

Α.Ε.Ι. ΠΕΙΡΑΙΑ Τ.Τ.

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΩΝ

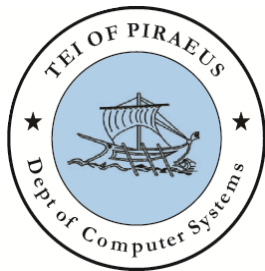
**ΚΑΜΑΡΙΝΟΥ ΣΤΑΥΡΟΥΛΑ
ΜΕΪΜΑΡΗ ΜΑΡΙΑ**

Επιβλέπων καθηγητής:

Αθήνα, Οκτώβριος 2016

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική
γλώσσα

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα



**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

«Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για την νοηματική γλώσσα.»

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΩΝ

ΚΑΜΑΡΙΝΟΥ ΣΤΑΥΡΟΥΛΑ ΜΕΪΜΑΡΗ ΜΑΡΙΑ

Επιβλέπων καθηγητής :

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

Αθήνα, Οκτώβριος 2016

(Υπογραφή)

(Υπογραφή)

.....

.....

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΩΝ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένες Μειμάρη Μαρία, του Γεωργίου, με αριθμό μητρώου 39196 και Καμαρινού Σταυρούλα, του Κωνσταντίνου, με αριθμό μητρώου 40273 φοιτήτριες του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

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

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

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

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

ΕΥΧΑΡΙΣΤΙΕΣ

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

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

ΠΕΡΙΛΗΨΗ

Η νοηματική γλώσσα είναι η φυσική γλώσσα της κοινότητας των Κωφών. Η ιδιαιτερότητά της συγκριτικά με τις υπόλοιπες γλώσσες, είναι πως οι κανόνες βάσει των οποίων επιτυγχάνεται η επικοινωνία, δεν είναι προφορικοί, αλλά οπτικο-κινησιακοί. Πρόκειται για ένα αυτόνομο γλωσσικό σύστημα που μπορεί να αναλυθεί όπως και κάθε άλλη φυσική γλώσσα. Το εύρος των ατόμων που έχουν επαφή με τη νοηματική γλώσσα είναι σχετικά μικρό. Συγκεκριμένα, την γνωρίζουν όσοι έχουν στο οικογενειακό τους περιβάλλον κωφά άτομα, φιλόλογοι, κοινωνικοί λειτουργοί, νοσηλευτές, εκπαιδευτικοί καθώς και άτομα που ενδιαφέρονται για την ιδιαίτερη αυτή γλώσσα.

Αφορμή για την πραγματοποίηση της έρευνας και συγγραφής αυτής της εργασίας αποτέλεσε ο προβληματισμός και το ενδιαφέρον μας για την έλλειψη γλωσσικών πόρων της ελληνικής νοηματικής γλώσσας (ΕΝΓ). Προκειμένου να καταρτιστεί ένα βασικό ηλεκτρονικό λεξιλόγιο, προσιτό σε όλες τις ομάδες ανθρώπων, μιας μη πλήρως καταγεγραμμένης γλώσσας όπως η ΕΝΓ, απαιτείται συγκεκριμένη προεργασία συλλογής σώματος γλωσσικών δεδομένων, ικανών να υποστηρίξουν, αφενός την εξαγωγή ενός πυρήνα λέξεων και αφετέρου την ανάλυση συχνοτήτων εμφάνισης των στοιχείων, που απαρτίζουν τον τελικό κατάλογο σε συνθήκες πραγματικής χρήσης. Πιο συγκεκριμένα, μπορεί κανείς εύκολα να παρατηρήσει μια σειρά ελλείψεων που κυρίως εντοπίζονται στην οργάνωση των λημμάτων, έχει γίνει μόνο με βάση τις χειρομορφές, όχι όμως και το χώρο νοηματισμού που μπορεί εξίσου να παρέχει έναν τρόπο κατάταξης των νοημάτων.

Για την υλοποίηση της επιλέξαμε μια web εφαρμογή γιατί έχει κάποια ελκυστικά χαρακτηριστικά όπως :

- ✓ Παρέχει ευκολότερη πρόσβαση από χρήστες κάθε είδους.
- ✓ Παρέχει ανεξαρτησία από το λειτουργικό σύστημα του χρήστη .
- ✓ Δεν απαιτούνται εγκαταστάσεις και ενημερώσεις(updates) από τον χρήστη
- ✓ Παρέχει ταυτόχρονη εξυπηρέτηση πολλών χρηστών.

Η υλοποίηση της web εφαρμογής θα γίνει με την χρήση ήδη έτοιμων

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

τεχνολογιών όπως το open source CMS Drupal. Επιλέχτηκε το drupal, καθώς εκτός από ένα CMS, είναι και ένα πλήρες framework για την ανάπτυξη διαδικτυακών εφαρμογών. Μας δίνει τη δυνατότητα ανάπτυξης των δικών μας modules (αρθρώματα), κάτι το οποίο το κάνει ευέλικτο και θα μειώσει τον χρόνο ανάπτυξης. Η αποθήκευση των δεδομένων θα είναι σε μια σχεσιακή βάση δεδομένων (DBMS).

Το τελικό αποτέλεσμα θέλουμε να είναι μια web εφαρμογή, λειτουργική και διαχειρίσιμη από χρήστες, μη σχετικούς με τον κλάδο της πληροφορικής.

Λέξεις κλειδιά: «Ελληνική νοηματική γλώσσα, χειρομορφές, CMS, Drupal, MySQL, εργαλεία διαχείρισης Βάσεων Δεδομένων, functions»

ABSTRACT

The sign language is the natural language of the Deaf community. The specificity compared to other languages, is that the rules by which communication is achieved, are not verbal, but opto-motional. It's about a unique language system, which can be analyzed like any other language. The number of people related to this kind of language is rather small. Specifically, those who interact with the deaf in their own families, scholars, social workers, nurses, teachers, as well as any other person who is interested in this particular language.

Motive for this research and the writing of this paper, was the concern and interest in the absence of linguistic resources of Greek sign language (GST). In order to establish a basic electronic vocabulary, reachable to all groups of people, of a non-fully recorded language such as the sign language, it is highly required the collection of body-moving data, capable of supporting, on one hand the extraction of a wholesome of words and on the other hand, the analysis on the frequency that each "word" appears and make the final register in real-using conditions. To add more, one can easily observe a number of deficiencies mostly located in the organization of the entries, has only been based on chirality, but not in sign-language, which can also provide a way to sort all kinds of different meanings.

For its materialization we have chosen a web application, because of its attractive benefits such as:

- ✓ Provides easier access to users of all kinds.
- ✓ Provides independency from the user's operating system
- ✓ There are not any requirements concerning the installation and updates from the users
- ✓ It can support multiple users simultaneously.

For this web application we will be using already made technologies such as the open source CMS Drupal. It has been chosen because it is a full framework system for the development of web applications in general. It gives us the possibility of developing our modules, a fact which is quite crucial since it can make our application flexible, reducing the development time. The data-saving will

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

be in a relational database (DBMS). The final result will be a web application, operative and manageable by users unrelated to the IT industry.

Keywords: «Greek sign language, body-moving data, CMS, Drupal, MySQL, Database management tools, συναρτήσεις»

Περιεχόμενα

1. Εισαγωγή.....	- 19 -
1.1 Ελληνική νοηματική γλώσσα.....	- 19 -
1.2 Χρήση της Γλώσσας.....	- 21 -
1.3 Αντικείμενο Πτυχιακής.....	- 21 -
2. «Η έννοια της γλωσσολογίας»	- 23 -
2.1 Τι είναι Γλωσσολογία.....	- 23 -
2.2 Υπολογιστική γλωσσολογία	- 24 -
2.2.1 Προέλευση.....	- 25 -
3. Θεωρητικό υπόβαθρο διαδικτυακών εφαρμογών	- 27 -
3.1 Web hosting.....	- 27 -
3.1.1 Τι είναι Web hosting	- 27 -
3.1.2 Πακέτα Φιλοξενίας.....	- 28 -
3.2 Οι Βάσεις Δεδομένων και τα ΣΔΒΔ (DBMS)	- 28 -
3.2.1 Η Αρχιτεκτονική των ΣΔΒΔ	- 29 -
3.2.2 Το Σχεσιακό Μοντέλο Βάσεων Δεδομένων	- 30 -
3.2.3 Τα σχεσιακά ΣΔΒΔ (RDBMS).....	- 31 -
3.3 Web Server (Διακομιστής / Εξυπηρετητής).....	- 31 -
3.3.1 Λειτουργία ενός Web Server.....	- 32 -
3.3.2 Ιστορία.....	- 33 -
3.3.3 Μετάφραση του Path	- 34 -
3.3.4 Kernel-mode και user-mode web servers.....	- 34 -
3.3.5 Όρια Φόρτωσης του Server	- 35 -
4. Πρακτικό υπόβαθρο.....	- 37 -
4.1 Apache Web Server	- 37 -
4.1.1 Λειτουργίες και χαρακτηριστικά	- 37 -
4.1.2 Το ξεκίνημα και η εξέλιξη του Apache Web Server.....	- 37 -
4.1.3 Ο Apache σήμερα.....	- 38 -
4.1.4 Χαρακτηριστικά και λειτουργίες του Apache HTTP	- 38 -
4.2 Drupal	- 39 -
4.2.1 Τι είναι το Drupal.....	- 39 -

4.2.2	Σχεδίαση του Drupal	- 40 -
4.3	SQL Injection	- 41 -
4.3.1	Μορφές.....	- 41 -
4.3.2	Πρακτικές υλοποιήσεις	- 42 -
4.3.3	Πώς το Drupal χειρίζεται τα SQL Injections και κάνει ασφαλή τα sites.-	44 -
4.4	PhpMyAdmin	- 46 -
4.5	MySQL Workbench	- 47 -
4.5.1	Σχεδίαση	- 47 -
4.5.2	Ανάπτυξη	- 47 -
4.6	Γιατί Drupal	- 48 -
5.	Προγραμματιστικά εργαλεία, ρυθμίσεις και εφαρμογές τους.....	- 53 -
5.1	Εγκατάσταση των εργαλείων που χρησιμοποιήθηκαν	- 53 -
5.1.1	Διαμόρφωση ενός MySQL χρήστη και βάσης δεδομένων για το Drupal -	53 -
5.1.2	Εγκατάσταση PHP Modules και βελτίωση της διαμόρφωσης.....	- 54 -
5.1.3	Ενεργοποίηση της Λειτουργίας Rewrite και των Htaccess Files στον Apache -	55 -
5.1.4	Εγκατάσταση των αρχείων του Drupal	- 55 -
5.1.5	Εγκατάσταση του phpMyAdmin	- 58 -
5.1.6	Εγκατάσταση του MySQL Workbench.....	- 59 -
5.2	Σχετικές εικόνες κατά την υλοποίηση των προαναφερθέντων εργαλείων ...-	60 -
5.2.1	Η αρχική σελίδα του Drupal site.....	- 60 -
5.2.2	Βάση δεδομένων – PhpMyAdmin	- 61 -
5.2.3	Μοντελοποίηση βάσης στο MySQL Workbench	- 63 -
5.2.4	Εγκατάσταση των modules.....	- 63 -
6.	Τεχνικές Λεπτομέρειες - Εφαρμογές σε κώδικα / modules.....	- 65 -
6.1.	Αρχιτεκτονική, Σχεδίαση και case – scenarios της εφαρμογής	- 65 -
6.1.1	Σύστημα Επιπέδων Layered Architecture.....	- 65 -
6.1.2	Εφαρμογή αρχιτεκτονικής επιπέδων στο διαδικτυακό λεξικό.....	- 67 -
6.1.3	Case scenarios και διαδρομές επιλογών του User	- 71 -

6.2. Περιγραφή των αρχείων «.info» και «.module» ως μέσον δημιουργίας του λεξικού μας.....	- 75 -
6.2.1 Ανάλυση του «.info» αρχείου ενός module	- 76 -
6.1.2 Ανάλυση και περιγραφή του «.module» αρχείου	- 79 -
6.2.2 Αρχείο «.info».....	- 80 -
6.2.3 Βασικά αποσπάσματα κώδικα «.module».....	- 81 -
6.3 Τελική μορφή του Λεξικού μας.....	- 89 -
7. Σύνοψη και Συμπεράσματα	- 97 -
7.1 Αποτελέσματα	- 97 -
7.2 Συμπεράσματα	- 97 -
8. Πηγαίος κώδικας	- 99 -
8.1 Αρχείο «general_module.info».....	- 99 -
8.1.1 Αρχείο «general_module.module».....	- 99 -
8.2 Αρχείο «functions_module.info»	- 100 -
8.2.1 Αρχείο «functions_module.module»	- 100 -
Βιβλιογραφία	- 115 -
Βιβλία	- 115 -
Διαδικτυακοί Ιστότοποι.....	- 115 -
Παράρτημα	- 117 -
«Γλωσσάρι».....	- 117 -
Επεξήγηση τεχνικών όρων.....	- 117 -

Περιεχόμενα Εικόνων

Figure 1-1 – Αλφάβητο Νοηματικής Γλώσσας.....	20 -
Figure 1-2 – Σύστημα Διαχείρισης Περιεχομένου (CMS)	22 -
Figure 2-1 – Η Γλωσσολογία και τα επιμέρους τμήματά της	23 -
Figure 4-1 - Apache Web Server	39 -
Figure 4-2 - PhpMyAdmin	46 -
Figure 4-3 - Μοντελοποίηση Βάσης με MySQL Workbench.....	47 -
Figure 5-1 – Εγκατάσταση των απαραίτητων εργαλείων στο τερματικό	54 -
Figure 5-2 - Εγκατάσταση του Drupal (1 ^ο Βήμα).....	56 -
Figure 5-3 - Εγκατάσταση του Drupal (2 ^ο Βήμα).....	57 -
Figure 5-4 - Εγκατάσταση του Drupal (3 ^ο Βήμα).....	57 -
Figure 5-5 - Ολοκλήρωση εγκατάστασης του Drupal (4 ^ο Βήμα).....	58 -
Figure 5-6 - PhpMyAdmin	59 -
Figure 5-7 - Εγκατάσταση του MySQL Workbench.....	60 -
Figure 5-8 - Localhost	61 -
Figure 5-9 - Η βάση του Drupal στο PhpMyadmin	62 -
Figure 5-10 - Η μορφή της βάσης μας στο Workbench.....	63 -
Figure 5-11 - Ενεργοποίηση των modules	64 -
Figure 6-1 - Layered Architecture	66 -
Figure 6-2 – Κλειστό σύστημα αρχιτεκτονικής και κατεύθυνση του αιτήματος .-	67 -
Figure 6-3 - Αρχιτεκτονικό σύστημα επιπέδων στην εφαρμογή μας	68 -
Figure 6-4 - Γενική διαδρομή όλης της εφαρμογής.....	71 -
Figure 6-5 - Διαδρομή για εισαγωγή νέας λέξης.....	72 -
Figure 6-6 - Διαδρομή για εισαγωγή νέας κατηγορίας.....	73 -
Figure 6-7 - Διαδρομή για αναζήτηση ανά λέξη.....	73 -
Figure 6-8 - Διαδρομή για αναζήτηση ανά κατηγορία.....	74 -
Figure 6-9 - Διαδρομή αναζήτησης όλων των λέξεων που υπάρχουν στη βάση της εφαρμογής	74 -
Figure 6-10 - Διαδρομή αναζήτησης όλων των κατηγοριών που υπάρχουν στη βάση της εφαρμογής.....	75 -
Figure 6-11 - Λειτουργία των hooks	80 -
Figure 6-12 - Βασική μορφή του αρχείου «.info».....	81 -
Figure 6-13 - Βασική μορφή του hook_menu()	81 -
Figure 6-14 - Drupal_get_form()	82 -
Figure 6-15 - Εισαγωγή πεδίων fieldset και submit buttons	82 -
Figure 6-16 - Ανακατεύθυνση σελίδων στα modules	83 -
Figure 6-17 - Λειτουργία αναζήτησης.....	83 -
Figure 6-18 - Σύνδεση λέξης - video προς διαγραφή	84 -
Figure 6-19 - Τελική διαγραφή λέξης - video.....	85 -
Figure 6-20 - Φτιάχνοντας τη μορφή για το πεδίο εισαγωγής νέας λέξης στο λεξικό	86 -
Figure 6-21 - Video uploading για την αντίστοιχη λέξη.....	86 -

Figure 6-22 - Μηχανισμός PagerDefault() για τα αποτελέσματα της αναζήτησης ... - 87 -	
Figure 6-23 - Τα αποτελέσματα της αναζήτησης με σωστή δόμηση - 88 -	
Figure 6-24 - Home Page της διαδικτυακής εφαρμογής..... - 89 -	
Figure 6-25 - Εισαγωγή νέας λέξης..... - 90 -	
Figure 6-26 - Εισαγωγή νέας κατηγορίας..... - 90 -	
Figure 6-27 - Αναζήτηση όλων των λέξεων ή κατηγοριών αντίστοιχα..... - 91 -	
Figure 6-28 - Αποτελέσματα αναζήτησης όλων των κατηγοριών..... - 91 -	
Figure 6-29 - Αποτελέσματα αναζήτησης όλων των λέξεων..... - 92 -	
Figure 6-30 - Αναζήτηση ανάλογα με την επιθυμητή κατηγορία..... - 92 -	
Figure 6-31 - Εκτέλεση αναζήτησης ανά κατηγορία..... - 93 -	
Figure 6-32 - Αποτέλεσμα αναζήτησης κατηγορίας..... - 93 -	
Figure 6-33 - Αναζήτηση επιθυμητής λέξης..... - 94 -	
Figure 6-34 - Εκτέλεση αναζήτησης της λέξης μας..... - 94 -	
Figure 6-35 - Εμφάνιση της λέξης μας..... - 95 -	
Figure 6-36 - Διαγραφή λέξης από τη βάση..... - 95 -	
Figure 6-37 - Επιτυχής λειτουργία της διαγραφής..... - 95 -	
Figure 0-1 - Επίπεδα πρωτοκόλλων..... - 119 -	

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Συντομογραφίες

XSS Cross-Site Scripting

UI User Interface

ΣΒΔ Σχεσιακές Βάσεις Δεδομένων

SQLI SQL Injection

DB Database

CMS Content Management System (Σύστημα Διαχείρισης Περιεχομένου)

API Application Programming Interface

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

ΚΕΦΑΛΑΙΟ 1

1. Εισαγωγή

1.1 Ελληνική νοηματική γλώσσα

Η Ελληνική Νοηματική Γλώσσα (Ε.Ν.Γ.) αποτελεί την επίσημη, φυσική γλώσσα των κωφών στην Ελλάδα και είναι αναγνωρισμένη με νομοθετική ρύθμιση του ελληνικού κοινοβουλίου (ν.2817/2000). Είναι ένα διαφορετικό γλωσσικό σύστημα με άλλους δικούς του γλωσσικούς κανόνες και διαφορετική σύνταξη. Η γλώσσα αυτή λοιπόν, δεν ταυτίζεται και δεν απεικονίζει την ελληνική και σαφώς δεν προφανώς δεν στηρίζει τις ρίζες της σε αυτήν.

Όπως όλες οι νοηματικές γλώσσες, έτσι και η Ε.Ν.Γ. βασίζεται σε ένα οπτικο-κινησιακό σύστημα, ανάλογα με την κίνηση των χεριών, τη στάση και την κίνηση του σώματος, καθώς επίσης και τις εκφράσεις του προσώπου.

Τα νοήματα που αποδίδονται εν τέλει, βασίζονται κυρίως στις χειρομορφές, δηλαδή το σχήμα της παλάμης και η διάταξη των δαχτύλων σε συνδυασμό με:

- Τον προσανατολισμό της παλάμης
- Την κίνηση του χεριού
- Την έκφραση του προσώπου
- Την στάση και κίνηση του σώματος
- Και την θέση της χειρομορφής (πάνω στο σώμα η στον χώρο)

Η Ελληνική Νοηματική Γλώσσα δεν αποτελεί και τη μοναδική στον κόσμο. Κάθε χώρα έχει το δικό της τρόπο να εκφράζεται στη νοηματική, παρά το γεγονός ότι όλες βασίζονται στην ίδια αρχή (χειρομορφές, κίνηση, εκφράσεις), δεν είναι κατανοητές μεταξύ τους. Για παράδειγμα, ένας νοηματιστής της ελληνικής δε μπορεί να κατανοήσει έναν άλλον της αγγλικής ή ισπανικής κ.ο.κ. και αντίστροφα. Αυτό συμβαίνει διότι, όπως και οι ομιλούσες, έχουν δημιουργηθεί με σκοπό τη συνεννόηση μεταξύ των ατόμων μιας κοινότητας/ομάδας και να ικανοποιήσουν τις ανάγκες τους.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

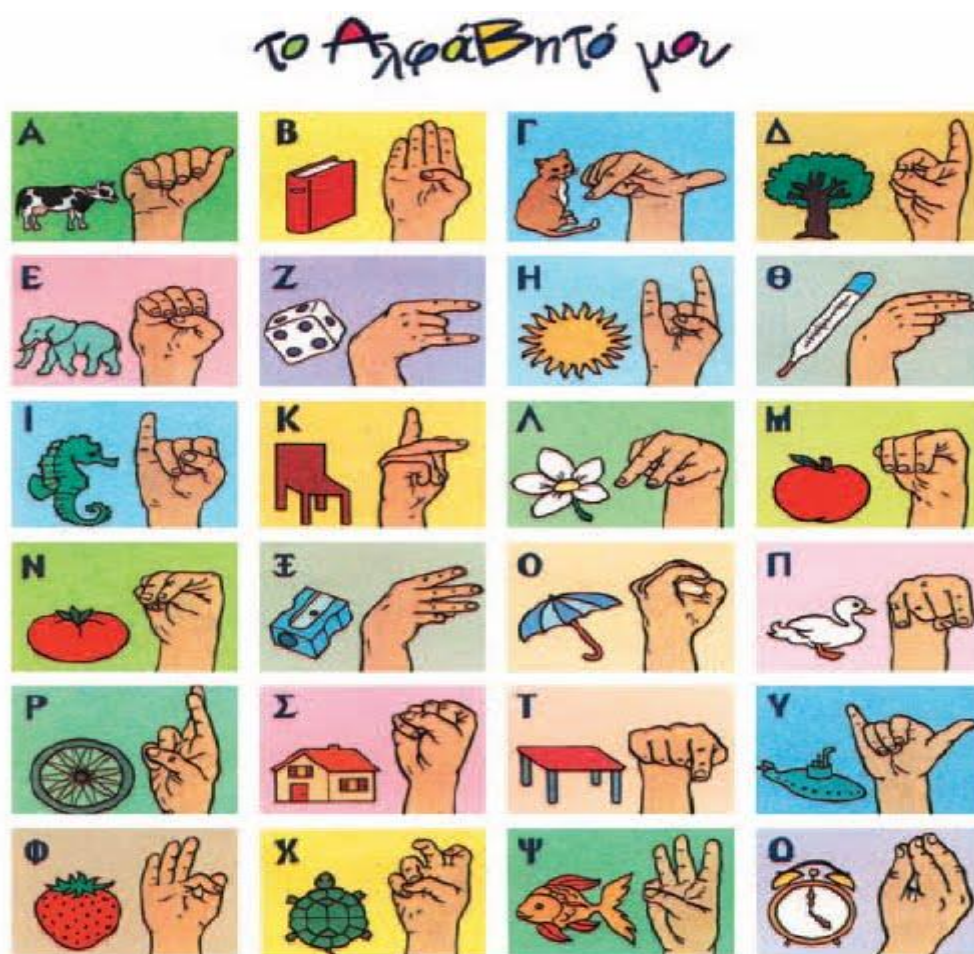


Figure 1-1 – Αλφάβητο Νοηματικής Γλώσσας

1.2 Χρήση της Γλώσσας

Όπως αναφέραμε και προηγουμένως, η Ε.Ν.Γ δεν είναι διεθνής όπως λανθασμένα πιστεύεται. Κάθε χώρα διαθέτει και αναπτύσσει τη δική της νοηματική με διαφορετικά νοήματα και διαφορετική συντακτική δομή. Υπάρχουν ορισμένα κοινά χαρακτηριστικά αλλά πολλές διαφορές σε μορφολογικό επίπεδο. Παρόλα αυτά, οι κωφοί διαφόρων κρατών μπορούν να συνεννοούνται άνετα μέσω της Διεθνούς Νοηματικής Γλώσσας, η οποία πρόκειται για έναν κώδικα για απλές καθημερινές ανάγκες επικοινωνίας.

Τα τελευταία χρόνια έχει παρατηρηθεί μεγάλη αύξηση στους χρήστες της νοηματικής γλώσσας, λόγω του μεγάλου ενδιαφέροντος από ακούοντες. Η συνεχής αύξηση της διάδοσής της πιστεύεται ότι οφείλεται στην επίσημη ανακήρυξή της, αλλά και στη προβολή ειδήσεων και εκπομπών από την Ελληνική τηλεόραση μέσω διερμηνείας, καθώς και συνεδριάσεων της Βουλής των Ελλήνων που προβάλλονται από το τηλεοπτικό σταθμό της Βουλή - Τηλεόραση.

Θα πρέπει να τονισθεί, πως παρά το γεγονός ότι η ελληνική νοηματική γλώσσα δεν απεικονίζει άμεσα την Ελληνική Γλώσσα, βασίζεται εν μέρει σε αυτή και μπορεί να αποδώσει, όπως κάθε ξεχωριστή γλώσσα, την οποιαδήποτε έκφραση της Ελληνικής ομιλούμενης γλώσσας.

1.3 Αντικείμενο Πτυχιακής

Αντικείμενο τη εργασίας αυτής αποτελεί η ανάπτυξη μιας web εφαρμογής με θέμα ένα λεξικό στη γλώσσα των Κωφών, μέσω ενός συστήματος διαχείρισης περιεχομένου (CMS). Το CMS είναι ακρωνύμιο για το Content Management System (Σύστημα Διαχείρισης Περιεχομένου). Στην πραγματικότητα, ένα CMS είναι μια διαδικτυακή εφαρμογή, που επιτρέπει την online τροποποίηση του περιεχομένου ενός δικτυακού τόπου και προσφέρει στον δημιουργό ενός website την δυνατότητα να οργανώνει, να χειρίζεται και να τροποποιεί το website. Η δημιουργία ενός ιστότοπου αποκτά εντελώς διαφορετικό νόημα αν χρησιμοποιηθεί ένα CMS.

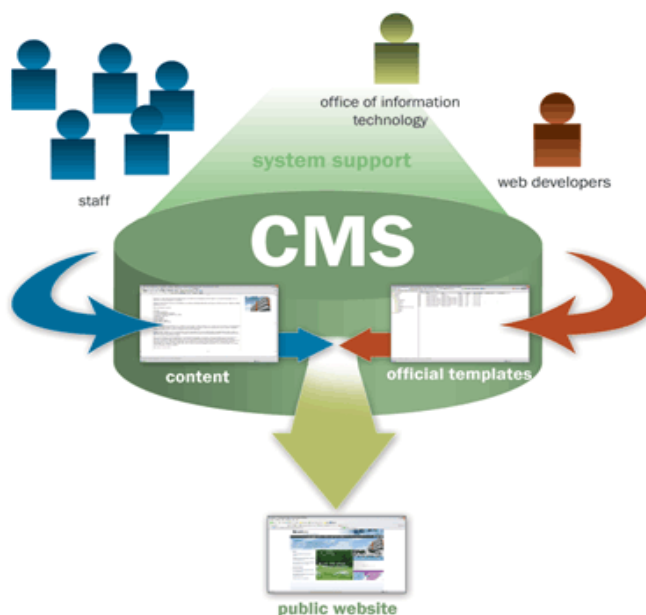


Figure 1-2 – Σύστημα Διαχείρισης Περιεχομένου (CMS)

Ένα Σύστημα Διαχείρισης Περιεχομένου (CMS), είναι το σύστημα που φροντίζει για όλο το περιεχόμενο σε μια ιστοσελίδα και το και το επεξεργάζεται με έναν απλό και εύκολο τρόπο χωρίς να απαιτούνται εξειδικευμένες γνώσεις στη διαχείρισή του. Με άλλα λόγια είναι ένα «αντικείμενο» υψηλής συμπερίληψης. Υλοποιείται με την λογική του μοντέλου «WYSIWYG» (What You See Is What You Get - Ό,τι βλέπεις είναι ό,τι παίρνεις), αποτελεί, δηλαδή, μια αντικειμενοστραφή λογική, όπου το «αντικείμενο» είναι στο υψηλότερο επίπεδο και όπου η υλοποίηση είναι μια εικονική λογική.

Η διαχείριση περιεχομένου ιστοσελίδων είναι ένα σύστημα, σχεδιασμένο με τέτοιο τρόπο ώστε να απλοποιεί τη διαχείριση του site και επιτρέπει στους administrators να δημοσιεύουν το δικτυακό περιεχόμενο χωρίς τεχνική γνώση html ή μεταφόρτωση των αρχείων.

Με ένα CMS, είναι πολύ εύκολη η λειτουργία δημοψηφισμάτων μέσω του website, το στήσιμο ενός forum, η δημιουργία ενός blog, η χρήση news feeds, η δημιουργία βάσης δεδομένων με εικόνες, αρχεία, κτλ. Αν λοιπόν, ο χρήστης θέλει να δημιουργήσει ένα δυναμικό, εύκολα αναβαθμίσιμο και «μοντέρνο» website, η εύκολη και σίγουρη λύση είναι ένα πρόγραμμα CMS.

ΚΕΦΑΛΑΙΟ 2

2. «Η έννοια της γλωσσολογίας»

2.1 Τι είναι Γλωσσολογία

Η γλωσσολογία αποτελεί μία επιστήμη καινούργια που βρίσκει την ανάπτυξή της τα τελευταία χρόνια. Σκοπό έχει α) στην ουσιαστική γνώση μιας γλώσσας, β) στο πώς κατακτάται αυτή η γνώση και γ) πώς χρησιμοποιείται. Η γλώσσα αποτελεί μέρος του ανθρώπινου χαρακτήρα και συμπεριφοράς του και κατ' αυτόν τον τρόπο πρέπει και να εξετάζεται. Πιο συγκεκριμένα, πρέπει να αποζητά τα χαρακτηριστικά που καθορίζουν τη μοναδικότητα και την καθολικότητά της, τα στοιχεία που την ξεχωρίζουν και τη διαφοροποιούν από τις άλλες. Σε μερικές γλώσσες η εξέλιξή τους θεωρείται δεδομένη, λόγω της ανάπτυξης των συγκεκριμένων κοινοτήτων και της συνεχούς αύξησης των αναγκών τους, κάποιες πάλι, δημιουργούνται εκ νέου και άλλες χάνουν την αξία τους και με την πάροδο των χρόνων ξεχνιούνται και «πεθαίνουν». Όπως όλες οι επιστήμες η γλωσσολογία διαιρείται σε θεωρητική και εφαρμοσμένη.

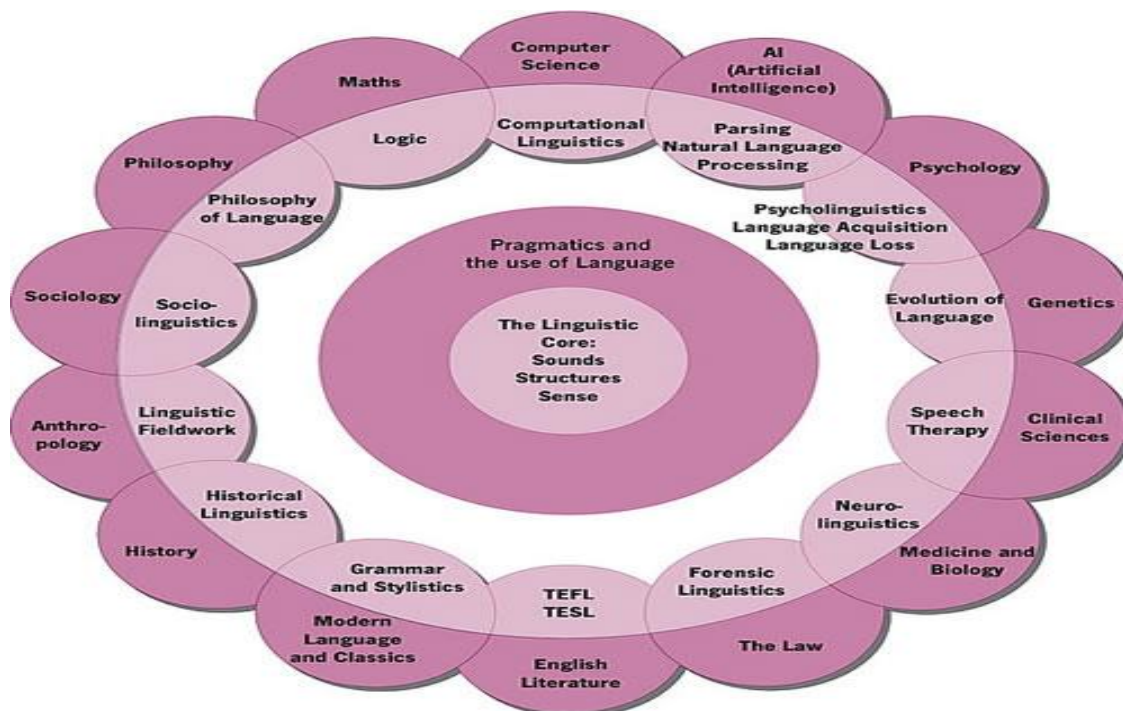


Figure 2-1 – Η Γλωσσολογία και τα επιμέρους τμήματά της

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Η θεωρητική Γλωσσολογία είναι η επιστήμη που μελετά τη γλώσσα βάσει των επιπέδων ανάλυσής της. Πυρήνας της θεωρητικής γλωσσολογίας θεωρούνται τα πεδία που περιλαμβάνουν τη σύνταξη, τη φωνολογία, τη μορφολογία και τη σημασιολογία της γλώσσας. Τέλος, η θεωρητική γλωσσολογία αποτελεί τον κλάδο που εξετάζει την καθολικότητα της γλώσσας, προσπαθεί δηλαδή να εξηγήσει τα κοινά χαρακτηριστικά της με τις υπόλοιπες γλώσσες.

Η εφαρμοσμένη Γλωσσολογία, με βάση τα πορίσματα της θεωρητικής Γλωσσολογίας και σε συνδυασμό με άλλες επιστήμες όπως η Ψυχολογία, τα Μαθηματικά, η Κοινωνιολογία κ.τ.λ., καλείται να εξετάσει τις συνθήκες υπό τις οποίες κατακτάται και χρησιμοποιείται η γλώσσα από τη γλωσσική κοινότητα, να προτείνει νέες μεθόδους και προσεγγίσεις για τη γλώσσα και να υποστηρίξει τη γλωσσική κοινότητα σε οποιοδήποτε ζήτημα ή πρόβλημα σχετικό με τη γλώσσα.

2.2 Υπολογιστική γλωσσολογία

Η υπολογιστική γλωσσολογία αποτελεί σημαντικό κομμάτι στον τομέα της γλωσσολογίας, καθώς την αναλύει από διαφορετική σκοπιά βασισμένη στη στατιστική, σε υπολογιστικούς κανόνες και στις θετικές επιστήμες. Πρόκειται για την μελέτη των υπολογιστικών συστημάτων, με σκοπό την κατανόηση και την παραγωγή φυσικών γλωσσών.

Η μελέτη για το συγκεκριμένο κομμάτι του κλάδου καλυπτόταν από επιστήμονες της Πληροφορικής και της θετικής κατεύθυνσης, με εξειδίκευση στους υπολογιστές και τα μαθηματικά. Οι υπολογιστικοί γλωσσολόγοι επεξεργάζονται τη φυσική γλώσσα και ερευνούν σε επιστημονικές ομάδες που διέπεται από γλωσσολόγους, ειδικούς εκπαιδευμένους στη γλωσσολογία, ειδικούς της γλώσσας και άτομα με υψηλό επίπεδο ικανοτήτων σε γλώσσες για κάποια δεδομένη εργασία και επιστήμονες της Πληροφορικής. Ουσιαστικά, πρόκειται για τη συνεργασία μεταξύ γλωσσολόγων, επιστημόνων υπολογιστών, ειδικών στην τεχνητή νοημοσύνη, μαθηματικών, επιστημόνων της λογικής, φιλοσόφων, γνωσιακών επιστημόνων, γνωσιακών ψυχολόγων, ανθρωπολόγων και νευροεπιστημόνων.

Η υπολογιστική γλωσσολογία έχει τόσο θεωρητικές, όσο και εφαρμοσμένες συνιστώσες. Πιο συγκεκριμένα, η θεωρητική υπολογιστική γλωσσολογία ασχολείται με θέματα της θεωρητικής γλωσσολογίας, ενώ η εφαρμοσμένη

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

υπολογιστική γλωσσολογία εστιάζει στα πρακτικά αποτελέσματα της μοντελοποίησης της χρήσης της φυσικής γλώσσας.

Οι επιστήμονες καταλήγουν στο συμπέρασμα, ότι ο μόνος τρόπος για την αντιμετώπιση της έκρηξης πληροφοριών και την συνεχή παρακολούθηση των δεδομένων και εξελίξεων είναι η βέλτιστη αξιοποίηση υπολογιστικών συστημάτων για το χειρισμό μεγάλης κλίμακας ηλεκτρονικής πληροφορίας. Η συστηματική και αυτοματοποιημένη αξιοποίηση επισημειωμένων κειμένων με υπολογιστικά μέσα, βοηθά στην κατεύθυνση για καλύτερη πρόβλεψη και αποφυγή «παγίδων», καθώς επίσης και στην ενδοσκόπηση στα κίνητρα και τις ιδεολογίες των συνομιλούντων και συγγραφέων.

2.2.1 Προέλευση

Η υπολογιστική γλωσσολογία ως κλάδος προϋπάρχει της τεχνητής νοημοσύνης, ένα πεδίο με το οποίο πλέον συνήθως συνεργάζεται στενά. Πρωτοεμφανίστηκε και γεννήθηκε στις Ηνωμένες Πολιτείες της Αμερικής την δεκαετία του 1950 όταν γίνονταν προσπάθειες να χρησιμοποιούνται υπολογιστές, για την αυτόματη μετάφραση ξενόγλωσσων κειμένων, και κυρίως επιστημονικών άρθρων και βιβλίων, στα αγγλικά. Οι γρήγοροι και ακριβείς αριθμητικοί υπολογισμοί των υπολογιστών προσπερνά πολλές τεχνικές λεπτομέρειες που εντοπίζονται στην αδυναμία του ανθρώπινου νου, όπως για παράδειγμα να λύνει πολλά και διαφορετικά προβλήματα τα ταυτόχρονα σε πολύ μικρό χρονικό διάστημα.

Όταν η αυτόματη μετάφραση – γνωστή και ως μηχανική μετάφραση – απέτυχε στο να παρέχει ακριβείς μεταφράσεις άμεσα, η αυτοματοποιημένη και απλή επεξεργασία των ανθρώπινων φυσικών γλωσσών απέκτησε έναν πιο περίπλοκο χαρακτήρα. Φάνηκε ότι τελικά η μελέτη της γλώσσας δεν είναι τόσο απλή, και έτσι η υπολογιστική γλωσσολογία γεννήθηκε ως το καινούργιο όνομα του επιστημονικού πεδίου, αφιερωμένο στην ανάπτυξη νέων αλγορίθμων και λογισμικών για την ευφυή επεξεργασία των γλωσσικών δεδομένων. Με την είσοδο της τεχνητής νοημοσύνης στο χώρο της τεχνολογίας, το 1960, η υπολογιστική γλωσσολογία έγινε υποκλάδος της και ασχολούταν με την κατανόηση και την παραγωγή των φυσικών γλωσσών σε ανθρώπινο επίπεδο.

Κατά τη μετάφραση μίας γλώσσας σε μία άλλη, παρατηρήθηκε πως είναι απαραίτητη η κατανόηση της Γραμματικής των επιμέρους γλωσσών, συμπεριλαμβανομένης και της μορφολογίας και της σύνταξης. Προκειμένου

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

κάποιος να αντιληφθεί τη σύνταξη μιας πρότασης, έπρεπε πρώτα να καταλάβει τη σημασιολογία και το λεξικό της γλώσσας. Όλη αυτή η προσπάθεια ξεκίνησε για τη μετάφραση δύο ξένων γλωσσών, αλλά με την έρευνα και τη μελέτη εξελίχθηκε σε κάτι ακόμη περισσότερο, σε μία ολόκληρη επιστήμη, η οποία αποσκοπεί στο να κατανοήσει πώς μπορούν να αντιπροσωπευτούν και να υποστούν επεξεργασία οι φυσικές γλώσσες με τη χρήση των υπολογιστών.

Βλέπουμε, λοιπόν, πως η κατανόηση της φυσικής γλωσσικής εξέλιξης, έχει βελτιωθεί σημαντικά με την πάροδο των χρόνων, χάρη στις εξελίξεις πάνω στον κλάδο της γλωσσολογίας. Η ανάπτυξη των συστημάτων και των υπολογιστών σε συνδυασμό με την ικανότητα του ανθρώπου να μοντελοποιεί και να τροποποιεί τα επιμέρους συστήματα ανάλογα με τις γλωσσικές συνιστώσες που προκύπτουν, δίνουν τη δυνατότητα να επαληθεύονται οι διάφορες επιστημονικές μελέτες.

Σήμερα, η έρευνα της υπολογιστικής γλωσσολογίας διεξάγεται σε εξειδικευμένα πανεπιστημιακά τμήματα υπολογιστικής γλωσσολογίας, σε πανεπιστημιακά τμήματα πληροφορικής και σε ευρύτερα πανεπιστημιακά τμήματα γλωσσολογίας.

ΚΕΦΑΛΑΙΟ 3

3. Θεωρητικό υπόβαθρο διαδικτυακών εφαρμογών

3.1 Web hosting

3.1.1 Τι είναι Web hosting

Η φιλοξενία ιστοσελίδων (Web Hosting ή απλά hosting) είναι μια διαδικτυακή υπηρεσία που επιτρέπει σε ιδιώτες και εταιρείες να διαθέτουν μία ιστοσελίδα συνεχώς αναρτημένη στο Διαδίκτυο, χωρίς να χρειάζεται να επιβαρύνεται με το κόστος του ανάλογου εξοπλισμού (π.χ. εξυπηρετητές) ή την ανάγκη εξυπηρέτησης μεγάλου αριθμού εξωτερικών συνδέσεων και εύρους σύνδεσης (bandwidth). Αυτό το αναλαμβάνουν οι εταιρίες φιλοξενίας ιστοσελίδων (web hosts) που προσφέρουν χώρο στον διακομιστή τους καθώς και μέρος της σύνδεσής τους προς το internet.

Ο όρος Web Hosting αναφέρεται στη διαδικασία με την οποία ο ιδιοκτήτης μίας ιστοσελίδας ενοικιάζει χώρο σε υπολογιστές (διακομιστές) για να τοποθετήσει τα αρχεία του. Τα αρχεία αυτά, που στοιχειοθετούν την ιστοσελίδα του, προσφέρονται μέσω ασφαλούς δικτύου συνεχόμενης παροχής στους επισκέπτες του.

Η διαχείριση του ιστότοπου από τον ιδιοκτήτη του γίνεται μέσω εμπορικού ή ανοικτού κώδικα γραφικού περιβάλλοντος (πίνακας ελέγχου hosting, control panel), το οποίο παρέχει δυνατότητα διαχείρισης του email, του ftp, των στατιστικών επισκέψεων του ιστότοπου, των εγκατεστημένων διαδικτυακών εφαρμογών και τεχνολογιών κ.α.

Ο ιδιοκτήτης του ιστότοπου μπορεί να ανεβάζει τα αρχεία του μέσω προγράμματος **FTP** στο διακομιστή φιλοξενίας και να τα τροποποιεί, να έχει ηλεκτρονική αλληλογραφία (ηλεκτρονικό ταχυδρομείο, e-mail) και να εγκαθιστά τις επιθυμητές διαδικτυακές εφαρμογές στον ιστότοπό του (ιστολόγιο/blog, forum, βιβλίο επισκεπτών κλπ). Μερικοί από αυτούς τους πίνακες ελέγχου φιλοξενίας είναι το Plesk, το Cpanel, το Webmin κ.α.

Από τις αρχές της δεκαετίας του '90 εταιρείες hosting ξεκίνησαν να δραστηριοποιούνται στον τομέα της παροχής φιλοξενίας καθώς η ανάγκη για σταθερό και συνεχές δίκτυο άρχισε να γίνεται επιβεβλημένη. Τη δεκαετία του 2000 η βιομηχανία του web hosting γνώρισε τεράστια άνθηση πρώτα στην Αμερική κι

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

έπειτα και στην Ευρώπη, ακολουθώντας την μεγάλη ζήτηση για υπηρεσίες hosting από ιδιώτες κι επιχειρήσεις για τη στέγαση της ιστοσελίδας τους.

3.1.2 Πακέτα Φιλοξενίας

Κάθε εταιρεία που παρέχει φιλοξενία σε ιδιοκτήτες ιστοσελίδων διαθέτει πακέτα hosting με ιδιαίτερα χαρακτηριστικά και τιμές. Ανάλογα με τη χρήση και την αναμενόμενη κίνηση της ιστοσελίδας ο ενδιαφερόμενος επιλέγει το πακέτο που του ταιριάζει καλύτερα. Η προσφορά χαρακτηριστικών εξελίσσεται παράλληλα με την τεχνολογία των υπολογιστών αλλά και τη διάδοση του διαδικτύου ως μέσου επικοινωνίας και συναλλαγών. Δεν είναι ασυνήθιστο τα χαρακτηριστικά των πακέτων να αλλάζουν 2 φορές τον χρόνο, προσφέροντας ακόμα περισσότερες δυνατότητες, ταχύτητα και υπηρεσίες. Τα πακέτα φιλοξενίας μπορούν χοντρικά να κατηγοριοποιηθούν στις παρακάτω οικογένειες:

1. Shared Hosting, όπου παρέχεται μέρος του (διακομιστή) στον οποίο φιλοξενούνται και άλλοι χρήστες.
2. Reseller Hosting, όπου παρέχεται η δυνατότητα να μεταπωληθεί χώρος και λοιπά εργαλεία φιλοξενίας ιστοσελίδων.
3. Virtual Private Server, όπου μέσω λογισμικού Εικονικοποίησης διακομιστή (Virtualization) παρέχεται ένας απομονωμένος χώρος στον διακομιστή, με δικούς του -αποκλειστικής χρήσης- πόρους συστήματος (μνήμη, επεξεργαστική ισχύ) και κεντρική πρόσβαση (web hosts)
4. Dedicated Servers, όπου παρέχεται ολόκληρος ο διακομιστής για αποκλειστική χρήση και διαχείριση από τον κάτοχο του ιστότοπου
5. Cloud Hosting, όπου με τεχνολογία διαμοιρασμού φόρτου εργασίας σε πολλούς διακομιστές ταυτόχρονα, τα εισερχόμενα αιτήματα εξυπηρέτησης διαμοιράζονται σε εκείνα τα μηχανήματα που έχουν το μικρότερο φόρτο εργασίας ενώ ταυτόχρονα τα αντίγραφα σε κάθε server εξασφαλίζουν την ακεραιότητα και τη διαθεσιμότητα των αρχείων.

3.2 Οι Βάσεις Δεδομένων και τα ΣΔΒΔ (DBMS)

Μια Βάση Δεδομένων (ΒΔ) είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, έτσι ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

χρήστες ταυτόχρονα. Υπάρχει ένα ειδικό λογισμικό το οποίο μεσολαβεί ανάμεσα στις αρχεία δεδομένων και τις εφαρμογές που χρησιμοποιούν οι χρήστες και αποκαλείται Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ) ή DBMS (Data Base Management System). Το ΣΔΒΔ είναι στην ουσία ένα σύνολο από προγράμματα και υπορουτίνες που έχουν να κάνουν με τον χειρισμό της βάσης δεδομένων, όσον αφορά τη δημιουργία, τροποποίηση, διαγραφή στοιχείων, με ελέγχους ασφαλείας κ.ά.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ' αυτήν. Το ΣΔΒΔ παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του ΣΔΒΔ μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα ΣΔΒΔ μπορεί να είναι εγκατεστημένο σ' έναν μόνο υπολογιστή ή και σ' ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν χρήστη ή και από πολλούς χρήστες.

3.2.1 Η Αρχιτεκτονική των ΣΔΒΔ

Ένα ΣΔΒΔ (Σύστημα Διαχείρισης Βάσης Δεδομένων) έχει σαν αποστολή τη διαχείριση των δεδομένων των αρχείων της βάσης, δηλ. την προσθήκη, διαγραφή, τροποποίηση εγγραφών, την αναζήτηση μέσα στις εγγραφές κ.ά.). Το ΣΔΒΔ δέχεται αιτήσεις από τους χρήστες των εφαρμογών και επικοινωνεί με τα αρχεία της βάσης δεδομένων για να τις διεκπεραιώσει.

Αυτή η κοινή διεπαφή (interface) των εφαρμογών με τα αρχεία αποκαλείται *λογική διεπαφή*. Οι εφαρμογές που δημιουργούμε δεν απασχολούνται με τον τρόπο που είναι αποθηκευμένα τα δεδομένα, πόσο χώρο καταλαμβάνουν κ.ο.κ και αυτή η ιδιότητα είναι γνωστή ως *ανεξαρτησία δεδομένων*.

Αυτό σημαίνει πρακτικά ότι οποιαδήποτε αλλαγή στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων δεν θα συνεπάγεται και αλλαγή στις εφαρμογές· ένα πρόβλημα που ταλαιπωρούσε πολύ τους προγραμματιστές παλαιότερων εποχών. Ακόμη, η προσθήκη, η κατάργηση ή και η τροποποίηση κάποιων εφαρμογών δεν θα έχει καμία επίπτωση στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων. Στα ΣΔΒΔ έχει επικρατήσει η λεγόμενη αρχιτεκτονική των τριών επιπέδων (βαθμίδων), όπου τα τρία επίπεδα είναι τα εξής :

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

- *Εσωτερικό επίπεδο (internal level)*: έχει να κάνει με την αποθήκευση των αρχείων στον σκληρό δίσκο, δηλ. την πραγματική ή φυσική κατάστασή τους.
- *Εξωτερικό επίπεδο (external level)*: έχει να κάνει με τους χρήστες είτε αυτοί είναι απλοί χειριστές, είτε προγραμματιστές ή και οι διαχειριστές της βάσης δεδομένων.
- *Εννοιολογικό επίπεδο (conceptual level)*: είναι ένα ενδιάμεσο επίπεδο που διασυνδέει τα δύο άλλα επίπεδα και έχει να κάνει με τη λογική σχεδίαση των αρχείων της βάσης δεδομένων.

3.2.2 Το Σχεσιακό Μοντέλο Βάσεων Δεδομένων

Το σχεσιακό (relational) μοντέλο έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα με αυτό αποκαλούνται σχεσιακές (relational databases). Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Υστερούν μόνο σε ταχύτητα υπολογισμών και σε χώρο αποθήκευσης, αλλά μόνο όταν έχουμε να κάνουμε με πολύ μεγάλες βάσεις δεδομένων.

Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων (relations) αντίστοιχα.

Ένας πίνακας (table) αποτελείται από γραμμές (rows) και στήλες (columns), όπου τοποθετούμε τα στοιχεία σε οριζόντια και κάθετη μορφή. Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται χαρακτηριστικό (attribute) ή πεδίο (field), ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες (στήλες) που αφορούν ένα στοιχείο της οντότητας και αποκαλείται πλειάδα (tuple) ή εγγραφή (record).

Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ, για παράδειγμα. Το σύνολο των αποδεκτών τιμών μιας οντότητας αποκαλείται πεδίο ορισμού (domain).

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

3.2.3 Τα σχεσιακά ΣΔΒΔ (RDBMS)

Τα Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ) ή RBMS (Relational Database Management Systems) αναπτύχθηκαν με βάση το σχεσιακό μοντέλο και έχουν επικρατήσει πλήρως στον χώρο. Κατά τον σχεδιασμό και τη δημιουργία μιας σχεσιακής βάσης δεδομένων, οι πίνακες αποτελούν το μοναδικό δομικό και απαραίτητο στοιχείο για να μπορέσουν να αναπαρασταθούν οι πληροφορίες που περιέχονται στη βάση δεδομένων.

Για να μπορέσουμε να προσθέσουμε, διαγράψουμε ή τροποποιήσουμε τα στοιχεία που περιέχονται σε μια βάση δεδομένων, χρησιμοποιούμε ειδικές γλώσσες προγραμματισμού που αποκαλούνται *γλώσσες ερωταπαντήσεων (query languages)*. Η γλώσσα που αποτελεί σήμερα ένα διεθνές πρότυπο για την επικοινωνία των χρηστών με τα Σχεσιακά ΣΔΒΔ είναι η *SQL (Structured Query Language)* ή *Δομημένη Γλώσσα Ερωτημάτων*. Μπορεί να λειτουργήσει αυτόνομα αλλά και σε συνεργασία με άλλες γλώσσες προγραμματισμού.

Μια άλλη, φιλική προς τον χρήστη γλώσσα προγραμματισμού για να μπορούμε να υποβάλουμε ερωτήματα σε σχεσιακές βάσεις δεδομένων και να λαμβάνουμε απαντήσεις είναι η *QBE (Query By Example)*, η οποία χρησιμοποιεί φόρμες για τη γραφική απεικόνιση των ερωτημάτων μας.

Σήμερα, υπάρχουν εξελιγμένα εργαλεία διαχείρισης σε γραφικό και φιλικό προς τον χρήστη περιβάλλον για να κάνουμε τα εξής :

- Δημιουργία πινάκων
- Δημιουργία φορμών
- Δημιουργία ερωτημάτων
- Δημιουργία εκθέσεων (αναφορών)

Τα Σχεσιακά ΣΔΒΔ τα διακρίνουμε στα *μεγάλα*, τα οποία αφορούν κυρίως μεγάλους οργανισμούς και επιχειρήσεις, έχουν τεράστιο όγκο δεδομένων και πολλούς χρήστες ταυτόχρονα, και τέτοια συστήματα είναι τα Oracle, Ingres, Informix, SQL Server κ.ά. και τα *μικρά*, τα οποία αφορούν κυρίως απλούς χρήστες, όπως είναι η Microsoft Access, η Paradox, η FoxPro κ.ά.

3.3 Web Server (Διακομιστής / Εξυπηρετητής)

Ένας διακομιστής web είναι μια τεχνολογία πληροφοριών (information technology) που επεξεργάζεται τις αιτήσεις μέσω *HTTP*, το βασικό πρωτόκολλο διαδικτύου που χρησιμοποιείται για την διανομή πληροφοριών στο World Wide

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Web. Ο όρος μπορεί να αναφέρεται είτε σε ολόκληρο το σύστημα υπολογιστών, σε μια συσκευή, ή ακόμα και στο λογισμικό που δέχεται και εμποπτεύει τις αιτήσεις HTTP.

3.3.1 Λειτουργία ενός Web Server

Η κύρια λειτουργία ενός web server είναι να αποθηκεύει, να επεξεργάζεται και να παρέχει ιστοσελίδες στους πελάτες. Η επικοινωνία μεταξύ του πελάτη και του διακομιστή πραγματοποιείται χρησιμοποιώντας το HTTP (Hypertext Transfer Protocol). Οι σελίδες που παρέχονται είναι συνήθως έγγραφα κειμένου HTML, τα οποία μπορεί να περιλαμβάνουν, εκτός του κειμένου, εικόνες και scripts.

Ένας web browser ή web crawler, ξεκινάει επικοινωνία στέλνοντας αίτημα για έναν συγκεκριμένο πόρο χρησιμοποιώντας http, και ο διακομιστής απαντάει με το περιεχόμενο αυτού του πόρου ή με ένα μήνυμα σφάλματος (error message) εάν δεν μπορέσει να ανταποκριθεί. Ο πόρος είναι βασικά ένα πραγματικό αρχείο σε δευτερεύον αποθηκευτικό χώρο του server, αλλά αυτό δεν είναι απαραίτητο και εξαρτάται από το πώς ο web server έχει εφαρμοστεί.

Ενώ η κύρια λειτουργία του είναι να παρέχει περιεχόμενο, μία πλήρης εφαρμογή του HTTP περιλαμβάνει επίσης και τρόπους λήψης περιεχομένου από τους πελάτες. Το χαρακτηριστικό αυτό χρησιμοποιείται για την υποβολή web φόρμας (web forms), συμπεριλαμβανομένου και uploading αρχείων.

Πολλοί web servers επίσης υποστηρίζουν server-side scripting χρησιμοποιώντας **Active Server Pages (ASP)**, PHP, ή άλλες γλώσσες για scripts. Αυτό σημαίνει ότι η συμπεριφορά του web server μπορεί να προγραμματιστεί σε διαφορετικά αρχεία, διαφορετικά scripts, ενώ το λογισμικό του server παραμένει αμετάβλητο. Συνήθως αυτή η λειτουργία χρησιμοποιείται για την δημιουργία html εγγράφων δυναμικά και αντίθετα επιστρέφει στατικά έγγραφα.

Στη μία περίπτωση, ο διακομιστής έχει την δυνατότητα να ανακτήσει ή να μεταποιήσει πληροφορίες στη βάση δεδομένων, ενώ στην δεύτερη είναι κυρίως πιο γρήγορη η διαδικασία, αλλά δεν μπορεί να παρέχει δυναμικό περιεχόμενο.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα



Εικόνα 3.1 - Επικοινωνία Server - PC

Οι Web διακομιστές δεν χρησιμοποιούνται πάντα για να εξυπηρετούν στο World Wide Web. Μπορούν επίσης να βρεθούν ενσωματωμένοι σε συσκευές όπως εκτυπωτές, δρομολογητές, κάμερες και εξυπηρετούν μόνο ένα τοπικό δίκτυο. Ο web server μπορεί να χρησιμοποιηθεί στη συνέχεια ως μέρος ενός συστήματος για την παρακολούθηση ή/και διαχείριση της εν λόγω συσκευής. Αυτό συνήθως σημαίνει ότι δεν χρειάζεται να εγκατασταθεί κανένα πρόσθετο λογισμικό στον υπολογιστή του client, αφού απαιτείται μόνο ένας web browser (ο οποίος περιλαμβάνεται στα περισσότερα λειτουργικά συστήματα).

3.3.2 Ιστορία

Το 1989 ο Tim Berners-Lee πρότεινε ένα νέο σχέδιο για τον εργοδότη του CERN, με στόχο τη διευκόλυνση της ανταλλαγής πληροφοριών μεταξύ των επιστημόνων χρησιμοποιώντας ένα σύστημα υπερκειμένου (hypertext system). Το σχέδιο κατέληξε με τον Berners-Lee να γράφει δύο προγράμματα το 1990:

- Ένα browser, ονομαζόμενο World Wide Web
- Και τον πρώτο στον κόσμο ASP web, αργότερα γνωστό ως CERN httpd, η οποία έτρεξε στο NeXTSTEP (object-oriented, multitasking operating system).

Μεταξύ του 1991 και του 1994, η απλότητα και η αποτελεσματικότητα των πρώιμων τεχνολογιών που χρησιμοποιήθηκαν για το σερφάρισμα και την ανταλλαγή πληροφοριών μέσω του World Wide Web, συνέβαλε στη μεταφορά τους σε πολλά διαφορετικά λειτουργικά συστήματα και στην εξάπλωσή τους σε πολλές επιστημονικούς οργανισμούς και πανεπιστήμια και έπειτα και στην βιομηχανία.

Το 1994 ο Tim Berners-Lee αποφάσισε να αποτελέσει το World Wide Web Consortium (W3C), για να ρυθμίσει την περαιτέρω ανάπτυξη των πολλών τεχνολογιών περιεχομένου html, http κτλ, μέσω μιας τυποποιημένης διαδικασίας.

3.3.3 Μετάφραση του Path

Οι web servers είναι σε θέση να χαρτογραφήσουν το path του Uniform Resource Locator (URL) σε:

- Ένα τοπικό σύστημα αρχείων πόρων (για στατικά αιτήματα)
- Ένα εσωτερικό ή εξωτερικό όνομα προγράμματος (για δυναμικά αιτήματα)

Για στατικό αίτημα η URL διαδρομή που ορίζεται από τον πελάτη είναι σχετική με το root directory του web server.

Σκέψου το ακόλουθο URL όπως θα είχε ζητηθεί από έναν client:

<http://www.example.com/path/file.html>

Ο **user agent** του client θα το μεταφράσει σε μια σύνδεση του www.example.com με το ακόλουθο HTTP 1.1 αίτημα:

GET /path/file.html HTTP/1.1

Host: www.example.com

Ο web διακομιστής του www.example.com θα προσαρτήσει το δεδομένο μονοπάτι στο path του root directory. Σε έναν Apache server είναι συνήθως /home/www (Σε μηχανήματα unix συνήθως είναι /var/www). Το αποτέλεσμα είναι το τοπικό σύστημα αρχείων πόρων:

</home/www/path/file.html>

Ο web διακομιστής μετά διαβάζει το αρχείο, εάν υπάρχει και στέλνει απάντηση στον web browser του client. Η απάντηση θα περιγράφει το περιεχόμενο του αρχείου και θα περιέχει το ίδιο το αρχείο ή θα εμφανίζει ένα μήνυμα λάθους εάν το αρχείο δεν υπάρχει ή δεν είναι διαθέσιμο.

3.3.4 Kernel-mode και user-mode web servers

Ο web server (πρόγραμμα) μπορεί να είναι είτε ενσωματωμένος στον πυρήνα Kernel του λειτουργικού συστήματος, είτε σε έναν χώρο χρήστη (όπως οι άλλες συνηθισμένες εφαρμογές).

Ένας web server στον Kernel (όπως ο Microsoft IIS στα Windows ή ο TUX σε GNU/Linux) συνήθως είναι πιο γρήγορος, διότι ως μέρος του συστήματος, μπορεί απευθείας να χρησιμοποιήσει όλους τους πόρους hardware που χρειάζεται.

Οι web servers που τρέχουν σε user-mode πρέπει να ζητήσουν την άδεια του συστήματος, για να χρησιμοποιήσουν περισσότερη μνήμη ή περισσότερους

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

πόρους στη CPU, Αυτά τα αιτήματα όχι μόνο απαιτούν χρόνο, αλλά δεν ικανοποιούνται και πλήρως, επειδή το σύστημα κρατάει αποθέματα για δική του χρήση και έχει την ευθύνη να μοιράσει πόρους του hardware και στις υπόλοιπες εφαρμογές που τρέχουν.

3.3.5 Όρια Φόρτωσης του Server

Ένας διακομιστής web έχει όριο φόρτωσης, επειδή μπορεί να χειριστεί μόνο ένα περιορισμένο αριθμό ταυτόχρονων συνδέσεων πελατών-clients (συνήθως μεταξύ 2 και 80.000, από default μεταξύ 500 και 1000) ανά διεύθυνση IP (και **TCP** θύρα) και μπορεί να εξυπηρετήσει ένα συγκεκριμένο μέγιστο αριθμό αιτημάτων ανά δευτερόλεπτο και εξαρτάται από:

- ✓ τις ρυθμίσεις του,
- ✓ τον τύπο αιτήματος HTTP,
- ✓ εάν το περιεχόμενο είναι στατικό ή δυναμικό,
- ✓ εάν το περιεχόμενο έχει αποθηκευτεί προσωρινά (cache), και
- ✓ τα όρια hardware και software του λειτουργικού συστήματος του υπολογιστή στον οποίο τρέχει ο web server.

Όταν ο διακομιστής web είναι κοντά ή έχει ξεπεράσει τα όριά του, τότε δεν θα ανταποκριθεί.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

ΚΕΦΑΛΑΙΟ 4

4. Πρακτικό υπόβαθρο

4.1 Apache Web Server

4.1.1 Λειτουργίες και χαρακτηριστικά

Ο Apache Web Server είναι αυτό ακριβώς που δηλώνει το όνομά του. Πρόκειται δηλαδή 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). Στο κεφάλαιο αυτό θα ασχοληθούμε μόνο με το software 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 «σερβίροντας» 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 σύμφωνα με τα πρότυπα που ορίζει το πρωτόκολλο HTTP (Hypertext Transfer Protocol).

4.1.2 Το ξεκίνημα και η εξέλιξη του Apache Web Server

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 εγκαταλελειμμένο, πέρα από κάποιες διορθώσεις (patches) που ανέπτυξαν 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.

4.1.3 Ο Apache σήμερα

Ο Apache HTTP αναπτύσσεται από την «Κοινότητα Ανοιχτού Λογισμικού» και η εποπτεία, υποστήριξη, και διάθεση του προγράμματος γίνεται από τον 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.

4.1.4 Χαρακτηριστικά και λειτουργίες του 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 τόσα 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 κ.λπ. 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 απ'ο το Apache Software Foundation, είτε από τρίτες εταιρίες λογισμικού.

Ένα άλλο χαρακτηριστικό – δυνατότητα του Apache HTTP, όπως έχω αναφέρει πιο πάνω, είναι ότι μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα. Ο Apache HTTP υποστηρίζει επίσης αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως MySQL, PHP, Perl, Python κ.λπ.

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 κ.α.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

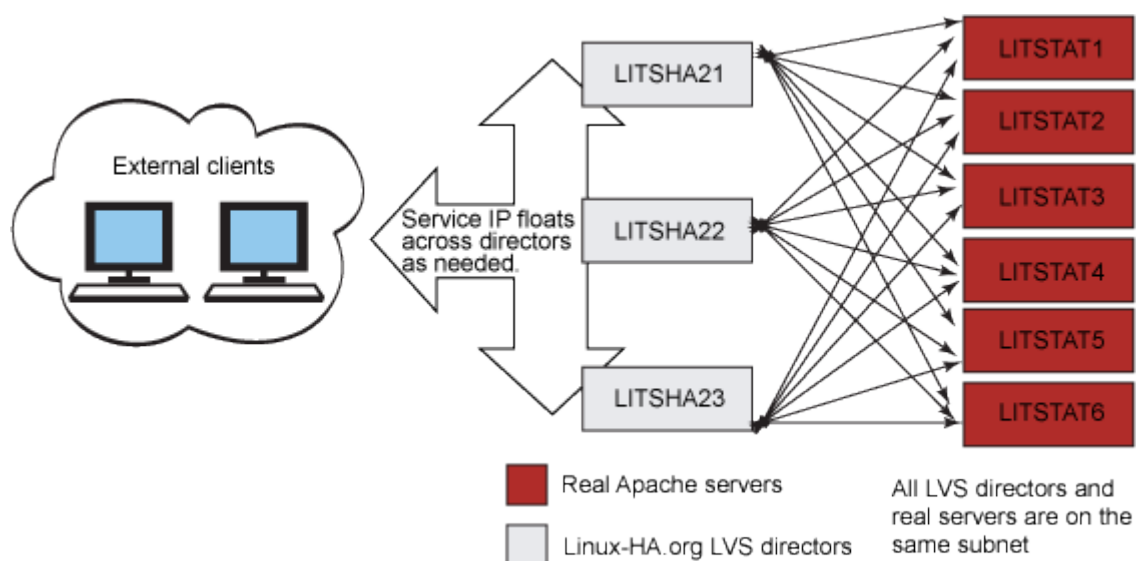


Figure 4-1 - Apache Web Server

4.2 Drupal

Αρχικά γραμμένο από τον Dries Buytaert ως σύστημα πίνακα ανακοινώσεων, το Drupal μετατράπηκε σε εγχείρημα ανοικτού κώδικα το 2001. *Drupal* είναι η διατύπωση στην Αγγλική γλώσσα της Ολλανδικής λέξης "druppel", που σημαίνει «σταγόνα». Το όνομα έχει προέλευση από τον ξεπερασμένο πλέον ιστοτόπο Drog.org, του οποίου ο κώδικας εξελίχθηκε στο Drupal. Ο Buytaert ήθελε να ονομάσει τον ιστότοπο "drog" (στα Ολλανδικά σημαίνει «χωριό», αναφερόμενος στη διάσταση της κοινότητας), αλλά έκανε ένα ορθογραφικό λάθος κατά τη διαδικασία ελέγχου του ονόματος χώρου (domain name) και τελικά σκέφτηκε ότι ακούγεται καλύτερα.

Από το Μάιο του 2006 ως τον Απρίλιο του 2007, χρήστες κατέβασαν το Drupal από τον επίσημο ιστότοπο περισσότερες από 600.000 φορές. Μια μεγάλη κοινότητα χρηστών λαμβάνει πλέον μέρος στη συνεχή εξέλιξη του Drupal.

4.2.1 Τι είναι το Drupal

Το **Drupal** είναι ένα αρθρωτό σύστημα διαχείρισης περιεχομένου (Content Management System, CMS) ανοικτού/ελεύθερου λογισμικού, γραμμένο στη γλώσσα προγραμματισμού PHP. Το Drupal, όπως πολλά σύγχρονα CMS,

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

επιτρέπει στο διαχειριστή συστήματος να οργανώνει το περιεχόμενο, να προσαρμόζει την παρουσίαση, να αυτοματοποιεί διαχειριστικές εργασίες και να διαχειρίζεται τους επισκέπτες του ιστότοπου και αυτούς που συνεισφέρουν. Παρόλο που υπάρχει μια πολύπλοκη προγραμματιστική διεπαφή, οι περισσότερες εργασίες μπορούν να γίνουν με λίγο ή και καθόλου προγραμματισμό. Το Drupal, ορισμένες φορές περιγράφεται ως «υποδομή για εφαρμογές ιστού», καθώς οι δυνατότητές του προχωρούν παραπέρα από τη διαχείριση περιεχομένου, επιτρέποντας ένα μεγάλο εύρος υπηρεσιών και συναλλαγών.

4.2.2 Σχεδίαση του Drupal

Το Drupal έχει λάβει επαίνους από τους διαχειριστές ιστοσελίδων, σχεδιαστές και προγραμματιστές για τον αρθρωτό σχεδιασμό του, που παρέχει το βασικό του στρώμα, ή «πυρήνα», να παρέχει τα βασικά χαρακτηριστικά του Drupal στην προεπιλεγμένη εγκατάστασή του. Πρόσθετα χαρακτηριστικά λειτουργικότητας και παρουσίασης μπορούν να επεκταθούν στον πυρήνα με την πρόσθεση προσαρτώμενων μονάδων και θεματικών παραλλαγών.

Οι μονάδες του Drupal χρησιμοποιούνται για να «υπερβούν» τα ενσωματωμένα χαρακτηριστικά του πυρήνα, επεκτείνοντας έτσι ή και αντικαθιστώντας την εξ' ορισμού συμπεριφορά του Drupal, χωρίς την επέμβαση στον αυτούσιο κώδικα των αρχείων του πυρήνα του Drupal. Η δυνατότητα αυτή της τροποποίησης της λειτουργικότητας του πυρήνα έχει επίπτωση στην προσαρμοστικότητα του Drupal καθώς και στην ασφάλειά του, ειδικότερα σε θέματα ασφαλείας, όπως η έγχυση εντολών SQL (**SQL injection**).

Προσαρμοσμένες θεματικές παραλλαγές, που μπορούν να προστεθούν χωρίς να επηρεάζουν τον πυρήνα του Drupal, χρησιμοποιούν προτυποποιημένες μορφές που μπορούν να δημιουργηθούν από μηχανές σχεδίασης θεματικών παραλλαγών τρίτων.

4.3 SQL Injection

Πολλοί κακόβουλοι χρήστες επιδιώκουν να εκμεταλλευτούν «αδύναμα σημεία» και ευπάθειες στην ασφάλεια μια διαδικτυακής εφαρμογής, ώστε να αποκτήσουν πλήρη πρόσβαση στα δεδομένα της βάσης. Η μέθοδος αυτή ονομάζεται SQL Injection. Δεν είναι καινούργια, ούτε και πολύπλοκη ως μέθοδος. Για την αποτελεσματική εφαρμογή αυτής της μεθόδου, αρκεί ένας web browser και η κοινή λογική. Η μέθοδος, λοιπόν, των SQLI κάνει την είσοδό της μαζί με τις πρώτες web εφαρμογές.

Όπως αναφέραμε, οι επιθέσεις με αυτή τη μέθοδο, έχουν στόχο τα «αδύναμα σημεία» της ασφάλειας στην εφαρμογή, με σκοπό την εκτέλεση επερωτήσεων SQL (SQL queries) που δεν έχουν προβλεφθεί από τους προγραμματιστές.

Παραδείγματα των SQL queries όπως θα δούμε και σε επόμενο υποκεφάλαιο είναι η προβολή δεδομένων (SELECT) που δεν έχει προβλεφθεί να προβάλλονται στους χρήστες, η ανεξέλεγκτη εισαγωγή δεδομένων (INSERT), καθώς και η αντίστοιχη διαγραφή τους (DELETE).

Ακόμα και σήμερα, είναι πολλά τα συστήματα που είναι ευάλωτα στις επιθέσεις των SQLI. Όμως, το θετικό είναι ότι μπορεί πλέον, πολύ εύκολα να διερευνηθούν σε μια web εφαρμογή και να ληφθούν τα κατάλληλα μέτρα για την αντιμετώπισή τους.

4.3.1 Μορφές

Υπάρχουν 5 κυρίως υπο-κλάσεις της SQL injection:

- Classic SQLI
- Blind ή Inference SQL injection
- Database management system-specific SQLI
- Compounded SQLI
 - SQL injection + insufficient authentication
 - SQL injection + DDoS attacks^[8]
 - SQL injection + DNS hijacking^[9]
 - SQL injection + XSS^[10]

4.3.2 Πρακτικές υλοποιήσεις

Ένας κακόβουλος χρήστης εκμεταλλεύεται τη λανθασμένη επαλήθευση των δεδομένων εισόδου (input) μιας εφαρμογής με τα SQL injections. Η ευπάθεια των SQLI δεν εμφανίζεται μόνο σε διαδικτυακές εφαρμογές, αλλά το φαινόμενο αυτό είναι πιο σπάνιο.

Συνήθως οι χρήστες δίνουν δεδομένα εισόδου (input data) σε μια διαδικτυακή εφαρμογή μέσω των φορμών με τις μεθόδους HTTP GET και POST, καθώς επίσης και μέσω συνδέσμων (links) με τη μέθοδο HTTP GET. Όταν δε γίνεται επαρκής επαλήθευση στα δεδομένα εισόδου, τότε υπάρχει περίπτωση να περαστούν νέες εντολές SQL στην web εφαρμογή ή ακόμα και να τροποποιηθούν μέρη ενός SQL query, με απρόσμενο τρόπο.

Παρακάτω μπορούμε να δούμε το πρόβλημα μέσω υλοποιήσιμων απλών παραδειγμάτων σε PHP:

Παράδειγμα 1:

Ας θεωρήσουμε την περίπτωση προβολής στοιχείων ενός υπαλλήλου από έναν πίνακα με όνομα «employees» σε μια βάση δεδομένων MySQL.

Το παρακάτω (ευπαθές) block κώδικα PHP εκτελεί μια επερώτηση SELECT στη βάση δεδομένων, με σκοπό την ανάγνωση από τη βάση δεδομένων των στοιχείων ενός συγκεκριμένου υπαλλήλου.

```
<?php
$query = "SELECT employeeid, fullname, salary FROM employees " .
        "WHERE employeeid = " . $_GET['employeeid'];
$result = mysql_query($query);
?>
```

Σκοπός του προγραμματιστή εδώ είναι η εκτέλεση SQL queries της μορφής:

```
SELECT employeeid, fullname, salary FROM employees WHERE employeeid = 3
SELECT employeeid, fullname, salary FROM employees WHERE employeeid = 352
SELECT employeeid, fullname, salary FROM employees WHERE employeeid = 590
```

όπου το employeeid (πρωτεύον κλειδί στον πίνακα employees) είναι τιμή που δίνεται στην πράξη από τον χρήστη της εφαρμογής (μέσω του browser, με χρήση της μεθόδου GET του HTTP). Για παράδειγμα, στην τυπική περίπτωση, το employeeid μπορεί να δίνεται με ένα link της μορφής:

```
http://www.example.com/employees.php?employeeid=3
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Το πρόβλημα είναι ότι η τιμή της παραμέτρου GET "employeeid" που δίνεται στο URL, δεν επαληθεύεται επαρκώς πριν την εκτέλεση της επερώτησης από τον κώδικα.

Έτσι ένας κακόβουλος χρήστης μπορεί να γράψει το εξής URL (χειρονακτικά) στον browser:

<http://www.example.com/employees.php?employeeid=3 OR 1=1>

όπου θα έχει ως αποτέλεσμα να εκτελεστεί στη βάση δεδομένων η εξής επερώτηση:

```
SELECT employeeid, fullname, salary FROM employees WHERE employeeid=3 OR 1=1
```

έτσι όμως ο κλάδος WHERE θα ισχύει για κάθε εγγραφή του πίνακα employees, οπότε επιστρέφονται όλες οι εγγραφές στη μεταβλητή \$result.

Ανάλογα με το πως χρησιμοποιείται λοιπόν στη συνέχεια η μεταβλητή \$result, για να παρουσιάσει τα δεδομένα στους χρήστες της web εφαρμογής, ενδέχεται ο κακόβουλος χρήστης να δει πληροφορίες που δεν είναι σκόπιμο.

Παράδειγμα 2:

Το παράδειγμα αυτό, εκμεταλλεύεται τη δυνατότητα εκτέλεσης πολλαπλών εντολών SQL ως μια επερώτηση (SQL query) στο σύστημα. Οι εντολές διαχωρίζονται μεταξύ τους με τον χαρακτήρα ";" (semicolon). Η εκτέλεση πολλαπλών εντολών ως μια επερώτηση είναι μια συνηθισμένη δυνατότητα στο χώρο των βάσεων δεδομένων, όμως αποτελεί και τον μεγαλύτερο κίνδυνο στην περίπτωση των SQLI.

Θα βασιστούμε στο παράδειγμα 1 και θα θεωρήσουμε ότι το σύστημα βάσεων δεδομένων είναι η PostgreSQL. Έτσι, έχουμε το εξής (ευπαθές) block κώδικα:

```
<?php
$qry = "SELECT employeeid, fullname, salary FROM employees " .
      "WHERE employeeid =" . $_GET['employeeid'];
$result = pg_query($qry);
?>
```

ο προγραμματιστής αναμένει δεδομένα εισόδου από links της μορφής:

<http://www.example.com/employees.php?employeeid=3>

που θα έχει ως αποτέλεσμα την εκτέλεση της επερώτησης:

```
SELECT employeeid, fullname, salary FROM employees
WHERE employeeid = 3
```

αυτή τη φορά όμως ο κακόβουλος χρήστης δίνει την εξής URL στον browser:

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

<http://www.example.com/employees.php?employeeid=3;DELETE FROM users;>

οπότε θα εκτελεστούν οι εξής 2 εντολές στην κλήση της `pg_query()`:

```
SELECT employeeid, fullname, salary FROM employees
```

```
WHERE employeeid = 3;
```

```
DELETE FROM users;
```

όπου θα έχει ως αποτέλεσμα να διαγραφούν όλα τα δεδομένα του πίνακα `users` από τη βάση.

Φυσικά αντί του `DELETE` του παραδείγματος, οποιαδήποτε άλλη SQL εντολή θα μπορούσε στην περίπτωση αυτή να εκτελεστεί. Αυτό είναι ένα καλό παράδειγμα, του γιατί δεν είναι ασφαλές να συνδέεται η εφαρμογή μας στη βάση δεδομένων με τα στοιχεία ενός χρήστη της βάσης που έχει αυξημένα δικαιώματα πρόσβασης. Με πλήρη δικαιώματα πρόσβασης, κάποιος θα μπορούσε να διαγράψει έως και άλλες βάσεις δεδομένων του συστήματος μη σχετιζόμενες με τη συγκεκριμένη εφαρμογή web).

4.3.3 Πώς το Drupal χειρίζεται τα SQL Injections και κάνει ασφαλή τα sites

I. Χρήση των `check functions` σε `output` για την αποφυγή επιθέσεων `cross-site scripting (XSS)`.

Δεν πρέπει ποτέ να τοποθετείται στην HTML περιεχόμενο που έχει υποβληθεί από χρήστες.

- Χρήση του `check_plain` για απλό text
- Χρήση του `filter_xss`, στην περίπτωση του το text περιέχει markups, όπου γίνεται φιλτράρισμα string στην HTML και χρήση του `filter_xss_admin` εάν η εισαγωγή του περιεχομένου γίνεται από admin, για να επιτρέπονται περισσότερα markups. Σχεδόν ποτέ δεν είναι σωστό να χρησιμοποιούμε `check_markup` σε theme ή module παρά μόνο σε περιεχόμενο σε text area με σχετικό text format.
- Χρήση της `t()` function ή τα προκαθορισμένα % για την ασφαλών strings.

II. Χρήση του `database abstraction layer` για την αποφυγή επιθέσεων τύπου `SQL injection`.

Χρησιμοποίηση του database layer σωστά. Για παράδειγμα, ποτέ μην ενσωματώνουμε δεδομένα σε SQL query, με αυτόν τον τρόπο:

```
<?php
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
db_query('SELECT foo FROM {table} t WHERE t.name = ' . $_GET['user']);
```

?>

Αντί αυτού, μπορούμε να χρησιμοποιούμε κατάλληλη αντικατάσταση των ορισμάτων με `db_query`:

Το επίπεδο της βάσεως δεδομένων (Database Layer) λειτουργεί στην κορυφή του PHP PDO και χρησιμοποιεί ένα array από προκαθορισμένες θέσεις ονομάτων:

```
<?php
```

```
db_query("SELECT foo FROM {table} t WHERE t.name = :name", array(':name' => $_GET['user']));
```

?>

Για ένα όρισμα με μεταβλητό αριθμό, χρησιμοποιείται array με ορίσματα ή **`db_select()`**:

```
<?php
```

```
db_query("SELECT t.s FROM {table} t WHERE t.field IN (:users)", array(':users' => $from_user));
```

?>

Ή

```
<?php
```

```
$result = db_select('table', 't')
```

```
->fields('t', array('s'))
```

```
->condition('t.field', $from_user, 'IN')
```

```
->execute();
```

?>

III. Χρήση `db_rewrite_sql` για node access restrictions.

Τα περισσότερα SQL Statements που αναφέρονται στα nodes ή στο {node} table θα έπρεπε να περικλείεται σε μία συνάρτηση

db_rewrite_sql_function_call:

```
<?php
```

```
$result = db_query(db_rewrite_sql("SELECT n.nid, n.title FROM {node} n"));
```

?>

Ο μηχανισμός πρόσβασης σε nodes του Drupal απαιτεί τέτοιες συναρτήσεις. Χωρίς αυτές, οι επισκέπτες μπορούν να αποκτήσουν πρόσβαση σε nodes στα οποία δεν έχουν δικαιώματα.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

4.4 PhpMyAdmin

Πρόκειται για ένα δωρεάν εργαλείο software για την διαχείριση της MySQL στο διαδίκτυο. Το phpmyadmin υποστηρίζει μεγάλο εύρος λειτουργιών σε MySQL, MariaDB και Drizzle. Συχνά, χρησιμοποιούμενες λειτουργίες διαχείρισης βάσεων δεδομένων, πινάκων, στηλών, σχέσεων, ευρετηρίων, χρηστών, δικαιωμάτων, κ.λπ. μπορούν να εκτελεστούν σε user interface, ενώ εξακολουθεί να υπάρχει η δυνατότητα άμεσης εκτέλεσης οποιαδήποτε SQL statement.

Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> id	tinyint(4)		No		auto_increment	
<input type="checkbox"/> person_name	varchar(200)		No			
<input type="checkbox"/> town_code	varchar(5)		Yes	0		
<input type="checkbox"/> country_code	char(1)		Yes	NULL		
<input type="checkbox"/> car_code	char(3)		No			

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	2		id
town_code	INDEX	2		town_code
country_code	INDEX	2		country_code
pays-ville	INDEX	2		country_code town_code

Space usage:
Type Usage
Data 16,384 Bytes
Index 49,152 Bytes
Total 65,536 Bytes

Row Statistic:
Statements Value
Format dynamic
Next Autoindex 5

More than one INDEX key was created for column 'country_code'
Create an index on 1 columns Go

Figure 4-2 - PhpMyAdmin

Στα χαρακτηριστικά του εργαλείου συμπεριλαμβάνονται:

- Προσπέλαση των βάσεων δεδομένων και μετατροπή των data που περιλαμβάνουν.
- Διαχείριση αποθηκευμένων διαδικασιών και εναυσμάτων.
- Εισαγωγή δεδομένων μέσω CSV και SQL.
- Εξαγωγή σε μορφές, όπως CSV, SQL, XML, PDF, OpenDocument Text και Spreadsheet, Word, Excel, LATEX.
- Υποστήριξη αιτημάτων μέσω παραδείγματος (QBE).
- Διαχείριση πολλαπλών servers

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

4.5 MySQL Workbench

Το Workbench είναι ένα εργαλείο αρχιτεκτονικής βάσεων δεδομένων, για τους προγραμματιστές και DBAs. Με το MySQL Workbench φτιάχνουμε το μοντέλο της DB (βάσης δεδομένων) της εφαρμογής, αναπτύσσουμε την SQL και μέσω απλών εργαλείων διαχείρισης διαμορφώνουμε το μοντέλο και τη βάση και κρατάμε αντίγραφο ασφαλείας. Είναι διαθέσιμο σε Windows, Linux και Mac OS X.

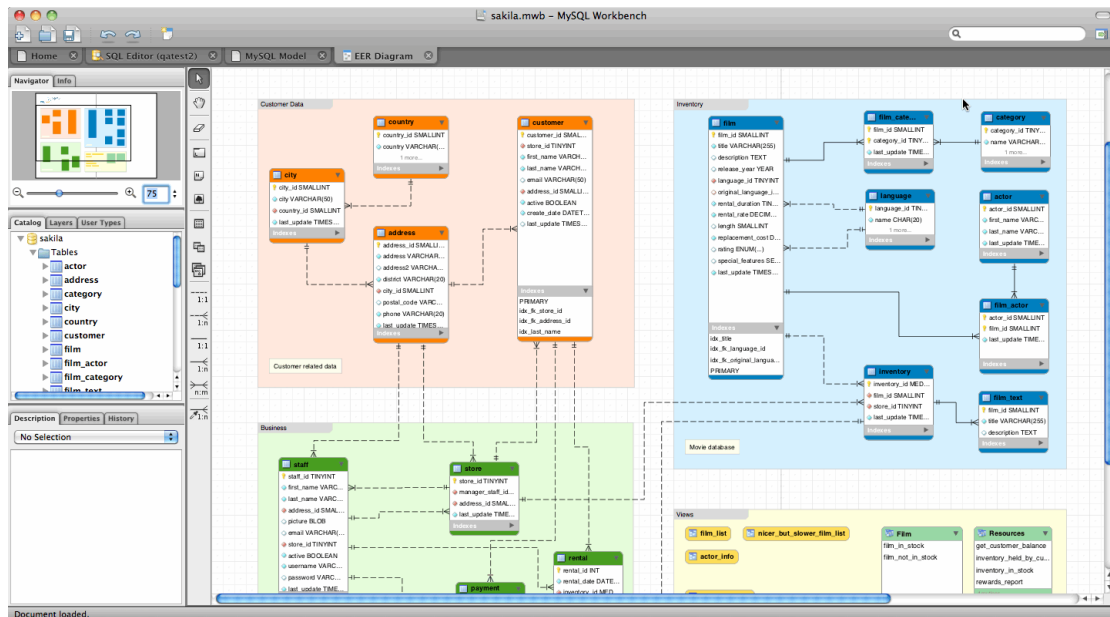


Figure 4-3 - Μοντελοποίηση Βάσης με MySQL Workbench

4.5.1 Σχεδίαση

Το MySQL Workbench δίνει τη δυνατότητα σε DBA, προγραμματιστές ή αρχιτέκτονες βάσεων να σχεδιάσουν, να μοντελοποιήσουν, να δημιουργήσουν, και να διαχειριστούν βάσεις δεδομένων. Περιλαμβάνει οτιδήποτε ένας αρχιτέκτονας δεδομένων χρειάζεται για την δημιουργία ενός πολύπλοκου ER μοντέλου και επίσης έχει χαρακτηριστικά για την εκτέλεση δύσκολων αλλαγών και εγγράφων, που συνήθως απαιτούν πολύ χρόνο και προσπάθεια.

4.5.2 Ανάπτυξη

Παρέχει εργαλεία για την δημιουργία, την εκτέλεση και τη βελτιστοποίηση των SQL queries. Το SQL Editor παρέχει χρωματική επισήμανση σύνταξης, αυτόματη ολοκλήρωση, επαναχρησιμοποίηση των SQL στιγμιότυπων και ιστορικό

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

εκτέλεσης της SQL. Ο πίνακας συνδέσεων της βάσης δεδομένων βοηθάει τους προγραμματιστές να χειριστούν εύκολα τη σύνδεση με την βάση δεδομένων, συμπεριλαμβανομένου του MySQL Fabric. Το Object Browser παρέχει στιγμιαία πρόσβαση στο σχήμα της βάσεως και στα αντικείμενα.

4.6 Γιατί Drupal

Μετά από μελέτη όλων των Συστημάτων Διαχείρισης Περιεχομένου καταλήξαμε ότι το CMS που παρουσιάζει ιδιαίτερο ενδιαφέρον είναι το Drupal λόγω των ιδιαίτερων χαρακτηριστικών του, μερικά από τα οποία αναλύονται παρακάτω:

1) Σύστημα Ανοιχτού Κώδικα

Λόγω του ότι το Drupal είναι ανοιχτού κώδικα σύστημα, έχει χαμηλό κόστος κτήσης, καθώς είναι το λιγότερο δαπανηρό για την προμήθεια αδειών λογισμικού. Παρέχει μεγαλύτερη αξιοπιστία, αφού ο κώδικας διατίθεται δημοσίως και ελέγχεται καθημερινώς όσον αφορά την ποιότητά του από μηχανικούς λογισμικού.

2) Συμβατό με διάφορες τεχνολογίες υποδομής

Το Σύστημα Διαχείρισης του Drupal συνεργάζεται με διάφορα λειτουργικά Συστήματα (Linux, Windows, MacOS), με διάφορες τεχνολογίες HTTP Server (Apache, Nginx ή Microsoft IIS HTTP Server), με διάφορες υλοποιήσεις επίσης PHP (Zend ή HHVM PHP), καθώς επίσης και με ΣΔΒ (MySQL, PostgreSQL, Maria DB, κ.λ.π).

3) Περιλαμβάνει ανεπτυγμένο Υποσύστημα Διαχείρισης της Δομής του Περιεχομένου

Οτιδήποτε καταχωρήσει ο χρήστης στην εφαρμογή ορίζεται ως αντικείμενο περιεχομένου. Κάθε αντικείμενο αντιμετωπίζεται στο συγκεκριμένο CMS ως ξεχωριστή «οντότητα».

Οι τυπικές οντότητες που ορίζονται είναι οι κόμβοι περιεχομένου (content nodes), οι όροι ταξινόμησης (taxonomy terms), τα στοιχεία μενού (menu items), οι χρήστες (users), τα σχόλια (comments) και τα αρχεία (files).

Οι «κόμβοι περιεχομένου» αποτελούν τον βασικό τύπο οντότητας. Διαχωρίζονται κυρίως σε 2 υπο-τύπους, τα Basic Pages και τα Articles, αλλά οι διαχειριστές (administrators) της εφαρμογής μπορούν να προσθέσουν ακόμα περισσότερους.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Όσον αφορά τους «όρους ταξινόμησης», με αυτόν τον τρόπο χαρακτηρίζεται το περιεχόμενο, ανάλογα με τον υπο-τύπο που ανήκει. Κατ' αυτόν τον τρόπο γίνεται και η ιεραρχική καταχώρηση διαφορετικών όρων. Οι administrators έχουν τη δυνατότητα να δημιουργήσουν πολλά vocabularies, που μοντελοποιούν ανεξάρτητες διαστάσεις χαρακτηρισμού των content types.

Τα «στοιχεία μενού» αποτελούν παρόμοιο τύπο με τον «όρο ταξινόμησης». Οι οντότητες καταχωρούνται με ιεραρχική δομή, όπως είδαμε και παραπάνω, αλλά ο συγκεκριμένος τύπος εστιάζει περισσότερο στο μενού πλοήγησης της εφαρμογής.

Στον τύπο «χρήστες» συμπεριλαμβάνονται όλοι οι χρήστες του Drupal. Έτσι, κατά τη χρήση των πεδίων επιτυγχάνεται ευελιξία στο ποια στοιχεία έχουν καταχωρηθεί από κάθε χρήστη.

Ο τύπος των «σχολίων» αφορά τα σχόλια που μπορούν να προσθέσουν οι χρήστες σε άλλες οντότητες περιεχομένου.

Τέλος, είναι ο τύπος των «αρχείων» που αφορά εικόνες και έγγραφα σε άλλες οντότητες.

4) Αρθρωτή αρχιτεκτονική (modular) και άρτια προγραμματιστική διεπαφή (API)

Με τα modules επιτυγχάνεται η βέλτιστη λειτουργικότητα του Drupal παρέχοντας συγκεκριμένες λειτουργίες. Οι μηχανικοί λογισμικού που αναπτύσσουν μια εφαρμογή στο Drupal χρησιμοποιούν μόνο τα modules που απαιτούνται για τη λειτουργία της συγκεκριμένης εφαρμογής.

Μέσω του API οι μηχανικοί δημιουργούν τα κατάλληλα modules για την εφαρμογή. Πιο συγκεκριμένα, χρησιμοποιούν τα hooks για να αλλάξουν τη συμπεριφορά της εφαρμογής, χωρίς να πειράξουν τον βασικό κώδικα του Drupal.

5) Άριστη γραφική διεπαφή χρήστη (UI)

Το Drupal χρησιμοποιεί από default τη μηχανή θέματος PHP Template. Το «theming» είμαι απλό και δεν χρειάζεται ιδιαίτερη γνώση σε PHP. Το ενεργοποιημένο theme ορίζει το πώς θα παρουσιάζεται το περιεχόμενο στην εφαρμογή.

Κάθε theme αποτελείται από ένα σύνολο διαφορετικών περιοχών όπως είναι οι παρακάτω:

- Κύρια περιοχή περιεχομένου

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

- Περιοχή κεφαλίδας
- Περιοχή μενού πλοήγησης
- Περιοχές πλευρικών ζωνών
- Περιοχή προβολής λαθών και ενδείξεων
- Περιοχή υποσέλιδου

Όλες οι περιοχές που αναφέρθηκαν μπορούν να τροποποιηθούν ανάλογα με τις γραφιστικές ανάγκες της εφαρμογής. Πάνω στις περιοχές μπορούν να υπάρχουν blocks, τα οποία ενδεχομένως να εμπεριέχουν κάποιο στατικό HTML για ένα ενημερωτικό κείμενο, είτε κάποιες λειτουργίες που παράγονται από ένα εγκατεστημένο module, όπως μια φόρμα login, προβολή ενός μενού κ.α.

Το Drupal επίσης περιέχει και ειδικές βιβλιοθήκες για την επεξεργασία εικόνας ώστε να μετατρέπει τις εικόνες στην κατάλληλη μορφή για τη χρησιμοποίησή τους.

6) Χρήστες και δικαιώματα πρόσβασης

Διαθέτει πλήρες υποσύστημα χρηστών, των ρόλων τους στην εφαρμογή και των δικαιωμάτων πρόσβασής τους.

7) Άρτια υποδομή για αναζήτηση

Με το module «search API» το Drupal υποστηρίζει:

- ✓ Πλήρη text αναζήτηση περιεχομένου βάσει του τίτλου.
- ✓ Δυνατότητα καθορισμού λέξεων μικρής σημασίας που δεν γίνονται indexed (κοινές λέξεις).
- ✓ Εξελιγμένο search ranking που καθορίζουν οι διαχειριστές της εφαρμογής για κάθε πεδίο ανάλογα με τη σημασία του.
- ✓ Δυνατότητα παραγωγής του αποσπάσματος μετά την αναζήτησή του.

8) Ασφάλεια και αυθεντικότητα δεδομένων

Αρχικά το Drupal διαθέτει υποσύστημα διαχείρισης κατά των επιθέσεων XSS, κάτι που είναι ιδιαίτερα σημαντικό όταν οι φόρμες κάνουν χρήση ajax.

Κάθε πεδίο οντότητας ρυθμίζεται με τέτοιο τρόπο ώστε να δέχεται μόνο τιμές που πληρούν συγκεκριμένα κριτήρια, όπως είναι η επιλογή μόνο προκαθορισμένων τιμών με χρήση στατικής λίστας, η ημερομηνία αρχής να είναι νεότερη από την ημερομηνία τέλους κ.α.

Στο Drupal είναι ενσωματωμένα αρχεία .htaccess που ρυθμίζουν αυτόματα τον HTTP Server, ώστε να αποφευχθούν συνήθη κενά ασφαλείας.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Με την εγκατάσταση ειδικών modules ελέγχεται η ακεραιότητα του κώδικα για τυχόν αλλοιώσεις από κακόβουλο λογισμικό, και παρέχει μηχανισμούς για την αποφυγή επιθέσεων spam (μέθοδος captcha). Διαθέτει επίσης ολοκληρωμένο μηχανισμό λαθών και debugging πληροφοριών.

9) Στατιστικά στοιχεία επισκεψιμότητας

Το Drupal είναι ρυθμισμένο έτσι ώστε να καταγράφει στατιστικά επισκεψιμότητας ανά σελίδα.

10) Πλήρες σύστημα Newsletter

Μέσω εγκατεστημένων modules δίνεται η δυνατότητα εγγραφής ή διακοπής συνδρομής των χρηστών σε ένα newsletter με το πάτημα ενός html button, καθώς επίσης και μαζικής αποστολής newsletters με μηχανισμό throttling, όπου στέλνει συγκεκριμένο αριθμό newsletters ανά συγκεκριμένο χρονικό διάστημα (π.χ 30 κάθε 10 λεπτά), ώστε να μην επηρεάζεται η απόδοσή του server.

11) Διαθέτει πρόσθετες βιβλιοθήκες

Το Drupal Core περιλαμβάνει τις βασικές Javascript βιβλιοθήκες (jQuery UI), αλλά μπορούν να προστεθούν επιπλέον ξεχωριστές βιβλιοθήκες με εξειδικευμένα plugins για επιπλέον λειτουργίες της εφαρμογής.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

ΚΕΦΑΛΑΙΟ 5

5. Προγραμματιστικά εργαλεία, ρυθμίσεις και εφαρμογές τους

5.1 Εγκατάσταση των εργαλείων που χρησιμοποιήθηκαν

Πριν την εγκατάσταση Drupal στο web directory, απαιτείται προετοιμασία του συστήματος. Ενώ είναι ήδη εγκατεστημένα Apache, PHP, και MySQL, χρειάζονται κάποιες πρόσθετες αλλαγές.

5.1.1 Διαμόρφωση ενός MySQL χρήστη και βάσης δεδομένων για το Drupal

Είναι σημαντικό να διαμορφωθεί ένας αποκλειστικός χρήστης και βάση δεδομένων για λόγους ασφαλείας.

Αρχικά, κάνουμε login στην MySQL με την εντολή: `mysql -u root -p`.

Έπειτα, στο mysql prompt, δημιουργούμε τη βάση δεδομένων για το Drupal
`CREATE DATABASE Drupal;`

Στη συνέχεια, δημιουργούμε έναν χρήστη για την σύνδεση με την βάση δεδομένων.

```
CREATE USER drupaluser@localhost IDENTIFIED BY 'κωδικός';
```

Μολονότι σ αυτό το στάδιο, υπάρχει και ο user και η βάση δεδομένων που απαιτούνταν, ο χρήστης δεν έχει ακόμα δικαιώματα στην βάση δεδομένων.

Με το παρακάτω SELECT, θα επιτραπεί στο λειτουργικό να λειτουργήσει, χωρίς την έκθεση της βάσης μας άσκοπα:

```
GRANT
```

```
SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX,ALTER,CREATE  
TEMPORARY TABLES,LOCK TABLES ON Drupal.* TO drupaluser@localhost;
```

Με αυτή την εντολή δίνονται δικαιώματα στον user να διαχειριστεί την βάση που δημιουργήθηκε.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
root@stauroulini-HP-15-Notebook-PC: /home/stauroulini
boot  etc  initrd.img.old  lost+found  opt  run  sys  var
cdrom  home  lib  media  proc  sbin  tmp  vmlinuz
root@stauroulini-HP-15-Notebook-PC:/# cd home
root@stauroulini-HP-15-Notebook-PC:/home# ls
stauroulini
root@stauroulini-HP-15-Notebook-PC:/home# cd stauroulini
root@stauroulini-HP-15-Notebook-PC:/home/stauroulini# ls
Desktop  Downloads  Music  Public  Videos
Documents  examples.desktop  Pictures  Templates
root@stauroulini-HP-15-Notebook-PC:/home/stauroulini# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.41-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

Figure 5-1 – Εγκατάσταση των απαραίτητων εργαλείων στο τερματικό

5.1.2 Εγκατάσταση PHP Modules και βελτίωση της διαμόρφωσης.

Αφού ολοκληρώσουμε τα απαραίτητα updates στα παρακάτω πακέτα,

```
sudo apt-get update
```

```
sudo apt-get install php5-gd php5-curl libssh2-php
```

Συνεχίζουμε με βελτιώσεις στο αρχείο διαμόρφωσης PHP, κάνοντας τις εξής ενέργειες:

```
sudo nano /etc/php5/apache2/php.ini
```

Τέλος, μας ανοίγει ένα παράθυρο του text editor, για να κάνουμε τις απαραίτητες και επιθυμητές αλλαγές μας στη διαμόρφωση.

5.1.3 Ενεργοποίηση της Λειτουργίας Rewrite και των Htaccess Files στον Apache

Η ενεργοποίηση της λειτουργίας του rewrite, επιτρέπει στο Drupal site να τροποποιήσει τα URLs σε strings φιλικά προς τον άνθρωπο.

```
sudo a2enmod rewrite
```

Έπειτα προσαρμόζουμε το virtual host configuration (διαμόρφωση), έτσι ώστε να επιτρέπεται η χρήση ενός .htaccess αρχείου. Σε αυτό το αρχείο θα πριέχονται οι κανόνες rewrite και συμπεριλαμβάνεται από default στην εγκατάσταση του Drupal.

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Κάνοντας τις επιθυμητές αλλαγές μέσα στο αρχείο, έχουμε έτοιμες τις αλλαγές σε PHP και Apache και τελειώνουμε με μια επανεκκίνηση του τελευταίου.

```
sudo service apache2 restart
```

5.1.4 Εγκατάσταση των αρχείων του Drupal

Με όλες τις παραπάνω ενέργειες που έγιναν, το σύστημα τώρα είναι έτοιμο για την εγκατάσταση του Drupal στο web root. Αντιγράφοντας τα αρχεία από το home directory, στην κατάλληλη τοποθεσία, θα μας δώσει έτοιμη πρόσβαση στα αρχικά αρχεία, στην περίπτωση που γίνει κάποιο λάθος ή διαγραφούν κάποια αρχεία αργότερα από καθαρό λάθος.

Από την σελίδα download του Drupal κατεβάζουμε την τελευταία έκδοσή του και αποσυμπιέζουμε το αρχείο για να μπορέσουμε να αντιγράψουμε όλα τα αρχεία ασφαλώς στο web root directory του server μας.

```
cd drupal-7.34
```

```
sudo rsync -avz . /var/www/html
```

Για την ασφάλεια και την ευκολία του χρήστη χρειάζεται η δημιουργία ενός καινούργιου directory μέσα στο web root directory μας, όπως φαίνεται και στην παρακάτω εντολή:

```
mkdir /var/www/html/sites/default/files
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

και αφού κάνουμε το αρχείο με τα default settings copy σε αρχείο που χρησιμοποιεί το Drupal για το configuration, προσθέτουμε δικαίωμα εγγραφής για το καινούργιο αρχείο.

```
cp/var/www/html/sites/default/default.settings.php
```

```
/var/www/html/sites/default/settings.php
```

```
chmod 664 /var/www/html/sites/default/settings.php
```

Στη συνέχεια μπαίνουμε στη σελίδα localhost/drupal και επιλέγουμε *install Drupal in English*, και εμφανίζεται η παρακάτω σελίδα

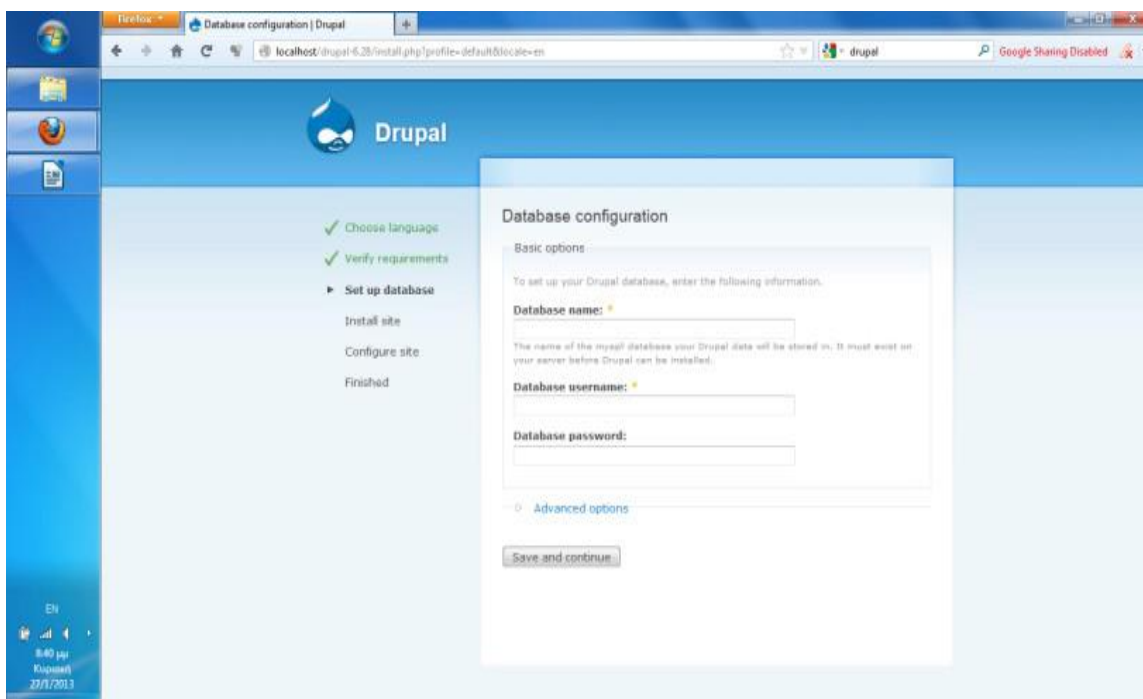


Figure 5-2 - Εγκατάσταση του Drupal (1^ο Βήμα)

Και συνεχίζοντας από εκεί που είχαμε μείνει όσον αφορά την εγκατάσταση του drupal, αφού ολοκληρώσουμε με την βάση δεδομένων μας, τώρα μπορούμε να προσθέσουμε τα στοιχεία.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα



Figure 5-3 - Εγκατάσταση του Drupal (2^ο Βήμα)

Και μετά το «Save» γίνεται η εγκατάσταση του drupal.

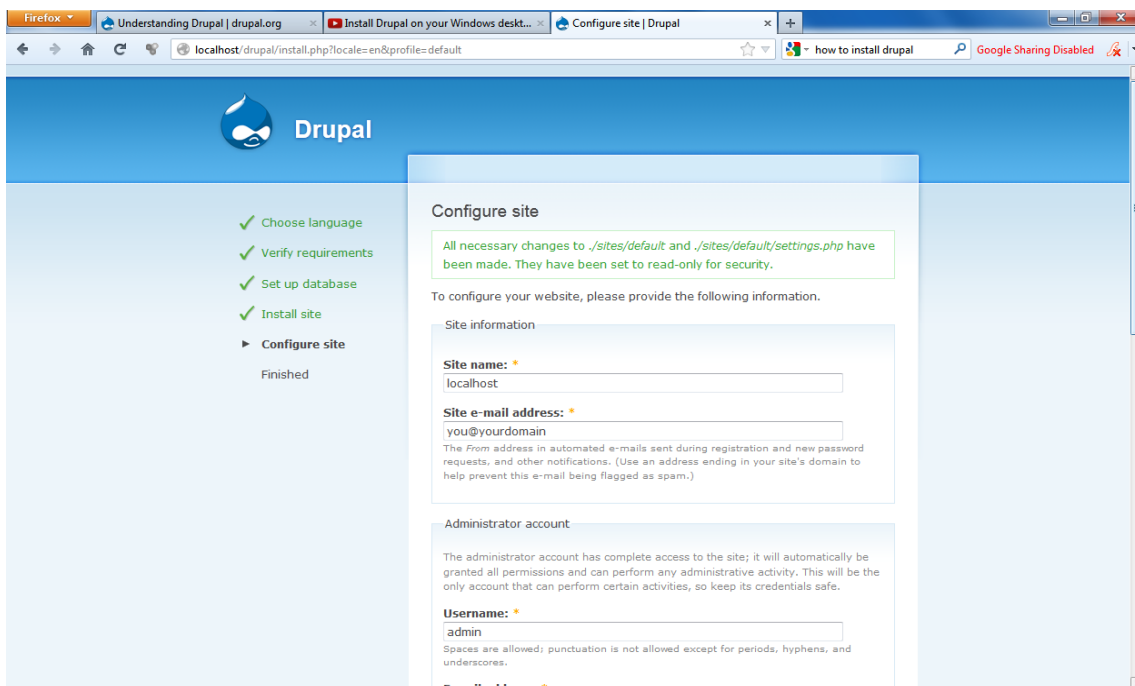


Figure 5-4 - Εγκατάσταση του Drupal (3^ο Βήμα)

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Συμπληρώνουμε τα υπόλοιπα στοιχεία για να τελειοποιήσουμε το configuration του site μας.

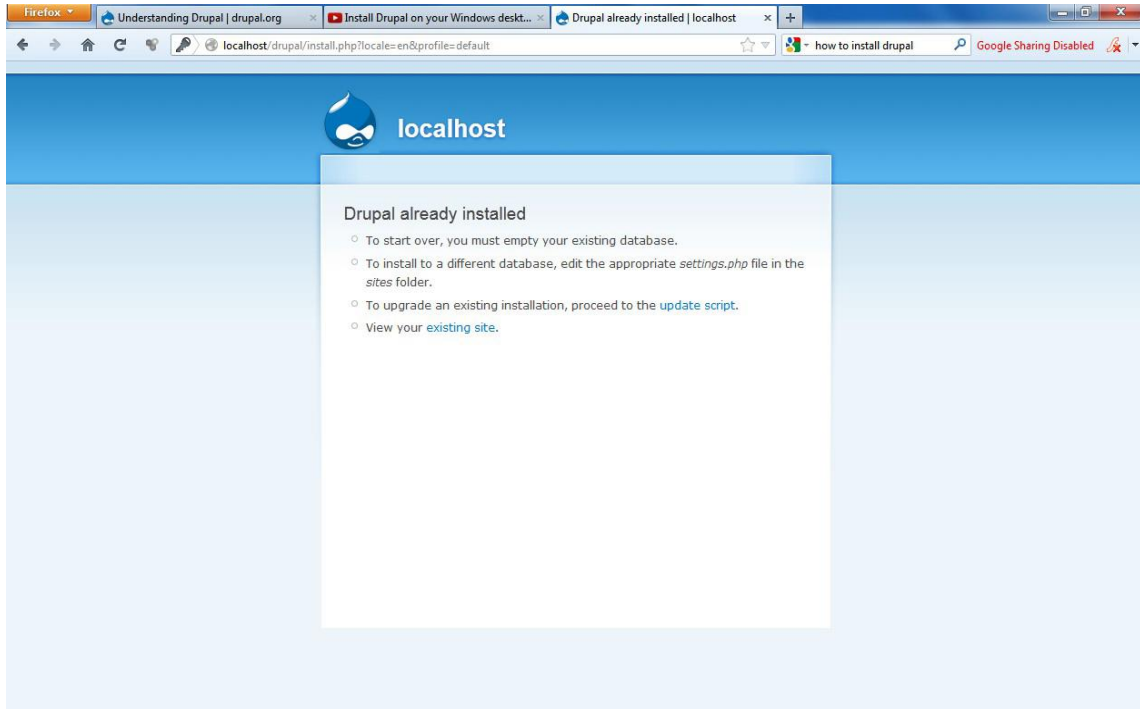


Figure 5-5 - Ολοκλήρωση εγκατάστασης του Drupal (4^ο Βήμα)

Και η εγκατάσταση έχει πλέον ολοκληρωθεί.

5.1.5 Εγκατάσταση του rhpMyAdmin

Μετά το τυπικό install

```
sudo apt-get update
```

```
sudo apt-get install rhpmyadmin
```

Συνεχίζουμε με ενεργοποίηση της επέκτασης *rhp5-mcrypt* σύμφωνα με την εντολή:

```
sudo rhp5enmod mcrypt
```

και τέλος, το μόνο που χρειάζεται είναι μια επανεκκίνηση του Apache για να ενεργοποιηθεί

```
sudo service apache2 restart
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Τώρα πλέον, υπάρχει πρόσβαση στο web interface βάζοντας το domain name του server ή την δημόσια IP address, ακολουθούμενο από */phpmyadmin*:

http://domain_name_or_IP/phpmyadmin

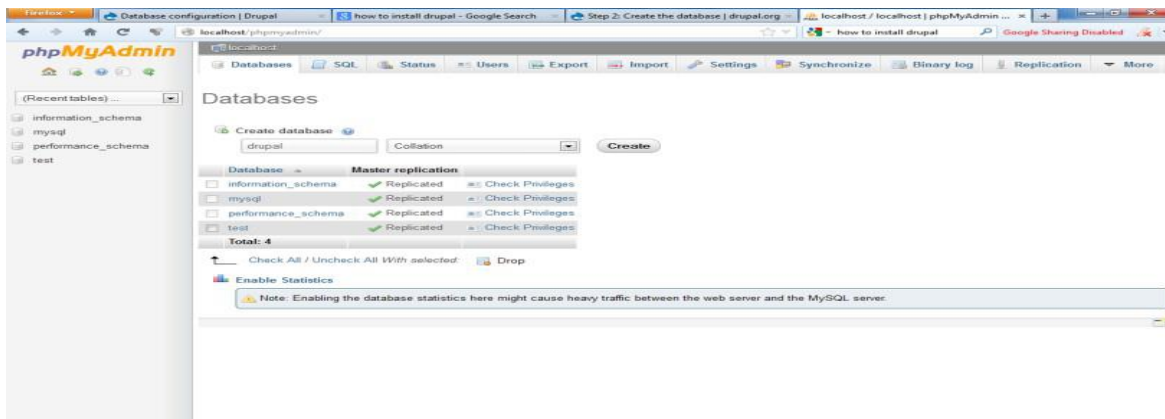


Figure 5-6 - PhpMyAdmin

Τέλος, προσθέτουμε τη βάση «**drupal**» και πατάμε «**create**», και η βάση μας έχει δημιουργηθεί.

5.1.6 Εγκατάσταση του MySQL Workbench

Η εγκατάσταση του συγκεκριμένου εργαλείου για την αρχιτεκτονική της βάσης δεδομένων γίνεται εύκολα ακολουθώντας τα εξής βήματα:

Αρχικά κάνουμε download το αρχείο από την επίσημη ιστοσελίδα <http://dev.mysql.com/downloads/workbench/> και διαλέγουμε ανάλογα με την έκδοση που επιθυμούμε και το λειτουργικό που χρησιμοποιούμε.

Έπειτα στο τερματικό τρέχουμε τις παρακάτω εντολές:

```
sudo apt-get remove mysql-workbench mysql-workbench-data
dpkg-deb -x mysql-workbench-community-6.*.deb tmp_mysql-workbench
dpkg-deb --control mysql-workbench-community-6.*.deb tmp_mysql-
workbench/DEBIAN
sed -i -e 's/mysql-client, //' tmp_mysql-workbench/DEBIAN/control
dpkg -b tmp_mysql-workbench mysql-workbench-community_patched.deb
sudo dpkg -i mysql-workbench-community_patched.deb
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα



Figure 5-7 - Εγκατάσταση του MySQL Workbench

5.2 Σχετικές εικόνες κατά την υλοποίηση των προαναφερθέντων εργαλείων

Στο υποκεφάλαιο αυτό θα δείξουμε την ακριβή εικόνα των εργαλείων που εγκαταστάθηκαν ,για την εκπόνηση της εργασίας.

5.2.1 Η αρχική σελίδα του Drupal site

Σε αυτή την εικόνα εμφανίζεται η αρχική σελίδα του Drupal site που δημιουργήσαμε μετά την εγκατάσταση του τοπικού μας server (Εικόνα 5.8).

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

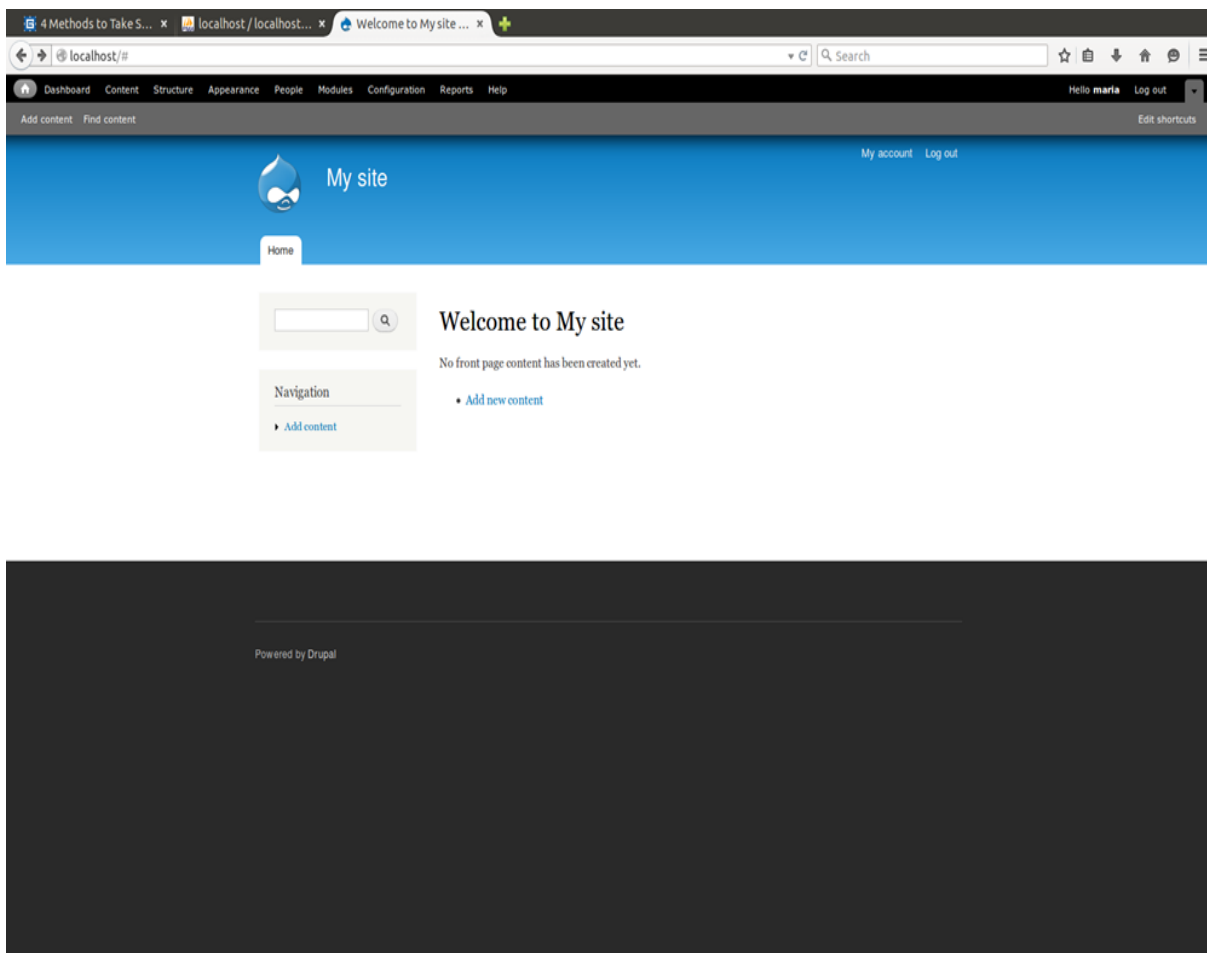


Figure 5-8 - Localhost

5.2.2 Βάση δεδομένων – PhpMyAdmin

Εδώ μας απεικονίζεται το εργαλείο, με το οποίο μπορούμε να δημιουργήσουμε τους πίνακες που χρειάζονται για την βάση δεδομένων της διαδικτυακής μας εφαρμογής (Εικόνα 5.9).

Από την αριστερή μεριά, όπως φαίνεται και στην εικόνα, μας δίνεται η δυνατότητα δημιουργίας πινάκων για την βάση μας.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

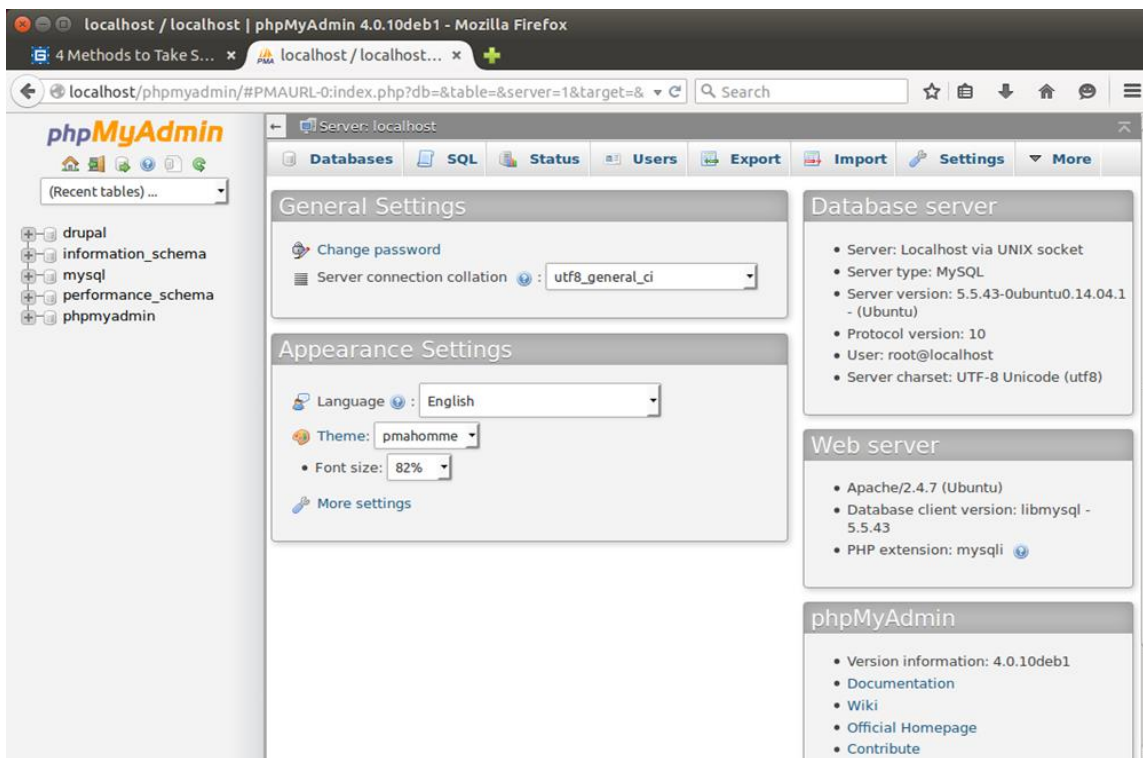


Figure 5-9 - Η βάση του Drupal στο PhpMyadmin

Στην περίπτωση της εργασίας μας, επιλέγουμε **drupal**, μετά **New** και στη συνέχεια δημιουργείται το επιθυμητό table και με τα χαρακτηριστικά που κρίνουμε εμείς ότι θα πρέπει να διαθέτει (όπως για παράδειγμα table για strings με συγκεκριμένο μήκος).

Σε περίπτωση που αποφασίσουμε να κρατήσουμε backup της βάσης μας, μπορούμε πάρα πολύ απλά να κάνουμε **export** το αρχείο **«όνομα».sql**.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

5.2.3 Μοντελοποίηση βάσης στο MySQL Workbench

Όπως προαναφέρθηκε σε προηγούμενη ενότητα, το εργαλείο αυτό μας βοηθά στην αρχιτεκτονική και μοντελοποίηση της βάσης δεδομένων που θέλουμε να αναπτύξουμε.

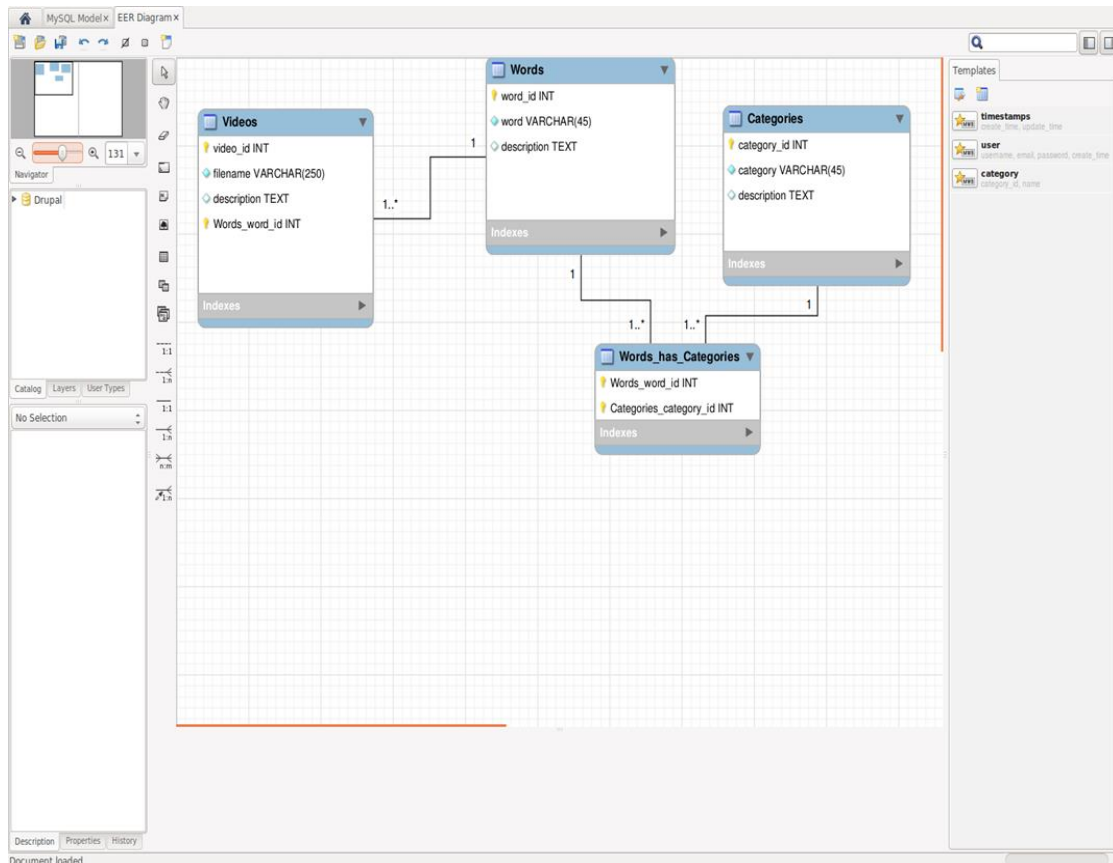


Figure 5-10 - Η μορφή της βάσης μας στο Workbench

Όπως φαίνεται και στην παραπάνω εικόνα, για την εκπόνηση της πτυχιακής μας έχουν δημιουργηθεί 3 πίνακες, videos, words και categories, καθώς επίσης και σχέσεις μεταξύ τους, που υποδηλώνονται με τις αντίστοιχες γραμμές.

5.2.4 Εγκατάσταση των modules

Για να ενεργοποιήσει ο server τα modules με τον κώδικα που έχουμε δημιουργήσει και βρίσκονται σε συγκεκριμένο path στο root web directory μας, πρέπει πρώτα να τα επιλέξουμε από τον localhost που φαίνεται στην ενότητα 4.2.1.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Για να ενεργοποιήσουμε τα modules, πατάμε «**Modules**» από το μενού στο localhost.

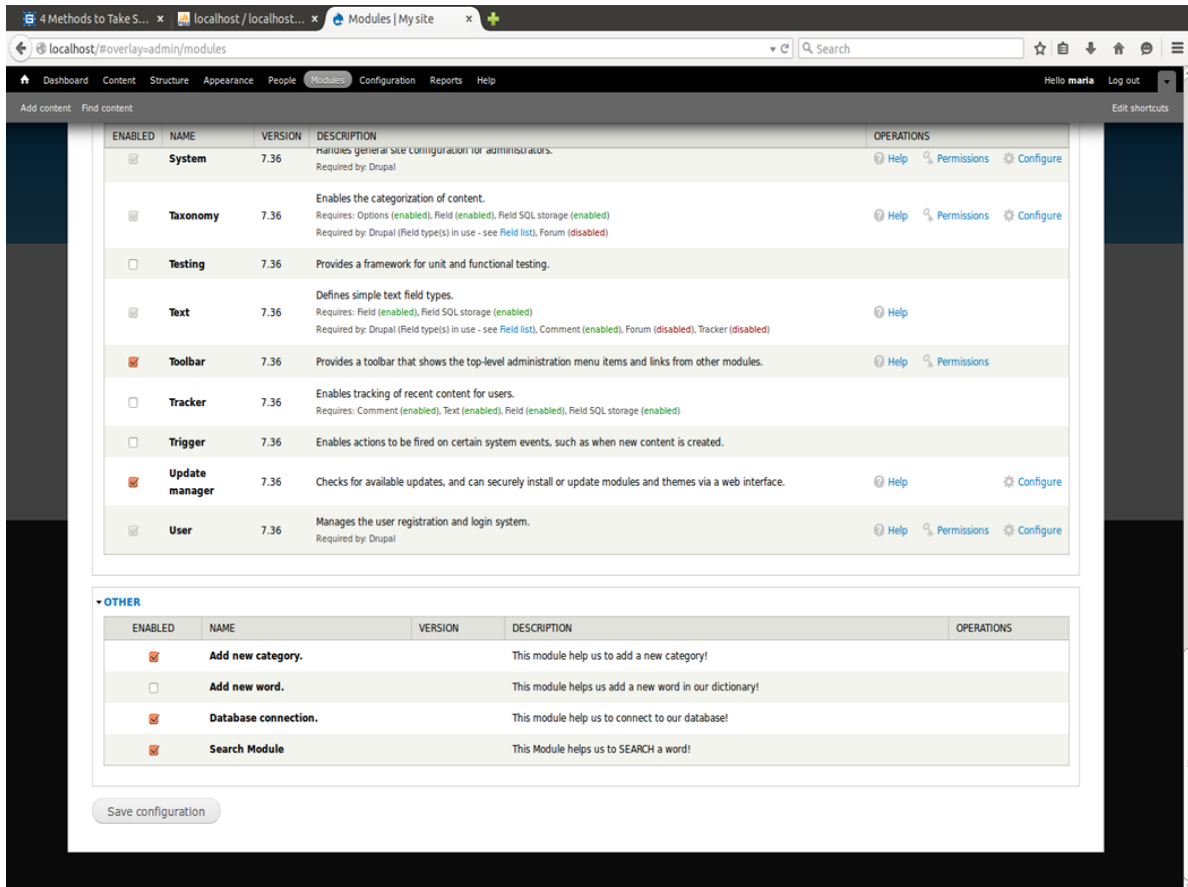


Figure 5-11 - Ενεργοποίηση των modules

Βλέπουμε τη λίστα με τα modules που έχουν εγκατασταθεί και σημειώνουμε με τικ το module που μας ενδιαφέρει.

Τέλος πατάμε το button «**Save configuration**».

ΚΕΦΑΛΑΙΟ 6

6. Τεχνικές Λεπτομέρειες - Εφαρμογές σε κώδικα / modules

6.1. Αρχιτεκτονική, Σχεδίαση και case – scenarios της εφαρμογής

Είναι κάτι πραγματικά απλό για τους developers, να ξεκινούν να προγραμματίζουν και να γράφουν κώδικα για μία εφαρμογή χωρίς μία σωστή αρχιτεκτονική τοποθέτηση. Σε μια τέτοια περίπτωση, συνήθως ο πηγαίος κώδικας είναι ανοργάνωτος και δύσκολα υποβάλλεται σε αλλαγές, καθώς είναι πολύ δύσκολο να καθοριστούν τα χαρακτηριστικά δόμησης της εφαρμογής χωρίς την πλήρη κατανόηση λειτουργίας των στοιχείων και modules από τα οποία αυτή αποτελείται.

Μέσα, λοιπόν, από τα αρχιτεκτονικά σχέδια γνωρίζουμε και κατανοούμε τη συμπεριφορά της εφαρμογής. Το πιο γνωστό σύστημα αρχιτεκτονικής είναι το σύστημα επιπέδων (**layered architecture**).

6.1.1 Σύστημα Επιπέδων Layered Architecture

Τα συστατικά στοιχεία στην αρχιτεκτονική επιπέδων οργανώνονται σε οριζόντια επίπεδα, εκ των οποίων το καθένα έχει τη δική του λειτουργία μέσα στην εφαρμογή. Παρά το ότι το σύστημα αρχιτεκτονικής επιπέδων δεν προσδιορίζει τον αριθμό και τον τύπο των επιπέδων που πρέπει να υπάρχουν, τα επίπεδα που συνήθως χρησιμοποιούνται είναι 4: *Presentation layer*, *Business layer*, *Persistence* και *Database layer* (εικόνα 6.1). Τα επίπεδα *Business* και *Persistence* μπορεί να συνδυάζονται σε ένα ενιαίο επίπεδο, κυρίως όταν το *Persistence* επίπεδο μπορεί να εμπεριέχεται μέσα στο *Business*.

Κάθε επίπεδο αυτού του συστήματος έχει το δικό του ρόλο λειτουργίας μέσα στην εφαρμογή. Για παράδειγμα, το *Presentation Layer* είναι υπεύθυνο για τα ζητήματα διεπαφής της εφαρμογής με τον χρήστη (*User – User Interface*). Κάθε επίπεδο έχει τη δική του ευθύνη και δεν ασχολείται με τις πληροφορίες ή τα αιτήματα των υπόλοιπων επιπέδων. Για παράδειγμα το *Presentation Layer* θα είναι υπεύθυνο για όλο το *User Interface* και την άμεση επικοινωνία του χρήστη

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

με τον browser, ενώ το Business Layer ασχολείται αποκλειστικά με την εκτέλεση συγκεκριμένων κανόνων ανάλογα με το κάθε αίτημα της εφαρμογής.

Κάθε στρώμα του συγκεκριμένου αρχιτεκτονικού προτύπου ενεργεί μόνο του, χωρίς να δέχεται αλληλεπιδράσεις από τα άλλα επίπεδα, έτσι ώστε να ολοκληρώσει το έργο που του έχει ζητηθεί και να ικανοποιήσει το αίτημα που έχει δοθεί από τον ίδιο τον χρήστη, ο οποίος αναμένει για το ανάλογο αποτέλεσμα.

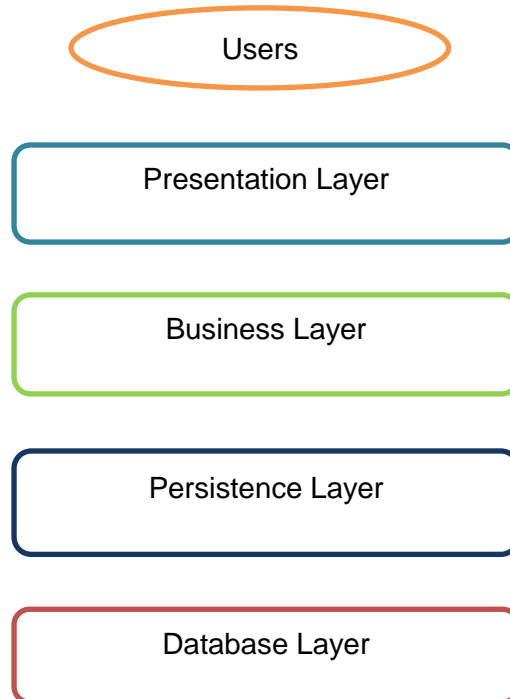


Figure 6-1 - Layered Architecture

Για παράδειγμα το *Presentation Layer* δεν χρειάζεται να γνωρίζει με ποιον τρόπο ο πελάτης θα πάρει τα ανάλογα δεδομένα που έχουν ζητηθεί. Απλά θα εμφανίσει την τελική πληροφορία στην οθόνη με το ανάλογο format. Παρομοίως, το *Business* επίπεδο, δεν χρειάζεται να γνωρίζει πώς θα εμφανιστούν τα αποτελέσματα στον πελάτη ή ακόμα και από πού προέρχονται. Το μόνο του έργο, είναι να πάρει τα δεδομένα από το *Persistence Layer*, να εφαρμόσει την αντίστοιχη λογική του business επιπέδου στα δεδομένα αυτά (να υπολογίσει τιμές, να αθροίσει κ.λ.π.), και να περάσει το τελικό αποτέλεσμα στο *Presentation Layer*.

Όπως έχουμε ήδη αναφέρει κάθε επίπεδο είναι αυτόνομο και **κλειστό**, όσον αφορά την επικοινωνία του με τα υπόλοιπα. Με τον όρο κλειστό εννοούμε ότι

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

καθώς το αίτημα μετακινείται από επίπεδο σε επίπεδο για την περάτωσή του, θα πρέπει πρώτα να περάσει από το ακριβώς επόμενο του. Δηλαδή, ένα request που φεύγει από το Presentation layer, θα πρέπει πρώτα να περάσει από το Business layer για να περάσει στο Persistence κ.ο.κ (εικόνα 6.2).

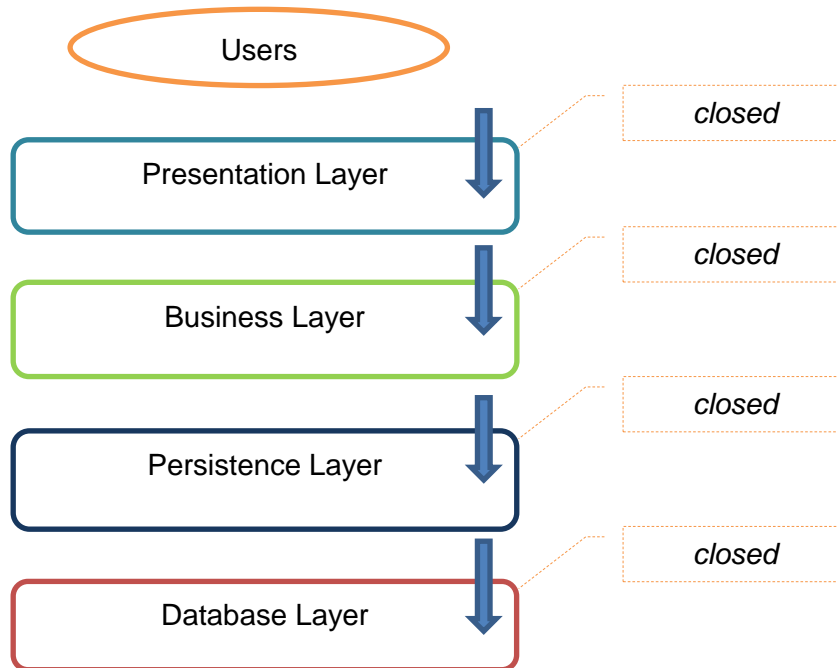


Figure 6-2 – Κλειστό σύστημα αρχιτεκτονικής και κατεύθυνση του αιτήματος

Αυτό σημαίνει ότι οποιαδήποτε αλλαγή προκύψει σε ένα επίπεδο δε θα επηρεάσει τα υπόλοιπα. Εάν, για παράδειγμα, επιτρεπόταν η πρόσβαση από το *Presentation layer* απευθείας στο *persistence*, ναι μεν θα ήταν πιο γρήγορη διαδικασία, αλλά τυχόν αλλαγές στον κώδικα του *Persistence* επιπέδου θα επηρέαζαν άμεσα και απρόβλεπτα το *Business* και το *Presentation*.

6.1.2 Εφαρμογή αρχιτεκτονικής επιπέδων στο διαδικτυακό λεξικό

Όπως αναφέραμε στην ενότητα 6.1.1, η πιο κοινή μέθοδος διαχωρισμού είναι το πολυεπίπεδο αρχιτεκτονικό μοτίβο, το οποίο χωρίζεται σε 4 πιο συχνά επίπεδα: Presentation Layer, Business Layer, Persistence Layer και Database Layer. Σε αυτή την ενότητα θα προσπαθήσουμε να εξηγήσουμε τη μέθοδο αυτή του αρχιτεκτονικού προτύπου πάνω στην διαδικτυακή εφαρμογή μας, που έχει ως θέμα ένα λεξικό, όπου ο χρήστης έχει τη δυνατότητα να εισάγει, αλλά και να λάβει δεδομένα (εικόνα 6.3).

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Άξιο να σημειωθεί είναι ότι για την περάτωση αυτής της εφαρμογής και των κατάλληλων συναρτήσεων έχει γίνει χρήση του API (Application Programming Interface). Είναι μια διεπαφή μεταξύ των στοιχείων ενός μεγάλου συστήματος λογισμικού. Για κάθε μεμονωμένο στοιχείο του συστήματος ένα API χειρίζεται την επικοινωνία μεταξύ των στοιχείων και του πυρήνα. Εμπεριέχονται ειδικές βιβλιοθήκες για την υλοποίηση και την εκτέλεση των φορμών, καθώς επίσης και την μορφοποίησή τους.

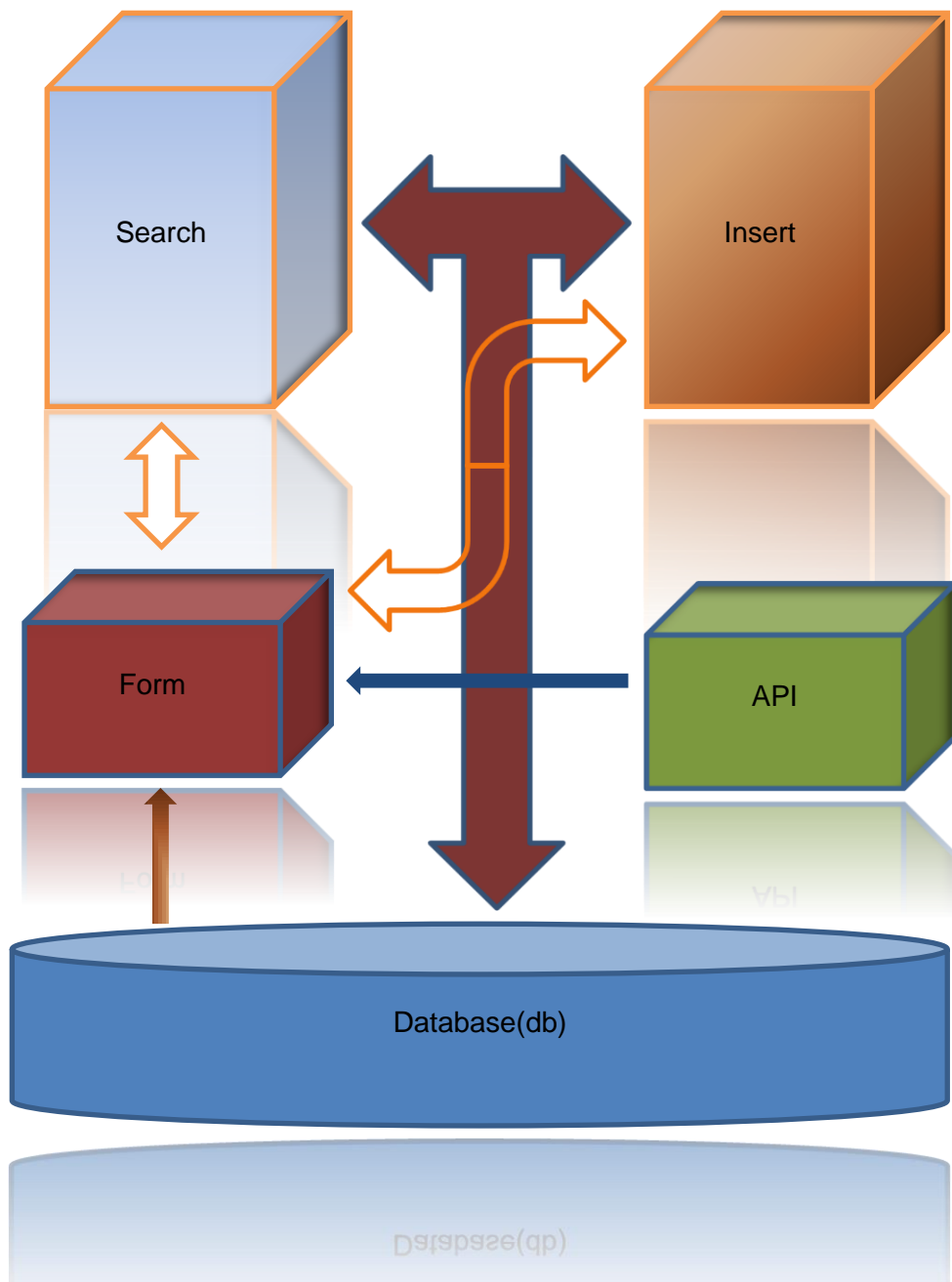


Figure 6-3 - Αρχιτεκτονικό σύστημα επιπέδων στην εφαρμογή μας

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Ξεκινώντας το αρχιτεκτονικό πρότυπο, υπάρχει το *Presentation Layer*, το οποίο ασχολείται αποκλειστικά με το User Interface (UI) της διαδικτυακής εφαρμογής. Στο σημείο αυτό ο χρήστης έχει το δικαίωμα της επιλογής, ανάμεσα σε 3 επιλογές:

- ✚ Αναζήτηση
- ✚ Εισαγωγή νέας λέξης
- ✚ Εισαγωγή νέας κατηγορίας

Με την επιλογή **Αναζήτηση**, εμφανίζονται στη φόρμα μας οι λέξεις ή οι κατηγορίες που υπάρχουν στο λεξικό - με την βοήθεια και των άλλων επιπέδων, τα οποία και θα αναλυθούν αργότερα - και υπάρχει και η **Εισαγωγή** λέξης ή κατηγορίας, όπου στο ανάλογο πεδίο ο χρήστης κάνει την ανάλογη ενέργεια.

Πρόκειται ουσιαστικά για το επίπεδο, που αποτελεί την άμεση διεπαφή του χρήστη με τη διαδικτυακή εφαρμογή, αυτά δηλαδή που μπορεί να δει. Πιο συγκεκριμένα, πρόκειται για το theme που έχει επιλεγθεί να έχει η web εφαρμογή μας, το στυλ και τα σημεία που επιθυμούμε να απεικονίζονται τα πεδία, όπου ο χρήστης είτε κάνει εισαγωγή λέξεων με την αντίστοιχη περιγραφή τους, είτε κάνει αναζήτηση μιας λέξης ή κατηγορίας.

Συνεχίζοντας με το *Business Layer*, αφορά το επίπεδο που περιέχει όλους τους μηχανισμούς, οι οποίοι υλοποιούν τη δημιουργία της φόρμας που εμφανίζεται στο προαναφερθέν επίπεδο. Από τη στιγμή που μπαίνει ο χρήστης στην διαδικτυακή εφαρμογή δίνοντας το URL, έχει ήδη στείλει request, το οποίο και ικανοποιείται άμεσα από το business επίπεδο, επιστρέφοντας και εμφανίζοντας όλες τις απαραίτητες φόρμες και το UI για την σωστή πλοήγηση του χρήστη στην εφαρμογή. Μια φόρμα μεταφράζεται συνήθως σε html με το πέρασμα του πίνακα στη συνάρτηση **drupal_get_form()**. Με τη συνάρτηση αυτή ανακτάται μία φόρμα από τη συνάρτηση κατασκευής της φόρμας ή από την κρυφή μνήμη, αν η φόρμα κατασκευάστηκε σε προηγούμενο φόρτωμα σελίδας. Έπειτα η φόρμα προωθείται για επεξεργασία και για μετάφραση σε html.

Τη στιγμή που ο χρήστης επιλέξει να κάνει κάποια αναζήτηση, ο μηχανισμός θα αντλήσει τα δεδομένα που έχουν ζητηθεί (σε συνεργασία με τα επόμενα επίπεδα), θα τα επεξεργαστεί δίνοντάς τα στην κατάλληλη συνάρτηση, ώστε να μπορέσει να μεταβιβάσει το αποτέλεσμα και να απαντήσει στο αίτημα του χρήστη. Στην περίπτωση που ο χρήστης επιλέξει να κάνει νέα εισαγωγή στην βάση του λεξικού μας το επίπεδο αυτό συμπίπτει με το επόμενο που είναι το Persistence και μέσω

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

της συνάρτησης **db_insert()** γίνεται όλη η διαδικασία, η οποία θα αναλυθεί στην επόμενη παράγραφο.

Ακολουθώντας και τη σωστή σειρά των επιπέδων, φτάνουμε στην ανάλυση του *Persistence Layer*. Το επίπεδο αυτό λειτουργεί ως ενδιάμεσο των Business και Database Layer. Πιο συγκεκριμένα, ελέγχει όλες τις ενέργειες, που απαιτούνται για την πρόσβαση στη Βάση αποθήκευσης των δεδομένων της εφαρμογής. Μέσω του επιπέδου αυτού η εφαρμογή εκτελεί τα requests στη βάση και περιμένει αποτελέσματα για να μεταφέρει πίσω τις αντίστοιχες απαντήσεις στα προηγούμενα επίπεδα. Οι περισσότερες ερωτήσεις προς την βάση γίνονται μέσω συναρτήσεων **db_query()**. Εάν πρόκειται για αίτημα εισαγωγής αντικειμένου με τη **db_insert()**, το επίπεδο αυτό είναι υπεύθυνο για την ένταξη της λέξης ή κατηγορίας στον αντίστοιχο πίνακα της βάσης με όλα τα επιπλέον αντίστοιχα πεδία που συμπληρώθηκαν από τον χρήστη στο UI, όπως είναι η περιγραφή της λέξης και το φόρτωμα του αντίστοιχου βίντεο. Εάν πρόκειται για να αντλήσει δεδομένα από την βάση μας μέσω της **db_select()**, τότε και αφού ολοκληρωθεί η διαδικασία αυτού του επιπέδου, ανοίγει νέο παράθυρο μέσω της **PagerDefault()**, όπου εμφανίζονται τα τελικά αποτελέσματα στο χρήστη αφού περάσουν από το Business επίπεδο, για το τελικό επιλεγμένο format.

Στην εφαρμογή αυτή επίσης, κάνουμε χρήση της συνάρτησης **db_delete()**, η οποία δημιουργεί έναν σύνδεσμο (link) με την αντίστοιχη λέξη που βρίσκεται στη βάση. Με αυτόν τον τρόπο όταν ο χρήστης πατήσει το button *delete*, θα διαγραφεί αυτόματα η λέξη από την βάση.

Το Database Layer περιέχει όλα τα απαραίτητα στοιχεία και τους πίνακες για μια σωστή και πλήρης βάση, με όλες τις κατάλληλες υποδομές. Περιέχει το πλήθος των πινάκων, το είδος των στοιχείων που απαρτίζουν τους πίνακες, καθώς επίσης και τη σύνδεση των πινάκων αυτών μεταξύ τους. Αρχικά, έχουμε έναν πίνακα που εμπεριέχει όλες τις λέξεις που θα περιλαμβάνει το λεξικό μας και έναν δεύτερο για όλες τις κατηγορίες. Σχηματίζουμε στη συνέχεια έναν τρίτο πίνακα που συνδέει τους δύο παραπάνω και έναν τέταρτο που συνδέει τις λέξεις με τα videos που εισάγει την ίδια στιγμή ο χρήστης.

6.1.3 Case scenarios και διαδρομές επιλογών του User

Σε αυτό το κεφάλαιο γίνεται η μελέτη όλης της διαδρομής που ακολουθεί ο client/user για να πάρει το αποτέλεσμα που θέλει και πώς αυτή επηρεάζεται από τις διαφορετικές επιλογές που θα μπορούσε να ακολουθήσει ανάλογα με τις εκάστοτε ανάγκες του.

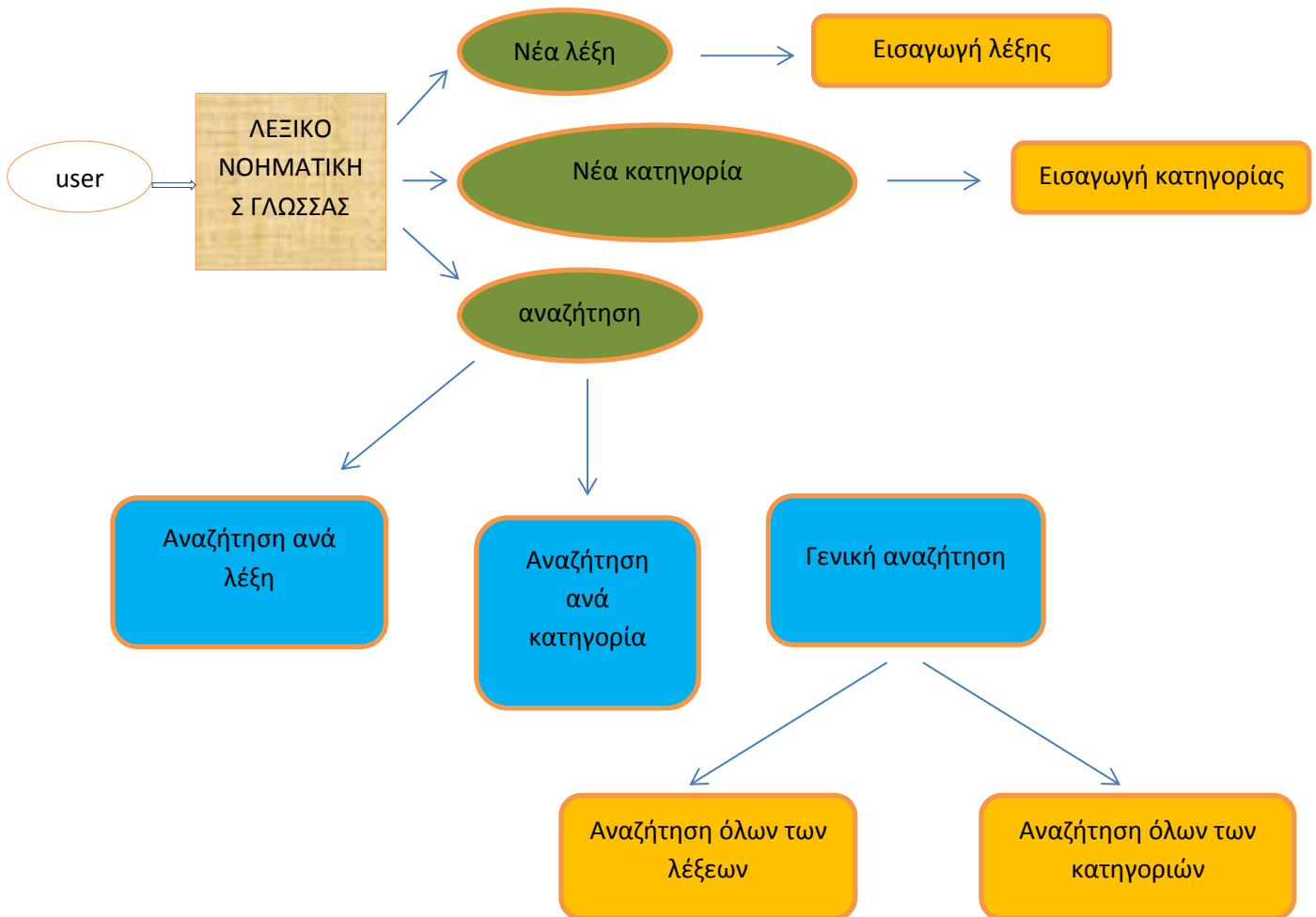


Figure 6-4 - Γενική διαδρομή όλης της εφαρμογής

Σε μια γενική εικόνα, ο χρήστης έχει πρόσβαση στο Λεξικό της Ελληνικής Νοηματικής γλώσσας. Η homepage της εφαρμογής θα του δώσει τρεις επιλογές, από τις οποίες θα μπορεί να επιλέξει μόνο τη μία, ανάλογα με το αν θέλει να εισάγει νέα λέξη, νέα κατηγορία ή να κάνει αναζήτηση συγκεκριμένου αντικειμένου μέσα στη βάση.

Από τις επιλογές που έκανε ανοίγουν νέες σελίδες, οι οποίες είτε του επιφέρουν το αποτέλεσμα που θέλει απευθείας, είτε μέσω άλλων επιλογών, όπως φαίνεται παρακάτω:

1. Εισαγωγή νέας λέξης:

Στην περίπτωση που αποφασίσει ο χρήστης να εισάγει νέα λέξη, θα τσεκάρει και την ανάλογη επιλογή στη home page του λεξικού. Πιο συγκεκριμένα, από τις τρεις επιλογές που θα έχει, νέα λέξη, νέα κατηγορία, και αναζήτηση, ο χρήστης θα επιλέξει νέα λέξη και θα ανακατευθυνθεί στη σελίδα, όπου θα του ζητηθεί να την εισάγει και να δώσει μια περιγραφή του αντικειμένου, μαζί με μια σύντομη περιγραφή του, την κατηγορία που ανήκει και το βίντεο με την ερμηνεία του στην ελληνική νοηματική.

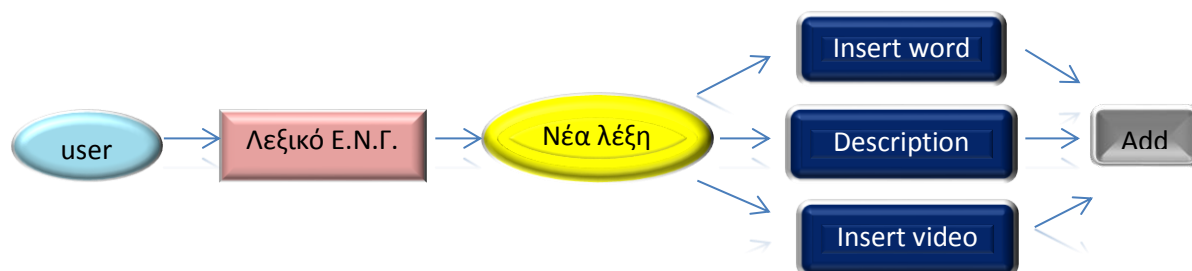


Figure 6-5 - Διαδρομή για εισαγωγή νέας λέξης

Μετά από αυτή τη διαδρομή ο χρήστης έχει εισάγει σωστά το αντικείμενο στη βάση δεδομένων και έτσι μια νέα λέξη έχει εισαχθεί στο λεξικό μας.

2. Εισαγωγή νέας κατηγορίας:

Η δεύτερη περίπτωση αφορά την επιλογή εισαγωγής νέας κατηγορίας στην αρχική σελίδα του λεξικού. Ο χρήστης θα επιλέξει νέα κατηγορία, όπου θα του ζητηθεί να συμπληρώσει 3 πεδία. Το ένα θα είναι το όνομα της νέας κατηγορίας που επιθυμεί να δημιουργήσει και το άλλο μια σύντομη περιγραφή του (εικόνα 6.6).

Το αποτέλεσμα αυτής της περίπτωσης είναι μιας επιπλέον κατηγορίας στο λεξικό.



Figure 6-6 - Διαδρομή για εισαγωγή νέας κατηγορίας

3. Αναζήτηση ανά λέξη:

Φτάνοντας στην περίπτωση της αναζήτησης από την αρχική σελίδα, ο χρήστης θα ανακατευθυνθεί σε νέα σε νέα σελίδα με νέες επιλογές, εκ των οποίων η μία θα είναι η *αναζήτηση ανά λέξη (search by word)*.

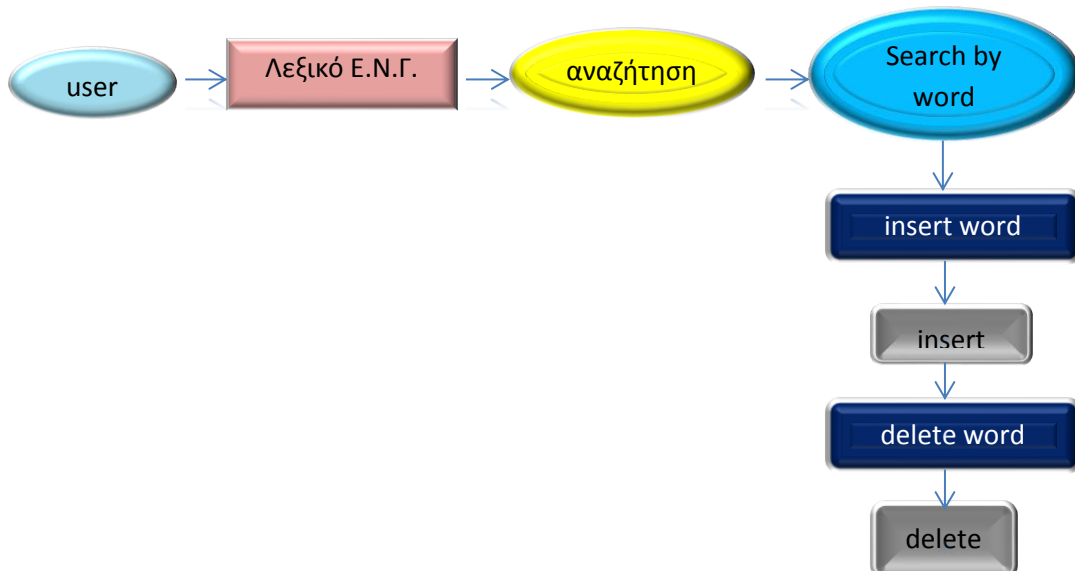


Figure 6-7 - Διαδρομή για αναζήτηση ανά λέξη

Αφού επιλέξει ο χρήστης την αναζήτηση και έπειτα την *αναζήτηση ανά λέξη (search by word)*, και πληκτρολογήσει την καινούργια λέξη, η επόμενη του κίνηση είναι να πατήσει το κουμπί insert. Μετά το πέρας της εισαγωγής θα έχει τη δυνατότητα, αν το επιθυμεί και ο ίδιος ο χρήστης, να κάνει διαγραφή γράφοντας τη λέξη στο ανάλογο πεδίο και πατώντας το κουμπί delete.

4. Αναζήτηση ανά κατηγορία:

Στην περίπτωση αυτή, ο χρήστης ακολουθεί την εξής διαδρομή: Στη home page της εφαρμογής επιλέγει αναζήτηση και έπειτα στη νέα σελίδα, στην οποία τον ανακατευθύνει επιλέγει *αναζήτηση ανά κατηγορία (search by category)*.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Έπειτα, στην καινούργια σελίδα που εμφανίζεται εισάγει στο πεδίο την κατηγορία που επιθυμεί να εμφανιστεί μαζί με μια περιγραφή του τι περιλαμβάνει του τι περιλαμβάνει.

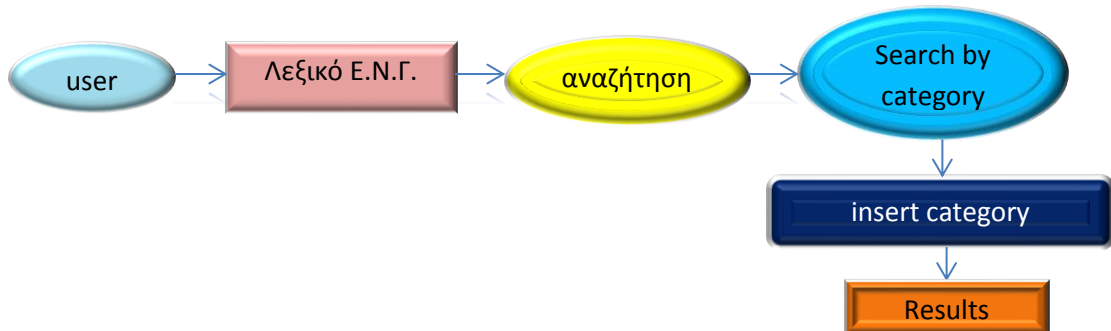


Figure 6-8 - Διαδρομή για αναζήτηση ανά κατηγορία

5. Γενική αναζήτηση όλων των λέξεων

Όπως ήδη έχουμε αναφέρει από την Αναζήτηση της αρχικής σελίδας του λεξικού, ο χρήστης ανακατευθύνεται στη φόρμα για το είδος της αναζήτησης που θέλει να κάνει. Στην περίπτωση που θέλει να δει από ποιες λέξεις απαρτίζεται το λεξικό μας, πολύ απλά πατάει στο κουμπί της γενικής αναζήτησης. Στη νέα φόρμα που θα εμφανιστεί επιλέγει το κουμπί «Όλες οι λέξεις (*all the words*)».

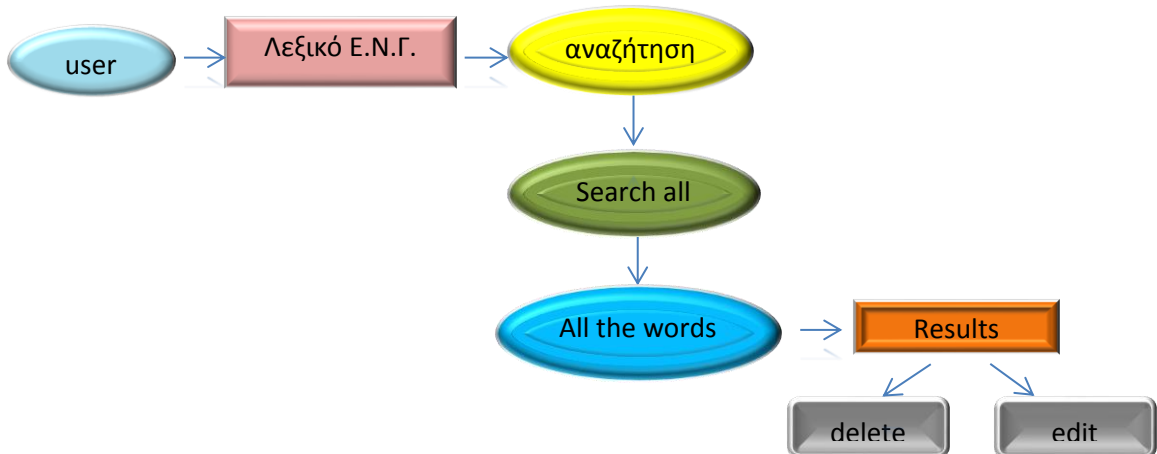


Figure 6-9 - Διαδρομή αναζήτησης όλων των λέξεων που υπάρχουν στη βάση της εφαρμογής

Τότε θα εμφανιστούν όλες οι λέξεις που είναι αποθηκευμένες στη βάση με 2 εξτρά επιλογές. Ανάλογα με το αν θέλουμε να διαγράψουμε κάποια λέξη ή αν θέλουμε να την επεξεργαστούμε θα υπάρχει και η ανάλογη λέξη, delete και edit αντίστοιχα.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

6. Γενική αναζήτηση όλων των κατηγοριών

Αντίστοιχα η ίδια διαδικασία και το ίδιο μονοπάτι θα κατευθύνει στην αναζήτηση των κατηγοριών. Η μόνη διαφορά είναι ότι θα επιλέξει «**Όλες τις κατηγορίες (all the categories)**».

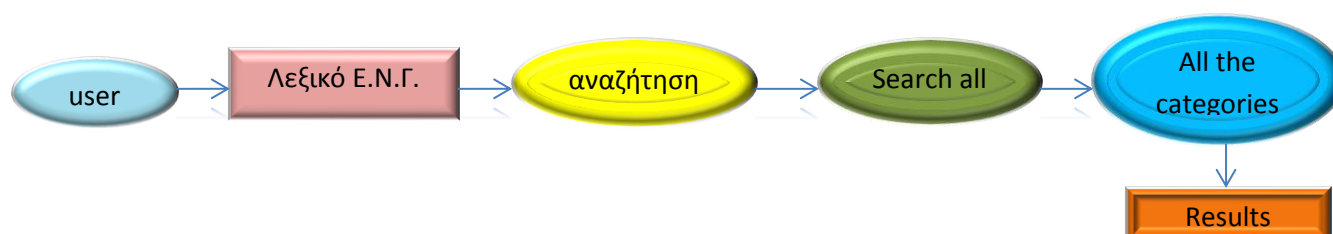


Figure 6-10 - Διαδρομή αναζήτησης όλων των κατηγοριών που υπάρχουν στη βάση της εφαρμογής

6.2. Περιγραφή των αρχείων «.info» και «.module» ως μέσον δημιουργίας του λεξικού μας

Το Drupal χρησιμοποιεί αρχεία .info για να αποθηκεύσει πληροφορίες σχετικά με τα themes και τα modules.

Για τα modules, το .info αρχείο χρησιμοποιείται για:

- Να καθιστά πληροφορίες στα gui των σελίδων του administrator
- Να μας παρέχει κριτήρια για το έλεγχο ενεργοποίησης και απενεργοποίησης του module
- Ενημερώνει το Drupal για την ύπαρξη ενός module
- Γενικές χρήσεις από τον admin σε άλλα περιεχόμενα

Αυτό το .info αρχείο απαιτείται για να αναγνωρίσει το σύστημα την ύπαρξη του module.

Δείγμα ενός τέτοιου αρχείου είναι το παρακάτω:

```
name = Really Neat Widget
description = Provides a really neat widget for your site's sidebar.
core = 7.x
package = Views
dependencies[] = views:views
dependencies[] = panels:panels
files[] = tests/example.test
configure = admin/config/content/example
```

Το .info αρχείο θα πρέπει να έχει το ίδιο όνομα με το αρχείο .module και να βρίσκονται στον ίδιο φάκελο. Για παράδειγμα, εάν το .module ονομάζεται example.module, τότε το .info αρχείο πρέπει να ονομαστεί example.info.

Επίσης, περιέχουν σχόλια. Ένα [;] στην αρχή κάθε γραμμής τη μετατρέπει σε σχόλιο και δεν θα αναλυθεί.

Συμπληρώνοντας σωστά το «.info», θα πρέπει να υπάρχουν και τα κατάλληλα χαρακτηριστικά του module, όπως :

- Name (απαραίτητο)
- Description (συνίσταται)
- Core (απαραίτητο)
- Stylesheets (προαιρετικό)
- Scripts (προαιρετικό)
- Files (προαιρετικό)
- Dependencies (προαιρετικό)
- Package (προαιρετικό)
- Php
- Configure
- Required
- Hidden

6.2.1 Ανάλυση του «.info» αρχείου ενός module

✓ **Name(απαραίτητο)**

Εδώ φαίνεται το όνομα του module, που θα εμφανιστεί και στη σελίδα των modules. Εφόσον οι ονομασίες των modules είναι κατά μία έννοια ονόματα, θα πρέπει να αναγράφονται και με τους ανάλογους κανόνες που αρμόζουν σε αυτά. Για παράδειγμα, «Really Neat Widget» είναι το σωστό και όχι «really neat widget».

Name = Really Neat Widget

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

✓ **Description (συνίσταται)**

Μία σύντομη, κατά προτίμηση μίας γραμμής περιγραφή που θα εξηγεί στη σελίδα των modules του administrator τη λειτουργία του συγκεκριμένου module. Το πεδίο είναι περιορισμένο για 255 χαρακτήρες.

Description = Provides a really neat widget for your site's sidebar

Τα descriptions ενδέχεται να περιέχουν και links σε συγγράματα και πηγές. Το παρακάτω παράδειγμα δείχνει ένα link στον συγγραφέα.

Description = Domain manager by <a href=<http://petermoulding.com>> Peter Moulding.com

✓ **Core(απαραίτητο)**

Η έκδοση του Drupal που προορίζεται το module. Για το Drupal 7 είναι 7.x, κτλ.

Core = 7.x

✓ **Stylesheets (προαιρετικό)**

Το Drupal 7 επιτρέπει να προσθέσουμε αρχεία CSS στο «.info» εάν πρόκειται για όλες τις σελίδες, ακριβώς όπως ένα theme.info αρχείο.

Stylesheets[all][] = node.css

✓ **Scripts (προαιρετικό)**

Μπορούμε να προσθέσουμε Javascript.

Scripts[] = somescript.js

✓ **Files (προαιρετικό)**

Το Drupal υποστηρίζει ένα μητρώο για φόρτωση δυναμικού κώδικα. Για την υποστήριξή του όλα τα modules πρέπει να δηλώσουν όλα τα αρχεία κώδικα με δηλώσεις κλάσεων στο .info αρχείο, όπως:

Name = Really Neat Wdget

...

Files[] = tests/example.test

Όταν ένα module ενεργοποιείται, το Drupal θα ξανασκανάρει όλα τα δηλωμένα αρχεία και περιεχόμενα των κλάσεων που βρίσκει. Οι κλάσεις φορτώνονται αυτόματα από την rhr με την πρώτη πρόσβασή τους.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

✓ **Dependencies (προαιρετικό)**

Μια σειρά από άλλα modules που χρειάζεται το συγκεκριμένο module. Αν αυτά τα modules δεν υφίστανται, τότε δε μπορεί να ενεργοποιηθεί. Στην περίπτωση που τα modules υπάρχουν, αλλά δεν είναι ενεργά, ο administrator θα ενημερωθεί με μία λίστα από επιπλέον modules για να ενεργοποιήσει, ανάμεσα στα οποία θα υπάρχει και αυτό που απαιτείται.

`Dependencies[] = {project}:{module}`

Η τιμή που παίρνει πρέπει να είναι το όνομα του project όπως φαίνεται στο url στο drupal.org (drupal.org/project/views), μετά ο χαρακτήρας : και έπειτα το όνομα του module όπως φαίνεται στο .module αρχείο.

✓ **Package (προαιρετικό)**

Εάν το module πρόκειται να χρησιμοποιηθεί αποκλειστικά με άλλα modules, πρέπει να εισάγουμε το όνομα του package εδώ. Εάν μείνει κενό, το module θα θεωρηθεί «Άλλο». Γενικά αυτή η ιδιότητα θα πρέπει να χρησιμοποιείται από πακέτα πολλών modules ή από modules που προορίζονται για επέκταση άλλων πακέτων.

`Package = Views`

✓ **Php (προαιρετικό)**

Από την έκδοση 6.x, τα modules και themes μπορούν να προσδιορίσουν την ελάχιστη php έκδοση που απαιτούν.

`Php = 5.3`

Αυτό σημαίνει ότι η php δεν πρόκειται να λειτουργήσει με προηγούμενη έκδοσή της.

✓ **Configure (προαιρετικό)**

Από την 7.x, το path της κύριας σελίδας διαμόρφωσης του module. Όταν ένα module ενεργοποιείται εμφανίζεται ένα link «configure» και «permissions». Αυτό είναι το path του link «configure» για το συγκεκριμένο module στη σελίδα όλων των modules.

`Configure = admin/config/content/example`

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

✓ **Required (προαιρετικό)**

Τα modules και themes μπορούν να προσδιοριστούν ως απαιτούμενα και δεν πρέπει ποτέ να είναι ανενεργά, απλά προσθέτοντας `required = TRUE`.

Αυτά τα modules ενεργοποιούνται αυτόματα με το install.

✓ **Hidden (προαιρετικό)**

Μπορούμε να προσδιορίσουμε αν δε θέλουμε κάποια modules να είναι ορατά στη σείδα τους, προσθέτοντας `hidden = TRUE`. Αυτό συνήθως συμβαίνει για modules που τα έχουμε δημιουργήσει για πειραματικούς σκοπούς και μόνο.

6.1.2 Ανάλυση και περιγραφή του «.module» αρχείου

Τα modules του Drupal είναι plugins εγκατεστημένα στο site της βάσης του Drupal για την επέκταση ή την ενίσχυση της λειτουργικότητας του προγράμματος του πυρήνα του Drupal.

Σε έναν ιστότοπο του Drupal μπορούν να υπάρχουν τριών ειδών modules:

1. Οι μονάδες του πυρήνα που είναι φορτωμένες με το drupal και αποδεκτές από τους κατασκευαστές του πυρήνα και την κοινότητα του Drupal.
2. Οι συνεισφερόμενες μονάδες, αυτές που έχουν γραφτεί από την κοινότητα και διανέμονται κάτω από την GNU Public Licence (GPL) άδεια καθώς και το Drupal.
3. Οι κατ' απαίτηση μονάδες που κατασκευάζονται από τους δημιουργούς και administrators της εφαρμογής – ιστοσελίδας.

Ένα module αποτελεί μια συλλογή αρχείων που περιέχουν κάποια λειτουργικότητα και είναι γραμμένα σε PHP. Επειδή ο κώδικας του module εκτελείται στο πλαίσιο του ιστότοπου, μπορεί να χρησιμοποιήσει όλες τις λειτουργίες και να έχει πρόσβαση σε όλες τις μεταβλητές και τις δομές του πυρήνα του Drupal. Στην πραγματικότητα, μια μονάδα δεν διαφέρει από ένα κανονικό php αρχείο, το οποίο μπορεί ανεξάρτητα να δημιουργηθεί και να ελεγχθεί και να χρησιμοποιηθεί για πολλαπλές λειτουργίες.

Το σύστημα των modules του Drupal βασίζεται στην ιδέα των «**hooks**». Τα hooks είναι το μέσο με το οποίο τα modules μπορούν να αλληλεπιδράσουν με τον κώδικα του πυρήνα του Drupal. Δίνει τη δυνατότητα να καθορίσει τη URL της web εφαρμογής (μέσω της `hook_menu()`), να προσθέσει περιεχόμενο σε σελίδες

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

(μέσω των `hook_menu()`, `hook_footer()`, κ.λ.π), να δημιουργήσει προσαρμοσμένους πίνακες στη βάση μας (`hook_schema()`) και πολλά άλλα.

Το «hook» αποτελεί μία PHP function, με την ονομασία `foo_bar()`, όπου το «foo» είναι το όνομα του module (`foo.module`) και «bar» είναι το όνομα του hook. Κάθε hook έχει ορισμένες παραμέτρους και ένα συγκεκριμένο τύπο επιστροφής του αποτελέσματος.

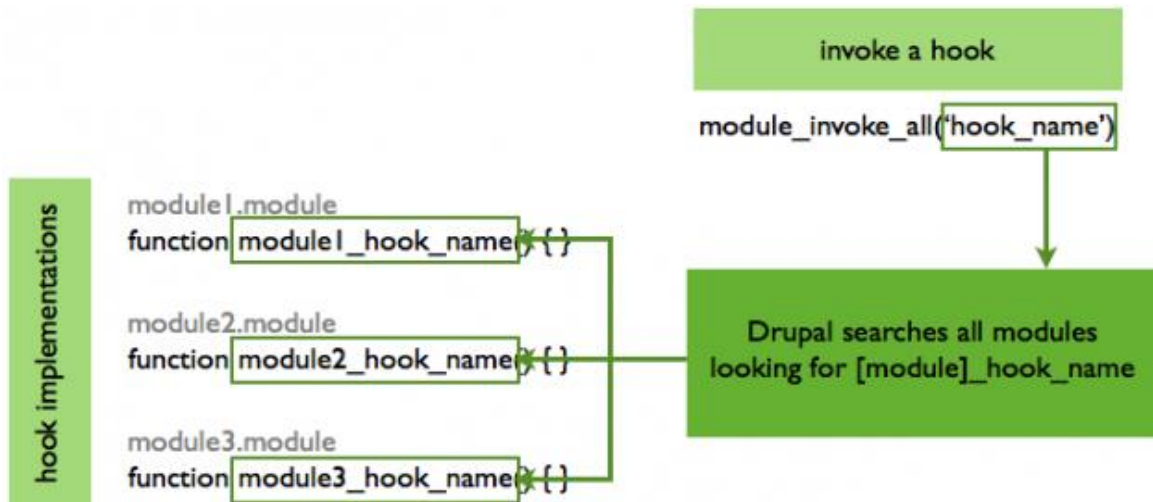


Figure 6-11 - Λειτουργία των hooks

Για την επέκταση του Drupal, χρειάζεται μόνο να συμπεριληφθεί στο module ένα hook. Όταν το Drupal θέλει να επιτρέψει την παρέμβαση από άλλα modules, καθορίζει ποια modules εμπεριέχουν αυτό το hook, και το καλεί.

Ένα **API** (Application Programming Interface) αποτελεί μια διεπαφή μεταξύ των στοιχείων ενός συστήματος λογισμικού. Για κάθε ένα στοιχείο του συστήματός μας ένα API αναλαμβάνει την επικοινωνία μεταξύ στοιχείων και πυρήνα. Με αυτόν τον τρόπο ένα στοιχείο του συστήματος μπορεί να απομονωθεί από τυχόν αλλαγές στον πυρήνα, μειώνοντας έτσι τις δοκιμές και το debugging για ένα στοιχείο και μόνο.

6.2.2 Αρχείο «.info»

Το .info αρχείο θα περιέχει τα παρακάτω χαρακτηριστικά, σύμφωνα με τα οποία ενημερώνει το Drupal με το όνομα του αρχείου .module, καθώς επίσης και την έκδοση του Drupal που χρησιμοποιήσαμε για την υλοποίηση της εργασίας.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Πιο συγκεκριμένα το όνομα που εμφανίζεται στη σελίδα των modules είναι *Our general Module*, έχουμε δώσει μια σύντομη περιγραφή του, την έκδοση και το αρχείο για τη φόρτωση του κώδικα.

```
Name=Our general Module!  
Description=This is our General module!  
core=7.x  
files[] = general_module.module
```

Figure 6-12 - Βασική μορφή του αρχείου «.info»

6.2.3 Βασικά αποσπάσματα κώδικα «.module»

Όπως ήδη αναφέραμε στη *function_hook_menu()*, εγγράφουμε το path για το URL Request. Τα paths εγγράφονται μόνο για τον χειρισμό των URL ή μπορούν να καταχωρήσουν link που θα τοποθετηθεί στο menu (συνήθως Navigation Menu). Επιστρέφει ένα array του οποίου τα κλειδιά ορίζουν τα μονοπάτια (paths) και του οποίου οι τιμές είναι ένα συνδεδεμένο array ιδιοτήτων για κάθε path. Ένα path είναι κοινώς ονομαζόμενο και ως «menu router item».

Στη συγκεκριμένη περίπτωση, για το παρακάτω module το path είναι το «*localhost/general_module*» και έχουμε ορίσει ως τίτλο *Λεξικό Νοηματικής Γλώσσας*.

```
<?php  
Function general_module__menu(){  
  $items=array();  
  $items['general_module']=array(  
    'title'=>t('Λεξικό Νοηματικής Γλώσσας'),  
    'page callback'=>'general_module_form',"  
    'access arguments'=>array('access content'),  
    'description'=> t('Λεξικό Νοηματικής Γλώσσας'),  
    'type'=>MENU_CALLBACK  
  );  
  return $items;  
}
```

Figure 6-13 - Βασική μορφή του hook_menu()

Όταν το path ζητηθεί η page callback function θα πάρει το general_module ως όρισμα.

Δηλώνοντας το πεδίο type ως *Menu_callback*, γίνεται εγγραφή του path έτσι ώστε η σωστή πληροφορία να δημιουργείται μετά την πρόσβαση σε αυτό.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Callback Functions: Ο ορισμός για κάθε path μπορεί να περιλαμβάνει μία *page callback function*, που ανακαλείται όταν ζητηθεί το καταχωρημένο μονοπάτι. Στην περίπτωση που δεν υπάρχει άλλο εγγεγραμμένο path, που να ταιριάζει στο ζητούμενο μονοπάτι, τότε τα επιπλέον στοιχεία του path περνάνε στο callback function.

Callback Arguments: Η *page callback* και άλλες *callback functions* μπορούν να προσδιορίσουν λίστες arguments ως arrays. Αυτές οι λίστες arguments μπορεί να εμπεριέχουν *fixed/hard-coded argument values* (καθορισμένες τιμές των arguments) καθώς επίσης και *integers* που αντιστοιχούν σε *path components* (στοιχεία του μονοπατιού).

```
function general_module_form(){
return drupal_get_form('general_module_my_form');
}
```

Figure 6-14 - Drupal_get_form()

Με την **drupal_get_form()** επιστρέφεται το αποτέλεσμα της function *general_module_menu()*. Ουσιαστικά, ζητήσαμε από το drupal να δημιουργήσει τη σελίδα και το περιεχόμενο αυτής της σελίδας είναι το περιεχόμενο που επιστρέφεται από την function *get_form()*, όταν της περνάμε το argument (όρισμα) 'general_module_my_form'.

Όταν θα έχουμε αποκτήσει πλέον πρόσβαση στο path *localhost/general_module*, εκτελείται η *drupal_get_form ('general_module_my_form')* και θα επιστρέψει, τα αποτελέσματα της form API του παρακάτω αποσπάσματος.

```
function general_module_my_form($form,&$form_state) {
$form['Fieldset1'] = array(
  '#type' => 'fieldset',
  '#title' => t('Επιλογές'),
  '#collapsible' => TRUE,
  '#collapsed' => FALSE,
);
$form['Fieldset1']['submit1'] = array(
  '#type' => 'submit',
  '#value' => 'Αναζήτηση',
  '#submit' => array('general_module_my_form_submit'),
}
}
```

Figure 6-15 - Εισαγωγή πεδίων fieldset και submit buttons

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Με την παραπάνω function προσθέτω στοιχεία της φόρμας και πεδία, όπως τα fieldset box και submit buttons, για την εισαγωγή των απαραίτητων λέξεων στη βάση του λεξικού μας και αντίστοιχα στην αναζήτησή τους.

Με τις functions που αναφέρονται παρακάτω, εκτελούμε path redirect. Μας ανακατευθύνει, δηλαδή, σε νέες σελίδες, όπου καλούμε άλλες συναρτήσεις (όπως θα δούμε παρακάτω σε άλλο module αρχείο), όσον αφορά την αναζήτηση στη βάση μας, την εισαγωγή νέας λέξης και νέας κατηγορίας.

```
function general_module_my_form_submit($form, &$form_state) {
  $form_state['redirect'] = 'functions_module/search';
}
```

Figure 6-16 - Ανακατεύθυνση σελίδων στα modules

Εδώ περιγράφονται αναλυτικά όλες οι functions που πρέπει να καλεστούν καθώς επίσης και όλες οι νέες σελίδες που έχουμε κάνει redirect από την home page που βλέπουμε στο general_module.module, των προηγούμενων αποσπασμάτων.

Το **hook_menu()**, το οποίο όπως αναφέραμε και σε προηγούμενη ενότητα εγγράφει το path για να ορίσει το URL Request.

Με το παρακάτω **MENU_CALLBACK()** ανακατευθυνόμαστε στο path και 'functions_module/search', όπως ζητήθηκε προηγουμένως με το αντίστοιχο path_redirect του Drupal, στη συγκεκριμένη περίπτωση κάνουμε αναζήτηση λέξης.

```
$items['functions_module/search'] = array(
  'title' => 'Search!',
  'description' => 'Search',
  'page callback' => 'drupal_get_form',
  'page arguments' => array('search'),
  'access callback' => TRUE,
  'type' => MENU_CALLBACK,
);
$items['functions_module/search_by_word'] = array(
  'title' => 'Search_by_word!',
  'description' => 'Search_by_word',
  'page callback' => 'drupal_get_form',
  'page arguments' => array('search_by_word'),
  'access callback' => TRUE,
  'type' => MENU_CALLBACK,
);
```

Figure 6-17 - Λειτουργία αναζήτησης

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Στο απόσπασμα της παρακάτω εικόνας βλέπουμε τη σύνδεση λέξης – video κατά τον εντοπισμό και η επιθυμητή διαγραφή τους . Επίσης, ορίζουμε τη θέση του video , το μέγεθος και την επέκταση που θα έχει το αρχείο.

```
$items['link_delete'] = array(
  'title' => t('Delete a word!'),
  'page callback' => 'drupal_get_form',
  'page arguments' => array('link_delete'),
  'type' => MENU_CALLBACK,
  'expanded' => TRUE,
  'access callback' => TRUE,
);
$items['link_video/%'] = array(
  'title' => 'Video page',
  'page callback' => 'link_video',
  'page arguments' => array(1),
  'access arguments' => array('access content'),
);
return $items;}
function link_video($id){
$template= '
<div><video width="320" height="240" controls autoplay type="video/mp4"
">
<source src="%s" type="video/mp4" >
Your browser does not support the video tag.
</video></div>';
$query = db_select('Videos', 'd');
$query->fields('d', array('filename'));
$query->condition('Words_word_id',$id, '=');
$result = $query
->execute();
$html = "";
while($record = $result->fetchAssoc()) {
  $file_uri = file_create_url(file_build_uri('videos/'.$record['filename']));
  $html .= sprintf($template,$file_uri);
}
if($html == "") return "no video";
return $html;
} function link_delete($form,&$form_state) {
  $form['first'] = array(
    '#type' => 'textfield',
    '#title' => t('Delete a word!'),
    '#required' => TRUE,
    '#default_value' => "Please enter the word you want to delete.",
    '#description' => "Please enter the word you want to delete.",
    '#size' => 60,
    '#maxlength' => 20,
  );
  $form['submit_del'] = array(
    '#type' => 'submit',
    '#value' => 'Delete',
    '#submit' => array('delete'),
  );
  return $form; }
```

Figure 6-18 - Σύνδεση λέξης - video προς διαγραφή

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Η παρακάτω function ευθύνεται για τη διαγραφή των videos και των λέξεων που θέλουμε να διαγραφούν από τη βάση, αφού πρώτα εντοπίσουμε τη λέξη μας. Στο τέλος εκτελείται και έλεγχος για τα 3 queries.

```
function delete($form,&$form_state) {
$query = db_select('Words', 'd');
$query->fields('d', array('word_id'))
->condition('word',$form_state['values']['first'],'=');
$result = $query
->execute();
$html="";
while($record = $result->fetchCol()) {
$html=$record;
}
$query2 = db_delete('Words')
->condition('word_id',$html,'=')
->execute();
$query3 = db_delete('Videos')
->condition('Words_word_id',$html,'=')
->execute();
$query4 = db_delete('W_V')
->condition('Word_id',$html,'=')
->execute();
if($query2&&$query3&&$query4){
drupal_set_message(print_r('The word ' . "".$form_state['values']['first']
. "".'that you requested has been deleted.', true));
}else{
drupal_set_message(print_r('The word that you requested does not exist in
our database.', true));
}
}
```

Figure 6-19 - Τελική διαγραφή λέξης - video

Στο επόμενο απόσπασμα κώδικα, φτιάχνουμε τη φόρμα όσον αφορά την εισαγωγή λέξης στο λεξικό. Πιο συγκεκριμένα, φτιάχνουμε ένα πεδίο με όνομα «New Word», το οποίο περιέχει α) ένα νέο πεδίο με αντίστοιχο όνομα για την εισαγωγή της λέξης μας, β) μία περιοχή στην οποία θα περιγράψουμε τη λέξη μας, γ) ένα πεδίο τύπου managed_field, για να φορτώσουμε το βίντεό μας και δ) το submit button για την προσθήκη λέξης.

```
function new_word() {
    $form['New_word'] = array(
        '#type' => 'fieldset',
        '#title' => t('New Word!!!!!!'),
        '#collapsible' => TRUE,
        '#collapsed' => FALSE,
    );
    $form['New_word']['Word'] = array(
        '#type' => 'textfield',
        '#title' => t('New word'),
        '#required' => TRUE,
        '#default_value' => "Please enter a new word.",
        '#description' => "Please enter a new word.",
        '#size' => 60,
        '#maxlength' => 20,
    );
    $form['Description'] = array(
        '#type' => 'textarea',
        '#title' => t('Description of the new word.'),
        '#default_value' => "Please enter the description of the new word.");
}
```

Figure 6-20 - Φτιάχνοντας τη μορφή για το πεδίο εισαγωγής νέας λέξης στο λεξικό

```
$form['file'] = array(
    '#type' => 'file',
    '#title' => t('Upload the video of the word!!!'),
    '#description' => t('Select the upload video to be imported. '),
);
$form['submit'] = array(
    '#type' => 'submit',
    '#value' => 'Add',
    '#submit' => array('my_form_button'),
);
$form['saved_value_1'] = array(
    '#type' => 'value',
    '#value' => $html,
);
$form['saved_value_2'] = array(
    '#type' => 'value',
    '#value' => $html2,
);
return $form;
}
```

Figure 6-21 - Video uploading για την αντίστοιχη λέξη

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Στο απόσπασμα αυτό εφαρμόζουμε τον κατάλληλο κώδικα για το κομμάτι της αναζήτησης και εφαρμόζουμε την επέκταση της κλάσης **PagerDefault()**. Είναι ο μηχανισμός pager και από default έχει 10 σειρές ανά σελίδα. Ουσιαστικά, πρόκειται για μία μέθοδο, με την οποία, λαμβάνοντας ως δεδομένο το αντικείμενο ενός query, π.χ. **\$query = db_select('Words', 'd')**, επιστρέφει ένα νέο αντικείμενο, που θα χρησιμοποιηθεί στη θέση του. Έτσι, λοιπόν, δημιουργείται ένα νέο PagerDefault query object που περιλαμβάνει το αρχικό μας select query.

```
function search_all_1($form, &$form_state) {
  $form['words_all'] = array(
    '#type' => 'submit',
    '#value' => 'All the words',
    '#submit' => array('words_all'),
  );
  return $form;
}
function search_all_2($form, &$form_state) {
  $header = array(
    array('data' => t('ID')),
    array('data' => t('Word')),
    array('data' => t('Description')),
    array('data' => t('Edit')),
    array('data' => t('Delete')),
  );
  $query = db_select('Words', 'd')->extend('PagerDefault');
  $query->fields('d', array('word_id', 'word', 'description'));
  $result = $query
    ->orderBy('d.word_id')
    ->execute();
}
```

Figure 6-22 - Μηχανισμός PagerDefault() για τα αποτελέσματα της αναζήτησης

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

Έπειτα φτιάχνουμε εμείς τη δομή των σειρών, όπως θέλουμε να εμφανίζονται οι λέξεις μας.

```
$rows = array();
foreach ($result as $row) {
$rows[] = array(
    $row->word_id,
    l($row->word,'link_video/'.$row->word_id),
    $row->description,
    l('edit','link_edit'),
    l('delete','link_delete')
);
}
$build['Words Region'] = array(
    '#type' => 'markup',
    '#markup' => t('All words in our database are below :'),
);
$build['pager_table'] = array(
    '#theme' => 'table',
    '#header' => $header,
    '#rows' => $rows,
    '#empty' => t('There are no WORDS found in the db'),
);
$build['pager_pager'] = array('#theme' => 'pager');
return $build;
}
```

Figure 6-23 - Τα αποτελέσματα της αναζήτησης με σωστή δόμηση

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

6.3 Τελική μορφή του Λεξικού μας

1. Αρχική σελίδα, όπου επιλέγουμε αν θέλουμε να κάνουμε αναζήτηση ή να εισάγουμε νέα λέξη ή νέα κατηγορία.

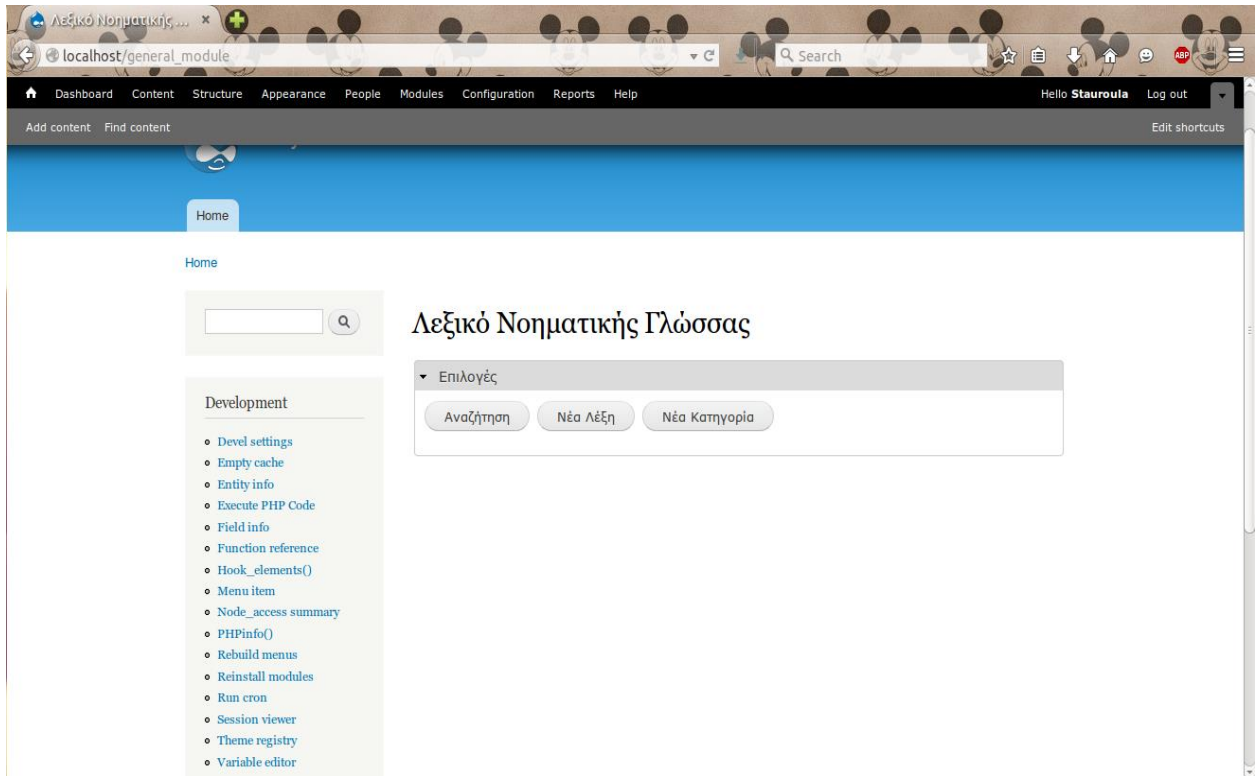


Figure 6-24 - Home Page της διαδικτυακής εφαρμογής

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

2. Αφού επιλέξουμε νέα λέξη μας ανακατευθύνει σε αυτή τη σελίδα όπου κάνουμε εισαγωγή της λέξης μας, κάνουμε μια σύντομη περιγραφή και ανεβάζουμε το αντίστοιχο βίντεο στη νοηματική γλώσσα.

Add a new word in our application!!

▽ New Word!!!!

New word *

Αλεπού
Please enter a new word.

Description of the new word.

Αλεπού

Choose the category!

Ζώα

Upload the video of the word!!!

Choose File alepou.mp4
Select the upload video to be imported.

Add

Figure 6-25 - Εισαγωγή νέας λέξης

3. Αντίστοιχα και την περίπτωση για εισαγωγή νέας κατηγορίας.

▽ New Category

New Category *

Δέντρα
Please enter new category.

Description of the new category.

Δέντρα

Add

Figure 6-26 - Εισαγωγή νέας κατηγορίας

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

- Όταν στην αρχική σελίδα επιλέξουμε την *Αναζήτηση* μας ανακατευθύνει στην παρακάτω σελίδα, όπου και επιλέγουμε το είδος της αναζήτησης που θέλουμε να κάνουμε.

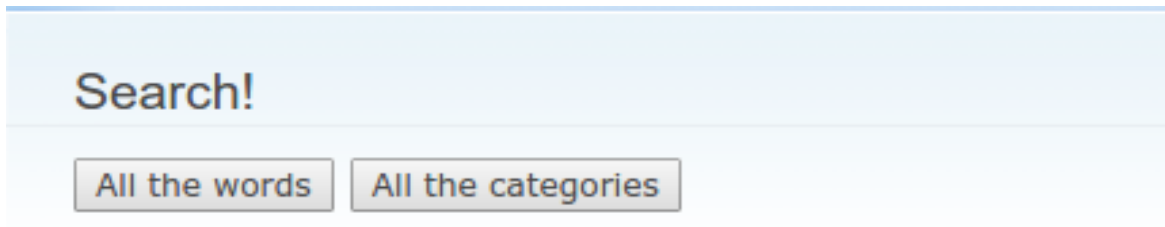


Figure 6-27 - Αναζήτηση όλων των λέξεων ή κατηγοριών αντίστοιχα

- Όταν επιλέγουμε να μας δείξει όλες τις κατηγορίες ή όλες τις λέξεις που υπάρχουν στη βάση μας, έχουμε τα παρακάτω αποτελέσματα αντίστοιχα:

All categories in our database are below :

ID	Category	Description
14	Ανθρώπινο Σώμα	Μέρη του ανθρώπινου Σώματος
15	Ποτά/Αναψυκτικά	Ποτά/Αναψυκτικά

Figure 6-28 - Αποτελέσματα αναζήτησης όλων των κατηγοριών

All words in our database are below :

ID	Word	Description	Edit	Delete
243	Ρινική Κοιλότητα	Ρινική Κοιλότητα	edit	delete
244	Στομάχι	Στομάχι	edit	delete
245	Οισοφάγος	Οισοφάγος	edit	delete
246	Αλάτι	Αλάτι	edit	delete
247	Αναψυκτικό	Αναψυκτικό	edit	delete
248	Γάλα	Γάλα	edit	delete
249	Καφές	Καφές	edit	delete
250	Στόμα	Στόμα	edit	delete

Figure 6-29 - Αποτελέσματα αναζήτησης όλων των λέξεων

6. Όσον αφορά την αναζήτηση ανά κατηγορία επιλέγουμε την αντίστοιχη λειτουργία και έπειτα παρουσιάζονται τα αποτελέσματα από τη βάση μας.

Home

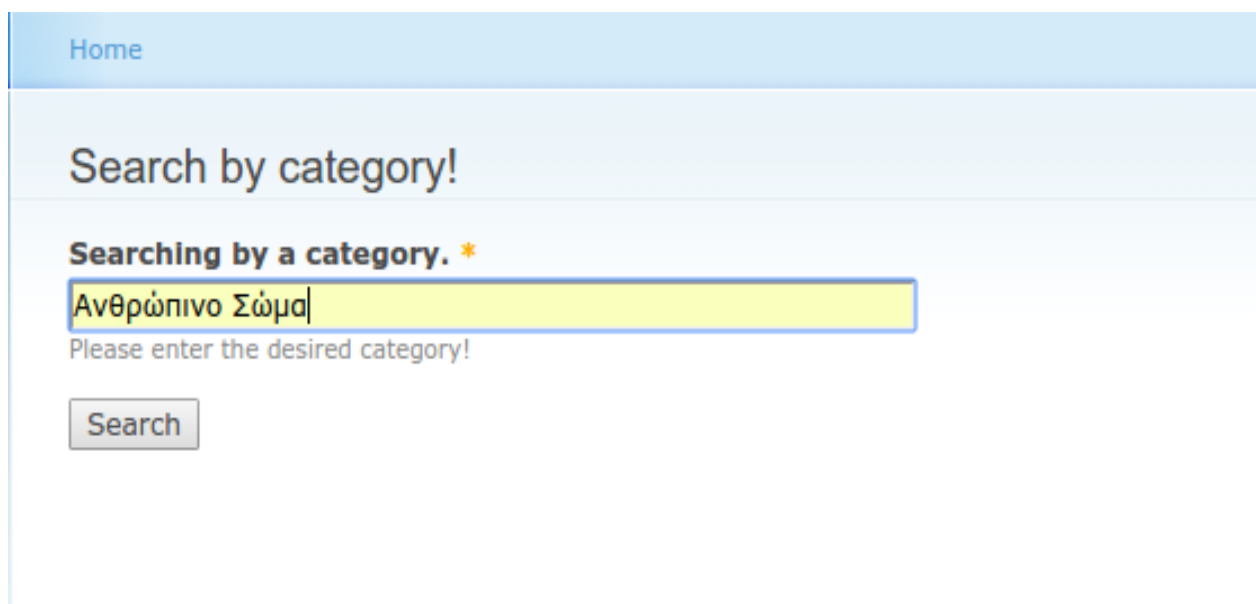
Search by category!

Searching by a category. *

Please enter the desired category!

Please enter the desired category!

Figure 6-30 - Αναζήτηση ανάλογα με την επιθυμητή κατηγορία



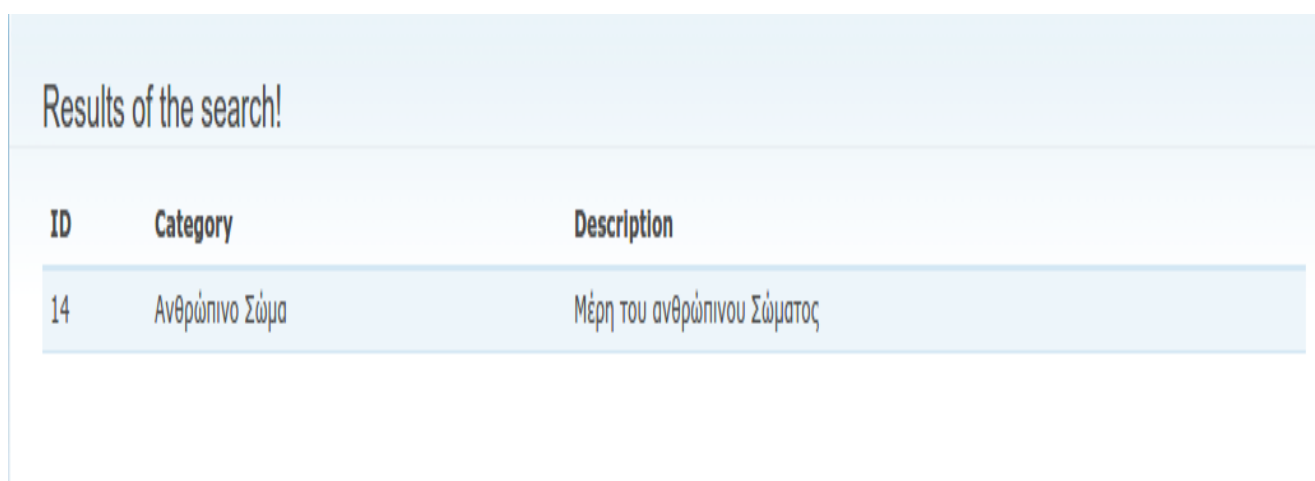
Home

Search by category!

Searching by a category. *

Please enter the desired category!

Figure 6-31 - Εκτέλεση αναζήτησης ανά κατηγορία



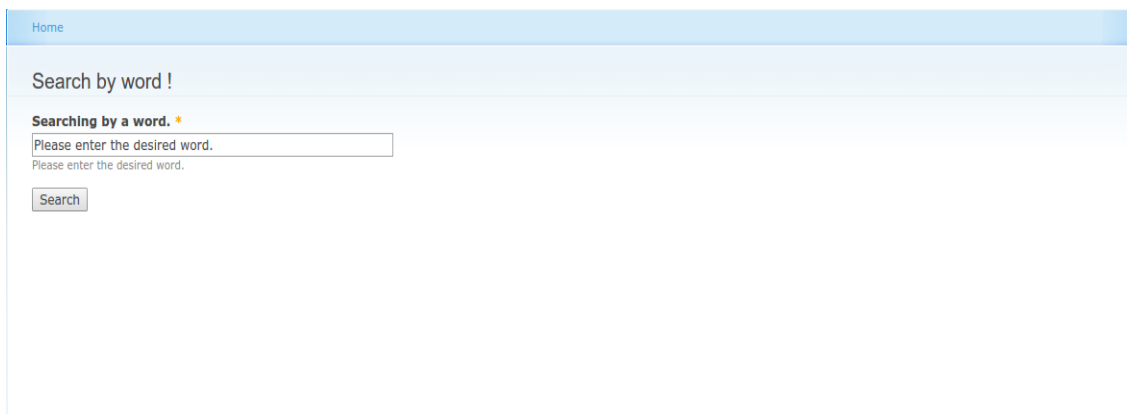
Results of the search!

ID	Category	Description
14	Ανθρώπινο Σώμα	Μέρη του ανθρώπινου Σώματος

Figure 6-32 - Αποτέλεσμα αναζήτησης κατηγορίας

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

7. Αντίστοιχα η διαδικασία για την αναζήτηση ανά λέξη:



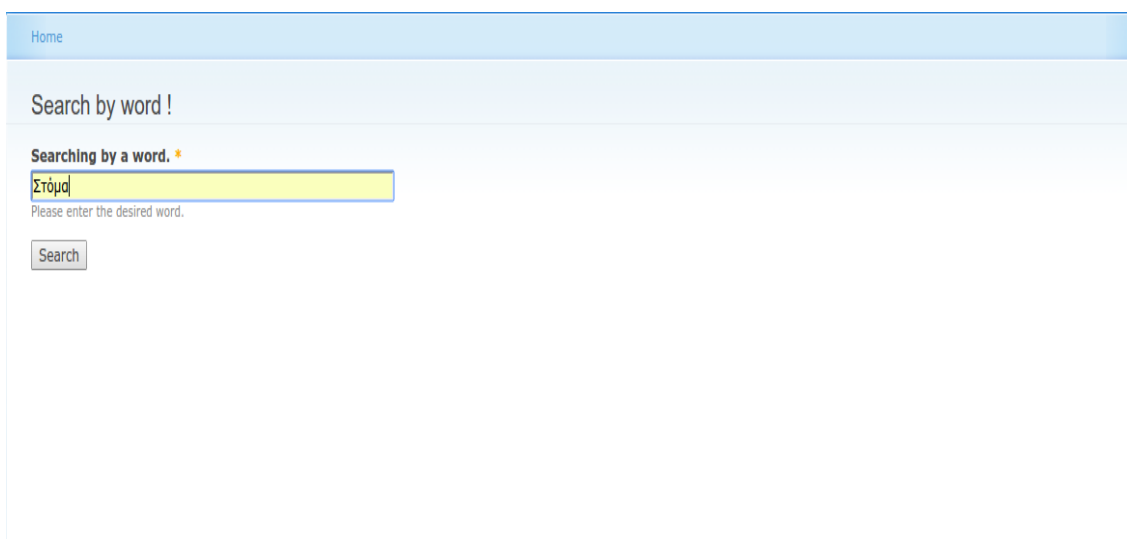
Home

Search by word !

Searching by a word. *
Please enter the desired word.

Search

Figure 6-33 - Αναζήτηση επιθυμητής λέξης



Home

Search by word !

Searching by a word. *
Please enter the desired word.

Search

Figure 6-34 - Εκτέλεση αναζήτησης της λέξης μας

Results of the search!

ID	Word	Description	Delete
244	Στόμαχι	Στομάχι	delete

Figure 6-35 - Εμφάνιση της λέξης μας

8. Τέλος, μας δίνεται η δυνατότητα να διαγράψουμε λέξη από τη βάση:

Home

Delete a word!

Delete a word! *

Please enter the word you want to delete.

Figure 6-36 - Διαγραφή λέξης από τη βάση

Home

Delete a word!

✔ The word 'Στόμαχι' that you requested has been deleted.

Delete a word! *

Please enter the word you want to delete.

Figure 6-37 - Επιτυχής λειτουργία της διαγραφής

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

ΚΕΦΑΛΑΙΟ 7

7. Σύνοψη και Συμπεράσματα

7.1 Αποτελέσματα

Με την υλοποίηση αυτής της πτυχιακής εργασίας μας δόθηκε η ευκαιρία να αποκομίσουμε γνώσεις αλλά και εμπειρία πάνω στο αντικείμενο την ανάπτυξης ιστοσελίδων. Πραγματοποιήσαμε έρευνα γύρω από το αντικείμενο της ανάπτυξης ιστοσελίδων και οτιδήποτε άλλο αυτό συμπεριλαμβάνει. Μάθαμε για το Open Source και γίναμε μέλη αυτής της τεράστιας κοινότητας. Γνωρίσαμε το Drupal και πώς να φτιάχνουμε ιστοσελίδες με αυτό το πολυδιάστατο εργαλείο. Αφιερώσαμε αρκετό χρόνο στην αναζήτηση modules, ελέγξαμε στην θεωρία και στην πράξη την λειτουργικότητα τους και καταλάβαμε πόσο σημαντικά είναι για την κατασκευή μίας ιστοσελίδας με χρήση του Drupal. Επίσης αποκομίσουμε γνώσεις πάνω στις γλώσσες προγραμματισμού που χρησιμοποιεί το Drupal και να επεξεργαστούμε κώδικα αυτών των γλωσσών.

Εκτός της τεχνικής πλευράς, με την πτυχιακή αυτή εργασία, εμπλουτίζουμε τους ορίζοντές μας και καταλαβαίνουμε τη σημασία της γλωσσολογίας. Κάθε άνθρωπος, κάθε λαός και κάθε τόπος έχει τα δικά του στοιχεία, τα οποία έχει αναπτύξει σε βάθος χρόνου σύμφωνα με τις ανάγκες που δημιουργούνταν. Έτσι λοιπόν, δημιουργώντας αυτό το λεξικό, μπορούμε να μάθουμε άλλη μία γλώσσα, όχι κατώτερη από άλλες, αλλά μία εξίσου σημαντική, για μία αποτελεσματική επικοινωνία.

7.2 Συμπεράσματα

Το Drupal είναι ένα ευρέως διαδεδομένο CMS στο διαδίκτυο και πάρα πολλές ιστοσελίδες έχουν δημιουργηθεί με την χρήση αυτού. Οι δυνατότητες του είναι σχεδόν απεριόριστες και μπορεί να χρησιμοποιηθεί για την κατασκευή ιστοσελίδων πολλών και διαφορετικών θεματικών περιεχομένων.

Πολύ βασικό πλεονέκτημα του Drupal είναι η επεκτασιμότητα του μέσω των modules. Υπάρχει μία μεγάλη ποικιλία σε modules που προσφέρουν επιπλέον

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

δυνατότητες και λειτουργικότητα και μπορούν να καλύψουν σχεδόν όλες τις ανάγκες λειτουργικότητας.

Επίσης, έχει δημιουργηθεί μία μεγάλη και διαρκώς αναπτυσσόμενη κοινότητα γύρω από το Drupal που απαρτίζεται από απλούς χρήστες αρχάριους, από προγραμματιστές και από ολόκληρες εταιρείες. Χάρη της νοοτροπίας αυτής της Open Source κοινότητας, οι πληροφορίες και οι γνώσεις πάνω στο συγκεκριμένο αντικείμενο μοιράζονται απλόχερα , δίνοντάς μας την ευκαιρία να ασχοληθούμε πιο πολύ εις βάθος και να έχουμε παράλληλα την βοήθεια και την υποστήριξη αυτής της κοινότητας.

ΚΕΦΑΛΑΙΟ 8

8. Πηγαίος κώδικας

8.1 Αρχείο «general_module.info»

```
name = Our general Module!  
description = This is our GENERAL module!  
core = 7.x  
files[] = general_module.module
```

8.1.1 Αρχείο «general_module.module»

```
<?php  
function general_module_menu() {  
  $items = array();  
  $items['general_module'] = array(  
    'title' => t('Λεξικό Νοηματικής Γλώσσας'),  
    'page callback' => 'general_module_form',  
    'access arguments' => array('access content'),  
    'description' => t('Λεξικό Νοηματικής Γλώσσας'),  
    'type' => MENU_CALLBACK,  
  );  
  return $items;  
}  
function general_module_form() {  
  return drupal_get_form('general_module_my_form');  
}  
function general_module_my_form($form,&$form_state) {  
  $form['Fieldset1'] = array(  
    '#type' => 'fieldset',  
    '#title' => t('Επιλογές'),  
    '#collapsible' => TRUE,  
    '#collapsed' => FALSE,  
  );  
  $form['Fieldset1']['submit1'] = array(  
    '#type' => 'submit',  
    '#value' => 'Αναζήτηση',  
    '#submit' => array('general_module_my_form_submit'),  
  );  
  $form['Fieldset1']['submit2'] = array(  
    '#type' => 'submit',  
    '#value' => 'Νέα Λέξη',  
    '#submit' => array('general_module_my_form_button'),  
  );  
  $form['Fieldset1']['submit3'] = array(  
    '#type' => 'submit',
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
'#value' => 'Νέα Κατηγορία',
'#submit' => array('general_module_my_form_button3'),
);
return $form;
}

function general_module_my_form_submit($form, &$form_state) {
$form_state['redirect'] = 'functions_module/search';
}
function general_module_my_form_button($form, &$form_state) {
$form_state['redirect'] = 'functions_module/new_word';
}
function general_module_my_form_button3($form, &$form_state) {
$form_state['redirect'] = 'functions_module/new_category';
}
?>
```

8.2 Αρχείο «functions_module.info»

```
name=The functions of our site!
description=All the functions that we need are here!
core = 7.x
files[] = functions_module.module
```

8.2.1 Αρχείο «functions_module.module»

```
<?php
function functions_module_menu() {
  $items = array();

  $items['functions_module'] = array(
    'title' => t('Our Functions'),
    'page callback' => 'drupal_get_form',
    'page arguments' => array('functions_module_my_form'),
    'description' => t('Our Functions'),
    'type' => MENU_CALLBACK,
    'expanded' => TRUE,
  );
  $items['functions_module/new_word'] = array(
    'title' => 'Add a new word in our application!!',
    'description' => 'Add a new word in our application!',
    'page callback' => 'drupal_get_form',
    'page arguments' => array('new_word',1),
    'access callback' => TRUE,
    'type' => MENU_CALLBACK,
  );
  $items['functions_module/new_category'] = array(
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
'title' => 'Add a new category in our application!',
'description' => 'Add a new category in our application!',
'page callback' => 'drupal_get_form',
'page arguments' => array('new_category'),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
);
$items['functions_module/search_all_1'] = array(
'title' => 'Search!',
'description' => 'Search',
'page callback' => 'drupal_get_form',
'page arguments' => array('search_all_1'),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
'expanded' => TRUE,
);
$items['functions_module/search_all_2'] = array(
'description' => 'Search',
'page callback' => 'drupal_get_form',
'page arguments' => array('search_all_2'),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
'expanded' => TRUE,
);
$items['functions_module/search_all_3'] = array(
'description' => 'Search',
'page callback' => 'drupal_get_form',
'page arguments' => array('search_all_3'),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
'expanded' => TRUE,
);
$items['functions_module/search'] = array(
'title' => 'Search!',
'description' => 'Search',
'page callback' => 'drupal_get_form',
'page arguments' => array('search'),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
);
$items['functions_module/search_by_word'] = array(
'title' => 'Search_by_word!',
'description' => 'Search_by_word',
'page callback' => 'drupal_get_form',
'page arguments' => array('search_by_word'),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
);
```

```
$items['functions_module/search_by_word_1'] = array(  
  'title' => 'Search by word !',  
  'description' => 'Search_by_word...',  
  'page callback' => 'drupal_get_form',  
  'page arguments' => array('search_by_word_1'),  
  'access callback' => TRUE,  
  'type' => MENU_CALLBACK,  
);
```

```
$items['functions_module/search_by_word_3'] = array(  
  'title' => 'Results of the search!',  
  'description' => 'Search_by_word2...',  
  'page callback' => 'drupal_get_form',  
  'page arguments' => array('search_by_word_3'),  
  'access callback' => TRUE,  
  'type' => MENU_CALLBACK,  
);
```

```
$items['functions_module/search_by_word_3/%'] = array(  
  'title' => 'Results of the search!',  
  'description' => 'Search_by_word2...',  
  'page callback' => 'search_by_word_3',  
  'page arguments' => array(2),  
  'access callback' => TRUE,  
  'type' => MENU_CALLBACK,  
);
```

```
$items['functions_module/search_by_category'] = array(  
  'title' => 'Search_by_category!',  
  'description' => 'Search_by_category',  
  'page callback' => 'drupal_get_form',  
  'page arguments' => array('search_by_category'),  
  'access callback' => TRUE,  
  'type' => MENU_CALLBACK,  
);
```

```
$items['functions_module/search_by_category_1'] = array(  
  'title' => 'Search by category!',  
  'description' => 'Search_by_category...!',  
  'page callback' => 'drupal_get_form',  
  'page arguments' => array('search_by_category_1'),  
  'access callback' => TRUE,  
  'type' => MENU_CALLBACK,  
);
```

```
$items['functions_module/search_by_category_3'] = array(  
  'title' => 'Search_by_category2...!',
```


Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
'description' => 'Search_by_category2...',
'page callback' => 'drupal_get_form',
'page arguments' => array('search_by_category_3'),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
);

$items['functions_module/search_by_category_3/%'] = array(
'title' => 'Results of the search!',
'description' => 'Search_by_cat...',
'page callback' => 'search_by_category_3',
'page arguments' => array(2),
'access callback' => TRUE,
'type' => MENU_CALLBACK,
);

$items['link_delete'] = array(
'title' => t('Delete a word!'),
'page callback' => 'drupal_get_form',
'page arguments' => array('link_delete'),
'type' => MENU_CALLBACK,
'expanded' => TRUE,
'access callback' => TRUE,
);
$items['link_video/%'] = array(
'title' => 'Video page',
'page callback' => 'link_video',
'page arguments' => array(1),
'access arguments' => array('access content'),
);
return $items;
}

function link_video($id){
$template= '
<div><video width="320" height="240" controls autoplay type="video/mp4 " >
<source src="%s" type="video/mp4" >

Your browser does not support the video tag.
</video></div>';
//$form['word_1']
//Query 1
$query = db_select('Videos', 'd');
$query->fields('d', array('filename'));
$query->condition('Words_word_id',$id, '=');
$result = $query
->execute();

$html = "";
while($record = $result->fetchAssoc()) {
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
        $file_uri = file_create_url(file_build_uri('videos/'.$record['filename']));
        $html .= sprintf($template,$file_uri);
    }
    if($html == "") return "no video";
    return $html;
    //$results = db_insert('Words')
    // -> fields(array('word', 'description'))
    // ->values(array(
    // 'word' => $result,
    //->execute());
}
function link_delete($form,&$form_state) {
    $form['first'] = array(
        '#type' => 'textfield',
        '#title' => t('Delete a word!'),
        '#required' => TRUE,
        '#default_value' => "Please enter the word you want to delete.",
        '#description' => "Please enter the word you want to delete.",
        '#size' => 60,
        '#maxlength' => 20,
    );
    $form['submit_del'] = array(
        '#type' => 'submit',
        '#value' => 'Delete',
        '#submit' => array('delete'),
    );
    return $form;
}

function delete($form,&$form_state) {
    //Query για το id της λεξης που πληκτρολογησα!
    $query = db_select('Words', 'd');
    $query->fields('d', array('word_id'))
    ->condition('word',$form_state['values']['first'],'=');
    $result = $query
    ->execute();
    $html="";
    while($record = $result->fetchCol()) {
        $html=$record;
    }
    //Query για το σβησιμο της λεξης που πληκτρολογησα απο τον πινακα WORDS!
    $query2 = db_delete('Words')
    ->condition('word_id',$html,'=')
    ->execute();

    //Query για το σβησιμο της εγγραφης απο τον πινακα VIDEOS!
    $query3 = db_delete('Videos')
    ->condition('Words_word_id',$html,'=')
    ->execute();
}
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
//Query για το σβησιμο της εγγραφης απο τον πινακα VIDEOS!
```

```
$query4 = db_delete('W_V')  
->condition('Word_id',$html, '=')  
->execute();
```

```
//Ελεγχος και των 3 Query για την σωστη εκτελεση τους!
```

```
if($query2&&$query3&&$query4){  
  drupal_set_message(print_r('The word ' . "" . $form_state['values']['first'] . "" . 'that you  
  requested has been deleted.', true));  
}else{  
  drupal_set_message(print_r('The word that you requested does not exist in our  
  database.', true));  
}  
}
```

```
function new_word() {  
  $form['New_word'] = array(  
    '#type' => 'fieldset',  
    '#title' => t('New Word!!!!!!'),  
    '#collapsible' => TRUE,  
    '#collapsed' => FALSE,  
  );  
  $form['New_word']['Word'] = array(  
    '#type' => 'textfield',  
    '#title' => t('New word'),  
    '#required' => TRUE,  
    '#default_value' => "Please enter a new word.",  
    '#description' => "Please enter a new word.",  
    '#size' => 60,  
    '#maxlength' => 20,  
  );  
  $form['Description'] = array(  
    '#type' => 'textarea',  
    '#title' => t('Description of the new word.'),  
    '#default_value' => "Please enter the description of the new word.",  
  );  
  $query = db_select('Categories', 'd');  
  $query->fields('d', array('category'));  
  $result = $query  
    ->execute();  
  
  $query2 = db_select('Categories', 'd');  
  $query2->fields('d', array('category_id'));  
  $result2 = $query2  
    ->execute();  
  
  while($record = $result->fetchCol()) {  
    $html=$record;  
  }  
}
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
while($record2 = $result2->fetchCol()) {
    $html2=$record2;
}
//drupal_set_message(print_r($html, true));
//drupal_set_message(print_r($html[0], true));

$form['category_s'] = array(
    '#title' => t("Choose the category!"),
    '#type' => 'select',
    '#options' => $html,
);
$form['file'] = array(
    '#type' => 'file',
    '#title' => t('Upload the video of the word!!!'),
    '#description' => t('Select the upload video to be imported. '),
);
$form['submit'] = array(
    '#type' => 'submit',
    '#value' => 'Add',
    '#submit' => array('my_form_button'),
);
$form['saved_value_1'] = array(
    '#type' => 'value',
    '#value' => $html,
);
$form['saved_value_2'] = array(
    '#type' => 'value',
    '#value' => $html2,
);
return $form;
}
```

```
function new_category() {
    $form['category'] = array(
        '#type' => 'fieldset',
        '#title' => t('New Category'),
        '#collapsible' => TRUE,
        '#collapsed' => FALSE,
    );
    $form['category']['Category'] = array(
        '#type' => 'textfield',
        '#title' => t('New Category'),
        '#required' => TRUE,
        '#default_value' => "Please enter a new category",
        '#description' => "Please enter new category.",
        '#size' => 60,
        '#maxlength' => 20,
    );
    $form['Description'] = array(
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
'#type' => 'textarea',
'#title' => t('Description of the new category.'),
'#default_value' => "Please enter the description of the new category.",
);
$form['submit'] = array(
  '#type' => 'submit',
  '#value' => 'Add',
  '#submit' => array('add_category'),
);
return $form;
}
```

```
function my_form_button(&$form, &$form_state) {
  $saved_value_1 = $form_state['values']['saved_value_1'];
  $saved_value_2 = $form_state['values']['saved_value_2'];
```

```
//Insert the new word in our database!
$result = db_insert('Words')
  -> fields(array('word', 'description'))
  -> values(array(
    'word' => $form_state['values']['Word'],
    'description' => $form_state['values']['Description'],))
  -> execute();
```

```
//Insert in our database the category of our new word !
$result2 = db_insert('C_W')
  -> fields(array('Category_id', 'Word_id'))
  -> values(array(
    'Category_id' => $saved_value_2[$form_state['values']['category_s']],
    'Word_id' => $result,))
  -> execute();
```

```
$file = file_save_upload('file', array( 'file_validate_extensions' => array('mp3 flv
mp4 jpg png mov'), 'file_validate_size' => array(1024*1280*800,)),);
```

```
// If the file passed validation:
if ($file) {
  // Move the file into the Drupal file system.
  if ($file = file_move($file, 'public://videos')) {
    // Save the file for use in the submit handler.
    $form_state['storage']['file'] = $file;
    drupal_set_message(t("The uploaded file was written to the site's file
folder."));
    $result3 = db_insert('Videos')
  -> fields(array( 'filename', 'description','Words_word_id'))
  -> values(array(
    'filename' => $file->filename,
    'description' => $form_state['values']['Description'],
    'Words_word_id' =>$result,))
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
->execute();
}
else {
    form_set_error('file', t("Failed to write the uploaded file to the site's file
folder."));
}
}
else {
    form_set_error('file', t('No file was uploaded.'));
}
$result4 = db_insert('W_V')
-> fields(array( 'Word_id', 'Video_id'))
-> values(array(
'Word_id' => $result,
'Video_id' => $result3,))
->execute();
}

function add_category($form, &$form_state) {
$result5 = db_insert('Categories')
-> fields(array(
    'category'=> $form_state['values']['Category'],
    'description' => $form_state['values']['Description'],
))
->execute();
drupal_set_message(t('Your new category has been added successfully!!'));
}

$form['search_by_word'] = array(
    '#type' => 'submit',
    '#value' => 'Search by Word',
    '#submit' => array('search_by_word'),
);
$form['search_by_category'] = array(
    '#type' => 'submit',
    '#value' => 'Search by Category',
    '#submit' => array('search_by_category'),
);
$form['search_all'] = array(
    '#type' => 'submit',
    '#value' => 'Search All',
    '#submit' => array('search_all'),
);
return $form;
}

function search_all($form, &$form_state) {
    $form_state['redirect'] = 'functions_module/search_all_1';
}
}
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
function search_all_1($form, &$form_state) {
    $form['words_all'] = array(
        '#type' => 'submit',
        '#value' => 'All the words',
        '#submit' => array('words_all'),
    );
    $form['categories_all'] = array(
        '#type' => 'submit',
        '#value' => 'All the categories',
        '#submit' => array('categories_all'),
    );
    return $form;
}
function words_all($form, &$form_state) {
    $form_state['redirect'] = 'functions_module/search_all_2';
}
function categories_all($form, &$form_state) {
    $form_state['redirect'] = 'functions_module/search_all_3';
}

function search_all_2($form, &$form_state) {
    $header = array(
        array('data' => t('ID')),
        array('data' => t('Word')),
        array('data' => t('Description')),
        array('data' => t('Edit')),
        array('data' => t('Delete')),
    );
    // We are extending the PagerDefault class here.
    // It has a default of 10 rows per page.
    // The extend('PagerDefault') part here does all the magic.
    $query = db_select('Words', 'd')->extend('PagerDefault');
    $query->fields('d', array('word_id', 'word', 'description'));
    // Change the number of rows with the limit() call.
    $result = $query
        ->orderBy('d.word_id')
        ->execute();

    $rows = array();
    foreach ($result as $row) {
        // Normally we would add some nice formatting to our rows
        // but for our purpose we are simply going to add our row
        // to the array.
        $rows[] = array(
            $row->word_id,
            l($row->word, 'link_video/' . $row->word_id),
            $row->description,
        );
    }
}
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
                l('edit','link_edit'),
                l('delete','link_delete')
            );
        }
    $build['Words Region'] = array(
        '#type' => 'markup',
        '#markup' => t('All words in our database are below :'),
    );
    $build['pager_table'] = array(
        '#theme' => 'table',
        '#header' => $header,
        '#rows' => $rows,
        '#empty' => t('There are no WORDS found in the db'),
    );

    // Attach the pager theme.
    $build['pager_pager'] = array('#theme' => 'pager');

    return $build;
}
function search_all_3($form, &$form_state) {
// We are going to output the results in a table with a nice header.
    $header = array(
        array('data' => t('ID')),
        array('data' => t('Category ')),
        array('data' => t('Description')),
    );
    // We are extending the PagerDefault class here.
    // It has a default of 10 rows per page.
    // The extend('PagerDefault') part here does all the magic.
    $query = db_select('Categories', 'd')->extend('PagerDefault');
    $query->fields('d', array('category_id', 'category', 'description'));

    // Change the number of rows with the limit() call.
    $result = $query
        ->orderBy('d.category_id')
        ->execute();

    $rows = array();
    foreach ($result as $row) {
        // Normally we would add some nice formatting to our rows
        // but for our purpose we are simply going to add our row
        // to the array.
        $rows[] = array(
            $row->category_id,
            $row->category,
            $row->description,
        );
    }
}
```


Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
// Create a render array ($build) which will be themed as a table with a
// pager.
$build['Words Region'] = array(
  '#type' => 'markup',
  '#markup' => t('All categories in our database are below :)'),
);
$build['pager_table'] = array(
  '#theme' => 'table',
  '#header' => $header,
  '#rows' => $rows,
  '#empty' => t('There are no WORDS found in the db'),
);
// Attach the pager theme.
$build['pager_pager'] = array('#theme' => 'pager');

return $build;
}
```

```
function search_by_word($form, &$form_state) {
  $form_state['redirect'] = 'functions_module/search_by_word_1';
}
function search_by_word_1($form, &$form_state) {
  $form['word_1'] = array(
    '#type' => 'textfield',
    '#title' => t('Searching by a word.'),
    '#required' => TRUE,
    '#default_value' => "Please enter the desired word.",
    '#description' => "Please enter the desired word.",
    '#size' => 60,
    '#maxlength' => 20,
  );
  $form['submit'] = array(
    '#type' => 'submit',
    '#value' => 'Search',
    '#submit' => array('search_by_word_2'),
  );
  return $form;
}
```

```
function search_by_word_2($form, &$form_state) {
  $form_state['redirect'] =
  'functions_module/search_by_word_3/'.$form_state['values']['word_1'];
}
function search_by_word_3($word,$form, &$form_state) {
  $header = array(
    array('data' => t('ID')),
    array('data' => t('Word')),
    array('data' => t('Description')),

    array('data' => t('Delete')),
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
);
// We are extending the PagerDefault class here.
// It has a default of 10 rows per page.
// The extend('PagerDefault') part here does all the magic.
$query = db_select('Words', 'd')->extend('PagerDefault');
$query->fields('d', array('word_id', 'word', 'description'));
$query->condition('word', $word, '=');
// Change the number of rows with the limit() call.
$result = $query
->limit(2)
->orderBy('d.word_id')
->execute();
$rows = array();
foreach ($result as $row) {
// Normally we would add some nice formatting to our rows
// but for our purpose we are simply going to add our row
// to the array.
$rows[] = array(

        $row->word_id,
        l($row->word, 'link_video/' . $row->word_id),
        $row->description,

        l('delete', 'link_delete')
    );
}
$build['pager_table'] = array(
    '#theme' => 'table',
    '#header' => $header,
    '#rows' => $rows,
    '#empty' => t('There are no WORDS found in the db'),
);
// Attach the pager theme.
$build['pager_pager'] = array('#theme' => 'pager');
return $build;
}
function search_by_category($form, &$form_state) {
$form_state['redirect'] = 'functions_module/search_by_category_1';
}

function search_by_category_1($form, &$form_state) {

$form['text_2'] = array(
    '#type' => 'textfield',
    '#title' => t('Searching by a category.'),
    '#required' => TRUE,
    '#default_value' => "Please enter the desired category!",
    '#description' => "Please enter the desired category!",
    '#size' => 60,
    '#maxlength' => 60,
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
);

$form['submit'] = array(
    '#type' => 'submit',
    '#value' => 'Search',
    '#submit' => array('search_by_category_2'),
);
return $form;
}

function search_by_category_2($form, &$form_state) {
    $form_state['redirect'] =
    'functions_module/search_by_category_3/'.$form_state['values']['text_2'];
}

function search_by_category_3($cat) {

// We are going to output the results in a table with a nice header.
$header = array(
    array('data' => t('ID')),
    array('data' => t('Category ')),
    array('data' => t('Description')),
);

// We are extending the PagerDefault class here.
// It has a default of 10 rows per page.
// The extend('PagerDefault') part here does all the magic.
$query = db_select('Categories', 'd')->extend('PagerDefault');
$query->fields('d', array('category_id', 'category', 'description'));

    $query->condition('category', $cat, '=');
    // Change the number of rows with the limit() call.
    $result = $query
        ->limit(2)
        ->orderBy('d.category_id')
        ->execute();
    $rows = array();
    foreach ($result as $row) {
        // Normally we would add some nice formatting to our rows
        // but for our purpose we are simply going to add our row
        // to the array.
        $rows[] = array(
            $row->category_id,
            $row->category,
            $row->description,
        );
    }
    // Create a render array ($build) which will be themed as a table with a
    // pager.
    $build['pager_table'] = array(
```

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

```
'#theme' => 'table',  
'#header' => $header,  
'#rows' => $rows,  
'#empty' => t('There are no WORDS found in the db'),  
);  
// Attach the pager theme.  
$build['pager_pager'] = array('#theme' => 'pager');  
return $build;  
}  
>
```

Βιβλιογραφία

Βιβλία

«Ανάπτυξη Διαδικτυακών Εφαρμογών : Θεωρία και Πράξη», 2009, Κεντερλής Π.

«Συστήματα Βάσεων Δεδομένων» 6η Έκδοση, Abraham Silberschatz, Henry F. Korth, S. Sudarshan.

«Συστήματα Βάσεων Δεδομένων : Θεωρία και Πρακτική Εφαρμογή», 2006, Ι. Μανωλόπουλος, Α.Ν. Παπαδόπουλος.

«Επικοινωνήστε με νοήματα, Χειρονομίες, μορφασμοί, στάσεις... που πρέπει να ξέρετε», 2008, Desmond Morris.

«Language and Linguistics», 2012, Lyons John.

«Εισαγωγή στη Θεωρητική Γλωσσολογία», 2012, Lyons John.

«Pro Drupal 7 Development», Apress.

«HTML & CSS – DESIGN and Build Web Sites», 2011, Duckett Jon.

«Πλήρες εγχειρίδιο της HTML 5 & CSS», 2011, Laura Lemay, Rafe Colburn

«Μάθετε PHP, MySQL και Apache», 2008, Melonie Julie C.

Διαδικτυακοί Ιστότοποι

Drupal. <https://www.drupal.org/>, Διαδικτυακός Ιστότοπος διάθεσης και πληροφοριών του Drupal.

Form Drupal Api <https://www.drupal.org/node/37775>
Apache. <http://www.apache.org/>, Διαδικτυακός Ιστότοπος διάθεσης και πληροφοριών του Apache Web Server.

MySQL. <https://www.mysql.com/>, Διαδικτυακός Ιστότοπος διάθεσης και πληροφοριών της MySQL.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

MySQL Workbench. <https://dev.mysql.com/downloads/workbench/> , Διαδικτυακός Ιστότοπος διάθεσης του Workbench για σχεδιασμό και μοντελοποίηση της βάσης.

phpMyAdmin. <https://www.phpmyadmin.net/>, Διαδικτυακός Ιστότοπος διάθεσης και πληροφοριών της phpMyAdmin, για διαχείριση της βάσης.

Γλωσσολογία. <http://www.philology-upatras.gr/department/glossology>, Διαδικτυακός ιστότοπος γενικών πληροφοριών με θέμα την γλωσσολογία.

Τομείς της γλωσσολογίας. <http://www.phil.uoa.gr/tomeis/tomeas-glwsologias/glwsologia-plhrofories.html> .

Database Security (SQLI). <http://php.net/manual/en/security.database.sql-injection.php>

Παράρτημα

«Γλωσσάρι»

Επεξήγηση τεχνικών όρων

- **FTP (File Transfer Protocol):**

Το Πρωτόκολλο Μεταφοράς Αρχείων είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο σε δίκτυα τα οποία υποστηρίζουν το πρωτόκολλο TCP/IP. Ο υπολογιστής που τρέχει εφαρμογή FTP client μόλις συνδεθεί με τον server μπορεί να εκτελέσει ένα πλήθος διεργασιών όπως ανέβασμα αρχείων στον server, κατέβασμα αρχείων από τον server, μετονομασία ή διαγραφή αρχείων από τον server κ.ο.κ. Το πρωτόκολλο είναι ένα ανοιχτό πρότυπο. Είναι δυνατό κάθε υπολογιστής που είναι συνδεδεμένος σε ένα δίκτυο, να διαχειρίζεται αρχεία σε ένα άλλο υπολογιστή του δικτύου, ακόμη και εάν ο δεύτερος διαθέτει διαφορετικό λειτουργικό σύστημα.

- **HTTP (HyperText Transfer Protocol):**

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client).

- **TCP (Transmission Control Protocol):**

Το Πρωτόκολλο Ελέγχου Μεταφοράς είναι ένα από τα κυριότερα πρωτόκολλα της Σουίτας Πρωτοκόλλων Διαδικτύου. Βρίσκεται πάνω από το πρωτόκολλο IP. Οι κύριοι στόχοι του πρωτοκόλλου TCP είναι να επιβεβαιώνεται η αξιόπιστη αποστολή και λήψη δεδομένων, επίσης να μεταφέρονται τα δεδομένα χωρίς λάθη μεταξύ του στρώματος δικτύου (network layer) και του στρώματος εφαρμογής (application layer) και, φτάνοντας στο πρόγραμμα του στρώματος εφαρμογής, να έχουν σωστή σειρά. Οι περισσότερες σύγχρονες υπηρεσίες στο Διαδίκτυο βασίζονται στο TCP. Για παράδειγμα το SMTP (port 25), το παλαιότερο (και μη-ασφαλές) Telnet (port 23), το FTP και πιο σημαντικό το HTTP (port 80), γνωστό ως

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

υπηρεσίες World Wide Web (WWW - Παγκόσμιος Ιστός). Το TCP χρησιμοποιείται σχεδόν παντού, για αμφίδρομη επικοινωνία μέσω δικτύου.

- **SSL (Secure Sockets Layer):**

Το πρωτόκολλο SSL αναπτύχθηκε από την εταιρεία Netscape και σχεδιάστηκε για να παρέχει ασφάλεια κατά την μετάδοση ευαίσθητων δεδομένων στο διαδίκτυο. Η έκδοση 3.0 του πρωτοκόλλου κυκλοφόρησε από την Netscape το 1996 και αποτέλεσε την βάση για την μετέπειτα ανάπτυξη του πρωτοκόλλου TLS (Transport Layer Security), το οποίο πλέον τείνει να αντικαταστήσει το SSL. Τα δύο αυτά πρωτόκολλα χρησιμοποιούνται ευρέως για ηλεκτρονικές αγορές και χρηματικές συναλλαγές μέσω του διαδικτύου.

Το SSL χρησιμοποιεί μεθόδους κρυπτογράφησης των δεδομένων που ανταλλάσσονται μεταξύ δύο συσκευών (συνηθέστερα Ηλεκτρονικών Υπολογιστών) εγκαθιδρύοντας μία ασφαλή σύνδεση μεταξύ τους μέσω του διαδικτύου. Το πρωτόκολλο αυτό χρησιμοποιεί το TCP/IP για τη μεταφορά των δεδομένων και είναι ανεξάρτητο από την εφαρμογή που χρησιμοποιεί ο τελικός χρήστης. Για τον λόγο αυτό μπορεί να παρέχει υπηρεσίες ασφαλούς μετάδοσης πληροφοριών σε πρωτόκολλα ανώτερου επιπέδου όπως για παράδειγμα το HTTP, το FTP, το telnet κοκ.

Η μετάδοση πληροφοριών μέσω του διαδικτύου γίνεται ως επί το πλείστον χρησιμοποιώντας τα πρωτόκολλα TCP/IP (Transfer Control Protocol / Internet Protocol). Το SSL λειτουργεί πριν το TCP/IP και μετά τις εφαρμογές υψηλού επιπέδου, όπως είναι για παράδειγμα το HTTP (προβολή ιστοσελίδων), το FTP (μεταφορά αρχείων) και το IMAP (email). Άρα λοιπόν αυτό που ουσιαστικά κάνει το SSL είναι να παίρνει τις πληροφορίες από τις εφαρμογές υψηλότερων επιπέδων, να τις κρυπτογραφεί και στην συνέχεια να τις μεταδίδει στο Internet προς τον Η/Υ που βρίσκεται στην απέναντι πλευρά και τις ζητήσει.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

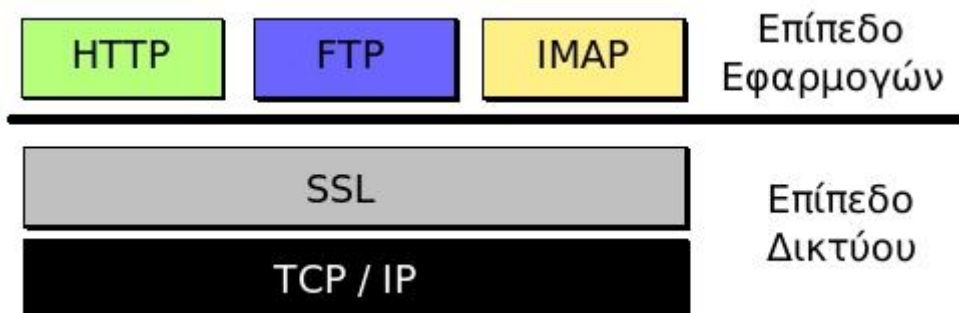


Figure 0-1 – Επίπεδα πρωτοκόλλων

- **TLS (Transport Layer Security):**

Είναι ένα πρωτόκολλο που εγγυάται ότι κατά την επικοινωνία server - client μέσω του Διαδικτύου δεν πρόκειται να μεσολαβήσει κάποιος τρίτος που θα «υποκλέψει» το περιεχόμενο της επικοινωνίας.

Το TLS έχει διαδεχθεί το Secure Sockets Layer (SSL). Χρησιμοποιούνται ως ενδιάμεσα πρωτόκολλα μεταξύ του επιπέδου εφαρμογών και του επιπέδου μεταφοράς.

Το TLS αποτελείται από δύο επίπεδα: το TLS Record Protocol και το TLS Handshake Protocol.

- **ASP (Active Server Pages):**

Η ASP, αποτελεί την πρώτη server-side script μηχανή της Microsoft δυναμικά παραγόμενες ιστοσελίδες. Η ASP.NET κυκλοφόρησε πρώτα τον Ιανουάριο του 2002 και έχει αντικαταστήσει την ASP.

- **User agent:**

Είναι ένα software (software agent) που δρα στη θέση ενός χρήστη. Για παράδειγμα, ένας email reader (αναγνώστης email) είναι ένας mail user agent και στο Session Indication Protocol (SIP) ο όρος user agent αναφέρεται και στα δύο τερματικά μιας επικοινωνίας.

- **Code injection:**

Είναι η εκμετάλλευση ενός bug του υπολογιστή, που προκαλείται από επεξεργασία μη έγκυρων δεδομένων. Το code injection, μπορεί να χρησιμοποιηθεί από έναν «εισβολέα» για να εισάγει κώδικα σε ένα πρόγραμμα του υπολογιστή για να αλλάξει την πορεία της εκτέλεσης. Τα

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα

αποτελέσματα μπορεί να είναι καταστροφικά. Για παράδειγμα, το `code injection` χρησιμοποιείται από μερικά worms για να πολλαπλασιαστεί.

- **Χαρακτήρας διαφυγής (Escape characters):**

Στην επιστήμη των υπολογιστών και στις τηλεπικοινωνίες καλούμε χαρακτήρα διαφυγής έναν μόνο χαρακτήρα ο οποίος σε μία ακολουθία χαρακτήρων σηματοδοτεί ότι οι χαρακτήρες που ακολουθούν λαμβάνουν διαφορετική ερμηνεία. Ο όρος ακολουθία διαφυγής αναφέρεται σε έναν χαρακτήρα διαφυγής και στους ακόλουθους χαρακτήρες ή χαρακτήρα που αλλάζει η ερμηνεία τους.

Οι χαρακτήρες αυτοί αποτελούν μέρος του συντακτικού για πολλές γλώσσες προγραμματισμού και πρωτόκολλα επικοινωνίας. Για ένα δεδομένο αλφάβητο, ο σκοπός ενός χαρακτήρα διαφυγής είναι να ξεκινήσει ακολουθίες χαρακτήρων (επονομαζόμενες ακολουθίες διαφυγής-escape sequences).

Συνήθως υπάρχουν δύο λειτουργίες για τις ακολουθίες διαφυγής. Η πρώτη είναι η κωδικοποίηση μιας συντακτικής οντότητας, όπως είναι οι εντολές συσκευής ή ειδικά δεδομένα, που δεν μπορούν απ' ευθείας να αντιπροσωπευτούν από ένα αλφάβητο. Η δεύτερη χρήση, που αναφέρεται ως παράθεση χαρακτήρων (*character quoting*), είναι να αντιπροσωπεύει χαρακτήρες που δεν μπορούν να τυπωθούν σε τωρινό περιεχόμενο, αλλιώς θα είχε μια δυσάρεστη ερμηνεία. Στην τελευταία περίπτωση μια ακολουθία διαφυγής είναι ένας δίφθογγος, που αποτελεί έναν χαρακτήρα διαφυγής από μόνος του.

- **Cross-site scripting ή XSS:**

Αναφερόμαστε στην εκμετάλλευση διάφορων ευπαθειών (vulnerabilities) υπολογιστών με εισαγωγή κώδικα HTML ή Javascript σε κάποιο ιστοχώρο. Κάποιος κακόβουλος χρήστης, θα μπορούσε να εισάγει κώδικα σε έναν ιστοχώρο, μέσω ενός κειμένου εισόδου για παράδειγμα, ο οποίος αφού δε θα φιλτραριζόταν από τον ιστοχώρο σωστά, θα μπορούσε να προκαλέσει προβλήματα στον διαχειριστή ή επισκέπτη του ιστότοπου.

Ανάπτυξη διαδικτυακής εφαρμογής δημιουργίας γλωσσικών πόρων για τη νοηματική γλώσσα