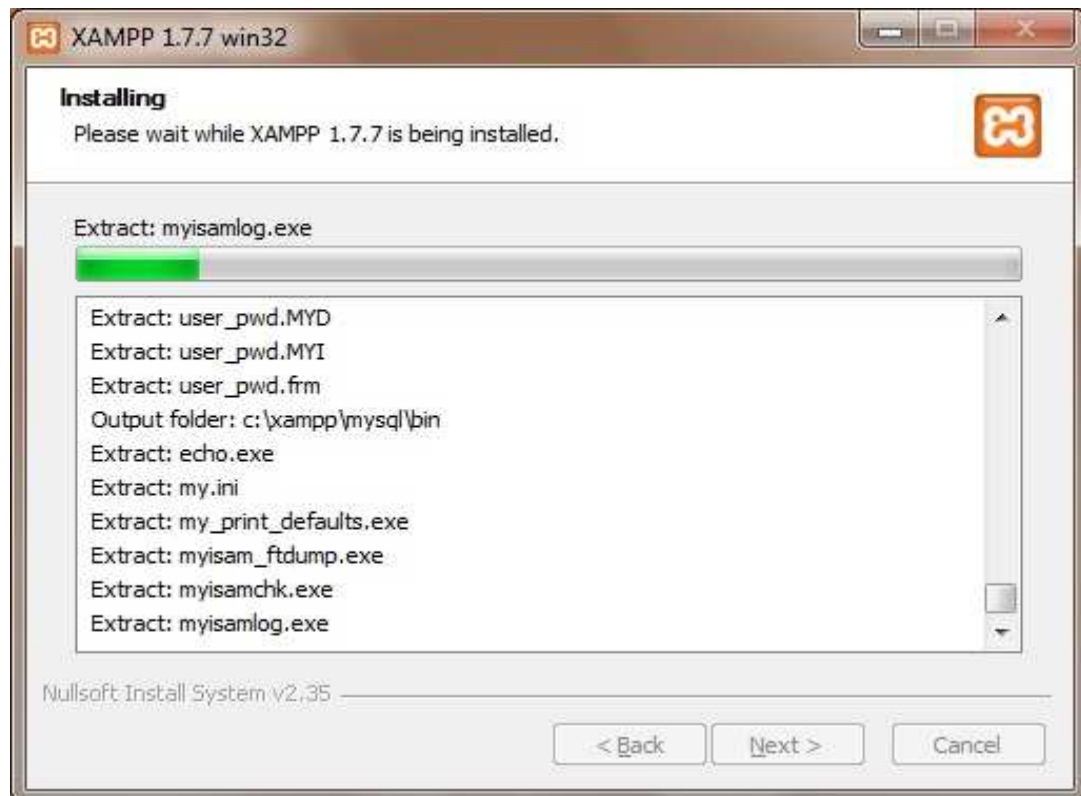


Εικόνα 4.2.7. Επιλογές XAMPP

Τσεκάρουμε τις επιλογές και επιλέγουμε «Install»



Η εγκατάσταση ξεκίνησε και παρατηρούμε την μπάρα προόδου

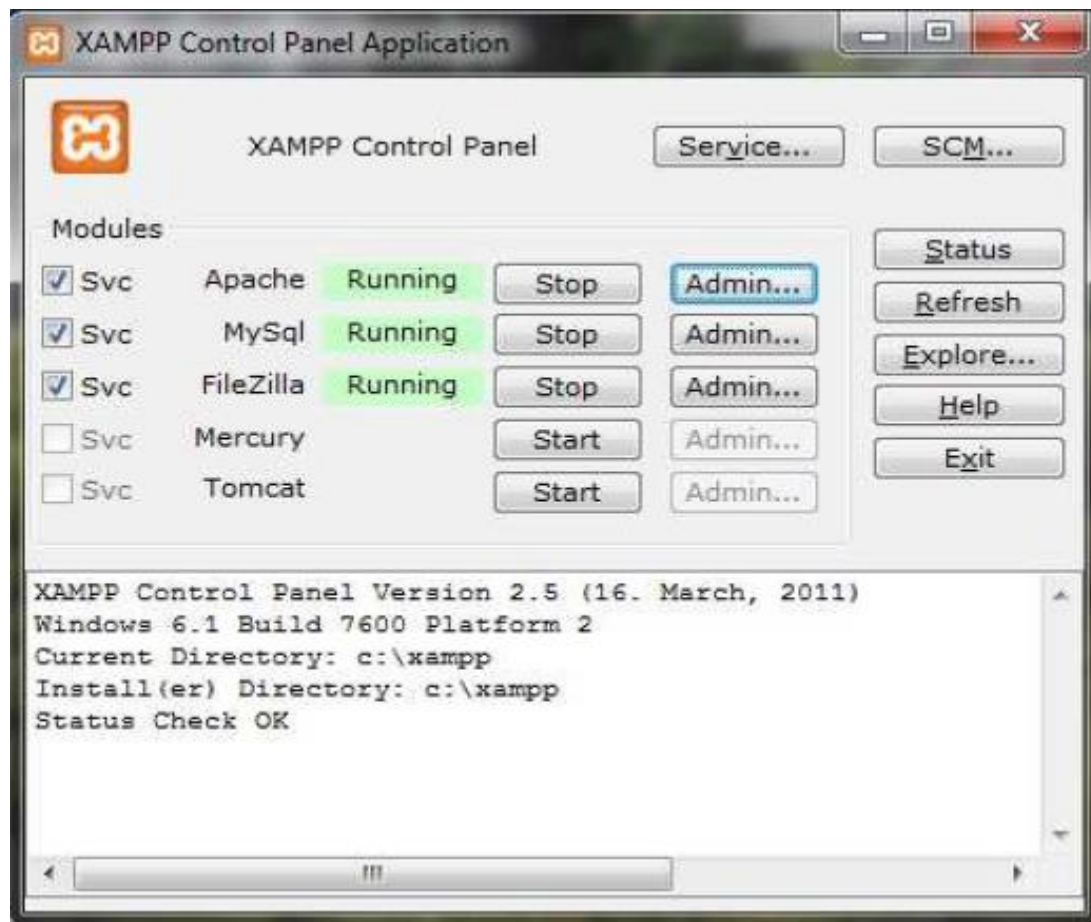


Εικόνα 4.2.8. Εκκίνηση εγκατάστασης



Εικόνα 4.2.9. Ολοκλήρωση εγκατάστασης

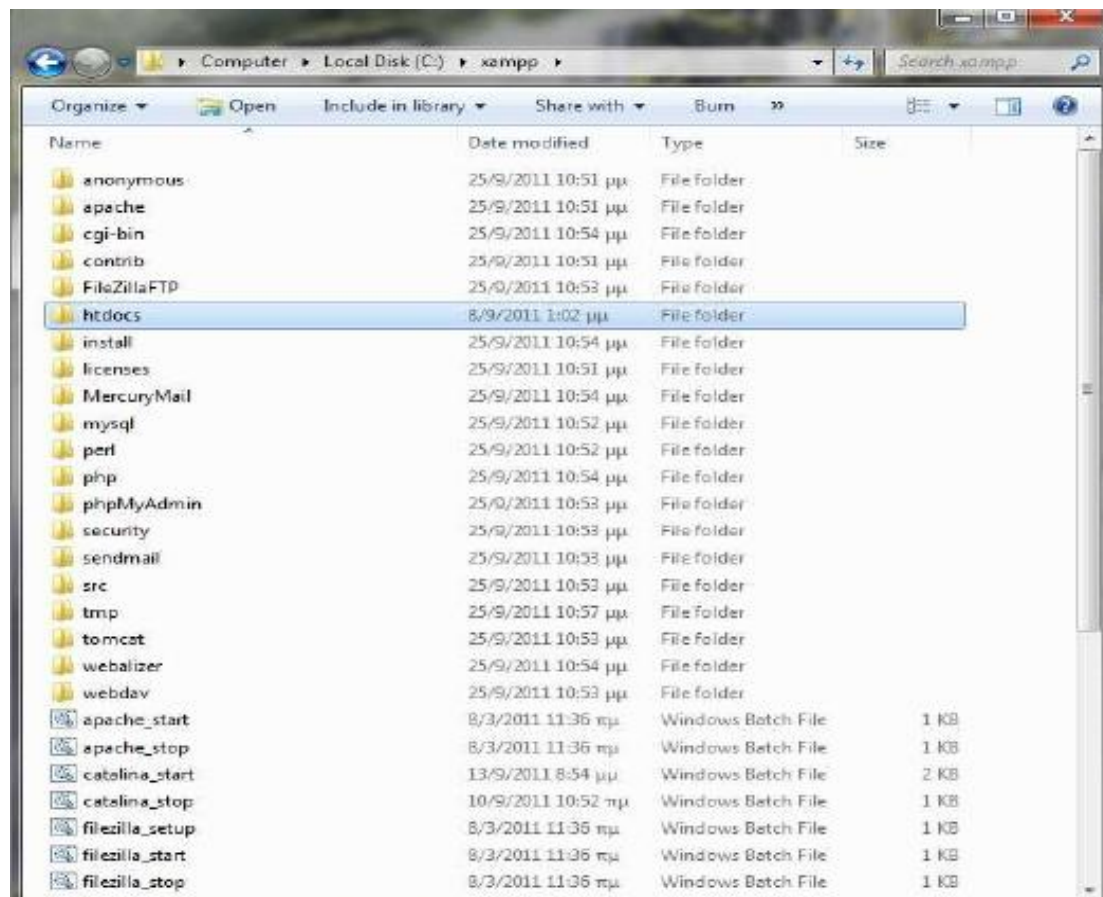
Η εγκατάσταση έχει ολοκληρωθεί επιλέγουμε το «**Finish**» και ξεκινάμε το λογισμικό



Εικόνα 4.2.10. XAMPP Control Panel Application

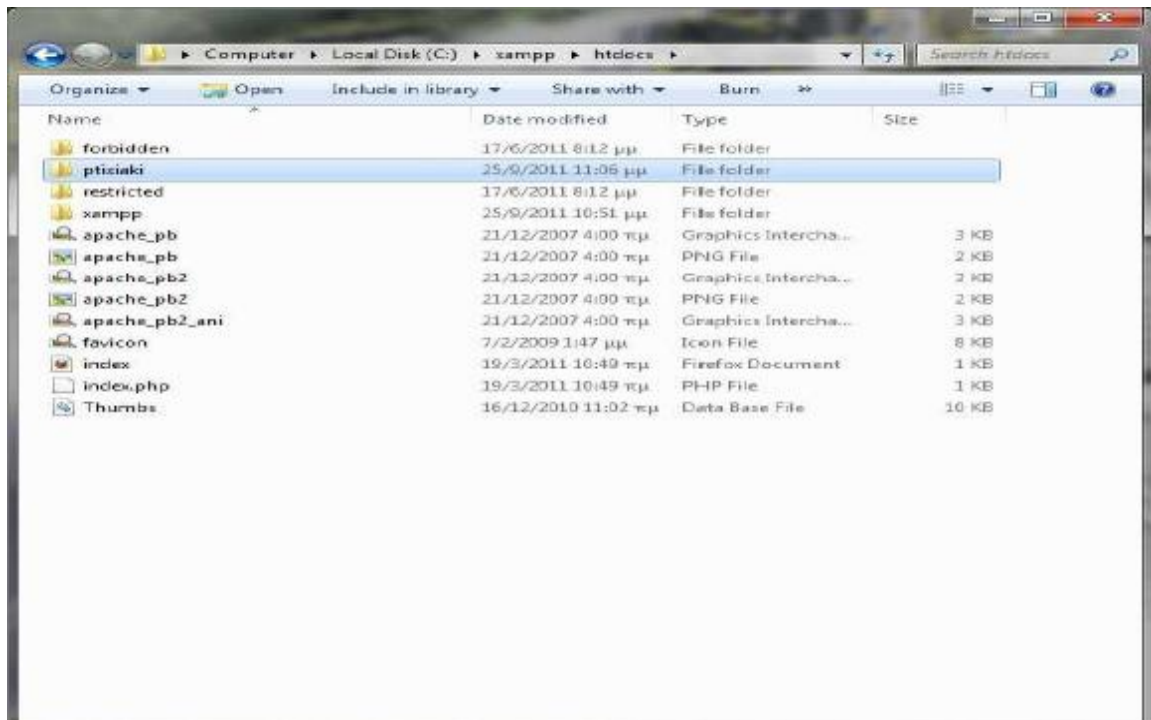
Από το «**XAMPP Control Panel Application**» επιλέγουμε ποιες διεργασίες θα ξεκινήσουν.

Στην συνέχεια πηγαίνουμε στα αρχεία του XAMPP και ανοίγουμε τον φάκελο htdocs όπου θα αποθηκεύουμε τα αρχεία της εφαρμογής που θα δημιουργήσουμε και από αυτόν τον φάκελο του server μας θα εμφανίζονται στον φυλλομετρητή (browser) μας.



Εικόνα 4.2.11. Αρχεία και φάκελοι XAMPP

Βλέπουμε τα αρχεία και τους φακέλους του XAMPP, επιλέγουμε και ανοίγουμε τον φάκελο **htdocs**.



Εικόνα 4.2.12. Δημιουργία και ονομασία νέου φακέλου στο htdocs

Δημιουργούμε ένα νέο φάκελο **ptixiaki** και σε αυτόν αποθηκεύουμε τα αρχεία της εφαρμογής μας , τα οποία θα εμφανίζονται στο φυλλομετρητή με τη διεύθυνση : <http://localhost/ptixiaki>

# Κεφάλαιο 5. Εισαγωγή στις Βάσεις Δεδομένων

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Είναι κοινός τόπος σήμερα η εκτίμηση ότι το αγαθό της πληροφορίας είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους, ώστε να είναι πιο αποδοτικοί, ανταγωνιστικοί αλλά και παραγωγικοί στην εργασία τους.

Τα Συστήματα Βάσεων Δεδομένων (DBS, DataBase System) τα χρησιμοποιούμε για να μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά.

## 5.1 Τα Δεδομένα και οι Πληροφορίες

Με τον όρο πληροφορία αναφερόμαστε συνήθως σε ειδήσεις, γεγονότα και έννοιες που αποκτάμε από την καθημερινή μας επικοινωνία και τα θεωρούμε ως αποκτηθείσα γνώση, ενώ τα δεδομένα μπορούν να είναι μη κατάλληλα επεξεργασμένα και μη ταξινομημένα σύνολα πληροφοριών. Ένας αυστηρός ορισμός για το τι είναι δεδομένα και τι είναι πληροφορία, σύμφωνα με την επιτροπή ANSI των ΗΠΑ, είναι ο εξής:

- **Δεδομένα** (data) είναι μια παράσταση, όπως γράμματα, αριθμοί, σύμβολα στα οποία μπορούμε να δώσουμε κάποια σημασία (έννοια).
- **Πληροφορία** (information) είναι η σημασία που δίνουμε σ' ένα σύνολο από δεδομένα, τα οποία μπορούμε να επεξεργαστούμε βάσει προκαθορισμένων κανόνων και να βγάλουμε έτσι κάποια χρήσιμα συμπεράσματα. Με τις πληροφορίες περιορίζεται η αβεβαιότητα που έχουμε για διάφορα πράγματα και βοηθιόμαστε έτσι στο να λάβουμε σωστές αποφάσεις.

Τα δεδομένα μπορούν να θεωρηθούν ως τρόποι αναπαράστασης εννοιών και γεγονότων που μπορούν να υποστούν διαχείριση και επεξεργασία. Η συλλογή και αποθήκευση ενός τεράστιου όγκου δεδομένων όπως απαιτούν οι κοινωνικές συνθήκες σήμερα, δε λύνει τελείως το πρόβλημα της σωστής οργάνωσης και ταξινόμησης των δεδομένων. Τα δεδομένα θα πρέπει να οργανωθούν με τέτοιο τρόπο έτσι ώστε να μπορούμε να τα εντοπίζουμε και να τα αξιοποιούμε εύκολα και γρήγορα τη στιγμή που τα χρειαζόμαστε.

Ένα κλασικό παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν ο τηλεφωνικός κατάλογος μιας πόλης, όπου οι συνδρομητές δεν θα ήταν καταχωρημένοι αλφαβητικά σύμφωνα με το επώνυμο και το όνομά τους, αλλά εντελώς τυχαία. Ένας τέτοιος τηλεφωνικός κατάλογος θα περιείχε μια τεράστια ποσότητα δεδομένων αλλά θα ήταν ουσιαστικά άχρηστος.

Εκτός λοιπόν από τη μόνιμη αποθήκευση των δεδομένων, χρειαζόμαστε και κάποιους τρόπους ευέλικτης και αποδοτικής οργάνωσής τους.

## 5.2 Βάσεις Δεδομένων

**Συστήματα Βάσης Δεδομένων (ΣΔΒ)** είναι ένα ηλεκτρονικό σύστημα τήρησης εγγραφών, δηλαδή είναι ένα σύστημα για υπολογιστές που ο γενικός σκοπός του είναι να τηρεί πληροφορίες και να δίνει αυτές τις πληροφορίες όταν του ζητούνται.

Οι πληροφορίες που τηρούνται σε ένα τέτοιο σύστημα μπορεί να είναι οτιδήποτε έχει σημασία για το άτομο ή τον οργανισμό που εξυπηρετεί το συγκεκριμένο σύστημα, με άλλα λόγια οτιδήποτε χρειάζεται για την υποβοήθηση των εργασιών αυτού του ατόμου ή οργανισμού.

Η ίδια η **Βάση Δεδομένων (DB, DataBase)** μπορεί να θεωρηθεί ένα είδος ηλεκτρονικής αρχειοθήκης, ένα χώρος για την αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων.

Ο χρήστης του συστήματος έχει στη διάθεσή του ορισμένα βοηθήματα για να εκτελεί στα αρχεία Βάσης Δεδομένων (ΒΔ) διάφορες εργασίες, στις οποίες συγκαταλέγονται ανάμεσα σε άλλες, και οι εξής:

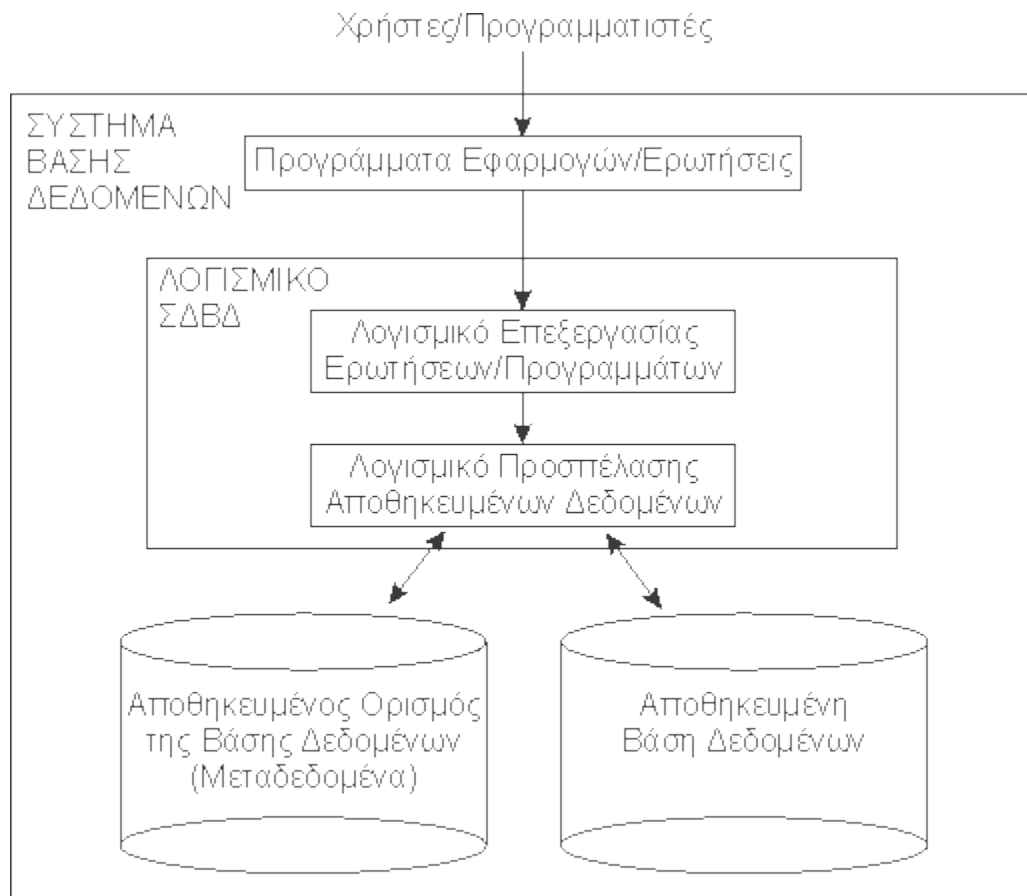
- Η προσθήκη νέων κενών αρχείων στη ΒΔ.
- Η εισαγωγή νέων δεδομένων σε υπάρχοντα αρχεία.
- Η ανάκληση δεδομένων από υπάρχοντα αρχεία.
- Η ενημέρωση δεδομένων σε υπάρχοντα αρχεία.
- Η διαγραφή δεδομένων από υπάρχοντα αρχεία
- Η αφαίρεση υπαρχόντων αρχείων, κενών ή όχι, από τη ΒΔ.

## 5.3 Κατηγορίες χρηστών

Ένα ΣΒΔ αποτελείται από το υλικό, το λογισμικό, τη ΒΔ και τους χρήστες. Είναι δηλαδή, ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια Η/Υ. Αναλυτικά:

- Το υλικό (hardware) αποτελείται όπως είναι γνωστό από τους Η/Υ, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες, όπου είναι αποθηκευμένα τα αρχεία της ΒΔ αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το λογισμικό (software) είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της ΒΔ.
- Η ΒΔ αποτελείται από σύνολο αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα (data integration), δηλ. δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και μερισμένα (data sharing), δηλ. υπάρχει δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της ΒΔ, ανάλογα με τον σκοπό για τον οποίο συνδέεται.
- Οι χρήστες (users) μιας ΒΔ χωρίζονται στις εξής κατηγορίες :
  - **Τελικοί χρήστες** (end users). Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια ΒΔ, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης ΒΔ, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της ΒΔ.
  - **Προγραμματιστές εφαρμογών** (application programmers). Αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
  - **Διαχειριστής δεδομένων** (DA, Data Administrator). Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της ΒΔ και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
  - **Διαχειριστής Βάσης Δεδομένων** (DBA, DataBase Administrator). Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΔΒΔ.





Εικόνα 5.3.1: Σύστημα διαχείρισης βάσεων δεδομένων

**Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, DataBase Management System)** είναι μια συλλογή από προγράμματα που επιτρέπουν στους χρήστες να δημιουργήσουν και να συντηρήσουν μία ΒΔ. Επομένως, το Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) είναι ένα γενικής χρήσης σύστημα λογισμικού που διευκολύνει τις διαδικασίες ορισμού, κατασκευής και χειρισμού ΒΔ για διάφορες εφαρμογές.

Το ΣΔΒΔ παρέχει στους χρήστες και τις εφαρμογές που αναπτύσσουν οι προγραμματιστές ένα επίπεδο λογισμικού για την επεξεργασία ερωτήσεων και προγραμμάτων. Για την πρόσβαση στον ορισμό της ΒΔ και τα δεδομένα χρησιμοποιείται το επίπεδο προσπέλασης αποθηκευμένων δεδομένων.

## 5.4 Γλώσσες Βάσεων Δεδομένων

Για να μπορέσουμε να δημιουργήσουμε και να διαχειριστούμε μια ΒΔ, μπορούμε να χρησιμοποιήσουμε ειδικές γλώσσες προγραμματισμού, τις λεγόμενες γλώσσες ερωτημάτων (query languages). Είναι γλώσσες μη διαδικαστικές, τέταρτης γενιάς (4th generation languages). Εμείς απλά διατυπώνουμε με απλές και κατανοητές εντολές το τί πληροφορίες ζητάμε και το ΣΔΒΔ αναλαμβάνει να μας απαντήσει.

Η SQL (Structured Query Language, δηλ. Δομημένη Γλώσσα Ερωτημάτων) είναι σήμερα η πιο δημοφιλής και πιο διαδεδομένη γλώσσα ανάπτυξης και διαχείρισης σχεσιακών ΒΔ.

Η SQL αποτελείται από εντολές με τα ορίσματά τους, τις οποίες μπορούμε να χρησιμοποιήσουμε με συγκεκριμένους κανόνες σύνταξης για να πάρουμε τα αποτελέσματα που θέλουμε. Με την SQL μπορούμε να δημιουργήσουμε μια ΒΔ και τους πίνακές της με τα αντίστοιχα πεδία, να καταχωρήσουμε δεδομένα στους πίνακες, να τροποποιήσουμε και να διαγράψουμε τα δεδομένα αυτά, να αλλάξουμε τη δομή των πινάκων με προσθήκη και διαγραφή πεδίων και να εμφανίσουμε πληροφορίες (συνδυασμούς από δεδομένα).

Η SQL έχει διάφορα τμήματα, τα πιο βασικά είναι τα παρακάτω:

- Τη Γλώσσα Ορισμού Δεδομένων (DDL, Data Definition Language), η οποία περιέχει τις απαραίτητες εντολές για τον ορισμό και την τροποποίηση του σχεσιακού σχήματος, καθώς και για τη δημιουργία, την τροποποίηση και τη διαγραφή σχέσεων. Περιέχει ακόμη τις εντολές δημιουργίας και επεξεργασίας όψεων και ορισμού περιορισμών ακεραιότητας.
- Τη Γλώσσα Χειρισμού Δεδομένων (DML, Data Manipulation Language), η οποία περιέχει τις απαραίτητες εντολές για την εμφάνιση (αναζήτηση) δεδομένων, καθώς και για την καταχώρηση, τροποποίηση και διαγραφή των εγγραφών (πλειάδων) μιας σχέσης.
- Τέλος, περιέχει εντολές για τον ορισμό και την επεξεργασία συναλλαγών (transactions) και εντολές για την ασφάλεια (authentication).

## 5.5 Μοντέλο Οντοτήτων Συσχετίσεων

Το Μοντέλο Οντοτήτων-Συσχετίσεων (ER Model, Entity-Relationship Model) προτάθηκε από τον P.P. Chen το 1976 για να περιγράψει με γραφικά σύμβολα τα δεδομένα ως οντότητες, συσχετίσεις και γνωρίσματα. Από τότε έχει αποτελέσει το αντικείμενο εκτεταμένων ερευνών με αποτέλεσμα τη διαρκή ανάπτυξή του.

Σήμερα θεωρείται ένα από τα πιο σημαντικά εργαλεία κατασκευής εννοιολογικών μοντέλων ΒΔ, σε περιβάλλοντα με ποικίλες απαιτήσεις, και αποτελεί βασική συνιστώσα μεθοδολογιών ανάπτυξης πληροφοριακών συστημάτων.

Χρησιμοποιείται κατά το στάδιο του εννοιολογικού σχεδιασμού (conceptual design) και του λογικού σχεδιασμού της ΒΔ.

Ένα μοντέλο δεδομένων είναι ένα εργαλείο που επιτρέπει την αφαίρεση περιττών πληροφοριών από το σύνολο των πραγματικών δεδομένων. Κάθε μοντέλο βασίζεται σε ορισμένους κανόνες και διαδικασίες.

Ένα μοντέλο δεδομένων έχει τρεις βασικές συνιστώσες:

1. Δομές (structures): Είναι τα αντικείμενα του μικρόκοσμου.
2. Πράξεις (operations): Είναι οι ενέργειες που γίνονται στις δομές.
3. Περιορισμοί (constraints): Είναι οι περιορισμοί που μπαίνουν πάνω στις δομές.

Άρα τα μοντέλα δεδομένων είναι μία συλλογή από εργαλεία για την περιγραφή δομών δεδομένων, τη σημασιολογία (semantics) δεδομένων και τους περιορισμούς δεδομένων.

Υπάρχουν τρεις διαφορετικές ομάδες που αντιστοιχούν στις ανάλογες φάσεις σχεδιασμού:

1. Εννοιολογικά μοντέλα βασισμένα σε αντικείμενα.
2. Λογικά μοντέλα βασισμένα σε εγγραφές.
3. Φυσικά μοντέλα δεδομένων βασίζονται στις πραγματικές τιμές [28].

## 5.6 Σχεσιακό Μοντέλο Δεδομένων

Τα λογικά μοντέλα βασισμένα σε εγγραφές χρησιμοποιούνται για να περιγράψουν τα δεδομένα από τον εννοιολογικό σχεδιασμό στον λογικό σχεδιασμό. Αντίθετα από τα εννοιολογικά μοντέλα, τα λογικά μοντέλα καθορίζουν τη γενική λογική δομή της ΒΔ και παρέχουν μία υψηλότερου επιπέδου περιγραφή της υλοποίησής της.

Ονομάζονται έτσι επειδή η ΒΔ είναι δομημένη σε εγγραφές διάφορων τύπων. Κάθε τύπος εγγραφής καθορίζει έναν σταθερό αριθμό πεδίων (γνωρίσματα - ιδιότητες). Κάθε πεδίο είναι συνήθως σταθερού μήκους (αυτό απλοποιεί την εφαρμογή).

Στο σχεσιακό μοντέλο οι οντότητες και οι συσχετίσεις αντιπροσωπεύονται από μία συλλογή από κανονικοποιημένους πίνακες (σχέσεις). Κάθε κανονικοποιημένος πίνακας έχει διάφορες στήλες με μοναδικά ονόματα.

Οι σημαντικότεροι από τους όρους που χρησιμοποιούνται με τα "αντικείμενα του σχεσιακού μοντέλου" είναι η σχέση (relation) που αντιστοιχεί σε εκείνο που μέχρι τώρα ονομαζόταν πίνακας, μία συστοιχία ή πλειάδα (tuple) αντιστοιχεί σε μία γραμμή ενός τέτοιου πίνακα και ένα γνώρισμα (attribute) αντιστοιχεί σε μία στήλη-γνώρισμα. Το πλήθος των συστοιχιών ονομάζεται πληθικότητα (cardinality) και το πλήθος των γνωρισμάτων ονομάζεται βαθμός (degree).

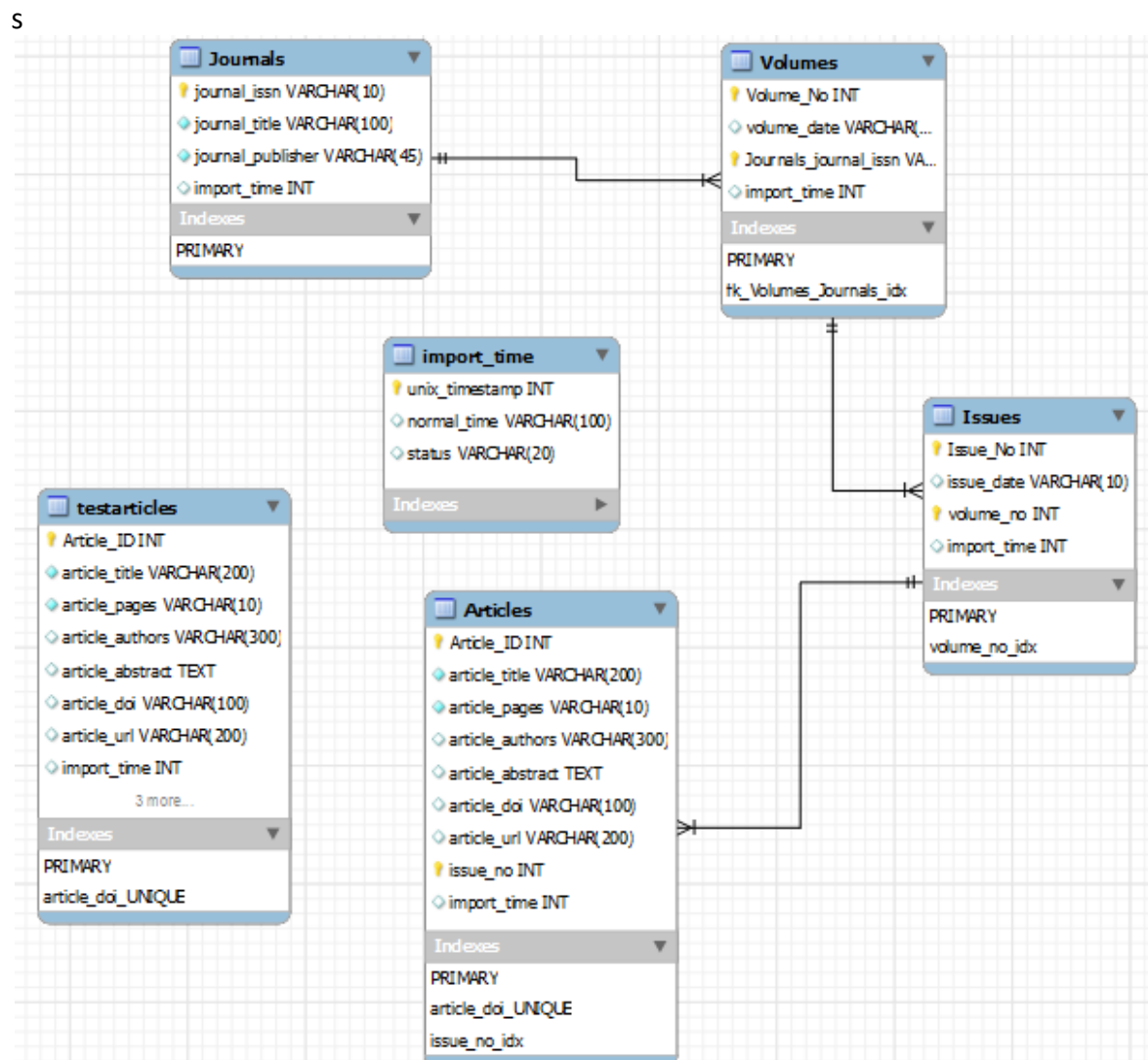
Το πρωτεύον κλειδί (primary key) είναι ένα μοναδικό αναγνωριστικό για τον πίνακα, δηλαδή μία στήλη ή συνδυασμός στηλών που έχει την ιδιότητα οι τιμές του να είναι μοναδικές.

Τέλος, ένα πεδίο ορισμού (domain) είναι μία δεξαμενή τιμών από την οποία τα συγκεκριμένα γνωρίσματα των συγκεκριμένων σχέσεων αντλούν τις συγκεκριμένες τιμές τους.

# Κεφάλαιο 6. Σχεδίαση της Βάσης Δεδομένων

## 6.1 Σχεδίαση διαγράμματος ER

Για την δημιουργία του διαγράμματος ER, χρησιμοποίησα τον πρόγραμμα MySQL Workbench (version 5.2.41), το οποίο προσφέρει πολλές δυνατότητες για το σχεδιασμό της βάσης δεδομένων.



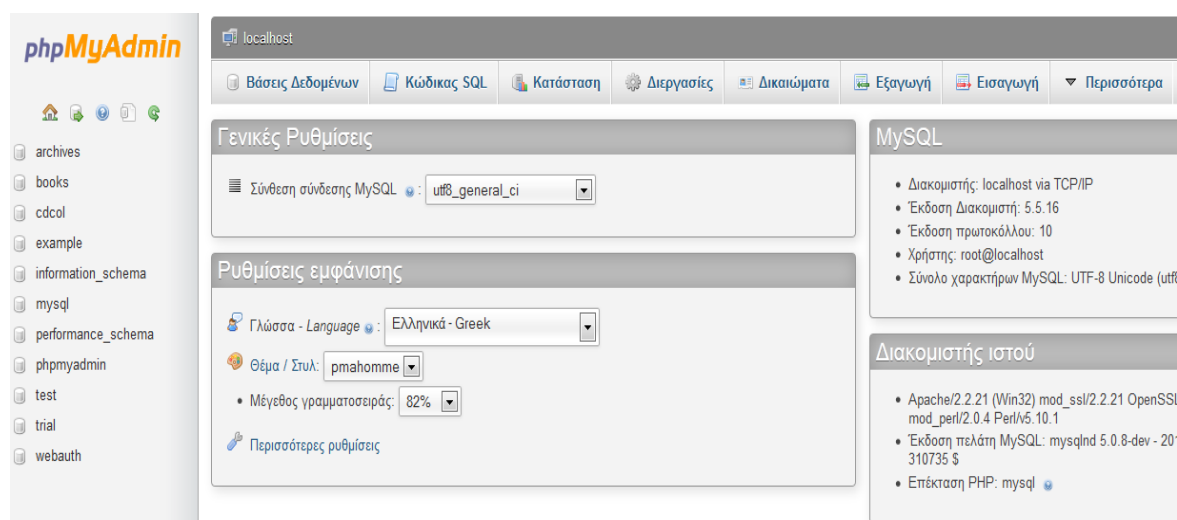
6.1. Διάγραμμα ER

## 6.2 Σχεδίαση Σχεσιακού Σχήματος Βάσης Δεδομένων

Για την αποθήκευση των πληροφοριών που ανακτώνται από τα αρχεία του εκάστοτε εκδότη είναι απαραίτητη η δημιουργία μιας βάσης δεδομένων. Η βάση που θα δημιουργήσουμε έχει το όνομα `books` και θα περιέχει τους εξής πίνακες:

- `journal`
- `volume`
- `issue`
- `article`
- `duplicates`
- `session`

Στον φυλλομετρητή πληκτρολογούμε την διεύθυνση <http://localhost/phpmyadmin/>, η οποία μας εμφανίζει ένα graphical interface για την δημιουργία της βάσης μας.



Εικόνα 6.2.1. Το interface του phpMyAdmin

Δημιουργούμε την βάση δεδομένων (database) με όνομα “**books**” με σύνθεση σύνδεσης MySQL την default: `utf8_general_ci`.

## 6.2.1 Αναλυτική Περιγραφή Πινάκων Βάσης Δεδομένων

Σε περαιτέρω ανάλυση, εξετάζουμε τον κάθε πίνακα ξεχωριστά. Συγκεκριμένα:

Ο πίνακας **session** περιέχει πληροφορίες σχετικά με την ώρα εισαγωγής των εγγραφών:

Πίνακας 6.2.1. **session**

Στήλη	Τύπος	Κλειδί
unix_timestamp	int(11)	PRIMARY
normal_time	varchar(100)	
status	varchar(20)	

Ο πίνακας **journal** περιέχει όλες τις πληροφορίες σχετικά με το περιοδικό:

Πίνακας 6.2.1.1. **journal**

Στήλη	Τύπος	Κλειδί
issn	varchar(20)	PRIMARY
j_title	varchar(100)	
j_publisher	varchar(30)	
Import_time	int(11)	

Ο πίνακας **volume** περιέχει όλες τις πληροφορίες σχετικά με τον τόμο:

Πίνακας 6.2.1.2. **volume**

Στήλη	Τύπος	Κλειδί
volume_no	int(11)	PRIMARY
issn	varchar(20)	PRIMARY, FOREIGN
year	int(11)	
Import_time	int(11)	

Ο πίνακας **issue** περιέχει όλες τις πληροφορίες σχετικά με το τεύχος:

Πίνακας 6.2.1.3. **issue**

Στήλη	Τύπος	Κλειδί
issue_no	int(11)	PRIMARY
issue_pages	varchar(10)	
issue_date	varchar(10)	
issn	varchar(20)	PRIMARY, FOREIGN
volume_no	int(11)	PRIMARY, FOREIGN
Import_time	int(11)	

Ο πίνακας **articles** περιέχει όλες τις πληροφορίες σχετικά με το άρθρο:

Πίνακας 6.2.1.4. **articles**

Στήλη	Τύπος	Κλειδί
art_id	int(11)	PRIMARY
art_doi	varchar(100)	UNIQUE
title	LONGTEXT	
url	varchar(200)	UNIQUE
authors	varchar(200)	
university	TEXT	
abstract	LONGTEXT	
keywords	TEXT	
pages	varchar(10)	
issn	varchar(20)	PRIMARY, FOREIGN
volume_no	int(11)	PRIMARY, FOREIGN
issue_no	int(11)	PRIMARY, FOREIGN
Import_time	int(11)	

Ο πίνακας **duplicates** περιέχει όλες τις διπλότυπες εγγραφές:

Πίνακας 6.2.1.4. **duplicates**

Στήλη	Τύπος	Κλειδί
art_id	int(11)	
art_doi	varchar(100)	
title	LONGTEXT	
url	varchar(200)	
authors	varchar(200)	
university	TEXT	
abstract	LONGTEXT	
keywords	TEXT	
pages	varchar(10)	
issn	varchar(20)	
volume_no	int(11)	
issue_no	int(11)	
Import_time	int(11)	



## 6.2.2 Δημιουργία της βάσης δεδομένων με χρήση της MySQL

Αφαίρεση Πινάκων:

```
DROP TABLE session;
DROP TABLE articles;
DROP TABLE duplicates;
DROP TABLE issue;
DROP TABLE volume;
DROP TABLE journal;
```

Για την δημιουργία των πινάκων χρησιμοποιούμε την μηχανή αποθήκευσης InnoDB, γιατί υποστηρίζει τα foreign keys, σε αντίθεση με την MyISAM που δεν τα υποστηρίζει.

Δημιουργία του πίνακα **session**:

```
CREATE TABLE IF NOT EXISTS session (
  unix_timestamp int(11) NOT NULL,
  normal_time varchar(100) NOT NULL,
  status varchar(20) NOT NULL
) ENGINE=InnoDB;
```

Δημιουργία του πίνακα **journal**:

```
CREATE TABLE IF NOT EXISTS journal (
  issn varchar(20) NOT NULL,
  j_title varchar(100) NOT NULL,
  j_publisher varchar(30) NOT NULL,
  import_time int(11),
  PRIMARY KEY (issn)
) ENGINE=InnoDB;
```

Δημιουργία του πίνακα **volume**:

```
CREATE TABLE IF NOT EXISTS volume (
  volume_no int(11) NOT NULL,
  issn varchar(20) NOT NULL,
  year int(11) NOT NULL,
  import_time int(11),
  PRIMARY KEY (issn, volume_no),
  FOREIGN KEY (issn) REFERENCES journal(issn)
) ENGINE=InnoDB;
```

### Δημιουργία του πίνακα **issue**:

```
CREATE TABLE IF NOT EXISTS issue (
  issue_no int(11) NOT NULL,
  issue_pages varchar(10) NOT NULL,
  issue_date varchar(10) NOT NULL,
  issn varchar(20) NOT NULL,
  volume_no int(11) NOT NULL,
  import_time int(11),
  PRIMARY KEY (issue_no,issn,volume_no),
  FOREIGN KEY (issn,volume_no) REFERENCES
volume(issn,volume no)
) ENGINE=InnoDB;
```

### Δημιουργία του πίνακα **articles**:

```
CREATE TABLE IF NOT EXISTS articles (
  art_id INT(11) NOT NULL,
  issn varchar(20) NOT NULL,
  volume_no int(11) NOT NULL,
  issue_no int(11) NOT NULL,
  art_doi varchar(100) NOT NULL UNIQUE,
  title LONGTEXT NOT NULL,
  url varchar(200) NOT NULL UNIQUE,
  authors varchar(200) NOT NULL,
  university TEXT NOT NULL,
  abstract LONGTEXT NOT NULL,
  keywords TEXT NOT NULL,
  pages varchar(10) NOT NULL,
  import_time int(11),
  PRIMARY KEY (art_id,issn,volume_no,issue_no),
  FOREIGN KEY (issn,volume_no,issue_no) REFERENCES
issue(issn,volume_no,issue_no)
) ENGINE=InnoDB;
```

### Δημιουργία του πίνακα **duplicates**:

```
CREATE TABLE IF NOT EXISTS duplicates (  
  art_id INT(11) NOT NULL,  
  issn varchar(20) NOT NULL,  
  volume_no int(11) NOT NULL,  
  issue_no int(11) NOT NULL,  
  art_doi varchar(100) NOT NULL,  
  title LONGTEXT NOT NULL,  
  url varchar(200) NOT NULL,  
  authors varchar(200) NOT NULL,  
  university TEXT NOT NULL,  
  abstract LONGTEXT NOT NULL,  
  keywords TEXT NOT NULL,  
  pages varchar(10) NOT NULL,  
  import_time int(11)  
) ENGINE=InnoDB;
```

# Κεφάλαιο 7. Ανάλυση, επεξεργασία και αποθήκευση των δεδομένων

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τα δεδομένα του εκάστοτε εκδότη. Θα πραγματοποιήσουμε για τα δεδομένα του κάθε ένα τις ακόλουθες διαδικασίες:

- Ανάλυση (parsing)
- Επεξεργασία
- Αποθήκευση στην βάση δεδομένων

## 7.1. ELSEVIER

Κάθε γραμμή ξεκινά με underscore και 2 χαρακτήρες που δηλώνουν τι είδους πληροφορία μας δίνει η γραμμή:

Για παράδειγμα ο τίτλος του περιοδικού ξεκινάει με `_jn`. Για να πάρουμε την πληροφορία που μας δίνει μέσω της `rh` εκτελούμε τα παρακάτω βήματα:

- Παίρνουμε τα περιεχόμενα του αρχείου με χρήση της μεθόδου `file()`.
- Στη συνέχεια κρατάμε τον αριθμό των γραμμών σε μια μεταβλητή, την οποία θα χρησιμοποιήσουμε στο `for loop`, με το οποίο αναλύουμε τα όλα τα περιεχόμενα, γραμμή προς γραμμή.
- Ελέγχουμε αν η γραμμή αρχίζει με τα 3 γράμματα που μας ενδιαφέρουν (εδώ, αν αρχίζει με `_jn`).
- «Σπάμε» την κάθε γραμμή, χρησιμοποιώντας το κενό διάστημα σαν διαχωριστή (π.χ. `_jn The Lancet`), ώστε να πάρουμε μόνο την πληροφορία που μας χρειάζεται (`The Lancet`) και να την αποθηκεύσουμε σε μία μεταβλητή.

```
$filename = "dataset3.toc";  
$contents = file ($filename);  
$l=count ($contents);
```

Εικόνα 5.1.1 Παίρνουμε τα περιεχόμενα του αρχείου

```
else if ($rest[0]=='_jn'){  
  
    $jn_title=substr ($contents[$i], 3, strlen ($contents[$i]));  
  
    $jn_title=trim ($jn_title);
```

Εικόνα 7.1.2 Παίρνουμε τον τίτλο του περιοδικού

Η μέθοδος **substr** παίρνει τρεις παραμέτρους. Το string από το οποίο θέλουμε να πάρουμε ένα μέρος του, ένα νούμερο που δείχνει από πια θέση να αρχίσουμε το «κόψιμο» και το μήκος του string (σε αυτήν την περίπτωση το **strlen()** το οποίο μας δίνει εύκολα το μήκος του string).

Τέλος, για να μην υπάρξουν προβλήματα αργότερα, αφαιρούμε τυχόν κενά δεξιά ή αριστερά του string με τη βοήθεια της μεθόδου **trim()**.

Σε μια πιο δύσκολη περίπτωση, για παράδειγμα για την DOI πληροφορία η οποία έχει την εξής παρακάτω μορφή, χρειαζόμαστε κάποια επιπλέον βήματα.

```
_ii S0196-6553(97)90031-6  
_ii [DOI] 10.1016/S0196-6553(97)90031-6
```

Εικόνα 7.1.3. Η μορφή της DOI πληροφορίας

Εδώ χρειαζόμαστε μόνο το υπο-string 10.1016/S0196-6553(97)90031-6.

```
else if ($rest[0]=='_ii'){  
    $art_doi=substr($contents[$i], 3, strlen($contents[$i]));  
    $is_ii=true;  
    while($is_ii){  
        $i++;  
        $rest = explode(" ", $contents[$i]);  
        if ($rest[0]=='_ii'){  
            $art_doi.=substr($contents[$i], 3, strlen($contents[$i]));  
        }  
        else{  
            $is_ii=false;  
        }  
    }  
    $art_doi=trim($art_doi);  
  
    $art_doi=preg_replace("/\//", "", $art_doi);  
    if (preg_match('/.*[DOI]/', $art_doi))  
        $art_doi=substr($art_doi, 28, strlen($art_doi));  
    $art_doi=trim($art_doi);  
    //echo "<b>DOI: </b>".$art_doi."<br>". "<br>";  
  
    $i--;  
}
```

Εικόνα 7.1.4. Παίρνουμε την DOI πληροφορία

Εδώ προσθέτουμε και την 2<sup>η</sup> γραμμή του DOI για να έχουμε ολόκληρη την πληροφορία. Εδώ με την μέθοδο **preg\_match()** βρίσκουμε αν υπάρχει το string [DOI]. Αν υπάρχει, με την μέθοδο **substr()** παίρνουμε το DOI.

## 7.2. SPRINGER

Εδώ έχουμε Xml documents τα οποία τα αναλύουμε (parse) με τον XML DOM parser, παίρνοντας τις πληροφορίες από κάθε tag που μας ενδιαφέρει και στην συνέχεια τις αποθηκεύουμε.

Εδώ ακολουθούμε τα εξής βήματα:

- Αρχικά, δημιουργούμε ένα DOM document.
- Στην συνέχεια διασχίζουμε όλους τους φακέλους που υπάρχουν στο μονοπάτι μας (path), για να βρούμε όσα αρχεία τελειώνουν σε .xml ή .Meta, με σκοπό να τα αναλύσουμε.
- Μετά βρίσκουμε το tag που μας ενδιαφέρει με χρήση της μεθόδου **getElementsByTagName()** του XML DOM parser.
- Τέλος, αποθηκεύουμε το περιεχόμενο του tag σε μια μεταβλητή.

Π.χ., για το όνομα του εκδότη (PublisherName tag):

```
if(substr($fil, -3) == ".xml" || substr($fil, -4) == ".Meta"){
    $articl_id++;
    //echo '<b>URL: </b>'. $fil. '<br>';

    $doc->load($fil);
    $publish = $doc->getElementsByTagName('PublisherName');
    if($publish){
        foreach ($publish as $publ) {

            $pub_name=$publ->nodeValue;

            $pub_name=preg_replace("/\'/", "'", $pub_name);
            //echo "<b>PublisherName: </b>". $pub_name. '<br>';
        }
    }
}
```

Εικόνα 7.2.1. Παίρνουμε το όνομα του εκδότη

Με την μέθοδο load του XML DOM parser φορτώνουμε το αρχείο xml και στην συνέχεια ψάχνουμε για το tag που θέλουμε.

Η μέθοδος **getElementsByTagName()** μας επιστρέφει έναν πίνακα τον οποίο τον περνάμε με μία foreach και επιστρέφουμε την τιμή του με την ιδιότητα **nodeValue**. Στην συνέχεια, για να μην έχουμε πρόβλημα στα sql queries αντικαθιστούμε το μονό εισαγωγικό με διπλά εισαγωγικά.

## 7.3. KLUWER

Εδώ έχουμε δεδομένα σε αρχεία .oa3 τα οποία μοιάζουν με xml αλλά δεν είναι. Δεν μπορούμε να τα αναλύσουμε με τον XML DOM parser ή με το SimpleXML, οπότε το παίρνουμε σαν απλό αρχείο text και το αναλύουμε με την βοήθεια των κανονικών εκφράσεων (regular expressions)

Εδώ υπάρχουν αρκετές δυσκολίες, που οφείλονται ως επί το πλείστον στο ότι δεν μπορούμε να τα αναλύσουμε σαν αρχεία XML.

Ακολουθούμε τα εξής βήματα:

- Ελέγχουμε αν η γραμμή ξεκινάει με κάποιο πρότυπο.
- Στην συνέχεια, αποθηκεύουμε την γραμμή σε μια μεταβλητή και αφαιρούμε με την μέθοδο **str\_replace** τα opening και closing tags που τα έχουμε αποθηκεύσει σε άλλες μεταβλητές.
- Σε πολλές περιπτώσεις, όπως στην περίπτωση του τόμου (volume) ή του τεύχους (issue) πρέπει να χρησιμοποιήσουμε την μέθοδο **substr** για να πάρουμε μόνο ένα μικρό μέρος του string.
- Επίσης πολλές φορές για μια πληροφορία που μας ενδιαφέρει, π.χ. τον τόμο (volume), χρειάζονται παραπάνω από ένα πρότυπα (patterns) για να ταιριάξουν σε όλα τα αρχεία που υπάρχουν, τα οποία αρκετές φορές έχουν λίγο διαφορετική δομή.
- Στο τέλος, αν έχουμε string αντικαθιστούμε τα μονά εισαγωγικά με διπλά για να μην έχουμε προβλήματα στα sql queries, και αφαιρούμε τυχόν κενά, δεξιά και αριστερά ενός string ή ενός αριθμού.

Για παράδειγμα, για τον τίτλο του περιοδικού κάνουμε τα εξής:

```
else if ((preg_match('/^<jtl/', $data[$i])) || (preg_match('/^<JTL>/', $data[$i]))){
    $j_title=$data[$i];
    $input("<jtl>");
    $input1("</jtl>");
    $input2("<JTL>");
    $input3("</JTL>");
    $replace("");
    $j_title=str_replace($input, $replace, $j_title);
    $j_title=str_replace($input1, $replace, $j_title);
    $j_title=str_replace($input2, $replace, $j_title);
    $j_title=str_replace($input3, $replace, $j_title);
    $j_title=preg_replace("/\'/", "'", $j_title);
    $j_title=trim($j_title);
    // echo "JOURNAL TITLE: ".$j_title."<br>";
}
```

Εικόνα 7.3.1. Παίρνουμε το όνομα του περιοδικού

Για μια πιο δύσκολη περίπτωση, π.χ. αυτή του volume έχουμε τα εξής παραδείγματα:

```
<binding vid = "14"  
        iid = "4">
```

Εικόνα 7.3.2. Βλέπουμε πως δίνονται οι πληροφορίες για το volume σε ένα oa3 αρχείο

Σε αυτό το παράδειγμα, για να πάρουμε τον αριθμό του τόμου (14):

```
else if (preg_match('/^<binding vid/', $data[$i])) {  
    //echo "volume id."<br>;  
    $rest=explode(" ", $data[$i]);  
    $vol_id=preg_replace("/\\"/,"", $rest[3]);  
    $vol_id=trim($vol_id);  
    //echo "VOLUME ID: ".$vol_id.<br>;  
}
```

Εικόνα 7.3.3. Παίρνουμε τον αριθμό του volume

Ενώ στο παρακάτω παράδειγμα, για να πάρουμε το νούμερο του τόμου (2)

```
<binding  
        vid = "2"  
        iid = "1">
```

Εικόνα 7.3.4. Βλέπουμε πως δίνονται οι πληροφορίες για το volume σε ένα άλλο oa3 αρχείο

Κάνουμε τα εξής:

```
else if (preg_match('/^.*vid =/', $data[$i])) {  
    $vol_id=substr($data[$i], 14, 18);  
    $vol_id=preg_replace(">/","", $vol_id);  
    $vol_id=preg_replace("/\\"/,"", $vol_id);  
    $vol_id=trim($vol_id);  
    //echo "VOLUME ID: ".$vol_id.<br>;  
}
```

Εικόνα 7.3.5. Παίρνουμε τον αριθμό του volume

Υπάρχουν και άλλες περιπτώσεις ακόμα και για το volume, αλλά και για το τεύχος (issue) και για τον αριθμό της πρώτης και τελευταίας σελίδας του άρθρου.



## 7.4. ΙΕΕΕ

Εδώ έχουμε Xml documents τα οποία τα αναλύουμε (parse) με τον XML DOM parser, παίρνοντας τις πληροφορίες από κάθε tag που μας ενδιαφέρει και στην συνέχεια τις αποθηκεύουμε.

Ακολουθούμε τα εξής βήματα:

- Αρχικά, δημιουργούμε ένα DOM document.
- Στην συνέχεια διασχίζουμε όλους τους φακέλους που υπάρχουν στο μονοπάτι μας (path), για να βρούμε όσα αρχεία τελειώνουν σε .xml ή .Meta, με σκοπό να τα αναλύσουμε.
- Ελέγχουμε αν κάποιο tag έχει παιδιά (children) με την μέθοδο **hasChildNodes()**.
- Διασχίζουμε τον πίνακα με τα παιδιά μέχρι να βρούμε το tag που ψάχνουμε.
- Αν θέλουμε κάποιο tag που έχει κάποιο γνώρισμα (attribute) χρησιμοποιούμε την μέθοδο **hasAttributes()** και ψάχνουμε τον πίνακα για να βρούμε το attribute που ψάχνουμε και την αντίστοιχη τιμή του.
- Στο τέλος, αν έχουμε string αντικαθιστούμε τα μονά εισαγωγικά με διπλά για να μην έχουμε προβλήματα στα sql queries, και αφαιρούμε τυχόν κενά, δεξιά και αριστερά ενός string ή ενός αριθμού.

Για παράδειγμα έχουμε το παρακάτω παράδειγμα ενός ISSN περιοδικού και του τίτλου ενός άρθρου:

```
<issn mediatype="Paper">1551-3203</issn>
```

```
<issn mediatype="Online">0000-0000</issn>
```

Εικόνα 7.4.1. Το ISSN του περιοδικού όπως παρουσιάζεται στο αρχείο xml

```
<article>  
<title><![CDATA[Table of contents]]></title>
```

Εικόνα 7.4.2. Ο τίτλος ενός άρθρου όπως παρουσιάζεται στο αρχείο xml

Για παράδειγμα, για να βρούμε το ISSN ενός περιοδικού ελέγχουμε αν κάποιο attribute του περιοδικού έχει την τιμή *Paper*:

```
$issnss = $doc->getElementsByTagName('issn');

if($issnss){

    foreach ($issnss as $issns) {
        if ($issns->hasAttributes()) {
            $attributes = $issns->attributes;

            foreach($attributes as $attr) {
                if($attr->nodeValue=="Paper"){
                    $jn_issn=$issns->nodeValue;

                    $jn_issn=preg_replace("/\'/", "'", $jn_issn);
                    $jn_issn=trim($jn_issn);
                    //echo "<b>JOURNAL ISSN: </b>".$jn_issn."<br>";
                    //print "Attribute Name: ".$attr->nodeName."<br>";
                    //print "Attribute Value: ".$attr->nodeValue."<br>";
                }
            }
        }
    }
}
```

Εικόνα 7.4.3. Παίρνουμε το ISSN του περιοδικού με χρήση της μεθόδου `hasAttributes()`

Για να βρούμε τον τίτλο ενός άρθρου κάνουμε την παρακάτω διαδικασία:

```
elseif($nodes->nodeName=="article"){
    if ($nodes->hasChildNodes()) {
        $children = $nodes->childNodes;
        foreach($children as $child) {
            if ($child->nodeType != XML_TEXT_NODE) {
                //echo "NODE NAME: ".$child->nodeName."<br>";
                if($child->nodeName=="title"){
                    $art_id++;
                    $art_title=$child->nodeValue;

                    $art_title=preg_replace("/\'/", "'", $art_title);
                    $art_title=trim($art_title);
                    //echo "<b>ARTICLE TITLE:</b> ".$art_title."<br>";
                }
            }
        }
    }
}
```

Εικόνα 7.4.4. Παίρνουμε τον τίτλο του άρθρου με χρήση της μεθόδου `hasChildNodes()`

# Κεφάλαιο 8. Scripts

Πέρα από την εισαγωγή των δεδομένων στην βάση μας, χρειάζονται και έλεγχοι πάνω στις διάφορες διαδικασίες. Αυτοί θα μας δείχνουν αν υπήρξε κάποιο πρόβλημα κατά την εισαγωγή των δεδομένων, και αν ναι να μας εμφανίζει το πρόβλημα. Επίσης θα γίνεται έλεγχος για να δούμε αν μπήκαν όλα τα δεδομένα στην βάση μας.

Τα scripts είναι παρόμοια και για τους τέσσερις εκδότες και συνολικά είναι τέσσερα για κάθε εκδότη.

## 8.1 1<sup>ο</sup> script: Εισαγωγή εγγραφών και έλεγχος για διπλότυπες εγγραφές

Στο 1<sup>ο</sup> script γίνεται η ανάλυση των δεδομένων όπως την είδαμε στο προηγούμενο κεφάλαιο, και η εισαγωγή στην βάση δεδομένων.

Στην αρχή παίρνουμε την ώρα και την ημερομηνία σε unix timestamp μορφή με χρήση της μεθόδου `getTimestamp()` και σε κανονική μορφή με χρήσης της μεθόδου `date()`, και τα αποθηκεύουμε στον πίνακα `session` (όπου διατηρούνται οι πληροφορίες κάθε session), αφήνοντας το πεδίο `status` (δηλαδή αν ολοκληρώθηκε ή όχι η εκτέλεση του script) κενό.

```
//take the date and time
$date = new DateTime();
// convert it to unix_timestamp
$unixtimestamp=$date->getTimestamp();
// take the date in normal format YYYY-MM-DD HH:II:SS
$dateformat=date('Y-m-d h:i:s',$unixtimestamp);

// Insert date, unix_timestamp and status in the session table
$sql0="insert into session(unix_timestamp,normal_time,status)values
$result0 = mysqli_query($con, $sql0);
```

Εικόνα 8.1.1 Παίρνουμε τις ημερομηνίες σε μορφή `unix timestamp` και σε κανονική μορφή

Σε περίπτωση σφάλματος αποθηκεύουμε την ώρα που έγινε το σφάλμα, τον κωδικό του mysqli σφάλματος και την αντίστοιχη περιγραφή του σε ένα αρχείο txt με χρήση του try catch.

```
try {
    $q3="insert into issue(issn,
    $vol_issue, '$iss_pages', '$is
    $counter++;
    $result3 = mysqli_query($con
}
catch (Exception $e)
{
    error_log($dateformat."\r\n"
```

Εικόνα 8.1.2 Αποθήκευση του κωδικού και της περιγραφής του σφάλματος mysqli

Η μέθοδος **mysqli\_errno()** μας δίνει τον κωδικό του σφάλματος. Η μέθοδος **mysqli\_error()** μας δίνει την περιγραφή του σφάλματος.

Σε περίπτωση διπλότυπης εγγραφής, αποθηκεύουμε τις πληροφορίες του σφάλματος σε ένα άλλο αρχείο txt.

```
if(mysqli_errno($con)==1062) {
// If duplicate insert it in table duplicates an
error_log($dateformat."\r\n".mysqli_errno($con).
```

Εικόνα 8.1.3 Αποθήκευση του κωδικού, της περιγραφής του σφάλματος mysqli και πληροφορίες της εγγραφής, αν βρεθεί διπλότυπη εγγραφή

Για παράδειγμα ένα σφάλμα που αποθηκεύεται στο αρχείο έχει την παρακάτω μορφή:

```
ErrorCode: 1062
Errormessage: Duplicate entry '97800027-02615614-16-1' for key 'PRIMARY'

Journal: : 02615614
Volume: : 16
Issue: : 1
Article ID: : 97800027
Article DOI: : 10.1016/S0261-5614(97)80002-7
```

Εικόνα 8.1.4 Ένα σφάλμα διπλοεγγραφής

Βλέπουμε ότι ο κωδικός σφάλματος είναι ο 1062 που αντιστοιχεί σε μια διπλότυπη εγγραφή. Βλέπουμε επίσης και γενικές πληροφορίες για την συγκεκριμένη εγγραφή, όπως το id του άρθρου, το ISSN του περιοδικού, το νούμερο του Volume, το νούμερο του issue και τον αριθμό DOI του άρθρου.

Αν ο κωδικός του σφάλματος είναι 1062 (που αντιστοιχεί σε διπλότυπη εγγραφή), εισάγουμε την εγγραφή στον πίνακα duplicates.

Σε περίπτωση που οι διπλοεγγραφές ενός session ξεπεράσουν ένα όριο (π.χ. 20), εμφανίζονται στον χρήστη 2 κουμπιά τα οποία ρωτάνε τον χρήστη αν θέλει να συνεχίσει την εισαγωγή, ή αν θέλει να την ακυρώσει, διαγράφοντας όλες τις μέχρι τώρα εγγραφές που έχουν αποθηκευτεί στην βάση δεδομένων, μεταφέροντάς τον σε μια νέα σελίδα.

```
@$q55="SELECT COUNT(*) FROM duplicates";

$result55 = mysqli_query($con, $q55);
$row = $result55->fetch_row();

if($row[0]>20&&$answer==false) {

    $_SESSION['count']=$counter;
    $_SESSION['timestamp']=$unixtimestamp;
    $_SESSION['date']=$dateformat;
```

```
echo '<form action="elsevier4.php" method="GET">';
echo "<label>The limit of 20 duplicates has been reached. Do you want to
cancel?</label><br />";
echo '<input type="submit" id="but1" name="but1" value="Continue">';
echo '<input type="submit" id="but2" name="but2" value="Cancel">';
echo '</form>';
exit;
```

Εικόνα 8.1.5 Αν οι διπλοεγγραφές ξεπεράσουν το όριο, εμφανίζεται μια φόρμα στον χρήστη

Στην περίπτωση του Elsevier βρήκαμε αρκετές διπλές εγγραφές. Υπήρχαν εγγραφές με το ίδιο DOI (το οποίο πρέπει να είναι μοναδικό) και μερικές με ίδιο article id, το οποίο μας έδινε μια εγγραφή με διπλότυπο primary key.

Αφού ολοκληρωθεί με επιτυχία το script, ενημερώνεται το πεδίο status στον πίνακα session και γίνεται 'Finished', που σημαίνει ότι ολοκληρώθηκε η εκτέλεση.

```
// If finished properly, update the status of the table session
sq6="update session set status='Finished'
    WHERE unix_timestamp=$unixtimestamp";
```

Εικόνα 8.1.6 Ενημέρωση του πεδίου status στον πίνακα session

## 8.2 2ο Script: Έλεγχος για τον ακριβή αριθμό των δεδομένων

Στο δεύτερο script περνάμε όλα τα αρχεία στον φάκελο του εκδότη με αναδρομή, τα αναλύουμε και βγάζουμε τον συνολικό αριθμό περιοδικών, τόμων, τευχών και άρθρων.

Αφού βγάλουμε αυτούς τους αριθμούς, ελέγχουμε την βάση για να δούμε αν πράγματι μπήκαν τόσες εγγραφές.

Για να μπορέσουμε να βγάλουμε τον σωστό αριθμό όσων θέλουμε, πρέπει να κάνουμε ελέγχους κατά την διάρκεια της ανάλυσης, όσον αφορά κυρίως τους τόμους και τα τεύχη των περιοδικών, έτσι ώστε να μην μετρήσουμε π.χ. κάποιον τόμο για δεύτερη φορά.

### 8.2.1. Journals

Όσον αφορά τα περιοδικά, αποθηκεύουμε, στο πρώτο πέρασμα, σε μια μεταβλητή το issn του περιοδικού και στα επόμενα περάσματα ελέγχουμε αν το issn έχει αλλάξει έτσι ώστε να αυξήσουμε τον αριθμό των περιοδικών κατά 1, ή αν έχει μείνει το ίδιο έτσι ώστε να κρατήσουμε τον αριθμό ως έχει.

Εδώ είναι επιτακτικός και ένας 2<sup>ος</sup> έλεγχος. Πρέπει να ελέγξουμε αν το ISSN που έχουμε αυτή τη στιγμή έχει ξανα-εισαχθεί στην βάση σε παλαιότερη φάση.

Για παράδειγμα, αν έχουμε εισάγει ένα ISSN 3000-0000, στην συνέχεια εισάγουμε ένα ISSN 2400-0000 και μετά εισάγουμε πάλι το ISSN 3000-0000, ο 1<sup>ος</sup> έλεγχος θα αυξήσει τον αριθμό των περιοδικών, επειδή το νέο ISSN (3000-0000) είναι διαφορετικό από το προηγούμενο (2400-0000). Το πρόβλημα είναι ότι έτσι θα βάλουμε το ίδιο περιοδικό 2 φορές.

Για να λύσουμε αυτό το πρόβλημα, αποθηκεύουμε σε κάθε πέρασμα το ISSN σε έναν πίνακα και σε κάθε έλεγχο, μέσα σε ένα for loop εξετάζουμε αν υπάρχει ήδη το τρέχων ISSN. Αν υπάρχει, τότε μειώνουμε κατά ένα τον αριθμό των περιοδικών που αυξήθηκε λανθασμένα από τον 1<sup>ο</sup> έλεγχο. Αν δεν υπάρχει, αφήνουμε τον αριθμό αυξημένο.

```
if($issn_no!=$jns) {
    //echo "<H1>Journal_ISSN: </H1>".$issn_no."
    $jn++;
    for($d=0;$d<$entryj;$d++){
        if($issn_no==$jnj_entries[$d]){
            $jn--;
            break;
        }
    }
}
$jnj_entries[$entryj]=$issn_no;
$entryj++;
```

Εικόνα 8.2.1.1 Έλεγχος των περιοδικών



## 8.2.2. VOLUMES

Όσον αφορά τους τόμους, αποθηκεύουμε, στο πρώτο πέρασμα, σε μια μεταβλητή το νούμερο του τεύχους (volume id) και στα επόμενα περάσματα ελέγχουμε αν το volume id έχει αλλάξει έτσι ώστε να αυξήσουμε τον αριθμό των τόμων κατά 1, ή αν έχει μείνει το ίδιο έτσι ώστε να κρατήσουμε τον αριθμό ως έχει.

Εδώ είναι επιτακτικοί τρεις έλεγχοι:

1. Αν το ISSN του περιοδικού είναι διαφορετικό από το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι διαφορετικό από αυτό του προηγούμενου περάσματος, τότε αυξάνουμε τον αριθμό των τόμων κατά 1.
2. Αν το ISSN του περιοδικού είναι διαφορετικό από το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι ίδιο με αυτό του προηγούμενου περάσματος, τότε αυξάνουμε τον αριθμό των τόμων κατά 1.
3. Αν το ISSN του περιοδικού είναι ίδιο με το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι διαφορετικό από αυτό του προηγούμενου περάσματος, τότε αυξάνουμε τον αριθμό των τόμων κατά 1 και στη συνέχεια ελέγχουμε αν υπάρχουν τα ίδια στοιχεία σε παλαιότερες εγγραφές.

```
if($vol_id!=$vols&&$issn_no!=$jns) {
    $vl++;
}
elseif($vol_id==$vols&&$issn_no!=$jns) {
    $vl++;
}
elseif($vol_id!=$vols&&$issn_no==$jns) {
    $vl++;

    // elegchos gia to an o titlos tou journal einai idios kai o arith
    // tote na mhn prostithetai kati ston arithmo twn synolikwn volur
    for($d=0;$d<$entryv;$d++) {
        if($issn_no==$jn_entries[$d]&&$vol_id==$vol_entries[$d]) {
            $vl--;
            break;
        }
    }
    $vol_entries[$entryv]=$vol_id; // kratame ton arithmo tou volume
    $jn_entries[$entryv]=$issn_no; // kratame ton titlo tou journal
    $entryv++;
}
```

Εικόνα 8.2.2.1 Έλεγχος των τόμων

### 8.2.3. ISSUES

Όσον αφορά τα τεύχη, είναι επιτακτικοί 7 έλεγχοι:

1. Αν το ISSN του περιοδικού είναι διαφορετικό από το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι διαφορετικό από αυτό του προηγούμενου περάσματος ΚΑΙ το issue id είναι διαφορετικό από το προηγούμενο, τότε αυξάνουμε τον αριθμό των τευχών κατά 1.
2. Αν το ISSN του περιοδικού είναι διαφορετικό από το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι διαφορετικό από αυτό του προηγούμενου περάσματος ΚΑΙ το issue id είναι ίδιο με αυτό του προηγούμενου, τότε αυξάνουμε τον αριθμό των τευχών κατά 1.
3. Αν το ISSN του περιοδικού είναι διαφορετικό από το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι ίδιο με αυτό του προηγούμενου περάσματος ΚΑΙ το issue id είναι διαφορετικό από το προηγούμενο, τότε αυξάνουμε τον αριθμό των τευχών κατά 1.
4. Αν το ISSN του περιοδικού είναι διαφορετικό από το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι ίδιο με αυτό του προηγούμενου περάσματος ΚΑΙ το issue id είναι ίδιο με το προηγούμενο, τότε αυξάνουμε τον αριθμό των τευχών κατά 1.

```
if($vol_id!=$vols&&$issn_no!=$jns&&$issue_id!=$iss) {  
    $is++;  
}  
elseif($vol_id!=$vols&&$issn_no!=$jns&&$issue_id==$iss) {  
    $is++;  
}  
elseif($vol_id==$vols&&$issn_no!=$jns&&$issue_id!=$iss) {  
    $is++;  
}  
elseif($vol_id==$vols&&$issn_no!=$jns&&$issue_id==$iss) {  
    $is++;  
}  
}
```

Εικόνα 8.2.3.1 Οι 4 πρώτοι έλεγχοι για τα τεύχη

1. Αν το ISSN του περιοδικού είναι ίδιο με το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι διαφορετικό από αυτό του προηγούμενου περάσματος ΚΑΙ το issue id είναι ίδιο με το προηγούμενο, τότε αυξάνουμε τον αριθμό των τευχών κατά 1 και στη συνέχεια ελέγχουμε αν υπάρχουν τα ίδια στοιχεία σε παλαιότερες τιμές 3 πινάκων.
2. Αν το ISSN του περιοδικού είναι ίδιο με το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι διαφορετικό από αυτό του προηγούμενου περάσματος ΚΑΙ το issue id είναι διαφορετικό από αυτό του προηγούμενου περάσματος, τότε αυξάνουμε τον αριθμό των τευχών κατά 1 και στη συνέχεια ελέγχουμε αν υπάρχουν τα ίδια στοιχεία σε παλαιότερες τιμές 3 πινάκων.
3. Αν το ISSN του περιοδικού είναι ίδιο με το ISSN του προηγούμενου περάσματος ΚΑΙ το volume id είναι ίδιο με αυτό του προηγούμενου περάσματος ΚΑΙ το issue id είναι διαφορετικό από το προηγούμενο, τότε αυξάνουμε τον αριθμό των τευχών κατά 1 και στη συνέχεια ελέγχουμε αν υπάρχουν τα ίδια στοιχεία σε παλαιότερες τιμές 3 πινάκων.

```

elseif($vol_id!=$vols&&$issn_no==$jns&&$issue_id==$iss){
    $is++;
    for($d=0;$d<$entryi;$d++){
        if($issn_no==$jni_entries[$d]&&$vol_id==$voli_entries[$d]&&$issue_id==$is_values[$d]){
            $is--;
            break;
        }
    }
}
elseif($vol_id!=$vols&&$issn_no==$jns&&$issue_id!=$iss){
    $is++;
    for($d=0;$d<$entryi;$d++){
        if($issn_no==$jni_entries[$d]&&$vol_id==$voli_entries[$d]&&$issue_id==$is_values[$d]){
            $is--;
            break;
        }
    }
}
elseif($vol_id==$vols&&$issn_no==$jns&&$issue_id!=$iss){
    $is++;
    for($d=0;$d<$entryi;$d++){
        if($issn_no==$jni_entries[$d]&&$vol_id==$voli_entries[$d]&&$issue_id==$is_values[$d]){
            $is--;
            break;
        }
    }
}
}

```

Εικόνα 8.2.3.2 Άλλοι 3 έλεγχοι για τα τεύχη

### 8.3 3ο Script: Έλεγχος εγγραφών

Στο 3<sup>ο</sup> script εμφανίζουμε το σύνολο των εγγραφών που έχουν εισαχθεί στη βάση δεδομένων στο τελευταίο session.

```
echo "Inserted number of journals: ".$jns[0]."</br>";  
echo "Inserted number of volumes: ".$vols[0]."</br>";  
echo "Inserted number of issues: ".$iss[0]."</br>";  
echo "Inserted number of articles: ".$art[0]."</br>";  
echo "Inserted number of duplicates: ".$dupl[0]."</br></br> ";
```

Εικόνα 8.3.1 Η εμφάνιση του συνόλου των εγγραφών

## 8.4 4ο Script: Τυχαίος έλεγχος εγγραφών

Στο 4<sup>ο</sup> script αποθηκεύουμε τις τιμές των κύριων κλειδιών σε 4 διαφορετικούς πίνακες για κάθε πέρασμα του αρχείου.

```
$jn_entries[$entry]="'$issn_no';  
$vol_entries[$entry]=$vol_id;  
$is_values[$entry]=$issue_id;  
$art_values[$entry]=$articl_id;  
  
$entry++;
```

Εικόνα 8.4.1 Αποθήκευση των τιμών των κύριων κλειδιών

Στο τέλος του αρχείου παίρνουμε 10 τυχαίες τιμές των πινάκων και τις συγκρίνουμε με τα αντίστοιχα κύρια κλειδιά στην βάση μας για να δούμε αν υπάρχουν οι εγγραφές. Εμφανίζεται κατάλληλο μήνυμα είτε υπάρχει, είτε δεν υπάρχει η εγγραφή.

Χρησιμοποιούμε την μέθοδο rand() για να πάρουμε τυχαίες τιμές, η οποία παίρνει σαν 1<sup>η</sup> παράμετρο την πιο μικρή τιμή που μπορεί να πάρει και σαν 2<sup>η</sup> την μεγαλύτερη τιμή που μπορεί να πάρει:

```
for ($d=0;$d<10;$d++) {  
    $random=rand(0, count($vol_entries));  
    echo "<h3>value: ".$random."</h3><br>";  
    $q3="SELECT * FROM articles  
        WHERE issn=$jn_entries[$random] AND volume_no=$vol_entries[$random]  
        AND import_time<= (SELECT unix_timestamp FROM import_time ORDER BY  
        AND import_time>=(SELECT unix_timestamp FROM import_time ORDER BY  
  
    //echo "Query: ".$q3."<br>";  
    echo "<b>Journal Issn: </b>".$jn_entries[$random]."<br>";  
    echo "<b>Volume No: </b>".$vol_entries[$random]."<br>";  
    echo "<b>Issue No: </b>".$is_values[$random]."<br>";  
    echo "<b>Article: </b>".$art_values[$random]."<br>";  
    $result3 = mysqli_query($con, $q3);  
    $num_rows3 = mysqli_num_rows($result3);  
  
    if($num_rows3==1) {  
        echo '<h2 style="color:red"> Exists </h2>';  
        echo "=====<br>";  
    }else{  
        echo '<h2 style="color:red"> It does not exist! </h2>';  
        echo "=====<br>";  
    }  
}
```

Εικόνα 8.4.2 Έλεγχος τυχαίων τιμών

## 8.5 5ο Script: Διαγραφή των πιο πρόσφατα εισαγμένων εγγραφών

Διαγράφονται όλες οι εγγραφές στην βάση δεδομένων που ανήκουν στο τελευταίο session με την χρήση του unix timestamp.

Με την ολοκλήρωση του script, εμφανίζονται αντίστοιχες πληροφορίες στον χρήστη.

```
echo $num_rows1." duplicates were deleted<br />";  
echo $num_rows2." articles were deleted<br />";  
echo $num_rows3." issues were deleted<br />";  
echo $num_rows4." volumes were deleted<br />";  
echo $num_rows5." journals were deleted<br /><br />";  
echo $num_rows5." sessions were deleted<br /><br />";
```

Εικόνα 8.5.1 Ενημέρωση για το πόσες εγγραφές διεγράφησαν

## 8.6 6ο Script: Διαγραφή όλων των εγγραφών

Διαγράφονται όλες οι εγγραφές στην βάση δεδομένων.

```
// delete the entries in table duplicates
$q1="DELETE FROM `duplicates`";

// delete the entries in table articles
$q2="DELETE FROM `articles`";

// delete the entries in table issue
$q3="DELETE FROM `issue`";

// delete the entries in table volume
$q4="DELETE FROM `volume`";

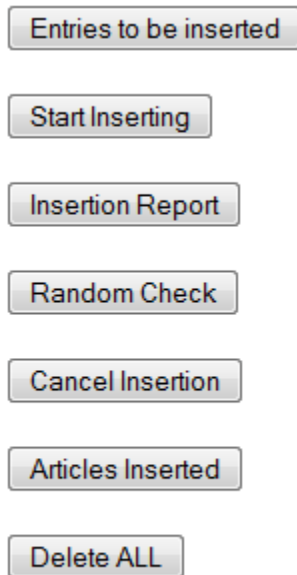
// delete the entries in table journal
$q5="DELETE FROM `journal`";

// delete the entries in table session
$q6="DELETE FROM `session`";
```

Εικόνα 8.6.1 Διαγραφή όλων των εγγραφών

## 8.7 7ο Script: Χρήση AJAX

Έχουμε ένα interface, όπου μας εμφανίζονται διάφορα κουμπιά για να κάνουμε τις αντίστοιχες ενέργειες.



Εικόνα 8.7.1 Το interface για την πραγματοποίηση των διάφορων ενεργειών



Με χρήση του συνδυασμού javascript-ajax με το πάτημα ενός κουμπιού δεν μεταφερόμαστε σε νέα σελίδα. Αντίθετα, μένουμε στην ίδια σελίδα και τα αποτελέσματα εμφανίζονται κάτω από τα κουμπιά.

```
if(window.XMLHttpRequest) {
    xmlhttp = new XMLHttpRequest();
} else {
    xmlhttp = new ActiveXObject('Microsoft.XMLHTTP')
}

xmlhttp.onreadystatechange = function() {
    if(xmlhttp.readyState == 1) {
        document.getElementById('content').innerHTML =
    }
    if(xmlhttp.readyState == 4 && xmlhttp.status ==
        document.getElementById('content').innerHTML =
    }
}

xmlhttp.open('GET', 'elsevier.php', true);
xmlhttp.setRequestHeader('Content-type','application/x-www-form-urlencoded');
xmlhttp.send(null);
```

Εικόνα 8.7.2 Ο συνδυασμός javascript-ajax

Όσο περιμένουμε να εκτελεστεί το script μας, εμφανίζεται ένα εικονίδιο φόρτωσης που δείχνει ότι το script εκτελείται.

# Κεφάλαιο 9. Μελέτη και σύγκριση CMS

## 9.1 Τι είναι ένα CMS

CMS ή Σύστημα Διαχείρισης Περιεχομένου (Content Management System) ονομάζουμε ένα λογισμικό που μας βοηθά να ελέγχουμε και να διαχειριζόμαστε έναν ιστότοπο δημόσιας ή περιορισμένης πρόσβασης. Περιεχόμενο είναι οτιδήποτε αποφασίζουμε να «ανεβάσουμε» στον ιστότοπό μας: κείμενα, φωτογραφίες, μουσική, βίντεο, έγγραφα ή οποιουδήποτε άλλου είδους αρχείο. Ένα ιδανικά σχεδιασμένο CMS μας επιτρέπει να διαχειριζόμαστε τον ιστότοπό μας χωρίς να διαθέτουμε περίπλοκες τεχνικές γνώσεις ή δεξιότητες.

### 9.1.1 Από τι αποτελείται

Τυπικά, ένα CMS εμπεριέχεται από δύο συστατικά: την εφαρμογή διαχείρισης περιεχομένου (Content Management Application) και την εφαρμογή παράδοσης περιεχομένου (Content Delivery Application).

Το πρώτο συστατικό, CMA, επιτρέπει στο διαχειριστή περιεχομένου ή συγγραφέα, ο οποίος μπορεί να μη γνωρίζει HTML γλώσσα (HyperText Markup Language), να διαχειρίζεται τη δημιουργία, τροποποίηση και απομάκρυνση περιεχομένου από έναν ιστοχώρο, χωρίς να απαιτείται να έχει γνώσεις ενός Webmaster. Το δεύτερο συστατικό, CDA, χρησιμοποιεί και μεταγλωττίζει αυτή την πληροφορία για να ενημερώσει τον ιστοχώρο.

### 9.1.2 Τα χαρακτηριστικά ενός CMS

Τα χαρακτηριστικά ενός CMS συστήματος ποικίλουν, αλλά τα περισσότερα αφορούν web-based δημοσιοποίηση, διαχείριση τυποποίησης, έλεγχο βελτιώσεων, τοποθέτηση δεικτών, έρευνα και ανάκτηση.

Η web-based δημοσιοποίηση επιτρέπει στον καθένα να χρησιμοποιεί ένα μοτίβο (template) ή ένα σύνολο από μοτίβα εγκεκριμένα από τον οργανισμό, καθώς επίσης οδηγούς (wizards) και άλλα εργαλεία για τη δημιουργία ή την τροποποίηση web περιεχομένου.

Η διαχείριση τυποποίησης επιτρέπει στα έγγραφα, συμπεριλαμβανομένου των νομικών ηλεκτρονικών εγγράφων και των σκαναρισμένων εγγράφων, να τυποποιούνται σε HTML ή PDF (Portable Document Format) μορφή για να μπορούν να προβληθούν στον ιστοχώρο. Ο έλεγχος βελτιώσεων επιτρέπει την ενημέρωση του περιεχομένου με καινούρια έκδοση ή την επαναφορά μιας προηγούμενης έκδοσης. Ο έλεγχος βελτιώσεων παρακολουθεί κάθε αλλαγή που γίνεται στα αρχεία από οποιονδήποτε.

Όσον αφορά το τελευταίο χαρακτηριστικό, δηλαδή την τοποθέτηση δεικτών, την έρευνα και την ανάκτηση, ένα CMS σύστημα τοποθετεί δείκτες σε όλα τα δεδομένα του οργανισμού. Έπειτα, ο καθένας μπορεί να ψάξει για δεδομένα χρησιμοποιώντας λέξεις κλειδιά και έτσι τελικά το CMS σύστημα τα ανακτά.

### 9.1.3 Το περιεχόμενο

Το διαχειριζόμενο περιεχόμενο αφορά αρχεία υπολογιστών, εικόνες, αρχεία ήχου, ηλεκτρονικά έγγραφα και web περιεχόμενο. Η ιδέα πίσω από ένα CMS είναι να καταστήσει αυτά τα αρχεία διαθέσιμα μεταξύ των ατόμων ενός γραφείου καθώς και μεταξύ ατόμων σε όλο το διαδίκτυο.

Το CMS χρησιμοποιείται ως μέσο αρχειοθέτησης και πολλές εταιρείες το χρησιμοποιούν για να αποθηκεύουν αρχεία σε μια μη-ιδιοκτησιακή μορφή. Οι εταιρείες χρησιμοποιούν επίσης ένα CMS για να διαμοιράζονται αρχεία με ευκολία, καθώς τα περισσότερα συστήματα χρησιμοποιούν λογισμικό βασισμένο σε servers.

Πολλά CMS περιλαμβάνουν ένα χαρακτηριστικό για Web Content και μερικά έχουν ένα χαρακτηριστικό για «διαδικασία ροής εργασίας» (“workflow process”).

### 9.1.4 Χρήση των CMS

Τα συστήματα διαχείρισης περιεχομένου χρησιμοποιούνται για αποθήκευση, έλεγχο, έκδοση και δημοσιοποίηση industry-specific εγγράφων όπως είναι τα άρθρα για νέα, εγχειρίδια χρήσης, sales οδηγοί και φυλλάδια μάρκετινγκ. Ένα CMS μπορεί να υποστηρίξει τα παρακάτω:

- Εισαγωγή και δημιουργία εγγράφων και πολυμεσικού υλικού.
- Ταυτοποίηση όλων των σημαντικών χρηστών και των ρόλων τους στο σύστημα διαχείρισης.
- Η ικανότητα για ανάθεση ρόλων και ευθυνών σε διαφορετικές κατηγορίες περιεχομένου ή τύπων.
- Προσδιορισμός των καθηκόντων ροής εργασίας του περιεχομένου, συχνά μαζί με ειδοποίηση γεγονότων έτσι ώστε οι διαχειριστές περιεχομένου να ειδοποιούνται έγκαιρα για αλλαγές στο περιεχόμενο.
- Η ικανότητα για εντοπισμό και διαχείριση πολλαπλών εκδόσεων ενός συγκεκριμένου instance ενός περιεχομένου.
- Η ικανότητα για δημοσιοποίηση του περιεχομένου σε ένα repository για να υποστηριχτεί η πρόσβαση στο περιεχόμενο. Το repository γίνεται όλο και περισσότερο ένα έμφυτο κομμάτι του συστήματος και ενσωματώνει εταιρική αναζήτηση και ανάκτηση.
- Μερικά συστήματα διαχείρισης περιεχομένου επιτρέπουν textual άποψη του περιεχομένου για να διαχωρίζεται από την τυποποίηση. Για παράδειγμα, το CMS μπορεί να θέτει αυτόματα χρώματα, γραμματοσειρές και διατάξεις σελίδας.

Γενικά ένα CMS μπορεί να χρησιμοποιηθεί για:

- Ειδήσεις (εφημερίδες, περιοδικά, πρακτορεία ειδήσεων κ.λπ.)
- Παρουσιάσεις εταιριών και προσωπικού
- Καταλόγους προϊόντων
- Παρουσιάσεις προϊόντων
- Online υποστήριξη
- Αγγελίες και ανακοινώσεις
- Παρουσιάσεις και προβολή γεωγραφικών περιοχών
- Διαφημίσεις
- Δελτία Τύπου
- Όρους και συμβόλαια
- Χάρτες, κατευθύνσεις, οδηγίες
- Image Galleries

### 9.1.5 Οι στόχοι ενός Συστήματος Διαχείρισης Διαδικτυακού Περιεχομένου (WCMS)

Οι περισσότεροι οργανισμοί αναγνωρίζουν την αναγκαιότητα ενός Συστήματος διαχείρισης διαδικτυακού Περιεχομένου (WCM) στην διαχείριση ενός μεγάλου εταιρικού ιστοχώρου ή ενδοδικτύου. Όμως η γενική ανάγκη για μια λύση WCM θα πρέπει να περιγραφεί σωστά με την θέσπιση συγκεκριμένων επιχειρηματικών στόχων. Εάν οι στόχοι που πρέπει να επιτευχθούν με την εφαρμογή ενός Συστήματος διαχείρισης διαδικτυακού Περιεχομένου δεν είναι ξεκάθαροι, θα είναι αδύνατον να υπολογισθεί η επιτυχία του όλου εγχειρήματος ενώ το πιο πιθανό είναι να μη γίνει τελικά πλήρης εκμετάλλευση των πλεονεκτημάτων που προσφέρει η εκάστοτε εφαρμογή.

Για τον λόγο αυτό είναι απαραίτητη η αναγνώριση των στόχων που καλείται να επιτύχει η επιχείρηση με την εφαρμογή ενός Συστήματος διαχείρισης διαδικτυακού Περιεχομένου.

Παρακάτω παρατίθενται οι κυριότεροι επιχειρηματικοί στόχοι που αναμένεται να πετύχει ένα Σύστημα διαχείρισης διαδικτυακού Περιεχομένου σε αντιστοιχία με τα προβλήματα ή τις ελλείψεις που παρουσιάζονται σε εφαρμογές εταιρικών ιστοχώρων ή εταιρικών ενδοδικτύων που δεν χρησιμοποιούν λύσεις WCM :

- **Υποστήριξη για αποκεντρωμένη συγγραφή:** Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να υποστηρίξει τον καταμερισμό ευθυνών για την συγγραφή περιεχομένου ώστε ο κάθε χρήστης να μπορεί να συντάσσει και να ανανεώνει το δικό του περιεχόμενο χωρίς να χρειάζεται να καλεί συνέχεια το τμήμα IT της επιχείρησης. Επίσης, το περιεχόμενο θα πρέπει να αποστέλλεται αυτόματα στους υπεύθυνους επιθεωρητές ώστε να εγκριθεί η δημοσίευσή του.
- **Απλούστευση της διαδικασίας συγγραφής και ανανέωσης του περιεχομένου:** Η παρούσα χειροκίνητη διαδικασία συγγραφής και ανανέωσης του περιεχομένου του ιστοχώρου είναι αργή και αναποτελεσματική. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να την διορθώσει ώστε να υποστηρίζεται η γρήγορη και απλή συγγραφή και ανανέωση της πληροφορίας σε ολόκληρο τον ιστοχώρο.
- **Βελτίωση της διαδικασίας δημοσίευσης περιεχομένου:** Οι επί τούτου (Ad-Hoc) διαδικασίες δημοσίευσης εμποδίζουν την αποτελεσματική διαχείριση και εντοπισμό δημοσιευμένου περιεχόμενου. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να εξαλείψει αυτό το πρόβλημα καθώς και να προσφέρει μεγαλύτερη διαφάνεια και λογοδοσία.

- **Αυξημένη ασφάλεια:** Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να παρέχει άριστη ασφάλεια περιεχομένων. Πρέπει να ελέγχεται απόλυτα ποιος μπορεί να δημοσιεύσει στον ιστοχώρο και ποιος μπορεί να διαβάσει την κάθε πληροφορία.
- **Καλύτερη συνοχή κατά την λειτουργία:** Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να εξαλείψει οποιαδήποτε απρόβλεπτη αντίδραση του συστήματος και να εξασφαλίσει την ομαλή λειτουργία των διαδικασιών ώστε να μειωθούν οι συχνές κλήσεις στο τμήμα IT της επιχείρησης.
- **Αύξηση της ευελιξίας του ιστοχώρου:** Εταιρικοί ιστοχώροι πρέπει να μπορούν να προσαρμόζονται γρήγορα ώστε να υποστηρίζουν την εμφάνιση νέων προϊόντων, υπηρεσιών ή εταιρικών στρατηγικών. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να υποστηρίζει την εύκολη αναδόμηση καθώς και τον εύχρηστο επανασχεδιασμό των διαμεσολαβητών. Αυτό περιλαμβάνει την δυνατότητα εύκολης ανανέωσης όλων των σελίδων ώστε να αντικατοπτριστεί μια τυχόν αλλαγή στην εταιρική ταυτότητα ή την εικόνα της εταιρίας.
- **Αύξηση της ακρίβειας της πληροφορίας:** Η ολική ποιότητα της πληροφορίας πρέπει να βελτιωθεί, τόσο στον ιστοχώρο όσο και στο ενδοδίκτυο. Όλες οι σελίδες θα πρέπει να είναι ακριβείς, ενημερωμένες και περιεκτικές.
- **Μείωση της διπλής εγγραφής πληροφορίας:** Η διπλή εγγραφή πληροφορίας στις επιχειρηματικές μονάδες και πλατφόρμες αυξάνει τα κόστη συντήρησης και το ποσοστό λαθών. Όποτε είναι δυνατόν, θα πρέπει η πληροφορία να αποθηκεύεται μία φορά και να χρησιμοποιείται πολλές.
- **Σύλληψη της επιχειρησιακής γνώσης:** Είναι γενικά γνωστό πως όταν φεύγει ένας υπάλληλος που βρίσκεται σε σημαντική θέση, μειώνεται η γνώση που είναι διαθέσιμη μέσα στον οργανισμό. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να υποστηρίζει την σύλληψη και καταγραφή αυτής της πληροφορίας.
- **Βελτίωση της ανταλλαγής γνώσης:** Η απ' ευθείας επικοινωνία μεταξύ του προσωπικού καθώς και η ανταλλαγή της πληροφορίας με την μέθοδο «peer-to-peer» αποτελούν δύο από τους πιο αποτελεσματικούς τρόπους για την κατανομή της γνώσης. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να παρέχει ένα κατάλληλο περιβάλλον και τα αντίστοιχα εργαλεία για να εξυπηρετούνται αυτές οι διαδικασίες.
- **Υποστήριξη της ανακάλυψης γνώσης:** Το προσωπικό συχνά αντιμετωπίζει έναν υπερβολικό φόρτο πληροφοριών. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου μπορεί να βοηθήσει το προσωπικό να εντοπίσει σημαντικές πληροφορίες με την χρήση ισχυρών μηχανών αναζήτησης, ξεφυλλίσματος και φιλτραρίσματος.

- **Βελτίωση της αποτελεσματικότητας του προσωπικού:** Η αποτελεσματικότητα του προσωπικού μπορεί να βελτιωθεί με την παροχή αρκετών πληροφοριών σε σημαντικές επιχειρησιακές διαδικασίες. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να στοχεύει στην προμήθεια του προσωπικού με πληροφορία που χρειάζεται, όταν την χρειάζεται. Αυτό μεταφράζεται άμεσα σε οικονομία χρόνου μέσω της αποφυγής άκαρπων ερευνών για τις απαιτούμενες πληροφορίες.
- **Μείωση της ποινικής έκθεσης:** Όλες οι πληροφορίες που παρουσιάζονται στον εταιρικό ιστοχώρο εκθέτουν την επιχείρηση σε ποινική ευθύνη. Αυτό θα πρέπει να ελαττωθεί εφαρμόζοντας μεγαλύτερο έλεγχο και λογοδοσία στις διαδικασίες της επιθεώρησης και της δημοσίευσης περιεχομένου.
- **Βελτίωση της πλοήγησης του ιστοχώρου:** Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να διευκολύνει την πλοήγηση του χρήστη είτε πρόκειται για συγγραφέα περιεχομένου είτε για απλό επισκέπτη ώστε να είναι εύκολος ο εντοπισμός του επιθυμητού περιεχομένου.
- **Βελτίωση της επιχειρηματικής δεκτικότητας:** Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να υποστηρίζει την ανάπτυξη νέων προϊόντων και υπηρεσιών καθώς και άλλων αλλαγών στην εταιρική κατεύθυνση. Αυτό μπορεί να επιτευχθεί με την χρήση ενός γρήγορου και αποτελεσματικού μηχανισμού για την ανανέωση των εσωτερικών εταιρικών πληροφοριών και πόρων.
- **Βελτίωση της εμπειρίας του πελάτη:** Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να βελτιώσει την ικανότητα του ιστοχώρου στο να παρέχει ένα πλούσιο διαδραστικό περιβάλλον για τους επισκέπτες. Αυτό θα περιλαμβάνει ενίσχυση της ποιότητας του ιστοχώρου καθώς και ευχρηστία.
- **Υποστήριξη του Marketing:** Οι ιστοχώροι έχουν γίνει βασικά κανάλια marketing για τις επιχειρήσεις. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να διευκολύνει την επικοινωνία του marketing καθώς και να υποστηρίζει την τρέχουσα εταιρική ταυτότητα.
- **Υποστήριξη των πωλήσεων :** Οι πωλήσεις του ηλεκτρονικού εμπορίου (E-commerce) μεγαλώνουν σταθερά και το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να παρέχει επιπλέον λειτουργίες για τις πωλήσεις. Επίσης θα πρέπει να τελειοποιήσει την παρούσα υποδομή του ηλεκτρονικού εμπορίου.
- **Αύξηση του κοινού του ιστοχώρου :** Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να επιτρέπει την πρόσβαση στον ιστοχώρο από ένα ευρύ κοινό, ανεξάρτητα από την μητρική τους γλώσσα. Αυτό επιβάλλει την παρουσίαση του περιεχόμενου σε περισσότερες από μία γλώσσες (η μία εκ των οποίων θα πρέπει να είναι και η αγγλική).

- **Μείωση του κόστους υποστήριξης πελατών** : Οι απαιτήσεις για υποστήριξη των πελατών θα πρέπει να μειωθούν. Αυτό θα επιτευχθεί εάν προσφέρονται περισσότερο ακριβείς και ολοκληρωμένες πληροφορίες στους πελάτες.
- **Μείωση του κόστους δημοσίευσης** : Πολλά εγχειρίδια επιχειρήσεων τυπώνονται ακόμα σε χαρτί. Μπορούν να μειωθούν άμεσα τα κόστη εάν αντικατασταθούν τα έντυπα εγχειρίδια με αντίστοιχα διαδικτυακά.
- **Μείωση του κόστους συντήρησης ιστοχώρου** : Με την αντικατάσταση των τωρινών κουραστικών διαδικασιών συντήρησης το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα μειώσει την ανάγκη για στελέχη που ασχολούνται με την διαχείριση του ιστοχώρου και άλλα σχετιζόμενα κόστη.
- **Υποστήριξη της ανάπτυξης του ιστοχώρου** : Υπάρχει μια στρατηγική ανάγκη για την ουσιαστική αύξηση της ποσότητας της πληροφορίας που δημοσιεύεται σε εταιρικούς ιστοχώρους. Το Σύστημα διαχείρισης διαδικτυακού Περιεχομένου θα πρέπει να παρέχει πλεονεκτήματα αποτελεσματικότητας και διαχείρισης ώστε να υποστηρίζεται η ανάπτυξη του ιστοχώρου.

### 9.1.6 Πλεονεκτήματα Συστημάτων Διαχείρισης Περιεχομένου

Ένα ολοκληρωμένο CMS πρέπει να μπορεί να διαχειρίζεται όλες τις δυναμικές πληροφορίες του site και να προσφέρει υπηρεσίες που εξυπηρετούν πλήρως τις ανάγκες των διαχειριστών του.

Επιγραμματικά, μερικά από τα πλεονεκτήματα και τα χαρακτηριστικά ενός ολοκληρωμένου CMS είναι:

- Γρήγορη ενημέρωση, διαχείριση και αρχειοθέτηση του περιεχομένου του δικτυακού τόπου.
- Ενημέρωση του περιεχομένου από οπουδήποτε.
- Ταυτόχρονη ενημέρωση από πολλούς χρήστες και διαφορετικούς υπολογιστές.
- Να μην απαιτούνται ειδικές τεχνικές γνώσεις από τους διαχειριστές του
- Εύκολη χρήση και άμεση γνώση του τελικού αποτελέσματος, όπως γίνεται με τους γνωστούς κειμενογράφους.
- Δυνατότητα αναζήτησης του περιεχομένου που καταχωρείται και αυτόματη δημιουργία αρχείου.
- Ασφάλεια και προστασία του σχεδιασμού του site από λανθασμένες ενέργειες, που θα μπορούσαν να δημιουργήσουν προβλήματα στην εμφάνισή του.
- Διαχωρισμός του περιεχομένου από το σχεδιασμό και την πλοήγηση (navigation) του δικτυακού τόπου.
- Αλλαγή σχεδιασμού ή τρόπου πλοήγησης χωρίς να είναι απαραίτητη η ενημέρωση όλων των σελίδων από τον ίδιο το χρήστη.



- Αυτόματη δημιουργία των συνδέσμων μεταξύ των σελίδων και αποφυγή προβλημάτων ανύπαρκτων σελίδων (404 error pages).
- Μικρότερος φόρτος στον εξυπηρετητή (server) και χρήση λιγότερου χώρου, αφού δεν υπάρχουν πολλές επαναλαμβανόμενες στατικές σελίδες, από τη στιγμή που η ανάπτυξη των σελίδων γίνεται δυναμικά.
- Όλο το περιεχόμενο καταχωρείται στην/στις βάσεις δεδομένων, τις οποίες μπορούμε πιο εύκολα και γρήγορα να τις προστατεύσουμε τηρώντας αντίγραφα ασφαλείας.

Ασφαλώς υπάρχουν και άλλα χαρακτηριστικά και πρόσθετες υπηρεσίες, ανάλογα με το **CMS**, που άλλοτε χρεώνονται επιπλέον και άλλοτε ενσωματώνονται και προσφέρονται δωρεάν προς χρήση, όπως:

- Εφαρμογή διαχείρισης και προβολής διαφημιστικών banners, δημοσκοπήσεων και παραμετροποίησης (personalisation)
- Δυνατότητα παρουσίασης του περιεχομένου σε συνεργαζόμενα sites
  1. Στατιστικά
  2. Διαχείριση μελών
  3. Newsletters
  4. Forum

## 9.2 Σύγκριση

Υπάρχουν εκατοντάδες Συστήματα Διαχείρισης Περιεχομένου (CMS) στον κόσμο. Τα πιο δημοφιλή συστήματα που χρησιμοποιούνται στο ελληνικό διαδίκτυο είναι κυρίως "κλειστά" συστήματα .ASP και το Joomla CMS ανοιχτού κώδικα.

Τα "κλειστά" συστήματα διαχείρισης περιεχομένου συρρικνώνονται εδώ και αρκετά χρόνια, τόσο λόγω μικρής βάσης ανάπτυξης και υποστήριξης, όσο και τεχνολογικής δυσκινησίας. Τα επιχειρηματικά μοντέλα πώλησης "αδειών χρήσης" είναι πια ξεπερασμένα και με ιδιαίτερη χαρά βλέπουμε πως η Ελλάδα σιγά-σιγά υιοθετεί τον ανοιχτό κώδικα, όπως και ο υπόλοιπος κόσμος.

Οι μεγαλύτεροι **δημόσιοι** και **ιδιωτικοί** φορείς σε όλο τον κόσμο αναθεωρούν τα "κλειστά" συστήματα διαχείρισης περιεχομένου και τα αναβαθμίζουν σε σύγχρονα συστήματα ανοικτού κώδικα.

### 9.2.1 Πλεονεκτήματα και μειονεκτήματα διαφόρων CMS

Εδώ παραθέτω τα πλεονεκτήματα και μειονεκτήματα από κάποια από τα πιο γνωστά CMS (Joomla!, Drupal, WordPress, TYPO3)..

#### Πλεονεκτήματα της Joomla:

- ♦ Απλή εγκατάσταση
- ♦ Έτοιμα εμπορικά θέματα
- ♦ Επιλογή παρόχου - Οι περισσότερες εταιρείες διαφημιστικών ιστοσελίδων στην Ελλάδα χρησιμοποιούν Joomla

#### Μειονεκτήματα της Joomla:

- ♦ Αδυναμία πραγματικής ενσωμάτωσης διαδικτυακών εφαρμογών
- ♦ Αδύνατη υποστήριξη SEO
- ♦ Φτωχή διαχείριση πολυγλωσσικότητας
- ♦ Πρωτόγονη διαχείριση αδειών πρόσβασης
- ♦ Περιορισμένη κατηγοριοποίηση/ταξινόμηση περιεχομένου
- ♦ Απουσία Enterprise δυνατοτήτων (αναθεώρηση περιεχομένου, caching, NBS κτλ)

### **Πλεονεκτήματα του Drupal:**

- ♦ Ελεύθερο λογισμικό - δεν χρειάζεται άδεια για την εγκατάσταση ή τη χρήση του
- ♦ Μηδενικά ετήσια τέλη χρήσης - ο προϋπολογισμός συγκεντρώνεται για τη συνεχή υποστήριξη του συστήματος
- ♦ Απεριόριστη επεκτασιμότητα και ευελιξία
- ♦ Μεγάλη κοινότητα που υποστηρίζει και αναπτύσσει το Drupal
- ♦ Μείωση του χρόνου ανάπτυξης σε σχέση με ιδιωτικούς φορείς παροχής και συστήματα
- ♦ Βελτίωση της ασφάλειας μέσω συνεχούς ενημέρωσης εκδόσεων
- ♦ Ανεξαρτησία - ελεύθερη επιλογή παροχών και συντηρητών συστήματος
- ♦ Τεράστια ποικιλία εφαρμογών για εμπορικούς, διαφημιστικούς και κοινωνικούς ιστοτόπους

### **Μειονεκτήματα του Drupal:**

- ♦ Δεν είναι φιλικό προς τον σχεδιαστή και τον απλό χρήστη. Για να κάνεις λίγο πιο σύνθετα πράγματα είναι απαραίτητη η γνώση κώδικα.
- ♦ Τα διαθέσιμα θέματα εμφάνισης μειονεκτούν δραματικά σε σχέση με των ανταγωνιστών.
- ♦ Η δημοσίευση ενός ιστότοπου με Drupal μπορεί να στοιχίσει περισσότερο χρόνο και χρήμα σε σχέση με το WordPress ή το Joomla!

### **Πλεονεκτήματα του Wordpress:**

- ♦ Απλή εγκατάσταση
- ♦ Απλές επεκτάσεις και θέματα
- ♦ Δωρεάν υπηρεσίες φιλοξενίας
- ♦ Επαρκές SEO

### **Μειονεκτήματα του Wordpress:**

- ♦ Δύσχρηστο API - Ανεπαρκές για ανάπτυξη πολυπλοκότερων λύσεων
- ♦ Προσανατολισμένο για διαχείριση ιστολογίου - Χρονική διάταξη περιεχομένου
- ♦ Ακατάλληλο για κάθετες λύσεις
- ♦ Απουσία Enterprise δυνατοτήτων

### **Πλεονεκτήματα του TYPO3:**

- ♦ Δυνατό API
- ♦ Ισχυρές Enterprise δυνατότητες
- ♦ Ισχυρό caching σε επίπεδο περιεχομένου
- ♦ Ισχυρό σύστημα αδειών πρόσβασης
- ♦ Typoscript

### **Μειονεκτήματα του TYPO3:**

- ♦ Δύσχρηστο περιβάλλον διαχείρισης
- ♦ Πολλαπλές βιβλιοθήκες Javascript με προβλήματα συμβατότητας
- ♦ Ελάχιστες πολυχρηστικές/κοινοτικές δυνατότητες
- ♦ Αδύνατες φόρμες εισαγωγής δεδομένων

# Κεφάλαιο 10. Συμπεράσματα και Μελλοντικές Προοπτικές

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Με το λογισμικό που αναπτύχθηκε, γίνεται εφικτός ο έλεγχος όλων των δεδομένων, τόσο των πιο πρόσφατων, όσο και παλαιότερων. Δίνεται πλέον η δυνατότητα ο έλεγχος αυτός να γίνεται συστηματικά στο μέλλον κάθε φορά που παραλαμβάνεται ψηφιακό υλικό από τους εκδότες.

Επιπλέον εφόσον τα αρχεία θα είναι φορτωμένα μπορεί να γίνεται περιστασιακά έλεγχος για την ακεραιότητα των δεδομένων και των μέσων αποθήκευσης, καθώς επίσης γίνεται και πιο εύκολη η μεταφορά σε νεώτερης γενιάς τεχνικό εξοπλισμό όταν παραστεί η ανάγκη.

Τέλος, τα δεδομένα είναι πλέον ομογενοποιημένα και μπορούμε να τα αποθηκεύσουμε όλα σε μία βάση δεδομένων για χρήση από οποιονδήποτε μέσω κάποιου web interface.

## ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΟΠΤΙΚΕΣ

Επομένως με την ανάπτυξη αυτού του λογισμικού παρίσταται η ανάγκη δημιουργίας ενός web interface για την διάθεση όλων αυτών των πληροφοριών σε όλους τους χρήστες που έχουν πρόσβαση σε ακαδημαϊκά εκπαιδευτικά ιδρύματα.

# Βιβλιογραφία

Χρησιμοποιήθηκαν οι παρακάτω σύνδεσμοι :

## AJAX

<http://www.wlearn.gr/index.php/ajax-intro2>

## Apache HTTP Server Project

<http://httpd.apache.org>

[http://en.wikipedia.org/wiki/Apache\\_web\\_server](http://en.wikipedia.org/wiki/Apache_web_server)

<http://mytwocents.gr/apache-web-server>

## CMS

<http://el.wikipedia.org/wiki/Drupal>

<http://www.cmsmatrix.org/matrix/cms-matrix>

<http://help.joomla.org/>

<http://tech.blorge.com/Structure:%20/2007/03/31/cms-web-design-mambo-joomla-ordrupal/>

<http://www.joomla.com>

## DATABASE

[http://anamorfosi.teiser.gr/ekp\\_yliko/e-notes/Data/database/main.htm](http://anamorfosi.teiser.gr/ekp_yliko/e-notes/Data/database/main.htm)

## DOM

<http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-DOM.html>

## JAVASCRIPT

<http://studentguru.gr/w/tutorials/a0-javascript-javascript-html.aspx>

## PHP

<http://www.php.net>

<http://gr2.php.net/tut.php>

<http://el.wikipedia.org/wiki/PHP>

<http://php.chinaunix.net/manual/el/intro-whatcando.php>

## PhpMyAdmin

[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)

<http://phpmyadmin.sourceforge.net/documentation>

## SQL

<http://el.wikipedia.org/wiki/SQL>

<http://el.wikipedia.org/wiki/MySQL>

[http://195.251.218.39/tm/wiki/index.php/Βάση\\_δεδομένων\\_DBMS\\_\(MySQL\)](http://195.251.218.39/tm/wiki/index.php/Βάση_δεδομένων_DBMS_(MySQL))

<http://en.wikipedia.org/wiki/MySQLi>

## W3Schools

<http://www.w3schools.com/>

## XML

<http://www.it.uom.gr/project/xml/Home%20Page.htm>