



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

TABLE OF CONTENTS

| | |
|---|-----------|
| 1. Proposal of the Project | 2 |
| 1.1 Proposed Project Description | 2 |
| 1.2 Clientele | 2 |
| 1.3 Plan Of Work | 3 |
| 2. Software Requirement Specifications | 3 |
| 2.1 Introduction..... | 3 |
| 2.2 Overall Description | 3 |
| 2.3 External Interface Requirements..... | 5 |
| 2.4 Analysis Models..... | 6 |
| 2.5 System Features..... | 7 |
| 2.6 Other Nonfunctional Requirements..... | 7 |
| 2.6 Other Requirements | 8 |
| 2.7 Glossary | 9 |
| 2.8 Requirement Traceability Matrix | 10 |
| 3. Project Plan | 11 |
| 3.1 Lifecycle | 11 |
| 3.2 Tools..... | 12 |
| 3.3 Work Breakdown Structure | 13 |
| 3.4 Effort Estimation | 13 |
| 3.5 Gantt Charts | 14 |
| 3.6 Sprints | 14 |
| 3.7 Product Backlog | 16 |
| 4. Design Diagrams | 17 |
| 5. Test Cases | 18 |
| 6. Screen Shots of Output | 19 |

Project Proposal

TITLE : Stock Market Predictor

GITHUB: <https://github.com/Mitul-Joby/Stonkinator>

Proposed Project Description

Stock market prediction and analysis are some of the most difficult jobs to complete. There are numerous causes for this, including market volatility and a variety of other dependent and independent variables that influence the value of a certain stock in the market. These variables make it extremely difficult for any stock market expert to anticipate the rise and fall of the market with great precision.

However, with the introduction of Machine Learning and its strong algorithms, the most recent market research and Stock Market Prediction advancements have begun to include such approaches in analyzing stock market data.

This article will walk through a simple implementation of analyzing and forecasting the stock prices of a Popular Worldwide Online Retail Store in Python using various Machine Learning Algorithms.

Clientele

The project's machine learning algorithms can be widely utilized by many organizations in Stock market prediction.

General public interested in the Stock market can make great use of this tool to aid them in their investment plans.

Plan Of Work

| | |
|--|----------|
| Login Page and Register Page | - Mihir |
| Predict Page, Model construction, backend server | - Mitul |
| Predictions Page, Model training, testing | - Navyae |
| Home Page and Land Page | - Noel |

Software Requirement Specifications

Introduction

Product Name: Stock Price Predictor

Version: 1.0

This document goes through the specifications of both the front end and the back end of the product.

Purpose

The stock market is a collection of buyers and sellers of stocks/shares, which represent ownership claims on businesses. Investments are usually made with an investment strategy in mind.

The stock market is always fluctuating and is known for being volatile, dynamic, and nonlinear. Accurate stock price prediction is challenging because of multiple external factors such as socioeconomic conditions, inflation, exchange rates, politics unexpected events, a company's financial performance, and so on.

However, this also means that there's a lot of data to find patterns and make models and predictions. These models don't even need to reach high levels of accuracy because even 60% accuracy can deliver solid returns. Thus, with the help of ML techniques, one can detect stock market trends.

Intended Audience and Reading Suggestions

This product is aimed to predict trends in the stock market and make predictions. The entire idea of predicting stock prices is to gain significant profits for the users.

It is intended that the document may be read by developers, project managers, marketing staff, users, testers, and documentation writers.

The remainder of the document contains all descriptions of product functions with functionality and basic Interface requirements. Readers are expected to read the stated references for a greater technical understanding.

DISCLAIMER

The Content is for informational purposes only, you should not construe any such information or other material as legal, tax, investment, financial, or other advice. This document, including but not limited to text, images, videos, spreadsheets, hyperlinks, and computer source code; is written and created solely for educational purposes and does not constitute a solicitation, recommendation, endorsement, to buy or sell any securities or other financial instruments in this or in any other jurisdiction in which such solicitation or offer would be unlawful under the securities laws of such jurisdiction.

Product Scope

This product aims to predict trends in the stock market and allow users to gain significant profits for such as predictions. The product hopes to support cross-platform as web-based clients and terminal clients.

The product shall analyze quantitative information that could identify trading signals and capture the movement patterns of the stock market.

The product hopes to be a relevant indicator for long-term investment. The fundamental analysis relies on both historical and present data to measure profits/losses.

References

Schwert, G.W., 1989. Why does stock market volatility change over time? The journal of finance, 44(5), pp.1115-1153.

Yadav, A., Jha, C.K. and Sharan, A., 2020. Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science, 167, pp.2091-2100.

Vanukuru, Kranthi. (2018). Stock Market Prediction Using Machine Learning. 10.13140/RG.2.2.12300.77448.

Overall Description

Product Perspective

Stock price analysis has been a critical area of research and is one of the top applications of machine learning. A stock market is a public market where you can buy and sell shares for publicly listed companies. The stocks, also known as equities, represent ownership in the company. The stock exchange is the mediator that allows the buying and selling of shares.

Stock market prediction using machine learning means the ability of the software to predict the future stock prices based on past events. It takes time series and based on patterns and trends; it predicts the future price of a particular stock of a particular company. This not only helps companies to raise capital but also helps in raising personal wealth.

Product Functions

Stock market prediction and analysis are some of the most difficult jobs to complete. There are numerous causes for this, including market volatility and a variety of other dependent and independent variables that influence the value of a certain stock in the market. These variables make it extremely difficult for any stock market expert to anticipate the rise and fall of the market with great precision.

However, with the introduction of Machine Learning and its strong algorithms, the most recent market research and Stock Market Prediction advancements have begun to include such approaches in analyzing stock market data.

The main objective of this project is to do the following:

- Although predicting the stock market is difficult since it is volatile, however with the help of machine learning, it is possible to help investors to make better decisions
- It gives investors a new perspective into the current trends in the stock market
- It also gives investors a new perspective into patterns of the stock market based on historical data (long term data)
- It helps in correct prediction of a stock; a correct prediction of stocks can lead to huge profits for the seller and the broker. Frequently, it is brought out that prediction is chaotic rather than random, which means it can be predicted by carefully analyzing the history of respective stock market.

User Classes and Characteristics

Our target audience as mentioned above is for anyone and everyone that surfs through the internet, who are looking to gain profits from the predictions.

There will also be an Admin class to manage the product who will have majority permissions.

Operating Environment

The product can operate in:

- Web Browser
- Command Line

The product is designed this way to make it accessible to almost anyone with the knowledge of surfing the internet.

Design and Implementation Constraints

- The limitation of this project is that it only considers the closing price of the stock and no other attributes such as opening price, high and low price and volume.
- The stock market is very volatile and it is hard to predict even with a very good machine learning algorithm.
- The stock market is very noisy and it might throw some unexpected results.

DISCLAIMER

The Content is for informational purposes only, you should not construe any such information or other material as legal, tax, investment, financial, or other advice. This document, including but not limited to text, images, videos, spreadsheets, hyperlinks, and computer source code; is written and created solely for educational purposes and does not constitute a solicitation, recommendation, endorsement, to buy or sell any securities or other financial instruments in this or in any other jurisdiction in which such solicitation or offer would be unlawful under the securities laws of such jurisdiction.

2.6 Assumptions and Dependencies

- Training data is either collected from past stock values.
- All users have the basic knowledge of the stock market.
- It is assumed the system is equipped with a GNU.
- It is assumed that there won't be any major noises or anything extremely unpredictable.

External Interface Requirements

User Interfaces

The user will be able to select the stock on the home screen that they want to predict using our product, upon the selection another page opens showing the projected graph inferred using the stock's past trends. Simple buttons like selection and enter are going to be used. Error messages show if not enough data is available on the stock or the wrong stock name is entered.

Software Interfaces

We use python to code the network model, which will be running on a server. We will be using various libraries such as:

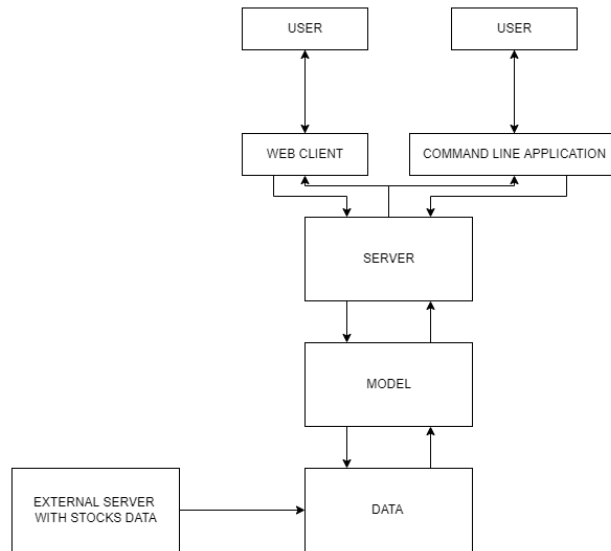
- Math
- Pandas_datareader
- NumPy
- Pandas
- Sklearn.preprocessing
- Keras.models
- Keras.layers
- Matplotlib.pyplot

We plan on using React JS along with redux for the web browser clients. We will also be using MongoDB as our database for storing user information, sessions, and predictions.

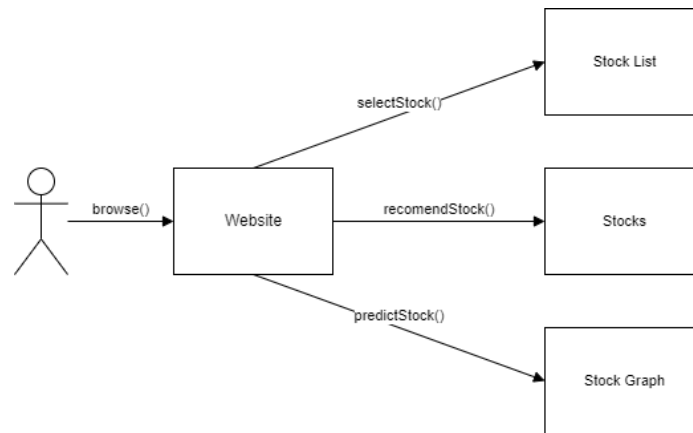
Communications Interfaces

Consists of a webpage which allows the user to access the predictions and will be using HTTP. The product will require an FTP connection as all computations will happen on a server. A command prompt, and a Python IDE downloaded version 3.8 above for development. set up of a working monitor for display and keyboard for user input. No cross-platform implementation.

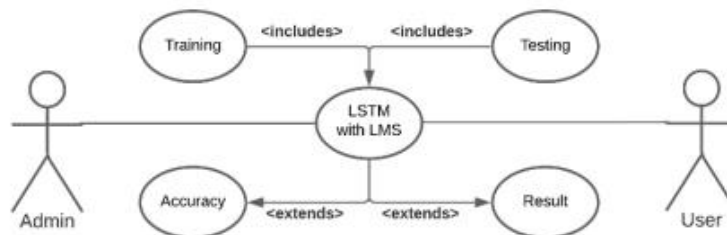
Analysis Model



UML Diagrams



Website



LSTM Model Predictor

System Features

5.1.1 Description and Priority

System is implemented using python language

- Data is first scraped and then put into a data frame
- The data is then scaled to normalize it using MinMaxScaler()
- Create the training dataset
- Train the dataset
- Get the models predicted value
- The data is then plotted

5.1.2 Stimulus/Response Sequences

- User enters the stock to be predicted
- The algorithm looks for past trends
- Our product displays the projected graph

5.1.3 Functional Requirements

FR1: Landing page- The user should be on a responsive and professional looking landing page. The website should be responsive and mobile compatible.

FR2: Login Page- The user should be able to login and only authorized users should be able to predict the price

FR3: Choosing stock- The user should be able to choose the stock they want to predict from the list given

FR4: Prediction page- The software must perform pre-processing on input for model training, such as checking for missing data values. It uses LSTM. It analyses the input data by generating the closing stock price that is most likely to occur.

FR5: Past Prediction page- The software stores user's predictions with stock along with the date it was predicted along with the 5 days predicted value and the then module's RMSE or accuracy value.

Other Non-functional Requirements

Performance Requirements

NFR1: It is expected that the time taken to analyze the stock market does not exceed an appropriate time limit.

NFR2: The system is expected to give results within an acceptable error margin so it does not affect the investors and lead them to take a wrong decision.

Safety Requirements

NFR3: It is expected that in order to make the system safe and reliable, the data chosen and the preprocessing that takes place, happens correctly.

NFR4: It is expected that the machine learning algorithm that is implemented is error free and handle cases within a certain tolerance.

NFR5: Users should use the product with caution because of the volatile nature of the stock market.

Security Requirements

The Content is for informational purposes only, you should not construe any such information or other material as legal, tax, investment, financial, or other advice. This document, including but not limited to text, images, videos, spreadsheets, hyperlinks, and computer source code; is written and created solely for educational purposes and does not constitute a solicitation, recommendation, endorsement, to buy or sell any securities or other financial instruments in this or in any other jurisdiction in which such solicitation or offer would be unlawful under the securities laws of such jurisdiction.

Software Quality Attributes

- It is expected that time taken to make the predictions does not take too long.
- The system should be adaptable and must be able to react to new trends and patterns in the stock market.
- The model should work for various stocks.
- The prediction made by the machine learning algorithm should be reliable.

Business Rules

Taxes will be applicable on the transaction of selling a share. Hopes to be profitable, successful, high-yielding, gainful, remunerative for the user.

Other Requirements

- An in-depth knowledge of the stock market
- Risk Assessment
- Natural ability to make good judgements
- Decide a time horizon before buying a stock
- Be responsible and spend money wisely

Appendix A: Glossary

| | |
|-------------|--------------------------------------|
| ML | - Machine Learning |
| LSTM | - Long short-term memory |
| UML | - Unified Modelling Language |
| FTP | - File Transfer Protocol |
| HTTP | - Hypertext Transfer Protocol |
| IDE | - Integrated Development Environment |

Appendix B: Requirement Traceability Matrix

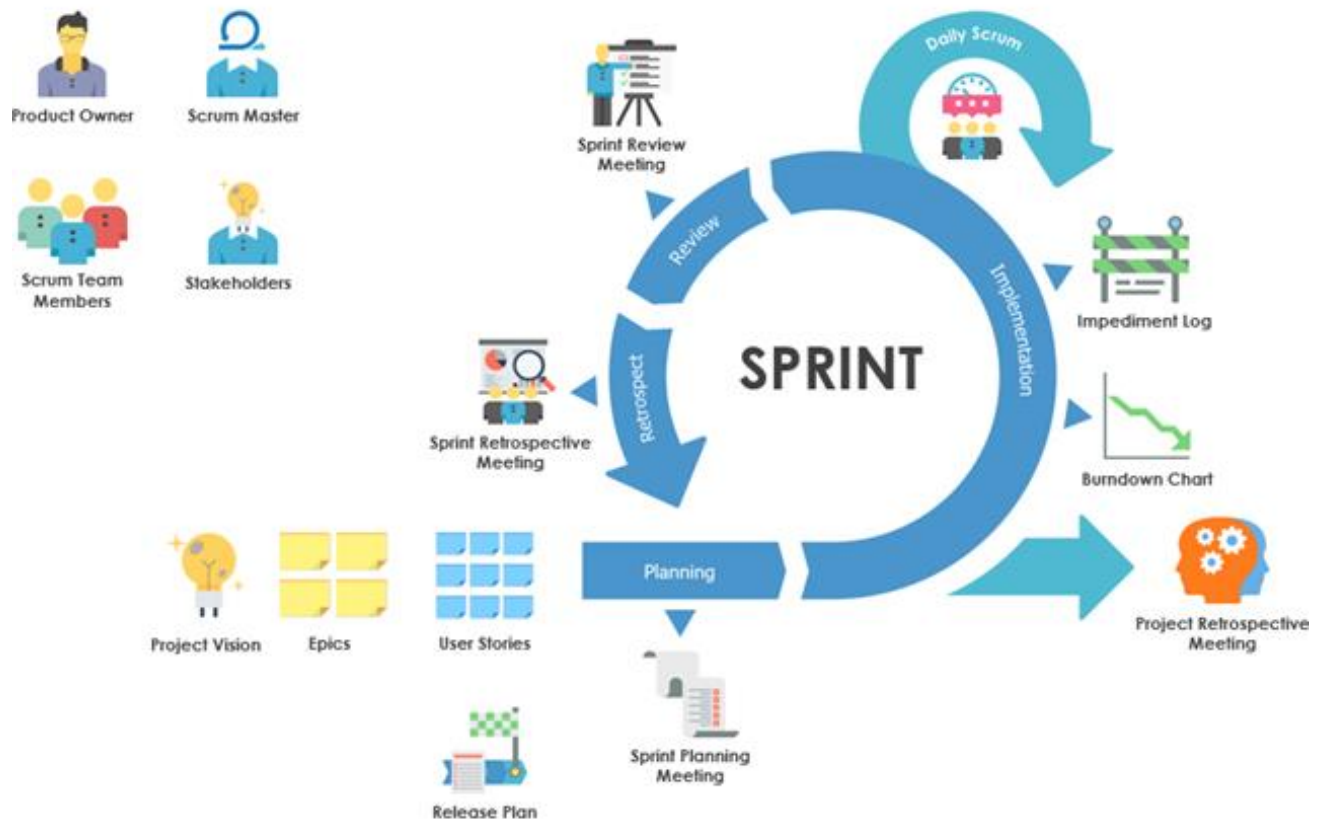
| SL NO | REQUIREMENT ID | BRIEF DESCRIPTION OF REQUIREMENT | ARCHITECTURE REFERENCE | DESIGN REFERENCE | CODE FILE REFERENCE | TEST CASE ID | SYSTEM TEST CASE ID |
|-------|----------------|----------------------------------|------------------------------|------------------|---------------------|--------------|---------------------|
| 1 | FR01 | Landing Page | Static Site Generation (SSG) | Self | - | UT01 | UT01 |
| 2 | FR02 | Login Page | Static Site Generation (SSG) | Self | - | UT02 | UT02 |
| 3 | FR03 | Search for Stock Page | Static Site Generation (SSG) | Self | - | UT03 | UT03 |
| 4 | FR04 | Prediction Page | LSTM | Self | - | UT04 | UT04 |
| 5 | FR05 | Past predictions page | Static Site Generation (SSG) | Self | - | UT05 | UT05 |

PROJECT PLAN

1. Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

The life cycle followed for the execution of our project is the Agile Scrum. This methodology involves multiple iterations of certain steps and a daily scrum to update each other on the progress.

These iterations are the sprints and are pre decided before starting the project. We follow a continuous deliverable format which verifies tasks for the day and if the sprint completion is up to date.



2. Identify the tools which you want to use throughout the life cycle like planning tool, design tool, version control, development tool, bug tracking and testing.

Planning tool: JIRA

Design Tools: PowerPoint, Canva

Version Control: git

Development Tool: VS Code, Python

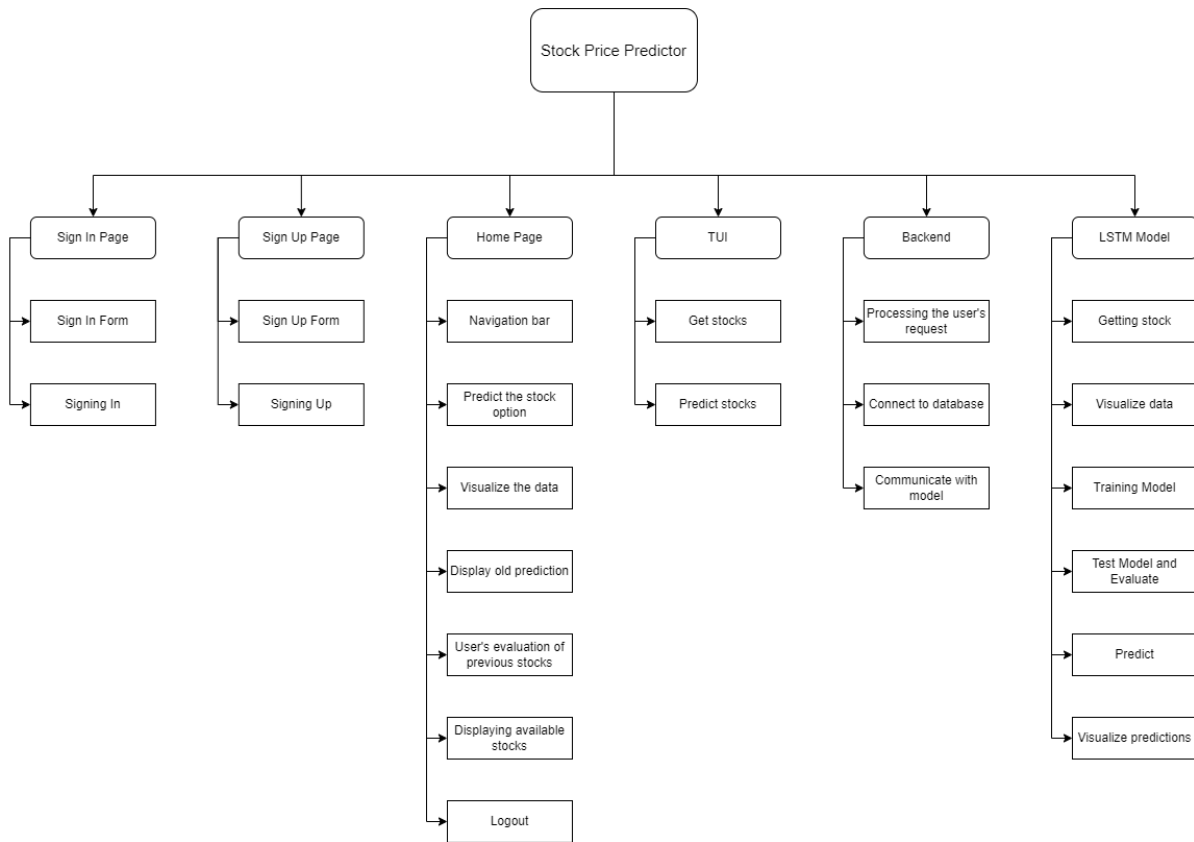
Bug Tracking: JIRA

Testing: Selenium

3. Determine all the deliverables and categorize them as reuse/build components and justify the same

- Login Page, Register Page: Reuse
There are many existing login page components that we can use and change to our requirements.
- Update the databases with companies' stock: Build
- Try to add as many companies' stock prices to the database: Build
A simple process of adding data to an existing database and connecting them
- Generate plots for historical stock prices: Build
Deliverable that takes care of generating historical data such as closing price, opening price, high and low and then based on that prints the stock prices.
- Generate future prices of the stock: Build
Generating the future prices of the stock using machine learning.

4. Create a WBS for the entire functionalities in detail.



Rough estimate of effort required

Organic models are used when the problem statement is well understood and have been used in the past, and works well with small teams. The module selected for testing was the LSTM model.

This model was selected because it is the core module and is a dependency for other modules.

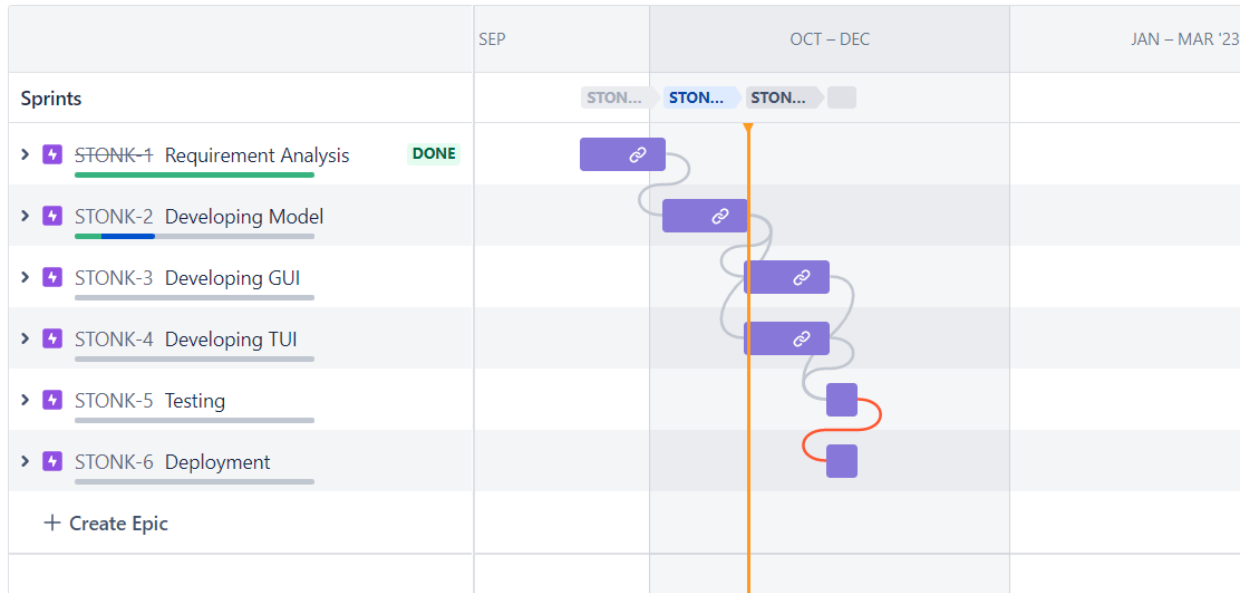
KLOC = 2

a = 2.4

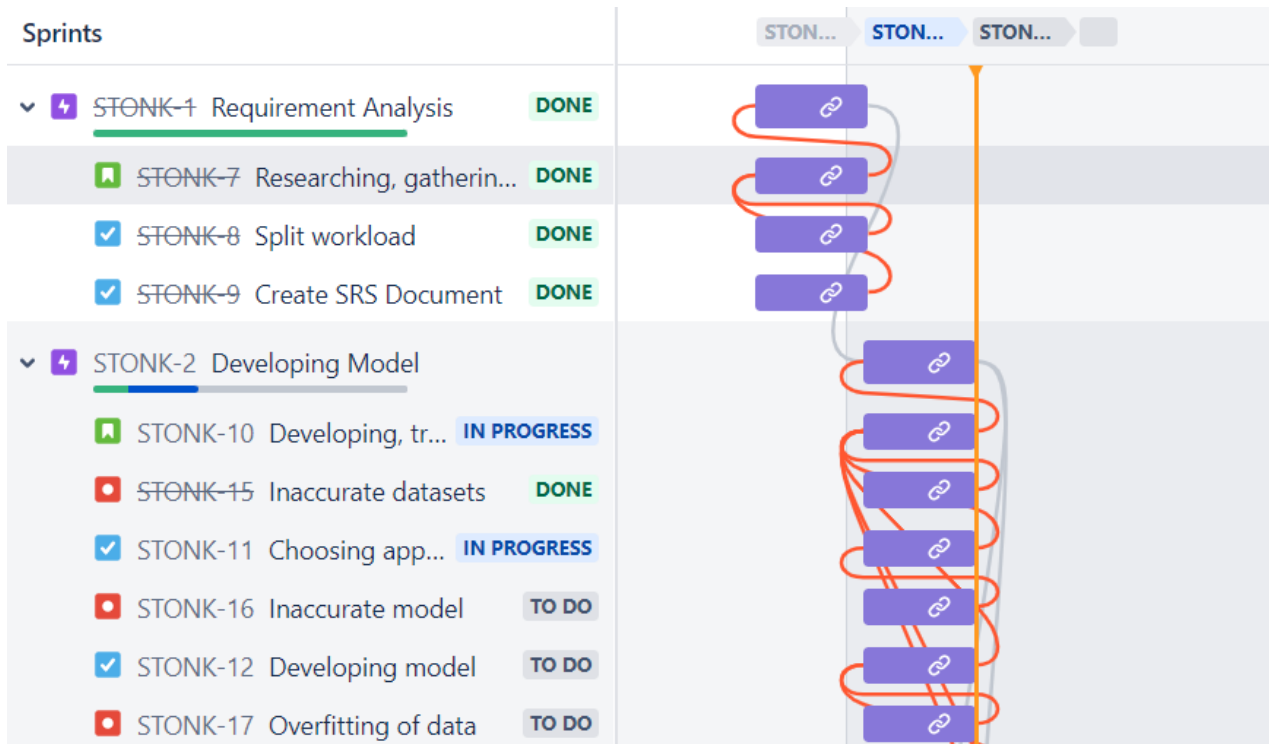
b = 1.05

Effort = $a \cdot (KLOC)^b = 4.96$

Gantt chart



Sprints



| | | | | | |
|--|--|--|--|--|--|
| <ul style="list-style-type: none"> ✓ STONK-12 Developing model TO DO ✗ STONK-17 Overfitting of data TO DO ✗ STONK-18 Slow Implementation TO DO ✓ STONK-13 Training the model TO DO ✓ STONK-14 Testing the model TO DO | | | | | |
| <ul style="list-style-type: none"> ✓ STONK-3 Developing GUI <ul style="list-style-type: none"> ✗ STONK-19 Create Frontend Ap... TO DO ✓ STONK-20 Create components TO DO ✗ STONK-23 Browser compatibili... TO DO ✗ STONK-25 Input form not work... TO DO ✓ STONK-21 Connect to model A... TO DO ✓ STONK-22 Create responsive UI TO DO ✗ STONK-24 Improve UI TO DO | | | | | |
| <ul style="list-style-type: none"> ✓ STONK-4 Developing TUI <ul style="list-style-type: none"> ✗ STONK-26 Create terminal app... TO DO ✓ STONK-27 Connect to backend... TO DO ✗ STONK-29 Server connection is... TO DO ✓ STONK-28 Optimize performan... TO DO | | | | | |
| <ul style="list-style-type: none"> ✓ STONK-5 Testing <ul style="list-style-type: none"> ✗ STONK-30 Testing and integrat... TO DO ✓ STONK-31 Seamless integration TO DO | | | | | |
| <ul style="list-style-type: none"> ✓ STONK-6 Deployment <ul style="list-style-type: none"> ✗ STONK-32 Deploying and final ... TO DO ✓ STONK-33 Deploy to a server TO DO ✗ STONK-34 Did not deploy pro... TO DO | | | | | |

Product Backlog

▼ STONK Sprint 2 4 Oct – 25 Oct (9 issues) 0 0 0 Complete sprint ...

| | | | | |
|----------|--|------------------|-------------|--|
| STONK-10 | Developing, training and testing the model | DEVELOPING MODEL | IN PROGRESS | |
| STONK-15 | Inaccurate datasets | DEVELOPING MODEL | DONE | |
| STONK-11 | Choosing appropriate model | DEVELOPING MODEL | IN PROGRESS | |
| STONK-16 | Inaccurate model | DEVELOPING MODEL | TO DO | |
| STONK-12 | Developing model | DEVELOPING MODEL | TO DO | |
| STONK-17 | Overfitting of data | DEVELOPING MODEL | TO DO | |
| STONK-18 | Slow Implementation | DEVELOPING MODEL | TO DO | |
| STONK-13 | Training the model | DEVELOPING MODEL | TO DO | |
| STONK-14 | Testing the model | DEVELOPING MODEL | TO DO | |

+ Create issue

▼ STONK Sprint 3 25 Oct – 15 Nov (11 issues) 0 0 0 Start sprint ...

| | | | | |
|----------|------------------------------|----------------|-------|--|
| STONK-19 | Create Frontend Application | DEVELOPING GUI | TO DO | |
| STONK-20 | Create components | DEVELOPING GUI | TO DO | |
| STONK-23 | Browser compatibility issues | DEVELOPING GUI | TO DO | |
| STONK-25 | Input form not working | DEVELOPING GUI | TO DO | |
| STONK-21 | Connect to model API | DEVELOPING GUI | TO DO | |
| STONK-22 | Create responsive UI | DEVELOPING GUI | TO DO | |
| STONK-24 | Improve UI | DEVELOPING GUI | TO DO | |
| STONK-26 | Create terminal application | DEVELOPING TUI | TO DO | |
| STONK-27 | Connect to backend server | DEVELOPING TUI | TO DO | |
| STONK-29 | Server connection issues | DEVELOPING TUI | TO DO | |
| STONK-28 | Optimize performance | DEVELOPING TUI | TO DO | |

+ Create issue

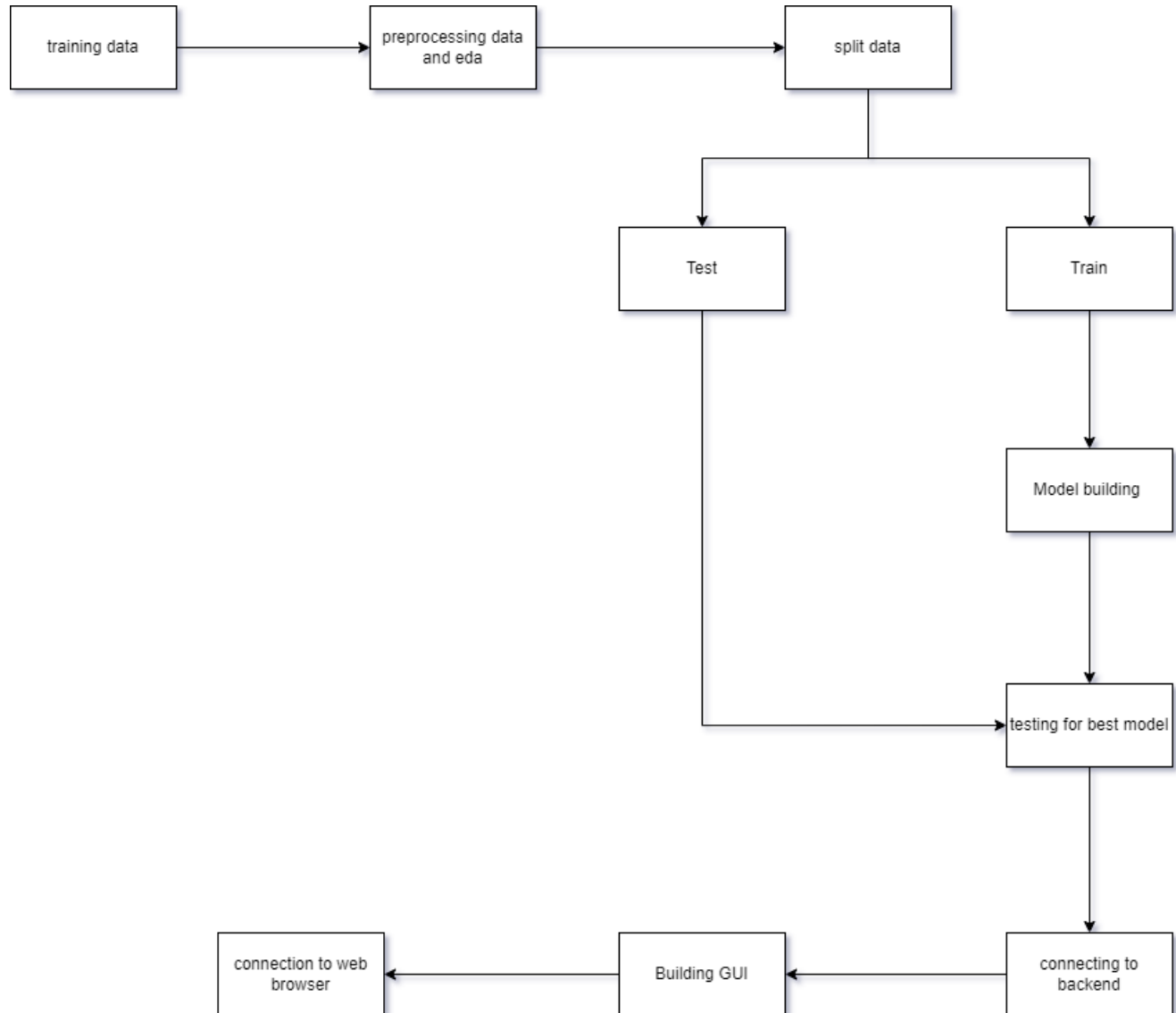
Quickstart

▼ STONK Sprint 4 15 Nov – 22 Nov (5 issues) 0 0 0 Start sprint ...

| | | | | |
|----------|----------------------------|------------|-------|--|
| STONK-30 | Testing and integrating | TESTING | TO DO | |
| STONK-31 | Seamless integration | TESTING | TO DO | |
| STONK-32 | Deploying and final checks | DEPLOYMENT | TO DO | |
| STONK-33 | Deploy to a server | DEPLOYMENT | TO DO | |
| STONK-34 | Did not deploy properly | DEPLOYMENT | TO DO | |

+ Create issue

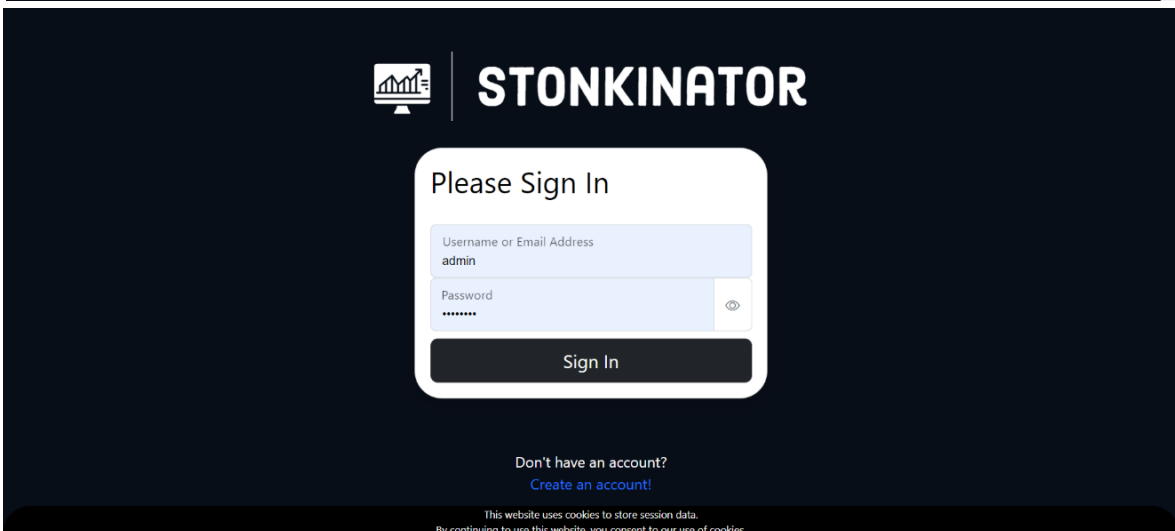
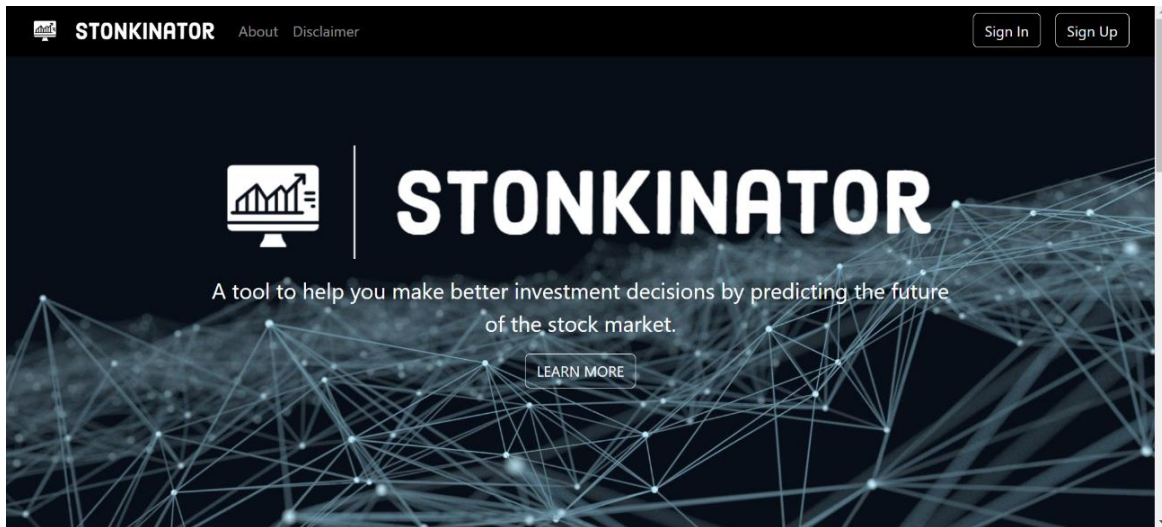
DESIGN DIAGRAM



TEST PLAN DOCUMENT

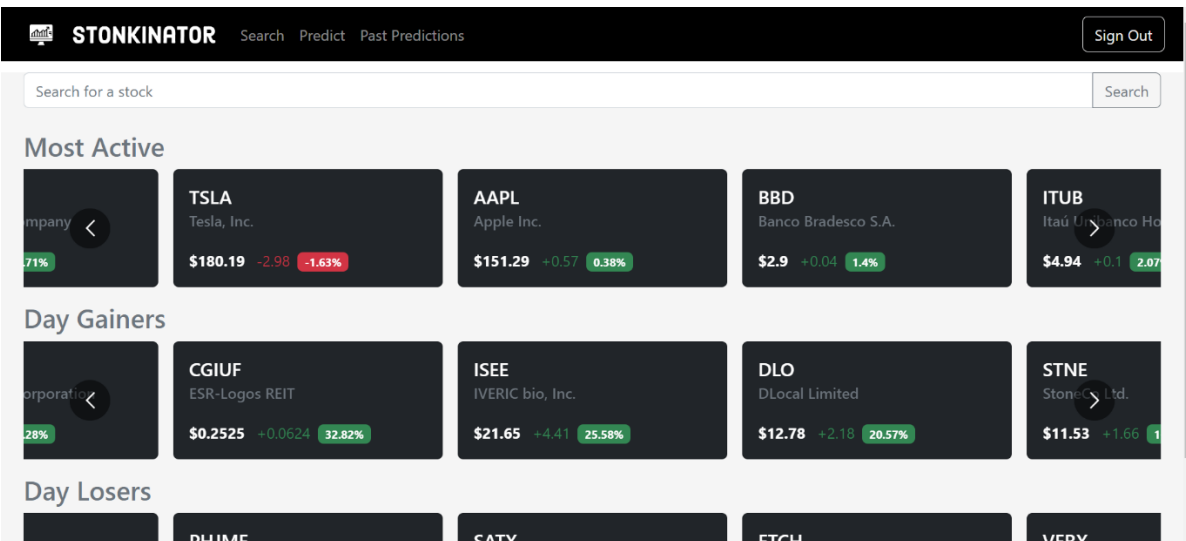
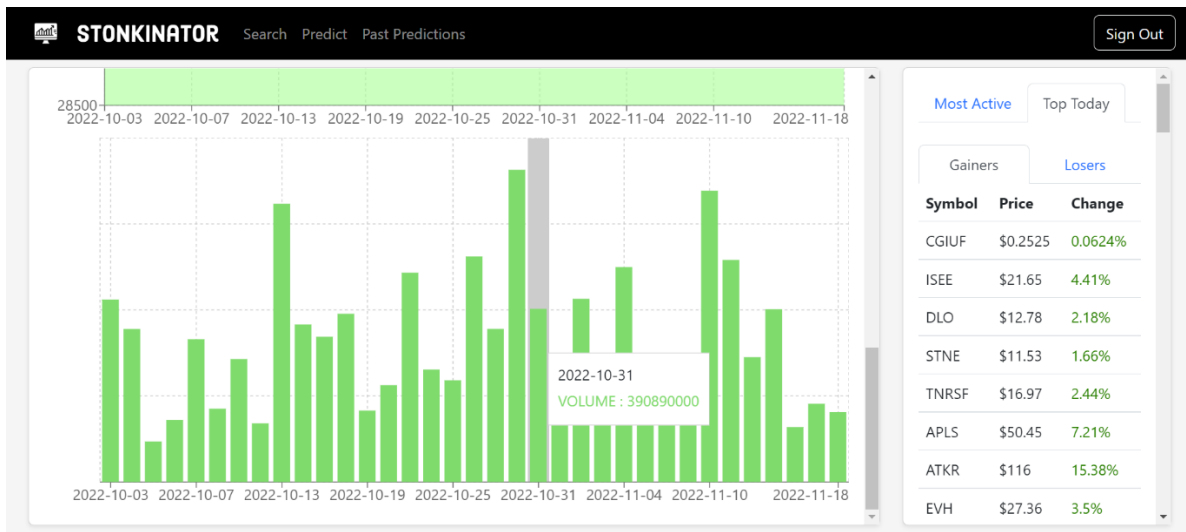
| TEST CASE ID | NAME OF MODULE | TEST CASE DESCRIPTION | PRE-CONDITIONS | TEST STEPS | TEST DATA | EXPECTED RESULTS | ACTUAL RESULT | TEST RESULT |
|--------------|---------------------------------------|--|---|--|--|--|---|-------------|
| UT-01 | Landing Page | Testing different browser compatibilities. | Access to browser. | 1.Navigate to website. 2.Resize browser to different sizes with developer tools and pre-set mobile size configurations. | | Render all components and scale to various browser sizes. | Rendered all components and scaled to various browser sizes as expected. | PASSED |
| UT-02 | User registration module | To test the signing up functionality | Access to browser. | 1.Navigate to sign up. 2.Enter Name, email, Username and Password 3.Click Submit | Name: Admin Email: admin@email.com Username: admin Password: admin | Should reject password and say it is too short. | Rejected password and said it was too short. | PASSED |
| UT-03 | User registration module | To test the signing up functionality | Access to browser. | 1.Navigate to sign up. 2. Enter Name, email, Username and Password 3.Click Submit | Name: Admin Email: admin@email.com Username: admin Password: df23hjka78 | Tell user account has been created and redirect to sign in page. | Said user account had been created and redirected to sign in page. | PASSED |
| UT-04 | User login module | To test the signing in functionality | Access to browser. Registered account on website. | 1.Navigate to sign in. 2: Enter Username and Password 3: Click Submit | Username: admin Password: df23hjka78 | Sign in to website, show client homepage. | Signed in to website and showed client homepage. | PASSED |
| UT-05 | Searching and choosing a stock module | To test the search feature functionality. | Access to browser. Registered account on website. Signed in. | 1.Navigate to search page 2.Search for a stock. | Stock ticker: TSLA | Show graph of past 50 days of Tesla and information about the stock. | Showed graph of past 50 days of Tesla and information about the stock. | PASSED |
| UT-06 | Searching and choosing a stock module | To test the search feature functionality. | Access to browser. Registered account on website. Signed in. | 1. Navigate to search page 2.Search for a stock. | Stock ticker: NOTREALSTOCK | Say that stock searched for is invalid. | Said that stock searched for is invalid. | PASSED |
| UT-07 | Predicting a stock module | To test the predict feature functionality. | Access to browser. Registered account on website. Signed in. | 1.Navigate to predict page 2.Predict a stock. | Stock ticker: TSLA | Show graph of past 50 days of Tesla along with next 5 days prediction along with model accuracy and information about the stock. | Showed graph of past 50 days of Tesla along with next 5 days prediction along with the model accuracy and info about the stock. | PASSED |
| UT-08 | Past predictions module | To test the past predictions, feature functionality. | Access to browser. Registered account on website. Signed in. Already predicted some valid stocks. | 1.Navigate to past predictions page. | | Show list of stocks along with days on which it was predicted along with predictions. | Showed list of stocks along with days on which it was predicted along with predictions. | PASSED |

SCREENSHOTS



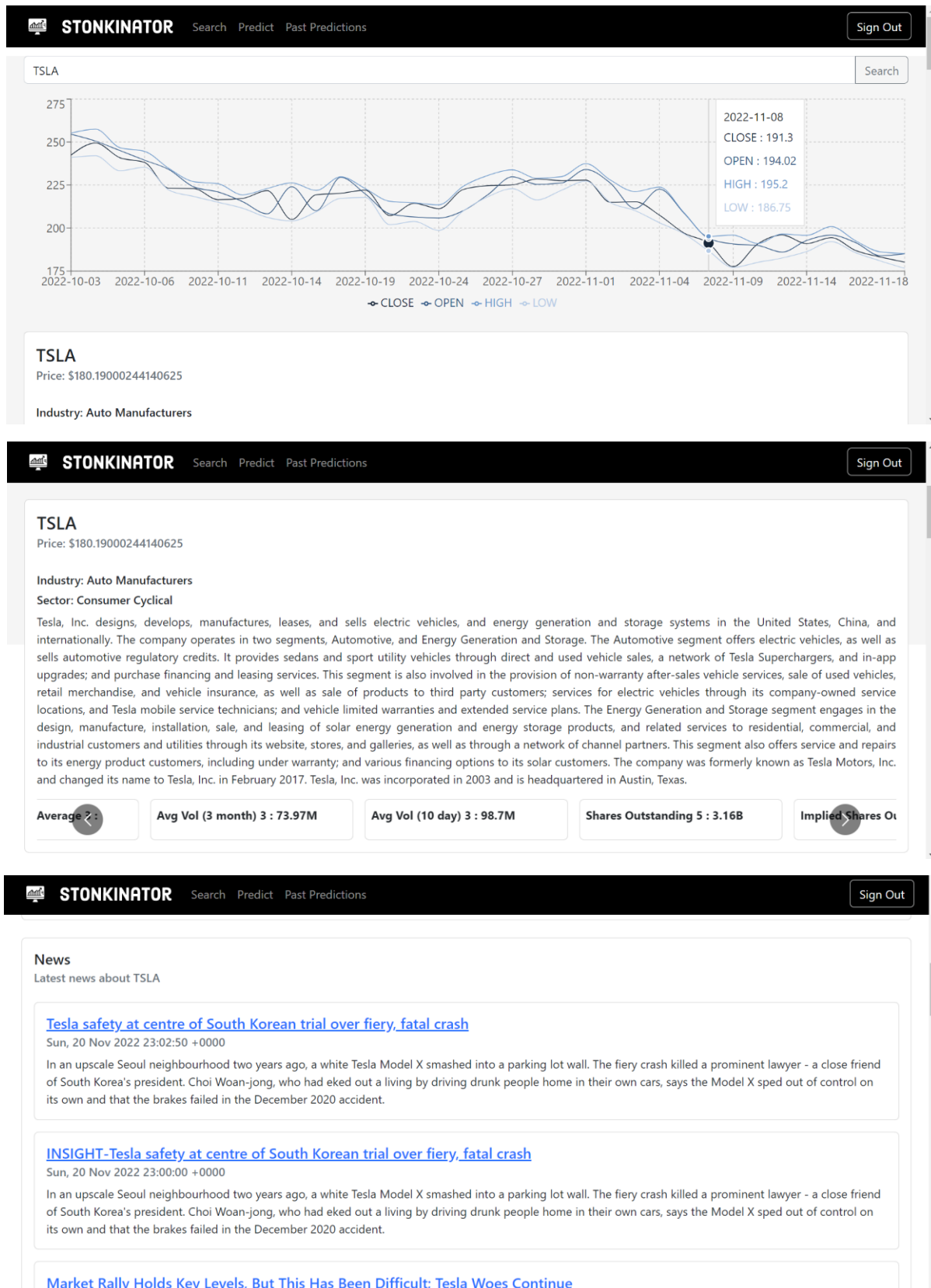
PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering



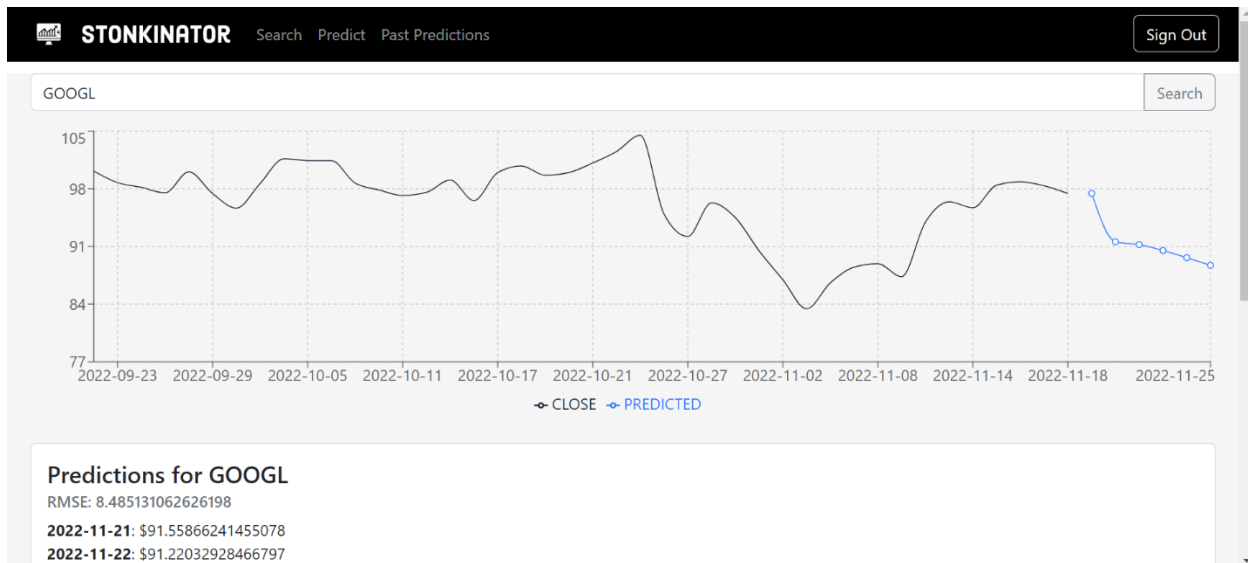
PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering



STONKINATOR Search Predict Past Predictions Sign Out

Predictions for GOOGL

RMSE: 8.485131062626198

2022-11-21: \$91.55866241455078

2022-11-22: \$91.22032928466797

2022-11-23: \$90.50595092773438

2022-11-24: \$89.6326675415039

2022-11-25: \$88.70360565185547

GOOGL

Price: \$97.43000030517578

Industry: Internet Content & Information

Sector: Communication Services

Alphabet Inc. provides various products and platforms in the United States, Europe, the Middle East, Africa, the Asia-Pacific, Canada, and Latin America. It operates through Google Services, Google Cloud, and Other Bets segments. The Google Services segment offers products and services, including ads, Android, Chrome, hardware, Gmail, Google Drive, Google Maps, Google Photos, Google Play, Search, and YouTube. It is also involved in the sale of apps and in-app purchases and digital content in the Google Play store; and Fitbit wearable devices, Google Nest home products, Pixel phones, and other devices, as well as in the provision of YouTube non-

STONKINATOR Search Predict Past Predictions Sign Out

Past Predictions

| | | |
|---|---|---|
| <p>TSLA</p> <p>2022-11-21</p> <p>2022-11-21 : 197.26739501953125</p> <p>2022-11-22 : 198.49497985839844</p> <p>2022-11-23 : 201.23809814453125</p> <p>2022-11-24 : 204.82118225097656</p> <p>2022-11-25 : 208.93992614746094</p> <p>RMSE: 21.103382463576153</p> | <p>AAPL</p> <p>2022-11-21</p> <p>2022-11-21 : 144.99212646484375</p> <p>2022-11-22 : 145.0689697265625</p> <p>2022-11-23 : 144.77560424804688</p> <p>2022-11-24 : 144.2884063720703</p> <p>2022-11-25 : 143.7022247314453</p> <p>RMSE: 5.513463910253683</p> | <p>GOOGL</p> <p>2022-11-21</p> <p>2022-11-21 : 91.55866241455078</p> <p>2022-11-22 : 91.22032928466797</p> <p>2022-11-23 : 90.50595092773438</p> <p>2022-11-24 : 89.6326675415039</p> <p>2022-11-25 : 88.70360565185547</p> <p>RMSE: 8.485131062626198</p> |
|---|---|---|