**WebApp to Scrap and Display Stock’s Variations**

Felipe de Oliveira1

1 University of Memphis, Memphis TN 38152, USA

**Abstract**. The growth of the Internet has brought different approaches to the matter of distributed systems during the years. Thus, the world is putting all of their old and slow work over applications deployed in the web environment. As an example, years ago, a stock trader had the necessity of having different channels for checking the market, including newspaper, phone, magazines and street information. However, with the boom of the Internet those same people have all of their information right in front of them on a computer screen. Those applications have become popular, because of the increase on the ease in their deployment on the web environment. Each day that goes on, there are new tools that help the developers in focusing in their touchable product instead of wasting time in hard matters that might only delay its development. As a result, people have easier access to better, more secure, and confident applications.

**Keywords:** Web App, Docker, Django, Web Service, Stock.

1. **Introduction**

The project will bring through a webpage an application where its users might check the variation on stocks’ prices from many companies around the America. This will be a 2-tier application that will use 24/7 updated data coming from trustful sources that will give the users a graphic solution helping them to decide whether it is a positive or negative time to make a movement in the market. The project’s ideal is to become a useful tool for traders who want to improve the income derived from their trades using charts and numbers. To make all of this possible, this project has in its sight the use of the most used useful structure now days, the Web. With its growth in the last 20 years, the use of web applications has been growing at light speed. The mobility and distribution of powerful processing delivered by web apps to its users make them the golden eggs of our time.

Due to this new era, developers and companies are moving and building their applications using all that a Platform As A Service (PaaS) has to deliver. Therefore, there is an expansion in the numbers of companies providing to users/developers this kind of structure for their application, making the market and the opportunities sprouting from everywhere. As a result of this growth, this project follows the concept that we should concentrate the efforts, and money, in places where the people are looking at with interest. After checking the market and comparing these companies, I chose the service made available by Amazon, which is called Amazon Web Services (AWS). The AWS offers a range of applications reachable by anyone with an account with no need for payment, at least not at the beginning. One of these applications used by this project is called Elastic Beanstalk where the application tier will be deployed using distribution of processes between the user and the server. There are other applications that are used to build this project, and we are going to see them from now forward.

1. **Related Work**

Stocks and market are always a hot trend in the day-to-day life of many people around the globe. Due to this event, the market for tools that will help these people to grasp a better perspective about what is happening inside of the general market have growth with different tools showing up. To give examples of how the market for these tools are expanding we can quote many of the apps developed by some of the biggest tech companies like Stocks by Apple, Finance by Google, Yahoo Finance by Yahoo, just to name a few. Even though this kind of app is spreading, the same is happening with its costumers. As one of the costumers of these apps, it is always possible to notice that these common apps do not deliver the user a specialized environment. Instead, it works in a general way that might not fit the taste of some traders. Each user has its own way of reading and understanding data, what makes it very difficult for a general application to deliver a perfect experience to all users. Therefore, this project has not the purpose to please all the users, but it is more focused in delivering something special and different from out there by using the vision of its developer.

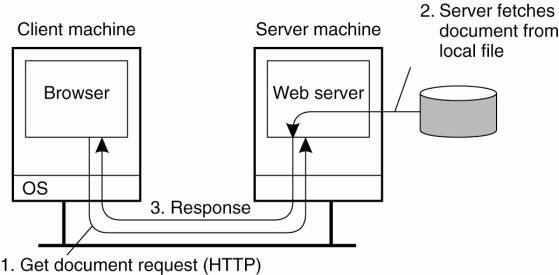
1. **Design**

At this section, we are going to see the implementation of a web model and how the project fits in this environment. As every project developed by a person, it is very important to know all the nuances that it implies. The Internet is a big network with different approaches available in the development and deployment of an application. There is so much content that needs to be understood before we start the creation, that we might not quote all of them in this article. Therefore, this project’s studies made over the distributed way that a web application works will be approached forward in this chapter.

* 1. **Distributed Architecture of the Web**

There has always been a discussion over the real starting point of the Internet. Some people believe that the U.S Army started it over the ARPA project. Others believe that it only might be consider today’s Internet after the studies of Berners-Lee. In this project, they believe that it initiated in one and was improved immensely by the other is what remains. Therefore, for purposes of keeping the content readable we only quote the last one. The Internet has never been the same since Tim Berners-Lee proposed at Switzerland in 1989 that CERN management adopt a distributed information system based on hypertext as a solution to the institution's communication problems. Berners-Lee demonstrated how information was lost daily at CERN proposing the development of a system for exchanging documents among the community, initially called Mesh. Eventually, it convinced the managers and was deployed at CERN and at 1990 it was named the World Wide Web. After few years, everything emerged on the Internet: academic and research websites, data sharing, video casting, etc. Today, Lee’s idea has spread all over the world with no need of longer presentation.

To start the explanation of the architecture over which the Internet is deployed, we initiate dividing the application between client and server. There are different approaches where client-server might fit in that would take an entire article only to explain it. As a result, we explain only the necessary content to give a better glimpse how this project’s architecture works. In computing, dividing the application in client-server is a distributed application framework that distributes the tasks and workloads between the providers of a resource or service, designated as servers, and the service requesters, clients.

***Figure 1****. The overall organization of a traditional Web site (Tanenbaum AS, Steen MV, 2014).*

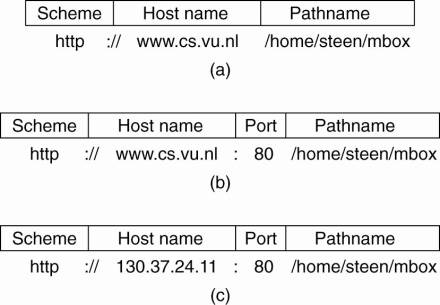
The above figure shows exactly how the overall web-based systems are organized. It shows the route of a client’s request to its server where the browser accepts input from and calling the server for the work. Also, the browser has the duty of displaying the server’s response to the user. The communication between a browser and Web server is standardized: they both adhere to the HyperText Transfer Protocol (HTTP) (Tanenbaum AS, Steen MV, 2014). These communications will be discussed below.

The communication between client and server is very complex and it also depends in the matter of what the client is requesting from the server. This article will dissert about the TCP/IP and HTTP only, but there are different protocols that are used to build the Internet as we see it today. First, we start defining the TCP/IP protocol. This protocol represents, to a certain extent, the set of Internet communication rules and is based on the notion of IP addressing, that is, providing an IP address to each machine on the network in order to be able to forward data packages of data. The TCP/IP protocol is designed to meet several criteria, including: The fractionation of messages in packets, the use of an address system, data routing in the network and control of data transmission errors. To perform this criteria, the protocol is divided into 4 layers of abstraction: Application (communication between the programs and the transport protocols), transport (controls host-to-host communication), internetwork (responsible for the connections between the local networks, thus establishing the interconnection) and network interface (it makes the connection between the networks and the installed applications of a machine as email, navigation, archives exchanges) (Dye, Mark; McDonald, Rick; Rufi, Antoon, 2007). During a transmission, the data crosses each layer at the issuing machine level. At each layer, information is added to the data packet, it is a header, a set of information that ensures transmission. At the receiving machine level, when passing in each layer, the header is read and then deleted. Thus, at the time of reception, the message is in its original state.

One of the protocols present in the TCP/IP is the HTTP (Hypertext Transfer Protocol) is the base for the communication in the World Wide Web. It is present in the Application layer quoted above. A client establishes a connection with another server and sends a request containing the URI, the protocol version, a MIME message containing the request modifiers, customer information, and possibly the content in the body of the message. The server responds with a status line including its protocol version and a successful operation code or error code, followed by server information, meta-information of the entity, and possible content in the body of the message. After the server sends the response the connection is terminated. To be able to perform this communication, server and client need to know each other. It is possible thanks to a property called Naming, that will be discussed in the chapter below.

* 1. **Naming**

At this chapter, we will be introduced to some concepts of how each participant on the Web can be found using naming. It starts describing the idea of how the Web, as Tanenbaum AS (2014) says, uses a naming system to refer to documents, which is called Uniform Resource Locator (URLs). The actual syntax of a URL is determined by its associated scheme. It usually contains information about how and where to access a document (through the embedded DNS), the port number on which the server will be servicing these requests is also part of the URI (if null, it gets the default) and also a URL also has the document’s name that will be searched by the server. The structure might be seen in the figure below:



***Figure 2****.  Often-used structures for URLs. (a) only a DNS name. (b) DNS name with a port number. (c) IP address with a port number (Tanenbaum AS, Steen MV, 2014).*

To resolve the URLs in the figure 2 the server has a straightforward task. If the server is called by it DNS name, this name will have to be resolved in the IP address of the. Using the port number contained in the URL, the client can then contact the server using the protocol named by the scheme, and pass it the document's name that forms the last part of the URL. As we can see, all the URLs in the figure 2 are using the HTTP protocol to be accessed. If a client wants to communicate using another protocol, lets says FTP (protocol File Transfer Protocol), it will have to change the scheme.

Now that we have a glimpse how the URL is build, we need discuss the work done by the DNS. Domain Name System (DNS) servers are responsible for locating and translating the addresses of websites we enter into web browsers into IP addresses. We can think of DNS as a layer of abstraction between what we want, how to get into a website, for example, and the gears necessary for that to happen. Simply enter the desired address that the servers responsible for locating and translating to the corresponding IP number will do the rest - and in a fraction of seconds. Due to the importance of DNS it is necessary to ensure its security, it is provided by DNS Security Extensions (DNSSEC), an encryption scheme that uses public and private keys to guarantee the authenticity of the addresses consulted, to ensure that when an address is consulted it will be translated to the correct IP number, avoiding internet scams (Icann.org, 2018). In addition to being able to make browsing faster, many DNS servers offer fake or infected site detection and even parental protection to block adult content sites.

In order for the IP and DNS address system to work properly, global coordination is required. For this purpose, there is an organization responsible for assigning domain names and IPS addresses worldwide. This is ICANN, a non-profit organization that tries to keep all registered sites running on the Internet. The ICANN site has an open-minded variety of content to understand their functioning and also the functioning of the DNS as all.

* 1. **Asynchronous approach**

In the client-server environment our applications work through requests and responses where the client (Browser) sends a request and the server, where our application is, returns a response as we saw in communication matter above. There are two approaches to make this request: using synchronous or asynchronous calls. This project uses the asynchronous call to extract the data from the stock market, giving to the user a more continually enjoyable experience.

In an asynchronous request, there is no synchronization between requests, so we can send several requests in parallel, where each response returns when ready. To make use of this form of communication, our application uses the AJAX method. AJAX is the acronym for Asynchronous Javascript and XML. It calls a resource on the server from a Javascript code in the web browser, so that the result refreshes only part of the page without having to make an entire update of the page. So, when using AJAX, you are using a set of techniques or, even better, a web-based programming approach focused on providing better interaction for the user (dos Prazeres Farias, 2012). Therefore, it is not a tool or framework that can be downloaded from any website, because the software components that are fundamental to this approach are present in all web browsers: HTML, XHTML and XML; CSS; Javascript; and, the main one, a component called XMLHttpRequest.

If you are reading this article, I assume that you already have knowledge over most of the software components mentioned above. However, the XMLHttpRequest might be one component forgotten by the majority and that is why we are going to look more in depth in it. XMLHttpRequest is the technical component that makes asynchronous communication with the server possible. It is used to send HTTP or HTTPS requests directly to a web server and load the server response data directly back to the script (dos Prazeres Farias, 2012). Despite the XMLHttpRequest name, data can be received from the server via JSON, XML, HTML, or as plain text. Response data can be used directly to change the DOM of the current active document in the browser window without loading a new page document. For example, if the web server has formatted it as JSON it might be easily converted into a client-side data object for later use.



***Figure 3****: Comparing synchronous and asynchronous (James Garrett, J. (2005).*

The figure 3 gives an example of the difference between the classical web application model and the Ajax application model. In the first approach, the client requests data from the server and stops its activities until the server’s response reach it back. At this time, the web page will be update entirely making necessary to reload content that has not been change. On the bottom side, the Ajax application allows its client to keep working in the web page while the request is being processed by at the server side without any interruption in the current situation. Also, when the same server responds back, the only part of the web content changed will be the part where the Ajax effects.

* 1. **Web Scraping**

Possibly you have already had to access multiple sites, extract information from them and play on a spreadsheet to do some analysis. This caused you to spend several hours in the copying and pasting content only to consolidate the information and other several hours to do some analysis. Using the Web Scraping technique, you could save much of your time in consolidating information and spending it on generating insights that create value for your business.

Data scraping is a computational technique in which a program extracts human-readable output data from a service or application (website). Through automated processes, implemented using a bot tracker, this type of "scraping" of information is a way of making copies of data in which specific information is collected and copied from the web, typically in a central local database or spreadsheet, for later retrieval or analysis.

This tool can be considered very useful for many professionals because it might generate valuable insights using the information. Also, it allows the scraper to collect more precise information about the matter it is searching for, such as marketing and sales. One example where scraping has succeeded is by facilitating the search, manipulation and analysis of data for sales optimization and customization of customer service so much that several legitimate businesses have used it with this purpose.

1. **Analysis**

In this section we are going to read more about the tools and technologies that were used in the development of this project. To be able to deliver what was written in the previous section and create a Web App for use of many users (at same time) some technologies were studied during all the process of building this project. Every tool was cautiously checked out in the matter of how the addition of it at the application body would improve the product at its end. Also, this project had the purpose of adding more background knowledge to my tech portfolio. Over the next lines we will concepts and application of these techs.

* 1. **Python and libraries**

When choosing a matter for this project one of the goals was to improve my technologic portfolio. After studying a new language for this development I decided going with Python by several reasons. Starting by something personal: my latest interest in data-science. As a web developer, my interest got into this field over the last years. However, after reading and talking with professionals in the data-science area it became one of my focal study points. As it is known, Python is the most used language for this matter, having gorgeous libraries and a big community to help new-interested developers. Another point that makes me want to create this project using Python is its growing market at the web developing. However, the use of the language alone at the web environment is not adequate and lack in performance, which is why the Django framework is present in this project.

Django is a free and open source web application framework written in Python. In addition, it is the main framework for Python web applications today. A web framework is a set of components that helps you develop websites faster and easier. Django takes care of the heavy part of web development, such as handling requests, object-relational mapping, preparing HTTP responses, so that you spend your effort on what really matters. Also, its community is very active and helpful for the first trip developers like me.

As mentioned before, the data-science field is becoming one of my study points lately. However, this is the first time I tried to create an application using these techniques. Therefore, I tried to mix my knowledge in web development and data-science to delivery a graphic demonstration over the stock’s variation. This was possible thanks to some JavaScript libraries: JQuery and Plotly. Starting by JQuery, it is a very common JavaScript framework used to facilitate the development at client-side. It delivers an easier way of working with the DOM document if comparable with raw JavaScript. Web developers have used JQuery for a very long time, and its community across the Internet is huge and still growing even with the growth of new JavaScript technologies.

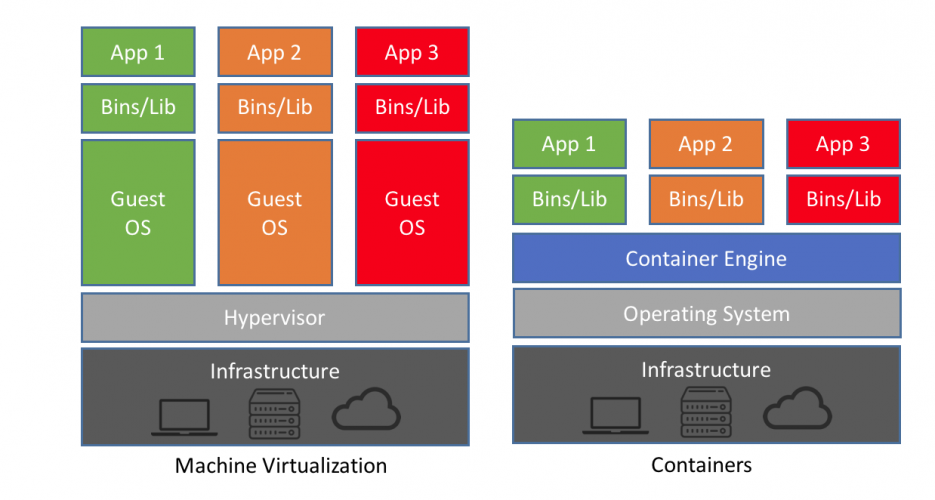
Plotly is data-analytics and visualization library known for its works with python. However, there is also a library in JavaScript with the same purpose at client-side. Plotly is able to create different kind of chart using numbers, matrixes, arrays, etc. The experienced had with this library during the development of the project has shown me how great and handy it turns out to be. Delivery of customizable charts and dashboards make the work with data faster and enjoyable.

Another library present is an API with the purpose of grabbing the stock’s data. This project started with the concept of having an API that grabs information directly from a database and shares it with the app’s user. However, after some studies in some APIs I started facing problems with it. As stock’s data is valuable information, most of the databases available are not sharing free data. And the free ones were closed by the bad use of this information for users around the world. One example is the database made available by Yahoo. This database was closed at the beginning of 2018 because the company was facing problems in sharing this data with outsiders. As a result, the company shut down the access to its server and all the APIs based on that got deprecated. After some research, I found out a new API called fix\_yahoo\_finance that, instead of having access to the database, uses web scraping (mentioned in 3.4) as a method of getting data. This API scraps the Yahoo Finance application returning data based in the parameters passed in the method call (company’s name, initial date, final date, etc.). It grabs all the information available at the web page and returns it to the application in format of Pandas dataframe. This way, we are able to separate the useful information and share it through the application’s interface. More about how the application works will be defined over the session 4.4.

* 1. **Docker and containers**

As a web developer for the last four years I know how hard it might be to deploy and create an environment either for production or development, process that might turns out a headache sometimes. Until now, when I wanted to initiate a new project I used a virtual machine to install the requirements needed for the development of the application. However, after a long time of problems, I decided to use this project as a bridge to learn more about how Docker might improve the development/deployment and also why this tool is becoming so famous in its niche lately. Thus, we will discuss more about the Docker’s concepts and also how it helped in the construction of this project.

We start by defining what is not Docker. Docker is not a traditional virtualization system. While in a traditional virtualization environment we have a complete and isolated S.O., inside Docker we have isolated resources (containers) that use common kernel libraries that are only possible because Docker uses a known Linux component, called LXC, as a backend. Thus, it is important to note that the containers are located on top of a physical server and the host OS. Each of them shares the host OS kernel and usually shares libraries and binaries as well. Therefore, Docker enables the packaging of an entire application or environment within a container, from which point the entire environment becomes portable to any other Host that contains Docker installed. Another difference is that shared items are read-only, which makes the container very lightweight, especially compared to the VM. To get an idea, the size of the first is Mb and so it starts in a few seconds. The virtual machine already contains several Gb and takes minutes to execute. The figure 4 below shows the division of labor in virtual machine and containers environment.



***Figure 4.*** *Difference between process execution in virtual machine and containers (Brey, 2018).*

A better explanation of the above figure and what it represents is that containers are a form of operating system-level virtualization that allows you to run multiple "systems" on a single real operating system. These isolated systems can be effectively isolated and limited in use of disk, RAM and CPU. Containers use a Kernel share trick to save resources, and hence the fact that they are a form of virtualization at the level of operational system.

Docker became a big part of the studies that were done in the development of this project. Even though it has a big community and different tutorial spread on the Internet explaining how to build an application over the containers, it also transforms the seek for the right information very tedious sometimes. However, after going after the information needed to put up a Web App using Docker, I decided the use of container with Nginx as a reverse proxy server and also static files server. In addition to it, I created another container based in Gunicorn (Python WSGI HTTP server UNIX).

To understand better the importance of Nginx for this project, we start by explaining what is a reverse proxy. A reverse proxy is a network server usually installed to be in front of a Web server. All outbound connections are addressed to one of the Web servers through a proxy server routing, which can process the request itself or forward the request all or part of a Web server, which will deal with it. A reverse proxy passes the network traffic to a set of servers, making it the only interface for external requests. There are different proxy servers out there, but after reading and grabbing information about the products I decided by the use of Nginx, due some reasons. According to the Web Server Survey, conducted by Netcraft in October 2015, the Nginx server was the second most used web server among the active web sites surveyed, being used by 15.33% of these. Also, due the research made, most of the users and articles conclude that Nginx has a higher performance over the competition.

Another container implemented by Docker in this project is the Python web services using Gunicorn. James (2018) explains that Gunicorn is a Python Http server that interacts with the web application via [Web Server Gateway Interface](https://wsgi.readthedocs.io/en/latest/what.html" \t "_blank) (WSGI), it might be considered as a middleman between the Web Application and the Web Server. The figure 5 gives a representation of the containers present and how they communicate to each other in the project environment.

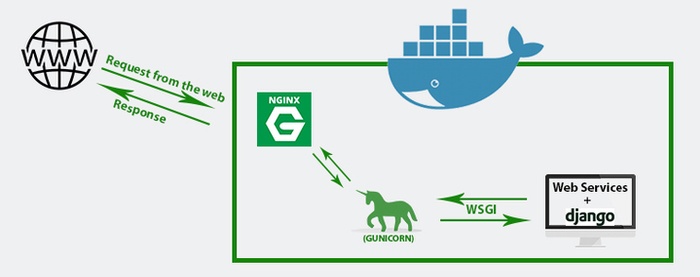


Figure 5. Representation of Docker containers present in the project (James, 2018).

The figure 5 shows the workflow of a request in this project. It starts by the client (Browser) making a request that heads directly to Nginx (Proxy Server), Nginx does it work and sends that request to Gunicorn (python Http server). Gunicorn receives and communicates with the web application via an interface (WSGI). After the work is done by the application, it responses return to the client using the same path which the request got in.

* 1. **Amazon AWS**

Every software development project comes to a point where you are faced with the following question: Where am I going to host my solution? Something that has shown lately not to be a good option is the shared hosting solutions, because they are no longer attractive because of their scalability and performance. Given this scenario and the evolution of technology, I turned to cloud providers that can meet this project needs. In addition, after looking for some cloud solutions available out there, I decided the use of the Amazon Web Service (AWS). The AWS Cloud infrastructure is built around regions and availability zones (AZs). Most of the information presented forward about this topic might be find in the AWS web site, which is an excellent tool to understand its functioning through tutorials, forum and articles helping its user to understand its potential and use all its power.

AWS consists of secure cloud products and services that provide computing power, database storage, network and content delivery services, developer tools, machine learning services, Internet of Things, augmented reality, and more to help organizations scale and grow. The AWS is divided, like others solutions seen, in regions. These regions are geographic regions that provide several separate availability zones, that is, separate, physically isolated datacenter facilities that are connected with low latency, high throughput and highly redundant networks. These availability zones offer AWS customers an easier and more efficient way to design and operate applications and databases. In addition, it makes them more highly available, fault-tolerant, and with much better scaling compared to traditional single-datacenter or multi-datacenter infrastructures. Today, AWS covers 53 availability zones in 18 regions around the world.

Although AWS has many available services to build and deploy an application in its domains, this project uses only a few of them. The main product used is the AWS Elastic Beanstalk. Following its documentation:

*AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.*

*You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.* (Amazon Web Services, Inc., 2018)

Elastic Beanstalks has some tools that transform and facilitate its use, making it unquestionable the best product for an agile deployment and scalable application. AWS Elastic Beanstalk makes use of Load Balancing and Auto Scaling to automatically scale and expand your application based on the specific needs of your application in a timely manner. In addition, the offer of execution in more than one region (mentioned above) offers an option to improve the reliability and availability of the application. The Auto Scaling tool provided by the AWS is an amazing approach to help user in delivering a full-scaled application

However, even with all the facilities about the deployment in the AWS serve mentioned so far, I was not able to create and deploy an instance in its services. After studying and developing solutions for this problem, I saw that, due my lack of knowledge in deployment, I should have started this project’s step with advance. Even though I still believe that the AWS has lots of tutorials and a helpful forum for questions, when it comes to deploy a Docker application it stills has a lack of information.

* 1. **The app tutorial**

From the beginning of this article we are seeing all the steps and technologies used to create the application. Now, we can have a brief presentation of the final application. We start with the client access that is done by the browser. Once the access is done, the application automatically searches for data using the access date in name of the Apple’s. This call is made through an asynchronous method with Ajax and JQuery framework. The application receives the call, formats the data and, using the fix\_yahoo\_finance API, performs an HTTP call on the Yahoo Finance application web page by reading the information and returning it in the form of a pandas dataframe. This information is then read, filtered and returned by the server to the client in the form of JSON.

When the client receives the JSON information it uses the Plotly library to generate a graphical demonstration about the data. Once the first automatic call is complete the user will have the chance to choose his next call.

It has as parameters some companies, stock exchange, and crypto currencies from where the data will be requested. It can also choose the start and end date of the search on which the data is related. In addition, by using the Plotly the user has the opportunity to change the presentation form of the same chart, choosing between candlestick and time series. This change does not require another call being that it is stored in variables for future uses. Once chosen, the parameters are sent to the application with the same path and form the first request.

1. **Future work**

This project has opened some doors and added so much knowledge that I am still thinking about where it can go and grow. The possibility of adding more features to it, as a paid stock’s database to have more secure and reliable source of data is one of the futures improvements. With this database onboard, I might create an entire application and made it available to other users. Also, after reading many articles about technologies that I did not use at this project due the lack of time, I might add some at the future creating an even more powerful and respected application.

After the development and studies made over this project I can see a bright future not only for this project but also for other web applications that I will be able to create now on. The knowledge acquired here is going to become very important in futures jobs and projects.

1. **Conclusion**

After the studies made, the conclusion reached on all the knowledge absorbed is that the Internet is a place to develop any distributed application. With the techniques and technologies that exist today, the development of applications in the web environment is increasingly intuitive and high performance. The Internet as a distributed system has eased users and aided companies in terms of processing. With the possibility of access to scalable and secure resources with mechanisms of fault tolerance this environment is perfect for any development at any time.

All the technologies present in this project will help the clearing the vision of development in this field now on. This view now encompasses not only the creation of functions and methods but also all deployment of the application. I feel capable of building a web application from the floor and allocate it for user access. This ability will allow me to be able to discuss the process as a whole improving my skills as a developer and, if everything goes right, an entrepreneur in the future.

**References**

1. Amazon Web Services, Inc. (2018). *AWS Elastic Beanstalk – Deploy Web Applications*. [online] Available at: https://aws.amazon.com/elasticbeanstalk/?nc1=h\_ls [Accessed 23 Nov. 2018].
2. Brey, P. (2018). *Containers vs. Virtual Machines (VMs): What's the Difference? | NetApp Blog*. [online] NetApp Blog. Available at: https://blog.netapp.com/blogs/containers-vs-vms/ [Accessed 14 Nov. 2018].
3. dos Prazeres Farias, J. (2012). *AJAX Tutorial: Um guia Completo sobre AJAX*. [online] DevMedia. Available at: https://www.devmedia.com.br/ajax-tutorial/24797 [Accessed 14 Nov. 2018].
4. Dye, Mark; McDonald, Rick; Rufi, Antoon (2007). Network Fundamentals, CCNA Exploration Companion Guide. Cisco Press.
5. Icann.org. (2018). *What Does ICANN Do? - ICANN*. [online] Available at: https://www.icann.org/resources/pages/what-2012-02-25-en [Accessed 22 Nov. 2018].
6. James Garrett, J. (2005). *Ajax: A New Approach to Web Applications | Adaptive Path*. [online] Adaptivepath.com. Available at: http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/ [Accessed 14 Nov. 2018].
7. James, S. (2018). *Nginx: Setting Up a Simple Proxy Server Using Docker and Python/Django... | Codementor*. [online] Codementor.io. Available at: https://www.codementor.io/samueljames/nginx-setting-up-a-simple-proxy-server-using-docker-and-python-django-f7hy4e6jv [Accessed 22 Nov. 2018].
8. News.netcraft.com. (2018). *October 2015 Web Server Survey | Netcraft*. [online] Available at: https://news.netcraft.com/archives/2015/10/16/october-2015-web-server-survey.html [Accessed 22 Nov. 2018].
9. Tanenbaum AS, Steen MV (2014) Distributed systems: principles and paradigms, 2nd edition. Pearson Education Inc, London, UK;