

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Development of Decision Support Web Application

Charalampos Avramidis

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΨΑΡΡΑΣ ΙΩΑΝΝΗΣ

Καθηγητής Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων
Αποφάσεων

October 2021

ΣΥΝΟΨΗ

Το αντικείμενο της προτεινόμενης διπλωματικής εργασίας είναι η υποστήριξη αποφάσεων εμπορίας ενός επιχειρηματία μέσω της ανάπτυξης διαδικτυακής εφαρμογής, η οποία θα έχει τη δυνατότητα να τον ενημερώνει για το πότε η κατώτατη προσφερόμενη από κάποιον ανταγωνιστή του τιμή ενός προϊόντος από αυτά που έχει αναρτημένα σε μία μηχανή αναζήτησης τιμών προϊόντων έχει μειωθεί περισσότερο από την δικιά του τιμή. Επιπλέον, θα δίνεται η δυνατότητα να αλλάξει την τιμή του εν λόγω προϊόντος κατα βούληση σε πραγματικό χρόνο πρώτα με ενημέρωση του διαδικτυακού μαγαζιού του (eShop) και στη συνέχεια της πλατφόρμας μέσω ενός «GET request» το οποίο θα είναι προγραμματισμένο να "χτυπάει" ανά 2 ώρες στο API του eShop. Ταυτόχρονα τα δεδομένα θα αποθηκεύονται σε μία βάση δεδομένων για να μπορέσουν να υποστηρίξουν τον χρήστη στην λήψη αποφάσεων με γράφους.

SUMMARY

The objective of the master's thesis is the support of marketing decision of an entrepreneur through the development of a web application, which will give the ability for real time information and update on when a competitor places a lowest price of a product, lower than the supporting entrepreneur, inside a price search engine platform. Furthermore the web application will provide the opportunity to the entrepreneur to change the price of the product firstly by changing the business eShop product price and then the search engine platform will update its prices when the scheduled "GET request" occurs after the 2 hour interval against the eShop's API. The data will simultaneously be saved inside the database in order to be able to support the decision maker alongside with the provision of graphs.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	5
INTRODUCTION.....	9
CHAPTER 1: Building the BackEnd of the application	
1.1 What is a BackEnd framework	10
1.2 Which are the most popular BackEnd frameworks around?.....	11
1.3 Why Django for building the API.....	12
1.4 Django vs Laravel.....	12
1.5 Django vs RoR (Ruby on Rails).....	13
1.6 Django vs Flask.....	18
CHAPTER 2: Database System	
2.1 Which database system to choose.....	20
2.2 PostgreSQL vs MySQL.....	20
CHAPTER 3: Web Scraper / Crawler	
3.1 Why is web crawling/scraping important for enterprises.....	22
3.2 How web scraping works.....	22
3.3 Choosing appropriate Programming Language for the scraper development.....	23
3.3.1 Node.js.....	23
3.3.2 Ruby.....	24
3.3.3 PHP.....	24
3.3.4 C & C++	25
3.3.5 Python	26
3.4 Which is the best Crawling toolkit for our project	27

Development of Decision Support Web Application

3.4.1 Requests.....	27
3.4.2 Selenium.....	28
3.4.3 BeautifulSoup	28
3.4.4 Scrapy.....	28
3.5 Scraping at a larger scale : Proxies.....	31

CHAPTER 4: Building the FrontEnd of the application

4.1 Choosing the right FrontEnd framework.....	34
4.1.1 React: The JS library backed by Facebook.....	35
4.1.2 Angular.JS: The framework created by Google.....	38
4.1.3 Vue.js: Maybe the easiest FrontEnd framework.....	40
4.2 Summary.....	42

CHAPTER 5: Presentation of the Application – Skroutz_Robot_App

5.1 The Login Page.....	44
5.2 The Welcome Page.....	45
5.3 Application’s Utilities.....	45
5.3.1 Adding a new product.....	46
5.3.2 Initializing the Robot / Scraper.....	48
5.3.3 Data Analytics Utilities.....	51
5.3.4 Adding Competitor Utility Manual.....	53
5.4 More potential utilities.....	54
6.Sources.....	55

Περίληψη

Το κύριο αντικείμενο της διπλωματικής εργασίας είναι η δημιουργία ενός δυναμικού web application για την χορήγηση υποστήριξης κατά την ανάληψη επιχειρηματικής απόφασης από κάποιον decision maker. Πρόκειται για ένα full stack application το οποίο ενσωματώνει στην κλασική αρχιτεκτονική ενός web application (DataBase - BackEnd - FrontEnd), μία σειρά από εργαλεία data engineering και data analytics, αφενός με το integration με ένα robot το οποίο θα κάνει web scraping, τις διευθύνσεις που θα του παράσχει ο χρήστης και στην συνέχεια, εφόσον ο χρήστης επιθυμεί θα μπορεί να αλλάζει τις τιμές όπως επιθυμεί είτε μέσω πλοήγησης στις λειτουργικότητες του application (επιλογή από την λίστα), είτε μέσω API call που θα πραγματοποιείται στο e-shop της επιχείρησης, αφετέρου θα μπορεί να ενημερώνεται για την πορεία των τιμών κάθε προϊόντος με διάφορους τρόπους (χρονοσειρά, line chart, competitor analysis με pie charts κτλ). Από πλευράς business intelligence, μπορούν να εφαρμοστούν περαιτέρω reports με γράφους, εφόσον κάτι τέτοιο κριθεί αναγκαίο, για παραδείγμα επεκτείνοντας τα πεδία δεδομένων που εξορύσσει ο scraper (για παράδειγμα με την εξόρυξη της διαθεσιμότητας των προϊόντων από τους ανταγωνιστές, η εξόρυξη πληροφορίας για το ποιοι ανταγωνιστές υποστηρίζουν την επιλογή της αντικαταβολής, ή ακόμα και άντληση δεδομένων για το κόστος μεταφορικών), αλλά και με ανάλυση χρονοσειρών, εφαρμογή Machine Learning, ή Deep Learning με χρήση νευρωνικών δικτύων, ή χρήση στατιστικών πακέτων με σκοπό την πρόβλεψη της πορείας των τιμών στο μέλλον κλπ. Η διαδικασία του scraper initiation είναι αρκετά "user friendly" ώστε να είναι απλή η διαδικασία για το μέσο χρήστη, αλλά συμπληρώνεται εδώ πως μπορεί να χρησιμοποιηθεί και scheduler ώστε να πραγματοποιείται η συλλογή ιστορικών δεδομένων χωρίς καμία ανάμειξη από πλευράς του χρήστη.

Έμφαση δόθηκε στην επιλογή των καλύτερων δυνατών εργαλείων από πλευράς της επιχείρησης, αλλά και από πλευράς δυναμικότητας των εργαλείων αυτών στην αγορά σήμερα. Σημαντικός είναι και ο συντελεστής βαρύτητας όσον αφορά την δημοφιλία των εκάστοτε εργαλείων από την data science κοινότητα.

Βάση Δεδομένων – DataBase system

Ως βάση του application θα επιλεγθεί σε πρώτο επίπεδο μία MySQL βάση η οποία θα λειτουργεί πάνω στον server που είναι εγκατεστημένο το eshop της επιχείρησης. Στην βάση θα φιλοξενούνται τα μοντέλα που έχουν δημιουργηθεί από το Django κατά το integration που θα πραγματοποιηθεί, περιλαμβάνοντας το μοντέλο των προϊόντων, των χρηστών και όλης της γενικότερης υποδομής του django, καθώς και δεδομένα που τόσο θα εισάγονται από τον χρήστη, όσο και δεδομένα που θα εισάγονται από το robot μέσω της διαδικασίας scraping. Επιπροσθέτως μπορούν να δημιουργηθούν stored procedures οι οποίες θα χρησιμοποιηθούν για την ανάκτηση δεδομένων από την βάση. Από πλευράς optimization η MySQL δεν είναι η καλύτερη επιλογή (η καλύτερη επιλογή εδώ θεωρείται η PostgreSQL η οποία θεωρητικά κάνει καλύτερο integration με το Django το οποίο θα επιλεγθεί ως BackEnd framework). Η PostgreSQL έχει σημαντικά πλεονεκτήματα σε σχέση με την MySQL για τους παρακάτω λόγους:

1. Τα περισσότερα data types που υποστηρίζονται από την PostgreSQL είναι primitives (integer, numeric, etc) κάτι που αποτελεί και κοινό σημείο της PostgreSQL με το Django.
2. Τα πιο "ταιριαστά" για λόγους optimization και ασφάλειας constraints που υποστηρίζονται στην PostgreSQL (UNIQUE, NOT NULL, Primary Keys, Exclusions etc

-) προτιμώνται από τους Django developers σαν γενικά εφαρμόσιμη "βέλτιστη πρακτική".
3. Μία σειρά από features όπως : υψηλό concurrency που μπορεί να υποστηριχθεί οδηγεί σε καλύτερο performance, και μία σειρά από τις πιο εξελιγμένες indexing μεθόδους (B-Tree, Multicolumn, Expressions etc). Επιπροσθέτως υποστηρίζει multi-version concurrency control και παραλληλισμο κατά την πραγματοποίηση των queries.
 4. Διαθέτει WAL (Write-Ahead Logging) το οποίο παρέχει υψηλή ασφάλεια δεδομένων. Μπορεί κάποιος να δημιουργήσει αντίγραφα των δεδομένων του χρησιμοποιώντας replication Master/Slave το οποίο δύναται να είναι τόσο Synchronous όσο και Asynchronous.

Με την MySQL ωστόσο όμως έχει καλύτερη εξοικείωση το προσωπικό της εταιρείας καθώς το eshop δουλεύει σε WordPress το οποίο είναι Integrated με MySQL και συνεπώς δεν θα χρειαστεί να χρησιμοποιηθούν εργατώρες για την εκμάθηση νέων συστημάτων στην περίπτωση που θα είναι αναγκαίο να πραγματοποιηθούν νέα queries για την εξαγωγή συμπερασμάτων από το business ή το marketing της επιχείρησης. Επίσης, η εγκατάσταση της βάσης του application θα είναι "διπλά" σε αυτήν που ήδη έχει συνηθίσει να δουλεύει ο Database Administrator της επιχείρησης. Τέλος πρέπει να αναφερθεί εδώ πως η MySQL επιλογή προσφέρει μία σειρά από πλεονεκτήματα που ενδεχομένως να ταιριάξουν στο project το οποίο είναι oriented στο Business Intelligence (OLAP - OnLine Analytical Processing & OLTP - OnLine Transactional Processing), εάν γίνει επέκταση της χρησιμότητάς του σαν πλατφόρμα αναζήτησης βέλτιστων τιμών, μέσα από μία σειρά από πλατφόρμες / eshop από τις οποίες θα γίνεται το scraping ώστε ο χρήστης καταναλωτής να ενημερώνεται τακτικά για τα προϊόντα που ενδιαφέρεται να αγοράσει στο προσεχές μέλλον.

BackEnd - Middleware

Το BackEnd αποτελεί το ενδιάμεσο layer μεταξύ χρήστη και βάσης. Οι λειτουργίες που επιτελεί είναι πολυάριθμες, μεταξύ των οποίων η εξασφάλιση της ασφαλούς λειτουργίας του application και της προστασίας του από εξωτερικές επιθέσεις με token σύστημα για το authentication του χρήστη, και "φίλτραρισμα" των εισόδων που πραγματοποιεί ο χρήστης, αλλά και η εύρυθμη και αποτελεσματική λειτουργία middleware καθώς θα δημιουργηθεί API για την ανταλλαγή των δεδομένων με το FrontEnd πάνω στο οποίο δύναται να αναπτυχθούν data science queries από τον BackEnd developer, δημιουργία νέων μοντέλων data, καθώς το initiation του scraper θα πραγματοποιείται από API call το οποίο θα διαχειρίζεται το BackEnd. Για την επιλογή BackEnd framework εξετάστηκαν τα πιο δημοφιλή frameworks της αγοράς τα οποία είναι :

1. Django (Python)
2. Ruby on Rails (Ruby)
3. Laravel (PHP)
4. Flask (Python)

Επικρατέστερη των ανωτέρω επιλογών κρίθηκε το framework της Python Django για τους παρακάτω λόγους:

1. Υψηλή ταχύτητα.
2. Διαθέτει ομαλή μαθησιακή καμπύλη.

3. Διαθέτει το πιο ευρύ community το οποίο υποστηρίζει την περαιτέρω ανάπτυξη του framework και την επίλυση προβλημάτων που μπορεί να αντιμετωπίσει ο προγραμματιστής κατά την ανάπτυξη του application.
4. Υψηλά επίπεδα ασφάλειας προστατεύοντας από μία σειρά προβλημάτων (π.χ. SQL injection attacks κλπ).
5. Παρέχει δυνατότητες scalability για το μέλλον σε περίπτωση που αποφασισθεί να γίνει deploy το application για ευρύτερο κοινό πελατών.
6. Είναι open source.
7. Είναι πλούσιο framework και βασίζεται στην πιο πλούσια (από πλευράς orientation στο data science) γλώσσα προγραμματισμού, την Python, πάνω στην οποία έχουν αναπτυχθεί οι πιο δημοφιλείς βιβλιοθήκες για ML, DL, και DRL οι οποίες είναι open source επίσης.

Web Scraping – Web Crawling

Η Python αποτελεί και την πιο δημοφιλή επιλογή για την δημιουργία Web Scrapers/Crawler, τα λεγόμενα robots. Πράγματι η Python έχει την πιο εκτενή σειρά εργαλείων για web scraping (Scrapy, Selenium, BeautifulSoup κλπ) τα οποία είναι και τα πιο αποτελεσματικά σε σχέση με τα υπόλοιπα (άλλων γλωσσών προγραμματισμού όπως C++ κλπ) αλλά και πιο γρήγορα. Από τα εργαλεία που είναι διαθέσιμα για την Python επιλέχθηκε το Scrapy για τους παρακάτω λόγους :

1. Ομαλό learning curve.
2. Πλούσιο framework, δίνει πολλές δυνατότητες και επιλογές στο χρήστη με την εγκατάσταση του framework out-of-the-box.
3. Δίνει επιλογές για integration με το Django framework για την συγχώνευση μοντέλων εάν κάτι τέτοιο κριθεί αναγκαίο στο μέλλον.
4. Δίνει επιλογή για το initiation του robot από API call μέσα από το Django με την χρήση του libart Scrapyd.
5. Υποστηρίζει την επιλογή τόσο Xpath, όσο και CSS selectors.
6. Είναι ίσως το πιο αποτελεσματικό εργαλείο μεταξύ των υπολοίπων επιλογών που εξετάστηκαν (Selenium με BeautifulSoup και Requests)
7. Διαθετεί άριστο documentation καθώς και ευρύ community που το υποστηρίζει και αναπτύσει νέα plugins (π.χ. Splash) , άρθρα διαμορφωμένα tutorials, καθώς και μία πληθυσμική ερωταπαντήσεων που δίνουν λύσεις στο StackOverflow.

Το Scrapy τέλος είναι ξεκάθαρη επιλογή από πλευράς integration των robotics με proxy servers, στην περίπτωση που κάτι τέτοιο θα υπαγορευθεί ως αναγκαίο εάν προκύψουν scalability issues, καθώς υπάρχουν εκτενή και ευανάγνωστα tutorials και υπηρεσίες που υποστηρίζουν την εν λόγω ανάγκη.

FrontEnd - UI

Τέλος η εμπειρία του χρήστη αποτυπώνεται στο FrontEnd του application. Ο χρήστης στην αρχική σελίδα πρέπει να εισάγει τα credentials του ώστε να εισέλθει στο application. Οι λειτουργικότητες που παρέχονται είναι οι παρακάτω:

1. Button για το initiation του web scraper.
2. Δυνατότητα για εισαγωγής νέου προϊόντος από τον χρήστη.
3. Επισκόπηση των τιμών για όλα τα προϊόντα που έχει εισάγει ο χρήστης από την τελευταία scraping διαδικασία.
4. Επισκόπηση της χρονοσειράς των τιμών για συγκεκριμένο προϊόν.
5. Επισκόπηση της πορείας των τιμών για συγκεκριμένο προϊόν με την βοήθεια γράφου (Line Chart).
6. Competitor Analysis για συγκεκριμένο προϊόν, δηλαδή ποιοι ανταγωνιστές έχουν συχνότερα την πιο ανταγωνιστική τιμή στην πλατφόρμα του Skrutz, για κάποιο συγκεκριμένο προϊόν, με την βοήθεια γράφων (Bar Chart, Pie Chart).
7. Εισαγωγή των ανταγωνιστών με το id τους στην βάση.
8. Δημιουργία νέου χρήστη.
9. Αποσύνδεση χρήστη με ταυτόχρονη διαγραφή των Cookies (token) από τον browser, για περισσότερη ασφάλεια.

Για την επιλογή του FrontEnd framework εξετάστηκαν οι παρακάτω επιλογές:

1. React (JavaScript)
2. Vue (JavaScript)
3. Angular (JavaScript)

Ως επικρατέστερη επιλογή κρίθηκε η REACT.js. Το application θα έχει σχετικά απλό DOM (Single page application), και επίσης η state of the art performance θα είναι on point. Η React αποτελεί μία επιλογή που μας φέρνει το facebook που σημαίνει πως θα υποστηρίζεται για πολλά χρόνια ακόμα, σε ανταγωνιστικό επίπεδο. Επιπροσθέτως, η React είναι η καταλληλότερη επιλογή για γρήγορο development καθώς δίνει την δυνατότητα επαναχρησιμοποίησης των components. Επίσης δίνει στον προγραμματιστή μία πληθώρα επιλογών που δεν περιορίζεται λόγω του framework. Βιώσιμες θα ήταν και οι υπόλοιπες επιλογές, όμως δεν θα ήταν οι βέλτιστες καθώς από την μία η Angular θεωρείται overkill για μικρά application (και η REACT μας δίνει το good enough scalability) και από την άλλη μπορεί η VUE να θεωρείται το πιο εύκολο framework, το οποίο όμως το μεγαλύτερο ποσοστό του community/documentation που έχει είναι κινέζικο.

Introduction

The objective of the master's thesis is the support of marketing decision of an entrepreneur through the development of a web application, which will give the ability for real time information and update on when a competitor places a lowest price of a product, lower than the supporting entrepreneur, inside a price search engine platform (Skroutz.gr). Furthermore the web application will provide the opportunity to the entrepreneur to change the price of the product firstly by changing the business eShop product price and then the search engine platform will update its prices when the scheduled "GET request" occurs after the 2 hour interval against the eShop's API. The data will simultaneously be saved inside the database in order to be able to support the decision maker alongside with the provision of graphs.

The technologies that have been used are the following :

1. MySQL for the Data Base implementation
2. Python / Django for the BackEnd Implementation
3. Python / Scrapy for the robot implementation
4. React / Recharts for the FrontEnd implementation

Main goal of this project is to combine the powers of Information Technology (such as Web Development, Data scraping, Data engineering, Data Science) along with the Business Intelligence (Data Analytics, Economic Science) department of one company, in order to provide an entry level application (with the right parameterization could be enriched) that can substantially help data analysts to extract information from the updated data in order to provide the best assistance for the decision makers with graphs.

Chapter 1 : Building the BackEnd of the application

A software framework is a foundation where the developers can make applications in a faster and standardized way. It can accommodate all those facilities that are constantly being used in applications ,instead of performing the same task again and again, in one nice packet, hence providing the abstraction for your application and more importantly for many applications.The web application framework is a software tool that is intended to facilitate and support web application development which includes web resources, web services and web APIs.

1.1 What is a backend framework?

Backend frameworks are server-side frameworks that are used to create tasks hassle-free and convenient for developers. They are focused on scripting languages (Node.js, JavaScript, Ruby) or compiled languages (Java, C#). It also lays focus on business logic and security authorization and authentication.

According to Wikipedia a framework’s main objective is to automate the overhead correlated with software development activities. The core advantages of using a framework for development are :

1. Time-Saving
2. Scalability
3. Robustness
4. Security
5. Integrations

Below, the table showcases the best backend frameworks for web development

Framework	Programming Language	Famous Use Cases
Django	Python	Instagram Pinterest Coursera
Laravel	PHP	Deltanet Travel Neighborhood Lender MyRank
Ruby on Rails	Ruby	ZendDesk Shopify GitHub
ExpressJS	NodeJS	MySpace GeekList Storify
CakePHP	PHP	Mapme Educationunlimited Followmy Tv
Flask	Python	Red Hat Rackspace Reddit
Asp .NET	C#	Microsoft Godaddy Ancestry
Spring Boot	Java	Trivago Via Varejo Intuit
Koa	NodeJS	-
Phoenix	Elixir	Financial Times Fox 10 ABC15

1.2 Which are the most popular Backend Frameworks around?

The most popular backend frameworks are the following:

1. Django (Python)
2. Laravel (PHP)
3. Ruby on Rails (Ruby)
4. SpringBoot (Java)
5. ASP.Net (C#)

The best REST API framework is Express.js, and Django because it incorporates the Django REST Framework (DRF) that simplifies development of REST APIs and lets you browse your API to see your program in action.

Django is considered in most cases the best Backend framework that someone could use nowadays. It is used by many popular applications such as Instagram, Google, Youtube, Pinterest, Coursera, Mozilla, Disqus, Pinterest, National Geographic, The Washington Times, to name a few.

Django is a high-level framework based in Python language and the Model-View design architecture: almost everything any developer would require is already included.

Hence, Django does not require the hassle of installing and maintaining third-party plug-ins, as everything in the framework functions together. It is ideal for the development of large-scale, database driven, interactive web applications.

However, if someone's scheduling for building something small, Django might not be the best choice for websites as it will make that small project bloated with needless characteristics.

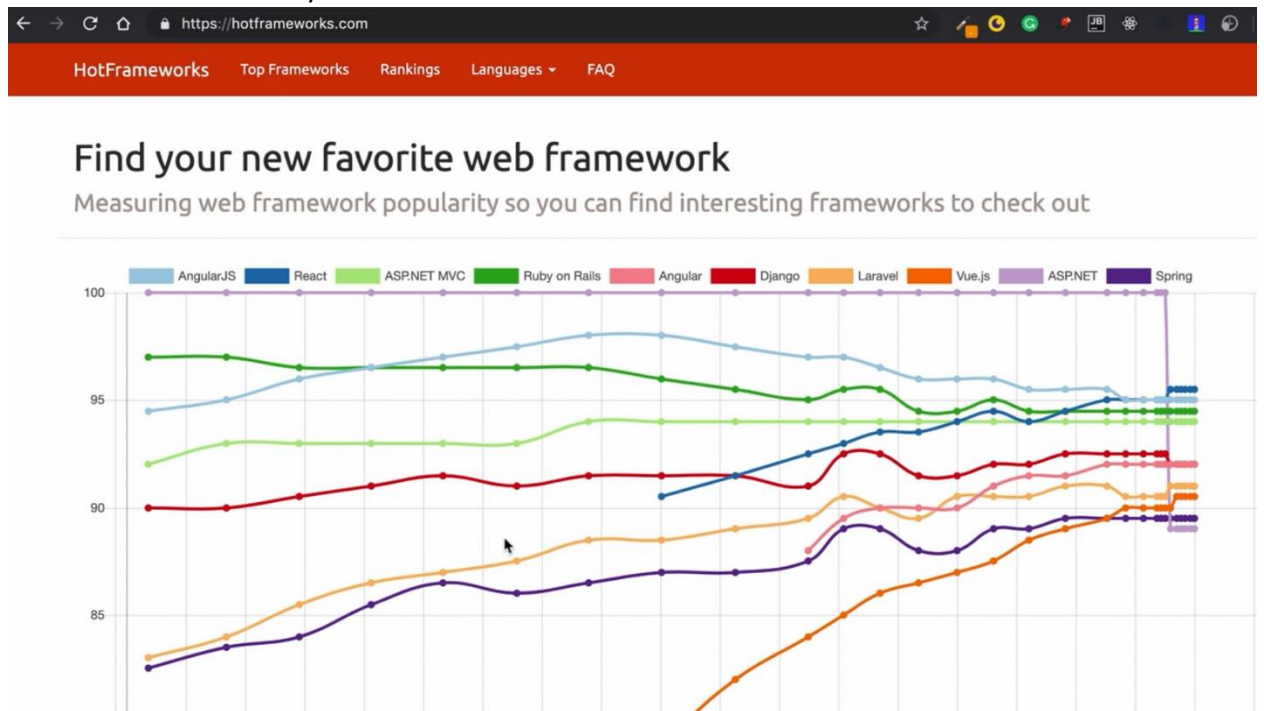
List of sites considering the Django Framework to be the best to build Backend API:

1. <https://www.monocubed.com/best-backend-frameworks/>
2. <https://blog.back4app.com/backend-frameworks/>

Backend frameworks are essential to application development for countless businesses across the world today. Finding the right backend framework can be quite crucial for developers to ensure optimal performance and scalability. With so many options available today, choosing the right one can be a hassle.

1.3 Why Django for building the API?

Django framework is a framework for “perfectionists with deadlines”. It is a very mature framework, having at least 10 years of development and it is not a framework that has just appeared and as well is not going to disappear any time soon Django is a full stack framework which means that can build both back end and front end. Django is very easy to use and it is really fast. Building an application will take at least half time as it would to build an application with any other framework. Django is packed with so many features that are ready to use in the moment when you install, that you don’t need to write your own code to cover it, but it doesn’t give you the freedom that other frameworks give you, and also it is considered to be a “heavy” framework.



1.4 Django vs Laravel

Django is an open-source framework for web development that is written in the Python programming language. The architecture pattern followed is Model View Template (MVT) in Django. The Django framework is used to develop complex web applications. Laravel is an open-source framework for web development that is written in the PHP programming language. The architecture pattern followed is the Model View Controller (MVC) in Laravel. Laravel provides a rich set of functionalities similar to that of Ruby on Rails. Laravel can be used to develop a Content Management System (CMS) applications.

Django is considered to be a little more faster, as it is built on Python programming language whereas Laravel uses PHP, which is a bit slower. Also Django is considered to have faster development and better/larger community users group. Using Laravel may face

compatibility issues since it's the only framework to be considered for PHP whereas there are several other frameworks for Python (e.g. flask etc).

The main differences between Django vs Laravel frameworks are that the language they support, and several different features and libraries exist to fulfill different requirements. There exist a lot of pros and cons as well as to understand the differences between Django vs Laravel performance. In terms of scalability, easier development, maintenance, and testing, Django is highly considered where Python also provides faster execution which further improves the speed of the application.

The choice, however, can be opted for based on the language and the features that fulfill the customer requirements. In terms of user community popularity, Django rates high, and Laravel comes second but close to Django, where it lacks some pros as compared to Django.

1.5 Django vs RoR (Ruby on Rails)

Written under MIT Licence, Ruby on Rails, is a server-side, open-source web app framework. Being a model-view-controller, Rails provides amazing default structures for a database, web services, and pages. It is considered as a time-saving method for developers to write code.

Ruby on Rails works on two principles mainly – Don't Repeat Yourself, and Conversion Over Configuration. DRY eliminates the need of doing the same task of coding again and again, where CoC means that the environment you work in, such as systems, libraries, languages and more, allows many logical situations by default.

Both languages provide clarity and readability of the code. In terms of speed and performance both Django and Rails are very competitive, as they leverage modern programming languages while they provide tools to optimize the code. Comparing RoR and Django on the basis of the installation process is not a hard nut to crack. Django's installation is very easy and it takes only few minutes to install it completely whereas RoR demand an understanding on what Bundle and Gems are, as they are needed to install the packages for RoR. In terms of security Python seems to be clearly the winner. NASA uses the Django framework python, which is a enough in itself advocating how secure it is. All in all though both Django and RoR are a reliable option and can be trusted for security. Regarding the scalability the two framework that can offer RoR is winner since it provides the freedom and flexibility of code. Both frameworks are really well documented so there is a tie in this basis. Depending on the type of usage, if someone is looking for a framework that helps in developing complex database-driven websites and web apps in less time, with efficiency in system administration, scientific programming, data analytics and data manipulation, then Django is the way to go. On the contrary RoR, which is flexible in nature, it is an ideal choice for meta programming and creating pleasing codes. In terms of the developer's learning curve Django with Python is the clear winner since it is really easy to learn among its contenders which also makes the learning curve of Django small. On the other hand RoR has a really steep learning curve due to independent concepts, that a developer needs to hone in order to become proficient.

Development of Decision Support Web Application

All in all, both Django and Ruby on Rails web frameworks are at the top of their category giving a hard competition to each other. Nevertheless, there are certain areas where one supersedes the other.

For example, if you want a highly detailed app loaded with remarkable features then you should go with Django. However, if you are thinking of a quick launch and then work on the details of the website or a web app then Ruby on Rails is your ideal bet. It is because it possesses shortcuts and automation features, which makes it easier to add complex features in the web applications.

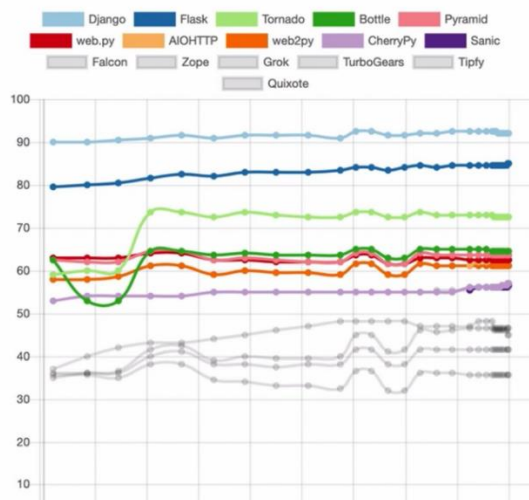
But the small learning curve, and the data driven usage that we are going to need for developing the current project, Django/Python seems to be the clear winner since the Web Crawlers will also be built on top of Python library Scrapy.

Rankings

Framework	Github Score	Stack Overflow Score	Overall Score
AngularJS	93	97	95
React	99	92	95
Ruby on Rails	90	99	94
ASP.NET MVC		94	94
Angular	91	93	92
Django	89	95	92
Laravel	91	91	91
Vue.js	100	81	90
ASP.NET	78	100	89
Spring	86	93	89
Express	90	85	87
Flask	90	80	85
Symfony	84	86	85
Meteor	89	80	84

Python

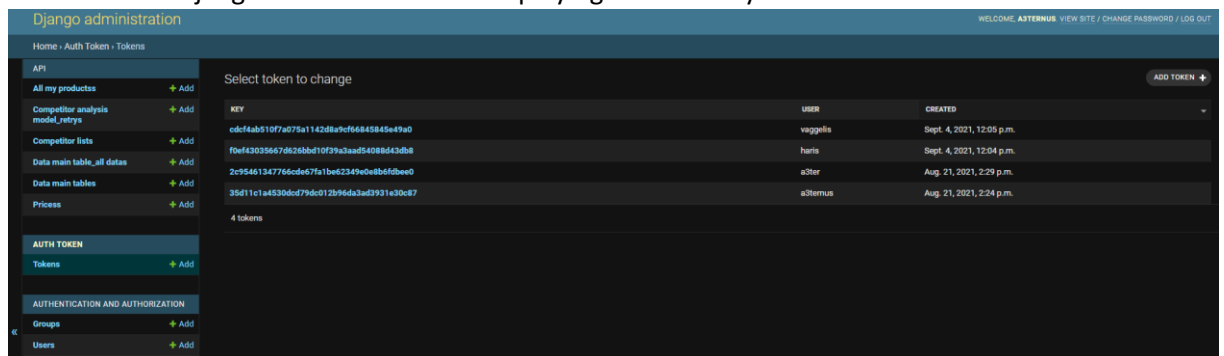
Framework	Score
Django	92
Flask	85
Tornado	72
Bottle	64
Pyramid	63
web.py	62
AIOHTTP	62
web2py	61
CherryPy	57
Sanic	56
Falcon	56
Zope	46
Grok	45



As you can see Django is one of the most popular choices amongst others and it is the most popular for python developers.

The most important advantages of Django are the following :

1. It is fast. Django is easy to use and a low learning curve framework.
2. Has many features, to help users take care of common requirements (user authentication, sitemaps, content administration and much more).
3. High security. It helps users prevent several security issues (cross-site scripting, clickjacking, SQL injection, request forgery to name a few). It provides a system for user authentication to enable users to store and manage passwords securely. Below screenshot of Django Administration UI displaying the token system.



4. Can provide high level of scalability to its users, crucial for many websites that rely the need to meet their high operational demands easily.
5. It is versatile as it can be used for the developing a wide array of application types such as networking applications, content management systems, and computing platforms.
6. It is Open-Source.
7. Has an easy to use administration panel in comparison to Yii and Laravel.
8. Has simple syntax.
9. Provides MVC (Model-View-Controller) core architecture. The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects. (Wikipedia)
10. Object Relational Mapper. Object-relational mapping (ORM, O/RM, and O/R mapping tool) in computer science is a programming technique for converting data between incompatible type systems using object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.
11. Middleware support.
12. Encourages clean, pragmatic design.
13. Provides stability over the years, with no technical dept. Applications written a few years ago, still make sense and can be easily upgraded.
14. Python is very easy to learn and a very readable language, making the Django one of the best choice for beginners.

On the other hand, Django have some cons also:

1. Django is synchronous, making reactive stuff and long living responses can be tricky to handle. (This has been fixed with Django 3.0 release).
2. Is very restricted and often doesn't allow the developer to write vanilla python powered by the framework.
3. The documentation sometimes is felt that does not cover real-world scenarios. It is a wide documentation indeed, but is not deep and does not cover real problems or does not show any examples. Most of the times the problems will be solved by researching Google or StackOverflow.
4. Server performance is often felt to be inadequate. Python can not serve as many requests on the same hardware as a compiled language such as Java, C, C++ Erlang or Golang.
5. Many think that Django can feel bloated for small projects. While on the bright side Django provide sheer scale and functionality, it has too many "bells and whistles" which can get in the way when developing a small scale application.
6. Routing requires some knowledge of regular expressions.
7. One of the biggest complaints amongst it's users is it's speed. Python is relatively slow for computation in comparison to other languages. This is making Django slightly slower from some competitors like RoR (about 0.7 percent) but slightly faster than other competitors such as Laravel (about 0.7 percent).

Last but not least, here should be mentioned that Python is of the most in demand programming language of the past decade. Python is a general purpose programming language that empowers developers to use several different programming styles (e.g. functional, Object-oriented, reflective, etc) when creating programs. The language comes with an extensive library that supports common commands and tasks. Its interactive qualities allow programmers to test code on the go, reducing the amount of time wasted on creating and testing long sections of code. Python also has the highest popularity among data scientist, due to its wide variety of uses. It is often the go-to choice for a range of tasks for domains, such as, machine learning, deep learning, artificial intelligence, and other popular forms of technology. These tasks are made easier due to Python's powerful data science libraries, which include Keras, Scikit-Learn, matplotlib, TensorFlow, Pandas etc. Python can also support very important tasks, including data collection, data manipulation, data analysis, data modeling, data visualization which are all key factors to work with in big data, providing the support for one of the most popular ETL big data toolkit which is Data Bricks in Cloud Services (Azure, AWS). The language has also a large community for support which is another reason it holds a vital place among the top tools for data science.

Pros of Python

- General purpose language**
Python is a broadly useful language that is simple and natural
- Library**
Beautiful soup is a python library that's designed for fast and highly efficient web scraping
- Framework**
 - Scrapy is the widely used frameworks that makes scraping using this language such as an easy to take.
 - Scrapy framework support Xpath, good performance based on twisted has debugging tools.
- Notable features**
Some notable features are pythonic idioms for navigation, searching and modifying a parse tree.
- Reusability**
Reusability through carefully executing bundles and modules.

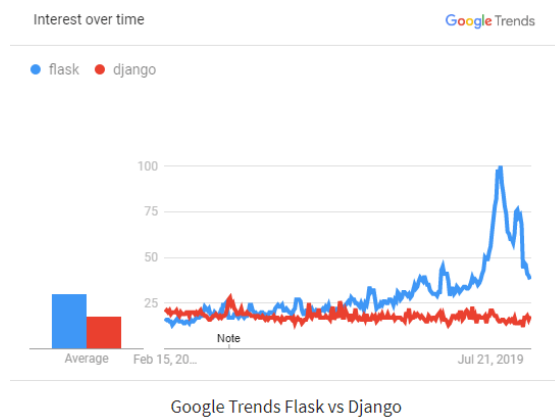
Cons of Python

- Speed**
Python is an interpreted language and it's slower than compiled languages but it's faster than other interpreted language.
- MultiCore/Multi-processor**
Python doesn't work good with multicore and multiprocessor
- Database**
Database access layer restrictions

One of the best use cases that python can be used is for automation. Automation tasks is extremely valuable in data science and will ultimately save a lot of time and provide valuable data. The biggest advantage of Python is its popularity among data scientists. This wide popularity means that there is endless support and a lot of resources available to continue the developer's education. Another thing to take into consideration when choosing a programming language to implement the project is the language that the web crawlers are going to be designed with. Python is known as the best web scraper language. It's more like an all-rounder and can handle most of the web crawling related processes smoothly. BeautifulSoup is one of the most widely used frameworks based on Python that makes scraping using this language such an easy task to take. Also scrapy, that is going to be used to implement this project's crawler, is another popular option, due to its performance, using a Twisted Library, supporting Xpath which is good for performance and carrying a set of amazing debugging tools. Last but not least Selenium is a great tool when it comes to automating boring stuff, and can be programmed to behave as a user and be combined with Scrapy or BeautifulSoup creating a great toolkit for Data Engineers. To sum up in the following table we can easily see the pros and cons of python programming language in a nutshell.

1.6 Django vs Flask

Another popular choice, given the Python programming language choice, would be flask framework, that it is a minimal framework which means that you must write all the code by yourself and add things manually. Flask is a micro framework which offers basic features of web app. Flask has no dependencies on external libraries. Some extensions are offered for form validation, Object-Relational-Mappers, open authentication system, uploading mechanism. Django on the other hand Django is a web development framework which offers efficient and fast method for website development. Provides assistance in building and maintaining quality web applications. I can make the development process smooth and time saving. The main goal of Django framework is to create complex database driven websites.

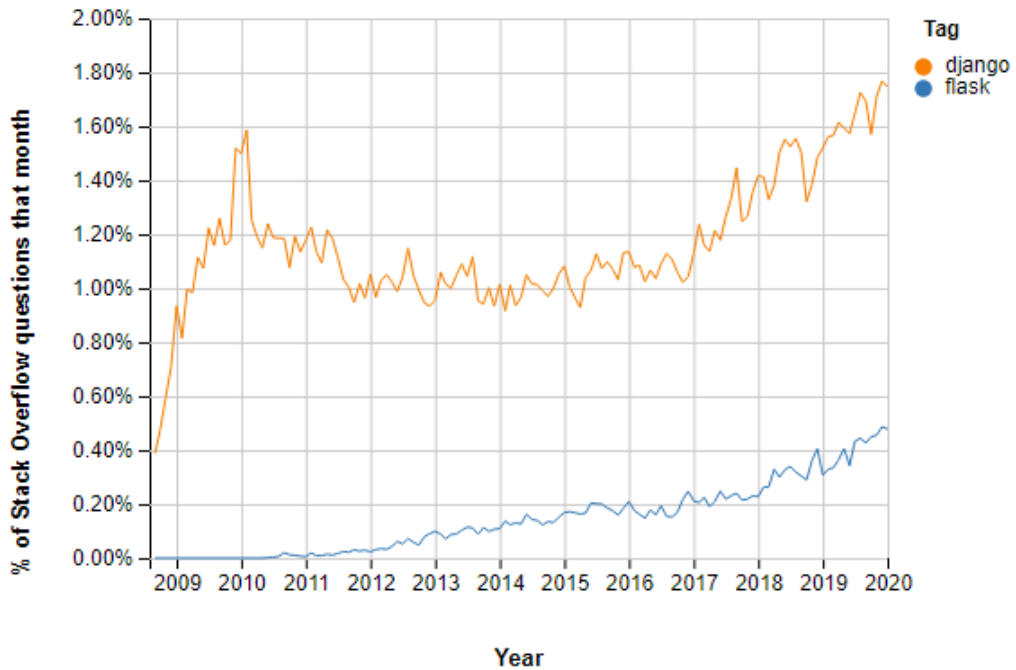


Flask and Django are most certainly the two most popular Python-frameworks.

The flask, in one hand, has many advantages. First of all provides high compatibility with state of the art technologies, and gives room to the developer for technical experimentation. For small size projects and for simple cases maybe better to use since the codebase size is relatively smaller compared to Django. Like Django it provides high scalability. Routing URL is easier compared to Django, which demands an understanding of Regular expressions. Moreover it is considered to be easier framework for database integration, developing and maintaining apps and got loads of resources available on GitHub.

On the other hand, Django is really easy to set up and run. It provides a Graphical User Interface for various administrative activities. It is a really mature framework, since it is 5 years older than flask, and offer end to end application testing. REST framework, which is provided, has rich support for several authentication protocols. It maybe more difficult in URL routing than Flask, but it helps in defining patterns for URLs and thus, can be a more appropriate choice for more complex applications. Offering a built-in authentication system can really provide a secure choice, plus can relieve some weight from developer's shoulders regarding security matters. Last but not least Django comes with a complete stack of tools, and a really rich community support.

Development of Decision Support Web Application



Stack Overflow Questions Flask vs. Django

On the downside, Flask offers limited support and smaller community compared to Django. It is considered slower regarding MVP development in most cases. For more complex systems can be really demanding in maintenance costs and setting up a large project requires previous knowledge of the framework. Django as well can feel really bloated for small size applications, and can have limited compatibility with latest technologies.

To sum up, Flask is more appropriate when a more granular level of control is needed, while Django on the other hand has an extensive community which can be really helpful when developing a unique website. Django combined with the REST framework, which will be used to implement this project's API, can really help creating powerful APIs, whereas Flask requires more work, so there is a high possibility to make mistake.

Django framework is in fact a full stack framework that could possibly build everything. But in this project it will only be used to create an API to serve the data from the database to front end. For that purpose the library that will be used is "Django Rest Framework", that is another library on top of Django and that's going to be used to build our API. For building the back-end Python programming language will be used.

Chapter 2 : Database System

2.1 Which database system to choose?

Having chosen our Back-End framework is a prerequisite in order to choose the database system which will support the creation of this project. In Django documentation the following database systems are supported by Django : PostgreSQL, MariaDB, MySQL, Oracle, SQLite. Also there are the following database backends provided by third parties : CockroachDB, FireBird, Microsoft SQL Server.

To begin with as Django is designed in Python, it works best with a RDBMS (Relational Database Management Systems) , than a DBMS, making the best choices the following databases : SQLite, MySQL, Oracle, and PostgreSQL.

Oracle is going to cost extra funds so will be omitted for the moment. That leaves PostgreSQL, SQLite, and MySQL as the remaining choices. If it is a simple project then most developers consider that SQLite is good enough, as it doesn't require a huge server running, and all data is stored in a file which is easy to transfer from system to system. MySQL and PostgreSQL are designed to support high load systems. Since the selection of Django was decided, among other reasons, upon its capability of upscaling the project, PostgreSQL and MySQL are both viable choices for the project.

2.1 PostgreSQL vs MySQL

MySQL is the world's most popular DBMS as 39% of developers using it in 2019, and it is quite fast, reliable, general purpose relational database management system. MySQL is considered the go-to-choice for scalable web applications due to the fact that it comes standard in the LAMP stack (an open source suite of web applications that consists of Linux, Apache, HTTP Server, MySQL and PHP). In addition, popular content management systems like WordPress rely on MySQL. MySQL is open-source, and it first became available in 1995. It has an extensive community of volunteers to help when troubleshooting is needed. It is considered a very stable RDBMS as long as you keep your databases "tidy" and perform regular maintenance.

On the other hand PostgreSQL is the go-to solution for performing complicated, high-volume data operations. PostgreSQL is better at handling extraordinary database situations, and it has a lot more features compared to MySQL. It is catalogue driven, which means that it doesn't just store information about tables and columns, but it lets you define data types, index types, and functional languages. Additional advocates in favor of PostgreSQL are the fact it is object-relational, ACID-compliant, highly concurrent. Both MySQL and PostgreSQL offer NoSQL support.

The Django community and official Django documentation state PostgreSQL as the preferred database for Django Web Apps. This is because Django provides support for a number of data types which will only work with PostgreSQL, which offers the richest feature set of the supported databases so its users have the most to gain.

On the other hand MySQL has a speed advantage in OLAP (Online Analytical Processing – which is going to be the case in this project) and OLTP (Online Transactional Processing),

which makes MySQL best suited for BI (Business Intelligence) applications. While PostgreSQL works well with BI applications, it's geared more towards Data Analysis and Data Warehousing applications.

The clear winner here between MySQL and PostgreSQL, regarding the compatibility with Django is PostgreSQL due to the following reasons:

1. All Django data types are supported in PostgreSQL as primitives (Integer, numeric etc).
2. The best needed constraints are supported in PostgreSQL like UNIQUE, NOT NULL, Primary Keys, Foreign Keys, and Exclusions.
3. Regarding the concurrency and performance, some of the most sophisticated of indexing methods (B-tree, Multicolumn, Expressions etc) are supported. Moreover, multi-version concurrency control, parallelization of reading queries and declarative table partitioning is available in PostgreSQL.
4. Having a Write-Ahead logging (WAL) to provide data assurance, Postgres never loses data. You can replicate your data using Master/Slave Native Replication, which can be Asynchronous, Synchronous and Logical.

In this project, regardless of the above, MySQL will be used to implement the project, since the collaborative company (Opto-Vision, Papaeythymioy) is using MySQL and the staff of the company is already familiar with the usage of MySQL since its WordPress Eshop is integrated with MySQL. This will encourage the possibility of the IT section of the company to experiment with the data, creating new queries, views or stored procedures, in order to empower furthermore the Data Science department of the company.

Chapter 3: Web Scraper / Crawler

3.1 Why is web crawling/scraping important for enterprises?

The explosion of the internet has been a boon for data science enthusiasts. The variety and the quantity of data that is available today through the internet is like a treasure trove of secrets and mysteries waiting to be solved. But there is no standard methodology of extracting the data which, in most cases, are unstructured and full of noise. Such conditions create the need of web scraping, making it a necessary technique for every data scientist's toolkit. Numerous enterprises rely on World Wide Web resources to extract data to create perfect data visualizations. But what web scraping really is?

Web scraping is data extraction from websites. Anything that someone could think of, can be extracted from the web, such as text, images, videos, e-mails, etc. Any content that can be viewed on a website can be scraped.

Web scraping, web harvesting, or web data extraction is a technique used for extracting data from websites. The web scraping software may directly access the world wide web using the Hypertext Transfer Protocol or a web browser. Web scraping can be performed manually by a software user, but the term typically refers to automated processes implemented using a bot or a web crawler. It is a technique of copying, in which specific data is gathered and copied from the web, typically into a database, spreadsheet for later retrieval or analysis.

3.2 How web scraping works

Web scraping works in two parts: a web crawler and a web scraper. The crawler leads the scraper, through the internet, where scraper extracts the data that are needed. The Crawler, or Spider, is an Artificial Intelligence that browses the World Wide Web to index and search for content by following links and exploring. In many cases the first thing to be done is to gather the URLs that need to be scraped and then pass them on to the scraper. In this project this part won't be automated, as the user will provide all the URLs using the application's interface, as will be projected in the application demonstration later on.

On the other hand, web scraper is a tool designed to extract data from a web page. Using the data locators, or selectors, the scraper will find the data that is needed to be extracted from the HTML file, which usually are XPath (that will be used for the needs of this project), CSS selectors, regex, or a combination of them if it is needed for more complex extractions. Web scrapers can vary widely in design and complexity depending on the project.

A typical web scraper would fetch a HTML page, downloading it, and then the extraction will take place. The content of a page may be parsed, searched, reformatted, and its data can be copied into a spreadsheet or loaded into a database and it mainly is designed in order to fulfill the requirements of the business intelligence of a company which then will retrieve the data to project them in a user friendly way for further analysis, providing support to the decision makers.

There are many ways that web scraping can provide assistance to the BI of a company. Price intelligence is one of the biggest use case for web scraping, and it will be the use case that we will focus on, in this project. Extracting pricing information from e-commerce websites,

like Skroutz or BestPrice, and then turning it into intelligence is an important part of modern e-commerce companies that want to make better pricing/marketing decisions based on the data. More specifically dynamic pricing, Revenue optimization, Competitor monitoring, Product trend monitoring, brand and MAP compliance are the most popular use cases of price intelligence. These are the use cases that most retail companies in Greece could focus and take the advantages of web scraping. More business sectors that are interested in the prospects of data extraction could include finance, real estate, social networking etc. This is due to the fact that data extraction can provide all the data that are needed as a fuel to power up various AI systems, even including algorithmic trading, sentiment analysis, news monitoring, brand monitoring, business automation, map monitoring etc.

But which is the most appropriate programming language for performing the above tasks these days?

3.3 Choosing appropriate Programming Language for the scraper development

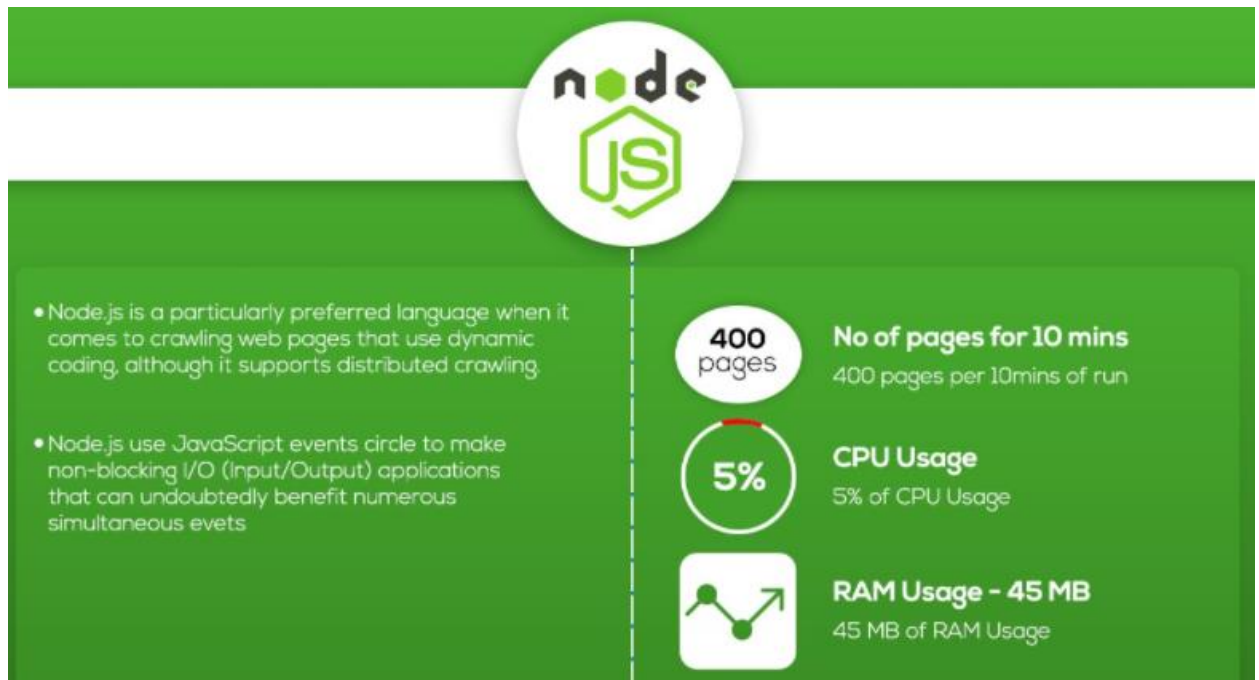
Firstly the type of data that are needed to be extracted from websites have to be determined. There are many popular choices that come at the top spot. No specific programming language can be claimed as the best option for a given project. However a wrong choice can lead to wasting time, money and energy on something which will not yield desired results. Therefore, it is important to carefully select a particular programming language for a specific project. Each language comes with features and limitations and the choice is strongly depended upon project type to put with a language to crawl data efficiently.

There are the following parameters that need to be precisely determined before selecting the appropriate language:

1. Flexibility
2. Operational ability to feed database
3. Crawling effectiveness
4. Ease of coding

3.3.1 Node.js

Node.js is a popular choice for web scraping, employing the use of dynamic coding practices. The framework supports distributed crawling, data extraction for larger-scale projects, and stable communication. Moreover it utilizes JavaScript events to counter non-blocking I/O applications which can benefit other data projects as well.



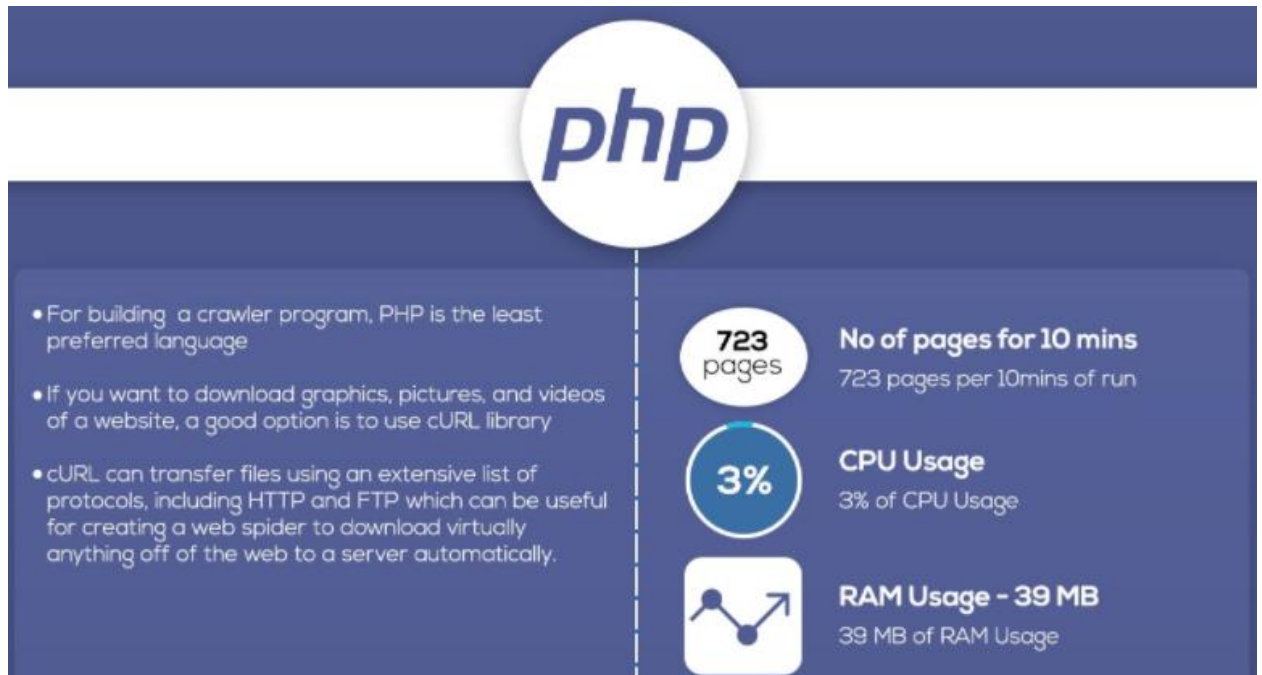
Node.js is recommended to be used for streaming, socket-based implementation and API. Many developers use multiple instances for the same scraping project as Node.JS takes only one core of the CPU. The framework, however, poses many limitations since it is not considered suitable for larger-scale data incentives, lacking maturity and stability for big data projects.

3.3.2 Ruby

Ruby is widely used due to its simplicity and productive nature as compared to other programming languages. Ruby keeps the functional balance of programming with aid of imperative programming. RoR helps to write less code and avoids any type of repetition and thus it is a choice that enables us to write a simple code. Ruby language is supported by a vast community of users in place of of any particular company. The language is a bit slower in comparison with the rest choice that we will take into consideration

3.3.3 PHP

Another option to build crawlers upon is PHP programming language. PHP is one of the least preferred languages to build web crawlers/scrapers. This is due to the weak support for async and multi-threading. The task scheduling and queueing issues can be associated with the usage of PHP language while web crawling for acquiring the desired data. One of its advantages though is that with the help of cURL, someone can extract videos, graphics, and photographs from websites efficiently with an extensive list of protocols involving FTP and HTTP.

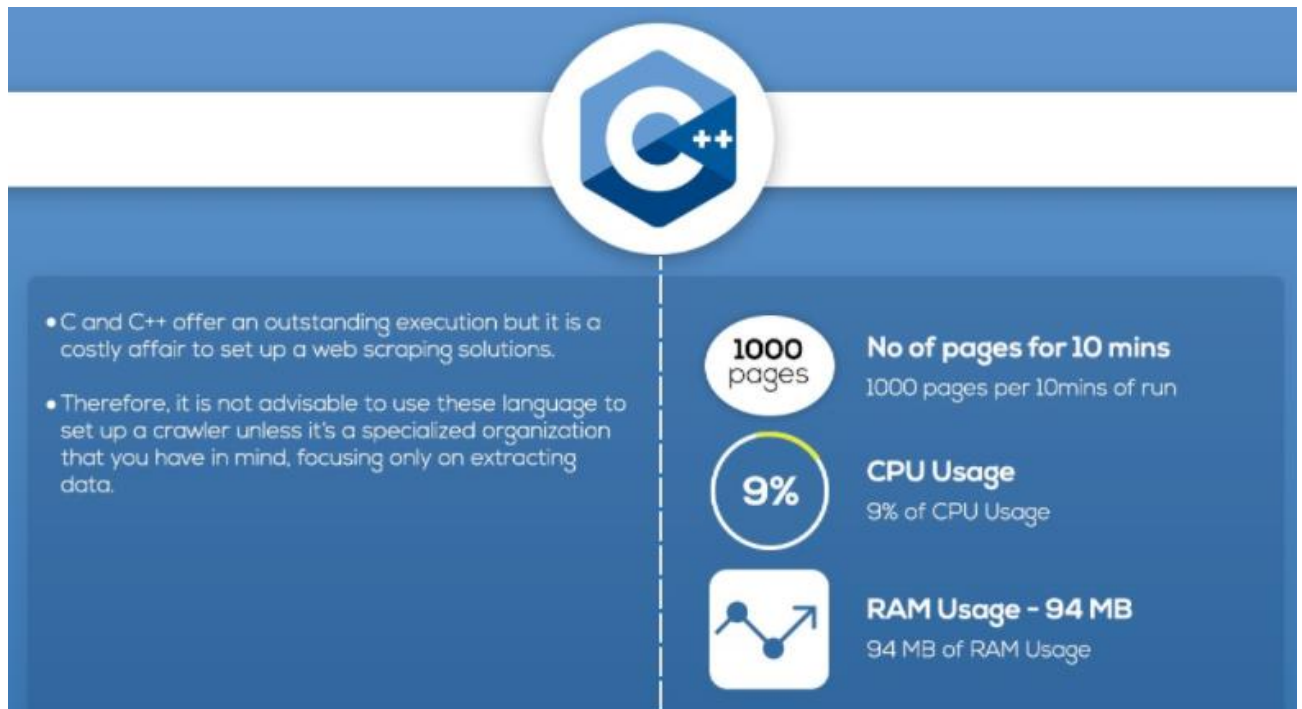


PHP is good enough for multimedia scraping, but due to its weakness in async and multi-threading makes it unsuitable for big data project, with lots of issues in queueing and task scheduling, making it less desired option for our project.

3.3.4 C & C++

C & C++ is considered to output the best results from constructing a unique web scraping set up. The limitations that these programming languages pose regarding the implementation of our web crawlers, is the cost that is a bit higher. Plus the languages are not perfect for creating web crawlers.

The high cost doesn't make this choice a good one when it comes to create smaller scale data projects. C++ would be the first choice of any professional web related data project since it is quite easy to understand as it supports a simple user interface and is efficient to parallelize the scraper via the C++ programming language.

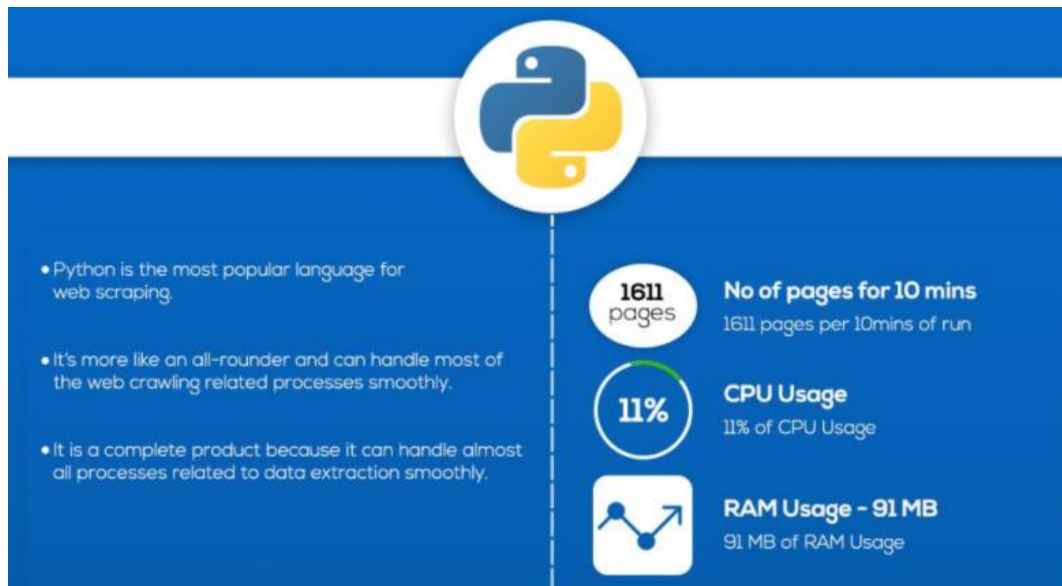


3.3.5 Python

Last but not least, here we should examine the Python programming language. Python is considered among the data science community as the number one web scraping programming language known today. It is a complete all-rounder platform where the developer can conduct data extraction smoothly.

It is maybe the most obvious choice when it comes to web scraping providing libraries/tools such as BeautifulSoup, Selenium, Scrapy which are supportive frameworks based on Python which further enables smoother web scraping than any other platform. Scrapy includes numerous features such as enhanced performance with a twisted library, support XPath, and a variety of debugging tools that makes the web scraping process feel easy. Other pros of python include the pythonic idioms that can aid in modifying, searching and navigating the results of a parse tree.

To sum up, the most efficient choice here is by far the Python programming language. Choosing Django as our BackEnd API framework, Python gets the edge over C++ (which is for more large projects) and Ruby (which poses a sharper learning curve and is not a data science programming language). Python comes also with the most popular web scraping tools which are supported from a vast community and many projects that are ready to use and to reverse engineer from github in assistance when it comes to design our Skroutz web crawler.



3.4 Which is the best Crawling toolkit for our project?

Python is our to-go-to programming language for developing our scraper. Having an extensive range of toolkits Python libraries for Machine-Learning, Deep-Learning, Data Science in general, and Web Development, Python continuously holds the trust of a lot of leading professionals in the fields of data extraction, web scraping and web mining and collection. Python also has also one of the most well documented and feature rich libraries as well as a robust support for Object Oriented Programming.

Making web scraping tools in Python using popular python libraries and packages like Selenium or BeautifulSoup is presently popular, due to the fact that it provides innovative functions, which are easy to use. Most of these libraries and functions are easy-to-learn as well as implement with the original applications; as these packages could be used later in the API formats to create custom-made web scrapers. Using these python libraries web scraping can be really easy in different fields including Twitter, Instagram, Amazon using other Python libraries and frameworks.

The following python libraries and functions which will be examined here are all open-source as well as they have rich documentation and a vast community support, which makes the usability and interfacing much easier.

3.4.1 Requests

Requests is a library designed to simplify the process of making HTTP requests. Requests comes handy in the first step of any web scraping workflow which is to send an HTTP request to the website's server to retrieve the data displayed on the target web page. Python comes out of the box with two built in modules (urllib and urllib2) which are designed to handle HTTP requests. Most developers, however, often prefer to use the Requests library due to the fact that using urllib and urllib2 need to be used together and the documentation can be overwhelming, often requiring a lot of code to be written in order to make a simple HTTP request. In order to build a complete functioning web scraping spider, a developer will need to write his own scheduling and parallelization logic, and use

other Python libraries such as BeautifulSoup to fulfill the other utilities of web scraping process.

Requests library is considered an important asset to be added in your data science toolkit. It is a very simple yet powerful HTTP library. Its easiness is probably its biggest advantage. Moreover, requests can be used to utilize API's, post, forms, and many other things.

3.4.2 Selenium

Another library that can be useful in web scraping is Selenium. The difference between selenium and the other libraries is that it wasn't developed for web scraping in the first place. Selenium is a web driver designed to render web pages as your web browser would for the purpose of automated testing of web applications. The fact that most of today's web pages make use of JavaScript to dynamically populate the page, make selenium a necessary asset to a data scientist toolkit. That's because most normal web scraping spiders have problem to execute this JavaScript code, and thus they are prevented from accessing all the available data, limiting their ability to extract all the available data. When a spider build in Selenium visits a web page, it will execute all the JavaScript available on the page before making it available for the parser to parse the data. The advantage of this approach is that it enables the spider to scrape data not available without JS or a full browser. However the web scraping process is a lot slower compared to making simple HTTP requests to the web browser, due to the fact that spider will execute all the scripts present on the web page. Selenium is a great choice if speed isn't a big concern, or the scale of the scraping isn't huge, but it is not the ideal choice. If speed is a big concern or you plan to scrape web in a large scale, then executing JS on every web page the scraper visits can be proven really impractical.

3.4.3 BeautifulSoup

BeautifulSoup is a python library designed to parse data from HTML or XML documents. Because BeautifulSoup can only parse the data and can't retrieve the web pages themselves. It is often combined with Requests library. One benefit of using BS is its capability to automatically identify encodings. It allows to elegantly deal HTML documents using special characters. Also BS can assist in navigating parsed documents and discover the data that may be useful. It is really easy to use and utilizing BS can make the process of creating general applications really fast.

3.4.4 Scrapy

Above stand the main python libraries used for scraping the web. Each of them is designed to accomplish one aspect of the web scraping process, resulting in having to combine multiple libraries in order to build a fully functioning spider. An easier approach would suggest a spider using a web scraping framework such as Scrapy, that includes all the core components to build a web scraper out of the box and has a huge range of plugins designed to deal with edge cases.

Scrapy is an open source and collaborative framework for extracting the data from websites. Scrapy is a fast high level web crawling and scraping framework for Python. It is useful for a wide variety of purposes, from data mining to monitoring and automate testing. It is an

application framework for writing web spiders that crawl web sites and extract data from them.

Scrapy was built by ScrapingHub co-founders P. Hoffman and S.Evans. Scrapy is a fully-fledged web scraping solution that takes a lot of the work off the data engineer shoulders when building and configuring the spiders, and best of all, it smoothly deals with the edge cases that he even haven't thought of yet. The installation process takes only few minutes, giving a fully functioning spider scraping the web framework. Out of the box, Scrapy spiders are designed to download HTML, parse and process the data and save it in either CSV, JSON, XML file formats. It supports multiple database connections, including MySQL, which is our choice in order to implement this project. It even can be integrated with Django seamlessly, and provide the functionality of scrapyd, which is a library that can initiate the Scrapy Spider from a url call from Django. There is also a wide range of built-in extensions and middlewares developed to handle cookies and sessions as well as HTTP features like compression, authentication, caching, user-agents, robots.txt and crawl depth restriction. Scrapy also provides the ability to extend through the development of custom middlewares or pipelines to your web scraping projects which can give you the specific functionality you may require.

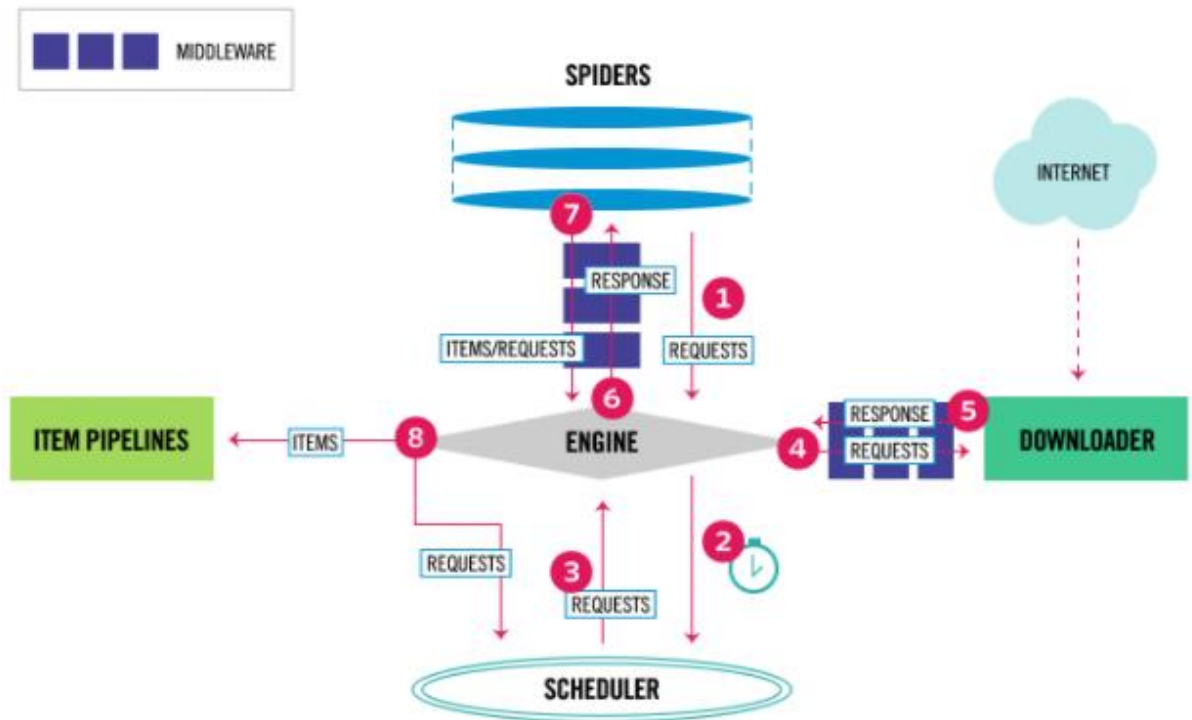
Built on Twisted, and asynchronous networking library, Scrapy framework provide the ability of making multiple HTTP requests in parallel and parse the data as it is being returned by the server. This significantly increases the speed and efficiency of a web scraping spider. Splash has also been created from ScrapingHub team, making Scrapy to be able to handle JavaScript, which is a n easy-to-integrate, lightweight, scriptable headless browser specifically designed for web scraping.

One drawback is the learning curve of Scrapy, which is steeper than, for example, that of BeautifulSoup's. On the other hand, however, Scrapy project has excellent documentation, and an extremely active community on GitHub and StackOverflow which is always releasing new plugins and help troubleshooting.

Development of Decision Support Web Application

	Scrapy	Requests	Beautiful Soup	Selenium
What is it?	Web scraping framework	Library	Library	Library
Purpose	Complete web scraping solution	Simplifies making HTTP requests	Data parser	Scriptable web browser to render javascript
Ideal use case	Development of recurring or large scale web scraping projects	Simple non-recurring web scraping tasks	Simple non-recurring web scraping tasks	Small-scale web scraping of javascript heavy websites
Built-in Data Storage Supports	JSON, JSON lines, XML, CSV	Need to develop your own	Need to develop your own	Customizable
Available selectors	JCSS & Xpath	N/A	CSS	CSS & Xpath
Asynchronous	Yes	No	No	No
Javascript support	Yes, via Splash library	N/A	No	Yes
Documentation	Excellent	Excellent	Excellent	Good
Learning curve	Easy	Very easy	Very easy	Easy
Ecosystem	Large ecosystem of developers contributing projects and support on Github and StackOverflow	Few related projects or plugins	Few related projects or plugins	Few related projects or plugins
Github stars	32,690	34,727	-	14,262

From the 4 python toolkits that have been presented Scrapy is considered the best choice. The learning curve maybe steeper than the rest, but the possibilities that provide out of the box, without the need of combining it with other libraries is one of the strongest advocates when it comes in making the choice. Also the community and the documentation that is backing the Scrapy framework is really great, and the scalability that can provide, really makes it the obvious choice. Below you can see the way that Scrapy works with the assistance of a flow chart.



3.5 Scraping at a larger scale : Proxies

Last but not least, here should be mentioned some problems that may occur, when the Scraper/Crawler will be deployed at a larger scale, than that we will use in UAT level. For example in order to work around anti-bot solutions you may want to access the targeted web site (in our case skrouz) from various countries/regions. To overcome obstacles like these there has to be some kind of proxies usage or management. IP banning is one of the most frequent reasons that the data flow maybe interrupted in production, when the scraper is banned. There are a number of proxy services available on the web where someone can set up his scraper and avoid such problems when he wants to scrape at larger scales.

But how does a proxy service work and how can its usage can help eliminate the risks when scraping at a large scale? A proxy server is like a mask to wear when accessing the web. It is like a whole new level between the user and the web. When using a proxy server, it channels on flow of internet traffic and gets you to the URL requested. The response to this request is through the same tunnel and then the HTML, thus the data, will be brought to you. This results in hiding your machine's/scraper's IP address. In this way when the scraper

Development of Decision Support Web Application

sends the requests, the target site will see the requests coming from a proxy IP and not the

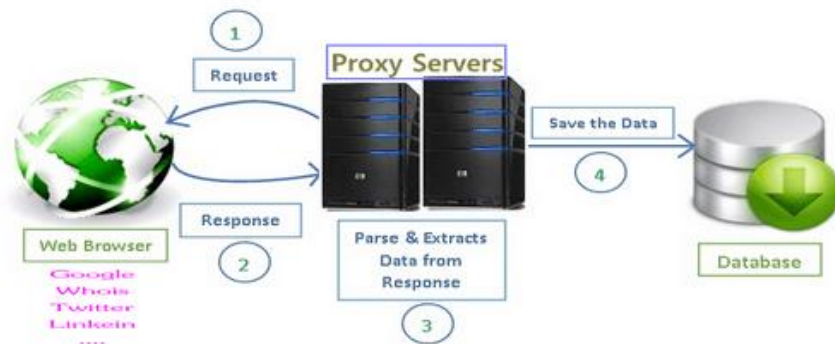


Source - Kimonolabs

original IP.

Web sites have different kind of mechanisms to detect the usage of a robot. Apps like honey pots identify and block scrapers. Most sites identify scraping when the following reasons occur:

1. Unusual traffic or download rate from a single address within a short time.
2. Humans do not perform, with the same pattern of behavior, the same tasks on and on, on a website. Any unhuman behavior can be easily identified (outlier detection, etc).
3. Honey pots are invisible fake links that are not visible for humans but only to a spider. When the spider crawls the links, it sets an alarm and alerts the site.



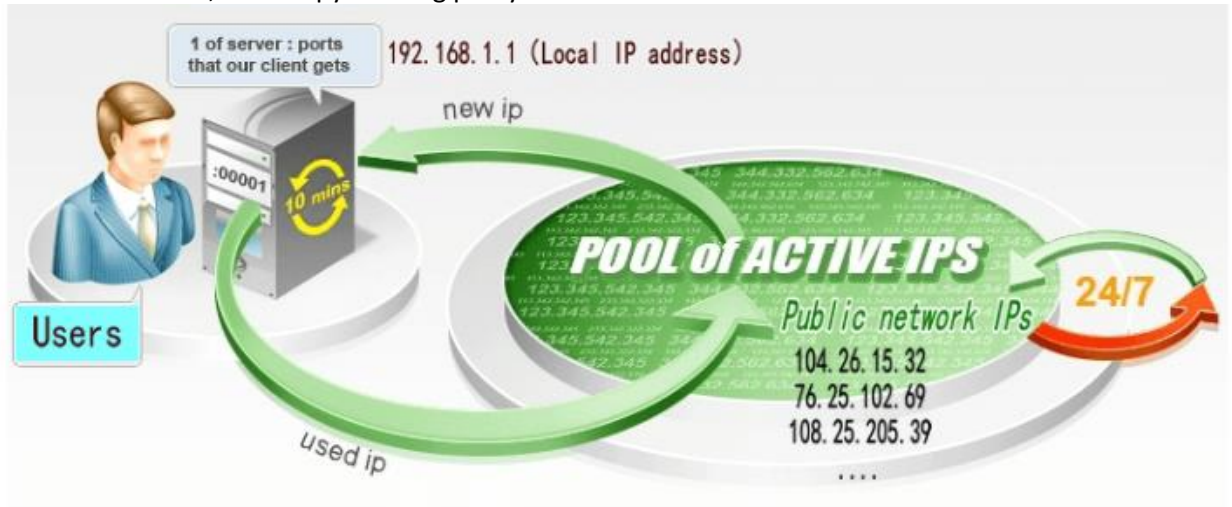
Source - [bestpaidproxies](#)

But of course it is a very possible to ban a proxy as well. In that case, the solution is to rotate IPs. When you rotate a bunch of IP address, they randomly pick an address and request for the web page.



Source: Smart Proxy

If it succeeds then the page is crawled and scraped. If not, then the IP is probably banned, and in that case another IP will be picked from the bundle. Managing this manually requires a lot of effort. But, if a Scrapy rotating proxy is installed then this effort can be automated.



Source: best paid proxies

Chapter 4: Building the FrontEnd of the application

Front-end is the part of the application with which the user interacts, in other words the UI, or app screen. When developing a web application, the client-side/front-end has great importance due to the fact that this is what the user mainly experiences. Yes, the back-end matters a lot, but the user can only witness what is happening on the front-end. Therefore, front-end is the key to improve the impression that it makes to a user, and hence the developer can not compromise on the quality of front end development. The front-end languages that are used to develop the front-end, play significant role to the overall client-side user experience.

The components that constitute a frontend framework are the following:

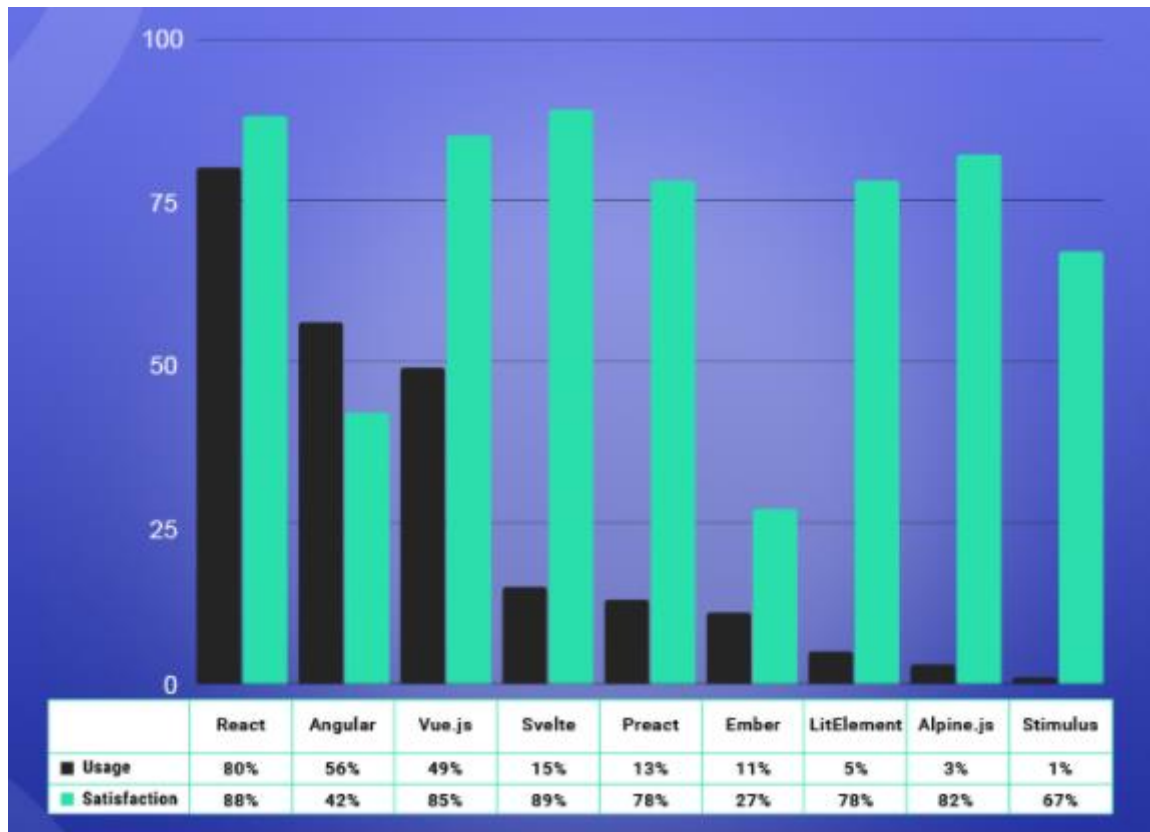
1. HTML, CSS, JavaScript are the components of the best UI frameworks.
2. A grid for designing and assembling the interface, aka DOM.
3. Font styles and sizes.
4. Website components e.g. navigation bars, hamburger menus etc.
5. Templates e.g. for registering a new user etc.
6. Buttons and side panels.

4.1 Choosing the right Frontend Framework

Building a frontend application requires the combined use of HTML, that is responsible for webpage basic layout, CSS to manage the visual formatting. JavaScript is used mainly to provide interactivity. Frontend frameworks are needed to facilitate the job of web developers: these software packages usually provide pre-written/reusable code modules, standardized front-end technologies, and ready-made interface blocks making it faster and easier for developers to craft sustainable web applications and UIs without coding every function or object from scratch.

Most of the front-end frameworks run on JS as their source language. There is too much commotion between the developer about which one is the best, but in order to choose the appropriate one, you must define, beforehand, your needs and take into account some factors.

In terms of usage and satisfaction that a framework has provided to its users/developers, the following bar chart can enlighten us about most popular choices of 2020.

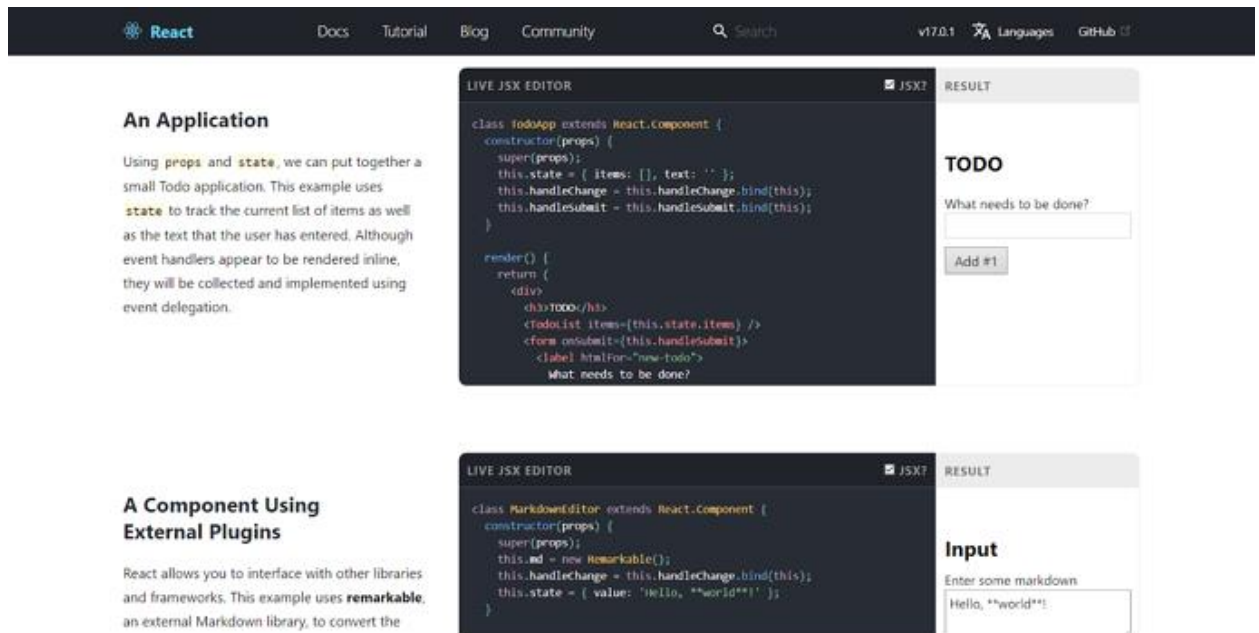


We can clearly see that React, Angular, View dominate the market in terms of popularity/usage among the developers, while Svelte, Preact, LetElement, and Alpine.js frameworks show promising levels of satisfaction.

React is still the most appreciated as a web Front-end framework, but Svelte is quickly rising to be one of its greatest rival as well. Each framework is unique and has its own advantages. Statistics may only assist one in decision making up to a point, but sheer numbers aren't enough in making the right decision while choosing the best framework.

4.1.1 React: The JS library backed by Facebook

React is probably the most popular front-end framework around. In a nutshell it is a JS component-based library featuring JSX syntax, which was developed by FaceBook and introduced in 2011, and becoming a open-source library in 2013. The most significant feature of React is a virtual Document Object Model with one-way data binding. React is also praised for its superior performance and it is considered to be one of the easiest frameworks to learn, making it a good choice for beginners or less seasoned developers. React is a library – not a framework, hence it does not provide some important features, making it essential to cooperate with other libraries in order to manage with routing, state, and interaction with the API. When the case is to save time creating reusable components, React is the ideal choice.



Programming languages used: JavaScript/JSX (HTML+JavaScript syntax).

React is steadily popular, with more than 3 million active users and supported by a vast community: 80% of developers have had a positive experience of using in their projects at least once, and more than 1.5 million websites have been made with it.

With 172k stars and 34.4k forks on GitHub, React is used by 1.2 million websites and 422k unique domains.

React is also considered superior in the following use cases:

1. Single page or cross-platform application development.
2. Data Visualization tools
3. Social Networks
4. Retail Services
5. Web apps
6. Messaging apps
7. Blogs

Due to its DOM capability, React is the best option for complex project, which require a large amount of components (navigation panels, buttons etc) that go through binary/variable states, such as active/inactive, expanded/collapsed, active/disabled, etc. It also backed by FaceBook, it is frequently updated, provides ease of migration with different versions, and is good for beginners.



Advantages

- Reusability of components
- Consistent and seamless performance
- React hooks allows to write components without classes and hence, easy to learn
- React dev tools are advanced and super useful

Disadvantages

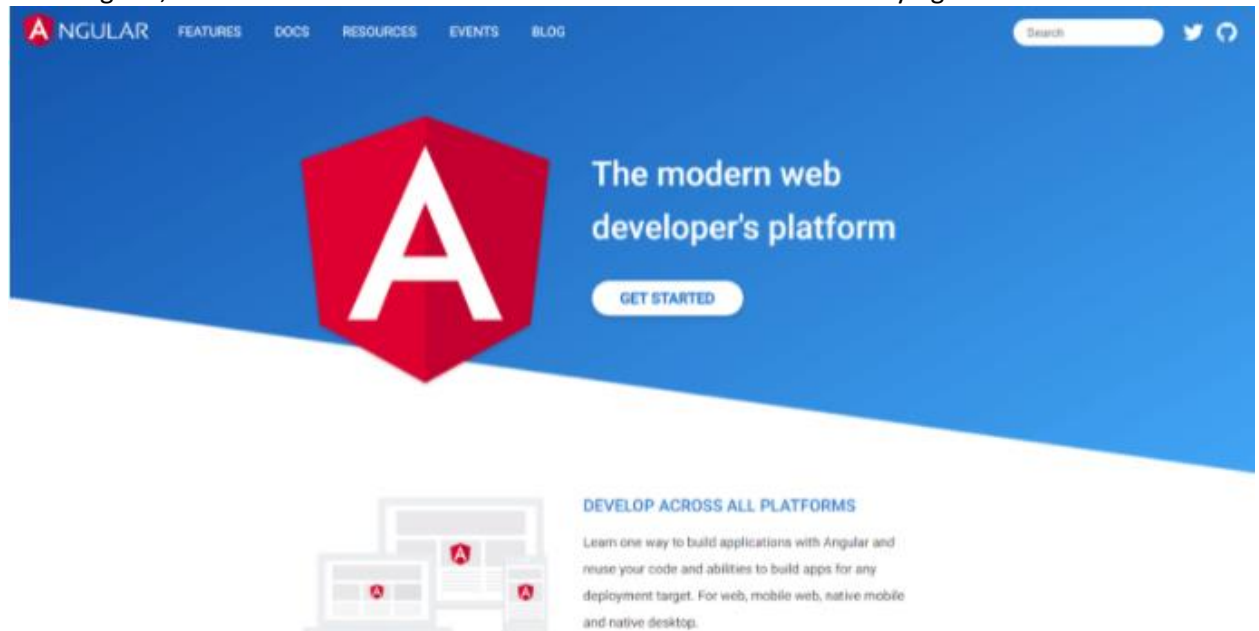
- Multiple and constant updates
- Complexities of JSX are hard to learn while beginning with the framework
- It only gives frontend solutions

Moreover, React is considered to have really easy debugging due to the fact that can be debugged before code is executed. React is also SEO friendly and can easily show up at a heavy load. This SEO support gives it a major advantage for high load applications in terms of Search Engine Optimization. React on the other hand may not be the appropriate choice for programmers that are not ready to develop on pure JS. JSX syntax can be an entry barrier for some developers that don't want to invest much time in learning it. In some cases also react is considered to have a less than necessary elaborated documentation.

Pros	Cons
<ul style="list-style-type: none">• Saving time while re-utilizing components• Virtual DOM enhances both the experience of the users and the work of the developer• An open-source library with a diversity of tools• The steady code is supplied by one direction data movement	<ul style="list-style-type: none">• Absence of documentation due to significant pace development• The comparatively long learning curve• The developers find it challenging to comprehend the complications of JSX

4.1.2 Angular.JS : The framework created by google

AngularJS is based on TypeScript, and is an open source framework, and one of the most popular software development tools nowadays, being the framework that more than 600 thousand sites have been developed with its help until now. Approximately 60% of experienced web developers have had an experience of building web applications or sites with Angular, while more than half of them believe it was efficient in satisfying their needs.



Angular supports two-way data binding for immediate synchronization between the model and the view, hence any change in the view will instantly be appeared in the model and vice versa. Featuring Directives, that help developers to program special behaviors of the DOM, Angular makes it possible to create rich and dynamic HTML content. Angular features a Hierarchical dependency injection function, which makes the developer's work easy when it comes to control, test, reuse their code components. It can help to define code dependencies as external elements decoupling components from their dependencies too.

Another strong aspect of Angular as well is its strong community, strong documentation, and good training materials that are available on the web – and it is back by Google itself too! Providing a strong server performance, Angular is all set to be the ideal choice when it comes to creating large, enterprise-scale applications. However since Angular is the complete dynamic solution, there are multiple ways which can be used in order to complete a task, so the learning curve is steeper, and it happens that the dynamic apps made with it, don't perform well sometimes because of their complexity and their size.

Angular feels to be the best choice when it come to creating browser-based application which incorporate dynamic updating of the contents fast, due to the fact that it features two-way data binding. For enterprise based applications and dynamic web apps, using Angular is a safe bet.



★ 58.4k

Made with
Angular



Forbes



Advantages

- In-built two-way data binding
- Less coding
- decoupling of components
- Usage of dependency injection and hence, more reusable components
- Vast community for learning and support

Disadvantages

- Steeper learning curve
- Code structure and size are complex in large-scale apps.

On the other hand, Angular may feel too difficult for beginners and may appear a bit overwhelming and complex for smaller teams, and maybe a bit too large in size making it overkill for smaller projects. If SEO is your primal concern then an SEO-friendlier alternative would be the best match for you.

Pros	Cons
<ul style="list-style-type: none">• Making the coding procedure easier due to its refactoring services and enhanced navigation• The component-based pattern of Angular sanctions forms a user interface with single components• Large Ecosystem• Angular Material reorganizes Material Design interface production• High Performance	<ul style="list-style-type: none">• Angular complication• Relocating legacy schemes from AngularJS to Angular• The CLI documentation is not fairly defined• The learning effort

4.1.3 Vue.js : Maybe the easiest Frontend framework

Vue was originally released in 2014. It was created by Evan You in an attempt to make a lightweight custom tool that brings the best parts of Angular, making it one of the most efficient and best UI frameworks for rapid web development. It is mostly used in order to develop one-page applications and web UIs, using an open source progressive JS framework. VueJS is used by 280 thousand websites and 174 thousand domains. It has 186 thousand stars and 29.6 thousand forks on GitHub.



Advantages

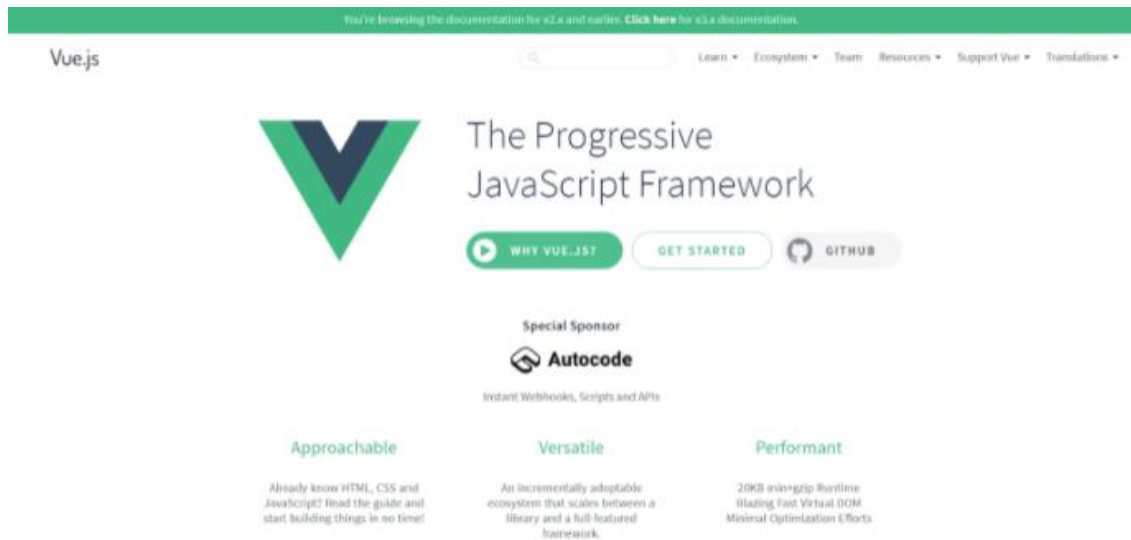
- Extensive and detailed documentation
- Simple syntax - programmers with javascript background can easily get started with Vue.js
- Flexibility to design the app structure
- Typescript support

Disadvantages

- Lack of stability in components
- Relatively small community
- Language barrier with plugins and components

Some of the most significant advantages of VueJS include it's really small size (only 20KB in size), it's simplicity – making the learning curve really smooth and it will take no time to get familiar with it. Moreover the apps that are developed in VueJS most of the times have no problem integrating with most existing apps. In terms of customization, VueJS provides the ability to assign each segment to different functions. Vue supports virtual DOM, component-based architecture, and two-way binding. Plus its speed makes a positive effect on SEO as well.

Development of Decision Support Web Application



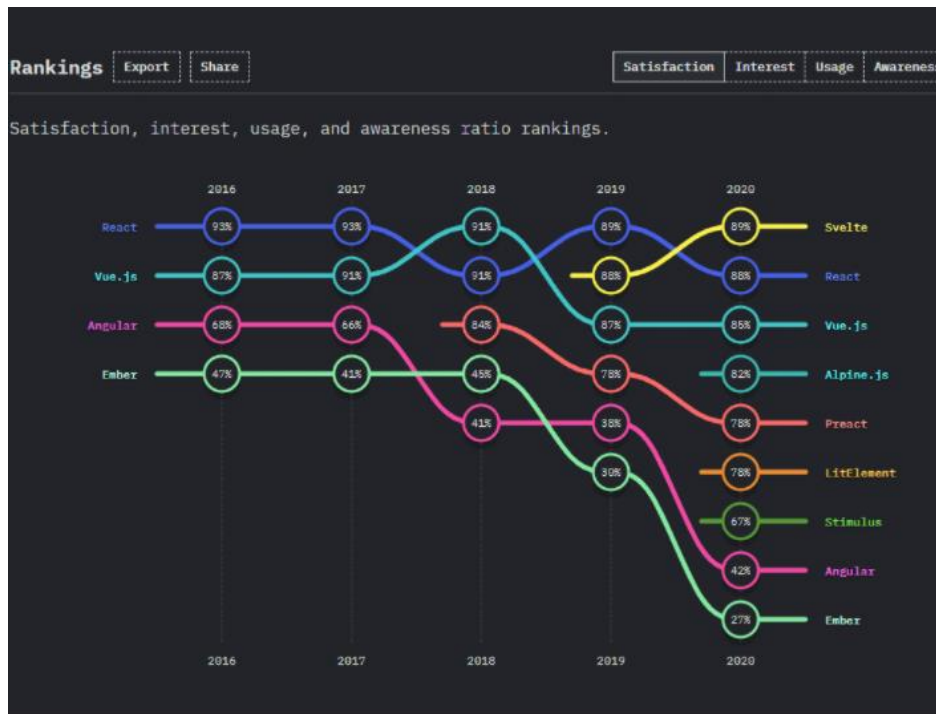
On the downside VueJS is relatively new framework and it has a small community backing it compared to the other choices. Too much content that can be found on web is Chinese as well, making it hard for non-chinese developers.

Also there is a really limited amount of plugins due to the fact that it is underdeveloped. Last but not least, not having a powerful business backing it, since it is developed by

Pros	Cons
<ul style="list-style-type: none">● Extensive and detailed documentation● Simplicity and clarity● Browser dev tools extensions● Code reusability and simple integration	<ul style="list-style-type: none">● Reduced developers' community● Flexibility leads to code irregularities

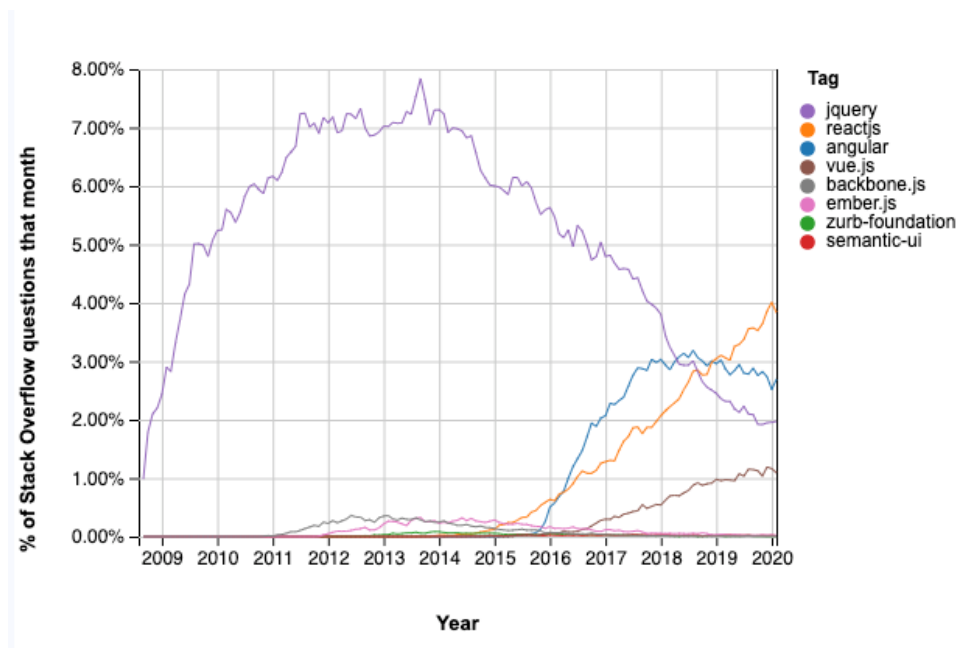
private individuals, it has limited applicability to bigger projects. And regardless the fact that it was created to tackle complexities and enhance app performance, it is not generally popular amongst the industry's giants. Nevertheless, Xiaomi, Alibaba, Reuters, and 9gag are using this framework.

VueJS shines through its simplicity and flexibility. It lets the developer create the whole project from scratch and is efficacious in building large scale projects as well. But if the lack of community support is a deal breaker for you then the Vue.js is not the right choice to make. Likewise, the applications necessitating steady components are not appropriate to be fabricated with Vue since the framework has presented difficulties with the firmness of parts.



4.2 Summary

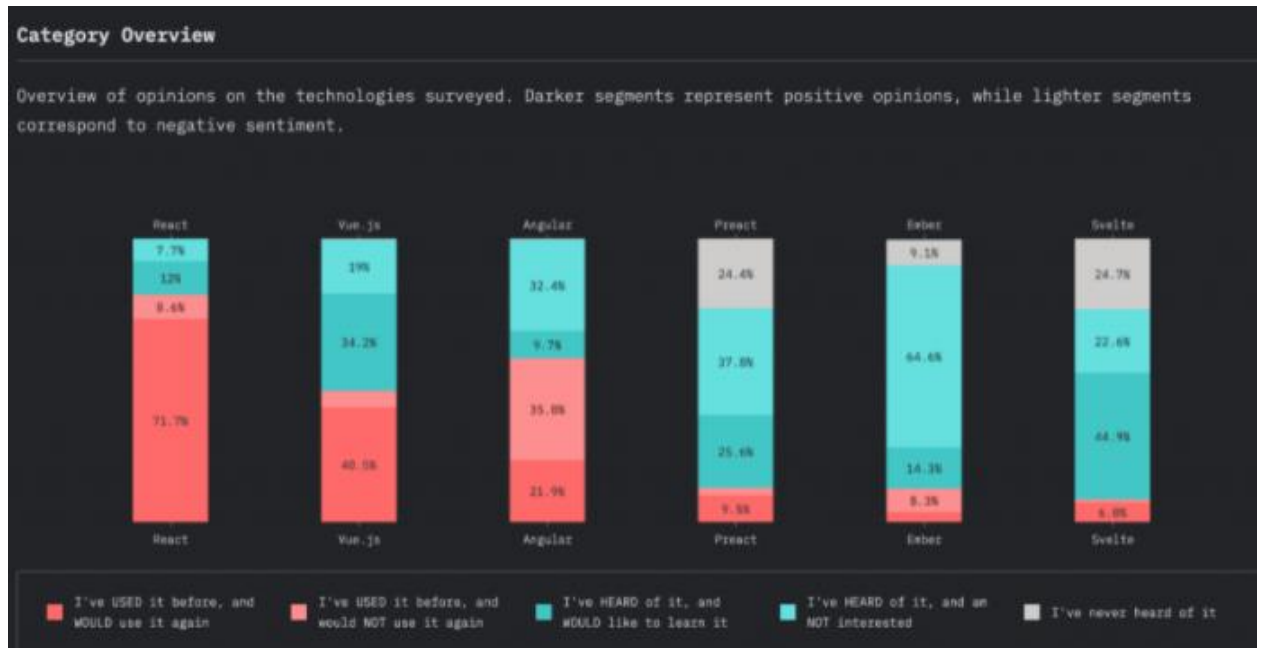
Our project will be a single page and not too complex. SEO won't be a top priority as well so top-notch performance won't be a priority as well. In such case Angular.JS seems like an overkill for building our application. Vue.JS limited community support seem a bit risky as well, since problems that may appear (I am not a seasoned Front-End developer) that may not have yet documented in English (Vue most users tend to be Chinese).



React seems a really robust choice for our project. Usage satisfaction exceeds all of the other frameworks and it is the most popular front end framework as well. Rich documentation is a

Development of Decision Support Web Application

strong aspect of the React.JS framework as well, and having a vast developer community backing it React is the safest bet in implementing the Skrouz_Robot_App web application.

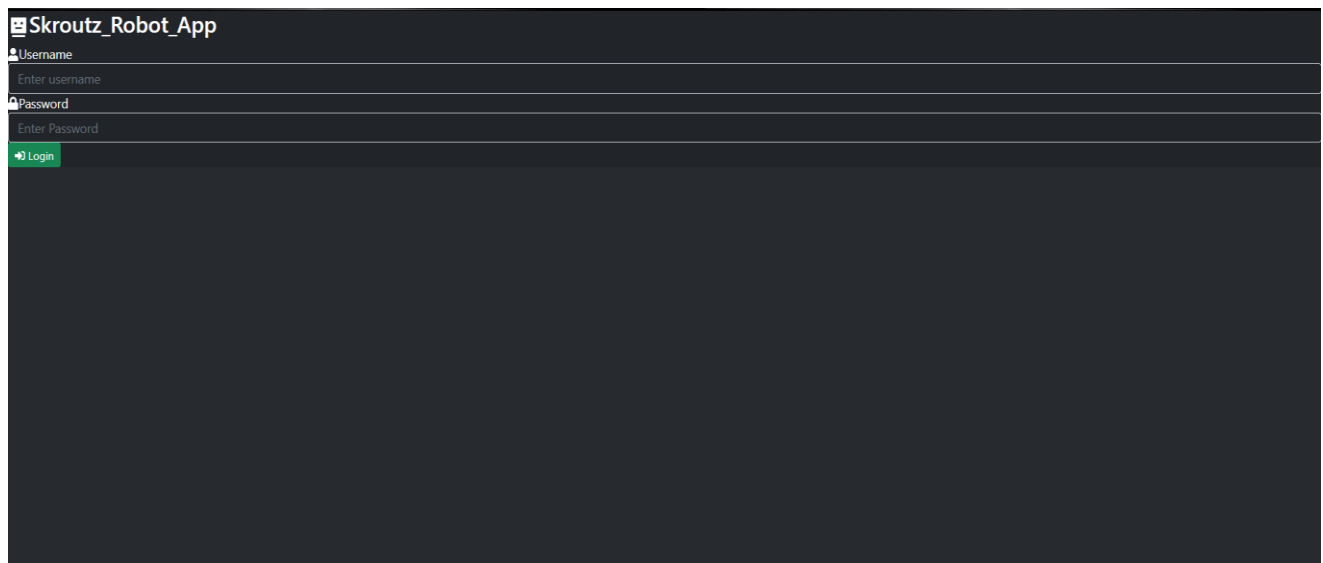


Chapter 5 : Presentation of the Application – Skroutz Robot App

5.1 The Log in Page

The application in the first level could help as an on-premise Data extraction / Data analysis tool that could potentially serve (with more additions to the data extracting model with more Data fields) as a data science tool.

The application is hosted on the local server and provides the authentication utility with username and password credentials for every user:

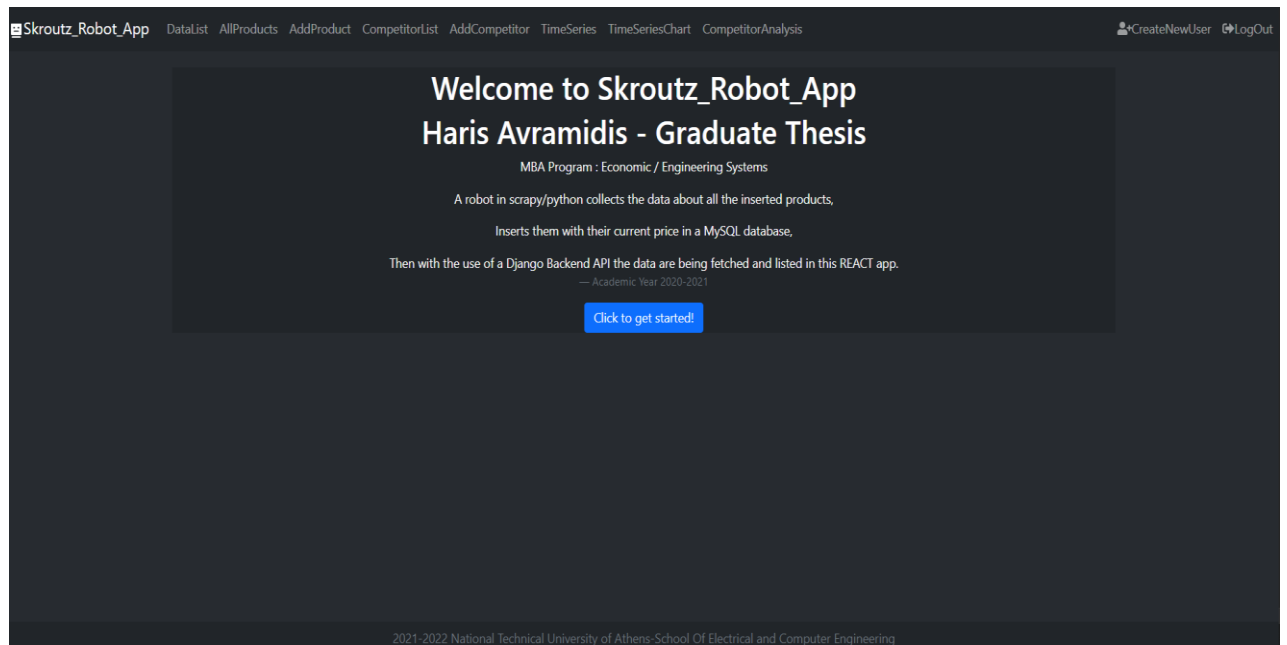


The authentication system is based on out-of-the-box Django token system.

The welcome page gives a brief explanation of how the application works from technology perspective.

5.2 Welcome page

The welcome page gives a brief explanation of how the application works from technology perspective.



The Click to get started button is designed to initialize the web scraper in order to fetch all the data for the products that have been added from the user, from the Skrouz.gr platform.

5.3 Application's Utilities

On the navigation bar the user have the following options:

1. DataList : Providing the latest data that have been extracted from the web crawler.
2. AllProducts : All products that have been added by the user.
3. AddProduct: Utility that enables user to add more products to the AllProducts list.
4. CompetitorList: A list of competitors added by the user.
5. AddCompetitor: Utility that allows the user to add a competitor.
6. TimeSeries: Providing the name of a product the user can fetch all the historical data from the database.
7. TimeSeriesChart: Providing the name of a product the user can see the historical timeseries data on a line Chart and easily compare the company's price with the best competitor price.
8. CompetitorAnalysis: Providing the name of a product the user can see a Competitor analysis report in a Bar and Pie Chart the competitors that have been detected to have to most competitive prices.

9. CreateNewUser: The user that has the administration privileges can create a new user that with his own credentials can enter and use the application.
10. LogOut : Utility that disconnects the user and deletes all the authentication token cookies.

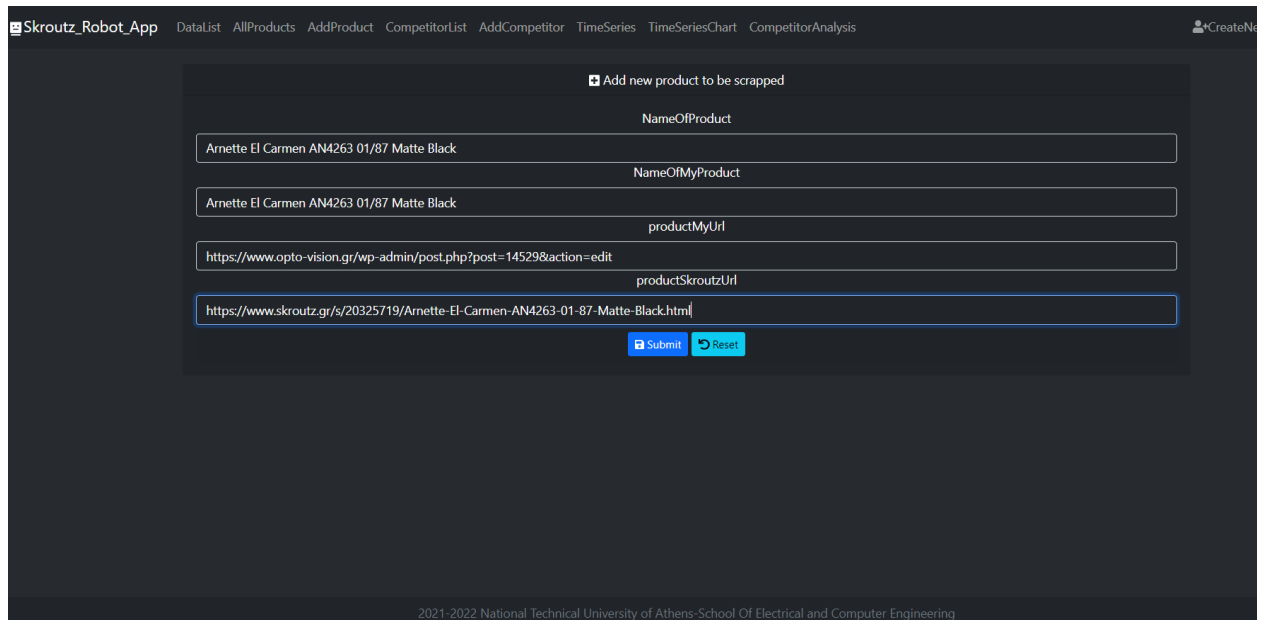
Getting Started!

5.3.1 Adding a new product.

The first the user should do is to navigate to the AddProduct tab of the application in order to add a new product.

The cooperating company (OptoVision Papaeythymioy) has selected the following product:

1. NameOfProduct (Skrouz's Product Name) : "Arnette El Carmen AN4263 01/87 Matte Black"
2. NameOfMyProduct (The eshop's product name) : "Γυαλιά Ηλίου Arnette AN 4263 01 87 Carmen"
3. productMyUrl (The Eshop's product URL) : "<https://www.opto-vision.gr/wp-admin/post.php?post=14529&action=edit>"
4. productSkrouzUrl : "<https://www.skrouz.gr/s/20325719/Arnette-El-Carmen-AN4263-01-87-Matte-Black.html>"



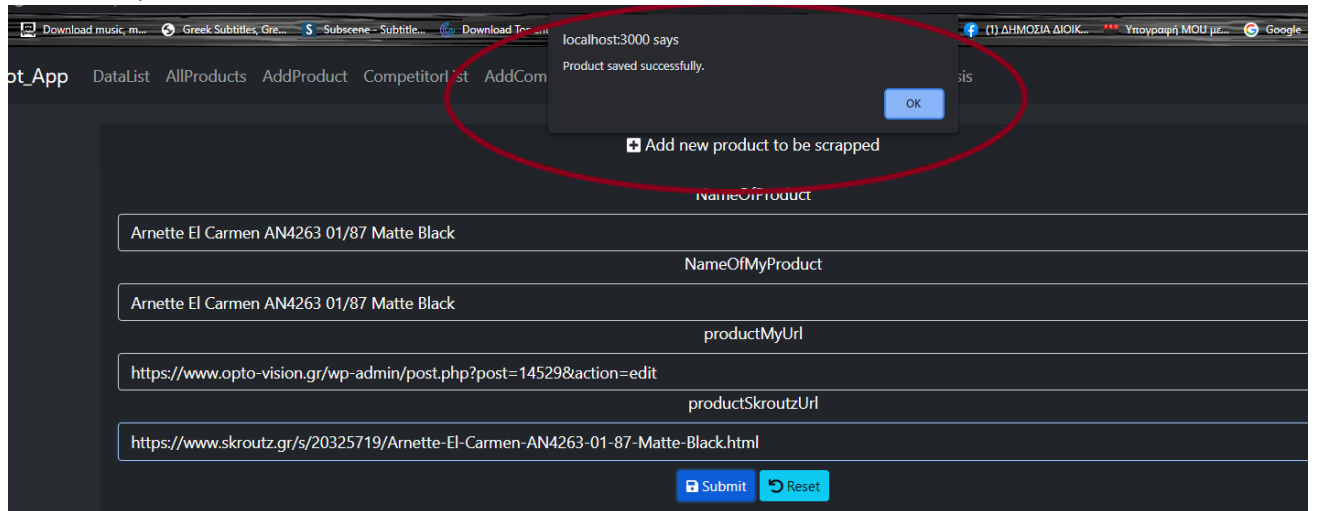
The screenshot displays the 'Add new product to be scrapped' form within the Skrouz Robot App. The form contains four input fields, each with a label above it and a value inside:

- NameOfProduct**: Arnette El Carmen AN4263 01/87 Matte Black
- NameOfMyProduct**: Arnette El Carmen AN4263 01/87 Matte Black
- productMyUrl**: <https://www.opto-vision.gr/wp-admin/post.php?post=14529&action=edit>
- productSkrouzUrl**: <https://www.skrouz.gr/s/20325719/Arnette-El-Carmen-AN4263-01-87-Matte-Black.html>

At the bottom of the form, there are two buttons: a blue 'Submit' button and a red 'Reset' button. The app's navigation bar at the top includes 'Skrouz_Robot_App', 'DataList', 'AllProducts', 'AddProduct', 'CompetitorList', 'AddCompetitor', 'TimeSeries', 'TimeSeriesChart', and 'CompetitorAnalysis'. A 'CreateNewUser' icon is visible in the top right corner. The footer text reads: '2021-2022 National Technical University of Athens-School Of Electrical and Computer Engineering'.

Development of Decision Support Web Application

After pressing the Submit Button the user can check if the product has been added successfully in the AllProducts tab:



The screenshot shows a table titled "Product List" with the following columns: "NameOfProduct", "NameOfMyProduct", "productMyUrl", "productSkrouzUrl", and "Actions". The table contains seven rows of product data, each with a corresponding "Actions" column containing edit and delete icons.

NameOfProduct	NameOfMyProduct	productMyUrl	productSkrouzUrl	Actions
Arnette-El-Carmen-AN4263-265887-Matte-Black	Arnette-El-Carmen-AN4263-265887-Matte-Black	Arnette-El-Carmen-AN4263-265887-Matte-Black	https://www.skrouz.gr/s/20314550/Arnette-El-Carmen-AN4263-265887-Matte-Black.html	[Edit] [Delete]
Arnette-AN4007-01	Arnette-AN4007-01	Arnette-AN4007-01	https://www.skrouz.gr/s/6585825/Arnette-AN4007-01.html?from=catspan	[Edit] [Delete]
Bausch-Lomb-EasySept-360ml-120ml	Bausch-Lomb-EasySept-360ml-120ml	Bausch-Lomb-EasySept-360ml-120ml	https://www.skrouz.gr/s/16035327/Bausch-Lomb-EasySept-360ml-120ml.html	[Edit] [Delete]
Ralph-Lauren-PH2224-5017	Ralph-Lauren-PH2224-5017	Ralph-Lauren-PH2224-5017	https://www.skrouz.gr/s/24403771/Ralph-Lauren-PH2224-5017.html	[Edit] [Delete]
Alcon-Opti-Free-PureMoist-300ml-60ml	Alcon-Opti-Free-PureMoist-300ml-60ml	Alcon-Opti-Free-PureMoist-300ml-60ml	https://www.skrouz.gr/s/8414500/Alcon-Opti-Free-PureMoist-300ml-60ml.html	[Edit] [Delete]
Bausch-Lomb-ReNu-Multiplus-360ml-60ml	Bausch-Lomb-ReNu-Multiplus-360ml-60ml	Bausch-Lomb-ReNu-Multiplus-360ml-60ml	https://www.skrouz.gr/s/12109270/Bausch-Lomb-ReNu-Multiplus-360ml-60ml.html	[Edit] [Delete]
Bausch-Lomb-EasySept-360ml-120ml	Bausch-Lomb-EasySept-360ml-120ml	Bausch-Lomb-EasySept-360ml-120ml	https://www.skrouz.gr/s/16035327/Bausch-Lomb-EasySept-360ml-120ml.html	[Edit] [Delete]

As you can see delete option is provided in order our user can easily have administrative utilities in the UI.

Development of Decision Support Web Application

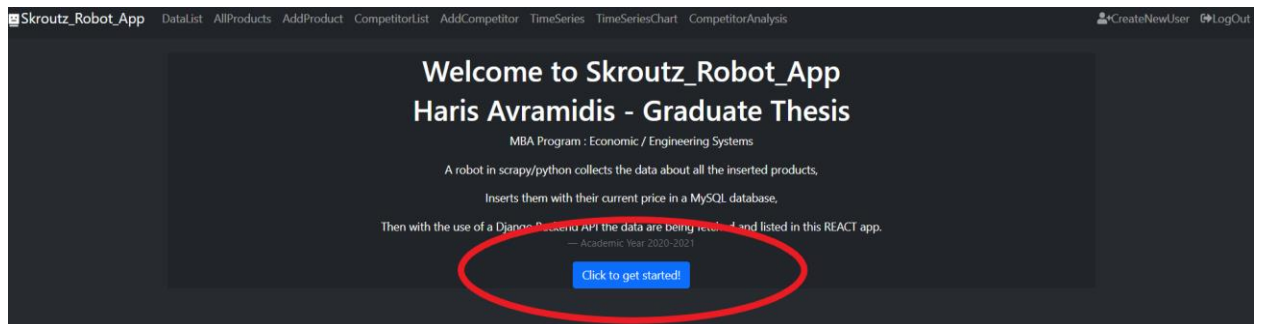
Scrolling down we can see the new product successfully added in our database:

Multiplus-360ml-60ml	Multiplus-360ml-60ml	60ml	Multiplus-360ml-60ml.html	 
Bausch-Lomb-EasySept-360ml-120ml	Bausch-Lomb-EasySept-360ml-120ml	Bausch-Lomb-EasySept-360ml-120ml	https://www.skrouztz.gr/s/16035327/Bausch-Lomb-EasySept-360ml-120ml.html	 
Alcon-Opti-Free-Express-355ml	Alcon-Opti-Free-Express-355ml	Alcon-Opti-Free-Express-355ml	https://www.skrouztz.gr/s/6869602/Alcon-Opti-Free-Express-355ml.html	 
Avizor-Ever-Clean-Pure-225ml	Avizor-Ever-Clean-Pure-225ml	Avizor-Ever-Clean-Pure-225ml	https://www.skrouztz.gr/s/6935506/Avizor-Ever-Clean-Pure-225ml.html	 
Bausch-Lomb-Biotrue-360ml-extra-Bottle-60ml	Bausch-Lomb-Biotrue-360ml-extra-Bottle-60ml	Bausch-Lomb-Biotrue-360ml-extra-Bottle-60ml	https://www.skrouztz.gr/s/11878545/Bausch-Lomb-Biotrue-360ml-extra-Bottle-60ml.html	 
Soleko-Biosee-Clear-All-In-One-Solution-380ml	Soleko-Biosee-Clear-All-In-One-Solution-380ml	Soleko-Biosee-Clear-All-In-One-Solution-380ml	https://www.skrouztz.gr/s/26087654/Soleko-Biosee-Clear-All-In-One-Solution-380ml.html	 
Amvis-AquaSoft-380ml-Extra-Bottle-60ml	Amvis-AquaSoft-380ml-Extra-Bottle-60ml	Amvis-AquaSoft-380ml-Extra-Bottle-60ml	https://www.skrouztz.gr/s/15079953/Amvis-AquaSoft-380ml-Extra-Bottle-60ml.html	 
Vogue	1	2	https://www.skrouztz.gr/s/13878315/Vogue-VO-5212S-W44-87.html	 
Arnette El Carmen AN4263 01/87 Matte Black	Arnette El Carmen AN4263 01/87 Matte Black	https://www.opto-vision.gr/wp-admin/post.php?post=14529&action=edit	https://www.skrouztz.gr/s/20325719/Arnette-El-Carmen-AN4263-01-87-Matte-Black.html	 

2021-2022 National Technical University of Athens-School Of Electrical and Computer Engineering

5.3.2 Initializing the Robot / Scraper

The new product now is ready to be scrapped and the fetched data. Then we need to initialize the Crawler from the Welcome page's button : Click to get started! to fetch the data (Daemon service Scrapyd needs to be up as well, otherwise the initiation will have to be executed from the terminal).



Development of Decision Support Web Application

The new product now is ready to be scrapped and the fetched data will be presented to us in the DataList tab:

The screenshot shows the 'DataList' tab in the 'Skroutz_Robot_App' application. The table contains the following data:

product	my_productName	price	my_price	date
Arnette El Carmen AN4263 265887 Matte Black	Not_Available	61.60000000	0.00000000	2021-09-13T12:26:35.862933Z
Arnette AN4007 01	Γυαλιά Ηλίου Arnette AN 4007 001 (Slide)	65.99000000	71.00000000	2021-09-13T12:26:36.192929Z
Bausch & Lomb EasySept 360ml + 120ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:36.420929Z
Ralph Lauren PH2224 5017	Γυαλιά Οράσεως Polo Ralph Lauren PH 2224 5017	110.00000000	113.00000000	2021-09-13T12:26:36.691928Z
Alcon Opti-Free PureMoist 300ml + 60ml	Opti-Free Pure Moist 300 + 60 ml Υγρό Φακών Επαφής (Καθαριστικό) (Alcon)	9.50000000	9.50000000	2021-09-13T12:26:37.167929Z
Bausch & Lomb ReNu Multiplus 360ml + 60ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:37.449932Z
Bausch & Lomb EasySept 360ml + 120ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:37.625929Z
Alcon Opti-Free Express 355ml	Opti-Free EXPRESS 355 ml Υγρό Φακών Επαφής (Καθαριστικό) (Alcon)	5.89000000	6.89000000	2021-09-13T12:26:38.790928Z

Scrolling down we will see the new added product data:

The screenshot shows the 'DataList' tab in the 'Skroutz_Robot_App' application, displaying a list of products. The new product data is highlighted in red:

	(Καθαριστικό) (Alcon)			13T12:26:37.167929Z
Bausch & Lomb ReNu Multiplus 360ml + 60ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:37.449932Z
Bausch & Lomb EasySept 360ml + 120ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:37.625929Z
Alcon Opti-Free Express 355ml	Opti-Free EXPRESS 355 ml Υγρό Φακών Επαφής (Καθαριστικό) (Alcon)	5.89000000	6.89000000	2021-09-13T12:26:38.790928Z
Avizor Ever Clean Pure 225ml	Ever Clean pure 225ml Υγρό Φακών Επαφής (Καθαριστικό) (Avizor)	10.00000000	11.80000000	2021-09-13T12:26:39.060932Z
Bausch & Lomb Biotrue 360ml + extra Bottle 60ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:39.388928Z
Soleko Biosee Clear All In One Solution 380ml	Βιολογικό Υγρό Φακών Επαφής Biosee ALL In One 380 ml (Καθαριστικό) (Soleko)	9.90000000	15.00000000	2021-09-13T12:26:39.594928Z
Amvis AquaSoft 380ml & Extra Bottle 60ml	AQUASoft 360+60 ml Υγρό Φακών Επαφής (Καθαριστικό) (AMVIS)	8.00000000	8.40000000	2021-09-13T12:26:39.927930Z
Vogue VO 5212S W44/87	Γυαλιά Ηλίου Vogue VO 5212S W44 87	85.00000000	96.00000000	2021-09-13T12:26:40.294929Z
Arnette El Carmen AN4263 01/87 Matte Black	Γυαλιά Ηλίου Arnette AN 4263 01 87 Carmen	54.00000000	62.00000000	2021-09-13T12:26:40.541928Z

Now to check the accuracy of the fetched data:

Development of Decision Support Web Application

Arnette El Carmen AN4263 01/87
Matte Black

★★★★★ Αξιολόγησε το προϊόν ΕΞΕΛΙΞΗ τιμής

Ανδρικό, Κατασκευαστής: Arnette

Μέγεθος: 63

Δεν είναι διαθέσιμο για αγορά με το Καλάθι του Skroutz

Διαθέσιμο σε 10 καταστήματα

Καταστήματα (10) Χαρακτηριστικά Αξιολόγησε το προϊόν Ρωτήσεις για το προϊόν Οι χρήστες είδαν επίσης Εξέλιξη τιμής

Φίλτρα Αμση Παραλαβή Ατοκες δόσεις Τελική τιμή

sky optik	ARNETTE AN4263 - 01/87 Γυαλιά με ολόκληρο σκελετό Μονόχρωμα Οργανικό Πλάστικο σκελετός 63-16-135 Παράδοση 1 έως 3 ημέρες Διαθέσιμα μεγέθη: 63	54,00 €
sunoptic	ARNETTE 4263 01/87 EL CARMEN Παράδοση 4 έως 10 ημέρες Διαθέσιμα μεγέθη: 63	61,60 €

The Best price is correctly scraped.

Regarding our price (Optovision Company):

Optovision

Γυαλιά Ηλίου Arnette AN 4263 01 87 Carmen
Παράδοση 4 έως 10 ημέρες

★★★★★ (6) Αθήνα, Αττική

62,00 €

Μεταφορικά +2,60 €
Αντακατοβολή +1,40 €
Σύνολο 66,00 €

Δες το στο κατάστημα

Opticacenter

Γυαλιά ηλίου Arnette AN 4263 El Carmen - AN4263/01/87/6316/135
Παράδοση 4 έως 10 ημέρες
Διαθέσιμα μεγέθη: 63

★★★★★ (243) Πύργος Ηλείας, Ηλεία GRECA Trustmark 0%

62,00 €

Μεταφορικά +0,00 €
Αντακατοβολή +0,00 €
Σύνολο 62,00 €

Δες το στο κατάστημα

EyeShop

Arnette - 4263 EL CARMEN 01/87 ΗΛΙΟΥ
Παράδοση 4 έως 10 ημέρες
Διαθέσιμα μεγέθη: 63

★★★★★ (22) Κορυδαλλός, Αττική

63,00 €

Μεταφορικά +0,00 €
Αντακατοβολή +0,00 €
Σύνολο 63,00 €

Δες το στο κατάστημα

Opticashop

ARNETTE AN4263 01/87 63 EL CARMEN
Παράδοση 4 έως 10 ημέρες
Διαθέσιμα μεγέθη: 63

★★★★★ (15) Εύοσμος, Θεσσαλονίκη 0%

64,00 €

Μεταφορικά +0,00 €
Αντακατοβολή +0,00 €
Σύνολο 64,00 €

Δες το στο κατάστημα

Idealeyes.gr

ARNETTE 4263 01/87
Αμση παραλαβή / Παράδοση 1 έως 3 ημέρες
Διαθέσιμα μεγέθη: 63

★★★★★ (15) Εύοσμος, Θεσσαλονίκη 0%

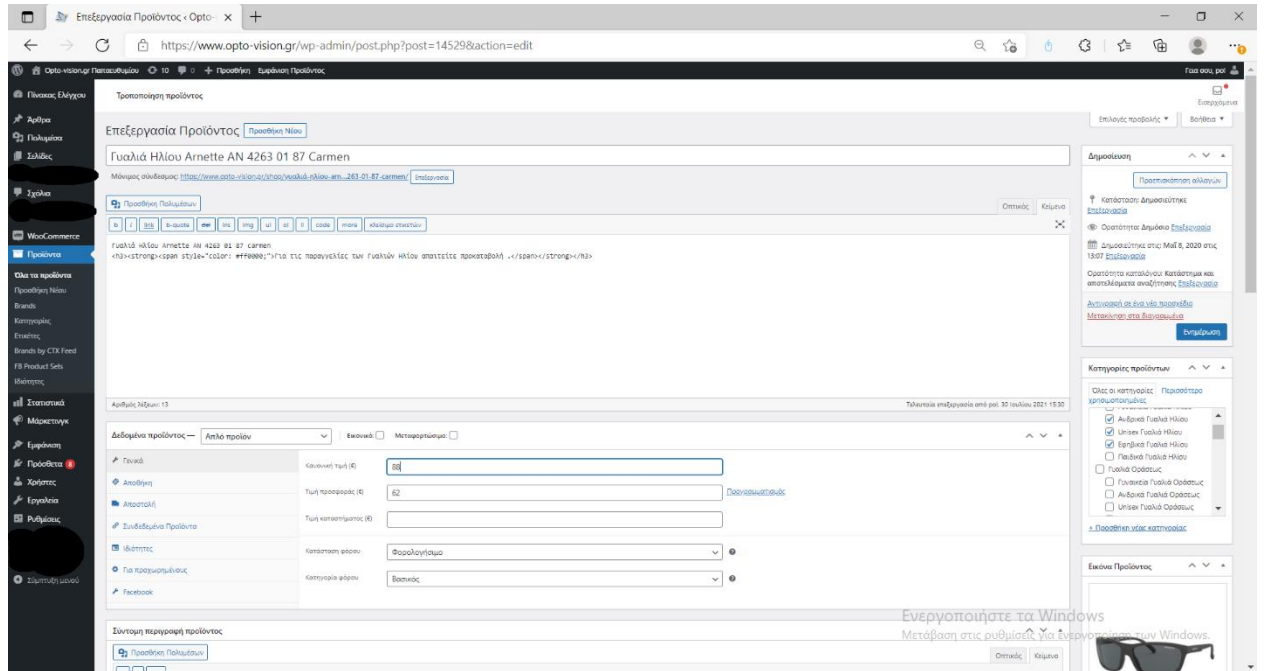
64,00 €

Μεταφορικά +0,00 €
Αντακατοβολή +0,00 €
Σύνολο 64,00 €

Δες το στο κατάστημα

Development of Decision Support Web Application

Then the user can change the price (if he wishes) if he visits the AllProducts Tab and copy/paste the url of the eshops product. (Skrutz is scheduled to refresh the prices fetching the data from the Eshop's API every 2 hours).



5.3.3 Data Analytics Utilities :

The user can review the historical data that have been fetched from the Skrutz platform.

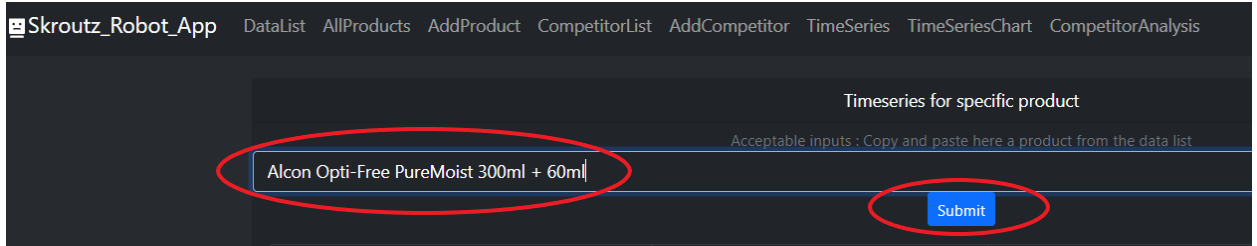
Visiting the “TimeSeries” tab the user can enter the productName and fetch the data for the selected product.

For example for the product : “Alcon Opti-Free PureMoist 300ml + 60ml”

product	my_productName	price	my_price	date
Arnette El Carmen AN4263 265887 Matte Black	Not_Available	61.60000000	0.00000000	2021-09-13T12:26:35.862933Z
Arnette AN4007 01	Γυαλιά Ηλίου Arnette AN 4007 001 (Slide)	65.99000000	71.00000000	2021-09-13T12:26:36.192929Z
Bausch & Lomb EasySept 360ml + 120ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:36.420929Z
Ralph Lauren PH2224 5017	Γυαλιά Οράσεως Polo Ralph Lauren PH 2224 5017	110.00000000	113.00000000	2021-09-13T12:26:36.691928Z
Alcon Opti-Free PureMoist 300ml + 60ml	Opti-Free Pure Moist 300+60 ml Υγρό Φακών Επαρχής (Καθαριστικό) (Alcon)	9.50000000	9.50000000	2021-09-13T12:26:37.167929Z
Bausch & Lomb ReNu Multiplus 360ml + 60ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:37.449932Z
Bausch & Lomb EasySept 360ml + 120ml	Not_Available	0.00000000	0.00000000	2021-09-13T12:26:37.675929Z

Development of Decision Support Web Application

The user should copy and paste the product name to in the TimeSeries tab.



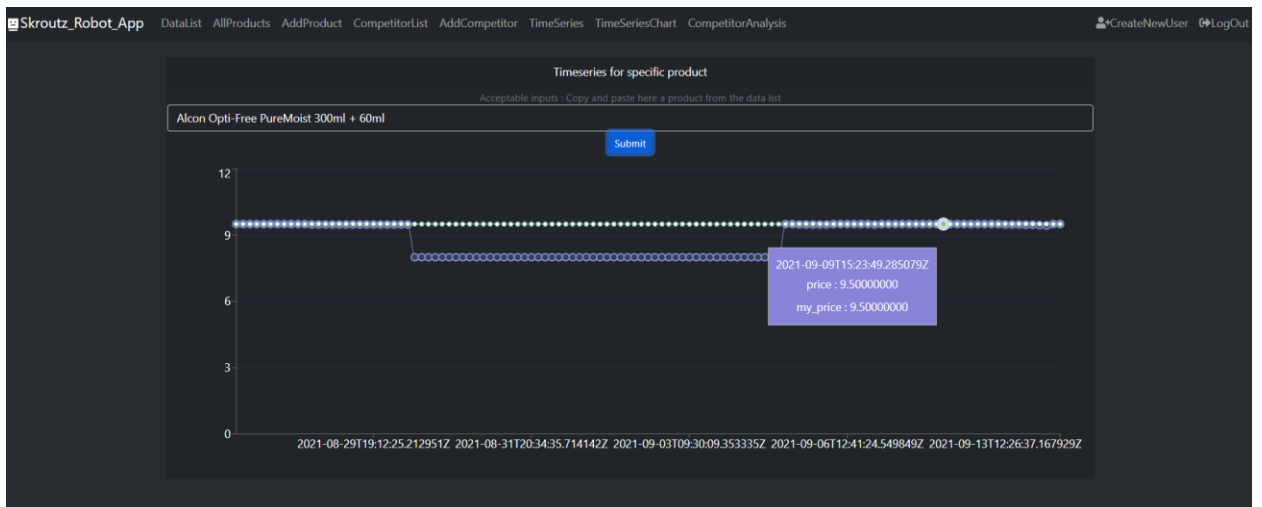
Pressing the submit button all the available data from our database will be filtered in our Django API and will be presented to us.

We can see the fetched data with the DateTime field along with the BestPriceCompany Id number (which is the skrouz's ID number for every company), MyPrice, SkrouzBestPrice.

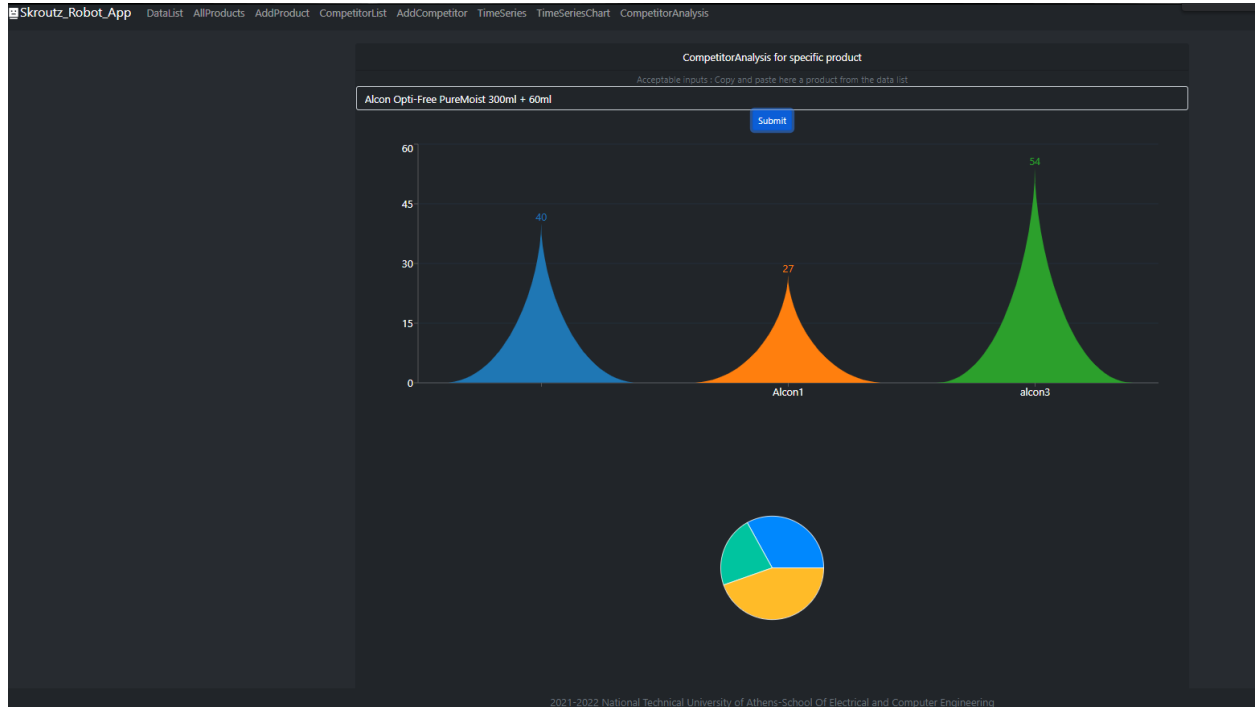
The screenshot shows the 'Skrouz_Robot_App' interface with the 'TimeSeries' tab selected. The input field contains the text 'Alcon Opti-Free PureMoist 300ml + 60ml' and a 'Submit' button is visible. Below the input field, a table displays the fetched data.

SkrouzProductName	BestPriceCompany	SkrouzBestPrice	MyPrice	DateTime
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-26T15:21:27.803525Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-26T15:22:05.634645Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-26T18:05:49.320928Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-26T18:43:19.186955Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-26T18:48:34.290091Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-27T11:55:23.876174Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-27T13:16:57.093074Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-27T15:50:15.043717Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-27T17:25:51.331248Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-27T18:50:33.391877Z
Alcon Opti-Free PureMoist 300ml + 60ml	5972	9.50000000	9.50000000	2021-08-28T09:55:20.032340Z
Alcon Opti-Free PureMoist 300ml + 60ml	14044	9.48000000	9.50000000	2021-08-28T19:19:30.272767Z

Furthermore the analyst can visit the TimeSeriesChart tab in order to see the historical data in a Line Chart:



And for a Competitor Analysis view he can visit the CompetitorAnalysis tab where bar chart and Pie Chart can assist his analysis report:

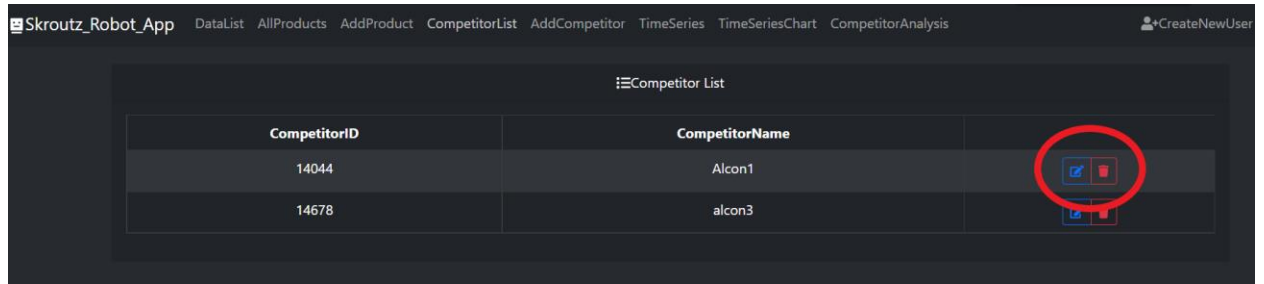


5.3.4 Adding Competitor Utility Manual:

The user can easily map the Other companies ID with a proper name in the AddCompetitor tab.

The screenshot shows the 'Add new competitor to be scrapped' form. The navigation menu at the top includes: Skrouz_Robot_App, DataList, AllProducts, AddProduct, CompetitorList, AddCompetitor, TimeSeries, TimeSeriesChart, CompetitorAnalysis, CreateNewUser, and LogOut. The form has a title 'Add new competitor to be scrapped'. It contains two input fields: 'CompetitorID' with the placeholder text 'Enter the competitor's id' and 'CompetitorName' with the placeholder text 'Enter the competitor's name'. At the bottom of the form are two buttons: 'Submit' and 'Reset'.

For when a competitor is added the user can always check his existence in our base from the CompetitorList tab where a delete option is provided for quick Administration even from our UI.



CompetitorID	CompetitorName	
14044	Alcon1	
14678	alcon3	

5.4 More potential utilities:

Of course the application is under development and with the proper additions could automate more utilities (e.g. adding new competitor) behind the scenes so could relieve work load from analyst's shoulders.

Furthermore more fields could be added and scraped, (such as Product Availability in days due to delivery, or prices/availability from other platforms e.g. BestPrice) providing new source of information for the data science team or the analyst create new reports correlating the prices with the availability etc.

Also the fact that many of the application's utilities can also be provided from the skroutz's analytic platform but that's not the case since this tool is designed for easy additions to it's current architecture, and with the proper reorganization in the user model in the backend could go online and be available for more than one companies, and for big data analytics if it were to be hosted in azure (In Virtual Machine – Synapse / Hadoop architecture and perform real time analytics using the DataBricks utility of the Azure Data Factory) .

Last but not least the structure of this project is also optimized in order to be used for every other sector of the business world (such as the commodities market, cryptocurrency market for arbitraging etc), and has also the dynamic to be fused with AI and algorithmic trading utilities (Data Science, Data manipulation with Pandas, ML with TensorFlow, Deep Learning with python etc) since its backend is implemented with Django utilizing all the available data science libraries from python.

Chapter 6 : Sources

1. <https://blog.wsi-emarketing.com/web-development-important-reasons/>
2. <https://ddi-dev.com/blog/programming/7-reasons-why-web-development-important-all-types-businesses/>
3. <https://www.checkerboard.com/seo-services/what-is-website-crawling/>
4. <https://www.grepsr.com/blog/why-is-web-crawling-important/>
5. <https://norconex.com/how-web-crawling-can-benefit-your-business/>
6. <https://medium.com/swlh/the-big-data-boom-explained-in-more-straightforward-terms-954f4eae3585>
7. <https://searchenterpriseai.techtarget.com/definition/data-science>
8. <https://www.jigsawacademy.com/blogs/business-analytics/importance-of-data-analytics/>
9. <https://businesspartnermagazine.com/5-reasons-why-data-analysis-is-important-for-every-business/>
10. <https://www.slant.co/topics/362/~best-backend-web-frameworks>
11. <https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/>
12. <https://xbytecrawling.medium.com/top-5-programming-languages-for-web-scraping-e592ce2192e4>
13. <https://flatironschool.com/blog/data-science-programming-languages>
14. <https://www.promptcloud.com/blog/best-programming-language-for-web-scraping/>
15. <https://www.educba.com/django-vs-laravel/>
16. <https://www.guru99.com/flask-vs-django.html>
17. <https://www.guru99.com/flask-vs-django.html>
18. <https://medium.com/@matkowaleczko/web-frameworks-in-analytics-and-data-science-a8b4d9bd7083>
19. <https://docs.djangoproject.com/en/3.2/ref/databases/>
20. <https://www.xplenty.com/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/>
21. <https://stackoverflow.com/questions/9540154/which-database-engine-to-choose-for-django-app/9540312>
22. <https://nickmccullum.com/best-database-django-web-apps/>
23. https://en.wikipedia.org/wiki/Web_scraping
24. <https://www.zyte.com/learn/what-is-web-scraping/>
25. <https://www.analyticsvidhya.com/blog/2017/07/web-scraping-in-python-using-scrapy/>
26. <https://it-s.com/5-best-programming-languages-for-web-scraping/>
27. <https://www.zyte.com/learn/what-is-web-scraping/>
28. <https://it-s.com/5-best-programming-languages-for-web-scraping/>

29. <https://yourstory.com/mystory/530e1cb78f-the-best-programming-l/amp>
30. <https://www.promptcloud.com/blog/best-programming-language-for-web-scraping/>
31. <https://geekflare.com/web-scraping-frameworks/>
32. <https://xbytecrawling.medium.com/top-8-python-based-web-crawling-libraries-14c08cba3fdf>
33. <https://www.bestproxyreviews.com/scrapy-vs-selenium-vs-beautifulsoup-for-web-scraping/>
34. <https://www.proxyrack.com/scrapy-vs-selenium-vs-beautiful-soup-which-is-best-for-web-scraping/>
35. <https://limeproxies.netlify.app/blog/selenium-vs-beautifulsoup>
36. <https://www.zyte.com/learn/what-python-web-scraping-tools-are-available/>
37. <https://elitedatascience.com/python-web-scraping-libraries>
38. <https://limeproxies.netlify.app/blog/everything-about-using-proxy-in-scrapy>
39. <https://www.zyte.com/blog/scrapy-proxy/>
40. <https://www.ideamotive.co/blog/best-frontend-frameworks>
41. <https://www.monocubed.com/best-front-end-frameworks/>

