

College of Business, Technology and Engineering

**Department of Computing  
Project (Technical Computing)  
[55-604708]  
2020/21**

<b>Author:</b>	Chrysovalantis Papadimitriou
<b>Student ID:</b>	B8019806
<b>Year Submitted:</b>	2021
<b>Supervisor:</b>	Steele Bob
<b>Second Marker:</b>	Mike Meredith
<b>Degree Course:</b>	Computer Science
<b>Title of Project:</b>	Stock Market analytics and general news for non-professional traders

**Confidentiality Required?**

YES / **NO**

I give permission to make my project report, video and deliverable accessible to staff and students on the Project (Technical Computing) module at Sheffield Hallam University.

**YES** / NO

# **Stock Market analytics and general news for non-professional traders**

By Chrysovalantis Papadimitriou

## Table of Contents

ABSTRACT: .....	5
1 INTRODUCTION .....	6
1.1 IDENTIFYING THE PROJECT.....	6
1.2 SOLUTION TO THE PROBLEM (PROPOSE) .....	6
1.3 AIMS AND OBJECTIVES. ....	7
1.3.1 WEBSITE DESIGN.....	7
1.3.2 DATA COLLECTION AND ANALYSIS. ....	8
1.3.3 FINALIZING PROJECT. ....	8
1.4 PROJECT OUTLINES .....	8
2. RESEARCH .....	9
2.1 WEB DEVELOPMENT.....	9
2.1.1 USER-FRIENDLY WEBSITE .....	9
2.1.2 WEBSITE DESIGN (NAVIGATION & COLOUR SCHEME) .....	10
2.1.3 BOOTSTRAP .....	11
2.2 STOCK MARKET.....	11
2.2.1 TECHNICAL ANALYSIS .....	11
2.2.2 SENTIMENT ANALYSIS.....	12
2.3 SOFTWARE/TOOLS OVERVIEW.....	13
2.3.1 JUPITER NOTEBOOK .....	13
2.3.2 NLTK.....	13
2.3.3 VADER .....	13
2.3.4 BS4 (BEAUTIFUL SOUP) .....	14
2.3.5 TENSORFLOW .....	14
2.3.6 DJANGO .....	14
2.4 ORGANISATIONAL TOOLS .....	15
2.4.1 GITHUB (VERSION CONTROL).....	15
2.4.2 TRELLO (ORGANIZATION TOOL).....	16
3. DESIGN/METHODOLOGY .....	16
3.1 DESIGN .....	16
3.2 PROGRAMMING LANGUAGE .....	17

3.3 PLATFORMS/SOFTWARE .....	17
3.3.1 ANACONDA .....	17
3.3.2 PYCHARM .....	18
3.3.3 VISUAL STUDIO CODE.....	18
4 DEVELOPMENT.....	20
4.1 ORGANISATIONAL TOOLS / VERSION CONTROL .....	20
4.2 WEBSITE .....	20
4.2.1 ENVIRONMENT SETUP .....	20
4.2.2 WEBSITE DESIGN DEVELOPMENT.....	21
4.3 DATA ANALYSIS .....	23
4.3.1 DATA COLLECTION .....	23
4.3.2 SENTIMENT ANALYSIS.....	25
4.3.3 ANALYSIS PANEL .....	27
4.3.4 ML FUTURE ANALYSIS (PREDICTION).....	28
4.4 PYCHARM .....	31
4.4.1 ENVIRONMENT SETUP .....	31
4.4.2 WEBSITE DESIGN MIGRATIONS .....	33
4.4.3 DATA ANALYSIS MIGRATIONS .....	34
5. TESTING & EVALUATION .....	36
5.1 WEBSITE .....	36
5.2 ALGORITHMS EVALUATION.....	37
6. CRITICAL REFLECTION .....	42
6.1 RESULTS/CONCLUSION .....	42
6.2 PROBLEMS/LIMITATIONS .....	42
6.3 PROPOSE SOLUTIONS .....	43
6.4 FUTURE IMPLEMENTATIONS .....	43
7. REFERENCE LIST .....	43
8. APPENDIX.....	48
8.1 IMPORTS .....	48
8.2 QUESTIONNAIRE .....	49

## ABSTRACT:

This project was to create a user-friendly website that aims to provide to the user's general stock analytics and predictions based on their desire companies. Inspiration was taken from my personal need for a good free-to-use tool that will be able to provide helpful information's to backup stock market research, evaluations, and save extensive time to the users from double-checking each research. The first part of the research was focused on the definition of a user-friendly website and what defines a good user-friendly website. The second part was focused on data collection and data analysis based on stock market topics. The research proved that big data collections can be collected, analysed, and visualised back to the website in a user-friendly way. With research complete, a prototype website was created and evaluated by a questionnaire focused on the design aspect of the website. After the completion of the website template. A merging process started to bind together the functional part of the project that was developed and tested using Jupiter Notebook (python programming language) with the Designed website using Django and PyCharm.

# 1 INTRODUCTION

This project aims to create a website to be used by stock traders to find relevant data and news by collecting datasets. Those datasets are used to present the selected company general stock price from 2012 until this day, Analyse the latest 100 articles titles and creating a sentiment score for each one then sorts them into days and present the average happiness score of each day on a bar graph. Adding to this a general graphic bar will present the happiness score for all 100 articles on a scale from sad to good using emojis. Next, a table will visualise the selected company price fluctuations for the last day, last week and last month. After the analytic panel users also have the option to use machine-learning algorithms to see a small prediction for the future 30 days of the selected stock price. **Disclaimer those data created and visualised should not be taken seriously without your own search evaluation and critical thinking. This research aims to back up your own research not to be used as a financial instruction!!**

## 1.1 IDENTIFYING THE PROJECT.

Inspiration for this project presented itself when started trading socks.

In Trading, one of the key points if not the most important one is to be able to access every possible information source and compare them before taking any actions. Each action performed without a logical assumption (knowledge) could be described as GAMBLING.

My first conclusion was, if I search the news/updates and be up-to-day with the market I would be able to predict the market.

But after some time in the industry, a new problem arose. Can you truly trust the easily accessible and mass directed information? Who can say that what he/she is reading is real/fake/hipped? There is so much information available these days that may lead many of us to a false conclusion and get easily fooled or misdirected.

### Questions:

- How an Individual can be surer that what he/she is reading is true or fake?
- How to identify fake news?
- How to label news as true?
- How an individual can be consistent in his/her decisions?
- Is there a way to answer all the above questions?

## 1.2 SOLUTION TO THE PROBLEM (PROPOSE)

A solution to the problem would be to compare massive data collections and produce some meaningful results out of the data collected (price graph, price analytics). Adding further to this

solution we need to also consider all the latest news that will also help us start picturing a view of the future and how the company performs in the present time!

Latest news analysis Example:

We are limiting the articles to 100 so we only take the latest available and by performing a sentiment analysis on their topic (Title) we can start generating a sentiment score or in other words a happiness score that we can then use to create and visualise graphs. Those graphs will demonstrate how positive negative or natural selected company news are. This is very useful news that affects the price of a stock company very often based on how positive or negative the news are.

So now the user will have access to the following. That can be used to back up his/her own research and have an extra perspective.

- A price graph from 2012 until this day with
- a bar Graph presenting how positive/natural/negative the articles are sorted by the day that the articles were published.
- Average happiness graphic bar based on all 100 articles.
- Table with daily, weekly, and monthly price fluctuations. Demonstrated by showing the percentages.

### *1.3 AIMS AND OBJECTIVES.*

1. Create a Modern and user-friendly website.
2. Collect data from the web.
3. Analyse data collected and create statistics (counts, percentages).
4. Finalizing the project

By achieving these aims, the project deliverable will either prove or disprove if collecting data and analysing them can create a decent analytic outcome and have consistent results.

To achieve these aims, objectives will need to be met. The project consists of 3 Big milestones creating the website, collecting and analysing data, and finalizing the project. Let us have a look at the objectives for each component.

#### *1.3.1 WEBSITE DESIGN.*

To complete this milestone, the following will need to be completed:

- Research modern designs for websites

- Research what makes a website user-friendly
- Create a website based on the above research.
- Create questionnaires to address the user-friendly aspect of the website.
- Change the website based on the feedback from the questionnaires.

#### Once the website has passed the finalizing steps

- Fix any issues that may arise after the process of finalizing.
- Second questionnaires for adjustments

### 1.3.2 DATA COLLECTION AND ANALYSIS.

To complete this milestone, the following will need to be completed:

- Research how to crawl data from the web.
- Research how to use data to produce analytics.
- Implement a solution for data capture.
- Implement functions for data analytics.

### 1.3.3 FINALIZING PROJECT.

To complete this milestone, the following will need to be completed:

- Merge of milestone 1 and 2.
- Fix errors from the merging process
- Finalize messages and users interactable.

## 1.4 PROJECT OUTLINES

The project can be summarised as:

Is there a way to help investors be more accurate and confident in their predictions? By also saving them time and providing them with useful resources that can be feather used and increase their overall research.

The proposed solution to this:

Is to create a modern user-friendly website which the users can use to access analytics and data on their preference searches.



The next chapter will focus on research. This will be on current solutions to the problem and similar projects.

## 2. RESEARCH

In this section, we will investigate various aspects related to the overall objectives. That could benefit in the selection of tools, software, and design of the project with brief explanations of their potential benefits.

### 2.1 WEB DEVELOPMENT

In this section, we will talk about the background research around websites that led to the formation of key bullet points that could serve as a guideline for the development progress of the Website design.

#### 2.1.1 USER-FRIENDLY WEBSITE

Research carried to be able to better understand the concept of websites and what defines a website as user-friendly and modern. This section will focus on all the technics collected to be used to achieve the desire website outcome. User-Friendly websites are all about offering the users/visitors the information they need quickly and easily. After extensive research on user-friendly websites, these are the top key points sorted from more popular to least popular, with their respectful benefits.

- Lighting speed loading x5

Slow and dawdle pages minimize user's satisfaction and experience rate as they are very frustrating and annoying to deal with. To improve on this compress images size before loading, test speed using Google chrome developer tool, remove additional elements, not in use.

- Mobile x5

Following the latest technological advancements also a good website has the needs to be mobile-friendly. Website these days also get penalized if are not optimized for mobile devices making it essential to use responsive tools and design methods. Use optimized images to minimize bandwidth and scaling issues. Suggested image types: GIF, PNG-8, and JPEG.

- Navigation x4

The importance of good navigation is also one of the top aspects of website usability as it provides to the user a way to orient themselves with pages. User-friendly navigation can be described by the Minimized number of items, easy to locate, remove sub-menus, and an informative header, and footer.

- Colour scheme x4

The use of one colour scheme to reduce confusion from the users also provides brand consistency and give a website identity that users can identify easily. Furthermore, colours need to make sense with the industry of the website and provide a clear separation between background colours and text so users can read easily the content.

- Information section x4

Readers need to grasp information quickly, using bullet points help users get the information needed quickly, solve their problems, and get key features all in a shorter time rather than extensive information sections. This also helps to isolate the most important parts of the information section needed to the user. Adding to this think like the user will also help to create the most reliable information points.

- Listen to your users x3

User feedback will help target or discover missing elements and problems that developer eyes cannot pick up. Also, the user may not be happy with the general design or some elements. Those feedbacks will help to turn negative opinions into positive ones and give the user what exactly they need. Also, listening to the users automatically make a website more user-friendly as the user controls or have an impact on the design of the website.

- Use of white spaces x2

White spaces enhance the experience of a user by enabling them to better focus on the elements, also gives a more modern, fresh, and open feel to the website. Adding more to this white space give relief to the user's eyes, extending the user period on the website and their experience.

### 2.1.2 WEBSITE DESIGN (NAVIGATION & COLOUR SCHEME)

A second stage of the theoretical website research focused on Navigation types and colour schemes. The aim was to explore and collect all possible options that could be used but also follow the previous guidelines of a user-friendly website so they can be used for the design.

#### Navigation

Website navigation has become an essential component in UX because it can either support or hinder the users' interaction with a website. It is similar to the foundation of a home. If a foundation plan fails, the house is at risk of collapsing. There are no fixed instructions or how-to organise navigation because it varies so much between websites. To help the selection of the navigation design a list with the most Modern and popular navigations was created.

- Interactive Navigation
- Sidebar navigation
- Hamburger menu
- Responsive Navigation
- Horizontal Navigation
- Vertical Menu
- Drop-Down Menu

## Colour Scheme

What does exactly the website colour scheme mean? It goes beyond the colour of the logo or the history of the website. It includes any colour that appears on the website, from the header and pages to the text and button colours and even the hues used in the images. Both elements contribute to the overall colour scheme, which influences the feel, mood, and aesthetic of the website. The website's colour scheme influences the user's view of the brand on many levels. It can influence anything from how long they stay on your page to whether they use any features of the website. Different colour schemes elicit different emotions, for example, red elicits confidence, yellow elicits fun, and blue elicits trust. This can ensure the tone or emotions that a website wants to pass to the users. Like with the Navigation a list with possible colour schemes was created.

- Neon tone and sharp contrast
- Neon and bold
- Warm and bold
- Earth tone red and green
- Classic double blue colour
- Greenery and pearl
- Intense green, blue, and red colour scheme

### 2.1.3 BOOTSTRAP

Bootstrap is a robust toolkit that includes HTML, CSS, and JavaScript tools for designing and developing web sites and web applications. It is free and open-source software that was developed by Twitter and is hosted on GitHub. Bootstrap is a very versatile and user-friendly framework. Its key benefits are that it retains broad browser stability, that it provides stable design by using re-usable modules, and that it is very easy to use and understand. It provides extensive JavaScript extensibility, including built-in support for jQuery plugins and a programmatic JavaScript API. Also Bootstrap includes simple HTML and CSS interface models with several different UI components that could be very helpful in the designing of a user-friendly and modern website. Typography, Tables, Forms, Buttons, Glyph icons, Dropdowns, Buttons and Input Groups, Navigation, Pagination, Labels and Badges, Alerts, Progress Bars, Modals, Tabs, Accordions, Carousels, and several other elements are used.

## 2.2 STOCK MARKET

In this section, we will talk about the logical research carried to achieve the necessary knowledge of how a stock market analysis could be performed with key points to be considered when developing the project.

### 2.2.1 TECHNICAL ANALYSIS

Technical analysis is the study of past and current pricing behaviour to forecast the likelihood of possible market fluctuations. Technical analysts examine the stock market as a whole, focusing on price and value, as well as the demand and supply forces that influence the market. Charts are an important instrument for technical analysts because they provide a graphical representation of a

stock's trend over a specified time frame. A technical analyst may use a map to label certain areas as a support or resistance level. Previous lows below the current share price serve as support markers, whereas previous peaks above the current stock price serve as resistance markers. A drop below the support level indicates a bearish pattern to the stock analyst, while a break above the resistance level indicates a bullish trend. Technical stock analysis is only useful when supply and demand forces affect the price pattern under consideration. When outside factors are influencing a price movement, stock analysis using technical analysis cannot be accurate. Other causes that can impact a stock price besides supply and demand include stock exchanges, mergers, dividend announcements, a class action case, news impacts, financial irregularities, managerial changes, monetary policy changes, and so on. Technological research may be performed separately or in tandem.

Key aspects of Technical analysis to be used for the functional part of the project:

- Monitor past and current pricing behaviour.
- Market fluctuations.
- Focus on the price and value.
- Use of charts for stock representation.

Drawback

- Outside factors decrease accuracy.

### 2.2.2 SENTIMENT ANALYSIS

Market sentiment is a qualitative indicator of investors' attitudes and moods toward capital markets in general, and individual industries or properties. Price movement is guided by positive and negative sentiment, which creates trading and investing opportunities for professional traders and long-term buyers. It can be described as the aggregated popular attitudes, beliefs, emotions, mood, or outlook that comprise market psychology at any given time. Since market sentiment cannot be precisely described or calculated, there is no right or wrong approach to perform sentiment analysis. However, there are methods like the Vader tool for using and combining other metrics that represent market sentiment.

#### Why emotions are important in the stock market

In the short term, markets are influenced by sentiment, especially fear and greed. Traders and buyers are often motivated by one or more psychological needs. The fear of losing out will cause investors to pay exorbitant rates for an asset. In that case, they are not purchasing because the commodity is a safe purchase, but because they need to do something to stop getting left out. During bear markets, buyers will often sell securities at values well below their intrinsic value to alleviate the pain of losing money. These are two examples of how feelings can cause investors to make irrational decisions. It also demonstrates that stock market peaks and lows are often characterised by intense levels of positivity and negativity. The sentiment is at its peak right before major market highs and at its lowest

just before major market bottoms. **Investors should use sentiment analysis to try to decide whether the market is being influenced by fear rather than sound decision making. They can detect shifts in the mood until there is any news to justify stock market conduct.**

## 2.3 SOFTWARE/TOOLS OVERVIEW

In this section, we will talk about the research of the software and technologies that could be used to implement some of the most important aspects of the project their background, and a list of benefits that could be beneficial to the project.

### 2.3.1 JUPITER NOTEBOOK

Originally designed for Python-based data science applications. Jupiter was selected for the benefits it could offer for the development of the functional part of the project.

#### Jupiter Benefits

- **Visualizations of data:** Most people are introduced to Jupiter Notebook with a data visualisation, which is a shared notebook that provides a graphic rendering of some data collection. Jupiter Notebook allows you to create visualisations, but also exchange them and make modifications to the shared code and data collection interactively.
- **Coding samples are being recorded:** You could embed a Jupiter Notebook if you have a piece of code and want to illustrate it line by line, with live comments all along the way. The best part is that the code will be entirely functional.

### 2.3.2 NLTK

The Natural Language Toolkit (NLTK) is a leading free open-source tool for developing Python programs that work with human language details. It offers user-friendly interfaces to over 50 corpora and lexical tools, including WordNet, as well as a suite of text processing libraries for grouping, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries. That could be used in the project analysis development.

#### NLTK Benefits

- Leading NLP library (Extensive 3<sup>rd</sup> party extensions)
- Well documented

### 2.3.3 VADER

Vader stance for Valence Aware Dictionary for Sentiment Reasoning is a text sentiment analysis model that is adaptive to both the polarity (positive/negative) and the force (strength) of emotion. It is included in the NLTK kit and can be used on unlabelled text data directly. VADER emotional research

is based on a dictionary that correlates lexical characteristics to emotion intensities, which are known as sentiment ratings. A text's sentiment score can be calculated by including the strength of each word in the text. VADER is also wise enough to recognise the simple meaning of these sentence. It is also able to recognise the importance of capitalization and punctuation. Vader is not going to try to figure out whether a statement is factual or subjective, fact or belief. Rather, it's only interested in whether the document expresses a constructive, negative, or neutral viewpoint. In more general terms Vader attempt to synthesise all the sentences in a text or paragraph to form an overarching view.

Vader can be applied using `".apply(vader.polarity_scores)"` on a data frame to calculate the sentiment score.

#### 2.3.4 Bs4 (BEAUTIFUL SOUP)

Beautiful Soup is a Python library that extracts data from HTML and XML formats. It integrates with your preferred parser to include idiomatic methods of browsing, scanning and changing the parse tree. In more general terms it's able to scrape information from the web.

##### Bs4 Benefits

- Quick data extraction
- Easy to use (Well documented)
- Community support

#### 2.3.5 TENSORFLOW

TensorFlow is an open-source library for machine learning and numerical computation on a wide scale. TensorFlow is a simple concept that incorporates a variety of machine learning and deep learning (also known as neural networking) models and algorithms. It uses Python to provide a user-friendly front-end API for creating platform-based applications.

##### Benefits

- TensorFlow will train and operate deep neural networks at scale to enable output prediction, using the same models that were used for testing. That prediction could be used with the stock price values to train and predict into the future.

#### 2.3.6 DJANGO

The Django web framework is a free, open-source project that can help speed up the creation of a web application written in Python. When installed on a web server, the Django web framework will assist developers in quickly creating a feature-rich, stable, and scalable web project.

##### Django Benefits

- Make database administration more Python-like.

Starting a Django project helps you to create the entire data model of your application in Python without having to use SQL. Django transforms conventional database structures into Python classes using an object-relational mapper (ORM) to make it simpler to operate within a completely Python framework. The JSON data form and associated functions are supported by Django-MySQL. Django Web Framework is a shortcut to complete database integration, it supports CRUD (create, read, edit, and delete) functionality, HTTP Response, cross-site scripting, user management, software administration, and other functions.

- Using Templates, to build interactive websites.

Since Django is intended for web app creation, it requires an easy way to generate dynamic HTML that shows the user's unique data. The Django framework generates complex HTML using a built-in templating engine known as the Django template language (DTL). An HTML prototype enables Django developers to merge static elements (such as colours, icons, or text) with data (such as usernames or locations) to generate a new web page on the fly. When the website is rendered, the dynamic elements will be combined with the static elements to create a unified user interface.

- Boost security

Django has tools that can help secure any project and its users. One of the most serious dangers for sites that allow user-entered data is that a malicious user can insert code into their data, which would have catastrophic consequences for the device. Django templates immediately escape standard HTML characters in any user-entered area to defend against such attacks. For example, it would automatically translate '<' to '&lt;' to make injecting malicious code into a web more difficult. Django similarly defends from SQL injection, reinterpreting unauthorised commands to prevent users from inserting their code into the database. Web developers may also rely on Django APIs to automatically inject user-specific hidden tokens into POST requests using cross-site request forgery (CSRF) security. As a result, web developers may prohibit malicious users from sending duplicate POST requests to masquerade as registered users. Django's security extends beyond its explicit security features: security efforts are bolstered by the vast knowledge and skills of the Django user base.

## 2.4 ORGANISATIONAL TOOLS

### 2.4.1 GITHUB (VERSION CONTROL)

GitHub hosts the source code projects in several programming languages and tracks the modifications made to each revision. As a result, the term "Git" refers to a version control system, which is a mechanism that helps developers to keep track of the continuous changes to their code. The "Hub" is the cluster of participants who have similar interests. It is all about the community's collective effort in updating, refining, and deriving new insights from the submitted code. It's certainly interesting to think about! Some benefits are:

Benefits:

- Documentation
- Markdown
- GitHub Repository
- Track changes (Use of versions)

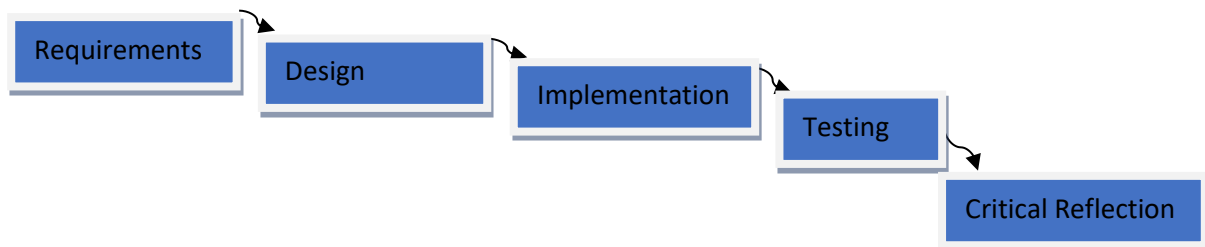
- Integration Options

### 2.4.2 TRELLO (ORGANIZATION TOOL)

Trello full potential is around group management. But it could be also used for Individual project in the same way. Trello can be used for planning, tracking, organizing and, managing, tasks and milestones very efficient and easy. As everything is very easy to create and use your board it can be used as a project calendar and task manager.

## 3. DESIGN/METHODOLOGY

In this segment, we will look at the approaches and procedures used for this project and discuss why they were selected. Providing relevant facts about the methods taken. The use of a waterfall methodology is applied as shown in figure 1.



**Figure 1: Waterfall methodology**

### 3.1 DESIGN

After the above research was concluded. A design pattern is formulated to create a website with the following features. All the above key points associated with user-friendly websites mentioned in subsection 2.1.1 are going to be present. Now for navigation, all options were evaluated and concluded to the use of a Responsive Navigation. This option was selected rather than the rest due to the many user-friendly and modern features that could provide. In general, responsive navigation is a combination of horizontal navigation and hamburger navigation that can adjust based on the resolution of the user screen. This ensures that the website looks fine on all platforms. For example, on mobile devices, the navigation will be transformed to hamburger navigation that is renowned for being fashionable, dynamic, mobile-friendly, and understandable. As a result, users are drawn to this style of the menu because it is both fashionable and practical. Moving now with the colour scheme, the most relevant ones would be Intense green, blue, and red. It fits perfectly with the concept as red and green especially reference a lot in the stock prices up/down of the stock market. And can be used to characterise impactful information's.

Hex numbers of the concluded colour palette:

Bright Green: #beef00

Electric Red: #ff0028

Deep Green: #657a00

Power Blue: #1400c6



### 3.2 PROGRAMMING LANGUAGE

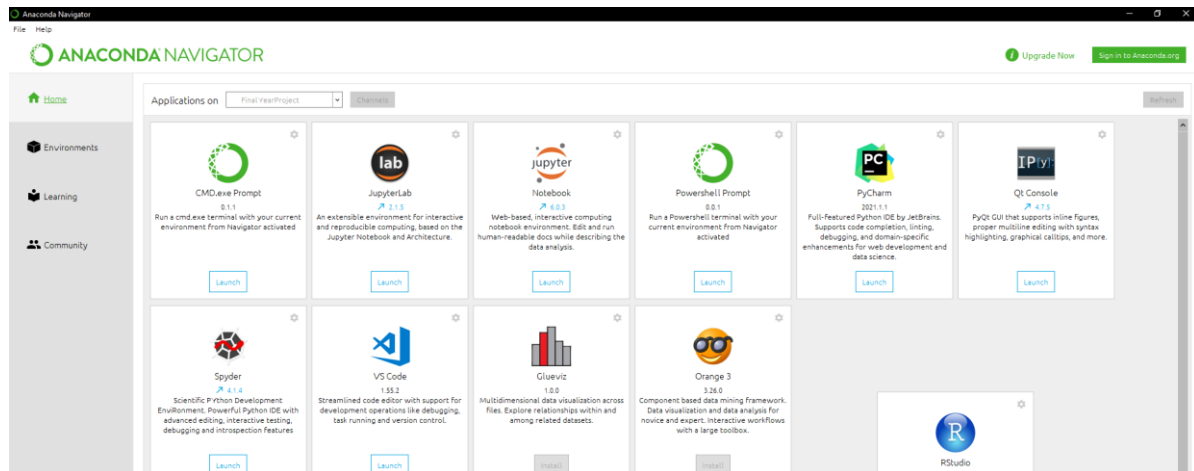
Python is known as the most frequently used language for data science projects. When it comes to Data Science, Python is a handy platform with many advantages. Since it is open-source, it is adaptable and always evolving. Python was the top pick for the development of this project due to the extensive benefits and libraries it can offer and the great compatibility with the anaconda environment through Jupyter Notebook and PyCharm. Some other benefits are the Simplicity of the language, Multi-paradigm approaches and, a massive community. All project functionalities will be developed and tested in a Jupyter Notebook which is a free and open-source web application that enables programmers to write interactive code. Also, Jupyter Notebook is a useful method for Data Science and Machine Learning that will be used a lot in this project. It allows you to present your conclusions and embed the outcomes (visualisations) in the same document that will be very useful for testing and evaluation. All Software/Tools described in research section 2.3 will be used in the development of the project.

### 3.3 PLATFORMS/SOFTWARE

The deliverable will be developed on various platforms. This comes down to the complexity of the project and the need for many technologies.

#### 3.3.1 ANACONDA

Anaconda Enterprise Figure 2 was selected because it provides developers and data scientists with access to cutting-edge technologies and the ability to use their favourite software and packages without compromising protection. Anaconda allows us to build environments in which we can install libraries and packages. This setting is entirely independent of the operating system and administrative libraries. This means we can build user-level environments with customised library versions for individual projects, allowing us to port the project through operating systems with minimal effort. Furthermore, Anaconda is known as one of the most popular Data Science platform that will be especially helpful as the functional aspect of the project is based around Data Science ideas and methods.



**Figure 2: Anaconda Navigator Environment**

### 3.3.2 PYCHARM

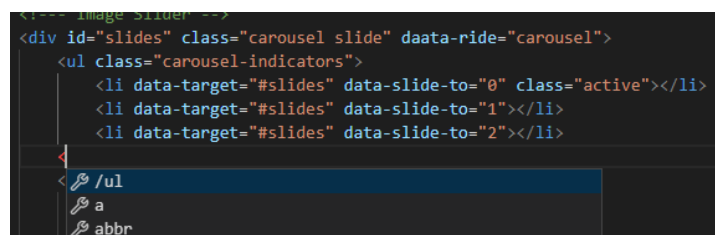
Pycharm was selected for the various python web development frameworks such as the Django templates debugger, manage.py and appcfg.py software, special autocompletion and navigation and, first-rate support for HTML and CSS. Also, PyCharm interfaces with IPython Notebook provides an integrated Python console and is compatible with Anaconda as well as a variety of science programmes such as Matplotlib that will be used for the illustration of the graphs and NumPy that will be used on many occasions. Pycharm will also serve as the binding platform for the final solution of the project.

### 3.3.3 VISUAL STUDIO CODE

Visual Studio was selected for many reasons. The visual studio developer community is massive, there is lots of documentation, there is outstanding user support in the event of a bug and, the GitHub extension available enables users to bind to their GitHub repository and push, pull, or add updates directly from the Visual Studio IDE. Furthermore, there also many benefits related to web development (HTML5/CSS) that will be especially useful in the development of the Website.

Benefits:

- HTML IntelliSense



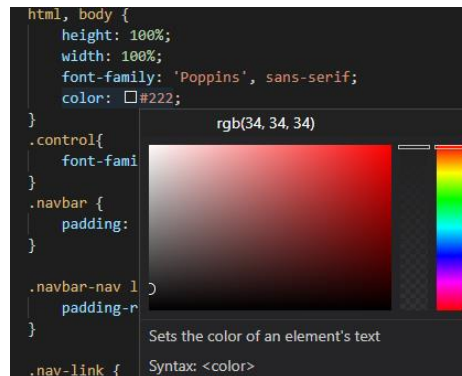
As can be seen, IntelliSense provides suggestions for elements, tags, and many more. Document symbols for HTML are also available, allowing you to easily navigate to DOM nodes by id and class name. Suggestions can be triggered at any time by pressing **Ctrl+Space**.

- Close Tags

```
<div class=AutoCloseTest></div>
```

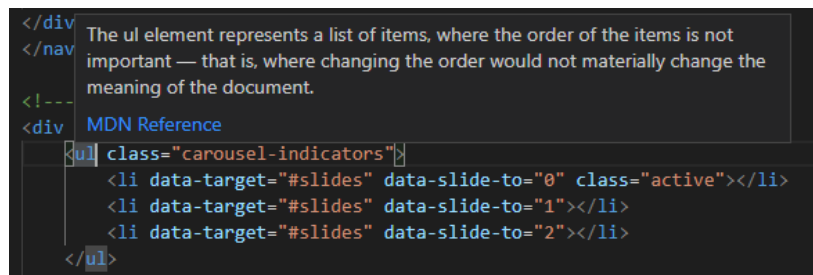
Automatic close tag when the opening tag is closed by ">". When this effect is triggered the matching closing tag will be auto written for the developer.

- Colour Picker



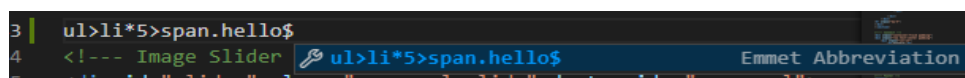
Colour picker UI is provided hue, saturation and opacity configurations to be picked from the editor. This can be triggered by hovering over the colour definition.

- Hover

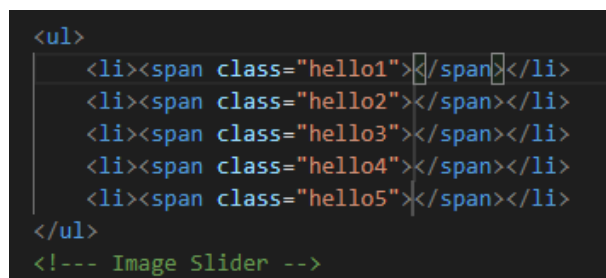


More information can be accessed by moving the mouse over HTML tags, embedded types, and JavaScript.

- Emmet Snippets



Emmet abbreviations are listed along with other suggestions and snippets. For example by typing the above code `ul>li*4>span.hello$` will give back the following outcome.



- Formatting Commands

1. `Shift+Alt+F` Format entire file
2. `Ctrl+K Ctrl+F` Format selection

## 4 DEVELOPMENT

### 4.1 ORGANISATIONAL TOOLS / VERSION CONTROL

First, the tools selected in subsection 2.4 had to be initialized before the production phase could begin.

#### Trello

At the start, a Trello board was created with all the objective that Had been set on project specification. And from their objective kept changing and changing categorise like need to be done and complete. Trello was also used for holding research data and their respectful links. That was retrieved and used to write section 2.

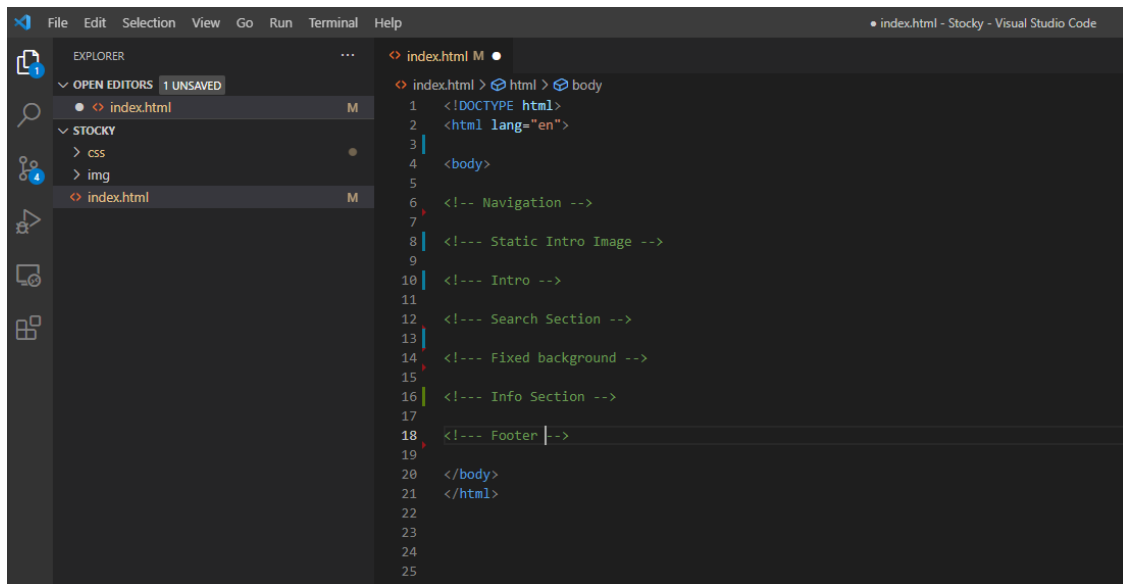
#### GitHub

GitHub was used mostly on the development of the website which was also needed due to public free website hosting that can provide. This was used to publish the website design for user to be able to evaluate it and give feedback. Also, code was always stored and manipulated using GitHub Desktop UI.

### 4.2 WEBSITE

#### 4.2.1 ENVIRONMENT SETUP

After the completion and setup of the GitHub and Trello board, a procedure started to set up the environment suggested in section 3.2.3 (Visual Studio Code) for the development of the website design template. A project was created in Visual Studio Code with the name STOCKY That is representing the website logo. Then 2 folders were created, the first one for the Stylesheet ("Style.css") and, a second folder for all the images that were going to be used in the website. After the structure of the Project solution was initialized, Index.html was created and commented on with sections for better management as shown in figure 3. By creating commented sections also proved to be very beneficial as it helps the project run smoothly and organised throughout the development giving it a more step-by-step feel.



**Figure 3: Environment setup and commented sections**

## 4.2.2 WEBSITE DESIGN DEVELOPMENT

Starting with the development of the website, first, a set of scripts and tags had to be loaded into the head of the website that provided all the necessary tools for the creation of the website shown in figure 4 (reference Appendix section Imports).

Imports and their proposal:

### 1. HTML Encoding

The HTML5 specification allows web developers to use the UTF-8 character set, which contains almost all characters and symbols! A web browser must be aware of the character set used in the page to properly view. To specify this the need of the following code snippet is required:

```
<meta charset="utf-8">
```

### 2. Viewport (Responsive web design)

The viewport is the portion of a web page that is open to the user. The viewport varies depending on the user, which is smaller on a cell phone than on a computer screen. Prior to tablets and smartphones, web pages were only optimised for computer screens, and web pages had a static nature and a fixed scale. When phones became popular, fixed format web pages became too big to match the viewport. To address this, browsers on certain computers reduced the size of the whole web page to suit the screen. The following code snippet instructs the browser on how to manage the page's dimensions and scaling.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

The width=device-width component configures the page's width to match the device's screen width (which will vary depending on the device). The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

### 3. Bootstrap

To be able to access bootstrap functionalities and benefits that were mentioned in section 2.2.3 the following code snippets are been used.

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
```

#### 4. jQuery CDN (Content Delivery Network)

jQuery is a compact, feature-rich JavaScript library. It simplifies HTML document traversal and manipulation, event management, animation, and Ajax with an easy-to-use API that operates through a wide range of browsers.

use of Google host jQuery.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

#### 5. Popper UI tools

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
```

#### 6. Fontawesome ToolKit (for Fonts and Icons)

```
7. <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
```

How the design proposed in section 3.1 was achieved:

Navigation and mobile features were achieved using Media Queries. Basically, by using media queries you can specify a max-width that when it exceed or fall below changes the sizing of the website. This will trigger automatically reducing the size of icons buttons and layers to fit the smaller resolutions. 3 different queries were creating first the larger one with a max-width of 992px that is suitable for Monitors followed by the second one with a max-width of 768px that is focused on tables and last the smallest one with a max-width of 576px for phones. Adding more to these when the Tittles in the navigation cannot feet anymore in a straight line the navigation will automatically also change from a Horizontal Navigation into a burger Navigation. Media queries can be seen in figure 6. Now moving on with the colour scheme was implemented by using the hex numbers proposed inside the style.css where was suitable. The speed test will also be carried on the test and evaluation. When the website was finished a questionnaire was created to receive users' feedback and adapt any changes needed to the design based on their responses. The questionnaire was posted on Reddit shown in figure 5. Reddit was selected due to the massive community and the need for real feedbacks with no feelings involved that may accrue from friends and family. Questions and results can be found in figure 43 (reference Appendix section Questionnaire).

```

300
301      /*---Media Queries ---*/
302      @media (max-width: 992px) {
303          .social a {
304              font-size: 4em;
305              padding: 2rem;
306          }
307      }
308      @media (max-width: 768px) {
309          .carousel-caption {
310              top: 45%;
311          }
312          .carousel-caption h1 {
313              font-size: 350%
314          }
315          .carousel-caption h3 {
316              font-size: 140%;
317              font-weight: 400;
318              padding-bottom: .2rem;
319          }
320          .carousel-caption .btn {
321              font-size: 95%;
322              padding: 8px 14px;
323          }
324          .display-4 {
325              font-size: 200%;
326          }
327          .fas fa-circle fa-xs{
328              font-size: 0.5rem;
329          }
330          .social a {
331              font-size: 2.5em;
332              padding: 1.2rem;
333          }
334      }
335      @media (max-width: 576px) {
336          .carousel-caption {
337              top: 40%;
338          }
339          .carousel-caption h1 {
340              font-size: 250%
341          }
342          .carousel-caption h3 {
343              font-size: 110%
344          }
345          .carousel-caption .btn {
346              font-size: 90%;
347              padding: 4px 8px;
348          }
349          .carousel-indicators {
350              display: none;
351          }
352      }
353

```

Figure 4: Media Queries

## 4.3 DATA ANALYSIS

### 4.3.1 DATA COLLECTION

Data are collected using python. This project accesses all data dynamically when a user post (search) a company using the corresponding ticker. Then the ticker ID is used to crawl data from FinViz and Yahoo finance that allows third party users to access their data by using their API.

#### FinViz data extraction:

FinViz extraction is performed to serve the sentiment analysis as is used to get news dates, times, and titles.

So now we have the website API and the user's ticker ID given. But we are not yet able to extract data. To do so we need to better understand the actual website that we are going to crawl the data and where we can find those data. This will provide the relevant knowledge needed to be able to extract

only the data needed and not irrelevant data. All the news is stored into a table with id = "news-table", in the example below the data are bounded by <tr> </tr>, Date and time are between the first <td> </td> tags and the news headline in the <a> </a> tags bellow.

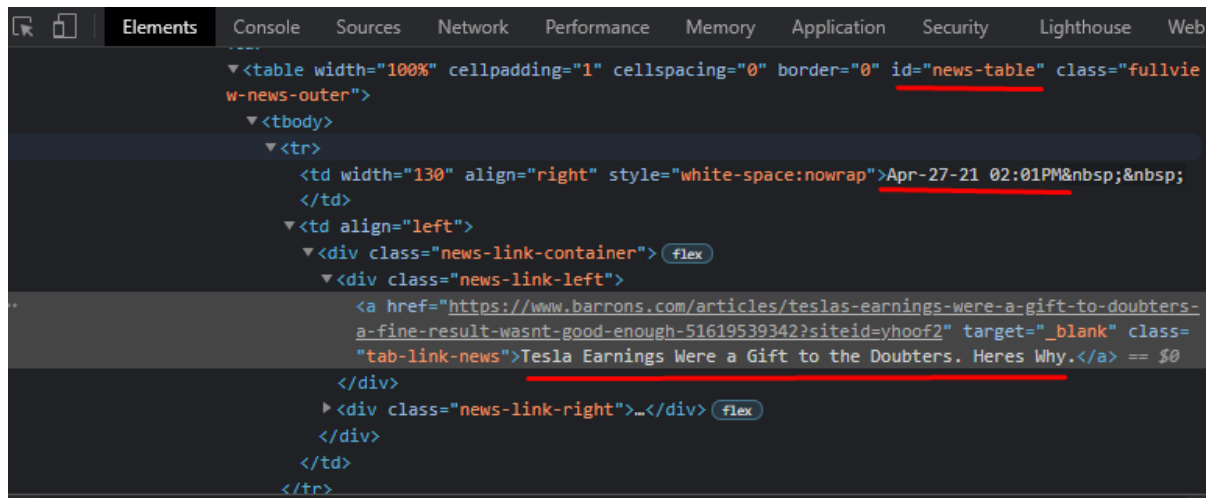


Figure 5: Locating data needed in FinViz Html

Now moving on figure 8 will demonstrate how the ticker ID is used with Finwiz\_url to get the URL content, find the "news-table" of the specific ticker given and, save them into a python dictionary named news\_table.

```
In [5]: news_tables = {}
search = 'TLRY'
tickers = [search]

for ticker in tickers:
    url = finwiz_url + ticker
    req = Request(url=url,headers={'user-agent': 'my-app/0.0.1'})
    response = urlopen(req)

    #Read the contents of the file into 'html'
    html = BeautifulSoup(response)

    #Find 'news-table' in the Soup and Load it into 'news_table'
    news_table = html.find(id='news-table')

    #Add the table to our dictionary
    news_tables[ticker] = news_table
```

Figure 6: Code snippet of data extraction

#### Yahoo Finance data extraction:

The purpose of the second data extraction is to get the close history values of the given ticker company. Those data are going to be used in various ways like company price visualization graph, price fluctuations, and future price predictions.

To get the close prices first a date frame has to be specified. Then by using the ticker, starting, and ending dates specified a data extraction can be performed as shown in figure 9 to get the data out of the yahoo finance API.



```
In [70]: #get the stock quote
ticker = 'BABA'
start = dt.datetime(2012,1,1)
end= dt.datetime.now()

df = web.DataReader(ticker, 'yahoo', start,end)
```

Figure 7: Second data extraction using Yahoo finance.

To keep things consistent the starting date is initialized to 01/01/2012 and the ending date is initialized to the current date. The selection of the starting date was my personal opinion for the most suitable choice.

#### 4.3.2 SENTIMENT ANALYSIS

After the data are extracted and stored time and headline will be parsed into a python list called `parsed_news` using iteration. An iteration is performed through each `<tr>` `</tr>` tags to get the headlines between the `<a>` `</a>` tags and the date and time between the `<td>` `</td>` tags before parsing them also an if, else loop will find each date label and group them with the rest news under the same date, this is required as the rest of the news with the same date only have a time label. The process is shown in figure 9.

```
: parsed_news = []

# Iterate through the news
for file_name, news_table in news_tables.items():
    # Iterate through all tr tags in 'news_table'
    for x in news_table.findAll('tr'):
        # read the text from each tr tag into text
        # get text from a only
        text = x.a.get_text()
        # splite text in the td tag into a List
        date_scrape = x.td.text.split()
        # if the length of 'date_scrape' is 1, Load 'time' as the only element

        if len(date_scrape) == 1:
            time = date_scrape[0]

        # else Load 'date' as the 1st element and 'time' as the second
        else:
            date = date_scrape[0]
            time = date_scrape[1]
        # Extract the ticker from the file name, get the string up to the 1st '_'
        ticker = file_name.split('_')[0]

        # Append ticker, date, time and headline as a list to the 'parsed_news' List
        parsed_news.append([ticker, date, time, text])
```

Figure 8: Code snippet for extracting and storing time, headline.

So far the data are crawled, grouped, and stored into a list. Now it's time to move into actually analysing the data by performing sentiment analysis. The sentiment scores are listed in the 'compound' column. The higher the value for positive scores, the more positive the sentiment.

Negative scores work in the same way: the lower the value, the more negative the sentiment. The scores range from -1 to 1. To achieve this first the list needs to be converted into a Panda DataFrame called “parsed\_and\_scored\_news”, then an iteration is performed through the headlines to get the sentiment scores by applying Vader polarity mention in subsection 2.3.3. After the scores are calculated a conversion is performed to change the scores from the python data structure to a Panda DataFrame and also convert the date column from string to also Panda DataFrame.

```
In [8]: # Instantiate the sentiment intensity analyzer
vader = SentimentIntensityAnalyzer()

# Set column names
columns = ['ticker', 'date', 'time', 'headline']

# Convert the parsed_news List into a DataFrame called 'parsed_and_scored_news'
parsed_and_scored_news = pd.DataFrame(parsed_news, columns=columns)

# Iterate through the headlines and get the polarity scores using vader
scores = parsed_and_scored_news['headline'].apply(vader.polarity_scores).tolist()

# Convert the 'scores' List of dicts into a DataFrame
scores_df = pd.DataFrame(scores)

# Join the DataFrames of the news and the List of dicts
parsed_and_scored_news = parsed_and_scored_news.join(scores_df, rsuffix='_right')

# Convert the date column from string to datetime
parsed_and_scored_news['date'] = pd.to_datetime(parsed_and_scored_news.date).dt.date
```

**Figure 9: Code snippet for storing and performing sentiment analysis.**

This concludes the steps needed to calculate the sentiment score and store it. Now moving into presenting the data. To visualize the score first data need to be group by date and ticker columns so it can be used to calculate the mean score. Then outcome will be plot into a bar chart using Pandas.

```
In [9]: plt.rcParams['figure.figsize'] = [10, 6]

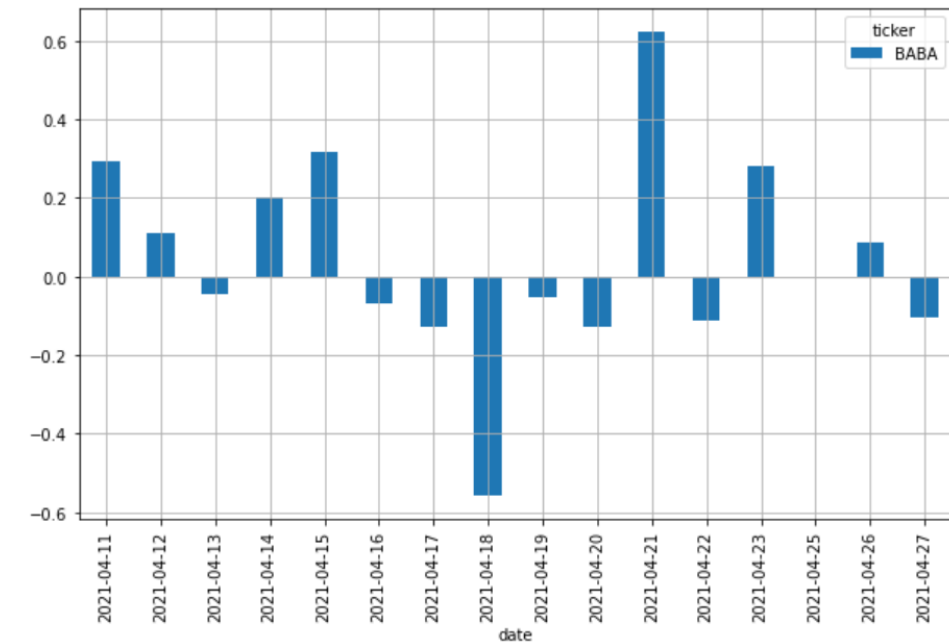
#Group by date and ticker columns from scored_news and calculate the mean
mean_scores = parsed_and_scored_news.groupby(['ticker', 'date']).mean()

#Unstack the column ticker
mean_scores = mean_scores.unstack()

#get the cross-section of compound in the 'columns' axis
mean_scores = mean_scores.xs('compound', axis='columns').transpose()
|
fig = mean_scores.plot(kind='bar')
plt.grid()
```

**Figure 10: Code snippet for visualizing the score.**

By running the above code shown in figure 12 the following bar chart is visualized showing the average sentiment score of each day from -1 to 1.



**Figure 11: Visualization of average sentiment score**

When the bar chart was ready a second progressive bar was created by calculating the average score of all news and adding 1 to the value. So now the lowest possible outcome will be 0 representing a negative score and 2 the highest representing a happy score. Then the result will pass throw and if else statement to return a string output with which is going to be positive, negative, or natural as shown in figure 14. These values are going to be used later in the development to visualize the progressive bar into the website using emojis.

```
In [11]: mean_scores = parsed_and_scored_news.groupby(['ticker', 'date']).mean()

In [12]: score = mean_scores.compound.mean()+1

In [13]: score
Out[13]: 1.0448496781994048

In [14]: if(score >= 1.3):
          outcome =("positive")
        elif (score<= 0.8):
          outcome =("negative")
        else:
          outcome =("natural")

In [15]: outcome
Out[15]: 'natural'
```

**Figure 12: Code snippet for progressive bar output**

### 4.3.3 ANALYSIS PANEL

The analysis panel is where price fluctuation is presented. Three if else statements are used to calculate the price fluctuations by performing some maths shown in figure 15. The logic behind it is simple. For example, if today's close price is 100£ and the last days were 110% the values are going to pass to the negative if statement as the value decreased and then by dividing the current value and the previous day's value followed by a multiplication of 100 will return the deference percentage between the two values which in this case will be 10% change. Then since the price was smaller also

a colour output will be initialised as red and info output will be initialised to decrease those values will be used in the information panel to represent the outcome followed by the info message in the respected colour. The same will apply also to the future prediction by doing the same thing but with the future prices.

```
In [108]: day = df1.tail(2).values.tolist()
firstday = np.array( day[1], dtype=float)
lastday = np.array( day[0], dtype=float)

if day[0] < day[1]:
    # positive
    value = firstday - lastday
    value = (value / lastday) * 100
    value = "%.2f" % value
    color = 'green'
    info = "increased"
elif day[0] > day[1]:
    # negative
    value = lastday - firstday
    value = (value / lastday) * 100
    value = "%.2f" % value
    color = 'red'
    info = "decreased"

week = df1.tail(5).values.tolist()
firstday = np.array( week[4], dtype=float)
lastday = np.array( week[0], dtype=float)

if week[0] < week[4]:
    # positive
    value1 = firstday - lastday
    value1 = (value1 / lastday) * 100
    value1 = "%.2f" % value1
    color1 = 'green'
    info1 = "increased"
elif week[0] > week[4]:
    # negative
    value1 = lastday - firstday
    value1 = (value1 / lastday) * 100
    value1 = "%.2f" % value1
    color1 = 'red'
    info1 = "decreased"

# Month price analysis
Month = df1.tail(22).values.tolist()
firstday = np.array( Month[21], dtype=float)
lastday = np.array( Month[0], dtype=float)

MonthIncrease = 0
if Month[0] < Month[21]:
    #positive
    value2 = firstday - lastday
    value2 = (value2 / lastday) * 100
    value2 = "%.2f" % value2
    color2 = 'green'
    info2 = 'increased'
elif Month[0] > Month[21]:
    # negative
    value2 = lastday - firstday
    value2 = (value2 / lastday) * 100
    value2 = "%.2f" % value2
    color2 = 'red'
    info2 = 'decreased'
```

**Figure 13: Code snippet calculating price fluctuations.**

#### 4.3.4 ML FUTURE ANALYSIS (PREDICTION)

The next step in development is to predict the future using neural network models for time series prediction in Python using the Keras deep learning library. To do so first the close price data need to be isolated and insert into a data frame using the data extraction process described in subsection 4.3.3. Then the close price dataset will be split into two datasets by calculating the index of the split point and separating the data into the training dataset with 65% of the observations that will be used to train the model and leave the remaining 35% for testing the model as shown in figure 16.

```
In [13]: ##splitting dataset into train and test split
training_size=int(len(df1)*0.65)
test_size=len(df1)-training_size
train_data,test_data=df1[0:training_size:],df1[training_size:len(df1),:1]
```

**Figure 14: Code snippet for splitting the dataset into train and test split**

Two arguments will be defined for a function. The dataset is a NumPy array that must be converted into a dataset, and the time step is the number of previous time steps to use as input variables to predict the next time period, which is set to 1 in this case shown in figure 17.

This default creates a dataset with X representing the number of values at a given time (t) and Y representing the number of values at the next time (t+1). The function will now be used to ready the train and test datasets for modelling.

```
In [16]: # convert an array of values into a dataset matrix
def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0] ###i=0, 0,1,2,3-----99 100
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return numpy.array(dataX), numpy.array(dataY)
```

**Figure 15:Code snippet converting an array of values into a dataset matrix.**

Then code from figure 18 will be used to reshape the data into our x\_train, y\_train, X\_test, test.

```
In [17]: # reshape into X=t,t+1,t+2,t+3 and Y=t+4
time_step = 100
X_train, y_train = create_dataset(train_data, time_step)
X_test, ytest = create_dataset(test_data, time_step)
```

**Figure 16:Code snippet creating training sets.**

An extra reshape will be performed to also add an extra dimension needed for LSTM “X\_train.shape[1] , 1” and “X\_test.shape[1] , 1” shown in figure 19.

```
In [20]: # reshape input to be [samples, time steps, features] which is required for LSTM
X_train = X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
X_test = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)
```

**Figure 17:Code snippet performing reshape input to be [samples, time steps, features]**

A stack LSTM (Long Short Term Memory) model will be created. The model will be Sequential followed by 3 more LSTM layers and one Dense layer which is the output layer. The model will be compiled using “mean\_squared\_error” which if we take the square root gives us an error score in the units of the dataset and for optimizer “adam” which is a combination of AdaGrad and RMSProp algorithms that can handle sparse gradients on noisy problems.

```
In [22]: model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

**Figure 18:Code snippet creation of LSTM model**

Moving on now the data are going to fit inside the model created and trained using 25 epochs, batch\_size equal to 64, and verbose equal to 1.

```
In [23]: model.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=25,batch_size=64,verbose=1)
```

**Figure 19:Code snippet fitting the model to be trained.**

Now predict X\_train and X\_test data are going to be used for the prediction and performance metrics. To achieve this the data will be stored into train\_predict and test\_predict to be inverse transform (unscaled) back to their original and pasted to the RMSE performance metrics. RMSE is used to calculate the difference between the target value and the value predicted model value by calculating the square root of the averaged squared between them.

With the above model training and data manipulation concluded and tested. It's time to move into the actual future prediction. First, the latest 100 data will be extracted to be used for the calculation of the next day into the future. Then the data will be reshaped between -1 and 1 and transform into a list.

```
In [46]: test = len(test_data)-100
x_input=test_data[test:].reshape(1,-1)
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
```

**Figure 20:Code snippet of data manipulation to get ready for future prediction.**

Then the data are reshaped and passed into the model to perform a prediction, that will return yhat value to be stored into the final output list and also adding it into the previous input. By doing this the previous list will increase to 101 and be shifted forward because we only need the 100 latest values. To explain this further when a shift is performed from the 101 values are going to be kept only the 99 actual prices plus the 1 predicted into the future and this process will repeat until we have 70 latest values and the 30 future predicted values into the list.

```

In [96]: lst_output=[]
n_steps=100
i=0
while(i<30):

    if(len(temp_input)>100):
        x_input=np.array(temp_input[1:])
        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1))
        yhat = model.predict(x_input, verbose=0)
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]
        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps,1))
        yhat = model.predict(x_input, verbose=0)
        temp_input.extend(yhat[0].tolist())
        lst_output.extend(yhat.tolist())
        i=i+1

```

Figure 21:Code snippet of predicting 30 days into the future.

## 4.4 PYCHARM

### 4.4.1 ENVIRONMENT SETUP

Moving On with the merge process of the website and the python data analysis algorithms a Django project is created using PyCharm as shown in figure 24.

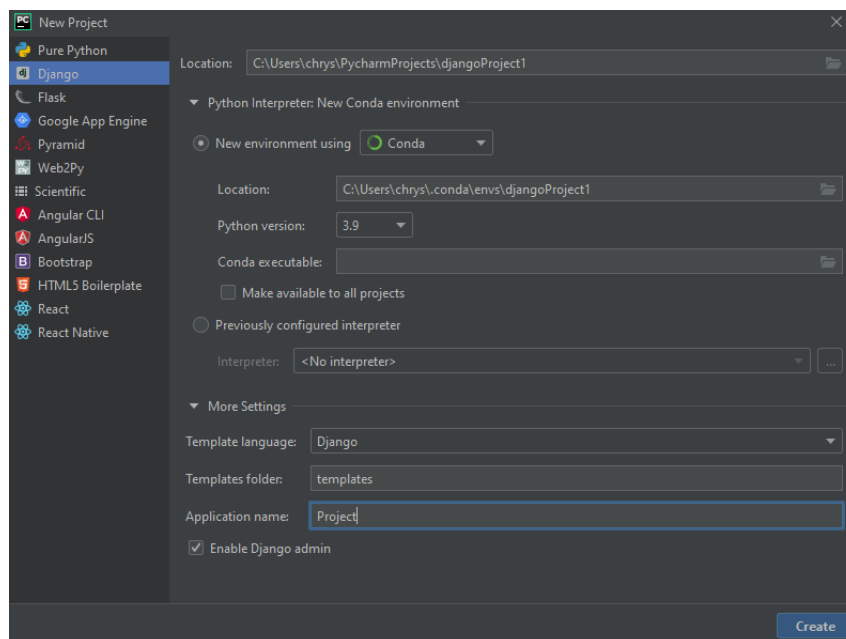
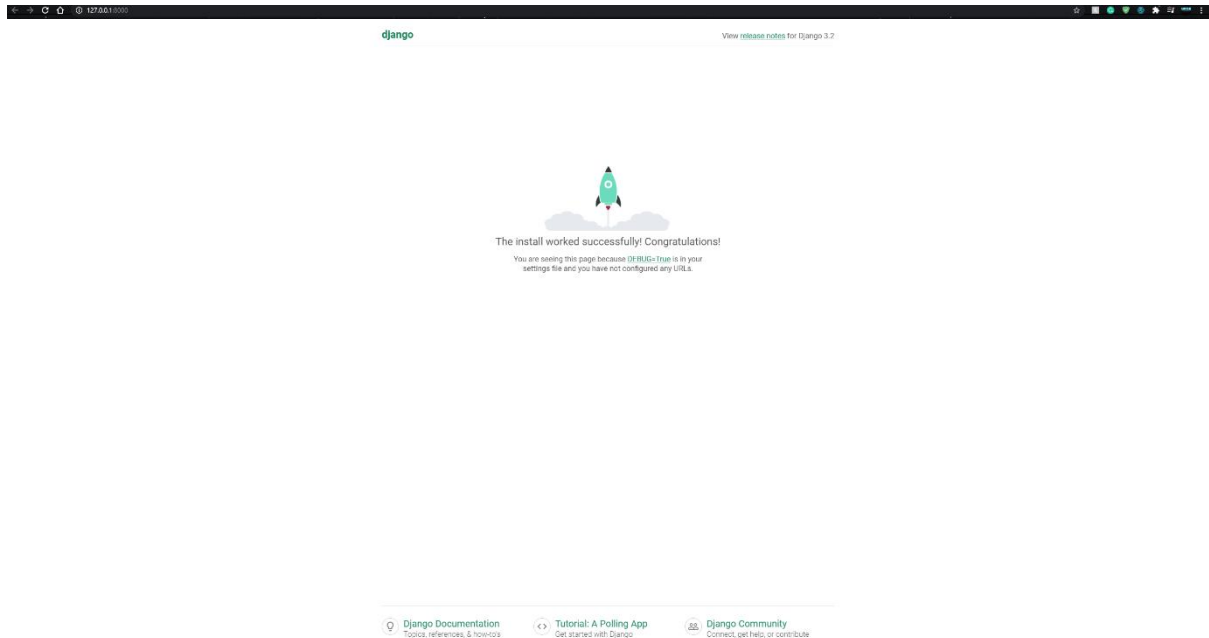


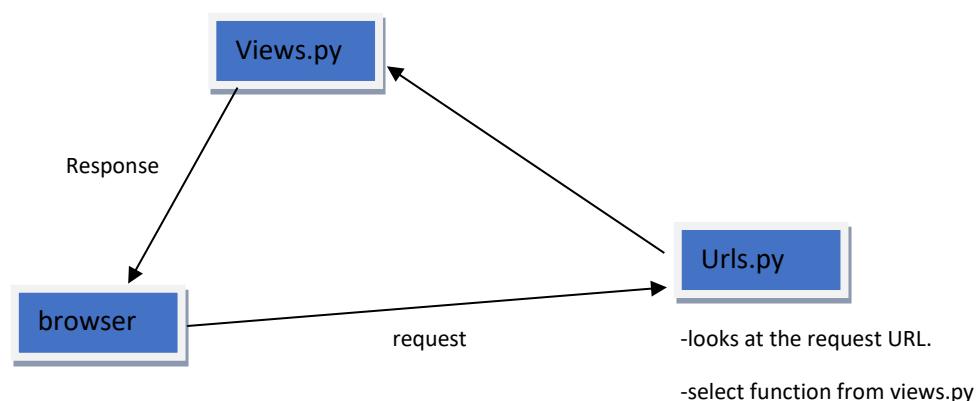
Figure 22: PyCharm project creation.

The following are created automatically when a PyCharm Django project is created. A directory structure, pip installed Django and its dependencies into the new virtual environments site packages, wired up the project for Django support, including the path to setting.py, and created a Django server run configuration with fields supporting Django specific server settings. By running the server a tool window can be accessed in the terminal by clicking the URL resulting in the website shown in figure 25.



**Figure 23: Terminal URL provided by Django.**

Now a few extra steps are needed to be done manually. First, inside the Website folder, a new file is created called views.py. Views are going to be linked with urls.py to provide the functionalities needed to link the website with the python data analysis algorithms as described in figure 26.

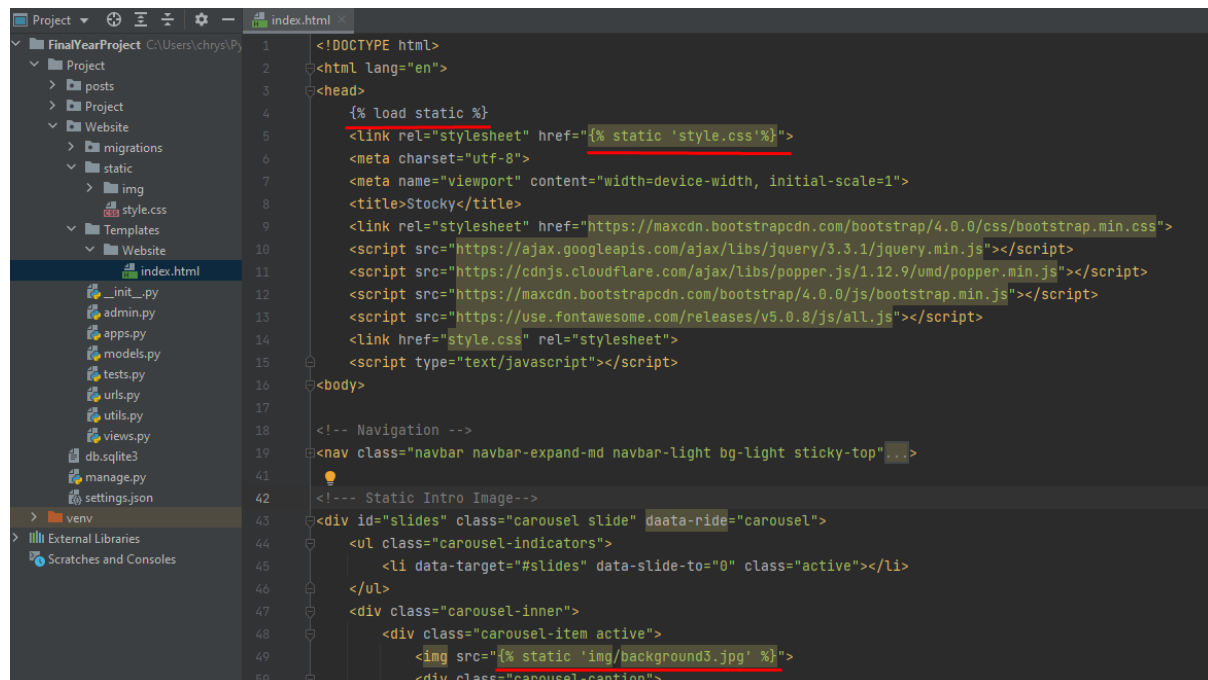


**Figure 24: From the website a request is passed to urls.py that is deciding which function to execute from views.py and then the results are responded back to the browser to the user.**



## 4.4.2 WEBSITE DESIGN MIGRATIONS

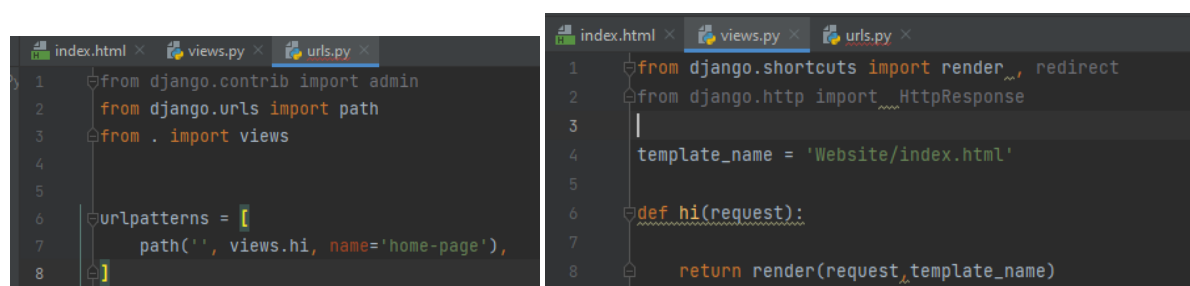
Now that all the files are ready it is time to start the migrations. As shown in figure 23 inside the Website folder there is a static folder in which all images and style.css are going to be transferred from a visual studio project, the same applies for the HTML which is going to be transferred inside the second subfolder called Website. But some changes must be made to allow the website to be loaded properly. Inside the index.html some Django structure code is included to allow the HTML to load and locate the files inside the static folder. Those changes are shown in figure 27.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     {% load static %}
5     <link rel="stylesheet" href="{% static 'style.css' %}">
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8     <title>Stocky</title>
9     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
10    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
11    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
12    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
13    <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
14    <link href="style.css" rel="stylesheet">
15    <script type="text/javascript"></script>
16 </head>
17 <body>
18 <!-- Navigation -->
19 <nav class="navbar navbar-expand-md navbar-light bg-light sticky-top">
41
42 <!-- Static Intro Image-->
43 <div id="slides" class="carousel slide" data-ride="carousel">
44     <ul class="carousel-indicators">
45         <li data-target="#slides" data-slide-to="0" class="active"></li>
46     </ul>
47     <div class="carousel-inner">
48         <div class="carousel-item active">
49             
50         </div>
51     </div>
52 </div>
```

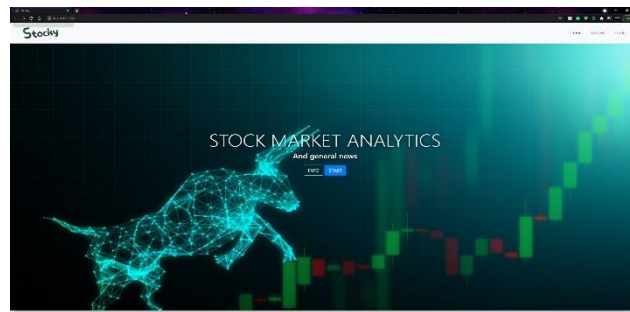
**Figure 25:**Code snippet showing how to use Django brackets "{% %}". Inside the brackets, all static images and files are loaded.

After all necessary changes made index.html was linked with views.py and urls.py to be able to render into the web using Django localhost as shown in figure 25. Resulting to figure 28.



```
index.html x views.py x urls.py x
1 from django.contrib import admin
2 from django.urls import path
3 from . import views
4
5 urlpatterns = [
6     path('', views.hi, name='home-page'),
7 ]
8

index.html x views.py x urls.py x
1 from django.shortcuts import render, redirect
2 from django.http import HttpResponse
3
4 template_name = 'Website/index.html'
5
6 def hi(request):
7
8     return render(request, template_name)
```



**Figure 26: Local server now load up our website instead of the previous welcome page generated by Django**

#### 4.4.3 DATA ANALYSIS MIGRATIONS

Data Analysis algorithms migrations designed in Jupiter notebook proved to be harder than expected. Code was migrated inside views.py explain in figure 29 (reference Appendix section Imports).

Function1 had all the python algorithms responsible for the Analysis panel and sentiment score analysis followed up with function 2 with the algorithms responsible for the price predictions and future analysis panel. The following explanation on how data are manipulated applies for both functions1 and function2 so I will explain it once.

What made this process harder was the process of getting the user input into the functions and then how the results would return and render back to the website.

To achieve this a post request had to be created inside index.html using forms. Then the post will be getting picked up from the input container (SearchBox) and pasted to the function specified in the form to be used as Input data. The input will be a company ticker ID that is going to be used by our algorithms to calculate the results. This process is shown in figure 30.

```
<!-- Search bar -->
<div class="d-flex justify-content-center h-50">
  <form action="/function1" method="post">
    {% csrf_token %}
    <div class="searchbar">
      <input class="search_input" type="text" name="data" placeholder="Search...">
      <button class="search_icon button3" data-toggle="collapse">Search</button>
      <a href="#test" class="search_icon" data-toggle="collapse"><i class="fa fa-chevron-circle-down"></i></a>
    </div>
  </form>
</div>
```

**Figure 27:Code snippet showing the creation of the form which calls an action to be made from views.py and post the user input.**

With the completion of this step, the only problem left was to find a way to return the data to index.html to be rendered into the browser. It took a lot of trial and error but in the end, a way was found to achieve this. The results calculated could be passed back to the HTML using Django render. Basically what this means when a function is called is like rendering a new page. On that page, you can return with the actual render request data values to be used inside the actual HTML those data will be grouped under a content container as shown in figure 31.

```
content= {'last':last, 'general':"General Prediction Range", 'pred':pred, 'predRange':predRange, 'color3P':color3P,
return render(request, 'Website/index.html', content)
```

Figure 28:Data values return to the requested HTML page.

Then those data can be used simply by using their names inside {{ }} brackets as shown in figure 32.

```
<div class="row">
<div class="column">
<br>
<h5>{{lastMsg}}</h5> <h4>{{last}}</h4> <h5>{{nextdayMsg}}</h5> <h4 style="color:{{colorP}}">{{nextday}} {{infoP}} {{valueP}}</h4>
<br>
<h5>{{scoreMsg}}</h5> <h4>{{avg}}</h4> <h5>{{weekMsg}}</h5> <h4 style="color:{{colorP}}">{{week}} {{info1P}} {{value1P}}</h4>
<br>
<h5>{{outcomeMsg}}</h5> <h4 style="color:{{color3}}">{{outcome}}</h4> <h5>{{last}}</h5>
```

Figure 29:Use of data produced in views.py inside the index.html

This method was able to pass data values back to HTML but wasn't able to pass the graphs produced using Matlab into the HTML to be render. To address this a new file called utils.py was created. Utils.py was used to pass graph values from views.py to be transformed into graphs and then passed back to views.py to be returned with the rest data values to the inde.html. this can be seen in figure 33. The same applies to all graphs created.

```
chart2 = get_plot(df.Close)
```

```
index.html x utils.py x views.py x urls.py x
1 import matplotlib.pyplot as plt
2 import base64
3 from io import BytesIO
4
5 def get_graph():
6     buffer = BytesIO()
7     plt.savefig(buffer, format='png')
8     buffer.seek(0)
9     image_png = buffer.getvalue()
10    graph = base64.b64encode(image_png)
11    graph = graph.decode('utf-8')
12    buffer.close()
13    return graph
14
15 def get_plot(scores):
16    plt.switch_backend('AGG')
17    plt.figure(figsize=(6, 3))
18    plt.title('close price history')
19    plt.plot(scores)
20    plt.xlabel('Date', fontsize=9)
21    plt.ylabel('Close Price USD ($)', fontsize=9)
22    plt.grid()
23    plt.tight_layout()
24    graph = get_graph()
25    return graph
26
```

Figure 30:get\_plot is called from views.py sending the data frame close values into get\_plot function in utils.py and the graph is returned to views.py

Furthermore, graphs can not be displayed by simply using them inside {{ }} brackets as the rest data values return to index.html. The graphs need to be printed using the following method figure 34.

```

<div id="test" class="collapse">
  <div class="container h-50">
    <h2 class="d-flex justify-content-center h-50">Analysis complete for {{data}}</h2>
    <div class="carousel-inner">
      {% block content1 %}
        {% if chart2 %}
          
        {% endif %}
        {% if chart3 %}
          
      {% if chart1 %}
        
    {% endif %}
  </div>
</div>

```

Figure 31:How graphs are structured inside the index.html

## 5. TESTING & EVALUATION

### 5.1 WEBSITE

The website was mostly been evaluated throw the questionnaire feedback. The questionnaire had 37 anonymous responses with a base average score of 3.84 out of 5 for the scale questions from very bad to very good, etc. This proved that the website design propose and implemented is what users expected for a user-friendly and modern website. Some bad feedbacks also was taken into consideration and changes were adapted to the website design those changes was. First, the search bar input was expanding when the user mouse was hover over it, some suggestions asked for it to be fixed, A contact button on navigation was also proposed and implemented to automatically scroll down to the section with the contact details, adding more information. Prove of the changes can be found in the appendix, with the link of the working questionnaire and the link of the hosted design of the website using Github pages. Furthermore, a speed test was performed using google chrome developer tools to see if the website speed is good enough (bellow 2 sec). The website was loaded below half of a second as shown in figure 35.

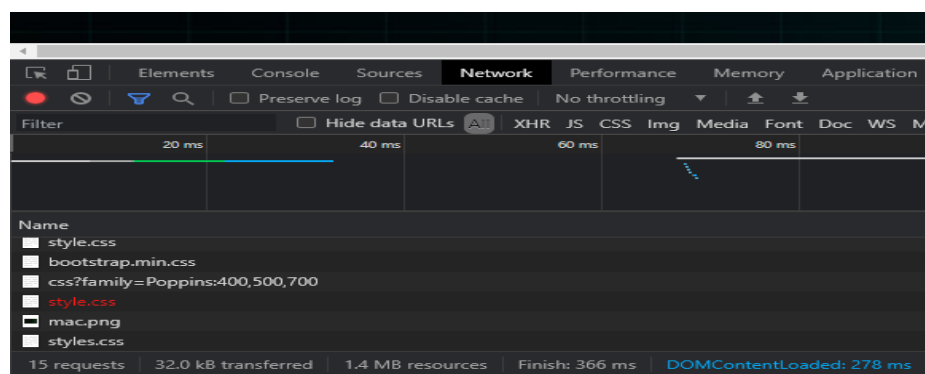


Figure 32: Website Load time

Searches were performed in under 2 seconds which is acceptable. Results figure 36.

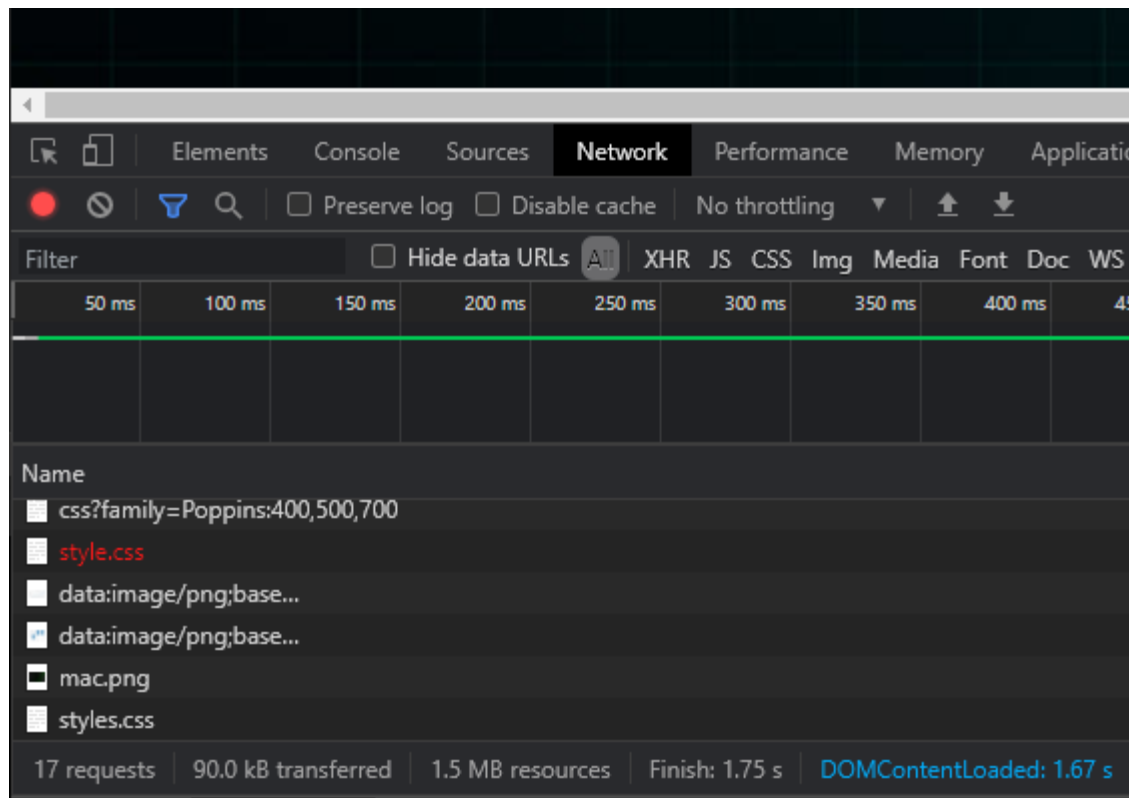


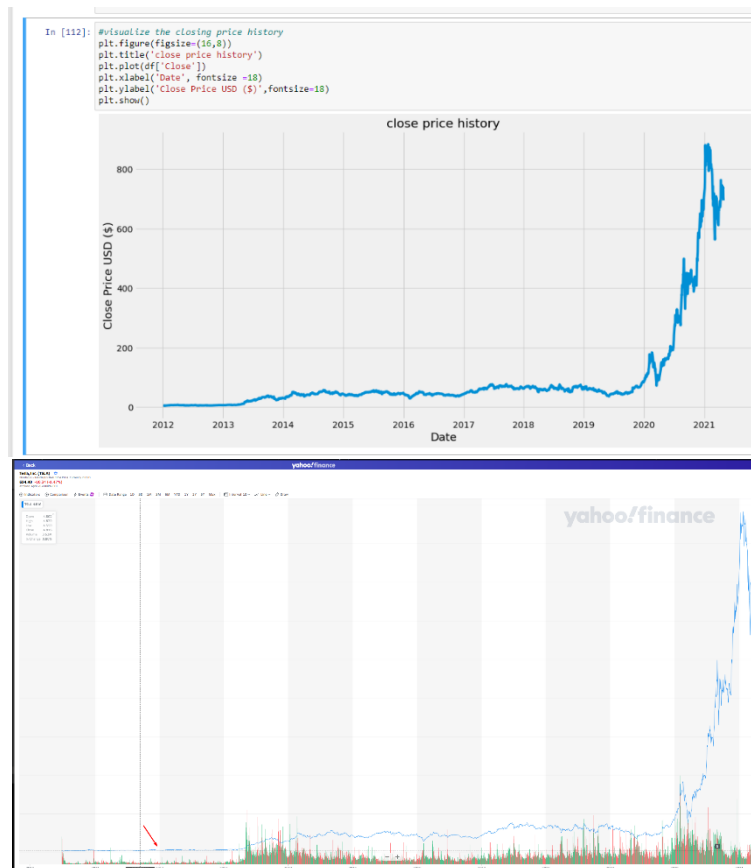
Figure 33: Website search load time

A speed test was not performed in the future prediction as the user was informed from the information section that a future prediction requires time to be performed since data are dynamic and are not stored so each time a new future analysis is performed the data needs to be trained again.

## 5.2 ALGORITHMS EVALUATION.

algorithms are been evaluated and tested using trial and error methods. All steps described in the development was evaluated parallel with the development, after each step data was printed to Jupiter Environment to observe their accuracy.

Let start with the close price graphs. To evaluate the accuracy of the close price history graph it was compared with the yahoo fiance graph of the same company. Both graphs were identical. Proving the accuracy of the graph created. Prove figure 37.



**Figure 34:comparison between project graph output and yahoo finance graph**

For the sentiment analysis algorithm data was evaluated by printing the parsed data and confirming that the data collected and parsed are. The ticker ID, date/time, and title. Prove figure 38.

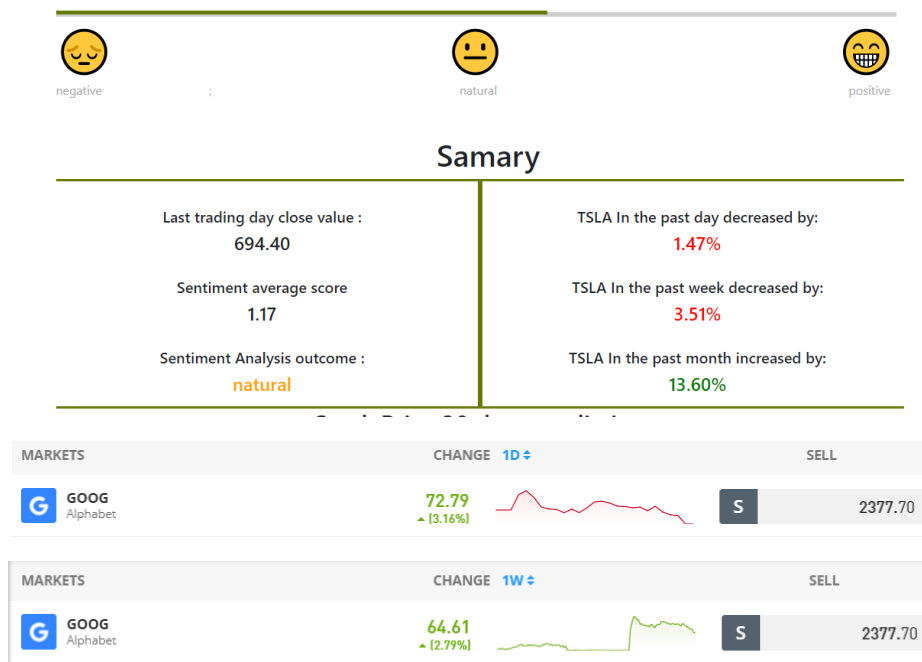
```

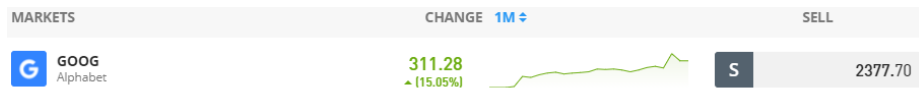
parsed_news
02:00AM',
'Xis Next Target in Tech Crackdown Is Chinas Vast Reams of Data'],
['BABA',
'Apr-22-21',
'08:00PM',
'Jack Ma/Ant Group: a disrupter who is no longer good for business'],
['BABA',
'Apr-22-21',
'08:00PM',
'Chinas central bank fights Jack Mas Ant Group over control of data'],
['BABA',
'Apr-21-21',
'06:27AM',
'Greenpeace Chides Alibaba on Energy as China Tech Cleans Up'],
['BABA',
'Apr-21-21',
'04:52AM',
'Tencent tops Greenpeace clean energy rankings for China Big Tech'],
['BABA',
'Apr-21-21',

```

**Figure 35:** Parsed\_news output proving that the data extraction was performed correctly.

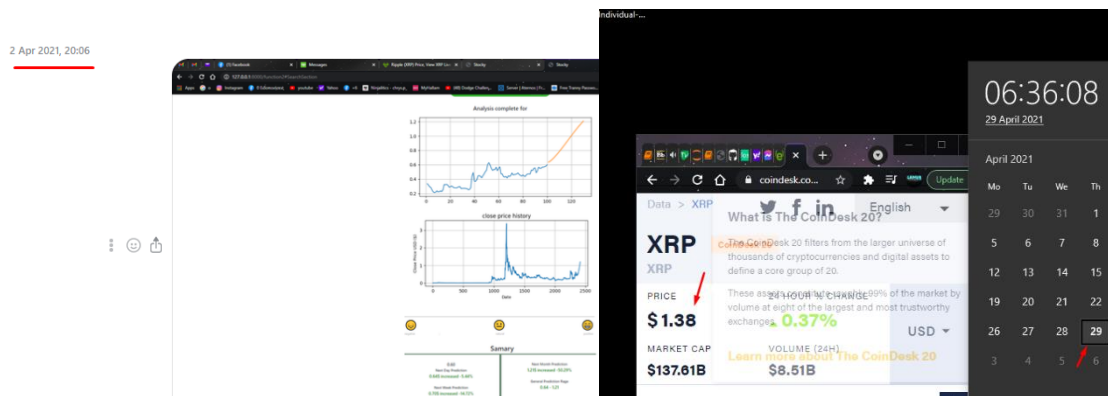
For the analysis panel last trading day close value, past day fluctuation, week fluctuation, and month function was also compared with the Etoro financial trading website to evaluate the results of the project. The price hear are not always identified as each fluctuation analysis may differ slightly but are all performed with the same logic. Prove figure 39.



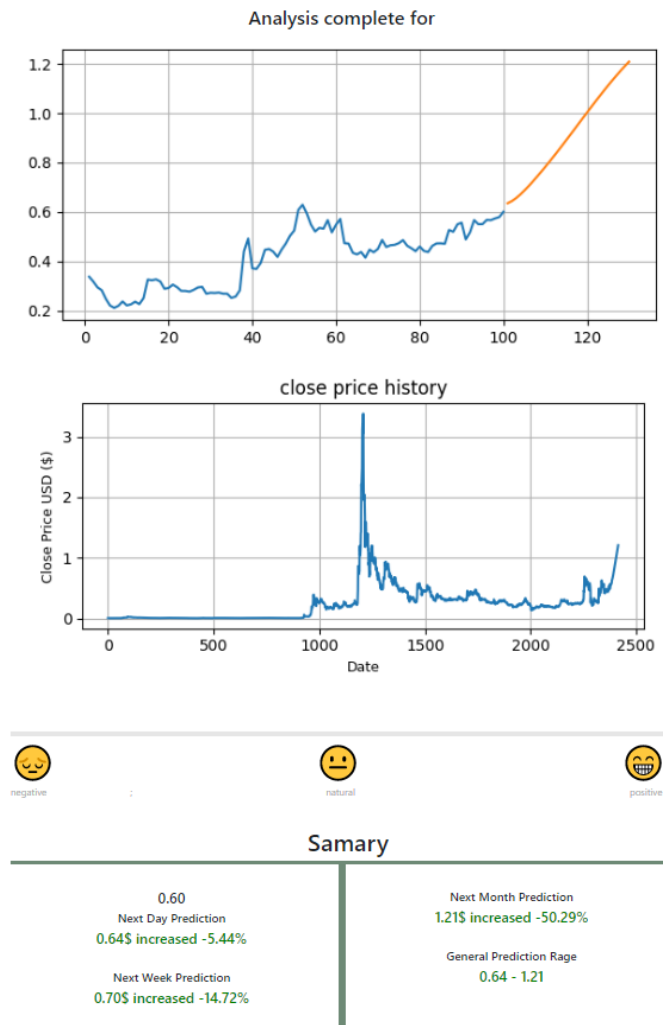


**Figure 36: Comparison of project fluctuations and Etoro fluctuations.**

Lastly to evaluate the future prediction a prediction was made one month ago and saved to be evaluated after a month is passed. The results were mind-blowing with the prediction predicting correctly that the price is going to increase but also the exact predictions fluctuations were very close to values. But this could not be counted as proof of working as one test is not enough to prove the accuracy of the future prediction. A lot of time is required for this aspect of the project to be proved accurate enough and trusted. Prove figure 40.

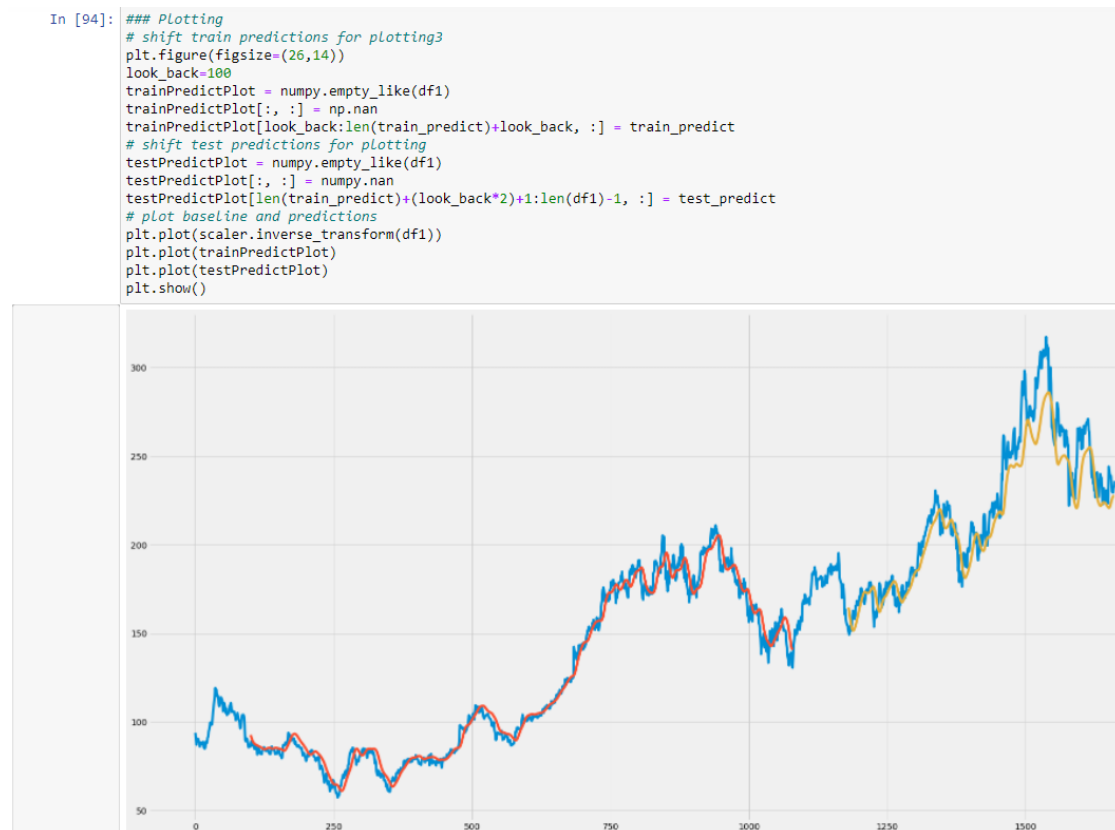






**Figure 37: Comparison between prediction made one month into the future with the future price after one month.**

Futhermore Future predictions was also evaluated by making predictions for past values and evaluating them with the accual values. This proved that the model is also capable of predicting very accurately the prices but still the prices are not taking into consideration news, and unforeseen circumstances. Prove figure 41.



**Figure 38:**Blue is the visualization of the original values and red and yellow are the predicted prices.

## 6. CRITICAL REFLECTION

In this section, a short discussion will summarise the results, outcomes, problems/limitations, and feature implementations.

### 6.1 RESULTS/CONCLUSION

The final project proved that data can be extracted dynamically by a simple search input in less than two seconds and produce some analytics for the user to evaluate. This was also the main objective of this project. Also, a user-friendly website was created for the algorithms to be hosted and accessed to a more simple environment for everyday people. All the functionalities could be run individually in python environments and get the same results but the main idea behind the project was to find a way to provide all those functionalities to everyone especially those with no programming skills and knowledge.

### 6.2 PROBLEMS/LIMITATIONS

The project deliverables were achieved but some problems could not be resolved. Problems with the Django environment especially was very hard to tangle. For example, if the user gives a non-existing ticker ID the website will return an error message and the page must be reloaded again. Also, I couldn't find a way to make Django load a page in a specific section and the progress, this was a problem

because after a search the page was reloaded and the user must scroll down and open the analysis panel manually which is not the ideal approach. One more limitation would be the need to type again the company ticker ID for the future prediction to be performed as a new post request must be performed to call the second function and pass the ticker ID again. If the training process was faster then all the analysis could be calculated simultaneously and presented to the user but this could not be achieved. Also, data are not saved so if a user refreshes the page while is view the future prediction the page will start training again resulting in the user waiting again for the training to be complete. Their also some HTML errors like the visualization progressive bar for the sentiment score is present all the time and could not find a way to dynamically show it and remove it due to Django rendering the page each time action is performed it could not be deleted or removed.

### *6.3 PROPOSE SOLUTIONS*

Starting with the Django environment problems a debug process could be included to refresh the page when a non-existing input is inserted. Also, a new page could be created with the results of the analysis that could be rendered to eliminate the problem of sections areas. But for future predictions, it could be proved very hard to solve it. One way it will be to manually perform thousands of predictions for different companies and storing them inside databased so when a user searches for a company the data will instantly present. But this will need massive computer power and storage space.

### *6.4 FUTURE IMPLEMENTATIONS*

For future implementations, the above problems could be resolve by using the proposed solutions. Further, there are dozen of opportunities and future implementations like login functionalities, giving users options like dynamic price graph or a dynamic number of articles to be used for sentiment analysis.

## *7. REFERENCE LIST*

Bangalore, R.S.S. is the P.D. for the U.-I. and Program, P.D.D.A. (2019). *7 Advantages of using Python for Data Science*. [online] upGrad blog. Available at: <https://www.upgrad.com/blog/advantages-of-using-python-for-data-science/>.

Beri, A. (2020). *SENTIMENTAL ANALYSIS USING VADER*. [online] Medium. Available at: <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>.

Bohmian (2020). *Sentiment Analysis of Stocks from Financial News using Python*. [online] Medium. Available at: <https://towardsdatascience.com/sentiment-analysis-of-stocks-from-financial-news-using-python-82ebdcefb638>.

Bowman, R. (2019). *Sentiment Analysis - Market sentiment and how it affects the stock market*. [online] LEHNER INVESTMENTS. Available at: <https://catanacapital.com/blog/sentiment-analysis-stock-market-sentiment/> [Accessed 29 Apr. 2021].

Brown, T. (2020). *7 Amazing Website Color Schemes 2021 + Color Mood Board*. [online] Hook Agency. Available at: <https://hookagency.com/website-color-schemes/>.

Brownlee, J. (2016). *Time Series Prediction With Deep Learning in Keras*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/time-series-prediction-with-deep-learning-in-python-with-keras/> [Accessed 29 Apr. 2021].

CHEN, J. (2019). *Stock Analysis*. [online] Investopedia. Available at: <https://www.investopedia.com/terms/s/stock-analysis.asp>.

Coyier, C. (2014). *Responsive Meta Tag*. [online] CSS-Tricks. Available at: <https://css-tricks.com/snippets/html/responsive-meta-tag/>.

Darling Jiminez (2018). *10 Tips That Can Drastically Improve Your Website's User Experience*. [online] Hubspot.com. Available at: <https://blog.hubspot.com/marketing/improve-your-websites-user-experience>.

Dive, H. (2014). *What makes a company attractive?* [online] Firstlinks. Available at: <https://www.firstlinks.com.au/makes-attractive-company> [Accessed 29 Apr. 2021].

Font Awesome 5 (2017). *Font Awesome 5*. [online] Fontawesome.com. Available at: <https://fontawesome.com/>.

<https://www.facebook.com/jason.brownlee.39> (2017). *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>.

IBM Cloud Education (2020). *Article us-en\_cloud\_AP\_django-explained*. [online] [www.ibm.com](https://www.ibm.com). Available at: <https://www.ibm.com/cloud/learn/django-explained>.

Inc, S. (2018). *Why Python & Django Are Your Top Choice for Web Development*. [online] Medium. Available at: <https://medium.com/swlh/why-python-django-are-your-top-choice-for-web-development-bff93e129448>.

Interactive, C. (2018). *Tips for Building a User-Friendly Website*. [online] [www.commonplaces.com](https://www.commonplaces.com). Available at: <https://www.commonplaces.com/blog/10-tips-for-building-a-user-friendly-website/>.

JetBrains (2020). *Features - PyCharm*. [online] JetBrains. Available at: <https://www.jetbrains.com/pycharm/features/>.

JOF, C. (2018). *11 Ultimate Tips That Will Make Your Website User-Friendly*. [online] Designhill. Available at: <https://www.designhill.com/design-blog/ultimate-tips-that-will-make-your-website-user-friendly/>.

JS Foundation (2019). *jQuery*. [online] JQuery.com. Available at: <https://jquery.com/>.

Kowalski, A. (2019). *How to Web Scrape with Python: Scrapy vs BeautifulSoup*. [online] [www.softkraft.co](https://www.softkraft.co). Available at: <https://www.softkraft.co/how-to-web-scrape-with-python/>.

Lu, L. (2018). *8 Ways to Make Your Website More User Friendly | Marketo*. [online] Marketo Marketing Blog - Best Practices and Thought Leadership. Available at: <https://blog.marketo.com/2018/06/8-ways-make-website-user-friendly.html>.

Miliushchanka, D. (2019). *10 Great Examples of Website Navigation Design | AGENTE*. [online] Agentestudio.com. Available at: <https://agentestudio.com/blog/website-navigation-design>.

Mishra, D. (2019). *Regression: An Explanation of Regression Metrics And What Can Go Wrong*. [online] Medium. Available at: <https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and-what-can-go-wrong-a39a9793d914>.

Novoseltseva, E. (2017). *Top 7 benefits you get by using Github / Apiumhub*. [online] Apiumhub. Available at: <https://apiumhub.com/tech-blog-barcelona/using-github/>.

Python, R. (2021). *Sentiment Analysis: First Steps With Python's NLTK Library – Real Python*. [online] realpython.com. Available at: <https://realpython.com/python-nltk-sentiment-analysis/> [Accessed 29 Apr. 2021].

Rains, J. (2013). *What Makes a Company an Attractive Investment?* [online] Wise Bread. Available at: <https://www.wisebread.com/what-makes-a-company-an-attractive-investment> [Accessed 29 Apr. 2021].

says, S. m (2019). *Python Anaconda Tutorial - Learn the most popular Data Science platform*. [online] DataFlair. Available at: <https://data-flair.training/blogs/python-anaconda-tutorial/>.

Serdar Yegulalp (2018). *What is TensorFlow? The machine learning library explained*. [online] InfoWorld. Available at: <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>.

Singh, K. (2013). *Web Design: 11 Characteristics of a User-Friendly Website*. [online] Social Media Today. Available at: <https://www.socialmediatoday.com/content/web-design-11-characteristics-user-friendly-website>.

Staff (2016). *8 Types of Modern Navigation Menus for Websites*. [online] TemplateToaster Blog. Available at: <https://blog.templatetoaster.com/types-modern-navigation-menus/>.

Twitter (2020). *5 Practical Types of Website Navigation to Consider*. [online] Lifewire. Available at: <https://www.lifewire.com/types-of-website-navigation-1697495> [Accessed 29 Apr. 2021].

Virtucio, M. (2019). *The Pros and Cons of using Trello Software*. [online] Bridge24.com. Available at: <https://bridge24.com/blog/the-pros-and-cons-of-using-trello-software/>.

Visual Studio Code Docs (2016). *Visual Studio Code*. [online] Visualstudio.com. Available at: <https://code.visualstudio.com/Docs/languages/html> [Accessed 1 Nov. 2019].

Yegulalp, S. (2019). *What is Jupyter Notebook? Data analysis made easier*. [online] InfoWorld. Available at: <https://www.infoworld.com/article/3347406/what-is-jupyter-notebook-data-analysis-made-easier.html> [Accessed 29 Apr. 2021].

## 8. APPENDIX

### 8.1 IMPORTS

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>TEST</title>
7   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
11  <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
12  <link href="css/style.css" rel="stylesheet">
13  <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
14  <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
15  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
16  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-B4dIYHKNBt8Bc12p+MkckhzcICoBwtJAoU8Y2TY5qE01d1G5seTk6s+L3B1XeVIU" cr
17 </head>
18 <body>
19
20 <!-- Navigation -->
```

Figure 39: HTML Head imports

```
1 from django.shortcuts import render, redirect
2 from django.http import HttpResponse
3 from io import StringIO
4 from .utils import get_plot, get_plot2, get_plot3, get_plot4
5 #My imports
6 import pylab
7 from urllib.request import urlopen, Request
8 from bs4 import BeautifulSoup
9 import os
10 import pandas as pd
11 import numpy as np
12 import matplotlib.pyplot as plt
13 from numpy import array
14 #imports for forecasting
15 import math
16 import pandas_datareader as web
17 from numpy import array
18 import datetime as dt
19 import tensorflow as tf
20 from sklearn.preprocessing import MinMaxScaler
21 from tensorflow.python.keras.models import Sequential
22 from tensorflow.python.keras.layers import Dense, LSTM
23 from nltk.sentiment.vader import SentimentIntensityAnalyzer
24
25 finviz_url = 'https://finviz.com/quote.ashx?t='
26
27 # Create your views here.
28 template_name = 'Website/index.html'
29
30 def hi(request):
31
32     return render(request, template_name)
33
34 #Close price history graph, sentiment analysis, Analysis panel
35 def function1(request):...
36
37 #Future price prediction, future analysis panel
38 def function2(request):...
```

Figure 40: All libraries were imported at the top of the views.py file followed by the two new functions with the algorithms created in Jupiter Notebook.



## 8.2 QUESTIONNAIRE

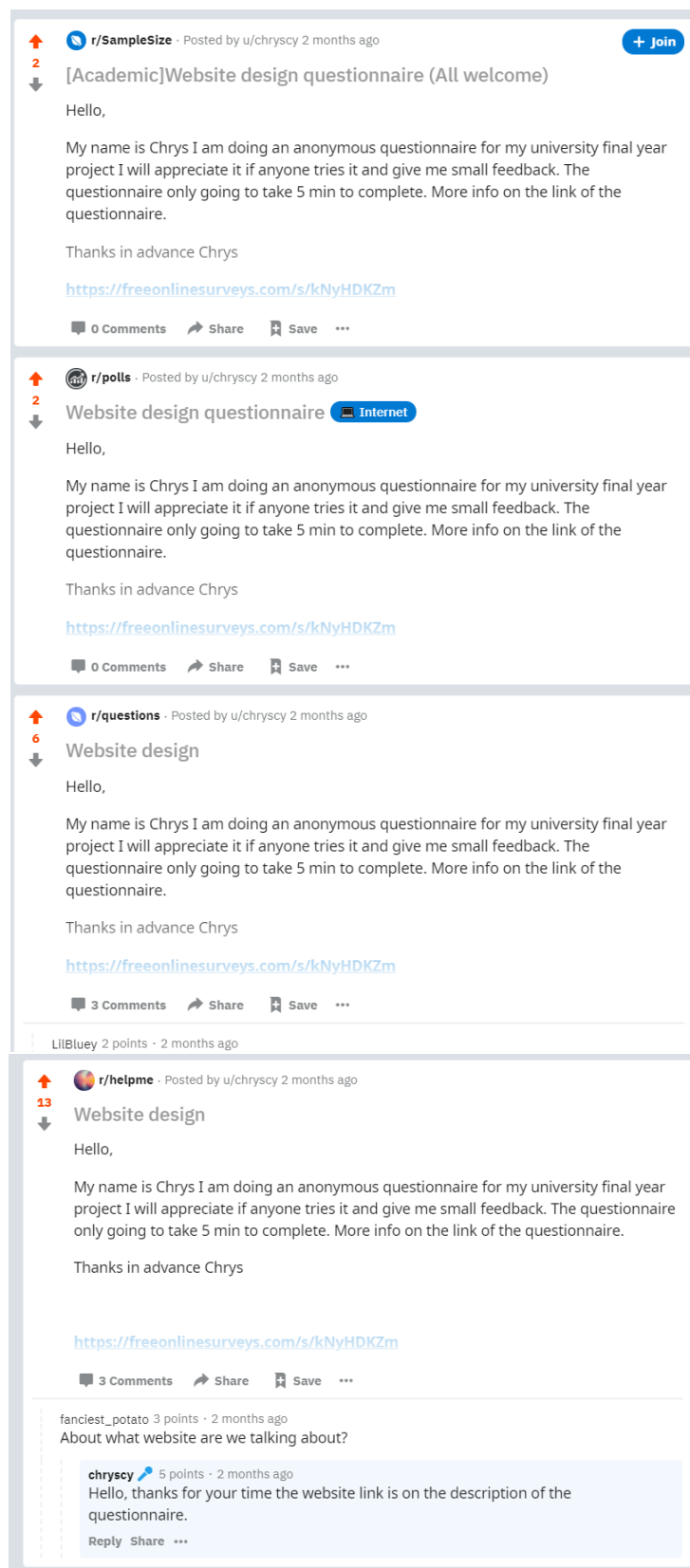


Figure 41: Post published in Reddit to get feedback for the website design.

# Stock Market analytics and general news for non-professional traders Questionnaire

My name is Chrysovalantis Papadimitriou and I am undertaking a research project as part of my studies for a first degree in Computer Science at Sheffield Hallam University. This questionnaire aims to gather information from the general public about aspects of the design of a website associated with this project. I hope that you will be able to find the time to complete the questionnaire as accurately as you can. It will not take much of your time. The University undertakes research as part of its function for the community under its legal status. Data protection allows us to use personal data for research with appropriate safeguards in place under the legal basis of public tasks that are in the public interest. A full statement of your rights can be found at

<https://www.shu.ac.uk/about-this-website/privacy-policy/privacy-notices/privacy-notice-for-research>.

After the completion of the project, all answers will be deleted. No personal information or any personal data is asked for in this questionnaire. Access to this questionnaire will be restricted to only myself and my supervisor.

It is up to you to decide if you want to take part. A copy of the information provided here is yours to keep, along with the consent form if you do decide to take part. You can still decide to withdraw at any time without giving a reason, or you can decide not to answer a particular question.

By completing my questionnaire you will be helping me by giving your own opinion. Later you can find the results of this questionnaire on my website.

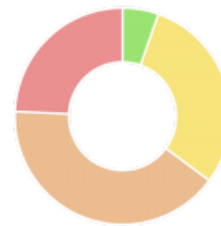
If you have any queries about my project or wish to comment on the contents of the questionnaire, please don't hesitate to contact me at [chrys700@yahoo.gr](mailto:chrys700@yahoo.gr)

Thank you for your participation.  
Chrysovalantis.

Early-stage of the website (Focused on design)

Link of the Website → Website Link

1 What was your first impression when you entered the website?



0% (0)  
Very Bad

5% (2)  
Bad

30% (11)  
Neutral

41% (15)  
Good

24% (9)  
Very Good

5.61  
Standard Deviation

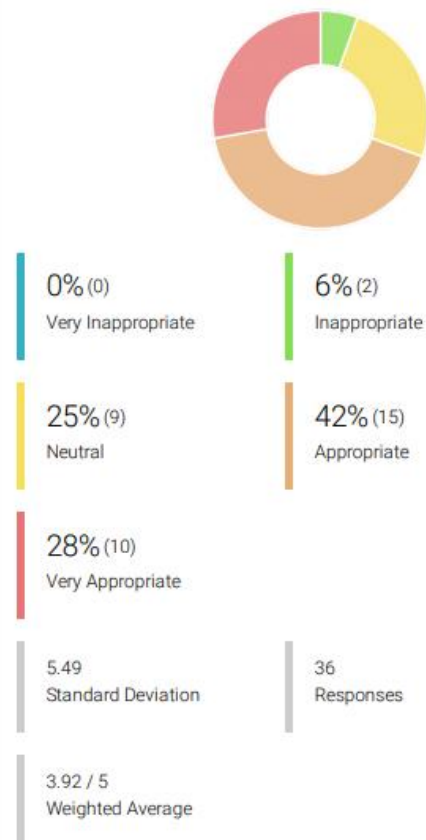
37  
Responses

3.84 / 5  
Weighted Average

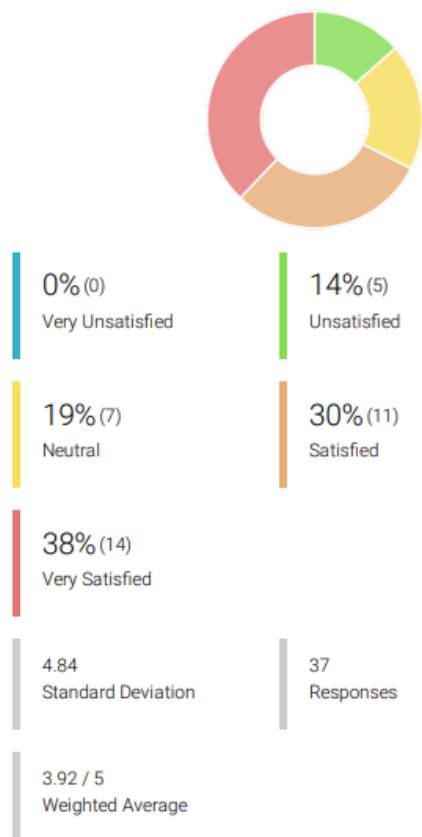
## 2 How did you find the design of the website?



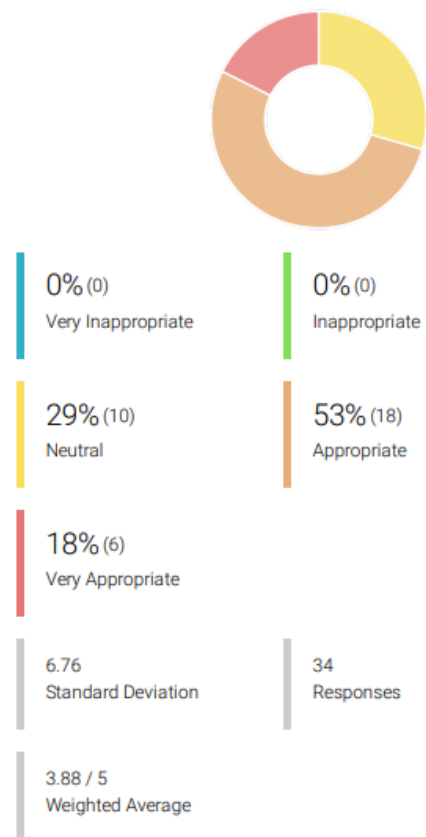
## 3 How Satisfied were you with the colors of the website?



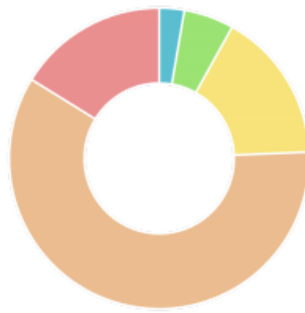
**4** How Satisfied were you with the navigation of the website?



**5** Were the images of the website relevant?



## 6 How did you find the information section?



3% (1)  
Very Useless

5% (2)  
Useless

16% (6)  
Neutral

59% (22)  
Useful

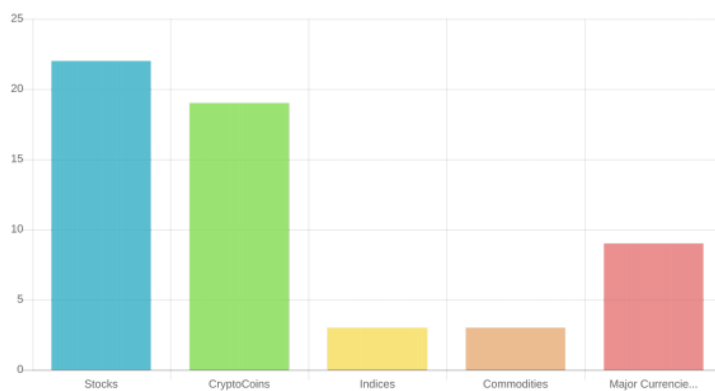
16% (6)  
Very Useful

7.58  
Standard Deviation

37  
Responses

3.81 / 5  
Weighted Average

## 7 Select the Markets that you are interested in



69% (22)  
Stocks

59% (19)  
CryptoCoins

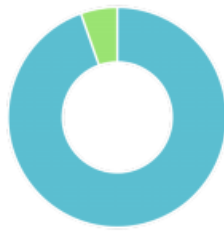
9% (3)  
Indices

9% (3)  
Commodities

28% (9)  
Major Currencies

56  
Responses

**8** Did the website render well in your browser?



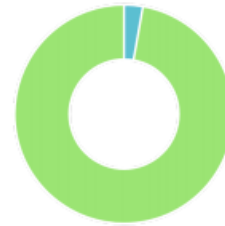
95% (35)  
Yes

5% (2)  
No

16.5  
Standard Deviation

37  
Responses

**9** Did it take too long to load the website?



3% (1)  
Yes

97% (35)  
No

17  
Standard Deviation

36  
Responses

## 10 Is there anything missing on this page?

No

Pricing, reviews, a reference to a white paper proving that the system works

No.

No

no

No, everything looks great :)

Not that I saw

no

No

No

Thoughtful proofreading

Don't think so

Not that I can think of

No

Different sections, as it seems everything is on a single page, which makes it a little thick to read through and find what you're looking for. Unless there are already different sections.

NO

A more dynamic look, it has an empty feel.

Not aware of anything missing.

No, its a great website

No

The purpose is not clear

It should inform the user when he types an invalid input

None

Yes i cant contact you :(

## 11 Any features suggestions?

No

None.

no

I enjoyed it.

I'm not a big fan of the "stocky" logo. I like the overall layout of the website but the logo is a little too much in my opinion. Something a little more minimalist would help

when you answer a question and the box forcefully drags you down to the next question directly below, remove it, its kinda annoying and its not that hard to scroll down yourself

A way to go back to the first view after scrolling down (Go up button). Some animations to make it feel less clunky.

If you're going to set up a website that's in English, you should get someone who speaks English natively to read it through. The site is full of typos and errors. If you are presenting yourselves as experts, it isn't useful to have a site with so many errors. It doesn't at all make me find you trustworthy. Also, I don't understand what makes this site more useful than any other finance site. What's different about it? It's hard to say from the content.

No, I really like it

None

No

Get someone to proofread the text and correct grammatical errors, use a slightly smaller font, don't bold as much words. Searchbar at top

NOT REALLY

The search bar extends more than it should (on phone).

Possibly a graph with popular stocks/stocks gaining popularity or a big growth

No

when typing the name of the stock, it would be better if the search bar stayed expanded.

Try to add different colors that will fit to each other

Not really

A contact button on the navigation menu.



## 12 What could we do to make this site more useful?

fix the layout bugs, fix the text

The header and buttons in the hero slideshow can be kept constant, i.e fixed while the background image changes.

nothing really

I'm not too sure! I thought it was all great.

The colour palette could be cleaner; blue for example doesnt fit at all. Light green is more modern in my opinion and a dark blue/white would give a nice and professional look

No suggestions

just add more information, make it interesting, right now its just a survey, make it useful

Add subcategories when fully developed

Too much to say, the site is basically empty right now

Better copy, more straightforward writing style.

It is useful as is.

Nothing

I don't know

More options

as simple as possible

It dies right

yes

Add more information

Add more data

Better information section.

### 13 Do you have any final comments you would like to raise?

the top gallery is useless, site scrolls horizontally on mobile, the text contents explains the website to your teacher but not to your users

None.

no

Nopes. :) looks amazing.

Keeping the design simplistic is great. I enjoyed scrolling and liked the not moving background image.

Nope

nope, have a nice day :)

Don't make the search bar disappear after you move the cursor away from it.

Keep trying!

Keep up the good work!

No

Seems good, atb:)

The font un the beginning title doesn't look very good. It's the forts and only thing I see on phone and I didn't appreciate it very much.

The design of the first picture could be improved.

Its a great website, very nice design as well.

No

no

**Figure 42: Questionnaire Questions and Responses.**