



Πανεπιστήμιο
Ιωαννίνων

ΤΜΗΜΑ: ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΦΑΡΜΟΓΗ ΜΗΤΡΩΟΥ ΕΠΙΣΤΗΜΟΝΙΚΩΝ
ΔΗΜΟΣΙΕΥΣΕΩΝ ΜΕΛΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ: ΠΑΥΛΟΣ ΤΙΦΤΙΚΙΔΗΣ
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΓΚΟΓΚΟΣ ΧΡΗΣΤΟΣ

ΑΡΤΑ 2020



Πανεπιστήμιο
Ιωαννίνων

DEPARTMENT: INFORMATICS AND
TELECOMMUNICATIONS

THESIS

A SOFTWARE APPLICATION THAT CAPTURES AND
TRACKS DATA RELATED TO SCIENTIFIC
PUBLICATIONS OF THE DEPARTMENT OF
INFORMATICS AND TELECOMMUNICATIONS

STUDENT: PAVLOS TIFTIKIDIS
SUPERVISOR: GOGOS CHRISTOS

ARTA 2020

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

ΤΙΦΤΙΚΙΔΗΣ ΠΑΥΛΟΣ

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως θέμα την ανάπτυξη Εφαρμογής μητρώου επιστημονικών δημοσιεύσεων μελών του τμήματος Μηχανικών Πληροφορικής. Συγκεκριμένα η ανάπτυξη εφαρμογής αποτελείται από τρία στάδια. Πρώτο στάδιο είναι η μελέτη, δεύτερο στάδιο η σχεδίαση και τρίτο στάδιο είναι η υλοποίηση της διαδικτυακής εφαρμογής στην οποία θα καταγράφεται το δημοσιευμένο ερευνητικό έργο των μελών του Τμήματος. Αναλυτικότερα το σύστημα θα δίνει τη δυνατότητα αυτόματης ενημέρωσης των δημοσιεύσεων και θα παράγει τα απαιτούμενα στατιστικά στοιχεία, πίνακες και γραφήματα που περιέχονται στις ετήσιες εκθέσεις αξιολόγησης του Τμήματος.

Λέξεις-κλειδιά: scholar, web scraping, puppeteer, php, mysql, επιστημονικό έργο, καθηγητές, διαδικτυακή εφαρμογή, ετήσια έκθεση

ABSTRACT

The present work aims at the elaboration of a study of application development of scientific publications, member of the department of Informatics and Telecommunications of the University of Ioannina. In particular, the complete development of the application consists of 3 stages. The first stage is study of the problem, the second is the planning and the third is the implementation of the web application that presents the published research work of the department's faculty. In more detail, the system allows the automatic update of publications and produce data needed for the annual evaluation reports of the department.

Keywords: scholar, web scraping, puppeteer, php, mysql, scientific work, professors, web application, annual report

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΕΙΣΑΓΩΓΗ.....	i
1.1	Σκοπός της πτυχιακής εργασίας.....	i
1.2	Δομή της πτυχιακής εργασίας	i
2	ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ.....	iii
2.1	Εισαγωγή.....	iii
2.2	Ανάπτυξη λογισμικού με το μοντέλο του καταρράκτη	vi
2.3	Ανάπτυξη λογισμικού με ευέλικτες τεχνικές.....	ix
2.3.1	Σκοπός	ix
2.3.2	Βασικές αρχές.....	ix
2.3.3	Εφαρμογή	x
2.3.4	Προβλήματα.....	x
2.4	Ακραίος προγραμματισμός	xi
2.4.1	12 πρακτικές ακραίου προγραμματισμού.....	xi
2.4.2	Γιατί ακραίος;.....	xii
3	Ανάπτυξη εφαρμογών διαδικτύου.....	xiii
3.1	Εισαγωγή.....	xiii
3.2	Βασικά εργαλεία ανάπτυξης λογισμικού	xiii
3.3	Εξειδικευμένα εργαλεία ανάπτυξης λογισμικού	xvi
3.4	Υπηρεσίες φιλοξενίας εφαρμογών στο διαδίκτυο.....	xix
4	Έρευνα και επιστημονικά άρθρα	xxiii
4.1	Εισαγωγή.....	xxiii
4.2	Επιστημονικά άρθρα	xxiii
4.3	Πεδία ενδιαφέροντος επιστημονικών άρθρων	xxiv
4.4	Πλατφόρμα καταλογογράφησης επιστημονικών άρθρων.....	xxv
4.4.1	Google Scholar	xxv
4.4.2	Academia Edu.....	xxv
4.4.3	Microsoft Academic.....	xxvi
4.4.4	Semantic Scholar	xxvi
4.5	Αξιολόγηση επιστημονικού έργου	xxvii
4.5.1	Αναφορές	xxvii
4.5.2	Μετρικές.....	xxvii
4.5.3	Οι μετρικές h-index, i10 κ.α.....	xxvii
5	Λογισμικά επιστημονικών δημοσιεύσεων	xxix

5.1	Λογισμικό που απευθύνονται σε χρήστες	xxix
5.1.1	Publish or Perish.....	xxix
5.1.2	Go Scholar for Google Scholar.....	xxx
5.1.3	Researcher.....	xxxι
5.2	Λογισμικά που απευθύνονται σε προγραμματιστές.....	xxxii
5.2.1	Χρήση API	xxxii
5.2.2	Web scraping	xxxiv
5.2.3	Παράκαμψη των anti-scraping τεχνικών	xliii
6	Σχεδιασμός εφαρμογής.....	xlν
6.1	Εισαγωγή.....	xlν
6.2	Προδιαγραφές.....	xlν
6.2.1	Περιπτώσεις χρήσης.....	xlvii
6.3	Τεχνολογίες και Εργαλεία Υλοποίησης	liii
6.3.1	HTML.....	liii
6.3.2	JavaScript.....	liii
6.3.3	CSS.....	liv
6.3.4	PHP	liv
6.3.5	MYSQL	lv
6.4	Framework.....	lvi
6.4.1	Τι είναι τα Frameworks	lvi
6.4.2	Χαρακτηριστικά	lvi
6.4.3	Πλεονεκτήματα	lvi
6.4.4	Λειτουργία.....	lvii
7	Υλοποίηση εφαρμογής	lix
7.1	Εγκατάσταση framework	lix
7.1.1	Βασικές ρυθμίσεις.....	lxi
7.2	Βάση δεδομένων	lxν
7.2.1	Migration.....	lxvi
7.3	Controllers	lxx
7.4	Views.....	lxxi
7.5	Models.....	lxxvi
7.6	Libraries	lxxviii
7.7	Web scraping	lxxx
8	Παρουσίαση εφαρμογής.....	lxxxvii
8.1	Εισαγωγή.....	lxxxvii
8.2	Αυθεντικοποίηση χρήστη.....	lxxxvii

8.3	Εισαγωγή καθηγητή.....	lxxxviii
8.4	Εισαγωγή δεδομένων	xc
8.5	Επισκόπηση ερευνητικού έργου.....	xcι
8.6	Στατιστικά στοιχεία	xcii
9	Συμπεράσματα και μελλοντικές επεκτάσεις	xciv
10	Αναφορές	xcv

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1.2: Φάσεις ανάπτυξης λογισμικού.	iii
Εικόνα 2.1.3: Πίνακα σύγκρισης βάση ορισμένων χαρακτηριστικών των μοντέλων κύκλου ζωής.	v
Εικόνα 2.1.4: Βήματα εκτέλεσης μιας διαδικασίας.	vi
Εικόνα 2.2.1: Διαδικασία του καταρράκτη.	vii
Εικόνα 2.4.1: 12 πρακτικές ακραίου προγραμματισμού.	xii
Εικόνα 3.2.1: Γραφικό περιβάλλον Notepad++.	xiv
Εικόνα 3.2.2: Γραφικό περιβάλλον Sublime.....	xv
Εικόνα 3.2.3: Γραφικό περιβάλλον Atom.....	xvi
Εικόνα 3.3.1: Γραφικό περιβάλλον Visual Studio.....	xvii
Εικόνα 3.3.2: Γραφικό περιβάλλον NetBeans.....	xviii
Εικόνα 3.3.3: Γραφικό περιβάλλον JetBrains.....	xix
Εικόνα 4.2.1: Στάδια ενός επιστημονικού έργου. (Διέτης, 2017)	xxiii
Εικόνα 5.2.1: Γραφικό περιβάλλον Publish or Perish.....	xxx
Εικόνα 5.3.1: Παράδειγμα επιλογής πεδίων.....	xxxvi
Εικόνα 5.3.2: Παροχή υπηρεσιών από την cloud υπηρεσία της Octoparse.	xxxvii
Εικόνα 5.3.3: Επιλογή πεδίων ενδιαφέροντος με το εργαλείο Webscraper.io	xxxviii
Εικόνα 5.3.4: Τελικό σχήμα επιλογής πεδίων και λειτουργίας διαδικασίας.....	xxxviii
Εικόνα 5.3.5: Τελικά δεδομένα ύστερα από εκτέλεση της διαδικασίας με την βοήθεια του εργαλείο Webscraper.io	xxxix
Εικόνα 5.3.6: Ιστότοπος από τον οποίο κάνουμε web scraping με την βοήθεια του puppeteer.....	xli
Εικόνα 6.4.1: Διάγραμμα λειτουργίας του μοντέλου MVC.....	lvii
Εικόνα 7.1.1: Τοποθεσία φιλοξενίας των αρχείων της εφαρμογής μας.....	lix
Εικόνα 7.1.2: Module τα οποία πρέπει να ενεργοποιήσουμε κάθε φορά που εκκινούμε το Xampp.....	lx
Εικόνα 7.1.3: Αποτέλεσμα που επιστρέφει το url: localhost μας διαβεβαιώνει πως το Xampp με το CodeIgniter λειτουργούν σωστά.....	lx
Εικόνα 7.1.4: Στη διαδρομή htdocs/application/config/autoload.php κάνουμε την εξής αλλαγή.....	lxi
Εικόνα 7.1.5: Στη διαδρομή htdocs/application/config/config.php	lxi
Εικόνα 7.1.6: Στη διαδρομή htdocs/application/config/database.php συμπληρώνουμε τα στοιχεία σύνδεσης μας με τη βάση δεδομένων.	lxi
Εικόνα 7.2.1: Πίνακες της βάσης δεδομένων.	lxvi
Εικόνα 7.2.2: Ενεργοποίηση migration στην εφαρμογή μας ακολουθώντας τη διαδρομή htdocs/application/config/migration.php και κάνοντας κάποιες αλλαγές όπως βλέπουμε στην παρακάτω εικόνα.	lxvii
Εικόνα 7.2.3: Οι πίνακες στη βάση δεδομένων.	lxix
Εικόνα 7.3.1: Κώδικας που εκτελείτε όταν βρει τον controller «dashboard» και από προεπιλογή θα εκτελέσει την function index.	lxx
Εικόνα 7.3.2: Επισκόπηση των Controllers που δημιουργήσαμε για την εφαρμογή μας.....	lxxi
Εικόνα 7.4.1: Στη γραμμή 24 με 26 φορτώνουμε τα views της κεντρικής σελίδας.....	lxxv

Εικόνα 7.4.2: PHP κώδικας όπου με την βοήθεια της foreach «εκτυπώνουμε» όλα τα δεδομένα που περιέχονται μέσα σε ένα array. Το παραπάνω κομμάτι περιέχεται μέσα σε κώδικα HTML.....	Ixxv
Εικόνα 7.4.3: Κεντρική σελίδα της εφαρμογής μας όπου απεικονίζει δύο διαγράμματα. .	Ixxvi
Εικόνα 7.4.4: Views τα οποία περιέχονται στην εφαρμογή μας.	Ixxvi
Εικόνα 7.5.1: Παράδειγμα ενός model το οποίο επικοινωνεί με τον πίνακα Articles.	Ixxvii
Εικόνα 7.5.2: Όλα τα models τα οποία περιέχονται στην εφαρμογή μας.	Ixxviii
Εικόνα 7.6.1: Βασικός κώδικας που πρέπει να περιέχεται μέσα στη κλάση mylib.php.....	Ixxix
Εικόνα 7.7.1: Στη γραμμή 60 βλέπουμε την κλείσει της function που φτιάξαμε η οποία περιέχεται μέσα στη βιβλιοθήκη mylib.php.....	Ixxxii
Εικόνα 7.7.2: Χρήση της τεχνολογίας CURL.	Ixxxii
Εικόνα 7.7.3: function η οποία μας επιστρέφει κάθε φορά μια διαφορετική διεύθυνση.	Ixxxiii
Εικόνα 7.7.4: Κομμάτι κώδικα της function prof_list_scraping.	Ixxxiii
Εικόνα 7.7.5: Κώδικας ο οποίος αφού έχουν ανακτηθεί όλοι η σελίδα html που ζητήσαμε επιλέγουμε τις πληροφορίες που μας ενδιαφέρουν.....	Ixxxiv
Εικόνα 7.7.6: Πεδία από τα οποία θα ανακτήσουμε τις πληροφορίες που μας είναι χρήσιμες.	Ixxxv
Εικόνα 7.7.7: Πεδία με την πληροφορία που θα ανακτήσουμε.	Ixxxvi
Εικόνα 8.2.1: Σελίδα αυθεντικοποίησης χρήστη.	Ixxnvi
Εικόνα 8.2.2: Κεντρική σελίδα της εφαρμογής με τα στατιστικά στοιχεία τα οποία έχουν ανακτηθεί από τη βάση δεδομένων.	Ixxnviii
Εικόνα 8.3.1: Σελίδα κατοχυρωμένων καθηγητών στη βάση δεδομένων.....	Ixxnviii
Εικόνα 8.3.2: Modal με τα πεδία του για την προσθήκη ενός καθηγητή στη βάση δεδομένων.	Ixxxi
Εικόνα 8.3.3: Εισαγωγή καθηγητή στη λίστα χωρίς Scholar Id στη βάση δεδομένων.	Ixxxi
Εικόνα 8.3.4: Πατώντας το καπελάκι ανοίγει ένα modal με τους καθηγητές που έχουν το ίδιο όνομα, παρέχοντας μας κάποιες παραπάνω πληροφορίες για την σωστή επιλογή του καθηγητή που μας ενδιαφέρει.	xc
Εικόνα 8.4.1: Καταχωρημένο Scholar Id το διακρίνουμε από το μπλε καπελάκι και τρεις επιλογές που μπορούμε να κάνουμε για έναν καθηγητή.	xc
Εικόνα 8.5.1: Πίνακας με το ερευνητικό έργο και τις λεπτομέρειες του κάθε έργου που υπάρχει στη βάση δεδομένων.	xc
Εικόνα 8.5.2: Στη στήλη citation πατώντας πάνω στον αριθμό μπορούμε να δούμε τις αναφορές του άρθρου με τις λεπτομέρειές του.....	xcii
Εικόνα 8.6.1: Απεικόνιση στατιστικών στοιχείων με την βοήθεια διαγράμματος και πίτας.	xciii

1 ΕΙΣΑΓΩΓΗ

1.1 Σκοπός της πτυχιακής εργασίας

Στη σημερινή εποχή η μεγαλύτερη πλειοψηφία των πληροφοριών διακινείται μέσω του διαδικτύου και αυτό γιατί η πληροφορία μεταφέρεται με μεγάλη ταχύτητα στον προορισμό της. Αυτό έχει ως αποτέλεσμα τη ραγδαία αύξηση των συσκευών αυτών που έχουν πρόσβαση στο διαδίκτυο τα τελευταία χρόνια. Τέτοιες συσκευές μπορεί να είναι ένας ηλεκτρονικό υπολογιστή, κινητές συσκευές μέχρι και το προσωπικό μας αυτοκίνητο.

Η μεγάλη ταχύτητα μεταφοράς δεδομένων μέσω του διαδικτύου δεν εξυπηρετεί μόνο σε προσωπικό επίπεδο κάθε έναν χρήστη μιας τέτοιας συσκευής αποκλειστικά αλλά εξυπηρετεί και ολόκληρες επιχειρήσεις που η βασική λειτουργία τους γίνεται μέσω του διαδικτύου.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η αυτοματοποιημένη εύρεση του δημοσιευμένου επιστημονικού έργου των μελών του τμήματος και της παραγωγής των απαιτούμενων στατιστικών στοιχείων, πινάκων και γραφημάτων που περιέχονται στις ετήσιες εκθέσεις αξιολόγησης του τμήματος.

1.2 Δομή της πτυχιακής εργασίας

Η δομή της πτυχιακής εργασίας αποτελείται χωρίς το εισαγωγικό, από επτά κεφάλαια, τα συμπεράσματα και τη βιβλιογραφία.

Εξαιρώντας το εισαγωγικό κεφάλαιο, στο 2^ο κεφάλαιο γίνεται αναφορά στα μοντέλα ανάπτυξης λογισμικού. Τα μοντέλα ανάπτυξης λογισμικού αποτελούν το αρχικό στάδιο της ανάπτυξης ενός οποιοσδήποτε λογισμικού και περιέχει την εύρεση μεθοδολογιών για την περιγραφή, τη κατασκευή και τη συντήρηση του λογισμικού.

Το 3^ο κεφάλαιο αφορά τα εργαλεία τα οποία χρησιμοποιεί ένας προγραμματιστής για την ανάπτυξη του λογισμικού που του έχουν αναθέσει. Τα εργαλεία χωρίζονται σε βασικά εργαλεία και εξειδικευμένα.

Στο 4^ο κεφάλαιο γίνεται αναφορά σχετικά με την έρευνα και τα επιστημονικά άρθρα τα οποία δημοσιεύει ένας ερευνητής, από ποια πεδία ενδιαφέροντος αποτελείται, πλατφόρμες καταλογογράφησης και πως αξιολογείται ένα επιστημονικό έργο.

Το 5^ο κεφάλαιο κάνει λόγο για τα εργαλεία τα οποία μπορεί να χρησιμοποιήσει ένας αναλυτής δεδομένων επιστημονικών άρθρων με σκοπό να παράγει στατιστικά στοιχεία καθώς επίσης και να κάνει ανάγνωση των έργων που έχουν αναρτηθεί.

Το 6^ο κεφάλαιο αφορά τον σχεδιασμό της εφαρμογής και συγκεκριμένα παρουσιάζονται οι προδιαγραφές της εφαρμογής που θα αναπτυχθεί για την παρούσα

πτυχιακή εργασία, τις περιπτώσεις χρήσης καθώς επίσης τις τεχνολογίες και τα εργαλεία που χρησιμοποιούνται για την ανάπτυξη της εφαρμογής.

Το 7^ο κεφάλαιο παρουσιάζει την υλοποίηση της εφαρμογής καθώς επίσης και τα πρώτα βήματα εγκατάστασης του λογισμικού που απαιτούνται για την έναρξη της ανάπτυξης της εφαρμογής.

Στο 8^ο κεφάλαιο γίνεται η παρουσίαση της εφαρμογής.

Τέλος στο 9^ο και 10^ο κεφάλαιο παρουσιάζονται τα συμπεράσματα και μελλοντικές επεκτάσεις οι οποίες προκύπτουν από την παρούσα πτυχιακή εργασία κατά τη διάρκεια της συγγραφής της μέχρι την ολοκλήρωσή της και αντίστοιχα στο τελευταίο κεφάλαιο περιέχονται η αναφοράς της πτυχιακής εργασίας.

2 ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

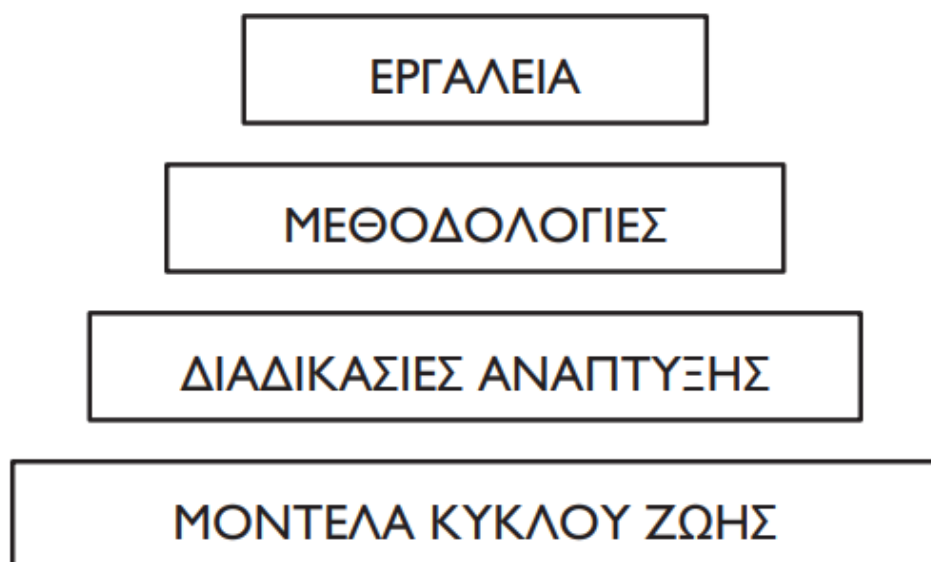
2.1 Εισαγωγή

Πριν ξεκινήσουμε με τον σκοπό του κεφαλαίου και τι είναι ανάπτυξη λογισμικού καλό θα ήταν να ορίσουμε μια βασική έννοια στον χώρο αυτόν, την Τεχνολογία Λογισμικού.

Τεχνολογία λογισμικού είναι το τμήμα αυτό της πληροφορικής το οποίο ασχολείται με την εύρεση μεθοδολογιών για την περιγραφή, κατασκευή και τη συντήρηση λογισμικού καλής ποιότητας με όσο το δυνατόν περισσότερη αυτοματοποίηση, παραγωγικότητα και ελάχιστο δυνατό κόστος.

Σκοπός αυτού του κεφαλαίου είναι να ορίσουμε τις έννοιες ανάπτυξη λογισμικού και μοντέλο κύκλου ζωής λογισμικού ώστε να δούμε πως σχετίζονται αυτές οι δύο έννοιες.

Ανάπτυξη λογισμικού δεν είναι μόνο η συγγραφή κώδικα αλλά η προετοιμασία και διαδικασία πριν τη συγγραφή αλλά και κατά τη συγγραφή.



Εικόνα 2.1.1: Έννοιες από τις οποίες αποτελείται η ανάπτυξη λογισμικού.

Ξεκινώντας από κάτω προς τα πάνω τα

- Τα Μοντέλα κύκλου ζωής λογισμικού περιγράφουν τις φάσεις από τις οποίες αποτελείται μια εφαρμογή λογισμικού από τη σύλληψη μέχρι και την απόσυρση της.



Εικόνα 2.1.1: Φάσεις ανάπτυξης λογισμικού.

- Διαδικασία ανάπτυξης λογισμικού ορίζονται οι ενέργειες που πρέπει να γίνουν για να εκτελεστεί κάποια από τις 2 φάσεις του μοντέλου κύκλου ζωής του λογισμικού.
 - (1) Κατασκευή
 - (2) Χρήση-συντήρηση
- Στη συνέχεια μια μεθοδολογία καθορίζει πως εκτελούνται οι διαδικασίες που αναφέραμε παραπάνω, δηλαδή ποια βήματα γίνονται σε καθεμία.
- Για την εκτέλεση των εργασιών δεν θα μπορούσαν να λείπουν τα ειδικά εργαλεία τα οποία υποστηρίζονται.

Τα μοντέλα με τα οποία μπορούμε να ξεκινήσουμε την ανάπτυξη λογισμικού είναι αρκετά. Παρακάτω θα κάνουμε μια απλή αναφορά στα πιο διαδεδομένα μοντέλα και θα αναλύσουμε περαιτέρω δύο από αυτά.

- Καταρράκτη
- Αυξητικό
- Επαναληπτικό
- Προτυποποίηση
- Σπειροειδές
- Rational Unified Process
- Ευέλικτο
- Scrum
- DevOps

Τα μοντέλα κύκλου ζωής λογισμικού που αναφέραμε παραπάνω χωρίζονται σε δύο κατηγορίες, τα

- ακολουθιακά
- επαναληπτικά

Στην περίπτωση των ακολουθιακών μοντέλων η ανάπτυξη γίνεται σε διακριτές φάσεις για όλο το σύστημα λογισμικού, ενώ στην περίπτωση των επαναληπτικών όλη η διαδικασία γίνεται τμηματικά.

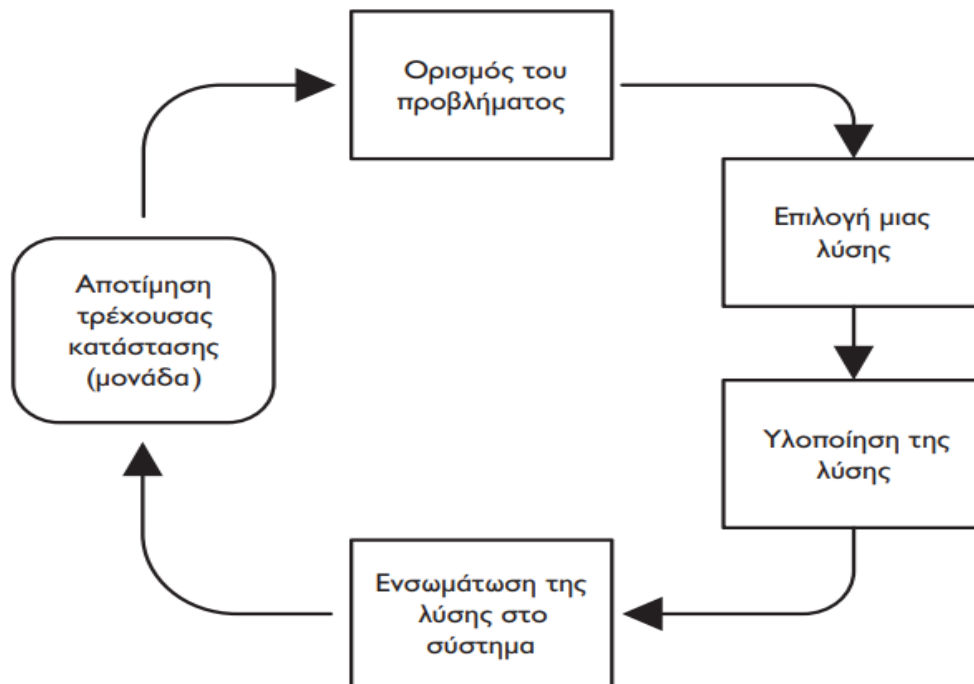
Με βάση όλα όσα αναφέραμε προκύπτει το εξής ερώτημα, υπάρχει το ιδανικό μοντέλο;

Όχι, ιδανικό μοντέλο δεν υπάρχει, το καθένα με τη σειρά του έχει τα αρνητικά και τα θετικά του σε σχέση με όλα τα άλλα. Ειδικότερα εύχρηστα μοντέλα θεωρούνται αυτά που μας δίνουν περισσότερη ελευθερία, δηλαδή εάν προκύψει μια αλλαγή κατά τη διάρκεια της ανάπτυξης θα είναι μικρό το κόστος διόρθωσης.

Μοντέλο	Μέγεθος εφαρμογών	Μεταβολές στις απαιτήσεις	Προσαρμοστικότητα στον κατασκευαστή	Διάδοση
Καταρράκτη	Μικρό έως μεσαίο	Ανεπιθύμητες	Καμία	Μεγάλη με τάση μείωσης
Πρωτοτυποποίησης	Μικρό ως μεσαίο	Δεκτές	Μικρή	Μικρή με τάση αύξησης
Λειτουργικής επαύξησης	Μεσαίο ως μεγάλο	Ανεπιθύμητες	Καμία	Μικρή με τάση μείωσης
Σπειροειδές	Μεσαίο ως μεγάλο	Δεκτές	Αρκετή	Μικρή με τάση μείωσης
Πίδακα	Οποιοδήποτε	Δεκτές	Αρκετή	Μικρή
Γενικό	Οποιοδήποτε	Δεκτές	Μεγάλη	Μικρή με ισχυρές τάσεις αύξησης

Εικόνα 2.1.2: Πίνακα σύγκρισης βάση ορισμένων χαρακτηριστικών των μοντέλων κύκλου ζωής.

Για την επιλογή του κατάλληλου μοντέλου κύκλου ζωής λογισμικού, εξαρτώνται σε μεγάλο βαθμό οι συνθήκες που επικρατούν στο περιβάλλον που θα γίνει η ανάπτυξη του λογισμικού, η εμπειρία και η τεχνογνωσία του κατασκευαστή ή των κατασκευαστών και τέλος το είδος της εφαρμογής που θα αναπτυχθεί.



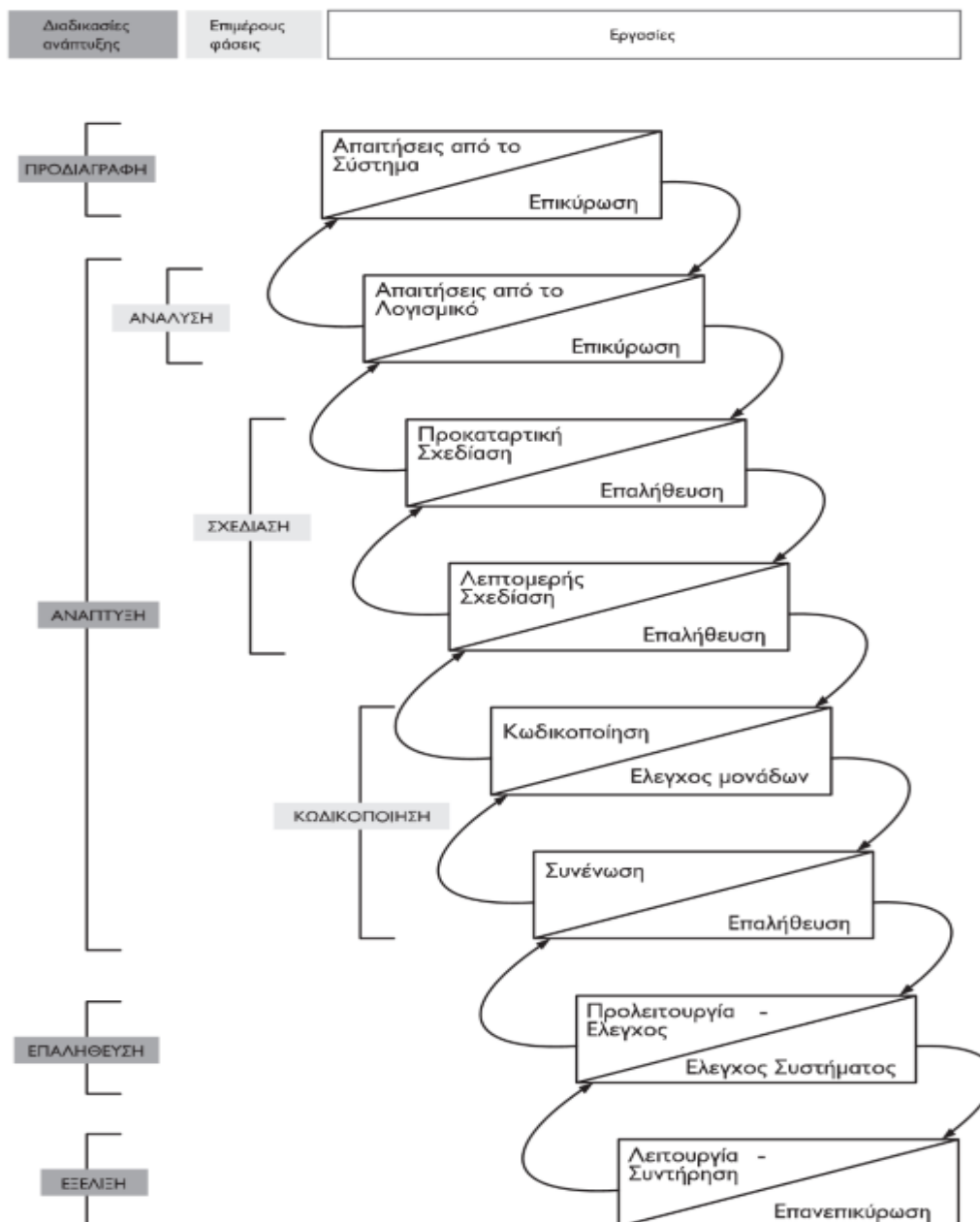
Εικόνα 2.1.3: Βήματα εκτέλεσης μιας διαδικασίας.

Ένας μηχανικός λογισμικού μέχρι την ολοκλήρωση μιας εφαρμογής εκτελεί συνεχώς τέτοιες διαδικασίες επίλυσης προβλημάτων. Τα παραπάνω βήματα εκτελούνται σε μικροσκοπικό επίπεδο, δηλαδή τμηματικά ή σε μακροσκοπικό δηλαδή σε ολόκληρο το σύστημα λογισμικού.

Τέλος η χρησιμότητα ενός μοντέλου κύκλου ζωής λογισμικού είναι η διευκόλυνση και καθοδήγηση του κατασκευαστή, με σκοπό να πετύχει την καλύτερη δυνατή υλοποίηση. Καλύτερη υλοποίηση είναι αυτή με τα λιγότερα δυνατά σφάλματα και όσον τον δυνατόν μικρότερο ρίσκο στις συνθήκες που αναπτύσσεται.

2.2 Ανάπτυξη λογισμικού με το μοντέλο του καταρράκτη

Το μοντέλο του καταρράκτη (Waterfall) θεωρείται ένα από τα πιο διαδεδομένα μοντέλα κύκλου ζωής ενός λογισμικού. Η λειτουργία που χαρακτηρίζει αυτό το μοντέλο είναι πως το σύστημα λογισμικού αναπτύσσεται περνώντας ολόκληρο από διαδοχικές φάσεις, εάν μια φάση δεν ολοκληρωθεί δεν μεταβαίνει στην επόμενη. Το συγκεκριμένο μοντέλο χαρακτηρίζεται ακολουθιακό, γιατί οι επιμέρους φάσεις από τις οποίες διέρχεται είναι διαφορετικές και ακολουθούν η μία την άλλη.



Εικόνα 2.2.1: Διαδικασία του καταρράκτη.

Σε μια πιο λεπτομερή ανάλυση βάση του σχήματος που θα γίνει παρακάτω θα προσπαθήσουμε να καταλάβουμε πως λειτουργεί το μοντέλο του καταρράκτη.

- Δημιουργία απαιτήσεων λογισμικού και συστήματος
Οι απαιτήσεις λογισμικού προδιαγράφουν τι θα κάνει το λογισμικό, εκφράζει δηλαδή τις ανάγκες του πελάτη. Οι απαιτήσεις συστήματος εκφράζουν τις απαιτήσεις που θα έχει το λογισμικό από το σύστημα για να λειτουργεί ομαλά. Με βάση αυτές τις απαιτήσεις λογισμικού μπορούμε να προχωρήσουμε παρακάτω.

- Προκαταρτική και λεπτομερής σχεδίαση
Προκαταρτική σχεδίαση του λογισμικού είναι η διαδικασία όπου καθορίζονται οι μονάδες που θα αποτελούν το λογισμικό όπως επίσης και ποιες είναι οι συσχετίσεις μεταξύ τους.
Λεπτομερής σχεδίαση καθορίζεται η εσωτερική δομή κάθε μονάδας λογισμικού η οποία αντιστοιχεί σε μονάδες πηγαίου κώδικα προγράμματος. Αυτή η διαδικασία περιλαμβάνει όλα τα απαραίτητα στοιχεία (αλγόριθμοι, δομές δεδομένων κ.λπ.) με αποτέλεσμα η συγγραφή κώδικα να είναι μια διαδικασία ολοκλήρωσης και μόνο.
- Συνένωση
Η συνένωση είναι η διαδικασία η οποία ενώνει τις μονάδες μεταξύ τους έτσι ώστε να προκύψει ένα ολοκληρωμένο σύστημα λογισμικού. Ένα πρόγραμμα αποτελείται από πολλές μονάδες ή μια μονάδα.
- Δοκιμές
Για να εξασφαλίσει ένας προγραμματιστής ότι μια εφαρμογή δουλεύει σωστά πρέπει να γίνουν πολλά είδη δοκιμών που προγραμματίζονται κατά τον σχεδιασμό του λογισμικού και πραγματοποιούνται κατά τη διάρκεια της ανάπτυξης και της εγκατάστασης. Σκοπός των δοκιμών είναι το συντομότερο δυνατόν να επιλυθούν πιθανά προβλήματα έτσι ώστε να αποφευχθεί η σπατάλη χρόνου και πολυπλοκότητα των προβλημάτων προς το τέλος της υλοποίησης.
- Συντήρηση
Αφού παραδοθεί το έργο στον πελάτη περνάμε στη φάση συντήρησης της εφαρμογής για τυχόν βελτιώσεις της λειτουργίας των διάφορων προβλημάτων τα οποία παραλήφθηκαν στη διαδικασία της σχεδίασης ή υλοποίησης.

Τα προτερήματα του συγκεκριμένου μοντέλου είναι:

- Διακριτότητα των φάσεων.
- Επικοινωνία μέσω επίσημων εγγράφων.
- Κάθε στάδιο ολοκληρώνεται με έλεγχο.
- Περιγραφή με ακρίβεια και σαφήνεια τις φάσεις ανάπτυξης ενός προϊόντος.

Τα μειονεκτήματα είναι:

- Δυσκολία στην πραγματοποίηση αλλαγών όταν η διαδικασία έχει ξεκινήσει.
- Χρησιμοποιείτε κυρίως για μεγάλα έργα ανάπτυξης.
- Χρησιμοποιείτε μόνο όταν οι απαιτήσεις είναι καλά κατανοητές και αναμένουν πολύ λίγες τροποποιήσεις.

Το βασικότερο συμπέρασμα μας για το συγκεκριμένο μοντέλο καταρράκτη είναι πως το μοντέλο καταρράκτη είναι χρήσιμο όταν ξέρουμε από την αρχή τις απαιτήσεις από

το λογισμικό και δεν μεταβάλλονται κατά τη διάρκεια της ανάπτυξης. (ΒΕΣΚΟΥΚΗΣ, 2015)

2.3 Ανάπτυξη λογισμικού με ευέλικτες τεχνικές

2.3.1 Σκοπός

Σκοπός ανάπτυξης των ευέλικτων τεχνικών είναι η γρήγορη παράδοση λειτουργικού λογισμικού και η ταχεία εξέλιξη σε ένα περιβάλλον με μεταβαλλόμενες απαιτήσεις. Αυτό έχει ως αποτέλεσμα την ελάττωση κόστους στη διαδικασία ανάπτυξης του λογισμικού και τη γρήγορη εξυπηρέτηση των απαιτήσεων των πελατών. Αυτές οι μέθοδοι επικεντρώνονται κυρίως στον κώδικα και όχι τόσο στον σχεδιασμό της εφαρμογής.

2.3.2 Βασικές αρχές

Χρησιμοποιώντας τις ευέλικτες τεχνικές κατά τη διάρκεια της υλοποίησης ανακαλύπτουμε καλύτερους τρόπους ανάπτυξης λογισμικού. Όλη αυτή η διαδικασία κάνει τους ανθρώπους να αναπτύσσουν καλύτερη επικοινωνία μεταξύ τους και κυρίως τις σχέσεις μεταξύ του προγραμματιστή και του πελάτη από τη διαπραγμάτευση συμβολαίων.

Παρακάτω θα δούμε αναλυτικότερα έναν πίνακα με τις αρχές των ευέλικτων τεχνικών:

Συμμετοχή του πελάτη	Οι πελάτες πρέπει να συμμετέχουν στενά σε όλη τη διαδικασία της ανάπτυξης. Ο ρόλος που έχουν οι πελάτες στον χώρο ανάπτυξης λογισμικού είναι να παρέχουν νέες απαιτήσεις συστήματος καθορίζοντας την προτεραιότητα τους, καθώς επίσης και να αξιολογούν τις επαναλήψεις του συστήματος.
Βαθμιαία παράδοση	Το λογισμικό αναπτύσσεται σε επαυξήσεις και ο πελάτης καθορίζει τις απαιτήσεις που θα συμπεριληφθούν σε κάθε επαύξηση.
Άνθρωποι, όχι διαδικασίες	Πάντοτε πρέπει να προσδιορίζονται και να αξιοποιούνται οι ικανότητες της ομάδας ανάπτυξης. Τα μέλη της ομάδας πρέπει να έχουν το ελεύθερο να

	αναπτύξουν τους δικούς τους τρόπους εργασίας, χωρίς προδιαγεγραμμένες διαδικασίες.
Πρόβλεψη για μεταβολές	Να προβλέπονται ότι οι απαιτήσεις του συστήματος θα αλλάξουν και γι' αυτό το λόγο το σύστημα πρέπει να έχει σχεδιασθεί με τρόπο που να μπορεί να αντιμετωπίσει εύκολα αυτές τις αλλαγές.
Διατήρηση απλότητας	Επικέντρωση στην απλότητα τόσο όσον αφορά το λογισμικό που αναπτύσσεται όσο και στη διαδικασία ανάπτυξης που ακολουθείται. Όπου είναι δυνατόν, πρέπει να γίνεται συνειδητή προσπάθεια να εξαλείφεται η πολυπλοκότητα του συστήματος.

(Παπαδόπουλος, n.d.)

2.3.3 Εφαρμογή

Οι συγκεκριμένες τεχνικές βρίσκουν εφαρμογές σε:

- Μικρά ή μεσαία συστήματα λογισμικών τα οποία ζητούνται από τους πελάτες και αναθέτονται σε μια εταιρία ανάπτυξης λογισμικού.
- Παραγγελίες ενός συστήματος σε εσωτερικό επίπεδο μιας εταιρίας με τη δυνατότητα συμμετοχής από μέρους του πελάτη με σκοπό να εμπλακεί σε όλη τη διαδικασία της ανάπτυξης μιας εφαρμογής.

2.3.4 Προβλήματα

Τα προβλήματα σε αυτές τις διαδικασίες είναι σημαντικά ειδικά στην περίπτωση που εμπλέκονται τρίτα άτομα που δεν γνωρίζουν πολλά πράγματα πάνω στην ανάπτυξη λογισμικού. Κάποια από τα προβλήματα των διαδικασιών αυτών είναι:

- Δύσκολη διατήρηση ενδιαφέροντος των πελατών στη διαδικασία.
- Η απλότητα της διαδικασίας απαιτεί επιπλέον εργασία πάνω στην ανάπτυξη λογισμικού.
- Οι συμβάσεις που γίνονται πριν την ανάπτυξη του λογισμικού μπορεί να αποτελέσουν πρόβλημα κατά την ανάπτυξη της εφαρμογής.

(Somerville, 2011)

2.4 Ακραίος προγραμματισμός

Ο ακραίος προγραμματισμός εντάσσεται στις ευέλικτες τεχνικές και είναι μια από τις πιο διαδεδομένες τεχνικές. Εν συντομία κάποιες λεπτομέρειες για τον ακραίο προγραμματισμό είναι:

- Δόμηση πολλών νέων καθημερινών εκδόσεων του λογισμικού.
- Ανάπτυξη και παράδοση μικρών επαυξήσεων λειτουργικών δυνατοτήτων της εφαρμογής που είναι στη διαδικασία της ανάπτυξης.
- Βασίζεται στη σταθερή βελτίωση του κώδικα
- Συμμετοχή του πελάτη συνεπάγεται την πλήρη απασχόληση του στην ομάδα ανάπτυξης του λογισμικού.
- Για κάθε δόμηση που δημιουργείτε πρέπει να εκτελεστούν όλες οι δοκιμές, διαφορετικά δεν γίνεται αποδεκτή.

(Παπαδόπουλος, χ.χ.)

2.4.1 12 πρακτικές ακραίου προγραμματισμού

Παρακάτω θα αναφέρουμε τις 12 πρακτικές από τις οποίες απαρτίζεται ο ακραίος προγραμματισμός.

1. Το παιχνίδι του σχεδιασμού
2. Μικρές εκδόσεις
3. Αρχιτεκτονική εικόνα
4. Απλή σχεδίαση
5. Έλεγχοι πριν την κωδικοποίηση
6. Ανακατασκευή κώδικα
7. Προγραμματισμός ανά ζεύγη
8. Συλλογική ιδιοκτησία κώδικα
9. Διαρκείς ενοποιήσεις του κώδικα
10. Υποφερτός ρυθμός εργασίας
11. Διαρκής παρουσία πελάτη
12. Πρότυπα κωδικοποίησης

Το παιχνίδι του σχεδιασμού  =συμβολή πελάτη	Προγραμματισμός ανά ζεύγη  =λιγότερα λάθη στον κώδικα	Έλεγχος πριν την κωδικοποίηση  =ποιοτικό λογισμικό	Ανακατασκευή κώδικα  =καθαρός κώδικας
Σταθερές κωδικοποίησης  =καλύτερη επικοινωνία	Απλή σχεδίαση  =ευελιξία	Μικρές εκδόσεις  =προσαρμοστικότητα	Διαρκείς ενοποιήσεις του κώδικα  =διαρκής ανάδραση
Συλλογική ιδιοκτησία κώδικα  =διαμοιρασμός γνώσης	Διακρή παρουσία πελάτη  =ξεκαθάρισμα απαιτήσεων	Αρχιτεκτονική εικόνα  =κατανόηση συστήματος	Υποφερτός ρυθμός εργασίας  =αποδοτικότητα

Εικόνα 2.4.1: 12 πρακτικές ακραίου προγραμματισμού.

Τέλος ο ακραίος προγραμματισμός θεωρείται μια πειθαρχημένη προσέγγιση που πρέπει να πληροί κάποια πρότυπα όπως:

- Δημιουργία των test πριν τη δημιουργία του κώδικα.
- Προγραμματισμός σε ζεύγη
- Ολοκληρώνετε τακτικά τον κώδικα.
- Ξεκούραση προγραμματιστών.
- Επικοινωνία των προγραμματιστών με τους πελάτες στον ίδιο χώρο εργασίας.
- Καθαρό και απλό λογισμικό πρέπει να παραδίδεται στο τέλος της μέρας.

2.4.2 Γιατί ακραίος;

Ο χαρακτηρισμός ακραίος δίνεται στην παραπάνω τεχνική γιατί γίνονται:

- Συνεχής επιθεωρήσεις
- Συνεχής έλεγχος
- Καλός ανασχεδιασμός
- Όσο το δυνατόν πιο απλούστερη εφαρμογή
- Συνεχή ανατροφοδότηση

(Παναγιώτης, χ.χ.)

(Ιωάννης, χ.χ.)

3 Ανάπτυξη εφαρμογών διαδικτύου

3.1 Εισαγωγή

Αφού έχει προηγηθεί όλη αυτή η ανάλυση της εφαρμογής με κάποιο μοντέλο, παραδείγματος χάρη το μοντέλο του καταρράκτη και έχει γίνει η σχεδίαση με κάποιο πρόγραμμα στον υπολογιστή ή με τον παραδοσιακό τρόπο στο χαρτί, το επόμενο μας βήμα είναι η ανάπτυξη του.

Σε αυτό το βήμα λοιπόν η εφαρμογή μας θα αρχίσει να παίρνει μορφή με τη βοήθεια ορισμένων εργαλείων. Με αυτά τα εργαλεία θα ξεκινήσουμε να γράφουμε τον κώδικα της εφαρμογής μας.

Η διαδικασία σύνταξης ενός προγράμματος ονομάζεται προγραμματισμός. Ένα από τα κύρια προγράμματα που χρησιμοποιεί ένας προγραμματιστής είναι ο συντάκτης κειμένου(editor).

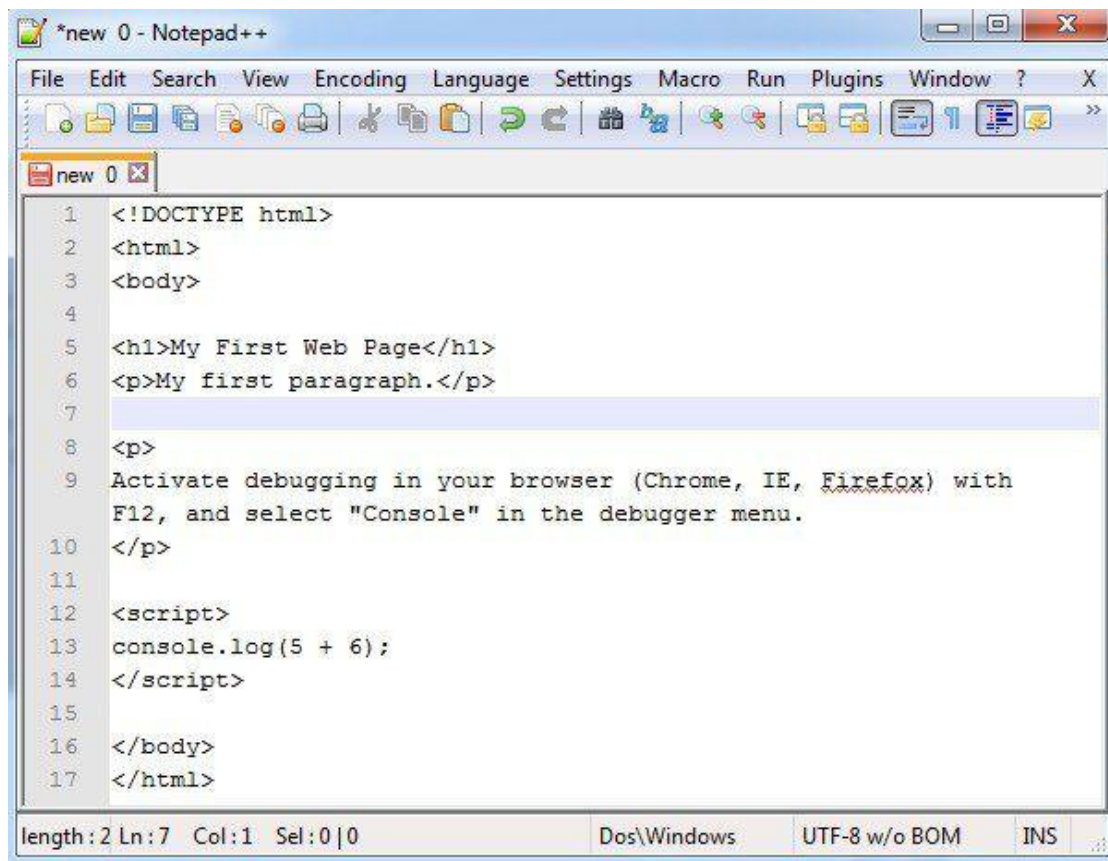
3.2 Βασικά εργαλεία ανάπτυξης λογισμικού

Σε πρώτη φάση για να ξεκινήσει η ανάπτυξη μιας εφαρμογής πρέπει γίνει επιλογή ενός συντάκτη κειμένου(editor). Όπως προδίδει και ο τίτλος θα κάνουμε αναφορά σε κάποιους απλούς συντάκτες κειμένου οι οποίοι μας βοηθούν μόνο στη συγγραφή του κώδικα και όχι στην απασφαλμάτωση του όπως θα δούμε στη συνέχεια με τα εξειδικευμένα εργαλεία.

Τα περισσότερα εργαλεία με τα οποία γράφουμε κώδικα συνήθως είναι opensource. Παρακάτω θα αναφέρουμε κάποια από αυτά με κάποιες λεπτομέρειες για το καθένα.

- Notepad++

Ένας συντάκτης κειμένου αρκετά απλός με τον οποίο όλοι γράψαμε τις πρώτες μας γραμμές είναι το Notepad++. Το Notepad++ έκανε την πρώτη του εμφάνιση στο διαδίκτυο τον Νοέμβριο του 2003 και είναι διαθέσιμο μόνο για το λειτουργικό σύστημα Windows, είναι γραμμένο σε γλώσσα προγραμματισμού C++. Ο συντάκτης κειμένου διατίθενται προς λήψη στον ιστότοπο <https://notepad-plus-plus.org>.

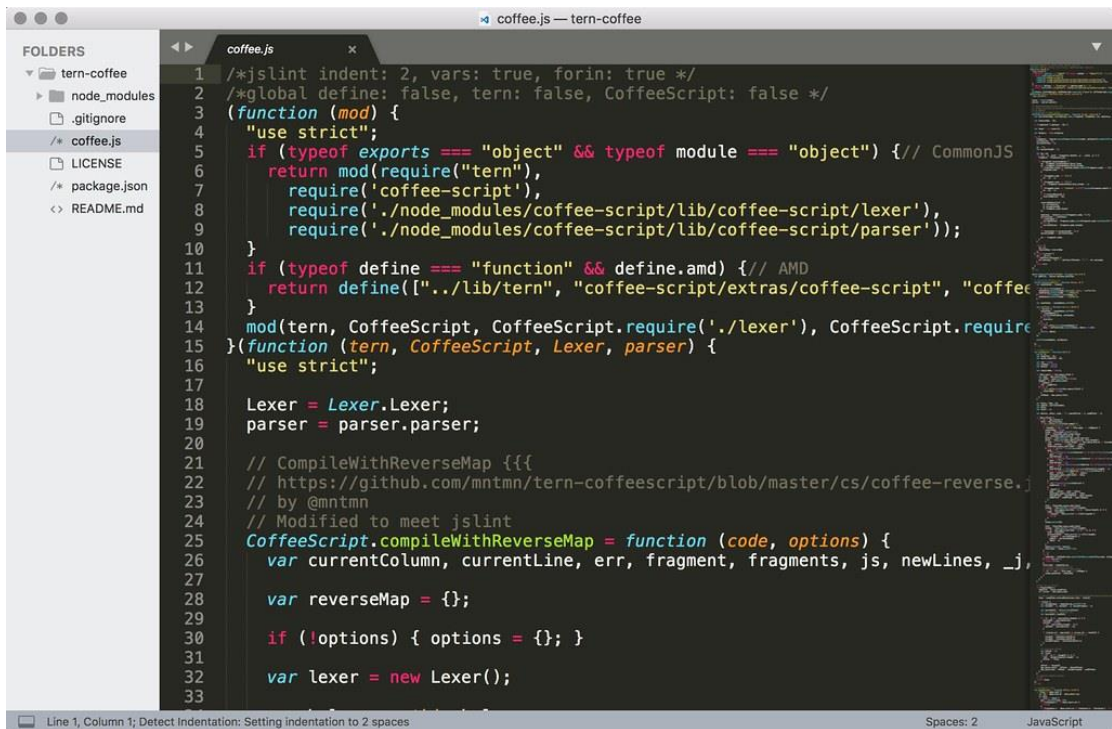


Εικόνα 3.2.1: Γραφικό περιβάλλον Notepad++.

- Sublime

Ένα άλλο πρόγραμμα συγγραφής κώδικα είναι το Sublime (<https://www.sublimetext.com>). Αναλυτικότερα, το sublime έκανε την εμφάνιση του τον Ιανουάριο του 2008 και είναι διαθέσιμο για τα λειτουργικά συστήματα: Windows, Linux και Mac OSX.

Το συγκεκριμένο πρόγραμμα υποστηρίζει αρκετές γλώσσες προγραμματισμού και γλώσσες σήμανσης, το δυνατό του προτέρημα σε σχέση με το Notepad++ είναι πως πλήθος Plugin έχουν αναπτυχθεί από τρίτους και είναι έτοιμα για να προσθέσουν περισσότερες δυνατότητες στο sublime όπως επίσης και η διάθεση κάποιων έτοιμων snippet(κομμάτια έτοιμου κώδικα τα οποία χρησιμοποιούμε συχνά). Γενικά θα λέγαμε πως το sublime αποτελεί μια πολύ καλή και ολοκληρωμένη λύση στην κατηγορία των απλών κειμενογράφων καθώς είναι πλήρως παραμετροποιήσιμο. Τέλος το sublime παρέχεται δωρεάν, στην περίπτωση που το λογισμικό χρησιμοποιηθεί μακροπρόθεσμα πρέπει να αγοραστεί άδεια χρήσης χωρίς όμως να υπάρχει κάποιος περιορισμός στην περίπτωση που δεν γίνει η αγορά κάποιας άδειας.

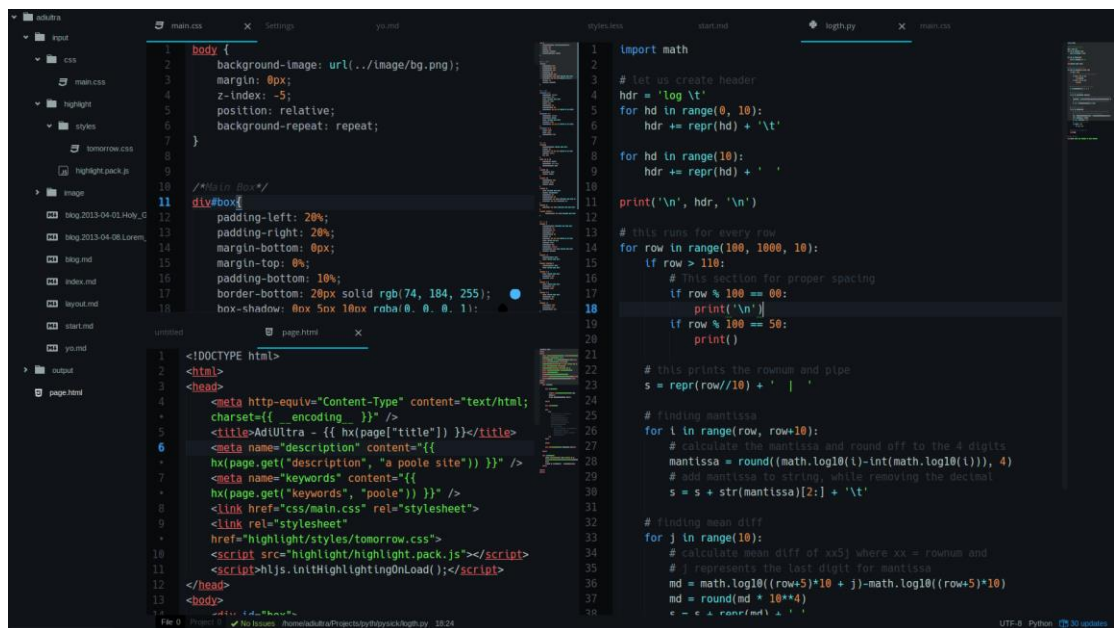


Εικόνα 3.2.2: Γραφικό περιβάλλον Sublime.

- Atom

Τέλος θα αναφερθούμε στο τελευταίο πρόγραμμα στην κατηγορία των απλών εργαλείων ανάπτυξης εφαρμογών στα πλαίσια της πτυχιακής εργασίας χωρίς όμως να εμβαθύνουμε περαιτέρω, καθώς υπάρχουν πάρα πολλά εργαλεία σε αυτήν την κατηγορία στο διαδίκτυο. Το τελευταίο εργαλείο είναι ο συντάκτης κειμένου Atom (<https://atom.io/>).

Ο συντάκτης Atom είναι ένα opensource πρόγραμμα το οποίο αναπτύχθηκε τον Αύγουστο του 2014. Ο Atom είναι γραμμένος σε Electron(back-end), CoffeeScript / JavaScript / Less / HTML (front-end/UI) και είναι διαθέσιμος για τα λειτουργικά συστήματα Windows, Linux και Mac OSX. Ο Atom διατίθεται εντελώς δωρεάν στους χρήστες και αυτό όπως το Sublime με πλήθος τρίτων κατασκευαστών Plugin για την προσθήκη περισσότερων λειτουργιών παραδείγματος χάρι FTP, Teletype(διαμοιρασμός του project με άλλους συνεργάτες), AutoSave κ.λπ.



Εικόνα 3.2.3: Γραφικό περιβάλλον Atom.

3.3 Εξειδικευμένα εργαλεία ανάπτυξης λογισμικού

Αφού γίνει η επιλογή του text editor που καλύπτει τις ανάγκες το επόμενο που έχουμε να κάνουμε είναι να επιλέξουμε τη γλώσσα προγραμματισμού ή τις γλώσσες προγραμματισμού που θα καλύψουν τις απαιτήσεις της εφαρμογής μας. Ένας βασικός συνδυασμός για την υλοποίηση μιας εφαρμογής για την κατασκευή του back-end είναι η PHP και για την κατασκευή του front-end είναι η HTML, CSS και JavaScript με μια βάση δεδομένων MySQL. Αυτός ο συνδυασμός μπορεί να υλοποιήσει μια ολοκληρωμένη εφαρμογή. Περισσότερες λεπτομέρειες θα δώσουμε σε επόμενο κεφάλαιο.

Εξειδικευμένα εργαλεία ανάπτυξης λογισμικού είναι και τα ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE) τα οποία μας βοηθούν στη συγγραφή του κώδικα. Είναι σαν αυτά που αναφέραμε στα βασικά εργαλεία (text editors) απλά είναι πιο «έξυπνα» και με περισσότερες δυνατότητες.

Στη συνέχεια θα αναφέρουμε κάποια από τα «Εξυπνα» εργαλεία καθώς και κάποιες λεπτομέρειες γι' αυτά.

- Visual Studio Code

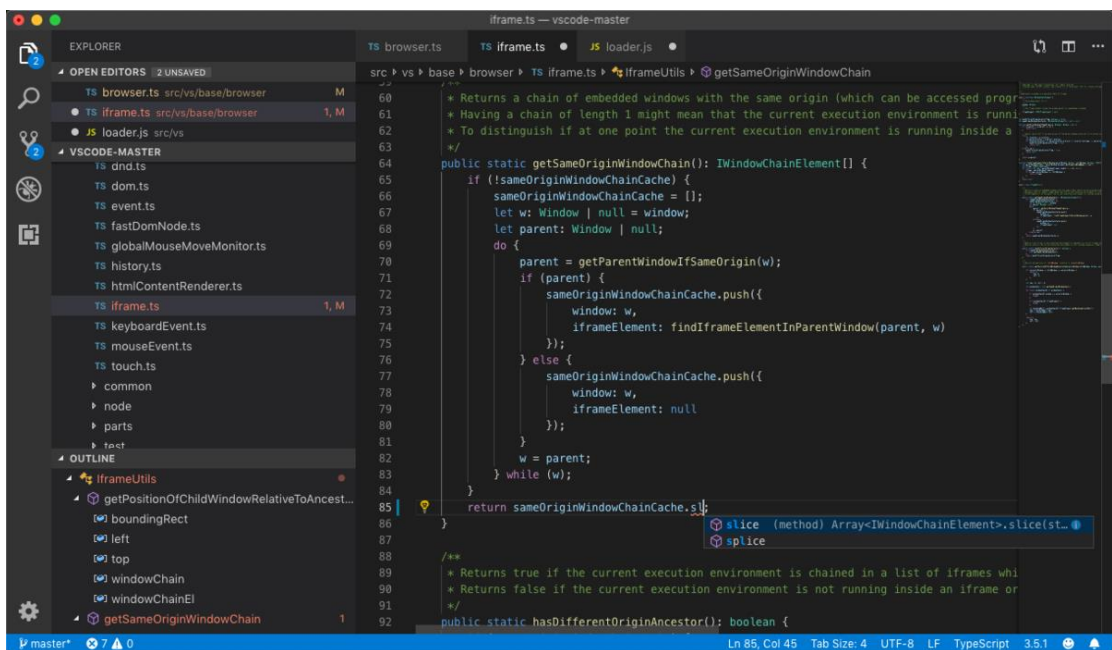
Το λογισμικό visual studio code (<https://visualstudio.microsoft.com>) έχει υλοποιηθεί από την Microsoft και συγκεκριμένα την πρώτη εμφάνιση στο διαδίκτυο την έκανε τον Απρίλιο του 2015. Αυτό το λογισμικό είναι διαθέσιμο για Windows, Linux, Mac OSX και είναι γραμμένο στις εξής γλώσσες, TypeScript, JavaScript και CSS.

Αναλυτικότερα αυτό το πρόγραμμα υποστηρίζει αν όχι όλες τις γλώσσες προγραμματισμού τις περισσότερες. Επίσης η εταιρία διαφημίζει κάποιες δυνατότητες όπως την αυτόματη συμπλήρωση(auto complete) και την επισήμανση της σύνταξης(syntax highlighting) με το όνομα IntelliSense.

Ακόμα ένα χαρακτηριστικό και ίσως το βασικότερο και αυτό γιατί το κάνει πιο «Έξυπνο» σε σχέση με τα άλλα βασικά εργαλεία είναι ότι με αυτής της κατηγορίας τα προγράμματα μπορείς να κάνεις debugging ή αλλιώς απασφαλμάτωση(μεθοδική διαδικασία εύρεσης και εξάλειψης σφαλμάτων κώδικα ενός προγράμματος υπολογιστή) χάρη στον ενσωματωμένο debugger που διαθέτει, μπορείς κατευθείαν να δεις τα συντακτικά λάθη που έχεις κάνει στον επεξεργαστή κειμένου.

Μία ακόμη αρκετά χρήσιμη δυνατότητα που έχει το συγκεκριμένο πρόγραμμα είναι η υποστήριξη Git Commands. Η συνεργασία με το Git είναι πλέον πιο εύκολη από ποτέ, το staging της εφαρμογής, push και το pull.

Τέλος υπάρχει μια βιβλιοθήκη με plugins από τρίτους κατασκευαστές η οποία μπορεί να ανταγωνιστεί ακόμα και τα προγράμματα που είναι επί πληρωμή.

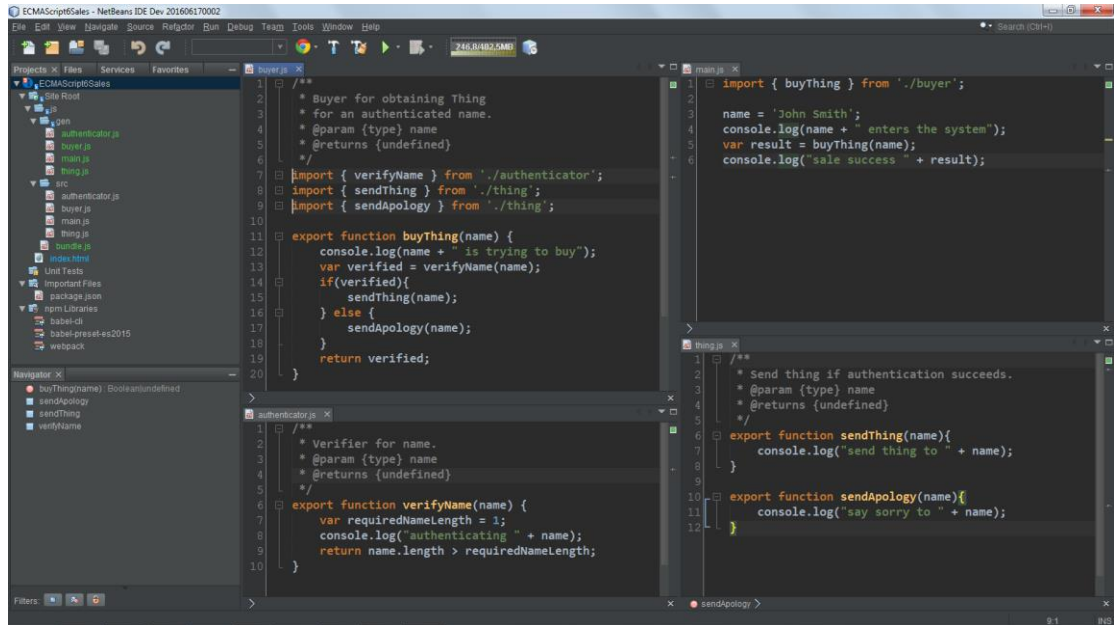


Εικόνα 3.3.1: Γραφικό περιβάλλον Visual Studio.

- NetBeans

Ακόμη ένα εργαλείο το οποίο δεν είναι ένας απλός κειμενογράφος είναι το NetBeans (<https://netbeans.org/>). Το NetBeans ξεκίνησε ο Roman Staněk το 1997 βγάζοντας στην παραγωγή τις πρώτες εμπορικές εκδόσεις του. Αρχικά το

πρόγραμμα κατασκευάστηκε για τη συγγραφή κώδικα σε Java. Έπειτα, μετά από κάποια χρόνια και αφού υπήρξαν σταδιακά διάφορες εκδόσεις του προγράμματος, το πρόγραμμα άρχισε να υποστηρίζει όλο και περισσότερες γλώσσες προγραμματισμού. Αξίζει να αναφέρουμε πως το πρόγραμμα είναι γραμμένο σε Java, είναι opensource και μπορεί να εγκατασταθεί στα λειτουργικά συστήματα Windows, Linux, Max OSX και Solaris.



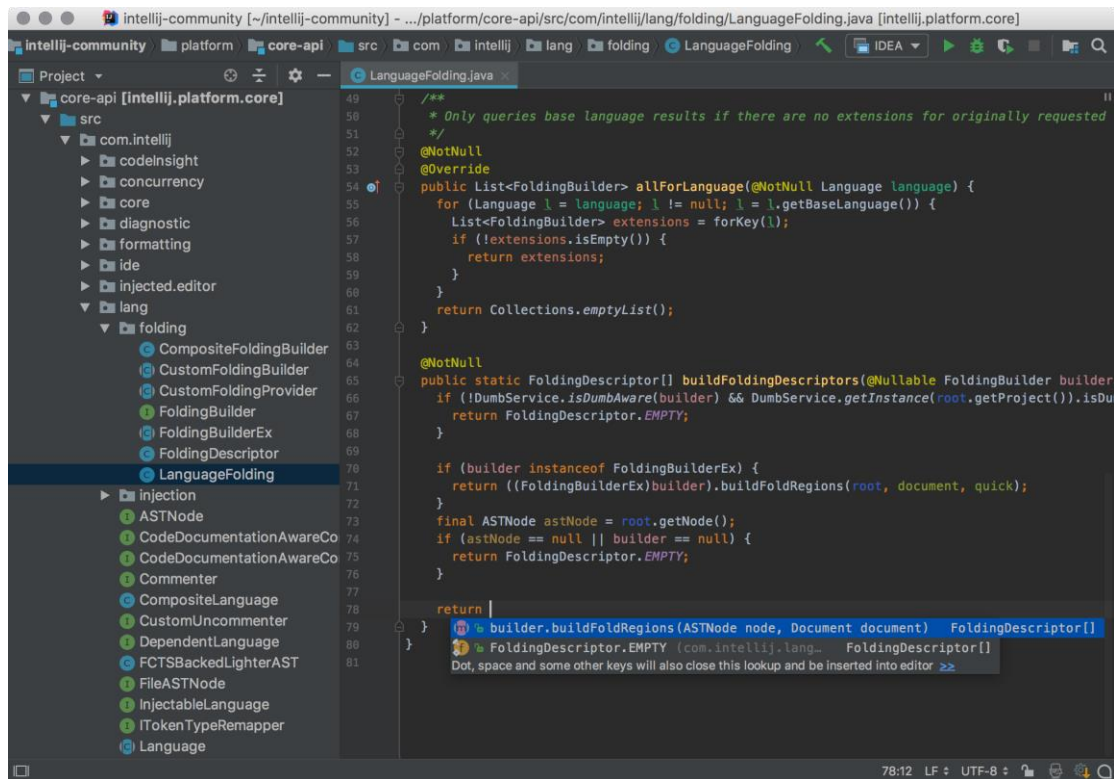
Εικόνα 3.3.2: Γραφικό περιβάλλον NetBeans.

- **JetBrains**

Η JetBrains (<https://www.jetbrains.com/>) είναι μια εταιρεία η οποία θα λέγαμε πως απευθύνεται σε εταιρίες ή σε επαγγελματίες καθώς πέρα από μια δωρεάν έκδοση πολύ «φτωχή», όλα τα υπόλοιπα προγράμματά της πως είναι επί πληρωμή. Ανάλογα σε ποια γλώσσα θα προγραμματίσεις αγοράζεις και το αντίστοιχο πρόγραμμα. Να πούμε πως ένα πρόγραμμα που υποστηρίζει μια γλώσσα προγραμματισμού θα δώσει στον χρήστη και κάποια άλλα εργαλεία έτσι ώστε να τον καλύψει πλήρως πάνω σε μια τεχνολογία, τα λεγόμενα bundles. Τα bundles είναι πακέτα τα οποία τα αγοράζεις και σου δίνουν ό,τι χρειάζεσαι για να καλύψεις όλο το εύρος απαιτήσεων μιας εφαρμογής.

Η JetBrains υποστηρίζει επίσης τις περισσότερες και τις νεότερες τεχνολογίες, γλώσσες προγραμματισμού και frameworks που υπάρχουν έξω στην αγορά. Δεν χρειάζεται να αναλύσουμε κάτι παραπάνω καθώς υποστηρίζει όλα τα παραπάνω που αναφέραμε στα προηγούμενα προγράμματα όπως debugging, Testing, Tracing and profiling και όλα αυτά μέσα από το περιβάλλον συγγραφής κώδικα.

Η JetBrains ιδρύθηκε τον Αύγουστο του 2000 και τα προγράμματά της είναι διαθέσιμα σε Windows, Linux και Mac OSX.



Εικόνα 3.3.3: Γραφικό περιβάλλον JetBrains.

3.4 Υπηρεσίες φιλοξενίας εφαρμογών στο διαδίκτυο

Αφού έχουμε τελειώσει με την υλοποίηση τοπικά στον υπολογιστή μας και έχουμε κάνει αρκετές δοκιμές τόσες ώστε να δούμε ότι η εφαρμογή μας λειτουργεί χωρίς κάποιο πρόβλημα, πρέπει να βρούμε μια υπηρεσία φιλοξενίας όπου θα ανεβάσουμε την εφαρμογή έτσι ώστε να υπάρχει πρόσβαση από όλο το διαδίκτυο.

Οι υπηρεσίες φιλοξενίας όπου κάποιες από αυτές είναι επί πληρωμή και άλλες δωρεάν μπορούν να φιλοξενούν και να «σερβίρουν» την εφαρμογή μας σε όποιον θέλει να τη χρησιμοποιήσει. Αυτήν τη διαδικασία θα μπορούσαμε να την αποφύγουμε εάν αποφασίζαμε ότι κάποιο μηχάνημα τοπικά θα «σερβίρει» την εφαρμογή μας αλλά εδώ δημιουργούνται άλλα θέματα όπως:

1. Την ταχύτητα που «σερβίρει» την εφαρμογή καθώς πρέπει να έχουμε πολύ καλή ταχύτητα upload στον χώρο όπου είναι ο υπολογιστής με την εφαρμογή.
2. Την ασφάλεια.
3. Την υπολογιστική ισχύ του μηχανήματος μας, δηλαδή εάν είναι ικανό σε ώρες αιχμής να «σερβίρει» σε πολλά άτομα ταυτόχρονα την εφαρμογή μας.

Παρακάτω θα αναφέρουμε τρεις διαφορετικές υπηρεσίες(Heroku, Firebase, Namecheap) όπου η κάθε μια θα φιλοξενήσει διαφορετική τεχνολογία. Έτσι θα καλύψουμε τρεις διαφορετικές περιπτώσεις εφαρμογών. Αναλυτικότερα:

- Heroku

Η Heroku (<https://www.heroku.com/>) είναι μια από τις πρώτες πλατφόρμες cloud που υποστηρίζει πολλές γλώσσες προγραμματισμού. Η εταιρεία ιδρύθηκε το 2007 η οποία υποστήριζε μόνο μια γλώσσα προγραμματισμού την Ruby, με τον καιρό είχε διαθέσιμες προς φιλοξενία και εφαρμογές με τις εξής γλώσσες προγραμματισμού: Java, Node.js, Scala, Clojure, Python, PHP και Go. Η Heroku θεωρείται μια πολύγλωσση πλατφόρμα καθώς υποστηρίζει όλες αυτές τις γλώσσες.

Η υπηρεσία έχει διάφορες επιλογές ως προς τα πακέτα τιμολόγησης που μπορεί να χρησιμοποιήσει ένας χρήστης, ανάλογα με τις απαιτήσεις της εφαρμογής καθώς επίσης και ένα δωρεάν πακέτο για τα πρώτα βήματα ενός προγραμματιστή.

Σε περισσότερες λεπτομέρειες σχετικά με την υπηρεσία πέρα από τις γλώσσες προγραμματισμού που υποστηρίζει η Heroku δίνει τη δυνατότητα συνεργασίας με Git Commands από CLI έτσι ώστε να κάνουμε deploy την εφαρμογή μας κατευθείαν από το GitHub. Στη συνέχεια ο προγραμματιστής μπορεί να επεκτείνει τις εφαρμογές του με τα Addons που έχει διαθέσιμα από τρίτους σχετικά με τα datastores, logging, monitoring. Η Heroku παρέχει τρία πλήρως ελεγχόμενα πρόσθετα υπηρεσίας δεδομένων:

1. Heroku Postgres
2. Heroku Redis
3. Apache Kafka

- Firebase

Η Firebase (<https://firebase.google.com/>) είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητά και ιστοσελίδες όπου ιδρύθηκε το 2011 και εξαγοράστηκε από την Google το 2014. Η συγκεκριμένη πλατφόρμα παρέχει πληθώρα υπηρεσιών στους υποψήφιους πελάτες της με ένα δωρεάν πλάνο και δύο πιο επαγγελματικά επί πληρωμή.

Η υπηρεσίες που διαθέτει η πλατφόρμα είναι:

1. Firebase Analytics
Είναι μια υπηρεσία η οποία παρέχεται δωρεάν και έχει παρόλα αυτά τη δυνατότητα η έκδοση αυτή να μετράει την επισκεψιμότητα μιας ιστοσελίδας.
2. Firebase Cloud Messaging

Είναι η υπηρεσία που παρέχει ειδοποιήσεις σε Android, IOS και σε διαδικτυακές εφαρμογές χωρίς κόστος από το 2016.

3. Firebase Auth

Η συγκεκριμένη υπηρεσία παρέχει αυθεντικοποίηση και είναι ανεξάρτητη από την εφαρμογή στην οποία χρησιμοποιείτε. Αποτελείτε από μια βάση δεδομένων με τους χρήστες και δίνει τη δυνατότητα σε μια εφαρμογή ή μια ιστοσελίδα να έχει ένα απλό σύστημα αυθεντικοποίησης. Ο συγκεκριμένος συνδυασμός χρησιμοποιείται από τις παρακάτω εταιρίες:

- Facebook
- GitHub
- Twitter
- Google

4. Firebase Realtime Database

Αποτελεί μια βάση δεδομένων σε πραγματικό χρόνο η οποία είναι μια αποτελεσματική λύση με χαμηλό χρόνο απόκρισης για εφαρμογές σε κινητά που απαιτούν συγχρονισμό κατάστασης σε clients σε πραγματικό χρόνο.

5. Firebase Cloud Firestore

Υπηρεσία αποθήκευσης και συγχρονισμού δεδομένων μεταξύ χρηστών και συσκευών. Η συγκεκριμένη αποτελεί μια βάση δεδομένων NoSQL που φιλοξενείται στο cloud και προσφέρει ζωντανό συγχρονισμό και υποστήριξη εκτός σύνδεσης με αποτελεσματικά data queries(Ερωτήματα). Η ενοποίηση του με άλλα προϊόντα της συγκεκριμένης πλατφόρμας δίνει τη δυνατότητα στο χρήστη να δημιουργεί πραγματικά εφαρμογές χωρίς διακομιστές.

6. Firebase Storage

Υπηρεσία αποθήκευσης και διαμοιρασμού περιεχομένου το οποίο δημιουργείται από χρήστες παραδείγματος χάρη εικόνα, ήχος και βίντεο με οικονομική και αποδοτική αποθήκευση η οποία έχει σχεδιαστεί από την Google. Επί πρόσθετα παρέχει ασφάλεια στα αρχεία ανεξάρτητα με την ποιότητα που έχει το δίκτυο στο οποίο πάει η συσκευή να κατεβάσει ένα αρχείο.

7. Firebase Hosting

Η φιλοξενία μιας σελίδας δεν ήταν ποτέ πιο απλή με τη συγκεκριμένη υπηρεσία. Ανεβάζοντας τα αρχεία της σελίδας στην υπηρεσία αυτόματα αποθηκεύονται στο global CDN έτσι ώστε να «σερβίρετε» το περιεχόμενο στους χρήστες ταχύτατα. Δεν θα μπορούσε να λείπει εν έτη 2019 η παροχή δωρεάν πιστοποιητικού SSL ώστε οι χρήστες να έχουν μια ασφαλή και αξιόπιστη εμπειρία περιήγησης.

8. ML KIT

Μια δυνατότητα που δεν περιέχεται σε άλλες αντίστοιχες εταιρίες είναι η δυναμική δυνατότητα εκμάθησης μηχανών στις εφαρμογές των κινητών. Ακόμα και ένας αρχάριος χρήστης πάνω σε αυτήν την τεχνολογία μπορεί να ξεκινήσει εύκολα χρησιμοποιώντας τα έτοιμα για χρήση API. Πέρα από των έτοιμων μοντέλων δίνεται η δυνατότητα στον χρήστη να προσθέσει και τα δικά του.

Αξίζει να πούμε πως παρέχει και κάποια εργαλεία για τη βοήθεια βελτίωσης των εφαρμογών αλλά δεν θα αναφέρουμε κάτι περισσότερο στη συγκεκριμένη πτυχιακή.

- **Namecheap**

Είναι μια εταιρία η οποία παρέχει όλες τις υπηρεσίες από την αγορά ονόματος(Domain) μιας σελίδας μέχρι τη φιλοξενία(Hosting), DNS Service, Reseller Hosting, VPS Hosting, Dedicated Server, κ.α. Τα δύο πρώτα είναι τα βασικότερα εργαλεία που χρειαζόμαστε για να βγάλουμε στην παραγωγή μια εφαρμογή διαδικτύου.

Άλλες παρόμοιες υπηρεσίες: mediahost, a2hosting, top.host, dnhost και άλλες.

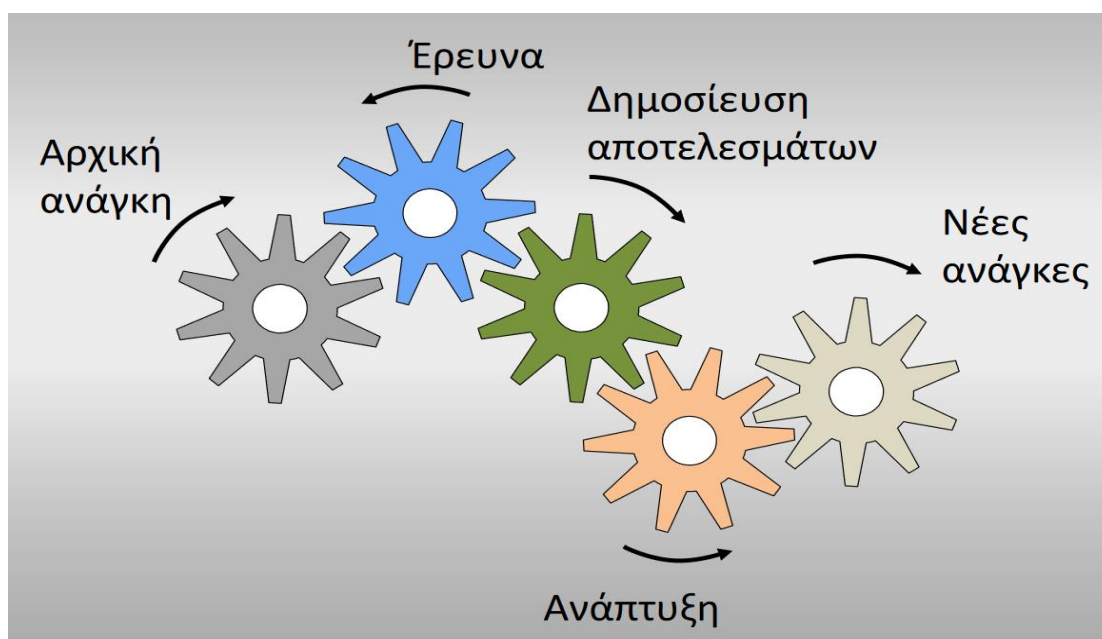
4 Έρευνα και επιστημονικά άρθρα

4.1 Εισαγωγή

Στις μέρες μας το ίντερνετ είναι το πιο διαδεδομένο μέσο επικοινωνίας, ενημέρωσης και διασκέδασης. Το ίντερνετ είναι ένα μέρος που ο καθένας μπορεί να ανεβάσει το περιεχόμενο του με σκοπό την προβολή του από άλλους χρήστες. Πολλές φορές θα λέγαμε πως δεν είναι και η πιο αξιόπιστη πηγή τουλάχιστον στον τομέα της ενημέρωσης. Η ενημέρωση είναι ένα σημαντικό κομμάτι της καθημερινότητας μας. Με βάση αυτή, μπορεί να πάρει ένας άνθρωπος κάποιες αποφάσεις μακροπρόθεσμα ή βραχυπρόθεσμα. Στις μέρες μας κυκλοφορούν πολλά «Fake news» απλά και μόνο για την αύξηση της επισκεψιμότητας μιας σελίδας. Σε αυτό το κεφάλαιο γίνεται αναφορά σε κάποια εργαλεία τα οποία χρησιμοποιούνται για τη σωστή ενημέρωση μας πάνω σε διάφορα θέματα, πρόσφατες μελέτες και γενικότερα σε διάφορες έρευνες που γίνονται στον κόσμο.

4.2 Επιστημονικά άρθρα

Τα επιστημονικά άρθρα είναι άρθρα τα οποία γράφτηκαν από κάποιον επιστήμονα, μια ομάδα επιστημόνων οι οποίοι ύστερα από μια έρευνα που έκαναν θέλουν να ενημερώσουν τους ενδιαφερόμενους πάνω σε κάποιο θέμα που τους ενδιαφέρει ανεβάζοντας σε μια πλατφόρμα αξιόπιστη τα συμπεράσματά τους.



Εικόνα 4.2.1: Στάδια ενός επιστημονικού έργου. (Διέτης, 2017)

Ανεβάζοντας τα συμπεράσματά τους πέρα από την ενημέρωση κάποιου χρήστη μπορεί να βοηθήσουν έναν επιστήμονα να συνεχίσει την έρευνα, όχι από το μηδέν αλλά από εκεί που την άφησαν οι προηγούμενοι κερδίζοντας έτσι πολύτιμο χρόνο.

Υπάρχουν 2 ειδών άρθρα, αυτά είναι:

- 1) Πλήρη άρθρα (Full articles): Αποτελούν τα πιο σημαντικά άρθρα, είναι τα πιο μακροσκελή(3000-5000 λέξεις) σε σχέση με τις άλλες κατηγορίες και περιγράφουν τα αποτελέσματα της μιας πρωτότυπης έρευνας.
- 2) Review papers: Η συγκεκριμένη κατηγορία δεν παρουσιάζει νέες πληροφορίες αλλά κάνει αναφορά στις πρόσφατες εξελίξεις πάνω σε ένα θέμα.

(Ανδρέας Φλούρης, n.d.)

4.3 Πεδία ενδιαφέροντος επιστημονικών άρθρων

Ένα επιστημονικό άρθρο για να θεωρείται έγκυρο πρέπει να ακολουθεί ένα συγκεκριμένο επιστημονικό πρωτόκολλο και δομή. Εν συντομία πέρα από τα γενικά στοιχεία στα οποία δεν θα κάνουμε αναφορά καθώς θα βγούμε εκτός θέματος. Θα αναφέρουμε παρακάτω τη δομή από την οποία αποτελείται ένα επιστημονικό άρθρο.

- Τίτλο
- Περίληψη
- Λέξεις κλειδιά
- Εισαγωγή
- Μέθοδος
- Αποτελέσματα
- Συζήτηση/Προβληματισμοί
- Συμπεράσματα
- Βιβλιογραφία

(Αθανασόπουλος, 2013)

Για την ανάπτυξη της εφαρμογής την οποία θα αναφερθούμε αναλυτικότερα στα παρακάτω κεφάλαια θα χρειαστούμε κάποιες πληροφορίες για το κάθε άρθρο η οποίες θα είναι αποθηκευμένες στη βάση δεδομένων. Αυτές οι πληροφορίες θα λέγαμε πως είναι τα πεδία ενδιαφέροντος επιστημονικών άρθρων τις εφαρμογής μας. Παρακάτω αναφέρουμε ποιες είναι αυτές:

- Τίτλος
- Ημερομηνία δημοσίευσης
- Τύπος του άρθρου
- Αναφορές

4.4 Πλατφόρμα καταλογογράφησης επιστημονικών άρθρων

Στο διαδίκτυο μπορεί να βρει κανείς πολλές μηχανές αναζήτησης επιστημονικών άρθρων. Άλλες η οποίες είναι εξιδεικευμένες πάνω σε έναν τομέα και άλλες στοχεύουν σε ένα ευρύ φάσμα κατηγοριών επιστημονικών άρθρων και άλλες μηχανές στις οποίες μπορείς να βρεις όλα τα επιστημονικά άρθρα τα οποία είναι σε ηλεκτρονική μορφή στο διαδίκτυο. Κάποιες δημοφιλής μηχανές αναζήτησης είναι:

- Openarchives.gr
- Heal-link.gr (Σύνδεσμο Ελληνικών Ακαδημαϊκών Βιβλιοθηκών)
- unioncatalog.gr/ucportal/ (Συλλογικός κατάλογος ελληνικών ακαδημαϊκών βιβλιοθηκών)
- Library.panteion.gr (Ανθρωποκοινωνικά)
- Ipl.org (Internet Public Library)
- Doabooks.org (Directory of open access book)
- ncbi.nlm.nih.gov/pubmed (Εθνικής Ιατρικής Βιβλιοθήκης των Η.Π.Α)

4.4.1 Google Scholar

Η μηχανή αναζήτησης Google Scholar ιδρύθηκε τον Νοέμβριο του 2004 και αποτελεί μια από τις μεγαλύτερες βάσεις δεδομένων επιστημονικού έργου η οποία πέρα από το επιστημονικό έργο περιέχει και κάποια στατιστικά στοιχεία για κάθε ένα έργο ξεχωριστά. Όταν λέμε στατιστικά στοιχεία εννοούμε ότι για κάθε έργο μπορούμε να δούμε πόσες αναφορές έχουν γίνει σε αυτό, ποιος είναι ο ερευνητής ή η ομάδα ερευνητών που κατέληξε σε κάποια αποτελέσματα μέσα από μια έρευνα. Το Google Scholar μας δίνει τη δυνατότητα να δημιουργήσουμε γραφήματα βάση του χρόνου για ένα άρθρο έτσι ώστε να βλέπουμε την απήχηση που έχει ένα άρθρο από την ημέρα που ανέβηκε στην πλατφόρμα μέχρι και σήμερα καθώς και κάποια άλλα στατιστικά στοιχεία τα οποία θα δούμε στη συνέχεια.

4.4.2 Academia Edu

Η Academia Edu έκανε την πρώτη της εμφάνιση στον διαδικτυακό χώρο τον Σεπτέμβριο του 2008. Academia Edu είναι μια εμπορική μηχανή αναζήτησης επιστημονικών άρθρων με σκοπό τη δικτύωση των ακαδημαϊκών ερευνητών και φοιτητών που συμμετέχουν σε μια έρευνα. Η συγκεκριμένη μηχανή για να τη χρησιμοποιήσει κάποιος πρέπει να κάνει εγγραφή. Κατά τη διάρκεια της εγγραφής και αφού δώσουμε τα στοιχεία σχετικά με το ακαδημαϊκό επίπεδο που βρισκόμαστε πρέπει να επιλέξουμε ανάμεσα σε δύο πλάνα, το δωρεάν και το επί πληρωμή. Παρακάτω θα αναφέρουμε τις δυνατότητες που μας δίνονται κατά την επιλογή ενός πλάνου. Η αναφορά θα γίνει όπως ακριβώς αναγράφονται στην ιστοσελίδα.

Free:

- Βρείτε, κατεβάστε και διαβάστε έγγραφα συνολικού κειμένου 23m
- Μοιραστείτε την έρευνά σας με το προφίλ Academia
- Μετρήστε τον αντίκτυπό σας με βασικά στοιχεία ανάλυσης

Premium:

- Βρείτε, κατεβάστε και διαβάστε έγγραφα συνολικού κειμένου 23m
- Μοιραστείτε την έρευνά σας με το προφίλ Academia
- Μετρήστε τον αντίκτυπό σας με βασικά στοιχεία ανάλυσης
- Μάθετε για τους αναγνώστες των εγγράφων σας
- Δείτε ποιος σας αναφέρει και αναφέρει τα έγγραφά σας
- Αποκτήστε βελτιωμένους επισκέπτες και προφίλ επισκεπτών
- Παρουσιάστε τη δουλειά σας με έναν προσωπικό ιστότοπο
- Αναζητήστε το πλήρες κείμενο και τις αναφορές των εγγράφων των 23m

Τέλος θα αναφέρουμε πως τον Ιανουάριο του 2019 κατέγραψε 39 εκατομμύρια μοναδικούς επισκέπτες και 21 εκατομμύρια αποσταλμένα αρχεία στην υπηρεσία.

4.4.3 Microsoft Academic

Η υπηρεσία Microsoft Academic αποτελεί και αυτή μια μηχανή αναζήτησης ιστού για ακαδημαϊκά άρθρα και λογοτεχνία. Ιδρύθηκε τον Φεβρουάριο του 2016 και αναπτύχθηκε από την Microsoft Research με τεχνολογίες Semantic search. Από την ημέρα ίδρυσης η Microsoft Academic μετράει 220 εκατομμύρια δημοσιεύσεις και τα 88 εκατομμύρια αποτελούν άρθρα περιοδικών.

Αξίζει να αναφέρουμε πως η συγκεκριμένη υπηρεσία αντικαθιστά το παλιό ερευνητικό πρόγραμμα Microsoft Academic Search.

4.4.4 Semantic Scholar

Semantic Scholar αποτελεί και αυτή μια μηχανή αναζήτησης επιστημονικών άρθρων η οποία είναι καινούρια στον χώρο σε σχέση με τις προηγούμενες που αναφέραμε. Ιδρύθηκε τον Νοέμβριο του 2015, υλοποιήθηκε στο Ινστιτούτο τεχνητής νοημοσύνης του Allen (AI2) και αποτελεί μια ελεύθερη μη κερδοσκοπική ακαδημαϊκή μηχανή αναζήτησης επιστημονικών έργων. Η συγκεκριμένη μηχανή έχει αναπτυχθεί σε συνδυασμό με μηχανική μάθηση (machine learning), επεξεργασίας φυσικής γλώσσας (natural language processing) και μηχανικής όρασης (machine vision).

Η συγκεκριμένη μηχανή αναζήτησης αναπτύχθηκε με σκοπό να επισημάνει τα σημαντικότερα άρθρα και να προσδιορίσει τις συνδέσεις μεταξύ τους σε σχέση με το Google Scholar.

Από την ημερομηνία ίδρυσης μέχρι τον Αύγουστο του 2019 μετράει 173 εκατομμύρια αρχεία μαζί με τα αρχεία από την Microsoft Academic Graph.

4.5 Αξιολόγηση επιστημονικού έργου

4.5.1 Αναφορές

Ως αναφορά ορίζεται μια ενέργεια δημόσιας γνωστοποίησης ενός συμβάντος. Μια αναφορά έχει σκοπό να γίνει γνωστό στον εκάστοτε αποδέκτη που στηρίζεται και σε τι αναφέρεται ένα επιστημονικό έργο ή ένα κομμάτι αυτού. Κάθε αναφορά σε ένα άλλο επιστημονικό έργο ή ένα οποιοδήποτε κείμενο το οποίο έχει χρησιμοποιηθεί πρέπει να αναφέρεται.

Οι αναφορές εκτός ότι διασφαλίζουν τα πνευματικά δικαιώματα ενός συγγραφέα, αποτελούν ένα μέσο αξιολόγησης ενός επιστημονικού έργου. Μέσω των αναφορών αναδεικνύουμε την ποιότητα και την αναγνώριση ενός επιστημονικού έργου.

4.5.2 Μετρικές

Είναι διάφορες μετρικές των αναφορών ενός επιστήμονα, δηλαδή δείχνουν την απήχηση ενός επιστημονικού άρθρου του συγγραφέα. Impact factor (Βαθμός επιρροής) ενός συγγραφέα υπολογίζεται από τη διαίρεση του συνολικού αριθμού των αναφορών του στις εργασίες ενός περιοδικού που έγιναν τα τελευταία δύο χρόνια προς τον συνολικό αριθμό εργασιών που δημοσιεύθηκαν από το περιοδικό. Το αποτέλεσμα από αυτή τη μαθηματική πράξη που προκύπτει είναι διαθέσιμο από το Journal Citation Report (JCR) (Μοναδική βάση δεδομένων από το Clarivate Analytics των πιο αναφερόμενων περιοδικών σε συγκεκριμένους τομείς). Τα στοιχεία των αναφορών της JCR είναι διαθέσιμα στο κοινό κάθε χρόνο σε δύο διαφορετικές κατηγορίες:

- JCR Science, αφορά περιοδικά στις φυσικές και εφαρμοσμένες επιστήμες, στις επιστήμες υγείας και τεχνολογίας.
- JCR Social Sciences, αφορά οικονομικές, κοινωνικές και ανθρωπιστικές επιστήμες.

4.5.3 Οι μετρικές h-index, i10 κ.α

Οι παρακάτω μετρικές αφορούν τους συγγραφείς και είναι διαθέσιμες από το Microsoft Academic και το Google Scholar:

- Δείκτης h (h -index) του Hirsch, είναι ένας από τους σημαντικότερους δείκτες που έχει στόχο στη μέτρηση της ποιότητας και απήχησης των ακαδημαϊκών δημοσιεύσεων ενός επιστήμονα. Δηλαδή κατά πόσο ποιοτικά είναι τα επιστημονικά άρθρα και πόσο παραγωγικός είναι ένας συγγραφέας. Για παράδειγμα, ένας επιστήμονας με δείκτη $h=50$ σημαίνει ότι έχει δημοσιεύσει μεταξύ άλλων, 50 επιστημονικά άρθρα, τα οποία το καθένα έχει 50 ή περισσότερες αναφορές από άλλους επιστήμονες.
- Δείκτης g (g -index) του Egghe, επιδιώκει να βελτιώσει τον δείκτη h , δίνοντας μεγαλύτερη βαρύτητα στα ιδιαίτερα αναφερόμενα άρθρα.
- Ο δείκτης $i10$ μας δείχνει τον αριθμό των άρθρων που έχουν 10 παραθέσεις τουλάχιστον.

(Anon., n.d.)

5 Λογισμικά επιστημονικών δημοσιεύσεων

Αυτό το κεφάλαιο απαρτίζεται από εργαλεία που μας βοηθούν στην ανάλυση δεδομένων και προβολή των επιστημονικών δημοσιεύσεων, δηλαδή θα αναφερθούμε σε εργαλεία που μπορούν να χρησιμοποιηθούν από τον απλό χρήστη που θέλει να κάνει ανάλυση δεδομένων μέχρι και τον προγραμματιστή. Ένας προγραμματιστής μπορεί να χρησιμοποιήσει αυτά τα εργαλεία για να εμπλουτίσει τη δική του εφαρμογή με διάφορα στατιστικά δεδομένα ή να κάνει τη δική του ανάλυση δεδομένων στη δική του εφαρμογή.

5.1 Λογισμικό που απευθύνονται σε χρήστες

Τα λογισμικά που θα αναφερθούν απευθύνονται σε απλούς χρήστες, σε χρήστες οι οποίοι θα κάνουν εγκατάσταση την εφαρμογή στη συσκευή τους και θα τη χρησιμοποιήσουν απευθείας. Στη συγκεκριμένη κατηγορία θα προταθούν 3 εφαρμογές με τις οποίες καλύπτονται 4 λειτουργικά συστήματα, έτσι ώστε να καλύψουν τον κάθε ένα χρήστη ξεχωριστά ανάλογα με τη συσκευή που έχει στη διάθεσή του.

5.1.1 Publish or Perish

Το λογισμικό αυτό αναπτύχθηκε για τα λειτουργικά συστήματα Windows, Linux και Mac OSX. Η εταιρεία έκανε την πρώτη της εμφάνιση στο χώρο του διαδικτύου το 1999 ξεκινώντας από μια απλή ιστοσελίδα και μέχρι σήμερα έχει καλύψει αυτά τα 4 λειτουργικά συστήματα που αναφέραμε παραπάνω.

Το Publish or Perish (<https://harzing.com/resources/publish-or-perish>) είναι ένα λογισμικό το οποίο ανακτά πληροφορίες από διάφορες πηγές όπως είναι:

- Crossref
- Google Scholar
- Microsoft Academic
- Scopus
- Web of Science

και αναλύει τις αναφορές από τα επιστημονικά έργα των ερευνητών που θα του δώσουν καθώς και την παρουσίαση των ακόλουθων μετρήσεων:

- Συνολικός αριθμός εγγράφων και συνολικός αριθμός παραπομπών
- Μέσες αναφορές ανά άρθρο, αναφορές ανά συγγραφέα, έγγραφα ανά συγγραφέα και αναφορές ανά έτος
- H-index του Hirsch.
- G-index του Egghe' s.
- Τρεις παραλλαγές των ατομικών h-index
- Η μέση ετήσια αύξηση του μεμονωμένου δείκτη h είναι χρήσιμη για τους παρακάτω λόγους:
 - i) Όπως συμβαίνει με τον δείκτη hI, norm , αφαιρεί σε σημαντικό βαθμό τυχόν μοντέλα δημοσιότητας και παραπομπής ειδικά για την πειθαρχία που αλλιώς αλλοιώνουν τον δείκτη h.
 - ii) Μειώνει επίσης την επίδραση του χρόνου σταδιοδρομίας και παρέχει μια δικαιότερη σύγκριση μεταξύ των κατώτερων και ανώτερων ερευνητών.
- Το σταθμισμένο ποσοστό ηλικίας αναφοράς
- Μια ανάλυση του αριθμού των συγγραφέων ανά χαρτί.

Όλες αυτές οι πληροφορίες είναι έτοιμες στην οθόνη του υπολογιστή και υπάρχει η δυνατότητα να γίνει αντιγραφή και να αποθήκευση των αποτελεσμάτων σε διάφορες μορφές.

The screenshot shows the 'Publish or Perish' software interface. At the top, it displays search terms, source (Google Scholar), and various citation metrics. Below this, the 'Google Scholar Profile search' section shows the profile of Christos Gogos, an Associate Professor at the University of Ioannina. The profile includes a list of publications with their titles, years, and citation counts. The 'Results' section at the bottom lists 22 publications, each with a checkbox for selection, a citation count, and a rank. The publications are sorted by citation count in descending order.

Publication years	Citation years	Papers	Cites/year	h	g	h/norm	h/annual	acc10	Search date	Cache date	Last	
2005-2019	14 (2005-2019)	42	378	27.00	9	19	5	0.36	1	7/11/2019	7/11/2019	0

Year	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
h-index	2	4	4	9	35	35	49	39	57	55	42	40	
Total	12	14	18	22	31	66	96	145	184	241	296	338	378

Results	Cites	Per year	Rank	Authors	Title	Year	Publication	Publisher	Type
Publication years: 2005-2019	h 93	13.29	1	C Valouxis, C Gogos	A systematic two phase approach ...	2012	European Journal of Operational Re...		Journal article
Citation years: 14 (2005-2019)	h 66	9.43	2	C Gogos, P Alefrag	An improved multi-staged algorit...	2012	Annals of Operations Research		Journal article
Papers: 42	h 47	9.40	3	G Post, JH Kingston...	XHSTT: an XML archive for high sc...	2014	Annals of Operations Research		Journal article
Cites/year: 27.00	h 35	3.18	4	C Gogos, P Alefrag	A multi-staged algorithmic proces...	2008	Practice and Theory of Automated T...		Journal article
Cites/year: 9.80	h 27	9.00	5	C Gogos, C Valouxi...	Scheduling independent tasks on ...	2016	Future Generation Computer Systems		Journal article
Authors/year: 4.81	h 19	3.17	6	G Foubitz, C Gogos...	Actuator location and voltages op...	2013	Actuators		Journal article
h-index: 9	h 12	1.20	7	C Gogos, G Goulas...	Pursuit of better results for the exa...	2009	Computational Intelligence in Sched...		Conference pa...
g-index: 19	h 11	1.57	8	C Valouxis, C Gogo...	Decomposing the high school tim...	2012	Practice and Theory of Automated T...		Journal article
h/norm: 5	h 9	1.80	9	V Kolonias, G Goul...	Solving the examination timetabl...	2014	Algorithms		Journal article
h/annual: 0.36	h 8	2.00	10	G Papantonopoulou...	Peri-implantitis: a complex conditi...	2015	Journal of clinical periodontology		Journal article
Papers with ACC >= 1.2,5,10,20: 12,7,4,1,0	h 7	0.50	11	C Gogos, P Alefrag	Public enterprise water pump sche...	2005	Emerging Technologies and Factory ...	10th IEEE ...	Conference pa...
	h 6	0.67	12	C Gogos, G Goulas...	Distributed Scatter Search for the ...	2010	The 8th International Conference on ...		Conference pa...
	h 5	0.83	13	C Valouxis, C Gogo...	DAG Scheduling using Integer Pro...	2013	MISTA 2013, Ghent, Belgium		Conference pa...
	h 5	0.42	14	C Gogos, P Alefrag	Appication Of heuristics, Genetic...	2007	11th Panhellenic Conference on Info...		Conference pa...
	h 4	0.67	15	G Foubitz, E Hadji...	Modal shape control of smart co...	2013	10th HSTAM (Hellenic Society for Th...		Conference pa...
	h 4	0.57	16	G Goulas, P Alefrag	From Scilab to multicore embedd...	2012	2012 International Conference on E...		Conference pa...
	h 4	0.33	17	C Kyritsis, I Sotirop...	Note on the effect of the 11 years ...	2007	Archives of Economic History		Journal article
	h 2	1.00	18	G Papantonopoulou...	Prediction of individual implant b...	2017	Clinical oral implants research		Journal article
	h 2	0.67	19	I Xanthopoulos, G ...	Highway Rest Areas simultaneous ...	2016	Proceedings of the 20th Pan-Helleni...		Conference pa...
	h 2	0.33	20	G Goulas, C Valouxi...	Coarse-Grain Optimization and C...	2013	2013 Euromicro Conference on Digit...		Conference pa...
	h 2	0.20	21	X Γκόγκος	Αλγόριθμοι συνδυαστικής βελτισ...	2009	University of Patras		Conference pa...
	h 2	0.17	22	C Gogos, P Alefrag	Sensor enabled rule based alarm s...	2007	2007 IEEE Conference on Emerging ...		Conference pa...

Εικόνα 5.1.1: Γραφικό περιβάλλον Publish or Perish

5.1.2 Go Scholar for Google Scholar

Το συγκεκριμένο λογισμικό έρχεται να καλύψει αυτούς που χρησιμοποιούν κινητές συσκευές με λειτουργικό σύστημα IOS. Το συγκεκριμένο λογισμικό ανακτά πληροφορίες από τη μηχανή αναζήτησης Google Scholar.

Το Go Scholar αναπτύχθηκε το 2017 για τις κινητές συσκευές Iphone και Ipad το οποίο αποτελεί ένα πολύ χρήσιμο εργαλείο για τους φοιτητές και ερευνητές. Οι δυνατότητες που μας παρέχονται από την εφαρμογή είναι οι εξής:

- Αναζήτηση άρθρων μέσω του Google Scholar με αρκετές επιλογές αναζήτησης.
- Αναζήτηση εγγράφων από οποιεσδήποτε βάσεις δεδομένων: IEEE, JSTOR, Web Science και άλλες.
- Αναζήτηση με κριτήριο την επέκταση του αρχείου PDF, DOC ή PPT.
- Αναζήτηση και πρόσβαση στην αμερικάνικη νομολογία.
- Διαχείριση και παρακολούθηση εγγράφων.
- Λήψη και οργάνωση των άρθρων σας μέσα στην εφαρμογή.
- Παροχή αναγνώστη για διάφορους τύπους αρχείων.
- Παρακολούθηση των ληφθέντων άρθρων στο Google Scholar.
- Αποθήκευση των άρθρων στο Google Scholar.
- Παροχή ενσωματωμένου σημειωματάριου για σύνταξη ακαδημαϊκών έργων στην εφαρμογή.
- Συγχρονισμός των σημειώσεων με όλες τις συσκευές μέσω του Dropbox.
- Αποστολή των άρθρων σε διάφορες εφαρμογές όπως email, Dropbox, PDF Reader.

Η συγκεκριμένη εφαρμογή θα λέγαμε πως αποτελεί μια πλήρη κινητή εφαρμογή που δεν της λείπει τίποτα από το πακέτο για τη δημιουργία, διαχείριση, αποστολή ακαδημαϊκών έργων εν κίνηση.

Το Go Scholar είναι μια εμπορική εφαρμογή την οποία μπορεί κανείς να αποκτήσει στην τιμή των 9,99 δολαρίων.

5.1.3 Researcher

Η εφαρμογή Researcher είναι μια εφαρμογή αναζήτησης επιστημονικού έργου για τις κινητές συσκευές Android καθώς επίσης παρέχεται πρόσβαση από έναν οποιονδήποτε browser στη διεύθυνση <https://www.researcher-app.com>. Ιδρύθηκε το 2017 και για τη χρήση της εφαρμογής απαιτείτε εγγραφή του χρήστη στην πλατφόρμα με τρεις τρόπους, email, google account ή Facebook account.

Στη συνέχεια επιλέγεις τον τομέα των επιστημονικών άρθρων που σε ενδιαφέρει καθώς επίσης και τους υποτομείς. Τέλος επιλέγεις βάση των υπό – τομών, ποια επιστημονικά περιοδικά θέλεις να παρακολουθείς και κάποια keywords τα οποία θέλεις να ενημερώνεσαι σχετικά με αυτά. Η εγγραφή είναι έτοιμη και πλέον μπορείς να περιηγηθείς στην εφαρμογή.

Πληροφοριακά μπορείς από την εφαρμογή να επιλέξεις πάνω από:

- 15000 περιοδικά υψηλής ποιότητας σε 10 διαφορετικές περιοχές θεμάτων δημιουργώντας μια προσωπική ροή στυλ κοινωνικής δικτύωσης.

- 1 εκατομμύριο ενεργούς χρήστες.
- 41000 καινούρια άρθρα προστίθενται κάθε εβδομάδα.
- 14800000 μηνιαίες προβολές περιεχομένου μέσα από την εφαρμογή.

Τέλος στην εφαρμογή συμπεριλαμβάνονται όλα τα περιοδικά και οι εφημερίδες με topimpact από όλα τα κορυφαία περιοδικά όπως Nature Springer, ACS, Elsevier, Wiley, Taylor and Francis, RSC, BMJ, IEEE, arXiv, PubMed, PLOS, PNAS, PubMed, F1000, Cell, Science AAAS και άλλα.

5.2 Λογισμικά που απευθύνονται σε προγραμματιστές

Αυτή η κατηγορία λογισμικών απευθύνεται σε όσους είναι επαγγελματίες, προγραμματιστές ή είναι «χομπίστες» πάνω σε αυτόν τον τομέα και ξέρουν προγραμματισμό ή ακόμα και σε αυτούς που έχουν μια μικρή εμπειρία πάνω σε αυτόν τον κλάδο. Στην ουσία αυτά τα λογισμικά έχουν αναπτυχθεί για να διευκολύνουν τη ζωή μας και ακολούθως το επάγγελμά μας. Παρακάτω θα δούμε ποια είναι αυτά, πόσο εύκολη είναι η χρήση κάποιων από αυτά τα εργαλεία και πως τα χρησιμοποιούμε στη δική μας εφαρμογή. Επίσης θα δούμε και κάποια άλλα που θέλουν λίγο ψάξιμο παραπάνω για να τα χρησιμοποιήσει ένας υποψήφιος προγραμματιστής.

5.2.1 Χρήση API

API είναι η συντομογραφία του Application Programming και η ακριβής ονομασία του στα ελληνικά είναι Διασύνδεση Προγραμματισμού Εφαρμογών. Το API είναι ένα ενδιάμεσο λογισμικό που επιτρέπει την επικοινωνία μεταξύ δύο εφαρμογών, δηλαδή είναι ο διαμεσολαβητής που παραδίδει το αίτημα στον πάροχο «πληροφοριών» και με τη σειρά του αυτό επιστρέφει την απάντηση από τον πάροχο πίσω στον χρήστη.

Ένα API καθορίζει τις λειτουργίες που είναι ανεξάρτητες από τις αντίστοιχες υλοποιήσεις του. Αυτό έχει ως αποτέλεσμα να δίνεις μόνο την πληροφορία που πρέπει και όχι τα προσωπικά δεδομένα κάποιου.

Παραδείγματος χάρη, εάν θέλεις να λάβεις τις δέκα πιο δημοφιλής ειδήσεις σήμερα, δίνοντας την εντολή «topnews» το API θα επιστρέψει τις δέκα κυριότερες ειδήσεις που προσφέρει μια υπηρεσία.

Στην περίπτωση που δεν υπήρχαν API η λειτουργικότητα μιας εφαρμογής θα ήταν πολύ περιορισμένη και ο χρόνος ανάπτυξης πολύ περισσότερος καθώς θα έπρεπε να αναπτυχθούν ανεξάρτητα κομμάτια από την κύρια εφαρμογή μας.

(Anon., χ.χ.)

5.2.1.1 Πλεονεκτήματα

Κάποια από τα πλεονεκτήματα του API είναι η

- Εξοικονόμηση χρόνου κατά την ανάπτυξη της εφαρμογής.
- Επιτρέπει την ανάκτηση πληροφοριών από ένα σύστημα στο άλλο.
- Το σύστημα το οποίο ανακτά τις πληροφορίες μπορεί να είναι ανεξάρτητης τεχνολογίας από το δικό σου σύστημα που θα το χρησιμοποιήσεις και αυτό γιατί ότι επιστρέφει το API το επιστρέφει σε μορφή JSON, XML. Δεδομένα σε μορφή JSON μπορούν να διαχειριστούν όλες οι γλώσσες προγραμματισμού και οι τεχνολογίες.

5.2.1.2 Τύποι API

- **Open APIs:** Είναι τα δημόσια API στα οποία δεν υπάρχουν περιορισμοί για την πρόσβαση καθώς είναι διαθέσιμα για το κοινό.
- **Partner APIs:** Είναι τα API τα οποία χρειάζεται να έχει ο χρήστης συγκεκριμένα δικαιώματα για να αποκτήσει πρόσβαση. Αυτά τα APIs δεν είναι διαθέσιμα για το κοινό.
- **Internal APIs:** Είναι τα ιδιωτικά APIs τα οποία χρησιμοποιούνται μόνο από εσωτερικά συστήματα και απευθύνεται για χρήση μέσα σε μια εταιρεία.
- **Composite APIs:** Αυτού του τύπου τα APIs συνδυάζει διαφορετικά API δεδομένων και υπηρεσιών. Αποτελείται από μια σειρά εργασιών που εκτελούνται συγχρόνως ως αποτέλεσμα της εκτέλεσης και όχι κατόπιν αιτήματος μιας εργασίας. Η βασική του χρησιμότητα είναι να επιταχύνει τη διαδικασία εκτέλεσης και τη βελτίωση της απόδοσης των δεκτών.

5.2.1.3 Endpoints(Τελικά σημεία)

Τα τελικά σημεία είναι τα βασικά στοιχεία της αλληλεπίδρασης της εφαρμογής με το API. Τα τελικά σημεία έχουν τη μορφή μιας διεύθυνσης URL (<http://newssite.com/topnews>) και στην περίπτωση αυτή μας επιστρέφει μια λίστα με τα κορυφαία νέα.

(<https://rapidapi.com/>, χ.χ.)

5.2.1.4 Πλατφόρμες επιστημονικού έργου με δυνατότητα API

Παρακάτω θα αναφερθούν κάποιες πλατφόρμες οι οποίες έχουν υλοποιήσει ένα σύστημα API για τη διευκόλυνση ανάκτησης πληροφοριών από τον προγραμματιστή

χωρίς την ανάγκη ανάπτυξης web scraping(εκτενέστερη αναφορά θα γίνει στη συνέχεια) εργαλείου για την ανάκτηση πληροφοριών.

- Microsoft Academic
- Semantic Scholar
- Scopus
- PubMed
- Elsevier

Αυτές είναι κάποιες από τις πολλές πλατφόρμες που υποστηρίζουν επικοινωνία με API.

Μέσα στη λίστα βλέπουμε πως δεν βάλαμε μια μεγάλη πλατφόρμα την Google Scholar και αυτό γιατί η Google δεν παρέχει κάποιο public API για την ανάκτηση των πληροφοριών από την πλατφόρμα. Με αφορμή την έλλειψη παροχής API κάνουμε τη συγκεκριμένη πτυχιακή δείχνοντας πως θα πάρουμε κάποια δεδομένα από αυτήν την ιστοσελίδα, αναλυτικότερα θα μιλήσουμε στο επόμενο κεφάλαιο.

5.2.2 Web scraping

Το Web Scraping είναι μια αυτοματοποιημένη τεχνική η οποία χρησιμοποιείται για την απόσπαση μεγάλων σε όγκο δεδομένων από μια ιστοσελίδα με σκοπό την αποθήκευση τοπικά ή σε μια βάση δεδομένων με σκοπό να αναλυθούν ή να χρησιμοποιηθούν περαιτέρω σε δεύτερο χρόνο.

Η τεχνική αυτή έρχεται να λύσει το πρόβλημα της χειροκίνητης αντιγραφής των δεδομένων από μια ιστοσελίδα όπου στην περίπτωση μεγάλου όγκου δεδομένων η διαδικασία γίνεται υπερβολικά χρονοβόρα, μπορεί να διαρκέσει ακόμα και ημέρες. Βέβαια ένα εργαλείο που κάνει scraping δεν θα χρειαζόταν στην περίπτωση που ο ιστότοπος παρέχει API έτσι ώστε τα δεδομένα να τα ζητάμε και να μας τα στέλνει σε μορφή JSON και έπειτα να τα αποθηκεύουμε στη βάση δεδομένων μας. Πολύ ιστότοποι δεν επιτρέπουν να εξάγει κανείς τα δεδομένα τους γι' αυτό δεν παρέχουν API έτσι ώστε η μόνη λύση είναι ένα τέτοιο εργαλείο.

Μερικές περιπτώσεις χρήσης του Web Scraping:

- E-commerce price monitoring
- News aggregation
- Lead generation
- SEO (Search engine result page monitoring)
- Bank account aggregation (Mint in the US, Bankin' in Europe)

Συχνό φαινόμενο όταν κάνουμε scraping είναι να αντιμετωπίσουμε κάποιες δυσκολίες σε αυτήν τη διαδικασία όπως είναι η εμφάνιση Captcha ή το μπλοκάρισμα της Ip

address μας για τον λόγο ότι κάνουμε πολλά requests(αιτήματα) για να πάρουμε τα δεδομένα στην περίπτωση που είναι πολλά.

5.2.2.1 Εμπορικά εργαλεία

Τα εμπορικά εργαλεία είναι κάποιες ολοκληρωμένες λύσεις οι οποίες μπορούν να αντιμετωπίσουν τις δυσκολίες που αναφέραμε παραπάνω. Οι εμπορικές λύσεις χρησιμοποιούνται κυρίως από χρήστες οι οποίοι δεν έχουν κάποια γνώση στον χώρο του προγραμματισμού καθώς όλη η διαδικασία γίνεται από το γραφικό περιβάλλον του προγράμματος.

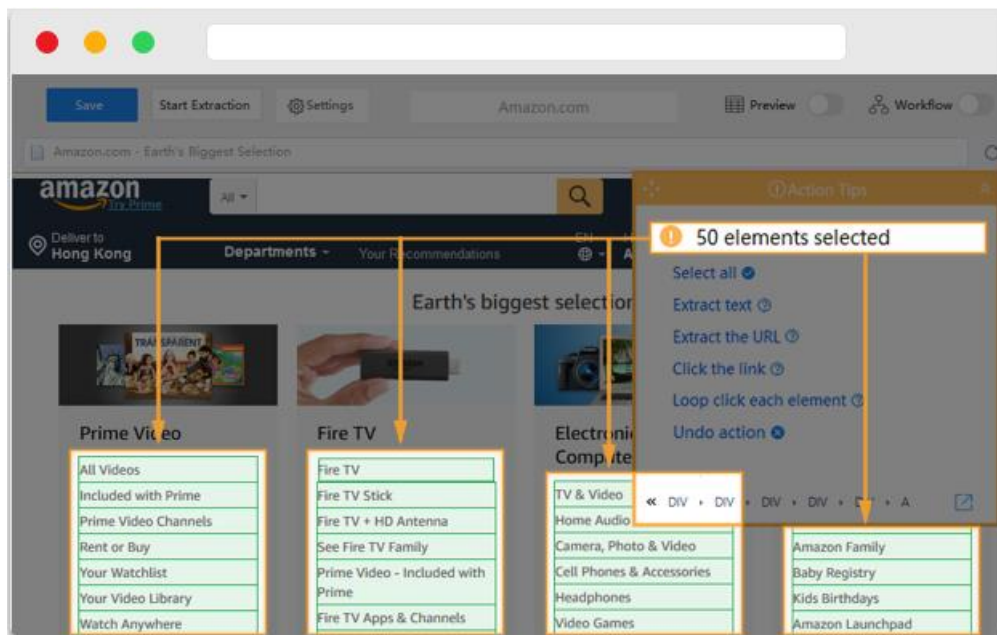
Όλα τα προγράμματα που θα αναφέρουμε παρακάτω διατίθενται στο κοινό από ένα πλήθος προγραμμάτων πληρωμής για την ανάγκη του κάθε project. Βέβαια κάποια από αυτά διαθέτουν πρόγραμμα το οποίο είναι δωρεάν με τους αντίστοιχους περιορισμούς.

Octoparse

Η Octoparse (<https://www.octoparse.com/>) ιδρύθηκε το 2016, είναι ένα εργαλείο web scraping με πολύ καλές κριτικές στον χώρο του ίντερνετ. Οι υπηρεσίες που παρέχει στους πελάτες είναι η εξής:

- Παρακολούθηση τιμών σε πραγματικό χρόνο.
- Εξαγωγή μαζικών δεδομένων από έναν διαδικτυακό χώρο χωρίς κώδικα.
- Generate Sales Leads για λόγους marketing.
- Εξαγωγή σε πραγματικό χρόνο για τροφοδότηση μιας έρευνας με δεδομένα.

Όλη η διαδικασία γίνεται με την επιλογή των πεδίων που θέλουμε να εξάγουμε τα δεδομένα όπως βλέπουμε στην παρακάτω εικόνα.

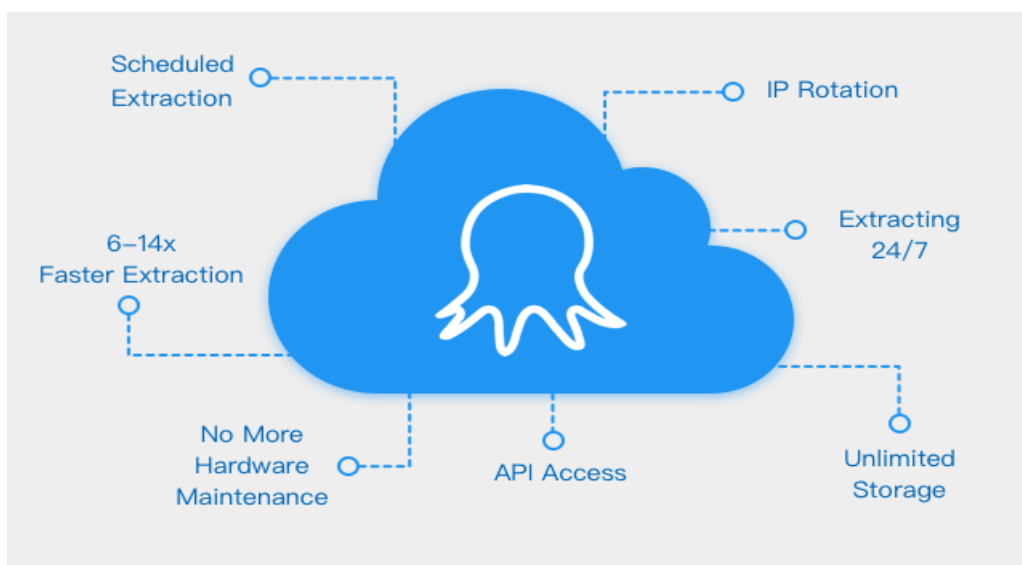


Εικόνα 5.2.1: Παράδειγμα επιλογής πεδίων.

Εξαγωγή δεδομένων από τον πιο περίπλοκο ιστότοπο καθώς και δημιουργία μιας διαδικασίας αυτοματοποιημένης εισόδου του χρήστη, συμπλήρωση φορμών, πεδία αναζήτησης, dropdown μέχρι να φτάσουμε στα δεδομένα που μας ενδιαφέρουν.

Το συγκεκριμένο λογισμικό παρέχει επίσης υπηρεσία cloud στα πακέτα επί πληρωμή έτσι ώστε να μην ανησυχούμε και ξοδεύουμε χρήματα για τη συντήρηση του μηχανήματός μας στην περίπτωση που τρέχουμε το λογισμικό τοπικά. Η cloud υπηρεσία είναι από 6 έως 20 φορές πιο γρήγορη από την τοπική στην εξαγωγή των δεδομένων, με το σημαντικότερο πλεονέκτημα μια cloud υπηρεσίας

- Πρόσβαση στα δεδομένα από ένα οποιοδήποτε μηχάνημα.
- Παροχή αυτόματης υπηρεσίας IP Rotation εξάγοντας δεδομένα με διαφορετική Ip address κάθε φορά για την αποφυγή του μπλοκαρίσματος από τον ιστότοπο στόχο.
- Παροχή προγραμματισμένης εξαγωγής δεδομένων κάνοντας το πρόγραμμα πλήρως ανεξάρτητο από την ανθρώπινη παρέμβαση από τη στιγμή που θα ρυθμιστεί σωστά.
- Εξαγωγή δεδομένων σε οποιαδήποτε μορφή θέλουμε, API, CSV, EXCEL, Database.



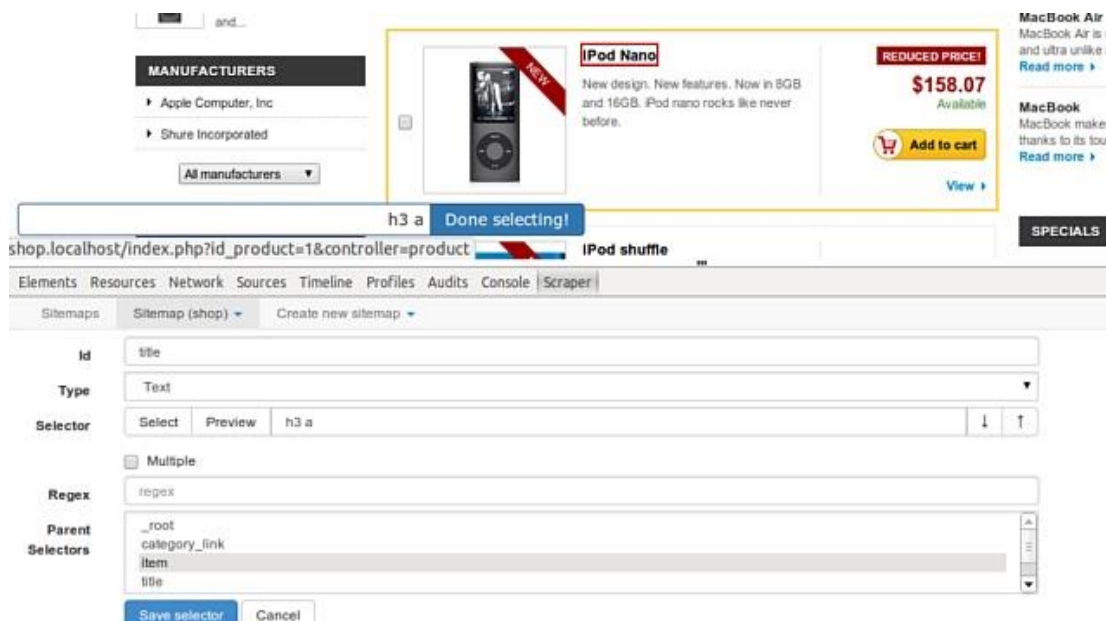
Εικόνα 5.2.2: Παροχή υπηρεσιών από την cloud υπηρεσία της Octoparse.

Τέλος παρέχεται όπως είπαμε και μια δωρεάν έκδοσή του λογισμικού. Για περισσότερες πληροφορίες σχετικά με το λογισμικό αυτό επισκεφθείτε τη σελίδα <https://www.octoparse.com>

Webscraper.io

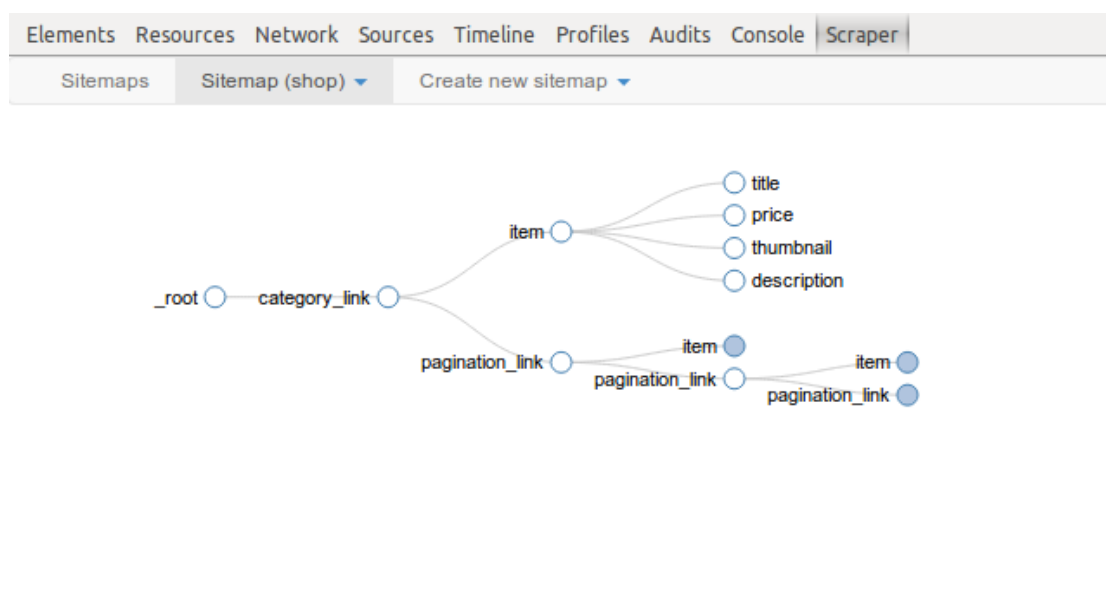
Το webscraper.io παρέχει ένα browser extension το οποίο κάνει την εξαγωγή των δεδομένων σε τοπικό επίπεδο και είναι διαθέσιμο στο Google Chrome.

Η βασική του ρύθμιση γίνεται μέσα από τον browser πατώντας δεξί κλικ στην περιοχή του και πατώντας έλεγχος στην περίπτωση μας. Έπειτα επιλέγουμε τις περιοχές που θέλουμε να εξάγουμε τα δεδομένα.



Create element selectors

Εικόνα 5.2.3: Επιλογή πεδίων ενδιαφέροντος με το εργαλείο Webscraper.io



Build sitemaps from multiple selectors

Εικόνα 5.2.4: Τελικό σχήμα επιλογής πεδίων και λειτουργίας διαδικασίας..

Πατώντας στο Sitemap της διαδικασίας που φτιάξαμε επιλέγουμε το κουμπί Scrape και θα ξεκινήσει η διαδικασία εξαγωγής των δεδομένων όπως θα δούμε παρακάτω.

Sitemaps				
Sitemap (shop) ▼		Create new sitemap ▼		
category_link	category_link-href	title	price	thi
Laptops	http://shop.localhost/index.php?id_category=5&controller=category	MacBook Air	504.18	htt
Laptops	http://shop.localhost/index.php?id_category=5&controller=category	MacBook	170.57	htt
Accessories	http://shop.localhost/index.php?id_category=4&controller=category	Belkin Leather Folio for iPod...	25.04	htt
Accessories	http://shop.localhost/index.php?id_category=4&controller=category	Shure SE210 Sound-Isolating...	124.58	htt
iPods	http://shop.localhost/index.php?id_category=3&controller=category	iPod Nano	158.07	htt
iPods	http://shop.localhost/index.php?id_category=3&controller=category	iPod shuffle	66.05	htt
iPods	http://shop.localhost/index.php?id_category=3&controller=category	iPod touch 2	241.64	htt
iPods	http://shop.localhost/index.php?	iPod Nano 2	158.07	htt

Browse and export scraped data

Εικόνα 5.2.5: Τελικά δεδομένα ύστερα από εκτέλεση της διαδικασίας με τη βοήθεια του εργαλείου Webscraper.io

Αφού ολοκληρωθεί η εξαγωγή μπορούμε να εξάγουμε τα δεδομένα μας σε μορφή CSV.

Τέλος να αναφέρουμε πως υπάρχει και η υπηρεσία επί πληρωμή η οποία μας δίνει τη δυνατότητα να αυτοματοποιήσουμε ακόμα περισσότερο τη διαδικασία παρέχοντάς μας:

- Δυνατότητα προγραμματισμού μιας διαδικασίας εξαγωγής δεδομένων.
- Διαχείριση του λογισμικού μας μέσω API.
- IP Rotation παρέχοντάς μας χιλιάδες IP addresses για την αποφυγή του μπλοκαρίσμάτος μας.

5.2.2.2 Ελεύθερα εργαλεία

Τα ελεύθερα εργαλεία είναι αυτά τα οποία δεν είναι ολοκληρωμένες λύσεις όπως τα εμπορικά, αντίθετα χρειάζονται προγραμματιστικές γνώσεις για να χρησιμοποιηθούν σε συνδυασμό με την εφαρμογή μας.

Όπως είδαμε στις εμπορικές εφαρμογές όλα γίνονται από το γραφικό περιβάλλον του λογισμικού ή από τον browser, σε αυτή την κατηγορία των εργαλείων ο χρήστης πρέπει να ξέρει να γράφει κώδικα για να προγραμματίσει μια διαδικασία έτσι ώστε να φτιάξει ένα αυτοματοποιημένο εργαλείο εξαγωγής δεδομένων από έναν ιστότοπο.

Αυτά τα εργαλεία δεν μας παρέχουν τίποτε άλλο παρά μόνο την επιλογή των πεδίων που θέλουμε να εξάγουμε, για παράδειγμα δεν διαθέτουν υπηρεσία IP Rotation με αποτέλεσμα να πρέπει να χρησιμοποιούμε άλλη υπηρεσία σε συνδυασμό με τα

εργαλεία για την αποφυγή του μπλοκαρίσματος. Περισσότερα για την αποφυγή του μπλοκαρίσματος του Web Scraping με αυτά τα εργαλεία θα μιλήσουμε στο τέλος του κεφαλαίου.

Puppeteer

Το puppeteer είναι μια Node βιβλιοθήκη που παρέχει υψηλού επιπέδου API για τον έλεγχο του Chrome μέσω του πρωτόκολλου DevTools. (Το Node.js είναι ένα εργαλείο ανάπτυξης λογισμικού υλοποιημένο σε περιβάλλον JavaScript.)

Τι μπορούμε να κάνουμε με το puppeteer?

- Δημιουργία Screenshot και PDF από ιστότοπους.
- Εξαγωγή δεδομένων από Single-Page Application και δημιουργία προκαθορισμένου περιεχομένου.
- Αυτοματοποίηση υποβολής φόρμας, UTesting και keyboardinput.
- Δημιουργία ενημερωμένου και αυτοματοποιημένου περιβάλλοντος δοκιμών.
- Εκτέλεση των δοκιμών απευθείας από τον Chrome.
- Καταγραφή ενός timeline με σκοπό τη βοήθεια διάγνωσης προβλημάτων απόδοσης.
- Test Chrome extension

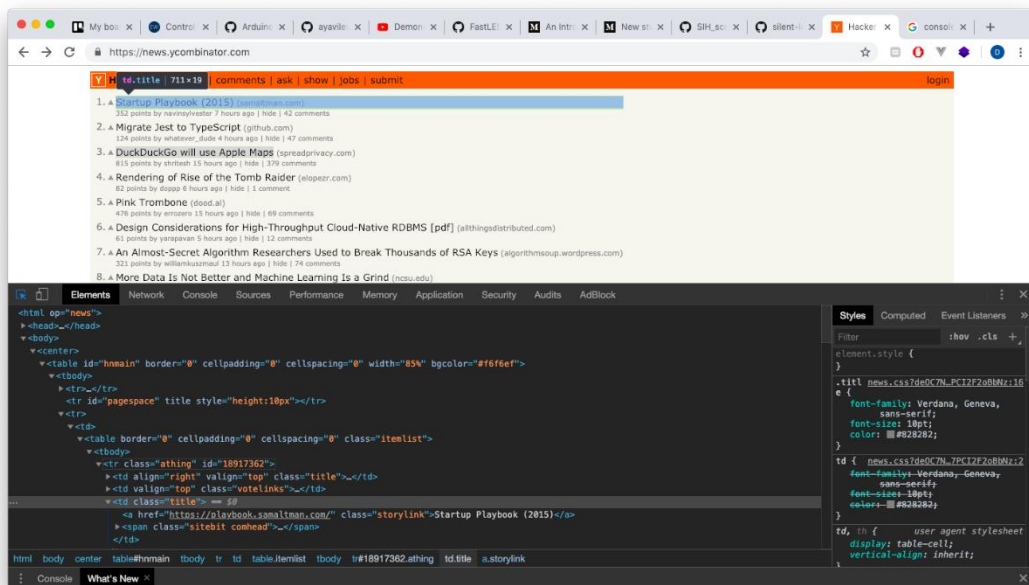
Παρακάτω θα δούμε ένα παράδειγμα κώδικα το οποίο κάνει scraping από έναν σύνδεσμο URL και αποθηκεύει και εκτυπώνει τα 10 πρώτα links που βρίσκονται στη λίστα του URL :

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch()
  const page = await browser.newPage()
  await page.tracing.start({
    path: 'trace.json',
    categories: ['devtools.timeline']
  })
  await page.goto('https://news.ycombinator.com/news')

  // execute standard javascript in the context of the page.
  const stories = await page.$$eval('a.storylink', anchors => {
return anchors.map(anchor => anchor.textContent).slice(0, 10) })
  console.log(stories)
  await page.tracing.stop();
  await browser.close()
}) ()

(https://github.com/checkly/puppeteer-
examples/blob/master/1.%20basics/get_list_of_links.js, n.d.)
```



Εικόνα 5.2.6: Ιστότοπος από τον οποίο κάνουμε web scraping με τη βοήθεια του puppeteer.

Τέλος το puppeteer υποστηρίζει την επίσκεψη ενός ιστότοπου με σκοπό το scraping μέσω proxy με σκοπό την αποφυγή του μπλοκαρίσματος.

Simple Html Dom Parser

Το Simple Html Dom Parser(<https://simplehtmldom.sourceforge.io/>) είναι μια δωρεάν βιβλιοθήκη γραμμένη σε PHP η οποία μας βοηθάει στον web scraping σε γλώσσα PHP. Όπως και οι προηγούμενες έτσι και αυτή η βιβλιοθήκη για να λειτουργήσει απαιτεί κώδικα.

Το συγκεκριμένο εργαλείο:

- Επιτρέπει τη διαχείριση του Html κώδικα με πολύ εύκολο τρόπο.
- Απαιτεί έκδοση PHP 5+.
- Υποστηρίζει Html κώδικα χωρίς να έχει περάσει το W3CValidator.
- Εύρεση των tags σε έναν κώδικα Html με selectors όπως στην jQuery.
- Εξαγωγή του περιεχομένου από κώδικα Html με μια γραμμή.

Παρακάτω θα δούμε ένα απλό κομμάτι κώδικα το οποίο εξάγει δεδομένα από τη γνωστή βάση δεδομένων με ταινίες IMDB, δίνοντας ένα URL που αντιστοιχεί σε μια ταινία το εργαλείο μας επιστρέφει πληροφορίες σχετικά με τον τίτλο αυτό.

```
<?php
```

```

include_once('../../simple_html_dom.php');

function scraping_IMDB($url) {
    // create HTML DOM
    $html = file_get_html($url);

    // get title
    $ret['Title'] = $html->find('title', 0)->innertext;

    // get rating
    $ret['Rating'] = $html->find('div[class="general rating"] b', 0)-
>innertext;

    // get overview
    foreach($html->find('div[class="info"]') as $div) {
        // skip user comments
        if($div->find('h5', 0)->innertext=='User Comments:')
            return $ret;

        $key = '';
        $val = '';

        foreach($div->find('*') as $node) {
            if ($node->tag=='h5')
                $key = $node->plaintext;

            if ($node->tag=='a' && $node->plaintext!='more')
                $val .= trim(str_replace("\n", '', $node-
>plaintext));

            if ($node->tag=='text')
                $val .= trim(str_replace("\n", '', $node-
>plaintext));
        }

        $ret[$key] = $val;
    }

    // clean up memory
    $html->clear();
    unset($html);

    return $ret;
}

// -----
// test it!
$ret = scraping_IMDB('http://imdb.com/title/tt0335266/');

foreach($ret as $k=>$v)
    echo '<strong>'.$k.' </strong>'.$v.'<br>';
?>

(https://github.com/sunra/php-simple-html-dom-
parser/blob/master/Src/Sunra/PhpSimple/simplehtml_dom_1_5/example/scra
ping/example_scraping_imdb.php, n.d.)

```

5.2.3 Παράκαμψη των anti-scraping τεχνικών

Ορισμένοι ιστότοποι προστατεύονται με διάφορες τεχνικές από το Web Scraping με σκοπό να προστατέψουν το περιεχόμενό τους. Για το λόγο αυτό θα κάνουμε αναφορά στις βασικότερες anti-scraping τεχνικές με σκοπό να παρακάμψουμε την οποιαδήποτε προστασία διαθέτει ένας ιστότοπος.

Οι τέσσερις τεχνικές είναι:

1. IP detection
2. IP rate limiting
3. Browser detection
4. Tracking user behavior

IP detection

Οι λόγοι οι οποίοι μπορεί ένας ιστότοπος να χρησιμοποιεί IP Detection είναι:

- Περιεχόμενο το οποίο απευθύνεται σε συγκεκριμένες χώρες.
- Περιεχόμενο το οποίο προστατεύεται από μη ανθρώπινη κίνηση, για παράδειγμα από ένα script το οποίο κάνει αυτόματα αιτήματα σε έναν ιστότοπο.

Αυτή η τεχνική μπορεί πολύ εύκολα να παρακαμφθεί από έναν proxy server.

Proxy server είναι μια υπηρεσία η οποία της ζητάμε να επισκεφθεί έναν ιστότοπο και να μας επιστρέψει το αποτέλεσμα μέσου αυτού, με μια λέξη θα λέγαμε πως είναι ένας διαμεσολαβητής.

IP rate limiting

Η δεύτερη πιο διαδεδομένη τεχνική είναι η IP Rate Limiting η οποία περιορίζει την πρόσβασή βάση του πόσα αιτήματα έχουν υποβληθεί από μια IP σε ένα συγκεκριμένο χρονικό διάστημα.

Αυτός ο τύπος προστασίας είναι είτε χειροκίνητος δηλαδή ένας υπάλληλος βλέπει αυξημένη κίνηση στα αρχεία καταγραφής και μπλοκάρει την πρόσβαση ή γίνεται αυτοματοποιημένα.

Παραδείγματος χάριν, ένας ιστότοπος όταν έχει ένα όριο 200 αιτήματα και εμείς κάνουμε 201 πιθανών να δούμε Captcha αντί το αποτέλεσμα μιας αίτησης.

Τέτοια προστασία μπορεί να είναι είτε προσωρινή είτε μόνιμη όταν δεν γίνεται από φυσικό άνθρωπο.

Οι τεχνικές παράκαμψης στο συγκεκριμένο μέτρο προστασίας είναι δύο:

1. Εισαγωγή καθυστέρησης ανάμεσα στα αιτήματα.
2. Χρησιμοποίηση ενός proxy server με IP Rotation με σκοπό την αυτόματη εναλλαγή των IP ύστερα από συγκεκριμένο αριθμό αιτημάτων.

User agents

Η τρίτη προστασία που μπορεί να χρησιμοποιήσει είναι η ανίχνευση κεφαλίδων HTTP User-Agent με σκοπό τον αποκλεισμό της πρόσβασης βάση συγκεκριμένων συσκευών. Ωστόσο, εάν ένα λογισμικό εντοπίσει μια κακόβουλη συσκευή, κατευθείαν θα τη μπλοκάρει.

Για την παράκαμψη αυτής της προστασίας πρέπει :

- Κάθε αίτημα να γίνεται με μια τυχαία κεφαλίδα HTTP User-Agent.
- Αφαίρεση των Cookies πριν γίνει η επίσκεψη στην ιστοσελίδα.

Ο προγραμματισμός και η επίσκεψη της σελίδας μέσω του puppeteer έρχεται να μας δώσει λύση στο συγκεκριμένο τρόπο προστασίας.

Tracking user behavior

Η τελευταία προστασία στην οποία θα κάνουμε αναφορά είναι η παρακολούθηση της συμπεριφοράς του χρήστη σε έναν ιστότοπο, με σκοπό να ανιχνεύσει οτιδήποτε δεν γίνεται από τον άνθρωπο. Παραδείγματος χάρη το κλικ σε έναν σύνδεσμο χωρίς να μετακινηθεί ο κέρσοντας.

Αυτός ο τύπος προστασίας συνήθως εφαρμόζεται με IP rate limiting και με τα αποτυπώματα που έχει αφήσει ένας browser (cookies), είναι πιο περίπλοκος από τις προηγούμενες τεχνικές προστασίας και εντοπίζεται βάση συγκεκριμένων μοτίβων.

(<https://help.apify.com/en/articles/1961361-several-tips-how-to-bypass-website-anti-scraping-protections>, n.d.)

6 Σχεδιασμός εφαρμογής

6.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί θα κάνουμε αναφορά σχετικά με τον σχεδιασμό της εφαρμογής που θα υλοποιήσουμε. Αναλυτικότερα θα ορίσουμε τις προδιαγραφές που έχει η εφαρμογή μας, καθώς επίσης και τις τεχνολογίες που έχουμε χρησιμοποιήσει για την υλοποίησή της.

Περίληπτικά στην παρούσα πτυχιακή εργασία θα αναπτύξουμε μια εφαρμογή της οποίας η βασική της λειτουργία θα είναι το web scraping από μια πλατφόρμα καταλογογράφησης, τη Google Scholar, του ερευνητικού έργου των καθηγητών.

Αναλυτικότερα, ο σκοπός της συγκεκριμένης εφαρμογής είναι η παραγωγή στατιστικών στοιχείων για τον κάθε καθηγητή ξεχωριστά ανά διδακτική περίοδο. Για να καταγράψουμε τα στατιστικά στοιχεία πρέπει να πάρουμε τις πληροφορίες που χρειαζόμαστε από το Google Scholar, να τις αποθηκεύσουμε με σωστό τρόπο στη βάση δεδομένων και έπειτα από τη βάση δεδομένων να διαχειριστούμε αυτές τις πληροφορίες ώστε να βγάλουμε έγκυρα αποτελέσματα.

6.2 Προδιαγραφές

Γενικά ο όρος προδιαγραφές σημαίνει όταν ένα υλικό, λογισμικό, απαιτεί κάποια χαρακτηριστικά και κάποιες προϋποθέσεις που πρέπει να πληροί έτσι ώστε να υφίσταται και γενικά να είναι ασφαλές προς τη χρήση του.

Αφού ορίσαμε τις προδιαγραφές σαν μια γενική έννοια πάμε να δώσουμε κάποιες περισσότερες λεπτομέρειες στον τομέα της ανάπτυξης λογισμικού.

Προδιαγραφές στην ανάπτυξη λογισμικού είναι το πρώτο πράγμα που σκέφτεται ένας προγραμματιστής τη στιγμή που του αναθέτουν ένα project. Τις προδιαγραφές δεν τις βγάζει ο προγραμματιστής αλλά δημιουργούνται από τα απαιτούμενα που έχει ο πελάτης από την εφαρμογή και σε κάποιες περιπτώσεις μπορεί να δημιουργήσει κάποιες από αυτές και ο προγραμματιστής με σκοπό την ομαλή χρήση της εφαρμογής κατά τη χρήση ή πιθανών ακόμα και μιας κακόβουλης χρήσης της εφαρμογής.

Οι προδιαγραφές όπως είδαμε ακόμα και στο δεύτερο κεφάλαιο είναι το πρώτο βήμα που γίνεται στα μοντέλα ανάπτυξης λογισμικού, οπότε θα λέγαμε πως αποτελεί τη βάση μιας εφαρμογής και βάσει αυτής στο τέλος αξιολογείται και η δουλειά του προγραμματιστή από τον πελάτη.

Έχουμε 3 μοντέλα καταγραφής των απαιτήσεων λογισμικού.

1. Λειτουργικές και μη λειτουργικές απαιτήσεις λογισμικού.
2. Περιπτώσεις χρήσης(Use Cases).
3. Ιστορίες χρηστών(User Stories).

Αρχικά αναφέρουμε κάποιες πληροφορίες σχετικά με το μοντέλο που επιλέξαμε και στη συνέχεια θα αναφέρουμε όλες τις περιπτώσεις χρήσης της εφαρμογής μας. Στη δική μας περίπτωση επιλέξαμε για την καταγραφή των προδιαγραφών της εφαρμογής μας το μοντέλο «περιπτώσεις χρήσης (Use Cases)» για την καταγραφή αυτών.

Περιπτώσεις χρήσης είναι ένα σύνολο σεναρίων τα οποία συνδέονται με έναν συγκεκριμένο σκοπό του χρήστη. Παρακάτω θα αναφέρουμε ειδικότερα ποιος είναι ο στόχος χρήσης των περιπτώσεων χρήσης.

- Να καθορίσουν και να περιγράψουν τις **λειτουργικές απαιτήσεις** του συστήματος.
- Να δώσουν μια σαφή εικόνα και περιγραφή για το **τι θα πρέπει να κάνει το σύστημα**.
- Να παρέχουν την κατάλληλη βάση για να γίνονται έλεγχοι για **επαλήθευση του συστήματος**.
- Να παρέχουν την ικανότητα να εντοπίζονται οι λειτουργικές απαιτήσεις μέσα **στις κλάσεις και τις λειτουργίες του συστήματος**.

Κάθε Use Cases αποτελείται από ένα δομημένο κείμενο. Μια ομάδα use cases μπορεί να συνοδεύεται από διαγράμματα UML.

Η βάση για τη δημιουργία ενός Use Case είναι:

Δράστης(actor) είναι ένα σύστημα ή αλλιώς ένας ρόλος είτε ανθρώπινος είτε άλλο λογισμικό.

- Εξωτερικό ως προς το λογισμικό που προδιαγράφουμε.
- Και που αλληλεπιδρά με το λογισμικό που προδιαγράφουμε, δίνοντας είσοδο στο ίδιο λογισμικό είτε παίρνοντας έξοδο απ' αυτό.

Δράστης: Τα Use Cases είναι μια ακολουθία ενεργειών ενός δράστη προς το λογισμικό για την υλοποίηση μιας λειτουργίας. Ο διάλογος του δράστη με το λογισμικό αποτελείται από διαδοχικά βήματα.

Εξωτερικό σημαίνει ότι το παραπάνω κομμάτι που αναφέρεται ως εξωτερικό δεν είναι μέρος του λογισμικού που σχεδιάζουμε. Παράδειγμα εξωτερικού θα λέγαμε πως αποτελεί ένα API το οποίο μας παρέχει κάποιες πληροφορίες και είναι ανεξάρτητο από το λογισμικό μας.

Τα Use Cases αποτελούνται από μια μορφή όπως θα δούμε παρακάτω:

- Use case id (ταυτότητα της περίπτωσης χρήσης)
- Use case title (τίτλος ή όνομα)
- Actor (δράστης) Preconditions
- Αρχικές προαπαιτούμενες συνθήκες
- Use case Triggers (γεγονότα που ενεργοποιούν το use case)
- Basic Flow (βασική ροή)

- Alternative Flows (εναλλακτικές ροές)
- Post conditions after basic flow (τελικές συνθήκες μετά τη βασική ροή)
- Post conditions after alternative flows (τελικές συνθήκες μετά τις εναλλακτικές ροές)

Γενικά η καταγραφή των περιπτώσεων χρήσης καλό θα ήταν να γίνεται προσεκτικά έτσι ώστε να μην περιοριστούμε, σε περίπτωση που θέλουμε να αλλάξουμε κάτι κατά τη διάρκεια της σχεδίασης, για τον λόγο αυτόν η καταγραφή των Use Cases γίνεται πρώτη σε σχέση με την υλοποίηση της εφαρμογής. Παραδείγματος χάρι δεν λέμε πατάμε το «X» κουμπί για να ξεκινήσει μια διαδικασία καθώς δεν έχουμε ξεκινήσει τον σχεδιασμό της εφαρμογής και δεν ξέρουμε τι κουμπιά θα έχει η εφαρμογή μας, απλά καταγράφουμε αυτό που θέλουμε να κάνουμε γενικότερα.

(ΒΕΣΚΟΥΚΗΣ, 2015)

(Anon., 2007)

6.2.1 Περιπτώσεις χρήσης

Στη συνέχεια θα αναφέρουμε όλες τις περιπτώσεις χρήσης του λογισμικού μας για να δούμε ποιες λειτουργίες πρέπει να αναπτύξουμε προγραμματιστικά.

Περίπτωση χρήσης «Σύνδεση διαχειριστή»

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Η κατάσταση του διαχειριστή είναι «ασύνδετος».

Βασική ροή:

B1α. Ο πελάτης ζητά να συνδεθεί.

B1β1. Το λογισμικό προτρέπει τον διαχειριστή να δώσει τα δεδομένα σύνδεσης.

B2α. Ο διαχειριστής εισάγει τα δεδομένα σύνδεσής του.

B2β1. Το λογισμικό ελέγχει αν τα δεδομένα που έδωσε ο διαχειριστής είναι έγκυρα.

B2β2. Αν είναι έγκυρα, το λογισμικό πραγματοποιεί σύνδεση.

Τελική συνθήκη μετά τη βασική ροή: Η κατάσταση του διαχειριστή είναι «Συνδεδεμένος».

Εναλλακτικές ροές:

Στο βήμα B1β1, αν τα δεδομένα σύνδεσης του διαχειριστή είναι λανθασμένα εμφανίσε μήνυμα «Λανθασμένα δεδομένα σύνδεσης, παρακαλώ προσπαθήστε ξανά».
Η ροή επιστρέφει στο βήμα B1α.

1. Στο βήμα B1β1, ο διαχειριστής επιλέγει να ακυρώσει τη λειτουργία σύνδεσης.
 - a) Το λογισμικό ακυρώνει τη λειτουργία και επιστρέφει τη ροή στην προηγούμενη λειτουργία.

Περίπτωση χρήσης «Εγγραφή καθηγητή στη βάση δεδομένων».

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Κανένας καθηγητής στη βάση δεδομένων.

Βασική ροή:

- B1α. Ο διαχειριστής ζητά να προσθέσει έναν καθηγητή στη βάση δεδομένων.
- B1β1. Το λογισμικό προτρέπει τον διαχειριστή να δώσει τα στοιχεία ενός καθηγητή.
- B2α. Ο διαχειριστής εισάγει τα προσωπικά στοιχεία του καθηγητή.
- B2β1. Το λογισμικό ελέγχει εάν όλα τα προσωπικά στοιχεία έχουν συμπληρωθεί.
- B2β2. Αν είναι όλα τα προσωπικά στοιχεία συμπληρωμένα τότε πραγματοποιήσε καταχώρηση του καθηγητή στη βάση δεδομένων.

Τελική συνθήκη μετά τη βασική ροή: Τα στοιχεία του καθηγητή έχουν καταχωρηθεί με επιτυχία στη βάση δεδομένων.

Εναλλακτικές ροές:

1. Στο βήμα B1β1, αν τα προσωπικά στοιχεία του καθηγητή ή κάποιο από αυτά δεν έχει συμπληρωθεί εμφανίσε μήνυμα «Δώσε όλα τα απαιτούμενα προσωπικά στοιχεία».
Η ροή επιστρέφει στο βήμα B1α.
2. Στο βήμα B1β1, ο διαχειριστής επιλέγει να ακυρώσει τη λειτουργία καταχώρηση καθηγητή στη βάση δεδομένων.
 - a. Το λογισμικό ακυρώνει τη λειτουργία και επιστρέφει τη ροή στην προηγούμενη λειτουργία.

Περίπτωση χρήσης «Εισαγωγή Scholar Id σε έναν καθηγητή».

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Καθηγητής χωρίς κατοχυρωμένο Scholar Id.

Βασική ροή

B1α. Ο διαχειριστής ζητά να προσθέσει Scholar Id σε έναν καθηγητή.

B1β1. Αφού γίνει web scraping από το scholar στη λίστα των καθηγητών με ένα συγκεκριμένο όνομα και εμφανιστεί η λίστα με τα ονόματα των καθηγητών, το λογισμικό μας παραπέμπει να επιλέξουμε πιο προφίλ αντιστοιχεί στον καθηγητή που θέλουμε να κατοχυρώσουμε το Scholar Id του.

B2α. Ο διαχειριστής επιλέγει το προφίλ που αντιστοιχεί στον καθηγητή.

B2β1. Το λογισμικό ελέγχει εάν έχει καταχωρηθεί το Scholar Id.

B2β2. Εάν δεν έχει καταχωρηθεί παλαιότερα Scholar Id τότε πραγματοποιείται ενημέρωση της καταχώρησης του καθηγητή με συμπληρωμένο το Scholar Id στη βάση δεδομένων.

Τελική συνθήκη μετά τη βασική ροή: Το Scholar Id του καθηγητή έχει καταχωρηθεί με επιτυχία στη βάση δεδομένων.

Εναλλακτικές ροές:

1. Στο βήμα B1β1, ο διαχειριστής επιλέγει να ακυρώσει τη λειτουργία καταχώρησης Scholar Id για τον καθηγητή.
 - a. Το λογισμικό ακυρώνει τη λειτουργία και επιστρέφει τη ροή στην προηγούμενη λειτουργία.
2. Στο βήμα B2β1, εάν υπάρχει καταχωρημένο Scholar Id για τον καθηγητή που επιλέξαμε τότε εμφανίσε μήνυμα «Το Scholar Id για τον συγκεκριμένο καθηγητή είναι καταχωρημένο στη βάση δεδομένων»

Περίπτωση χρήσης «Εισαγωγή άρθρων από το Google Scholar στη βάση δεδομένων για έναν καθηγητή».

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Κανένα άρθρο στη βάση δεδομένων για έναν καθηγητή.

Βασική ροή:

B1α. Ο διαχειριστής ζητά να προσθέσει όλα τα άρθρα από την πλατφόρμα Google Scholar.

B1β1. Το λογισμικό επιστρέφει μια λίστα με όλα τα άρθρα του καθηγητή.

B2α. Αν επιστρέψει επιτυχώς τη λίστα με τα άρθρα και εάν το συγκεκριμένο άρθρο δεν βρίσκεται κατοχυρωμένο ήδη στη βάση δεδομένων τότε ένα άλλο εργαλείο θα φέρει πληροφορίες για το κάθε άρθρο ξεχωριστά.

B2β1. Το λογισμικό επιστρέφει με επιτυχία τις πληροφορίες για ένα άρθρο.

B2β2. Αφού έχουμε όλες τις πληροφορίες για ένα άρθρο τότε το λογισμικό κάνει αποθήκευση στη βάση δεδομένων.

B3. Το λογισμικό πάει πάλι στο βήμα B1α για το επόμενο άρθρο στη λίστα.

Εναλλακτικές ροές:

1. Στο βήμα B1β1, αν δεν επιστρέψει τη λίστα με όλα τα άρθρα του καθηγητή το πρόγραμμα αυτόματα ξανά τρέχει τη διαδικασία ανάκτησης της λίστας μέχρι να φέρει αποτέλεσμα.
2. Στο βήμα B2α, εάν το άρθρο βρίσκεται στη βάση δεδομένων τότε πάμε στο βήμα B3.
3. Στο βήμα B2β1, αν το λογισμικό δεν επιστρέψει τις πληροφορίες για ένα άρθρο τότε θα λειτουργεί η διαδικασία μέχρι να φέρει τις πληροφορίες που χρειαζόμαστε.

Περίπτωση χρήσης «Εισαγωγή των κωδικών αναφορών από το Google Scholar στη βάση δεδομένων για κάθε άρθρο για έναν καθηγητή».

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Κανένας κωδικός αναφοράς στη βάση δεδομένων.

Βασική ροή:

B1α. Ο διαχειριστής ζητά να προσθέσει όλους τους κωδικούς αναφοράς για τα άρθρα του από την πλατφόρμα Google Scholar με σκοπό σε δεύτερη φάση να φέρουμε και τις πληροφορίες για την κάθε αναφορά.

B1β1. Το λογισμικό ανακτά όλα τα άρθρα του καθηγητή που έχουν τουλάχιστον μια αναφορά.

B2α. Έλεγχος βάσης του αριθμού αναφορών που υπάρχει στη βάση δεδομένων και στην πλατφόρμα Google Scholar.

B2β1. Εφόσον υπάρχει διαφορά μεταξύ των δύο αριθμών τότε για κάθε άρθρο του καθηγητή που έχει αναφορά έλεγξε τον αριθμό των αναφορών για το συγκεκριμένο άρθρο στη βάση δεδομένων και στην πλατφόρμα Google Scholar.

B2β2. Εφόσον και αυτοί οι αριθμοί είναι διαφορετικοί τότε κάνει Scraping όλη τη λίστα με τις αναφορές για το συγκεκριμένο άρθρο.

B3. Για κάθε μια αναφορά εάν δεν υπάρχει στη βάση δεδομένων αποθήκευσε την και αύξησε έναν μετρητή έτσι ώστε να γνωρίζουμε πόσες καινούριες αναφορές έχουν προστεθεί.

B3β1. Αποθήκευσε τον καινούριο αριθμό αναφορών για το συγκεκριμένο άρθρο.

B4. Μεταφέρσου στο επόμενο άρθρο B2β1.

B5. Αφού γίνει αυτή η διαδικασία για όλα τα άρθρα αποθήκευσε τον καινούριο συνολικό αριθμό αναφορών του συγκεκριμένου καθηγητή.

Εναλλακτικές ροές:

1. Στο βήμα B1β1, εάν δεν γίνει ανάκτηση κανενός άρθρου από τη βάση δεδομένων τότε εμφάνισε μήνυμα «Δεν υπάρχουν άρθρα στη βάση δεδομένων».
2. Στο βήμα B2β1, εάν δεν υπάρχει διαφορά στις αναφορές μεταξύ βάσης δεδομένων και πλατφόρμας Google Scholar πάμε στο βήμα B4.
3. Στο βήμα B3, εάν υπάρχει η αναφορά στη βάση δεδομένων τότε μην την αποθηκεύεις και μεταφέρσου στην επόμενη.

Περίπτωση χρήσης «Εισαγωγή επιπλέον πληροφοριών στις αναφορές της βάσης δεδομένων μας».

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Καμία επιπλέον πληροφορία στον πίνακα με τις αναφορές στη βάση δεδομένων.

Βασική ροή:

B1α. Ο διαχειριστής ζητά να προσθέσει επιπλέον πληροφορίες στις αναφορές ενός καθηγητή.

B1β1. Το λογισμικό επιστρέφει μια λίστα με όλες τις αναφορές ενός συγκεκριμένου καθηγητή.

B2α. Αν επιστρέψει επιτυχώς τη λίστα με τις αναφορές για κάθε μια αναφορά και αφού δεν υπάρχουν πληροφορίες για τη συγκεκριμένη αναφορά κάνει scraping από την πλατφόρμα Google Scholar.

B2β1. Το λογισμικό επιστρέφει με επιτυχία τις επιπλέον πληροφορίες

B3. Κάνει αποθήκευση των πληροφοριών.

B4. Έλεγχος επόμενης αναφοράς.

Εναλλακτικές ροές:

1. Στο βήμα B1β1, εάν δεν γίνει ανάκτηση καμίας αναφοράς για ένα άρθρο από τη βάση δεδομένων τότε εμφανίσε μήνυμα «Δεν υπάρχουν καταχωρημένοι κωδικοί αναφοράς σε άρθρα στη βάση δεδομένων».
2. Στο βήμα B2β1, εάν δεν επιστρέψει πληροφορίες το λογισμικό τρέχει μέχρι να φέρει κάποιο αποτέλεσμα.

Περίπτωση χρήσης «Εμφάνιση καταλόγου άρθρων».

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Ο διαχειριστής είναι συνδεδεμένος.

Βασική ροή:

B1α. Ο διαχειριστής ζητά να δει τον κατάλογο των άρθρων

B1β1. Το λογισμικό εμφανίζει τον κατάλογο των άρθρων.

Εναλλακτική ροή:

1. Στο βήμα B1β1, δεν υπάρχουν άρθρα. Το λογισμικό εμφανίζει μήνυμα «Δεν υπάρχουν άρθρα στη βάση δεδομένων».

Περίπτωση χρήσης «Αναζήτηση άρθρων».

Δράστης (actor): Διαχειριστής

Αρχική συνθήκη: Ο διαχειριστής είναι συνδεδεμένος.

Βασική ροή:

B1α. Από τον πίνακα των άρθρων, ο διαχειριστής μπορεί να κάνει αναζήτηση με δύο διαφορετικά κριτήρια ένα σε κάθε αναζήτηση.

- i) Αναζήτηση με κριτήριο το ονοματεπώνυμο ενός καθηγητή.
- ii) Αναζήτηση με κριτήριο ενός String.

B1β1. Το λογισμικό εμφανίζει τον κατάλογο των άρθρων με τα άρθρα που πληρούν τα κριτήρια αναζήτησης που αναφέραμε παραπάνω.

Εναλλακτική ροή:

Στο βήμα B1β1, δεν βρέθηκαν άρθρα που να πληρούν τα κριτήρια αναζήτησης. Το λογισμικό εμφανίζει αντίστοιχο μήνυμα.

6.3 Τεχνολογίες και Εργαλεία Υλοποίησης

6.3.1 HTML

Η HTML (HyperText Markup Language) είναι μια γλώσσα σήμανσης για τις ιστοσελίδες, γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags). Η γλώσσα σήμανσης έκανε για πρώτη φορά την εμφάνιση της το 1991, αυτήν την περίοδο βρίσκεται στην 5^η έκδοσή της. Η σύνταξη των ετικετών γίνεται με τα σύμβολα «<>» και ενδιάμεσα το όνομα της ετικέτας. Οι ετικέτες λειτουργούν ανά ζεύγη δηλαδή <ετικέτα></ετικέτα>. Με το σύμβολο «/» αναφερόμαστε στο κλείσιμο της ετικέτας. Ανάμεσα στις ετικέτες μπορούμε να προσθέσουμε περιεχόμενο όπως κείμενο, εικόνες, πίνακες, σύμβολα και άλλα.

Η HTML με τη βοήθεια ενός φυλλομετρητή μεταφράζει τις ετικέτες μαζί με το περιεχόμενο και παρουσιάζει το περιεχόμενο χωρίς τις ετικέτες. Παρακάτω θα γίνει αναφορά και σε μια τεχνολογία η οποία μπορεί να κάνει πιο καλαίσθητο τον ιστότοπο που κατασκευάζουμε.

6.3.2 JavaScript

Η JavaScript είναι μια διερμηνευμένη γλώσσα προγραμματισμού για τους ηλεκτρονικούς υπολογιστές, θεωρείται μια γλώσσα σεναρίων και είναι δυναμική. Έχει τυποποιηθεί βάση της γλωσσικής προδιαγραφής ECMAScript. Η σύνταξη της JavaScript είναι επηρεασμένη από τις πιο σημαντικές γλώσσες προγραμματισμού στην ιστορία την C.

Η JavaScript χρησιμοποιήθηκε με μεγάλη επιτυχία από τη μεριά του πελάτη (client-side). Client-side σημαίνει πως όλη η επεξεργασία γίνεται στον φυλλομετρητή μας και όχι στον server που φιλοξενείτε ο ιστότοπος. Σκοπός της είναι η εκτέλεση κώδικα σε ιστοσελίδες, τον χειρισμό των περιεχομένων μιας ιστοσελίδας δίνοντας τη δυνατότητα αλληλεπίδρασης των χρηστών με τον ιστότοπο.

Τι πετυχαίνουμε τελικά με αυτήν τη γλώσσα προγραμματισμού;

Όπως αναφέραμε και παραπάνω αυτή η γλώσσα είναι δυναμική, ο server θα μας «σερβίρει» το περιεχόμενο του ιστότοπου και στη συνέχεια μπορούμε να επεξεργαζόμαστε, να αλλάζουμε και να ανακτούμε τα δεδομένα χωρίς να απασχολούμε τον server με διάφορα αιτήματα σε κάθε μια αλλαγή και όλα αυτά με τη βοήθεια των scripts που είναι γραμμένα και ανεβασμένα στον ιστότοπο που χρησιμοποιούμε.

Η JavaScript επίσης χρησιμοποιείται σε εφαρμογές η οποίες δεν βασίζονται μόνο στο διαδίκτυο, κάποιες εφαρμογές είναι τα έγγραφα PDF, κάποιοι εξειδικευμένοι φυλλομετρητές και κάποιες μικρές εφαρμογές τις οποίες τις χρησιμοποιούμε στην επιφάνεια εργασίας. Η JavaScript μαζί με την HTML και CSS θεωρούνται οι τρεις βασικές τεχνολογίες παραγωγής διαδικτυακών τόπων.

(<https://el.wikipedia.org/wiki/JavaScript>, n.d.)

6.3.3 CSS

Η CSS (Cascading Style Sheets – διαδοχικά φύλλα ύφους) είναι μια γλώσσα η οποία μας βοηθάει στον έλεγχο της εμφάνισης μιας ιστοσελίδας η οποία έχει γραφτεί σε γλώσσα σήμανσης όπως είναι η HTML για την οποία κάναμε αναφορά παραπάνω. Η παραπάνω γλώσσα βρίσκεται στην κατηγορία των γλωσσών φύλλων ύφους.

Κάποια χαρακτηριστικά τα οποία μπορούμε να βελτιώσουμε με την παραπάνω γλώσσα είναι τα χρώματα, τη στοίχιση, το μέγεθος, τη γραμματοσειρά και πολλά άλλα. Η γλώσσα ύφους είναι αναγκαία για την περίπτωση που θέλουμε να φτιάξουμε έναν μοντέρνο ιστότοπο όπως επίσης είναι αναγκαία και για τον λόγο ότι αυτά που μας προσφέρει αυτή η γλώσσα δεν μας τα προσφέρει η γλώσσα σήμανσης HTML εκτός από κάποια πολύ βασικά χαρακτηριστικά.

Η σύνθεση της παραπάνω γλώσσας αποτελείται από κανόνες όπως:

```
<style>
p {
  color: red;
}
</style>
```

Ο παραπάνω κώδικας έχει την εξής λειτουργία, όπου βρεις την ετικέτα p (<p></p>) ότι περιέχει μέσα στην ετικέτα κάνε το περιεχόμενο κόκκινο. Όλοι οι κανόνες γράφονται μέσα στην ετικέτα style.

(<https://el.wikipedia.org/wiki/CSS>, n.d.)

6.3.4 PHP

Η PHP (Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού με σκοπό τη δημιουργία ιστοσελίδων με κύριο χαρακτηριστικό το δυναμικό περιεχόμενο. Αφού γράψουμε τον κώδικα για μια σελίδα και με τη βοήθεια ενός συμβατού διακομιστή (Apache server) παράγουμε σε πραγματικό χρόνο το περιεχόμενο που είναι προς παρουσίαση. Η έξοδος που μπορεί να μας δώσει η php είναι είτε σε έναν φυλλομετρητή αποστέλλοντας ένα HTML αρχείο όπου ο φυλλομετρητής με τη σειρά του παρουσιάζει με σωστό τρόπο το περιεχόμενο, είτε να μεταβιβάσει την έξοδο σε ένα PHP script.

Παρακάτω παραθέτουμε ένα πολύ απλό παράδειγμα κώδικα PHP μέσα σε HTML αρχείο.

```
<!DOCTYPE html>
<html>
<body>

<?php
```



```

$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>

</body>
</html>

```

Τέλος η γλώσσα προγραμματισμού PHP , η οποία είναι μια από της πιο διαδεδομένες τεχνολογίες στον Παγκόσμιο ιστό, βρίσκει αμέτρητες εφαρμογές και ιστότοπους οι οποίες χρησιμοποιούν αυτή τη τεχνολογία. Κάποιες τέτοιες εφαρμογές είναι τα Συστήματα Διαχείρισης Περιεχομένου (CMS – Content Management System). Τα CMS είναι εργαλεία τα οποία μας βοηθούν στη γρήγορη κατασκευή ενός απλού ιστότοπου σε ένα πιο οικείο περιβάλλον προγραμματισμού σε σχέση με τον παραδοσιακό τρόπο, ο οποίος απευθύνεται ακόμα και σε άτομα τα οποία δεν έχουν επαφή με τον προγραμματισμό. Κάποια CMS εργαλεία είναι Wordpress, Drupal, Joomla, Modx κ.α.

(<https://el.wikipedia.org/wiki/PHP>, n.d.)

6.3.5 MYSQL

Πριν διατυπώσουμε τι είναι η MYSQL πρέπει να κάνουμε μια αναφορά στην έννοια *βάσεις δεδομένων*. Βάσεις δεδομένων είναι ένα σύστημα το οποίο κρατά αποθηκευμένο περιεχόμενο το οποίο είναι έτοιμο να προσπελαστεί από περισσότερα από ένα άτομο, να προστεθεί περιεχόμενο και να ενημερωθεί με έναν εύκολο τρόπο. Δεν μπορούμε να διαχειριστούμε από οπουδήποτε αυτά τα δεδομένα, με τη βοήθεια ενός συστήματος διαχείρισης βάσεων δεδομένων (DataBase Management Systems - DBMS) η διαχείριση γίνεται εύκολη υπόθεση με ασφάλεια και με έναν φιλικό προς τον χρήστη πλέον τρόπο. Τώρα μπορούμε να δώσουμε περισσότερες λεπτομέρειες σχετικά με την MYSQL.

Η MYSQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων . Η γλώσσα SQL(Structured Query Language) αναπτύχθηκε για να υπάρχει ένα κοινό πρωτόκολλο επικοινωνίας μεταξύ του συστήματος διαχείρισης και του προγραμματιστή για να μπορεί με εύκολο τρόπο να προσπελάσει τα δεδομένα.

Με λίγα λόγια η MYSQL αποτελεί μια έκδοση της SQL, εντάσσεται στα ελεύθερα λογισμικά και χρησιμοποιείται τόσο από ερασιτέχνες όσο και από επαγγελματίες σε αυτόν τον χώρο. (<https://el.wikipedia.org/wiki/MySQL>, n.d.)

6.4 Framework

6.4.1 Τι είναι τα Frameworks

Τα frameworks είναι κάποιες βιβλιοθήκες οι οποίες είναι κατάλληλες για την ανάπτυξη διαδικτυακών εφαρμογών. Στην περίπτωση που η διαδικτυακή εφαρμογή μας απαιτεί βάση δεδομένων, φόρμες, sessions, cookies και αυθεντικοποίηση ακόμα και από εξωτερικές υπηρεσίες όπως Facebook, twitter, google account τότε η χρήση framework κρίνεται απαραίτητη για την ανάπτυξη της εφαρμογής μας. (<https://stigma.host/why-you-should-be-using-a-php-framework/>, n.d.)

6.4.2 Χαρακτηριστικά

Κάποια χαρακτηριστικά των frameworks είναι:

- Ο έλεγχος λειτουργίας-εκτέλεσης της διαδικτυακής εφαρμογής.
- Κοινή δομή κλάσεων-αρχείων κώδικα με σκοπό την επαναχρησιμοποίηση του.
- Εύκολη αλλαγή, μεταφορά δεδομένων από μια βάση δεδομένων σε μια άλλη.
- Εύκολη χρήση Cache με ποιο γενικές κλάσεις συμβατές με μηχανές για cache όπως Memcached(είναι ένα σύστημα απομνημόνευσης αντικειμένων μνήμης υψηλής απόδοσης, που έχει σκοπό την επιτάχυνση των δυναμικών εφαρμογών ιστού με την ελάφρυνση του φορτίου βάσης δεδομένων) και APC.
- Σωστή διαχείριση φορμών επιτρέπουν στον developer τη δημιουργία, την πιστοποίηση των δεδομένων και τη διασφάλιση της ασφάλειας των δεδομένων που δίνει ο χρήστης.
- Παροχή έτοιμου συστήματος αυθεντικοποίησης η οποία μπορεί να διαχειριστεί τη σύνδεση και αποσύνδεση του χρήστη με εύκολο τρόπο καθώς επίσης και τη δημιουργία του στη βάση δεδομένων.
- Ευκολία στην απασφαλμάτωση καθώς υπάρχει έτοιμο σύστημα το οποίο αναγνωρίζει το οποιοδήποτε σφάλμα στον κώδικα και μας ενημερώνει.

6.4.3 Πλεονεκτήματα

Τα πλεονεκτήματα της χρήσης των frameworks είναι:

- Φορητότητα
- Ταχύτερη ανάπτυξη
- Ασφάλεια της εφαρμογής
- Υποστήριξη από την κοινότητα πίσω από κάθε framework.
- Χρήση καλών μοντέλων προγραμματισμού.(MVC)
- Ευκολία στον κώδικα

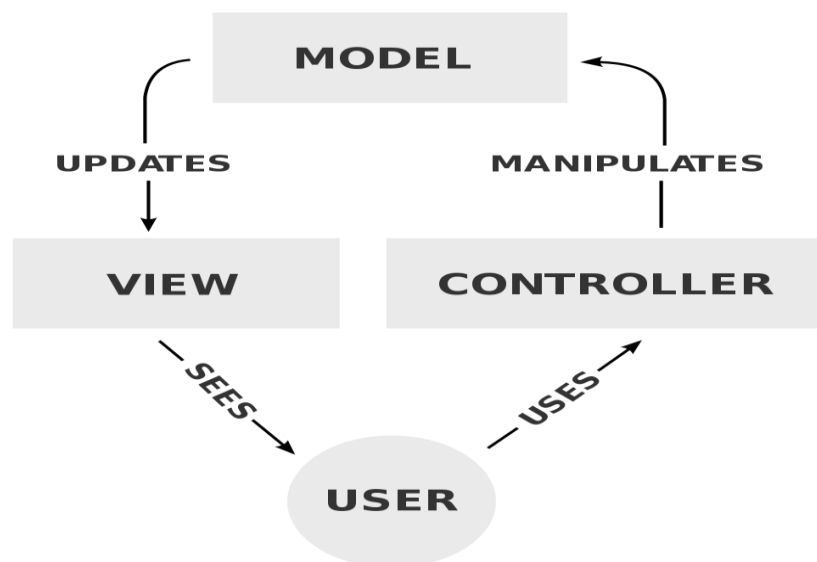
6.4.4 Λειτουργία

Αξίζει να δώσουμε βάση στο μοντέλο προγραμματισμού που αναφερθήκαμε παραπάνω, καθώς είναι ένα πολύ βασικό χαρακτηριστικό το οποίο χρησιμοποιούν τα περισσότερα frameworks.

Model View Controller (MVC) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού το οποίο συμβάλει στη δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Το μοντέλο αυτό διακρίνεται από τρία ξεχωριστά μέρη.

1. Κλάσεις για το model της βάσης δεδομένων.
2. Κλάσεις για τον Controller(αρχεία τα οποία φορτώνουν μια λειτουργία, υπηρεσία).
3. Κλάσεις για το template του τρόπου εμφάνισης.

Ο Controller φορτώνει το template(View) με σκοπό την εμφάνιση των δεδομένων της λειτουργίας – υπηρεσίας που έχει υλοποιηθεί.



Εικόνα 6.4.1: Διάγραμμα λειτουργίας του μοντέλου MVC.

6.4.4.1 Model

- Τα αρχεία στον φάκελο model είναι κλάσεις οι οποίες σχετίζονται με τις βάσεις δεδομένων.
- Όσους πίνακες έχουμε στη βάση δεδομένων τόσα είναι και τα αρχεία κλάσεων.
- Αυτά τα αρχεία περιέχουν μεθόδους σχετικά με κάποιες περιπτώσεις – αντικείμενα του πίνακα καθώς επίσης και με κάποιες ενέργειες οι οποίες ενδεχομένως να χρειαστούν(Select, Update, κλπ.).

6.4.4.2 *Controller*

- Όλα ξεκινούν από το url, δίνοντας ένα url καλούμε τη μέθοδο στον κατάλληλο controller. Υπεύθυνο για τη σωστή λειτουργία είναι το αρχείο routing το οποίο έχει σκοπό να καλέσει την κατάλληλη μέθοδο.
- Επίσης σε αυτό το αρχείο υπάρχουν τα actions.
- Για παράδειγμα εάν θέλουμε σε ένα action να εμφανίσουμε προϊόντα τότε δίνουμε μέσα στον controller την εντολή για να ανακτήσουμε τα δεδομένα από τη βάση δεδομένων και στη συνέχεια τα φορτώνουμε στο View.

6.4.4.3 *View*

- Σε αυτό το τμήμα των αρχείων περιέχεται το HTML αρχείο, δηλαδή η παρουσίαση μιας σελίδας.

Αναλυτικότερα για τη λειτουργία θα αναφερθούμε παρακάτω στην παρουσίαση της δικής μας εφαρμογής, η οποία αναπτύχθηκε για τις ανάγκες της πτυχιακής εργασίας.

Το framework το οποίο θα χρησιμοποιήσουμε στη δική μας εφαρμογή είναι το CodeIgniter.

(Χρήστος, 2011)

7 Υλοποίηση εφαρμογής

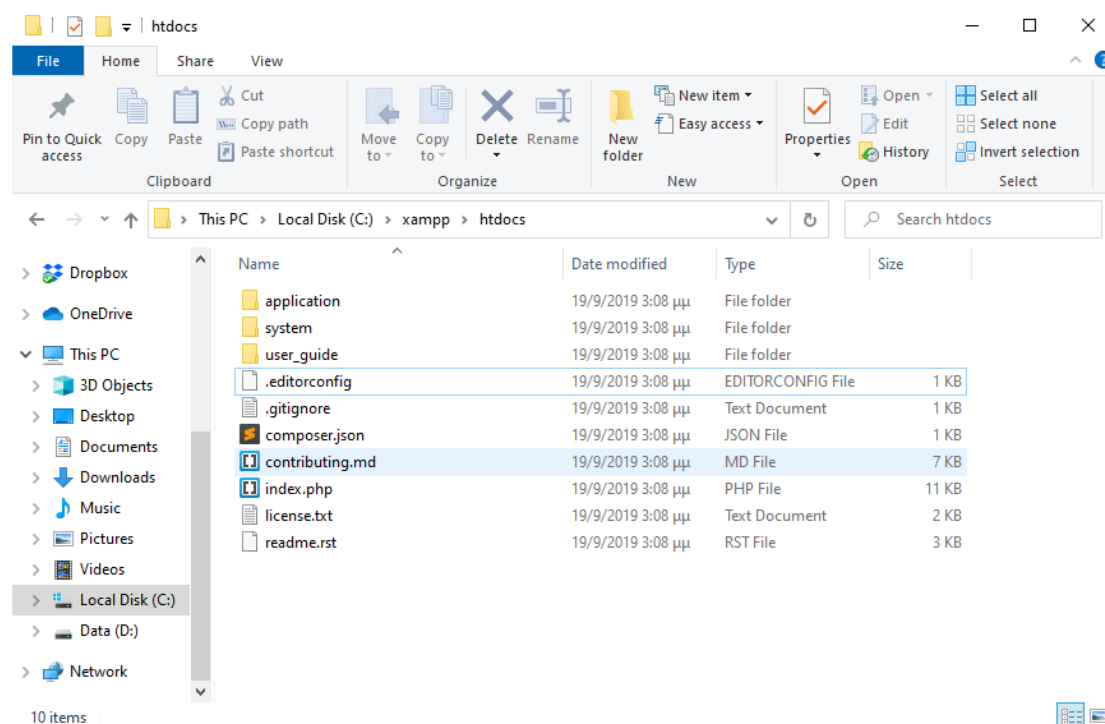
Παρακάτω θα δώσουμε πληροφορίες σχετικά με την εφαρμογή μας, τη δομή της, τη βάση δεδομένων, το framework που χρησιμοποιήσαμε για την υλοποίηση καθώς επίσης και πως χρησιμοποιούμε την εφαρμογή.

7.1 Εγκατάσταση framework

Για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε το CodeIgniter με σκοπό τη γρήγορη ανάπτυξη της εφαρμογής, παραπάνω πληροφορίες έχουμε αναφέρει στο προηγούμενο κεφάλαιο σχετικά με αυτό το framework.

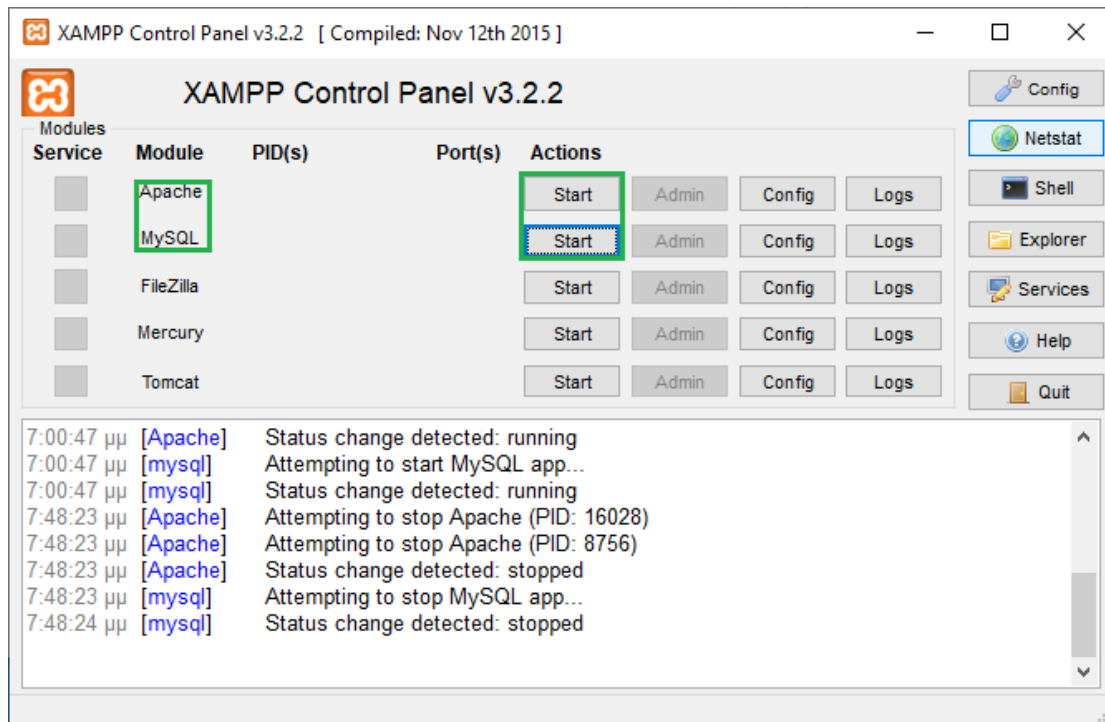
Αφού κατεβάσουμε και εγκαταστήσουμε το Xampp(www.apachefriends.org) το επόμενο βήμα είναι να κατεβάσουμε το CodeIgniter(www.codeigniter.com).

Έπειτα αφού έχει ολοκληρωθεί η λήψη του framework θα αποσυμπιέσουμε το περιεχόμενο που κατεβάσαμε στον φάκελο htdocs του Xampp.



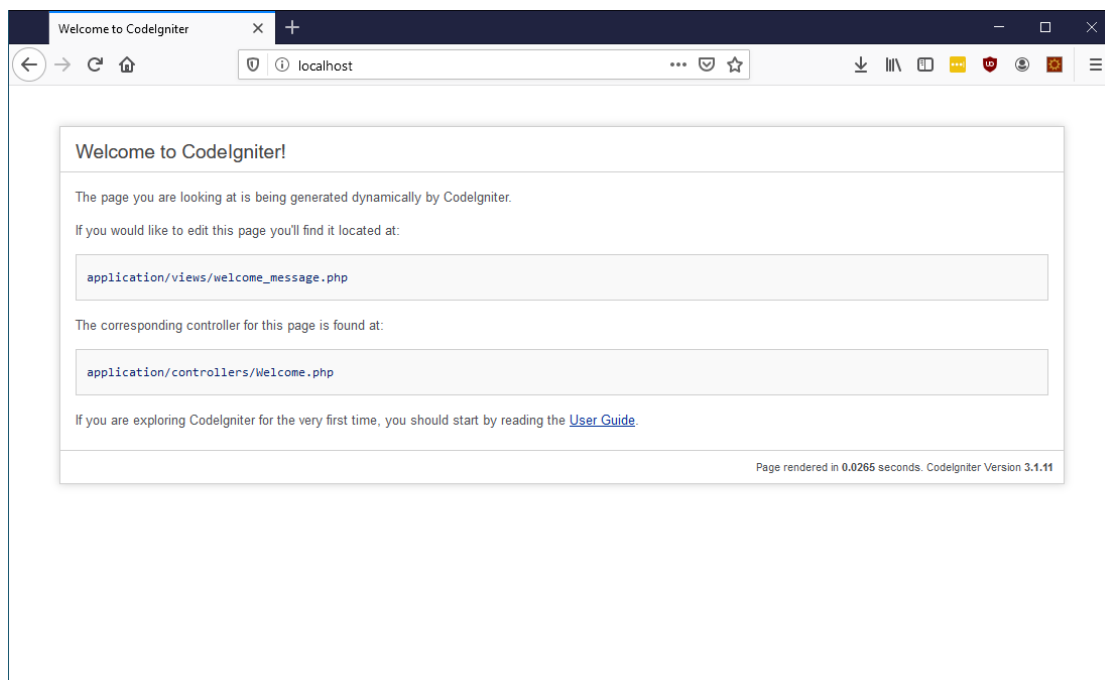
Εικόνα 7.1.1: Τοποθεσία φιλοξενίας των αρχείων της εφαρμογής μας.

Τέλος θα ελέγξουμε εάν λειτουργεί σωστά το framework μας. Πρώτα όμως πρέπει να εκκινήσουμε κάποια module που μας παρέχει το Xampp.



Εικόνα 7.1.2: Module τα οποία πρέπει να ενεργοποιήσουμε κάθε φορά που εκκινούμε το Xampp.

Πληκτρολογώντας σε έναν οποιονδήποτε φυλλομετρητή τη διεύθυνση localhost πρέπει να πάρουμε το παρακάτω αποτέλεσμα.



Εικόνα 7.1.3: Αποτέλεσμα που επιστρέφει το url: localhost μας διαβεβαιώνει πως το Xampp με το CodeIgniter λειτουργούν σωστά.

Το framework είναι έτοιμο προς χρήση.

7.1.1 Βασικές ρυθμίσεις

Πριν ξεκινήσουμε να προγραμματίζουμε πρέπει να κάνουμε κάποιες βασικές ρυθμίσεις στο CodeIgniter.

- Βήμα 1

```
$autoload['libraries'] = array('database');
```

Εικόνα 7.1.4: Στη διαδρομή `htdocs/application/config/autoload.php` κάνουμε την εξής αλλαγή.

- Βήμα 2

```
26 $config['base_url'] = '/localhost/';
27
28 /*
29 |-----|
30 | Index File
31 |-----|
32
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 |*/
38 $config['index_page'] = '';
39
```

Εικόνα 7.1.5: Στη διαδρομή `htdocs/application/config/config.php`

- Βήμα 3

```
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'test',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
```

Εικόνα 7.1.6: Στη διαδρομή `htdocs/application/config/database.php` συμπληρώνουμε τα στοιχεία σύνδεσης μας με τη βάση δεδομένων.

- Τέλος στη διαδρομή *htdocs/application/core* θα δημιουργήσουμε δύο αρχεία:

i) **MY_Controller** το οποίο μας παρέχει το μενού της εφαρμογής μας.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class MY_Controller extends CI_Controller {
    public $viewData = array();
    public function __construct()
    {
        parent::__construct();
        $this->viewData = [];
        //$this->getMenu();
        $this->load->library('session');
        $this->load->helper('url_helper');
        $this->viewData['baseUrl'] = $this->config->item('base_url');
    }

    public function getMenu()
    {
        if(isset($_SESSION['userid']))
        {
            $logged = true;
        }
        else
        {
            $logged = false;
        }

        $this->viewData['menu'] = [
            ['href' => 'dashboard', 'label' => 'Dashboard', 'icon' =>
            'wb-dashboard', 'modal' => false],
            ['href' => 'professors', 'label' => 'Καθηγητές', 'icon'
            => 'wb-user', 'modal' => false],
            ['href' => 'articles', 'label' => 'Επιστημονικό Έργο',
            'icon' => 'wb-book', 'modal' => false]
        ];

        $this->viewData['tags'] = ['url' => '', 'type'=>'', 'title'
        => '', 'description' => '', 'image' => ''];

        $this->viewData['nav'] = [['href' => 'blog', 'label' =>
        'Home']];
    }
}
```

ii) **MY_Modal** το οποίο μας παρέχει κάποιες βασικές κλάσεις για να παίρνουμε εύκολα τα αποτελέσματα, να αποθηκεύουμε και να κάνουμε διαγραφή από τη βάση δεδομένων μας.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/**
```


** This Class extends CI_Model class and is used in our Model classes for common tasks accross all application models.*

**/*

```
class MY_Model extends CI_Model {

    protected $table_name = 'undefined';

    public $id;
    public $inserted_at;
    public $updated_at;
    public $deleted_at;

    public function __construct($properties = [])
    {
        parent::__construct();
        $this->set_properties($properties);
        $this->load->database();
    }

    public function set_properties($properties = [])
    {
        foreach ($properties as $property => $value)
        {
            $this->$property = $value;
        }
    }

    private function nullify_empty_properties()
    {
        foreach (get_object_vars($this) as $property => $value)
        {
            if (strlen(trim($this->$property)) == 0)
            {
                $this->$property = NULL;
            }
        }
    }

    public function getRow($where = [])
    {
        if (!$where) return NULL;

        $called_class = get_called_class();

        return $this->db->get_where($this->table_name, $where)->custom_row_object(0, $called_class);
    }

    public function getRows($where = [], $order_by = NULL, $dir = 'ASC', $limit = NULL, $start = NULL)
    {
        $called_class = get_called_class();
        $where = array_merge($where, ['deleted_at' => NULL]);

        if ($order_by)
            $this->db->order_by($order_by, $dir);

        if (isset($limit) && isset($start))
            $this->db->limit($limit, $start);
    }
}
```

```

        return $this->db->get_where($this->table_name, $where) -
>custom_result_object($called_class);
    }

    public function countRows($where = [], $order_by = NULL, $dir =
'ASC', $limit = NULL, $start = NULL)
    {
        $where = array_merge($where, ['deleted_at' => NULL]);

        if ($order_by)
            $this->db->order_by($order_by, $dir);

        if (isset($limit) && isset($start))
            $this->db->limit($limit, $start);

        $this->db->where($where);
        $this->db->from($this->table_name);

        return $this->db->count_all_results();
    }

    public function save()
    {
        if ((int)$this->id > 0)
        {
            $this->un_delete();
            $this->update();
            $result = $this->id;
        }
        else
        {
            $this->insert();
            $result = $this->db->insert_id();
        }

        return $result;
    }

    public function update()
    {
        $this->nullify_empty_properties();

        $called_class = get_called_class();

        $datetime_now = new DateTime('now', new
DateTimeZone('Europe/Athens'));
        $this->updated_at = $datetime_now->format('Y-m-d H:i:s');

        $this->db->update($this->table_name, $this, ['id' => $this-
>id]);
    }

    public function insert()
    {
        $this->nullify_empty_properties();

        $called_class = get_called_class();

        $datetime_now = new DateTime('now', new
DateTimeZone('Europe/Athens'));

```

```

        if (!$this->inserted_at)
            $this->inserted_at = $datetime_now->format('Y-m-d
H:i:s');
        $this->updated_at = $datetime_now->format('Y-m-d H:i:s');

        $this->db->insert($this->table_name, $this);

        $this->id = $this->db->insert_id();
    }

    public function delete($where = null)
    {
        $called_class = get_called_class();

        if (!$where)
            $where = ['id' => $this->id];

        return $this->db->delete($this->table_name, $where);
    }

    public function soft_delete()
    {
        $called_class = get_called_class();
        $datetime_now = new DateTime('now', new
DateTimeZone('Europe/Athens'));
        $this->deleted_at_at = $datetime_now->format('Y-m-d H:i:s');

        return $this->db->update($this->table_name, $this, ['id' =>
$this->id]);
    }

    public function un_delete()
    {
        $called_class = get_called_class();
        $this->deleted_at = NULL;

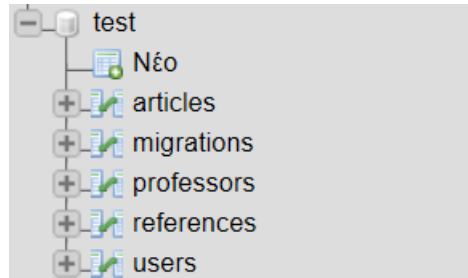
        return $this->db->update($this->table_name, $this, ['id' =>
$this->id]);
    }
}

```

7.2 Βάση δεδομένων

Κάθε εφαρμογή η οποία θεωρείται ολοκληρωμένη περιέχει και μια βάση δεδομένων έτσι ώστε να είναι ανεξάρτητη από εξωτερικές υπηρεσίες. Διαφορετικές υπηρεσίες οι οποίες προσφέρουν βάση δεδομένων είναι η Firebase της Google. Στην παρούσα υλοποίηση της εφαρμογής μας θα χρησιμοποιήσουμε το εργαλείο Xampp το οποίο μας παρέχει βάση δεδομένων MYSQL και των Apache server για να «σερβίρουμε» την εφαρμογή μας.

Όταν ξεκινάμε να γράφουμε τις περιπτώσεις χρήσης της εφαρμογής μας και αφού ολοκληρώσουμε τη διαδικασία καταγραφής όλων των περιπτώσεων χρήσης το επόμενο βήμα μας είναι να φτιάξουμε τη βάση δεδομένων. Η περιπτώσεις χρήσης θα μας βοηθήσουν να καταλάβουμε τι πεδία χρειαζόμαστε να έχουμε σε κάθε πίνακα της βάσης δεδομένων μας.



Εικόνα 7.2.1: Πίνακες της βάσης δεδομένων.

7.2.1 Migration

Το CodeIgniter μας περιέχει μια κλάση η οποία λέγεται Migration. Αυτή η κλάση μας παρέχει την εύκολη δημιουργία της βάσης δεδομένων αλλά και την τροποποίηση της βάσης μας. Τα βήματα και ο τρόπος λειτουργίας τον εξηγούμε παρακάτω.

```

$config['migration_enabled'] = TRUE;

/*
|-----
| Migration Type
|-----
|
| Migration file names may be based on a sequential identifier or on
| a timestamp. Options are:
|
| 'sequential' = Sequential migration naming (001_add_blog.php)
| 'timestamp'  = Timestamp migration naming (20121031104401_add_blog.php)
|               Use timestamp format YYYYMMDDHHIISS.
|
| Note: If this configuration value is missing the Migration library
| defaults to 'sequential' for backward compatibility with CI2.
|
*/
$config['migration_type'] = 'timestamp';

/*
|-----
| Migrations table
|-----
|
| This is the name of the table that will store the current migrations state.
| When migrations runs it will store in a database table which migration
| level the system is at. It then compares the migration level in this
| table to the $config['migration_version'] if they are not the same it
| will migrate up. This must be set.
|
*/
$config['migration_table'] = 'migrations';

/*
|-----
| Auto Migrate To Latest
|-----
|
| If this is set to TRUE when you load the migrations class and have
| $config['migration_enabled'] set to TRUE the system will auto migrate
| to your latest migration (whatever $config['migration_version'] is
| set to). This way you do not have to call migrations anywhere else
| in your code to have the latest migration.
|
*/
$config['migration_auto_latest'] = TRUE;

/*
|-----
| Migrations version
|-----
|
| This is used to set migration version that the file system should be on.
| If you run $this->migration->current() this is the version that schema will
| be upgraded / downgraded to.
|
*/
$config['migration_version'] = 20190701000035;

```

Εικόνα 7.2.2: Ενεργοποίηση migration στην εφαρμογή μας ακολουθώντας τη διαδρομή `htdocs/application/config/migration.php` και κάνοντας κάποιες αλλαγές όπως βλέπουμε στην παρακάτω εικόνα.

Αξίζει να αναφέρουμε περισσότερες λεπτομέρειες για τη γραμμή «migration_version». Αυτή η επιλογή μας λέει πιο αρχείο θα εκτελεστεί έτσι ώστε να δημιουργηθεί η βάση δεδομένων μας και έχει τη μορφή timestamp. Αφού αποθηκεύσουμε το αρχείο με τις αλλαγές πάμε να δημιουργήσουμε το αρχείο το οποίο περιέχει τον κώδικα για τη δημιουργία της βάσης μας και ακούει στο όνομα «20190701000035_initial_schema».

Το αρχείο αυτό θα το αποθηκεύσουμε στην εξής διαδρομή `htdocs/application/migrations/` με τον παρακάτω κώδικα.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Migration_Initial_schema extends CI_Migration {
    private $tables;
    public function __construct()
    {
        parent::__construct();
        $this->load->dbforge();
    }
    public function up()
    {
        // Create Users table
        $this->dbforge->add_field([
            'id' => [
                'type' => 'INT',
                'constraint' => 5,
                'unsigned' => TRUE,
                'auto_increment' => TRUE
            ],
            'email' => [
                'type' => 'VARCHAR',
                'constraint' => '50'
            ],
            'username' => [
                'type' => 'VARCHAR',
                'constraint' => '30'
            ],
            'firstname' => [
                'type' => 'VARCHAR',
                'constraint' => '30'
            ],
            'lastname' => [
                'type' => 'VARCHAR',
                'constraint' => '30'
            ],
            'password' => [
                'type' => 'VARCHAR',
                'constraint' => 50
            ],
            'inserted_at' => [
                'type' => 'date',
                'null' => TRUE
            ],
            'updated_at' => [
                'type' => 'date',
                'null' => TRUE
            ],
            'deleted_at' => [
                'type' => 'date',
                'null' => TRUE
            ]
        ]);
        $this->dbforge->add_key('id', TRUE);
        $this->dbforge->create_table('users');
```

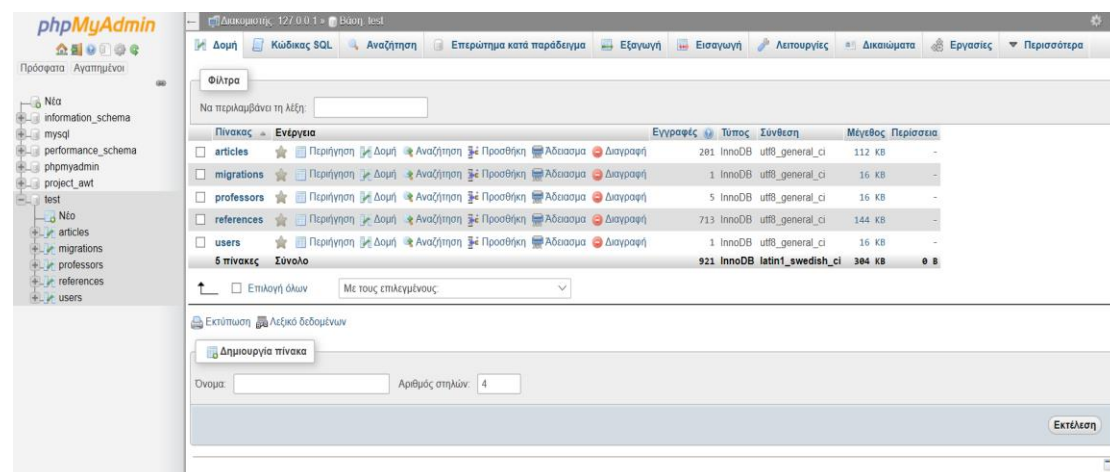
Ο παραπάνω κώδικας δημιουργεί τον πίνακα users αντίστοιχα δημιουργούμε και τους άλλους πίνακες, για λόγους συντομίας δεν θα περιλάβουμε των κώδικα για τους άλλους πίνακες καθώς μπορούμε να τον δούμε στα αρχεία της εφαρμογής μας. Αφού ολοκληρώσουμε και αυτό το αρχείο είμαστε έτοιμη να τρέξουμε την εντολή που θα μας δημιουργήσει τη βάση δεδομένων με τους πίνακες.

Στη διαδρομή htdocs/application/controllers θα δημιουργήσουμε ένα αρχείο Migrate.php. Όπως κάναμε αναφορά και στο προηγούμενο κεφάλαιο στον φάκελο Controllers περιέχονται τόσα αρχεία όσοι είναι οι πίνακες στη βάση δεδομένων καθώς επίσης και η βασική λειτουργία της κάθε σελίδας. Ο κώδικας του αρχείου μας είναι:

```
<?php
defined("BASEPATH") or exit("No direct script access allowed");
class Migrate extends CI_Controller
{
    public function index()
    {
        $this->load->library('migration');

        if ($this->migration->current() === FALSE)
        {
            show_error($this->migration->error_string());
        } else {
            echo "migration run successfully";
        }
    }
}
```

Οπότε για να εκτελέσουμε τη λειτουργία migrate με σκοπό τη δημιουργία της βάσης ηλεκτρολογούμε σε έναν φυλλομετρητή την εντολή *localhost/migrate* και οι πίνακες στη βάση δεδομένων μας είναι έτοιμες.



Εικόνα 7.2.3: Οι πίνακες στη βάση δεδομένων.

7.3 Controllers

Οι Controllers αποτελούν τη «καρδιά» της εφαρμογής μας, καθώς μπορούμε να καθορίσουμε τον τρόπο με τον οποίο θα χειριζόμαστε τις αιτήσεις HTTP. Ένας controller είναι ένα απλό αρχείο κλάσης το οποίο του δίνουμε όνομα με τρόπο που μπορεί να συσχετιστεί ένα URL.

Ένα παράδειγμα χρήσης URL είναι:

localhost/dashboard/

Στο παραπάνω παράδειγμα, ο CodeIgniter θα προσπαθήσει να βρει έναν controller με το όνομα Dashboard.php και να το φορτώσει.

Στην περίπτωση που θέλουμε να εκτελέσουμε μια άλλη function τότε δίνουμε τη διεύθυνση:

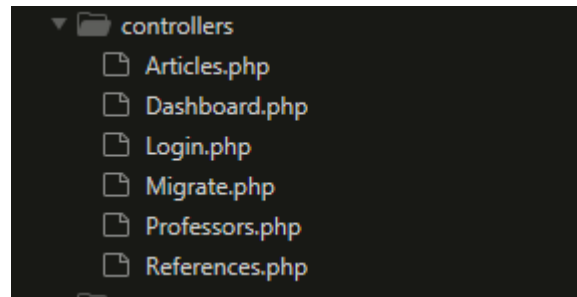
localhost/dashboard/anotherfunction

```
1 |<?php
2 | class Dashboard extends MY_Controller {
3 |
4 |     public function __construct()
5 |     {
6 |         parent::__construct();
7 |         $this->load->helper('url_helper');
8 |         $this->load->library('session');
9 |         $this->load->model('registered_model');
10 |        $this->load->model('article_model');
11 |
12 |        $this->getMenu();
13 |    }
14 |
15 |    public function index()
16 |    {
17 |
18 |        $query = $this->db->query("SELECT year FROM articles GROUP BY year");
19 |
20 |        $res = $query->result();
21 |        //$this->load->view('sign-in6');
22 |        $this->viewData['yearlist'] = $res;
23 |
24 |        $this->load->view('template/header', $this->viewData);
25 |        $this->load->view('dashboard');
26 |        $this->load->view('template/footer');
27 |    }
28 | }
29 |
```

Εικόνα 7.3.1: Κώδικας που εκτελείτε όταν βρει τον controller «dashboard» και από προεπιλογή θα εκτελέσει την function index.

Όπως βλέπουμε στην παραπάνω εικόνα στην function index αφού εκτελεστούν κάποια queries και στη γραμμή 22 φορτώσουμε τα αποτελέσματα σε ένα array, πρέπει να φορτώσουμε το view της σελίδας που θα αναπαραστήσουμε τα αποτελέσματα μας.

Για την υλοποίηση της εφαρμογής μας δημιουργήσαμε τους εξής Controllers:



Εικόνα 7.3.2: Επισκόπηση των Controllers που δημιουργήσαμε για την εφαρμογή μας.

7.4 Views

Τα Views είναι μια ιστοσελίδα ή κάποια κομμάτια σελίδας τα οποία ποτέ δεν φορτώνονται απευθείας παρά μόνο από τον Controller. Σε αυτό το σημείο αξίζει να αναφέρουμε πως στο πλαίσιο του MVC model, ο Controller λειτουργεί ως αστυνομικός κυκλοφορίας οπότε μπορούμε να καταλάβουμε τη σημαντικότητα που έχει ο Controller.

Τα αρχεία με τα Views τοποθετούνται στη διαδρομή `htdocs/application/views`. Δημιουργούμε ένα View με το όνομα `dashboard`. Ο κώδικάς μας είναι:

```
<!-- Page -->
<div class="page">
  <div class="page-header">
    <h1 class="page-title">Analytics Overview</h1>
  </div>

  <!-- Page Content -->
  <div class="page-content container-fluid">
    <div class="row" data-plugin="matchHeight" data-by-row="true">
      <div class="col-12">
        <!-- Product Overview Card -->
        <div id="productOverviewWidget" class="card card-shadow
card-md">
          <!-- The Header Of First Card -->
          <div class="card-header card-header-transparent py-20">
            <div class="btn-group dropdown">
              <a href="#" class="text-body dropdown-toggle blue-
grey-700 text-uppercase" data-toggle="dropdown">Product Sales</a>
              <div class="dropdown-menu animate" role="menu">
                <a class="dropdown-item" href="#"
role="menuitem">Sales</a>
                <a class="dropdown-item" href="#"
role="menuitem">Total sales</a>
                <a class="dropdown-item" href="#"
role="menuitem">Profit</a>
              </div>
            </div>
            <div class="card-header-actions">
```

```

        <ul class="nav nav-pills nav-pills-rounded product-
filters">
            <li class="nav-item">
                <a class="active nav-link" href="#scoreLineToDay"
data-toggle="tab">Day</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#scoreLineToWeek" data-
toggle="tab">Week</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#scoreLineToMonth"
data-toggle="tab">Month</a>
            </li>
        </ul>
    </div>
</div>
<!-- End The Header Of First Card -->

<!-- The Content Of First Card -->
<div class="card-block p-20">
    <div class="tab-content">
        <div class="ct-chart tab-pane active"
id="scoreLineToDay"></div>
        <div class="ct-chart tab-pane"
id="scoreLineToWeek"></div>
        <div class="ct-chart tab-pane"
id="scoreLineToMonth"></div>
    </div>
    <div id="productOptionsData" class="text-center">
        <div class="row no-space">
            <!-- The First Counter Block-->
            <div class="col-xl-3 col-md-6">
                <div class="counter">
                    <div class="counter-label">Vist</div>
                    <div class="counter-number-group text-
truncate">
                        <span class="counter-number-related red-
600">+</span>
                        <span class="counter-number">681</span>
                    </div>
                    <div class="ct-chart" data-counter-
type="productVist"></div>
                </div>
            </div>
            <!-- End The First Counter Block-->

            <!-- The Second Counter Block-->
            <div class="col-xl-3 col-md-6">
                <div class="counter">
                    <div class="counter-label">Unique Vistors</div>
                    <div class="counter-number-group text-
truncate">
                        <span class="counter-number-related green-
600">-</span>
                        <span class="counter-number">522</span>
                    </div>
                    <div class="ct-chart" data-counter-
type="productVistors"></div>
                </div>
            </div>
        </div>
    </div>

```

```

        <!-- End The Second Counter Block-->

        <!-- The Third Counter Block-->
        <div class="col-xl-3 col-md-6">
            <div class="counter">
                <div class="counter-label">Page Views</div>
                <div class="counter-number-group text-
truncate">
                    <span class="counter-number-related green-
600">--</span>
                    <span class="counter-number">1,622</span>
                </div>
                <div class="ct-chart" data-counter-
type="productPageViews"></div>
            </div>
        </div>
        <!-- End The Third Counter Block-->

        <!-- The fourth Counter Block-->
        <div class="col-xl-3 col-md-6">
            <div class="counter">
                <div class="counter-label">Bounce Rate</div>
                <div class="counter-number-group text-
truncate">
                    <span class="counter-number-related red-
600">+</span>
                    <span class="counter-number">843</span>
                </div>
                <div class="ct-chart" data-counter-
type="productBounceRate"></div>
            </div>
        </div>
        <!-- End The fourth Counter Block-->
    </div>
    </div>
    <!-- End The Content Of First Card -->
</div>
<!-- End Product Overview Card -->
</div>

<div class="col-xl-6 col-12">
    <!-- Example Line -->
    <div class="example-wrap">
        <h4 class="example-title">Ερευνητικό ανά έτος</h4>
        <p>Παρακάτω βλέπουμε το ερευνητικό έργο που έχει
ανέβει ανά έτος από το προσωπικό των καθηγητών του Πανεπιστημίου
μας</p>
        <div class="example text-center">
            <canvas id="chartByYear" height="300"
width="450"></canvas>
        </div>
    </div>
    <!-- End Example Line -->
</div>

<div class="col-xl-6 col-12">
    <!-- Example Pie -->
    <div class="example-wrap m-md-0">

```

```

<h4 class="example-title">ΕΡΓΟ ΠΟΥ ΕΧΕΙ ΠΑΡΑΧΘΕΙ ΤΗ
ΔΙΔΑΚΤΙΚΗ ΠΕΡΙΟΔΟ:

```

```

<form method="post">
  <select value="2019" class="custom-select"
onchange="graphCurrentYear(this.value)">
    <?php foreach($yearlist as $year): ?>
      <option value="<?= $year->year?"><?=
intval($year->year)?>-<?= intval($year->year) + 1 ?></a>
    <?php endforeach; ?>
  </select>
</form>
</h4>

```

<p>Pie charts are probably the most commonly used chart there are.

They are divided into segments, the arc of each segment shows

the proportional value of each piece of data.</p>

```

<div class="example text-center">
  <canvas id="chartByProfCurrentYear" height="300"
width="450"></canvas>
</div>
</div>
<!-- End Example Pie -->
</div>

```

```

<div class="col-xl-6 col-12">
  <!-- Example Bar -->
  <div class="example-wrap">
    <h4 class="example-title">ΕΡΕΥΝΗΝΗΤΙΚΟ ΕΡΓΟ ΑΝΑ ΜΗΝΑ
ΤΟΥ ΤΩΡΙΝΟΥ ΕΤΟΥΣ</h4>

```

<p>Παρακάτω βλέπουμε το ερευνητικό έργο που έχει ανέβει ανα μήνα του έτους που διανύουμε απο το προσωπικό των καθηγητών του Πανεπιστημίου μας.</p>

```

  <div class="example text-center">
    <canvas id="chartByMonth" height="300"
width="450"></canvas>
  </div>
</div>
<!-- End Example Bar -->
</div>

```

```

</div>
</div>
<!-- End Page Content -->
</div>
<!-- End Page -->

```

Για να φορτώσουμε ένα View πάμε στον Controller που κάναμε αναφορά παραπάνω

```

1 |<?php
2 |class Dashboard extends MY_Controller {
3 |
4 |    public function __construct()
5 |    {
6 |        parent::__construct();
7 |        $this->load->helper('url_helper');
8 |        $this->load->library('session');
9 |        $this->load->model('registered_model');
10 |        $this->load->model('article_model');
11 |
12 |        $this->getMenu();
13 |    }
14 |
15 |    public function index()
16 |    {
17 |
18 |        $query = $this->db->query("SELECT year FROM articles GROUP BY year");
19 |
20 |        $res = $query->result();
21 |        //$this->load->view('sign-in6');
22 |        $this->viewData['yearlist'] = $res;
23 |
24 |        $this->load->view('template/header', $this->viewData);
25 |        $this->load->view('dashboard');
26 |        $this->load->view('template/footer');
27 |    }
28 |}
29 |

```

Εικόνα 7.4.1: Στη γραμμή 24 με 26 φορτώνουμε τα views της κεντρικής σελίδας.

Και με τη βοήθεια της μεθόδου view που περιλαμβάνεται στο CodeIgniter ειδικότερα στη γραμμή 25 δίνοντας το όνομα του View που δημιουργήσαμε παραπάνω φορτώνουμε το View της σελίδας που επιθυμούμε στέλνοντας τα data που έχουμε αποθηκεύσει στο array “yearlist”.

Όταν στέλνουμε ένα array στο view στέλνουμε αρκετά δεδομένα τα οποία περιέχονται μέσα στο array με αποτέλεσμα να χρειαζόμαστε να κάνουμε μια επαναληπτική διαδικασία στο View που θα τα φορτώσουμε έτσι ώστε να μπορέσουμε να τα απεικονίσουμε όλα τα δεδομένα μας. Αυτό το πετυχαίνουμε με τη βοήθεια της Foreach η οποία αυτό που κάνει είναι όσο υπάρχουν δεδομένα τρέχει έναν κώδικα τον οποίο θα του ορίσουμε εμείς.

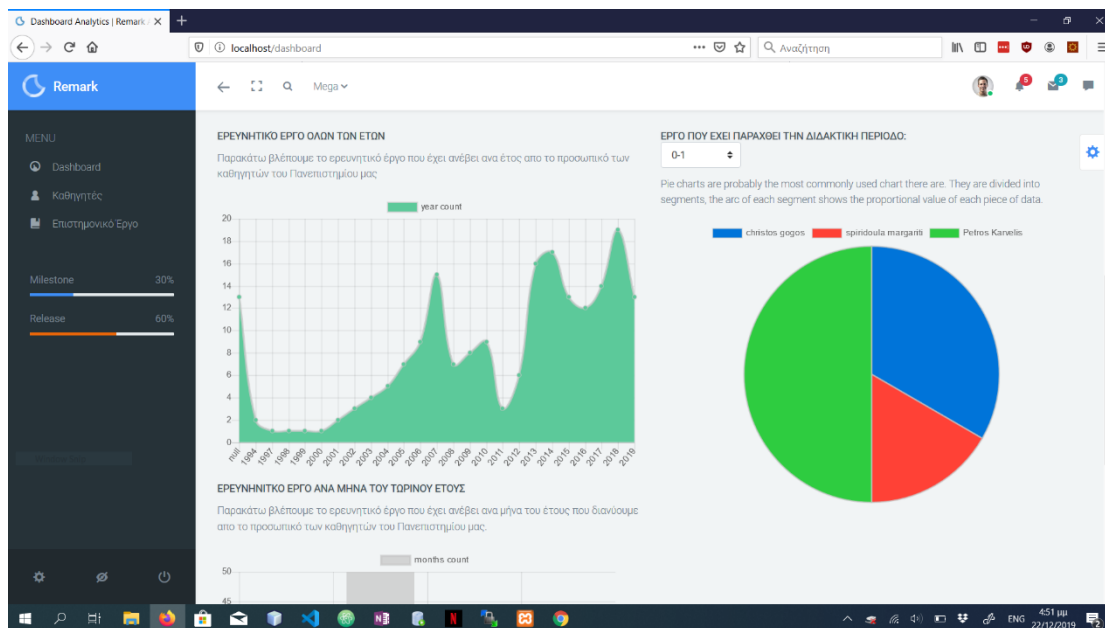
```

<?php foreach($yearlist as $year): ?>
    <option value="<?= $year->year?>"><?= intval($year->year)?>-<?= intval($year->year) + 1 ?></a>
<?php endforeach; ?>

```

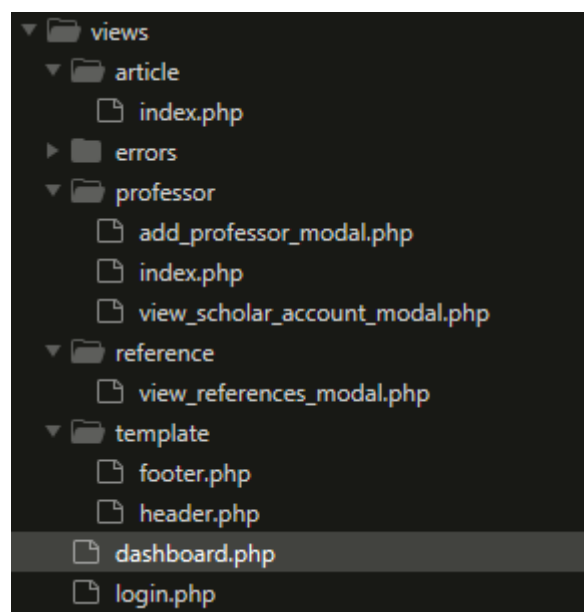
Εικόνα 7.4.2: PHP κώδικας όπου με τη βοήθεια της foreach «εκτυπώνουμε» όλα τα δεδομένα που περιέχονται μέσα σε ένα array. Το παραπάνω κομμάτι περιέχεται μέσα σε κώδικα HTML.

Σε αυτή τη φάση δίνοντας στο url ενός φυλλομετρητή τη διεύθυνση localhost/dashboard φορτώνουμε το View της σελίδας μας με τα δεδομένα που θέλουμε να απεικονίσουμε. Το αποτέλεσμα μετά από αυτήν την υλοποίηση θα είναι:



Εικόνα 7.4.3: Κεντρική σελίδα της εφαρμογής μας όπου απεικονίζει δύο διαγράμματα.

Για την υλοποίηση της εφαρμογής μας δημιουργήσαμε τα εξής Views:



Εικόνα 7.4.4: Views τα οποία περιέχονται στην εφαρμογή μας.

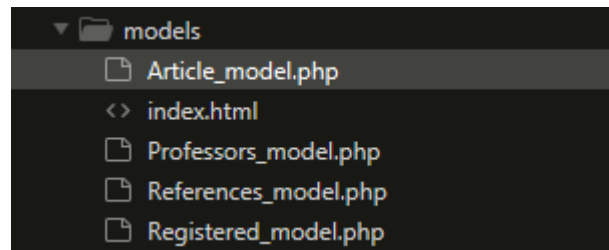
7.5 Models

Τα Models θα λέγαμε πως μας βοηθούν έτσι ώστε να επικοινωνήσουμε με τη βάση δεδομένων με σκοπό την ανάκτηση, διαγραφή, τροποποίηση δεδομένων. Τα models σύμφωνα με το CodeIgniter τα αποθηκεύουμε στη διαδρομή `htdocs/application/model` και είναι στο πλήθος όσοι είναι και οι πίνακες στη βάση δεδομένων μας.

```
1 <?php
2 class Article_model extends MY_Model
3 {
4     protected $table_name = 'articles';
5     public $title;
6     public $publication_date;
7     public $month;
8     public $year;
9     public $type_name;
10    public $type;
11    public $issue;
12    public $pages;
13    public $volume;
14    public $publisher;
15    public $description;
16    public $references_id;
17    public $citation;
18    public $scholarid_professor;
19    public $scholarid_article;
20
21
22
23    function __construct($properties = [])
24    {
25        parent::__construct($properties);
26    }
27
28
29
30    public function get_total_articles()
31    {
32        $query = $this->db->select("COUNT(*) as num")->get("articles");
33        $result = $query->row();
34        if(isset($result)) return $result->num;
35        return 0;
36    }
37
38    public function get_articles($start, $length, $order, $dir)
39    {
40        if($order != null) {
41            $this->db->order_by($order, $dir);
42        }
43
44        return $this->db
45            ->limit($length,$start)
46            ->get("articles");
47    }
48
49
50 }
```

Εικόνα 7.5.1: Παράδειγμα ενός model το οποίο επικοινωνεί με τον πίνακα Articles.

Για την υλοποίηση της εφαρμογής μας και σύμφωνα με τη βάση δεδομένων μας δημιουργήσαμε τα εξής Models:



Εικόνα 7.5.2: Όλα τα *models* τα οποία περιέχονται στην εφαρμογή μας.

7.6 Libraries

Το CodeIgniter μας παρέχει κάποιες βιβλιοθήκες, εν συντομία κάποιες από αυτές είναι:

- [Benchmarking Class](#)
- [Caching Driver](#)
- [Calendaring Class](#)
- [Shopping Cart Class](#)
- [Config Class](#)
- [Email Class](#)
- [Encrypt Class](#)
- [Encryption Library](#)
- [File Uploading Class](#)
- [Form Validation](#)
- [FTP Class](#)
- [Image Manipulation Class](#)
- [Input Class](#)
- [Javascript Class](#)
- [Language Class](#)
- [Loader Class](#)
- [Migrations Class](#)
- [Output Class](#)
- [Pagination Class](#)
- [Template Parser Class](#)
- [Security Class](#)
- [Session Library](#)
- [HTML Table Class](#)
- [Trackback Class](#)
- [Typography Class](#)
- [Unit Testing Class](#)
- [URI Class](#)
- [User Agent Class](#)
- [XML-RPC and XML-RPC Server Classes](#)

- [Zip Encoding Class](#)

(https://codeigniter.com/user_guide/libraries/index.html, n.d.)

Στην περίπτωση που δεν μας καλύπτει κάποια από τις παραπάνω που αναφέραμε μπορούμε να προσθέσουμε μια βιβλιοθήκη τρίτου ή ακόμα να φτιάξουμε και τη δική μας. Παρακάτω θα κάνουμε αναφορά και στις δύο περιπτώσεις καθώς πρέπει να προσθέσουμε στην εφαρμογή μας τη δυνατότητα του web scraping.

Αρχικά πρέπει να κατεβάσουμε μια βιβλιοθήκη που θα μας βοηθήσει να κάνουμε web scraping βάσει ενός δοθέντος URL. Το εργαλείο που είναι χρήσιμο γι' αυτή τη λειτουργία όπως αναφέραμε και στο προηγούμενο κεφάλαιο είναι το PHP Simple HTML Dom Parser(<https://sourceforge.net/projects/simplehtmldom>). Αφού το κατεβάσουμε και κάνουμε αποσυμπίεση τον φάκελο ψάχνουμε το αρχείο «simple_html_dom.php» και το αντιγράφουμε μέσα στην εφαρμογή μας και ειδικότερα στη διαδρομή htdocs/application/libraries. Από εδώ και πέρα μπορούμε να χρησιμοποιήσουμε κάποιες κλάσεις που μας παρέχει η συγκεκριμένη βιβλιοθήκη για να κάνουμε web scraping.

Έπειτα πρέπει να δημιουργήσουμε τη δική μας βιβλιοθήκη μέσα στην οποία θα υπάρχει όλη η λογική του web scraping στην οποία θα χρησιμοποιούμε τις κλάσεις που μας παρέχει πλέον η βιβλιοθήκη που προσθέσαμε παραπάνω. Στην ίδια διαδρομή htdocs/application/libraries θα δημιουργήσουμε ένα αρχείο mylib.php με κάποιες βασικές γραμμές κώδικα που πρέπει να εμπεριέχονται.

```
1  <?php
2
3  include_once('application/libraries/simple_html_dom.php');
4
5  defined('BASEPATH') OR exit('No direct script access allowed');|
6
7
8  class Mylib {
9
10     public function __construct()
11     {
12         $CI = & get_instance();
13         $CI->load->helper('url');
14     }
15
```

Εικόνα 7.6.1: Βασικός κώδικας που πρέπει να περιέχεται μέσα στη κλάση mylib.php

Στη γραμμή 3 δηλώνουμε που βρίσκετε η βιβλιοθήκη που κατεβάσαμε και προσθέσαμε καθώς όπως είπαμε και παραπάνω θα χρησιμοποιήσουμε κάποιες βοηθητικές κλάσεις. Παρακάτω έχουμε έναν κατασκευαστή ο οποίος θα καλείται κάθε φορά που θα καλούμε μια κλάση από τη βιβλιοθήκη μας. Αυτές τις κλάσεις που θα δημιουργήσουμε μέσα στη δικιά μας βιβλιοθήκη θα τις καλούμε μόνο μέσα από τους Controllers.

7.7 Web scraping

Όλη η διαδικασία του web scraping είναι μέσα στο αρχείο mylib.php που φτιάξαμε προηγουμένως. Όταν χρειαζόμαστε δεδομένα από το Google Scholar τότε μέσω του controller θα καλούμε την κατάλληλη function που έχουμε γράψει στο αρχείο mylib.php.

Παρακάτω θα αναλύσουμε μια function η οποία μας επιστρέφει μια λίστα από καθηγητές εάν υπάρχουν αρκετοί με το ίδιο όνομα σε διαφορετική περίπτωση θα μας επιστρέψει έναν καθηγητή δηλαδή αυτόν που μας ενδιαφέρει έτσι ώστε να ανακτήσουμε το Scholar Id του καθηγητή που θέλουμε να ανακτήσουμε τα άρθρα του. Σε κάθε έναν καθηγητή αντιστοιχεί ένα Id, η εύρεση των άρθρων του στην πλατφόρμα δεν γίνεται με το όνομα του αλλά με το Id που έχει.

```
public function prof_list_scraping($url)
{
    $i=0;
    start:
    $ch = curl_init();

    // set options
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_PROXY, proxyIp());
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false); // read more
    about HTTPS http://stackoverflow.com/questions/31162706/how-to-scrape-a-ssl-or-https-url/31164409#31164409
    curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U;
    Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311
    Firefox/2.0.0.13');

    // $output contains the output string
    $str = curl_exec($ch);
    $error_msg = curl_error($ch);
    if($error_msg == "Proxy CONNECT aborted") goto start;
    if(strpos($error_msg, 'Timed out') !== false) goto start;
    if(strpos($error_msg, '502') !== false) goto start;
    if(strpos($error_msg, '503') !== false) goto start;
    if(strpos($error_msg, '400') !== false) goto start;
    if(strpos($error_msg, '404') !== false) goto start;
    if(strpos($error_msg, '401') !== false) goto start;
    if(strpos($error_msg, '514') !== false) goto start;
    if(strpos($error_msg, '403') !== false) goto start;
    if(strpos($error_msg, 'failure') !== false) goto start;
    if(strpos($error_msg, 'SSL_ERROR_SYSCALL') !== false) goto
start;
    if(strpos($error_msg, 'Failed') !== false) goto start;
    if(strpos($error_msg, 'The document has moved') !== false)
goto start;
    echo "CURL ERROR Message: ".var_dump($error_msg);
    echo "<br>";
    echo "Returned result: ".var_dump($str);
    curl_close($ch);
}
```

```

$html_base = new simple_html_dom();

$html_base->load($str);
$prof = [];
foreach($html_base->find('.gsc_lusr') as $key => $element)
{
    $temp = [];
    //echo $element->plaintext . '<br>';
    $tds = $element->find('.gs_ai_t');
    foreach ($tds as $idx => $td) {
        switch ($idx)
        {
            case 0:
                $contents = $td->find('a');
                $temp['name'] = $contents[0]->plaintext;
                $link = trim($contents[0]-
>href, '/citations?hl=en&user=!');
                $temp['scholarid'] = trim($link, 'mp;user=');
                if (!$contents)
                    break;
            case 1:
                $contents = $td->find('.gs_ai_eml');
                $trimmed = trim($contents[0]->plaintext, "Η
διεύθυνση ηλεκτρονικού ταχυδρομείου έχει επαληθευτεί στον τομέα .");
                $temp['description'] = $trimmed;
                //$temp['description'] = $contents[0]-
>plaintext;
                if (!$contents)
                    break;
            case 2:
                $contents = $td->find('.gs_ai_cby');
                preg_match_all('!\d+', $contents[0],
$matches);
                $temp['citation'] = $matches[0];
                if (!$contents)
                    break;
        }
    }
    $prof[$i] = $temp;
    $i++;
}
$html_base->clear();
unset($html_base);
return $prof;
}

```

Όπως βλέπουμε στο παραπάνω κομμάτι κώδικα έχουμε μια function την οποία την έχουμε ονομάσει prof_list_scraping και έχει ένα όρισμα με το όνομα url. Άρα όταν θέλουμε να ανακτήσουμε τη λίστα των καθηγητών με σκοπό να επιλέξουμε το κατάλληλο Scholar Id το κάνουμε όπως βλέπουμε στην παρακάτω εικόνα.

```

43 public function insert_scholar_account_modal()
44 {
45     $firstname = $_GET['firstname'];
46     $lastname = $_GET['lastname'];
47     $id = $_GET['id'];
48
49     $url = 'https://scholar.google.gr/citations?hl=el&view_op=search_authors&mauthors=';
50     $name = $firstname.'+'.$lastname;
51     $stripped_name = preg_replace('/\s+/', '+', $name);
52     $name = mb_convert_encoding(urldecode($stripped_name), 'HTML-ENTITIES', 'UTF-8');
53
54     $url = $url.$stripped_name;
55
56     $this->viewData['title'] = 'Εισαγωγή Scholar Id';
57
58     $this->load->library('mylib');
59
60     $this->viewData['profScholarId'] = $this->mylib->prof_list_scraping($url);
61
62     $this->viewData['id'] = $id;
63
64
65     $this->load->view('professor/view_scholar_account_modal', $this->viewData);
66 }

```

Εικόνα 7.7.1: Στη γραμμή 60 βλέπουμε την κλείσει της function που φτιάξαμε η οποία περιέχεται μέσα στη βιβλιοθήκη mylib.php

Όπου URL είναι μια σειρά λέξεων συνδεδεμένα μεταξύ τους με χαρακτήρες και στο τέλος το όνομα του καθηγητή. Αυτό το URL θα είναι το όρισμα στην function που θα καλέσουμε. Αυτό το URL δεν το σκεφτήκαμε εμείς αλλά κάναμε μια ανάλυση έτσι ώστε να δούμε με ποιόν τρόπο ζητάμε από το Google Scholar να μας φέρει πληροφορίες για έναν καθηγητή. Αφού κάνουμε την κλήση της function μας στη γραμμή 60 πάμε να δούμε τι γίνεται σε αυτήν.

Επιστρέφουμε λοιπόν στο κομμάτι κώδικα που παραθέσαμε πάνω από την εικόνα. Όπως βλέπουμε αρχικοποιούμε μια συνεδρία curl, curl είναι η συντομογραφία των λέξεων command line tool και είναι μια τεχνολογία η οποία χρησιμοποιείται σε γραμμές εντολών ή σεναρίων για τη μεταφορά δεδομένων. Θα καταλαβαίναμε απολύτως κάποιον ο οποίος μας ρωτούσε γιατί να χρησιμοποιήσουμε αυτή τη τεχνολογία;

Αυτή την τεχνολογία τη χρησιμοποιούμε έτσι ώστε αντί να φορτώσουμε το περιεχόμενο που θέλουμε να ανακτήσουμε από το ίντερνετ με τη δική μας IP address με τη βοήθεια της βιβλιοθήκης simple_html_dom και την function που μας παρέχει file_get_html(\$url), το κάνουμε με τη χρήση proxy server έτσι ώστε σε περίπτωση που κάνουμε πάνω από έναν αριθμό αιτημάτων στην Google να μην μας θεωρήσει κακόβουλο χρήστη και μας περιορίσει την πρόσβαση.

```

38 // set options
39 curl_setopt($ch, CURLOPT_URL, $url);
40 curl_setopt($ch, CURLOPT_PROXY, proxyIp());
41 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
42 curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
43 curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');

```

Εικόνα 7.7.2: Χρήση της τεχνολογίας CURL.

Στα set options της τεχνολογίας curl βλέπουμε ότι στην πρώτη γραμμή φορτώνουμε το url που θέλουμε να φορτώσουμε έπειτα καλούμε μια function την proxyIp() η οποία μας επιστρέφει μια IP address και ένα port έτσι ώστε να ανακτήσουμε τη σελίδα μέσω

αυτής της διεύθυνσης και παρακάτω κάποιες άλλες παραμέτρους. Οπότε σε κάθε request που κάνουμε γίνεται μέσω αυτών των γραμμών.

```
7 function proxyIp()
8 {
9     var_dump("hello from proxy");
10    $url = 'http://pubproxy.com/api/proxy?api=N2JVT3ZYeXByd1ZEWlVwSVN2d29kZz09&google=true';
11    $curl = curl_init($url);
12    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
13    $response = curl_exec($curl);
14    curl_close($curl);
15    $res = json_decode($response);
16    $res = $res->data[0]->ipPort;
17    var_dump($res);
18    return $res;
19 }
```

Εικόνα 7.7.3: function η οποία μας επιστρέφει κάθε φορά μια διαφορετική διεύθυνση.

Στη γραμμή 10 όπως βλέπουμε χρησιμοποιούμε μια υπηρεσία η οποία μας παρέχει αυτές τις IP address παρέχοντάς μας ένα API έτσι ώστε με μια γραμμή να παίρνουμε κάποιες πληροφορίες σχετικά με τον server που έχουμε συνδεθεί πέρα από την IP address που χρειαζόμαστε και το port. Στις παρακάτω γραμμές απλά κάνουμε τον διαχωρισμό αυτών των πληροφοριών έτσι ώστε να πάρουμε μόνο αυτές που χρειάζεται η τεχνολογία curl έτσι ώστε να μας επιστρέψει το αποτέλεσμα μέσα από τον απομακρυσμένο server.

```
48 if($error_msg == "Proxy CONNECT aborted") goto start;
49 if(strpos($error_msg, 'Timed out') !== false) goto start;
50 if(strpos($error_msg, '502') !== false) goto start;
51 if(strpos($error_msg, '503') !== false) goto start;
52 if(strpos($error_msg, '400') !== false) goto start;
53 if(strpos($error_msg, '404') !== false) goto start;
54 if(strpos($error_msg, '401') !== false) goto start;
55 if(strpos($error_msg, '514') !== false) goto start;
56 if(strpos($error_msg, '403') !== false) goto start;
57 if(strpos($error_msg, 'failure') !== false) goto start;
58 if(strpos($error_msg, 'SSL_ERROR_SYSCALL') !== false) goto start;
59 if(strpos($error_msg, 'Failed') !== false) goto start;
60 if(strpos($error_msg, 'The document has moved') !== false) goto start;
61 echo "CURL ERROR Message: ".var_dump($error_msg);
62 echo "<br>";
63 echo "Returned result: ".var_dump($str);
64 curl_close($ch);
```

Εικόνα 7.7.4: Κομμάτι κώδικα της function *prof_list_scraping*.

Βλέπουμε πως γίνεται ένας έλεγχος έτσι ώστε εάν γίνει μια προσπάθεια ανάκτησης μέσω του proxy server και μας επιστρέψει κάποιο σφάλμα ή έναν κωδικό σφάλματος να ξανατρέξει ο κώδικας από την αρχή μέχρι να επιστρέψει επιτυχώς τα δεδομένα της σελίδας.

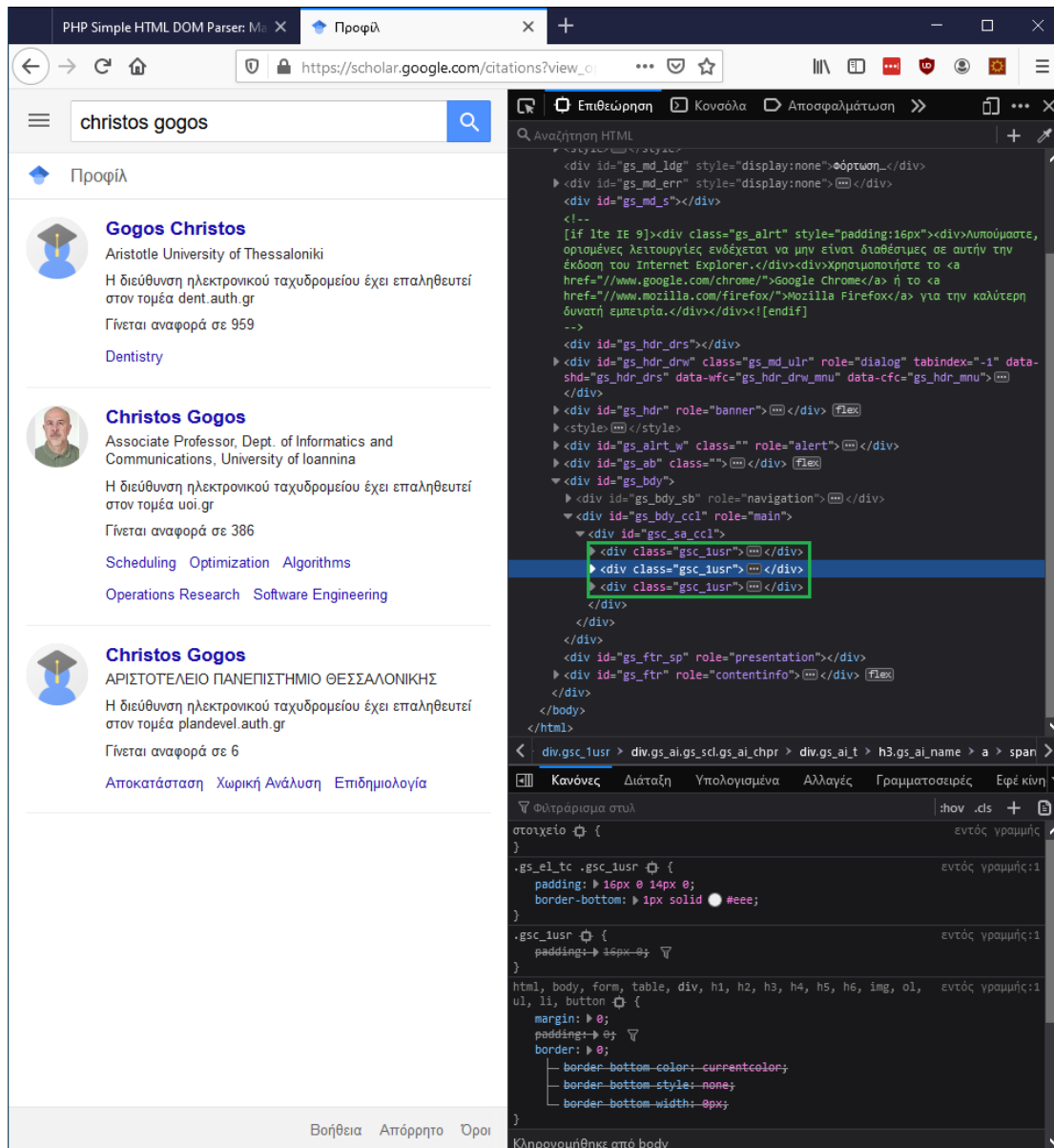
Στη συνέχεια αφού έχουμε ανακτήσει το περιεχόμενο που έχουμε ζητήσει(url) μέσω proxy server πρέπει να διαλέξουμε τις πληροφορίες που χρειαζόμαστε να εμφανίσουμε στην εφαρμογή μας έτσι ώστε ο χρήστης να έχει ένα φιλικό περιβάλλον επιλογής του καθηγητή που επιθυμεί.

```
66 $html_base = new simple_html_dom();
67
68 $html_base->load($str);
69 $prof = [];
70 foreach($html_base->find('.gsc_lusr') as $key => $element)
71 {
72     $temp = [];
73     //echo $element->plaintext . '<br>';
74     $tds = $element->find('.gs_ai_t');
75     foreach ($tds as $idx => $td) {
76         switch ($idx)
77         {
78             case 0:
79                 $contents = $td->find('a');
80                 $temp['name'] = $contents[0]->plaintext;
81                 $link = trim($contents[0]->href, '/citations?hl=el&user=!');
82                 $temp['scholarid'] = trim($link, 'mp;user=');
83                 if (!$contents)
84                     break;
85             case 1:
86                 $contents = $td->find('.gs_ai_eml');
87                 $trimmed = trim($contents[0]->plaintext, "Η διεύθυνση ηλεκτρονικού ταχυδρομείου έχει επαληθευτεί στον τομέα .");
88                 $temp['description'] = $trimmed;
89                 // $temp['description'] = $contents[0]->plaintext;
90                 if (!$contents)
91                     break;
92             case 2:
93                 $contents = $td->find('.gs_ai_cby');
94                 preg_match_all('/\d+!', $contents[0], $matches);
95                 $temp['citation'] = $matches[0];
96                 if (!$contents)
97                     break;
98         }
99     }
100     $prof[$i] = $temp;
101     $i++;
102 }
103 $html_base->clear();
104 unset($html_base);
105 return $prof;
106 }
107 }
```

Εικόνα 7.7.5: Κώδικας ο οποίος αφού έχουν ανακτηθεί όλοι η σελίδα html που ζητήσαμε επιλέγουμε τις πληροφορίες που μας ενδιαφέρουν.

Έπειτα στη γραμμή 66 δημιουργούμε ένα καινούριο αντικείμενο simple_html_dom στη μεταβλητή html_base και στη συνέχεια της φορτώνουμε το περιεχόμενο που ανακτήσαμε από το web scraping που κάναμε στη γραμμή 68.

Στη μεταβλητή html_base πλέον πρέπει να επιλέξουμε τους καθηγητές που θα μας επιστρέψει η λίστα στο Google Scholar. Λεπτομερέστερα εμείς έχουμε αποθηκεύσει στη μεταβλητή html_base τον κώδικα αυτής της σελίδας που βλέπουμε παρακάτω.



Εικόνα 7.7.6: Πεδία από τα οποία θα ανακτήσουμε τις πληροφορίες που μας είναι χρήσιμες.

Τα αποτελέσματα είναι περισσότερα από ένα οπότε πρέπει με τη βοήθεια της `foreach` να φτιάχνουμε μια επαναληπτική διαδικασία έτσι ώστε να επιλέξουμε τις πληροφορίες και από τα 3 αποτελέσματα με σκοπό να επιλέξουμε τον καθηγητή που επιθυμούμε. Η κλάση με τις πληροφορίες των αποτελεσμάτων είναι η «`gsc_usr`» οπότε βάση αυτής της κλάσεις θα γίνουν οι επαναλήψεις. Για να βρούμε τώρα αυτές τις κλάσεις μέσα στη μεταβλητή με τον κώδικά μας(`html_base`) θα χρησιμοποιήσουμε μια μέθοδο που παρέχεται από τη βιβλιοθήκη `simple html dom` και αυτή είναι η `find` όπως βλέπουμε στη γραμμή 70.

Αντίστοιχα θα γίνει ακόμα μια `foreach` με την κλάση `gs_ai_t` έτσι ώστε να βρεθούν οι πληροφορίες που χρειαζόμαστε συγκεκριμένα μέσα στην κλάση όπως βλέπουμε στη γραμμή 75. Αφού βρούμε τις πληροφορίες που θέλουμε τις αποθηκεύουμε σε ένα `array`

και το επιστρέφουμε στον Controller έτσι ώστε εκεί να αποφασίσουμε τι θα κάνουμε αυτήν την πληροφορία.

The screenshot shows a web browser window with the URL `https://scholar.google.com/citations?view_o...`. The page displays a Google Scholar profile for **Christos Gogos**. The profile information includes:

- Gogos Christos**: Aristotle University of Thessaloniki. H διεύθυνση ηλεκτρονικού ταχυδρομείου έχει επαληθευτεί στον τομέα `dent.auth.gr`. Γίνεται αναφορά σε 959. Dentistry.
- Christos Gogos**: Associate Professor, Dept. of Informatics and Communications, University of Ioannina. H διεύθυνση ηλεκτρονικού ταχυδρομείου έχει επαληθευτεί στον τομέα `uoι.gr`. Γίνεται αναφορά σε 386. Scheduling Optimization Algorithms, Operations Research, Software Engineering.
- Christos Gogos**: ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ. H διεύθυνση ηλεκτρονικού ταχυδρομείου έχει επαληθευτεί στον τομέα `plandevel.auth.gr`. Γίνεται αναφορά σε 6. Αποκατάσταση, Χωρική Ανάλυση, Επιδημιολογία.

The right sidebar shows the HTML source code of the page. A green box highlights the HTML structure for the first entry, which includes the name, affiliation, and citation count. The bottom of the page shows a CSS style sheet for the profile entries.

Εικόνα 7.7.7: Πεδία με την πληροφορία που θα ανακτήσουμε.

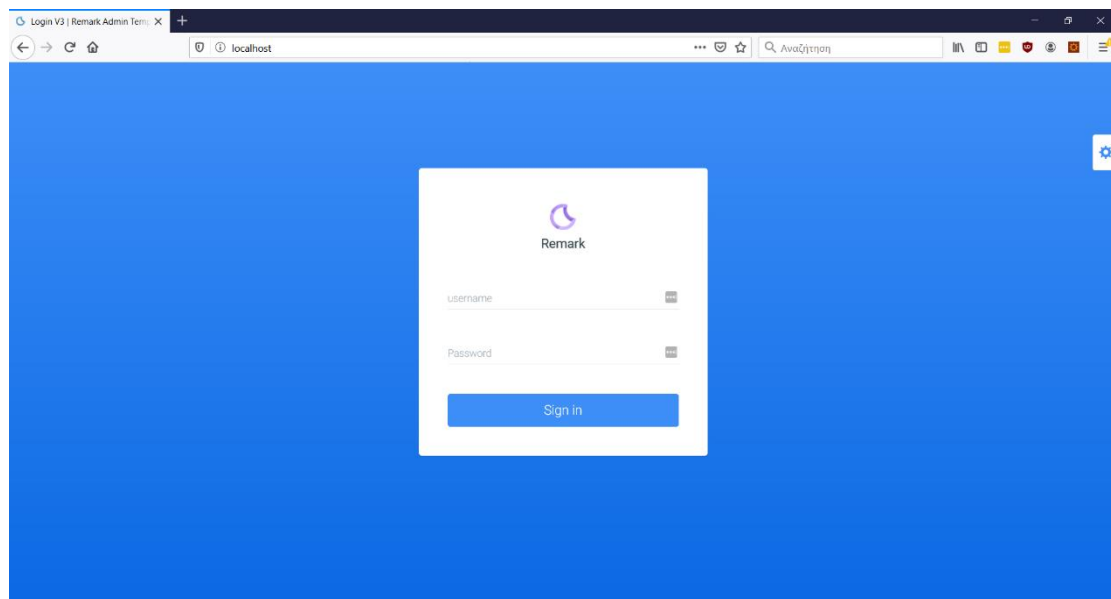
8 Παρουσίαση εφαρμογής

8.1 Εισαγωγή

Στο παρόν κεφάλαιο θα κάνουμε μια επίδειξη της εφαρμογής μας και των λειτουργιών της. Όπως θα δούμε η εφαρμογή περιέχει 4 σελίδες οι οποίες μας δίνουν διάφορες πληροφορίες για τα άρθρα, αναφορές των άρθρων και τα στατιστικά τα οποία προκύπτουν ανά ακαδημαϊκή περίοδο. Ο κώδικας της εφαρμογής υπάρχει αναρτημένος στο GitHub. (<https://github.com/pavlosti/ptyxiaki>)

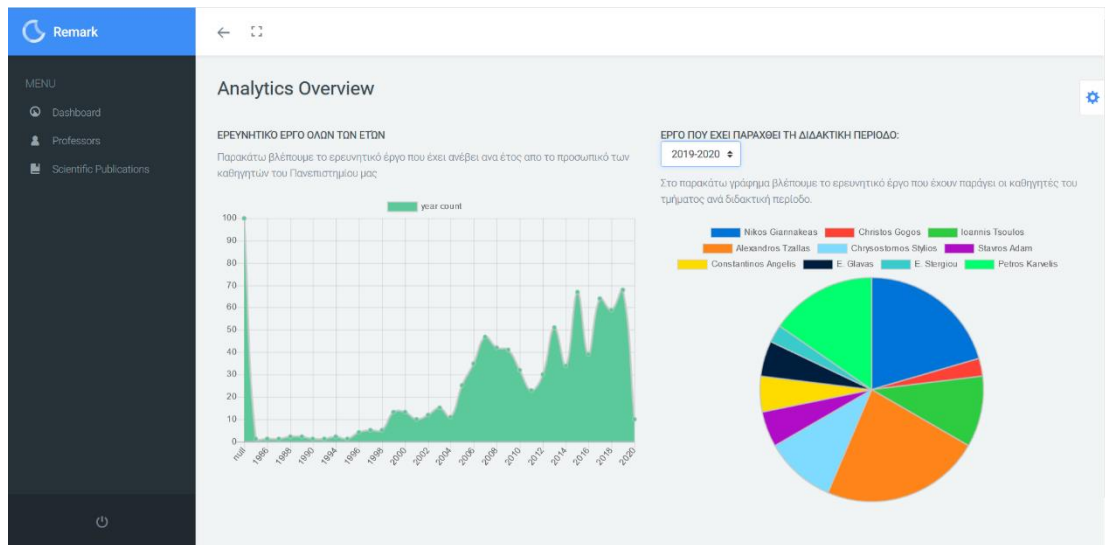
8.2 Αυθεντικοποίηση χρήστη

Αρχικά η εφαρμογή μας περιέχει σύστημα αυθεντικοποίησης χρήστη έτσι ώστε να μην έχει ο οποιοσδήποτε πρόσβαση στα δεδομένα της εφαρμογής σε περίπτωση που τρέχει εκτός τοπικού δικτύου. Πατώντας τη διεύθυνση localhost στον φυλλομετρητή της επιλογής μας εμφανίζεται το σύστημα αυθεντικοποίησης της εφαρμογής προτρέποντας τον χρήστη να βάλει τα στοιχεία του για να συνεχίσει.



Εικόνα 8.2.1: Σελίδα αυθεντικοποίησης χρήστη.

Αφού ο χρήστης πληκτρολογήσει τα σωστά διαπιστευτήρια τότε θα γίνει ανακατεύθυνση στην κεντρική σελίδα της εφαρμογής μας.



Εικόνα 8.2.2: Κεντρική σελίδα της εφαρμογής με τα στατιστικά στοιχεία τα οποία έχουν ανακτηθεί από τη βάση δεδομένων.

8.3 Εισαγωγή καθηγητή

Αριστερά έχουμε το μενού. Στην επιλογή «Καθηγητές» έχουμε μια λίστα με τους καθηγητές που υπάρχουν στη βάση δεδομένων και κουμπί που μας δίνει τη δυνατότητα να προσθέσουμε έναν καθηγητή.

Remark

MENU

Dashboard

Professors

Scientific Publications

Professors

+ Add professor

+ Update All

Name	Surname	Email	Citation(db)	Citation(web)	ScholarId(STATUS)	Update	Last Update	Articles/References	Action
Stavros	Adam	adamsp@uoi.gr	54	54		Article listReference listReference single	2020-02-25 / 2020-02-21		
Alexandros	Tzallas	tzallas@uoi.gr		2574		Article listReference listReference single	2020-02-17 /		
Chrysostomos	Stylios	stylios@uoi.gr		5792		Article listReference listReference single	2020-02-18 /		
Christos	Gogos	cgogos@uoi.gr		395		Article listReference listReference single	2020-02-17 /		
Spiridoula	Marganti	smargan@uoi.gr	109	109		Article listReference listReference single	2020-02-18 / 2020-02-25		
Constantinos	Angelis	kangelis@uoi.gr	465	465		Article listReference listReference single	2020-02-18 / 2020-02-24		
Nikos	Giannakeas	giannakeas@telep.gr	177	342		Article listReference listReference single	2020-02-19 / 2020-02-25		
Ioannis	Tsoulos	a@ia.com	231	1056		Article listReference listReference single	2020-02-19 / 2020-02-25		
E.	Glavas	a@ia.com		543		Article listReference listReference single	2020-02-19 /		
Petros	Kanelis	a@ia.com		643		Article listReference listReference single	2020-02-19 /		

Εικόνα 8.3.1: Σελίδα κατοχυρωμένων καθηγητών στη βάση δεδομένων.

Πατώντας “Add professor” ανοίγει ένα modal παροτρύνοντας μας να δώσουμε κάποια στοιχεία σχετικά με τον καθηγητή που θέλουμε να προσθέσουμε στη βάση.

Add
professor

Name:

Lastname:

Email:

Insert Professor

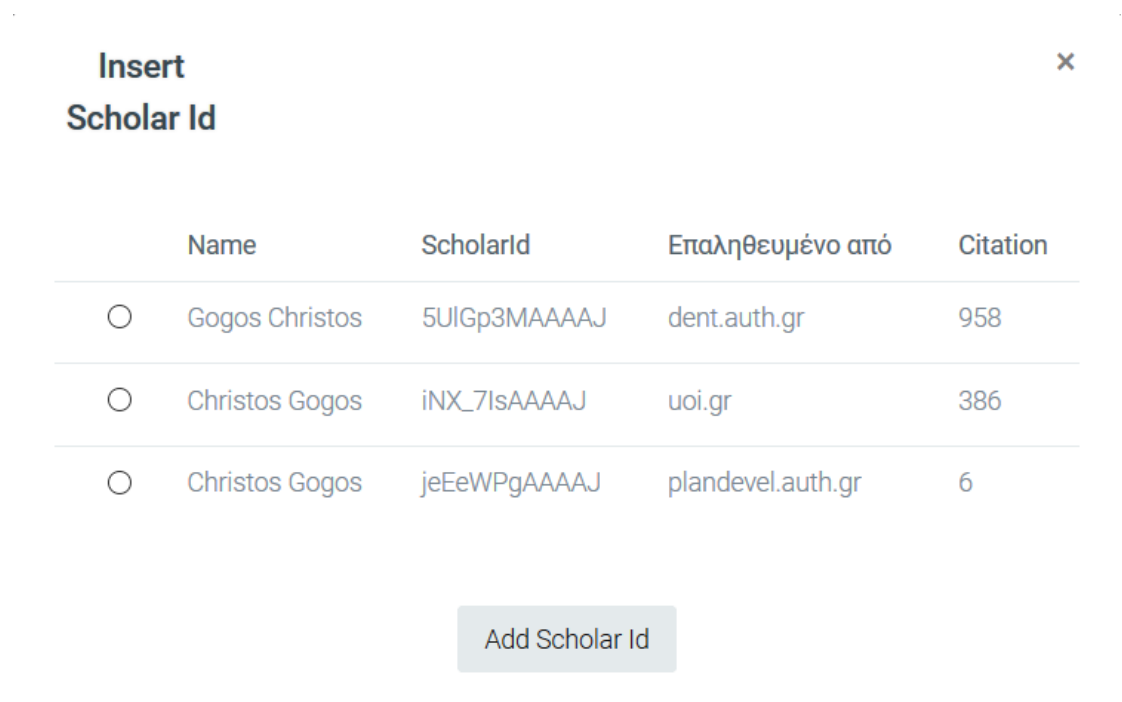
Εικόνα 8.3.2: Modal με τα πεδία του για την προσθήκη ενός καθηγητή στη βάση δεδομένων.

Αφού πληκτρολογήσουμε τα στοιχεία του κατευθείαν προστίθενται στη λίστα ο καθηγητής. Σε αυτό το σημείο βλέπουμε στη στήλη ScholarId ένα καπελάκι μαύρο, το οποίο σημαίνει πως δεν έχουμε συσχετίσει τον καθηγητή μας με το ScholarId που δίνει το Google Scholar.

Καθηγητές								
+ Add professor		+ Sync						
Όνομα	Επίθετο	Email	Citation(db)	Citation(web)	ScholarId(Status)	Update	Last Update Articles/References	Action
christos	gogos	cgogos@uoi.gr				Article list Reference list Reference single	/	

Εικόνα 8.3.3: Εισαγωγή καθηγητή στη λίστα χωρίς Scholar Id στη βάση δεδομένων.

Πατώντας το μαύρο καπελάκι θα γίνει web scraping και θα ανοίξει ένα modal με τους καθηγητές τους οποίους βρήκε το Google Scholar με το ίδιο όνομα.



Εικόνα 8.3.4: Πατώντας το καπελάκι ανοίγει ένα modal με τους καθηγητές που έχουν το ίδιο όνομα, παρέχοντας μας κάποιες παραπάνω πληροφορίες για τη σωστή επιλογή του καθηγητή που μας ενδιαφέρει.

Από το email το οποίο έχει δώσει ο καθένας μπορούμε να καταλάβουμε πιο αποτέλεσμα αντιστοιχεί στον καθηγητή που θέλουμε να εισάγουμε στην εφαρμογή μας. Αφού επιλέξουμε και πατήσουμε «Add Scholar Id» το modal θα κλείσει και το καπελάκι στη λίστα θα γίνει γαλάζιο.

Τέλος κάνοντας αριστερό διπλό κλικ στο όνομα ή επίθετο ή email μπορούμε να τροποποιήσουμε αυτά τα πεδία στη βάση δεδομένων σε περίπτωση που κάνουμε κάποιο λάθος ή ενημερωθεί το πεδίο email.

8.4 Εισαγωγή δεδομένων

Έχουμε καταχωρήσει τον καθηγητή στη βάση δεδομένων, πλέον μπορούμε να συνεχίσουμε τις διαδικασίες έτσι ώστε να ανακτήσουμε τις πληροφορίες των άρθρων από την πλατφόρμα Google Scholar. Σε κάθε γραμμή η οποία αντιπροσωπεύει από έναν καθηγητή στη βάση υπάρχουν 3 κουμπιά με τις εξής λειτουργίες:

1. Article list: Ανακτά μια λίστα με όλα τα άρθρα του καθηγητή και στη συνέχεια τα αποθηκεύει στη βάση δεδομένων.
2. Reference list: Ανακτά όλες τις αναφορές από κάθε ένα άρθρο ξεχωριστά και αποθηκεύει το id της αναφοράς.

3. Reference single: Βάση των αποθηκευμένων id τα οποία τα ανακτήσαμε από το προηγούμενο κουμπί σε αυτό θα φέρουμε όλες τις πληροφορίες για κάθε μια αναφορά η οποία δεν έχει περισσότερες πληροφορίες πέρα του id της.

Καθηγητές

+ Add professor + Sync

Όνομα	Επίθετο	Email	Citation(db)	Citation(web)	ScholarId(STATUS)	Update	Last Update Articles/References	Action
christos	gogos	cgogos@uoi.gr		386		Article list Reference list Reference single	2019-12-26 /	

Εικόνα 8.4.1: Καταχωρημένο Scholar Id το διακρίνουμε από το μπλε καπελάκι και τρεις επιλογές που μπορούμε να κάνουμε για έναν καθηγητή.

8.5 Επισκόπηση ερευνητικού έργου

Έχουμε προσθέσει καθηγητή, έχουμε κάνει scraping τα άρθρα του, τις αναφορές του. Ήρθε η στιγμή να παρουσιάσουμε τη σελίδα στην οποία μπορούμε να δούμε σε έναν πίνακα όλο το ερευνητικό έργο με τις αναφορές που υπάρχει στη βάση δεδομένων, να κάνουμε αναζήτηση ενός έργου και αναζήτηση βάση ενός συγκεκριμένου καθηγητή.

Scientific Publications

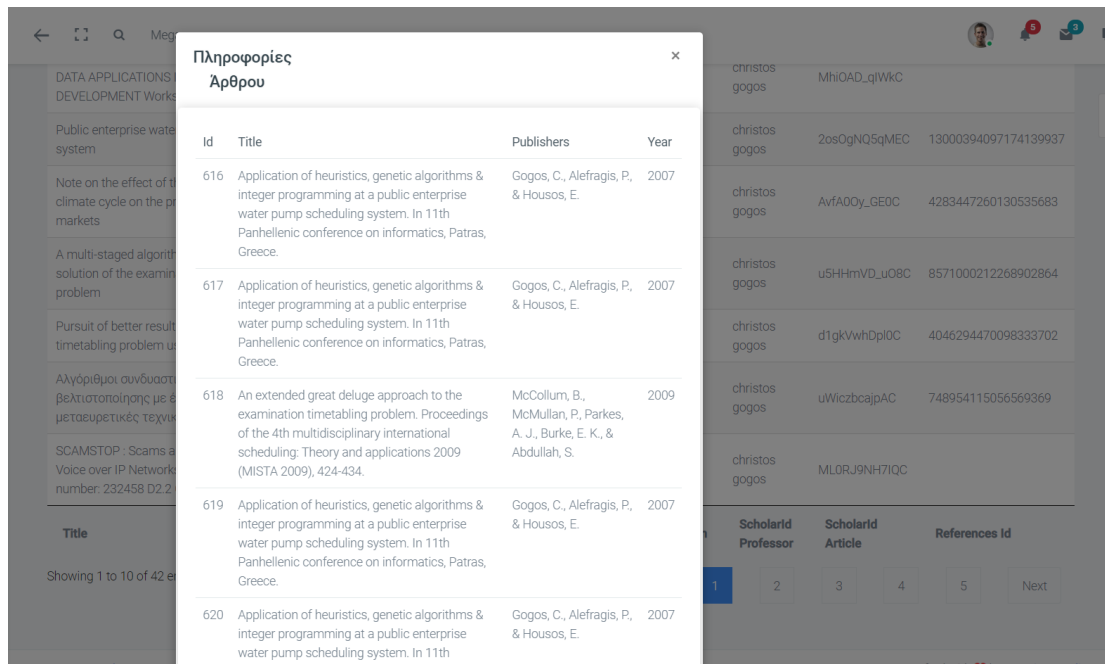
Καθηγητές ▾

Show 10 entries

Search:

Title	Publication Date	Type	Type_name	Citation	ScholarId Professor	ScholarId Article	References Id
Epileptic seizure detection in EEGs using time-frequency analysis	2009/3/16	Journal	IEEE transactions on information technology in bio	514	Alexandros Tzallas	u-x6o8ySG0sC	12790897782239216084
Modeling complex systems using fuzzy cognitive maps	2004/1/14	Journal	IEEE Transactions on Systems, Man, and Cybernetics	365	Chrysostomos Stylios	u5HlHmVD_u08C	12769382570728669199
Automatic seizure detection based on time-frequency analysis and artificial neural networks	2007	Journal	Computational Intelligence and Neuroscience	360	Alexandros Tzallas	u5HlHmVD_u08C	3871572851409004544
Active Hebbian learning algorithm to train fuzzy cognitive maps	2004/11/1	Journal	International journal of approximate reasoning	313	Chrysostomos Stylios	d1gkVwhDpl0C	13953849852154320492
Fuzzy cognitive maps in modeling supervisory control systems	2000/1/1	Journal	Journal of Intelligent & Fuzzy Systems	250	Chrysostomos Stylios	u-x6o8ySG0sC	3163940555028318306
A fuzzy cognitive map approach to			Artificial intelligence in		Chrysostomos		

Εικόνα 8.5.1: Πίνακας με το ερευνητικό έργο και τις λεπτομέρειες του κάθε έργου που υπάρχει στη βάση δεδομένων.



Εικόνα 8.5.2: Στη στήλη *citation* πατώντας πάνω στον αριθμό μπορούμε να δούμε τις αναφορές του άρθρου με τις λεπτομέρειές του.

8.6 Στατιστικά στοιχεία

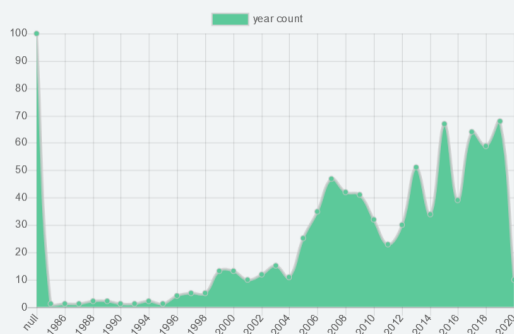
Ο κύριος λόγος ανάπτυξης της παρούσας εφαρμογής είναι τα στατιστικά στοιχεία τα οποία τα βρίσκουμε στην κεντρική σελίδα της εφαρμογής στη σελίδα dashboard. Η συγκεκριμένη σελίδα αποτελείται από δύο γραφικές παραστάσεις. Στην πρώτη γραφική παράσταση αριστερά βλέπουμε το ερευνητικό έργο όλων των ετών των άρθρων που υπάρχουν στη βάση δεδομένων. Δεξιά στο δεύτερο γράφημα βλέπουμε μια πίτα η οποία απεικονίζει το ερευνητικό έργο που έχουν παράγει οι καθηγητές ανά διδακτική περίοδο.

Analytics Overview



ΕΡΕΥΝΗΤΙΚΟ ΕΡΓΟ ΟΛΩΝ ΤΩΝ ΕΤΩΝ

Παρακάτω βλέπουμε το ερευνητικό έργο που έχει ανέβει ανά έτος από το προσωπικό των καθηγητών του Πανεπιστημίου μας

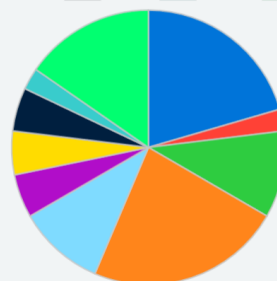


ΕΡΓΟ ΠΟΥ ΕΧΕΙ ΠΑΡΑΧΘΕΙ ΤΗ ΔΙΔΑΚΤΙΚΗ ΠΕΡΙΟΔΟ:

2019-2020

Στο παρακάτω γράφημα βλέπουμε το ερευνητικό έργο που έχουν παράγει οι καθηγητές του τμήματος ανά διδακτική περίοδο.

Nikos Giannakeas Christos Gogos Ioannis Tsoulos
Alexandros Tzallas Chrysostomos Stylios Stavros Adam
Constantinos Angelis E. Glavas E. Stergiou Petros Karvelis



Εικόνα 8.6.1: Απεικόνιση στατιστικών στοιχείων με τη βοήθεια διαγράμματος και πίτας.

9 Συμπεράσματα και μελλοντικές επεκτάσεις

Μετά τη συγγραφή των παραπάνω κεφαλαίων σε συνδυασμό με την ανάπτυξη της εφαρμογής έβγαλα τα εξής συμπεράσματα: Οι τεχνολογίες που χρησιμοποιήθηκαν στην εφαρμογή αποτελούν μια ολοκληρωμένη λύση ανάπτυξης ενός λογισμικού, με τη βασικότερη τεχνολογία να είναι η επιλογή του Framework που χρησιμοποιήσα(CodeIgniter) με σκοπό να μειώσω σημαντικά τον χρόνο ανάπτυξης μιας τέτοιας εφαρμογής.

Οι κύριες δυσκολίες που προέκυψαν στην παρούσα εφαρμογή ήταν η ανάκτηση πληροφοριών από το Google Scholar καθώς δεν παρέχεται κάποιο public API με σκοπό την εύκολη ανάκτηση και σωστή δομή των πληροφοριών από την πλατφόρμα. Επίσης μετά την ολοκλήρωση ενός συγκεκριμένου αριθμού αιτημάτων η πλατφόρμα μπλόκαρε την IP address του δικτύου από το οποίο γινόταν το Web Scraping. Τέλος ο συγκεκριμένος proxy server που χρησιμοποιήθηκε πολλές φορές μας επέστρεφε IP address όπου εξ αρχής ήταν μπλοκαρισμένες από την πλατφόρμα.

Πιθανές επεκτάσεις: Η εφαρμογή θα μπορούσε να περιέχει ρόλους χρηστών, δηλαδή θα είχε τη δυνατότητα κάθε ένας χρήστης(καθηγητές) να επεξεργαστεί τις πληροφορίες που αφορούν τον ίδιο σε αντίθεση με την τωρινή υλοποίηση όπου ο μόνος χρήστης της εφαρμογής είναι ο διαχειριστής. Επίσης στην περίπτωση που χρησιμοποιηθεί μια υπηρεσία proxy server η οποία επιστρέφει αποκλειστικά IP address οι οποίες δεν είναι μπλοκαρισμένες από την Google τότε θα μπορούσαμε να κάνουμε την εφαρμογή μας αυτόνομη από τον ανθρώπινο παράγοντα προγραμματίζοντας Cron Jobs έτσι ώστε να εκτελούνται όλες οι διαδικασίες αυτόματα με τον μόνο σκοπό της επίσκεψης του χρήστη να ήταν η ανάκτηση των στατιστικών στοιχείων ή αυτόματης αποστολής των δεδομένων μέσω email.

10 Αναφορές

Anon., 2007. <https://www.csd.uoc.gr>. [Ηλεκτρονικό]

Available at:

https://www.csd.uoc.gr/~hy351/2007/downloads/assisting_lectures/presentations/IS_351_UseCases_2007_11.pdf

Anon., χ.χ. <https://el.wikipedia.org>. [Ηλεκτρονικό]

Available at:

https://el.wikipedia.org/wiki/%CE%94%CE%B5%CE%AF%CE%BA%CF%84%CE%B7%CF%82_%CE%B5%CF%80%CE%B9%CF%83%CF%84%CE%B7%CE%BC%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE%CF%82_%CF%80%CE%BF%CE%B9%CF%8C%CF%84%CE%B7%CF%84%CE%B1%CF%82_h

Anon., χ.χ. <https://hellenictechnologies.com>. [Ηλεκτρονικό]

Available at: <https://hellenictechnologies.com/ti-einai-to-apis-kai-pos-chrisimopoietai/>

Anon., χ.χ. <https://help.apify.com/>. [Ηλεκτρονικό]

Available at: <https://help.apify.com/en/articles/1961361-several-tips-how-to-bypass-website-anti-scraping-protections>

https://codeigniter.com/user_guide/libraries/index.html, χ.χ. <https://codeigniter.com>. [Ηλεκτρονικό]

Available at: https://codeigniter.com/user_guide/libraries/index.html

<https://el.wikipedia.org/wiki/CSS>, χ.χ. s.l.: s.n.

<https://el.wikipedia.org/wiki/JavaScript>, χ.χ. s.l.: s.n.

<https://el.wikipedia.org/wiki/MySQL>, χ.χ. s.l.: s.n.

<https://el.wikipedia.org/wiki/PHP>, χ.χ. s.l.: s.n.

https://github.com/checkly/puppeteer-examples/blob/master/1.%20basics/get_list_of_links.js, n.d. s.l.: s.n.

https://github.com/sunra/php-simple-html-dom-parser/blob/master/Src/Sunra/PhpSimple/simplehtmlDom_1_5/example/scraping/example_scraping_imdb.php, n.d. s.l.: s.n.

<https://help.apify.com/en/articles/1961361-several-tips-how-to-bypass-website-anti-scraping-protections>, χ.χ. s.l.: s.n.

<https://rapidapi.com/>, χ.χ. <https://rapidapi.com/>. [Ηλεκτρονικό]

Available at: <https://rapidapi.com/blog/how-to-use-an-api/>

<https://stigma.host/why-you-should-be-using-a-php-framework/>, χ.χ. s.l.: s.n.

Somerville, I., 2011. *Software Engineering 9th*. 9η Έκδοση ed. s.l.:Pearson.

Αθανασόπουλος, Π., 2013. <https://panagiotisathanasopoulos.gr>. [Ηλεκτρονικό]

Available at:

<https://panagiotisathanasopoulos.gr/%CE%BF%CE%B4%CE%B7%CE%B3%CE%AF%CE%B5%CF%82-%CF%83%CF%85%CE%B3%CE%B3%CF%81%CE%B1%CF%86%CE%AE%CF%82->

[%CE%B5%CF%80%CE%B9%CF%83%CF%84%CE%B7%CE%BC%CE%BF%CE%BD%CE%B9%CE%BA%CE%BF%CF%8D-%CE%AC%CF%81%CE%B8/](#)

Ανδρέας Φλούρης, Ρ., χ.χ. <http://www.eevfa.gr>. [Ηλεκτρονικό]
Available at: <http://www.eevfa.gr/site/documents/Frontistirio%203.pdf>

ΒΕΣΚΟΥΚΗΣ, Β., 2015. <https://repository.kallipos.gr/>. [Ηλεκτρονικό]
Available at:
https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/3162/1/02_chapter_02.pdf

ΒΕΣΚΟΥΚΗΣ, Β., 2015. <https://repository.kallipos.gr/>. [Ηλεκτρονικό]
Available at:
https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/3164/1/02_chapter_04.pdf

Διέτης, Ν., 2017. <https://www.ucy.ac.cy>. [Ηλεκτρονικό]
Available at: https://www.ucy.ac.cy/ctl/documents/KEDIMA/TaxyrrythmaFall2017-18/3_steps_to_Publication.pdf

Ιωάννης, Σ., χ.χ. <https://opencourses.auth.gr>. [Ηλεκτρονικό]
Available at:
https://opencourses.auth.gr/modules/document/file.php/OCRS221/%CE%A0%CE%B1%CF%81%CE%BF%CF%85%CF%83%CE%B9%CE%AC%CF%83%CE%B5%CE%B9%CF%82/%CE%95%CE%9D%CE%9F%CE%A4%CE%97%CE%A4%CE%91_11.pdf

Παναγιώτης, Σ., χ.χ. <https://aetos.it.teithe.gr/~sfetsos/>. [Ηλεκτρονικό]
Available at: <https://slideplayer.gr/slide/2633261/>

Παπαδόπουλος, Γ. Α., χ.χ. <https://www.cs.ucy.ac.cy>. [Ηλεκτρονικό]
Available at: <https://www.cs.ucy.ac.cy/courses/EPL362/Notes/Enothta-1.pdf>

Χρήστος, Σ., 2011. <http://mis.uom.gr/>. [Ηλεκτρονικό]
Available at: http://mis.uom.gr/wp-content/uploads/2013/06/05_06_2013_Somaras.pdf

