

CS 687 Capstone Project

Progress Report

Serverless Stock Market Web Application

Priyanka Kamdar
Advisor: Sion Yoon
MS in Computer Science
School of Technology & Computing (STC)
City University of Seattle (CityU)
[kamdarpriyanka@cityuniversity.edu, yoonhee@cityu.edu]

Abstract

Recent advancements in stock market technology and trading apps have attracted more individual investors to invest in the stock market. Individual investors account for around 38 billion USD of daily trading volume in the US stock market (Adinarayan, 2021). As the financial market dataset is large and changing rapidly this paper aims at building integrative web applications with cloud computing and machine learning framework. The web application is designed in a Microservice architecture with the use of ReactJS as frontend and Python Flask as a backend service. The web application is hosted on AWS serverless framework to provide availability and scalability using the Docker container. To make stock prediction more precise, Linear Regression which is a Supervised machine learning algorithm was evaluated and integrated into the stock web application. The outcome of this web application will provide comprehensive and valuable information to the users such as historical stock data, financial statements, analyst ratings, financial news, organization information and prediction of stocks.

Keywords: Stock Market application, ReactJS, Python Flask, AWS Elastic Beanstalk, AWS S3 Bucket, Docker container, Machine Learning, AWS CloudFront.

1. INTRODUCTION

Problem Statement

The stock markets are volatile, but it plays a vital role in tracking economic growth as it's linked with the GDP [9]. Higher the growth in the stock market, higher the economy rate and higher the GDP rate of a country. As the financial market is growing rapidly, the investors need to get all the key information about the company to decide whether it's worth investing in a particular company or not. To overcome investors' problems, currently, there are some websites or applications that provide financial information to the investors. But there is no, one platform that provides all the consolidated information i.e., historical stock data, company information, financial ratios, stock predictions, financial news etc. As the pieces of information are scattered across many applications it does not provide fruitful results, especially for the new investors to decide about the investment. Also, some of the applications require a monthly/yearly subscription which new investors don't prefer to pay.

Motivation

Richer investors usually seek advice from Financial Advisors for investment opportunities or seek financial information by purchasing a membership from web applications such as Yahoo Finance [14], MarketWatch [18], Seeking Alpha [16]. But the new investors devote a lot of time in finding financial information and investment opportunities, which in a way is stressful. Hence, to cater to new investors' issues, the goal of the project is to prepare a Stock application for the new investors with historical and financial information.

With the newly emerging technology of the Serverless framework, the stock application will be built and deployed on AWS. This will make our project application cost-effective with high availability and scalability.

Approach

This paper aims to build a more efficient web application that will help the Investor get an overview of the company's details, its historical financial data, financial ratios, and financial news

in one place, which will help the investor to determine the growth of the company and decide for investment. With emerging technology such as Machine Learning, the paper will also aim at building a Supervised Machine Learning model to help investors to get Stock predictions for the next 5 days, which will ease their decision in buying or selling the stock. Further, this application will be hosted on the AWS Serverless framework to provide scalability, increase agility, reliability, and cost optimization.

Conclusion

This project aims at building a Stock application with ReactJS as a Front end and Python Flask as a Backend. The complete web application will be hosted on AWS serverless framework to build a full-stack application per the need of the new investors.

Hence, this project will allow exploring many technologies related to frontend, backend, cloud computing, Docker container, machine learning models. With this project, I will be able to showcase my knowledge and skills.

2. Background

The stock market plays an important role in the economic growth of a country. Good functioning stocks markets can boost economic growth by accumulating capital.[9] The fundamental concept, knowledge and information needed for this research has been described below: -

Stock Market – Stock Market is a platform where Investors buy and sell stocks. Stocks are often termed Equity which means ownership of a company. The reason companies go public in Stock Market as it provides capital which eventually helps them to grow and the opportunity to those investors who invested, to be part of the profit.

Stock Analysis – Stock Analysis is a method that assists the investors in studying and evaluating the past, present and future data of a company to make buying or selling decisions of stock [12]. Stock Analysis can be Fundamental Analysis or Technical Analysis [10]. Fundamental Analysis is an analysis of a Financial Statement that focuses more on company assets, financial ratios, historical data, economic reports. Technical Analysis focuses on the study of past and present stock prices and predict their future price movements.

S&P 500 – There are various stock market indexes, but this project aims at stocks from S&P 500. S&P 500 is a stock index that pertains 500

largest US companies. It includes companies across 11 sectors [11].

Hence, Stock Analysis is essential for any investors before investing in any company to understand the company's financial health. In this paper, both Functional and Technical data are collected from the S&P 500 companies. This will give a good overview to the new investors about the company's historical financial data and performance plus future price prediction of the largest 500 companies in the USA.

3. Related Work

Literature Review

This part focus on many kinds of research, approaches and work done by other people in the past.

Review on ReactJS and AngularJS

Kumar & Singh (2016) did a comparative analysis between the two popular front-end JavaScript based frameworks i.e., AngularJS and ReactJS. AngularJS is a Model-view-controller model which can create a customized Document Object Model (DOM). ReactJS is a UI library and uses virtual DOM [6]. Both these frameworks are used to create a single web page application. According to them, the important differences are related to debugging, fail-time, component model, DOM. The distinction between the two frameworks helps to understand when either of these frameworks should be used. The paper concludes that it depends upon the user requirement and developers as some features of ReactJS are better than AngularJS and vice-versa.

Review on Serverless Architecture

Cloud Computing is a model that involves hosting web services over the internet. In the article Rajan (2018, December), the author had proposed a new model over the cloud computing model which is a serverless architecture. In this architecture model, the cloud service provider handles all the resources of the server, and the consumer will be billed only for the resource used by them. This implementation reduces the cost, increase the ability of the application to scale in or out in the cloud and minimize the configuration expenses.

The paper demonstrated step by step implementation of serverless computing using AWS Lambda and other services such as SNS, IAM, Cloud Watch and Auto Scaling Group.

Review on Supervised Machine Learning Models for Stock Price Prediction

Madeeh & Abdullah (2021, February) proposed and built a model for forecasting the financial stock price of the companies listed in the New York Stock Exchange (NYSE). The data was collected from the Kaggle repository. The work was classified into 3 stages. In the first stage, the collected data was preprocessed. In the second stage, the two Supervised Machine Learning Models K-NN algorithm and the Random-Forest algorithm was executed. For this 70% of the dataset were considered for Training the model and the rest 30% dataset was considered for evaluating the model. The third stage involved accuracy rate. For the K-NN model the Precision, Recall and F-Measure were 90.21%, 90.13% and 90.15% whereas for Random Forest the ratio was 93.23%, 93.12% and 93.17%. The Mean Absolute Error and Root Mean Square Error for the K-NN model were 0.0156 and 0.0967, whereas for Random Forest was 0.0127 and 0.0809. The prediction output of Random Forest seems more promising.

The main aim of the paper Naeimuddin & Vijaykumar (2020, December) was to compare the algorithm with each other for both large and small datasets and analyze the best algorithm with a high accuracy rate. In this paper, they executed 4 Supervised Machine Learning Models i.e., Random Forest, SVM, Linear regression, and K-Nearest Neighbor algorithms for the stock prediction. The data was collected from Yahoo finance, Kaggle, towards Data Science, NSE India. The dataset contains 1260 rows and 12 columns. Different company data sets were used to compare the 4 algorithms. The dataset was divided into two parts where 70% of the dataset was the Training dataset and 30% dataset was the Testing dataset. The paper concludes that the Random Forest algorithm was the best algorithm which provided an accuracy rate between 91%-95%. The accuracy rate for the KNN model was between 90%-93%. The accuracy rate for Linear Regression was between 80%-90% and the last SVM accuracy rate was between 75% - 85%, which was less compared to other algorithm models.

Review on Prediction over Cloud Server

Nowadays analyzing huge data in real-time becomes difficult for cloud providers to allocate resources. To overcome this, the paper Yadav (2021) focuses on using the Time series analysis models to handle the workload in cloud servers. To identify future trends, the authors implement long-short-term memory (LSTM) which is a Deep Learning method. Further, the prediction of the

model was measured using the metrics Root Mean Square Error (RMSE), Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Review on Technical and Fundamental indicator on Stock Prices forecasting

For predicting stock prices, the paper Beyaz (2018, June) analyze whether the fundamental analyses approach was better or technical analysis. Further, even a combined approach model was tested using machine learning algorithms for 140 companies from the S&P 500 index for 6 months or 1 year.

Review Conclusions

By studying and exploring the research paper, we can conclude that ReactJS will be useful for front-end development. For stock price prediction Supervised Learning algorithm gives a promising accuracy rate, which will be good for our stock application. And Serverless framework will reduce the cost and increase the scalability of stock application

4. Approach

User Requirements

The main aim of this project is to build a highly scalable and available web application for users i.e., investors to provide them with consolidated and valuable stock market-related information. The stock web application will also forecast near future stock price trends using a Machine Learning algorithm to assist investors in making an investment decision. Based on the intent of the project, the User requirements are categorized into tangible and non-tangible requirements.

Tangible Requirements

- Stock search engine by Stock ticker name.
- Stock dashboarding with historical price, financial information, analyst ratings
- Company non-financial information
- Stock price forecasting for the near future.
- Related stock news

Non-Tangible Requirements

- The stock market web application should be highly available, scalable, and reliable
- Easy to deploy and maintain

Architecture Design

Based on the User's tangible and non-tangible requirements, the project technology stack is divided into two main categories.

End to End web application stack:

Stock web applications will use ReactJS, as it provides better performance with a reusable UI component for a single web page application [3]. For the backend service, Python Flask is used to fetch and run the Machine Learning model for predicting the stock price for the near future

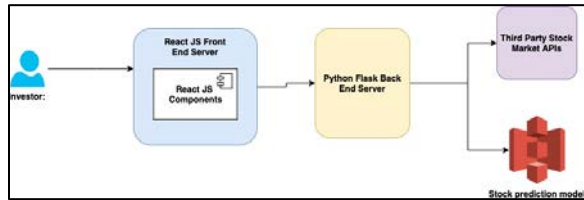


Figure 1. End to End Web application stack

According to the design shown in Figure 1. User i.e., the investor will use its device like a laptop or computer to interact with the web application through ReactJS which is a front-end server. ReactJS will render the views through different React components on receiving the user request. These components will also interact with Python flask (backend server) web APIs. The backend server APIs will then interact with Third-party APIs to get stock information. The back-end server API will also build a Machine Learning model on stock closing price, and it will forecast the stock price for the next 5 days. For building a Machine Learning model, a Linear Regression algorithm is used.

Serverless Architecture Design-

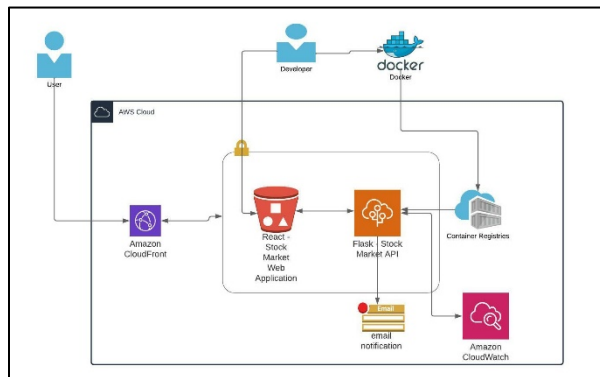


Figure 2. Serverless Architecture

Figure 2. implies the serverless architecture for hosting the stock market web application. According to this architecture diagram, the developer will push the frontend application to the S3 Bucket which will run the react application. Then the developer will create a docker image for the backend application (Python Flask API). This docker image will contain the required python libraries like Flask, Sklearn and Pandas as dependencies. The docker image will be pushed

to AWS Container Registries which is the repositories for the docker image. AWS Elastic Beanstalk is a platform service offered by AWS to host the docker containers. This service will fetch the docker image from registries. After fetching the docker image, Elastic Beanstalk will deploy and run the docker image as a backend service.

AWS CloudFront content delivery network service will connect front end application server hosted on cloud and back-end API running on AWS elastic beanstalk. End users will connect to AWS CloudFront URL to interact with the web application.

Technology Used

The software tools that have been implemented to prepare the full stack stock market application are as follows: -

- Operating System – Microsoft Window 10 and Linux
- Programming Languages – Typescript, Python
- Frontend framework - ReactJS
- Web Server Software – NodeJS server, Python Flask server
- Machine Learning Libraries – Pandas, NumPy, SKlearn
- Serverless Technology – AWS Elastic Beanstalk, AWS S3 Bucket, AWS CloudFront, AWS CloudWatch, AWS SNS, AWS Container Registry, Docker Container.

5. Data Collection

Data collection plays a vital role in any project. For this project, the data was collected from the Third-party API i.e., eodhistoricaldata. [17] The main advantage of using this API for this project are: -

- It provides robust, powerful, and reliable stock information [17]
- It provides more than 30+ years of data
- It provides different categories of Fundamental and Technical data of Stock
- It provides JSON/CSV format which can be easily integrated with Python, R, PHP, and many other web API

To collect essential stock information and for visualization below mentioned APIs were used from eodhistoricaldata [17]

- Stock Ticker and Name – To fetch the company stock tickers and the name “exchange-symbol-list” API [17] was used. To call the API the stock ticker name and exchange name are also provided as input.

- Stock price data – To fetch historical end of day stock price “eod” API [17] was used. To call the API the stock ticker name and exchange name are also provided as input.
- Stock Fundamental data – To fetch stock fundamental data such as company financial data like balance sheet, income statement, cash flow, analyst ratings, historical dividends “Fundamental” API [17] was used. This API takes stock ticker name and exchange name as input.
- Stock News – To fetch companies’ business news “News” API [17] was used so that investors know the latest information about the company. To call the API the stock ticker name and exchange name are also provided as input.

Data Collection Process

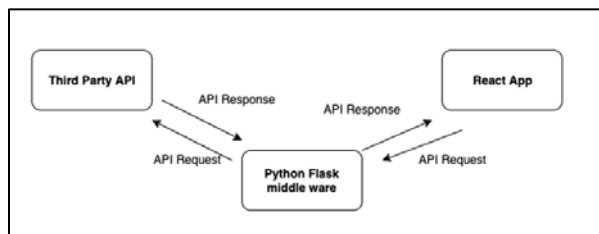


Figure 3. Data Collection Process

The process of data collection is crucial for any project. Hence, to get the accurate and appropriate data we have followed the process mentioned in figure 4. According to figure 3, React App will send the user request to Python Flask middleware. The Python Flask API will integrate with Third-party API i.e., eodhistoricaldata [17], to collect the required data. This data is in the form of JSON file format. The Python Flask API will return the data to React App to fulfil the user request. During this process, the data was inspected, transformed, modified, and prepared for various financial visualization.

6. Data Analysis

In this section, the user i.e., the investor can access the Stock Application to analyze and visualize the stock fundamentals data of any company from the S&P 500. S&P 500 is a stock index that relates to the 500 largest US companies. It includes companies across 11 sectors [11]. The requested company data is retrieved from the different API’s of eodhistoricaldata [17]. The requested data is presented in the form of the dashboard in the stock analysis web application. The dashboard

will give the users the visualization of company historical data, stock price trends, stock valuation, financial data, news, and stock prediction.

Stock Price Prediction with Machine Learning Algorithm

Stock price predictions help the users to determine the future stock price movement of the required company. With the Machine learning model, the prediction of stock price become promising. The historical stock price dataset is required, to train the machine learning algorithm. This dataset was collected from the Third-party API i.e., “eod” API [17]. It contains information such as Date, Open, High, Low, Close, Adj Close and Volume information according to Figure 4. The users use this information to measure the stock volatility.

Date	Open	High	Low	Close	Adj Close	Volume
3/5/2021	2073.120117	2118.110107	2046.415039	2108.540039	2108.540039	2195200
3/8/2021	2101.129883	2128.810059	2021.609985	2024.170044	2024.170044	1647400
3/9/2021	2070.000000	2078.040039	2047.829956	2052.699951	2052.699951	1697300
3/10/2021	2071.760010	2075.000000	2033.369995	2055.030029	2055.030029	1268600
3/11/2021	2074.060059	2125.699951	2072.379883	2114.770020	2114.770020	1239100

Figure 4. Historical Stock Price data

The description of the stock price data information is mentioned in Table 1

Open	The stock price at the beginning of the Trading Day
High	The highest point of the stock price on the Trading Day
Low	The lowest point of the stock price on the Trading Day
Close	The stock price at the end of the Trading Day
Adj Close	Stock closing price after accounting
Volume	Number of shares traded

Table 1. Historical stock price data

There are many Supervised Machine Learning algorithms to predict future stock prices. But this project uses a Linear Regression algorithm, as it’s a popular and widely used algorithm for predictive archetypical modelling [2][8]. The prediction was done using the “Close” stock price to get the forecast stock price for the next 5 days.

Execution and Result: -

The historical stock price dataset was collected from the Third-party API i.e., “eod” API [17]. This dataset was classified into two sets i.e., training and testing datasets on the proportion of 80% and 20% respectively to get a good accurate

prediction according to Figure 5. A linear regression algorithm was used. The result after testing the model is between 80% to 90%.

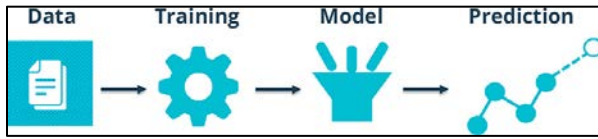


Figure 5. Machine Learning Prediction Model

7. Implementation

Since this project involves a full-stack application with serverless technologies, UI was built using the ReactJS framework. AWS S3 service is an object storage service that provides a feature to host the static web application. A new S3 bucket is created to host the front-end application. After bucket creation, react application was built and deployed using AWS CLI commands into the S3 bucket according to Figure 6.

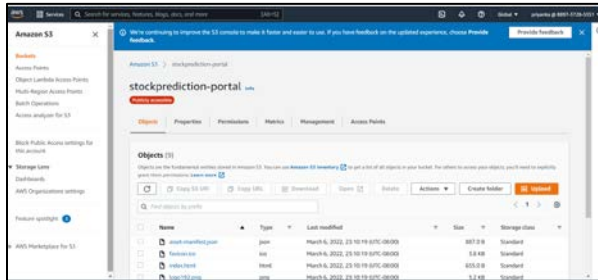


Figure 6. AWS S3 bucket configuration

The project uses the Python flask as a backend API server. Gunicorn web server is used as a backend server. A new container repository is created in AWS to store the Docker image file. Once the repository is created, a new Elastic beanstalk instance of type on-demand EC2 T2. micro is created to host the Docker container according to Figure 7.



Figure 7. Elastic Beanstalk configuration

AWS CloudFront service provides the content network delivery network to improve the website performance and to redirect the frontend requests to backend APIs. In the CloudFront service, a new distribution is created with the frontend URL as the origin. To re-redirect requests

to back-end APIs from the front-end, a new origin for back-end API URL is created within the same distribution with path pattern as `"/api/*"` according to Figure 8.

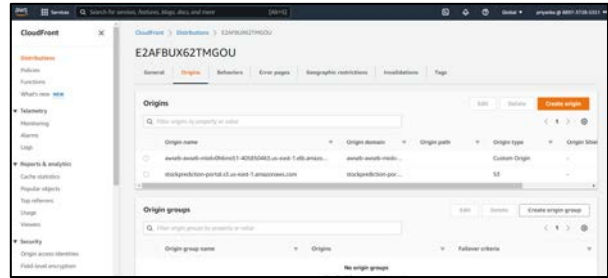


Figure 8. CloudFront configuration

8. User Interface

This section involves a complete visualization of the user interface of the stock application. It demonstrates various financial and technical data by company ticker symbol that will help the investors in taking the right decision. Some of the key visualization features are: -

1) Stock Ticker Search Function –

On the Stock web application, the user will be able to select the company by ticker name from the S&P 500, as shown in Figure 9. S&P 500 is a stock index that pertains to the 500 largest US companies [11]. Users can either enter the stock ticker which will display the closest match of the stock ticker or else the user can click on the required stock ticker to view the information.

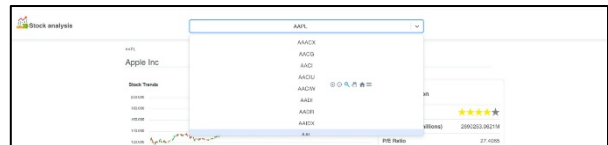


Figure 9. Stock Ticker Search Function

2) Historical Stock Price Trends –

Once the user selects the stock ticker, they will be able to visualize the selected company's historical stock price trends according to Figure 10.

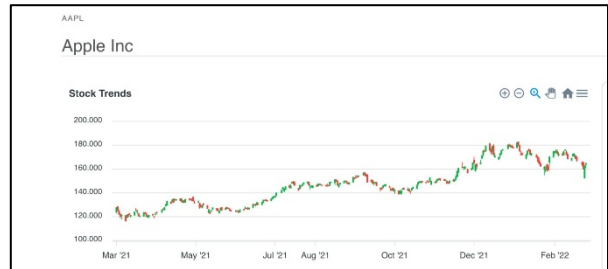


Figure 10. Stock Price Trends

For user convenience, there are some ad-hoc features added near to the Stock Trends such as:

- Zoom In and Zoom Out: - With this feature users will be able to zoom in or zoom out stock prices details for any specific day or month as per their requirement.
- Download the Stock Trends: - With this feature, the user will be able to download the stock price trends in different formats like SVG, PNG and CSV.
- Hoover function: - With this feature, once the user hovers on the stock price trends, they will be able to visualize the company stock Open, High, Low and Close price according to Figure 11. for any specific day or month per their requirement.

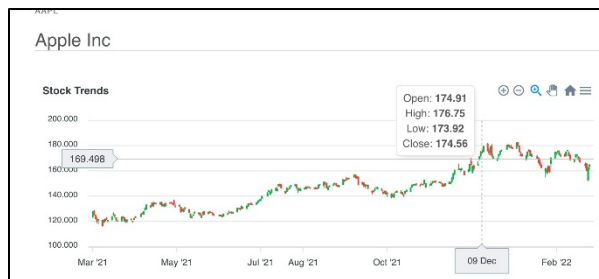


Figure 11. Stock Price Trends with Hoover function

3) Company Information –

According to Figure 12., the user will be able to get the complete company information including company CEO and Employees information. This helps the user to know about the basic company information, ownership and management team, mission statement, company history products they are involved in etc.

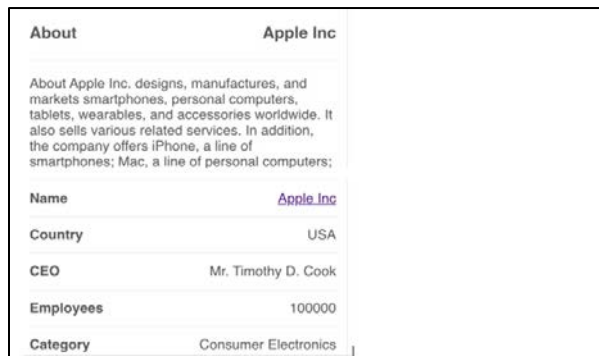


Figure 12. Company Information

4) Valuation and Analyst Rating –

According to Figure 13., the Stock web application dashboard will display the Stock Valuation and

Analyst Rating. Stock valuation provides the user with the Gross Profit, Revenue and PE ratio of the company. Analyst ratings play a key role as they measure the performance of the stock for a particular period. Hence, it serves as an important metric for the users.

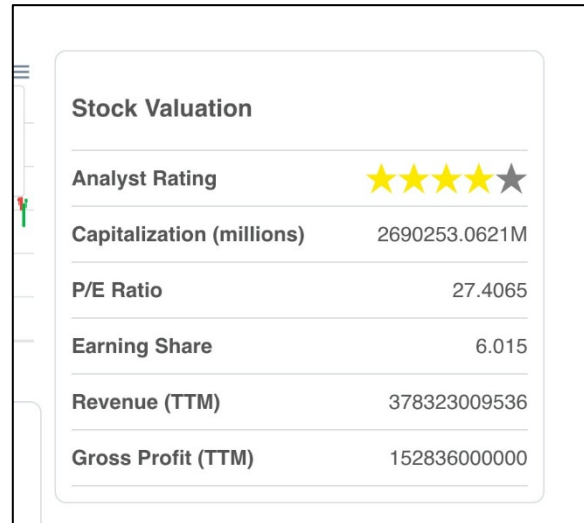


Figure 13. Stock Valuation and Analyst Rating

5) Financial data-

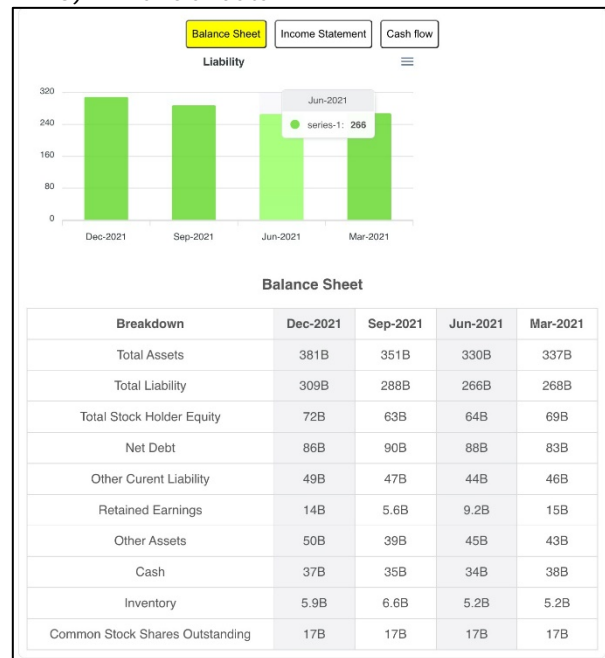


Figure 14. Financial Statements

According to Figure 14., the financial data section on the dashboard provides the user with a snapshot of the company's financial health. It gives a quick overview of the Balance Sheet,

Income Statement and Cash Flow. These 3 financial statements are important for the users to analyze the company's financial strength.

6) Trending News –

The Stock web application will show users the latest trending News of the selected company according to Figure 15. The news serves as a valuable tool for the users as positive news such as good earnings, companies' acquisitions, the launch of new products etc. denotes the buying opportunity for the users. It also helps users for sentimental analysis of the company.

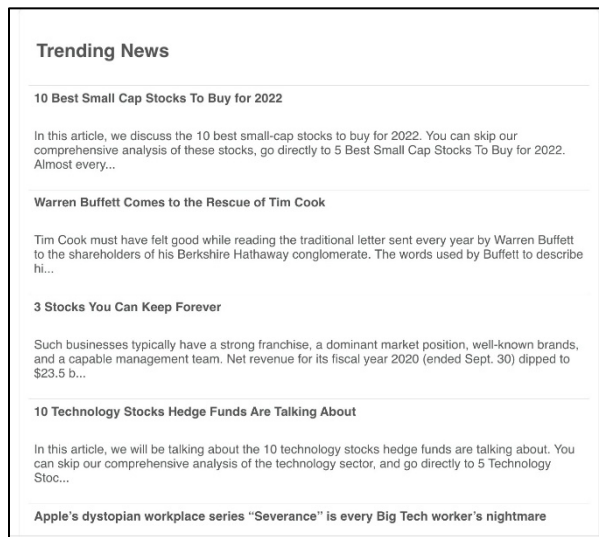


Figure 15. Trending News

7) Stock Prediction –

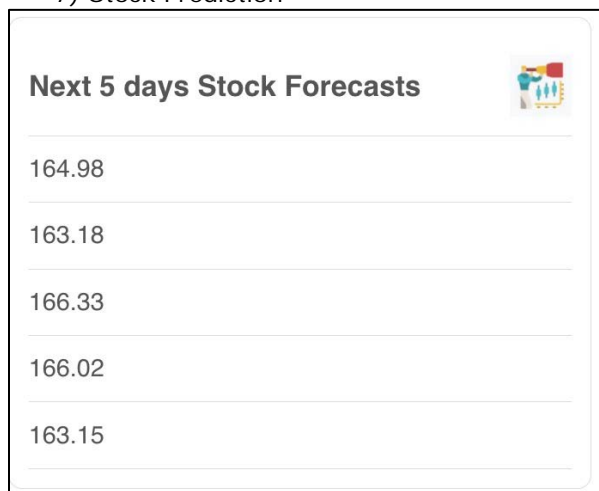


Figure 16. Stock Price Forecast

The user will be able to see the stock prediction price for the next 5 days on the dashboard according to Figure 16.

9. Findings

Cloud Implementation

The traditional approach is to have an on-premises server. But with this approach, the company need to purchase a physical server and hire staff to maintain the server which eventually increases the expenses. Keeping this in mind, the stock web application was hosted in the cloud to reduce the expenses and increase the high availability and scalability.

Stock Prediction

The Linear Regression algorithm proves beneficial for predicting the stock prices, as the accuracy for this model is between 80% to 90% which is a good rate. There are many other supervised machine learning algorithms that can be explored to get a higher accuracy rate.

The stock prediction helps the user to get an idea about the investment decision. But the user should not rely on the stock prediction, as the stock markets are volatile. There are many such factors that cannot be considered into account before it happens such as natural disasters, economic news, interest rate changes, economic factors etc.

10. Conclusions

As mentioned in the paper, the stock web application provides the user with basic stock investment knowledge and several useful data visualization and analysis tools for the new investors to help them decide which stock investment is worth investing in.

11. Future Work

To make the stock web application more effective and useful for the users below features can be included: -

Real-time stock prices: - To give users more reliable stock prices, the data process should be automated so that users can get the real-time stock prices which will help the investors to make a real-time decision.

Adding other indices: - Currently, the stock web application supports S&P 500. There are many other indices that can be included such as Dow Jones Industrial Average, Nasdaq Composite, Wilshire 5000 etc.

Ad-hoc Features: - To be more beneficial to the users, the stock web application should add some ad-hoc features such as stock comparison function, stocks bifurcated by industry sector, education function etc.

11. REFERENCE

- [1] Adinarayan, T. (2021, June 30). Business. Retrieved from Reuters: <https://www.reuters.com/business/retail-traders-account-10-us-stock-trading-volume-morgan-stanley-2021-06-30/>
- [2] Lawal, Z. K. (2020). Stock Market Prediction using Supervised Machine Learning Techniques: An Overview. 6.
- [3] Zammetti, F. (2020). Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker. Apress.
- [4] Rajan, D. P. (2018). Serverless Architecture - A Revolution in Cloud Computing. 2018 Tenth International Conference on Advanced Computing (ICoAC), 88-93.
- [5] Yadav, M. P., Pal, N., & Yadav, D. K. (2021). Workload Prediction over Cloud Server using Time Series Data. 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 267-272.
- [6] Kumar, A., & Singh, R. K. (2016). Comparative analysis of angularjs and reactjs. International Journal of Latest Trends in Engineering and Technology, 7(4), 225-227.
- [7] Madeeh, O. D., & Abdullah, H. S. (2021, February). An efficient prediction model based on machine learning techniques for the prediction of the stock market. In Journal of Physics: Conference Series (Vol. 1804, No. 1, p. 012008). IOP Publishing.
- [8] Naeimuddin, B. Vijaykumar (2020, December). Stock Market Trend Prediction using Supervised Machine Learning Algorithms. International Research Journal of Engineering and Technology, 7 (12)
- [9] Caporale, G. M., Howells, P. G., & Soliman, A. M. (2004). Stock market development and economic growth: the causal linkage. *Journal of economic development*, 29(1), 33-50.
- [10] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53(4), 3007-3057.
- [11] Davis, C. (2021, October 20). What Is the S&P 500? Retrieved from NerdWallet: <https://www.nerdwallet.com/article/investing/what-is-sp-500>
- [12] Chen, J. (2021, February 26). Stock Analysis. Retrieved from Investopedia: <https://www.investopedia.com/terms/s/stock-analysis.asp>
- [13] Beyaz, E., Tekiner, F., Zeng, X. J., & Keane, J. (2018, June). Comparing technical and fundamental indicators in stock price forecasting. In 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 1607-1613). IEEE.
- [14] y!Finance+. (n.d.). Retrieved from Yahoo! Finance: <https://finance.yahoo.com>
- [15] MarketWatch. (n.d.). Retrieved from MarketWatch: <https://www.marketwatch.com/>
- [16] Seeking Alpha. (n.d.). Retrieved from Seeking Alpha: <https://seekingalpha.com/>
- [17] eodhistoricaldata. (n.d.). Retrieved from eodhistoricaldata: <https://eodhistoricaldata.com/>