**CPSC531-01 Advanced RDBMS**

**Final Group Project Report**

**Stock Market Analysis**

**By: Group NR01**

**Anitha Chockalingam Student ID : 887350817**

**Abhiruchi Shinde ID : 885964049**

**CPSC 531-01 (13536)**

**Fall, 2021**

**Professor: Dr. Chun-l Phillip Chen**

**Department of Computer Science**

**California State University, Fullerton**

**Dec 9, 2021**

Table of Contents

[Abstract](#_heading=h.gjdgxs) **5**

[Chapter 1: Introduction](#_heading=h.30j0zll) **5**

[Project Requirements](#_heading=h.1fob9te) **5**

[Chapter 3: Methodology](#_heading=h.3znysh7) **7**

[Integrating Robo3T in MongoDB:](#_heading=h.lxokjb552fch) 9

[PyCharm:](#_heading=h.m0jdnvi7f3c5) 11

[**Document creation scripts:**](#_heading=h.9sdsb5w1lcpf) **15**

[Chapter 4: Results and Discussion](#_heading=h.2et92p0) **17**

[Chapter 5: Conclusions](#_heading=h.tyjcwt) **17**

[Reference](#_heading=h.3dy6vkm) **18**

[Appendices](#_heading=h.1t3h5sf) **18**

[Files (Source programs codes, Not Software Libraries)](#_heading=h.4d34og8) **18**

# 

**Table of Figures**

# Abstract

# Over the years, several investors, no matter beginner or experienced, have gained significant profit by planning and investing in stocks. When an investor makes the right and correct decisions, there are higher chances of booking profits from those investments.

# After all, the stock market is truly unpredictable. However, investing in stocks has several major benefits than other investments. Stock analysis helps traders to gain an insight into the economy, stock market, or securities. It involves studying the past and present market data and creating a methodology to choose appropriate stocks for trading. Stock analysis also includes the identification of ways of entry into and exit from the investments. For Analysing current and past data users will need an application which can show stocks price updates. This Project will help users in making better decisions as it provides a platform to check latest stock prices and also provides trends and performance comparison between all these stocks. It has an edge over other stock trading applications or apps as it uses MongoDB as a database to store stock data. MongoDB provides high throughput and performance with real time stock data. it supports ad-hoc querying. It is very flexible and supports all different kinds of data.

# Introduction

# 

# Project Requirements

**Data Requirements :**

There are two different types of documents used in this project which provide information related to stock price and stock company.

**Company** : The company document has the company's name and other significant information of major companies listed on NASDAQ or NYSE stock exchange.

**Stock\_price** :stock price information such as open , close , high and low.

| **Company Data** | | |
| --- | --- | --- |
| **Data Field Name** | **Description** | **Example** |
| \_Id | used to uniquely identify documents default format ObjectID | ObjectId |
| Name | Name of the company | Adobe Inc |
| Symbol | Unique series of letters or numbers used to identify a stock traded on stock exchange | ADBE |
| stock\_exchange | The stock exchange company listed on | XNAS |
| Industry | industry to which company belongs | Software |
| Currency | default currency used | Dollars |

| **Stock\_price data** | | |
| --- | --- | --- |
|  | | |
| **Data Field Name** | **Description** | **Example** |
| \_id | used to uniquely identify documents default format ObjectID | ObjectId |
| Symbol | Unique series of letters or numbers used to identify a stock traded on stock exchange | ADBE |
| Open | Stock open price | 120 |
| High | highest price of the stock for the day | 133 |
| Low | Lowest stock price for the day | 127 |
| Close | stock close price | 125 |
| Date | stock price date | 10/21/2021 |
| Volume | stock volume | 13639399 |
| Change | change in stock price and volume from previous day | 0.79 |
| Company\_id | company\_id a reference from the \_id field in the company document | ObjectId |

**Functional Requirements:**

| **Requirement ID** | **Requirement Statement** | **Must/want** | **Comments** |
| --- | --- | --- | --- |
| **1** | The Web application shall have a homepage that shows the stock price updates for 5 major stocks in tabular format | Must |  |
| **2** | The company symbol displayed on the homepage should have hyperlink | Must |  |
| **3** | The company symbol hyperlink shall redirect the user to the stocks trend page , which will show the company's overall performance over the past 2 weeks in tabular as well as chart format | Must |  |
| **4** | The homepage shall have dashboard button | Must |  |
| **5** | The dashboard page shall redirect user to the performance comparison page | Must |  |
| **6** | This performance comparison page shall show user performance comparison of 5 major stocks over the past 2 weeks in chart format | Must |  |
| **7** | The Web pages shall be refreshed with updated stock prices every 1 hour. | Must |  |
| **8** | Web applications shall use MongoDB to store and retrieve stock data. | Must |  |
| **9** | A MongoDB database should fetch the latest stock data from different sources, and the database must be updated every time there is a change in stock prices. | Must |  |
| **10** | The database should maintain a history of stock data over 2 weeks that can be queried and used by the web application to show the company's performance over a time period. | Must |  |
| **11** | The Application shall have a separate line chart that shows a performance comparison between different companies. | Must |  |
| **12** | The application shall always be up and running. | Want |  |

**Business Requirements**

This web application has three main Business requirements/functions

1. **stock price update :**

Input : stock price data stored in MongoDB documents

Output: Stock price updates in tabular format

Description:This function shows stock price updates on the homepage in tabular format. It fetches the latest information about the company and stocks from the MongoDB database and updates the webpage every one hour. The table shown on the homepage has different attributes such as company name , symbol , Date , Open , high, Low , close , adj close and volume. This function queries MongoDB databases to get the latest stock information and updates the Homepage.

1. **stock trends :**

Input : User click on company symbol displayed on homepage

Output: Stock Performance over 2 weeks

Description: when user clicks on company symbol displayed on homepage. This function will query history data stored in MongoDB related to specific stock and create a chart to show performance of the stock over 2 weeks. it will also show performance information in tabular format on stock trends page.

1. **stock performance comparison :**

Input : User click on Dashboard button on homepage

Output: comparison between stock performance of 5 major stocks over 2 weeks

Description: When a user clicks on the Dashboard button on homepage, this function will query history data for all 5 company stocks and redirect users to dashboard page. on dashboard page stock comparison for these 5 companies will be shown in chart format

# Methodology

This section will outline and briefly describe the various methods and techniques that was used throughout the project. The programming language used in this project is Python, one of the most preferred languages to build backend applications and NoSQL MongoDB for the database.

**Hardware requirements:**

* Minimum System Requirements for Python Installation:
  + Operating Systems and CPU architecture:
  + Windows 7 or 10
  + Mac OS X 10.11 or higher, 64-bit
  + Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
  + x86 64-bit CPU (Intel / AMD architecture)
  + Python v3.9.1 is the first version supporting macOS 11 Big Sur.
  + RAM and free disk space:
  + 4 GB RAM
  + 5 GB free disk space

**Software Requirements and set-up:**

* Python 3.7 recommended
* Pycharm community IDE
* MongoDB database
* Robo3T
* Flask
* PyMongo

**MongoDB:**

It’s a Document-oriented NoSQL database which means each record is not bound by a defined schema (as it would be for a Relational Database Management System). This characteristic enables MongoDB to be scaled-out for handling large volumes for an application such as StockMarketAnalysis which stores daily stocks data for many companies. MongoDB is also flexible to changes in the data structure.

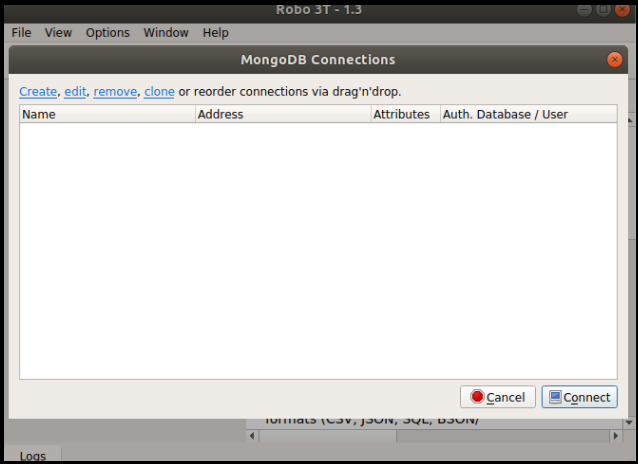
**Robo3t**:

Robo3T is one of several GUI tools that allow you to visualize your data which can be very hard to do through the MongoDB shell. Robo3T allows you to perform all of the primary functionality in a GUI without having to remember the MongoDB shell syntax. Robo3T also provides an embedded shell so users can still use the MongoDB shell if necessary. In the embedded shell Robo3T provides a great autocomplete feature which is invaluable for developers who jump back and forth between different languages and have a hard time remembering the exact syntax. There is a version of Robo3T for Mac OS X, Windows, and Linux systems.

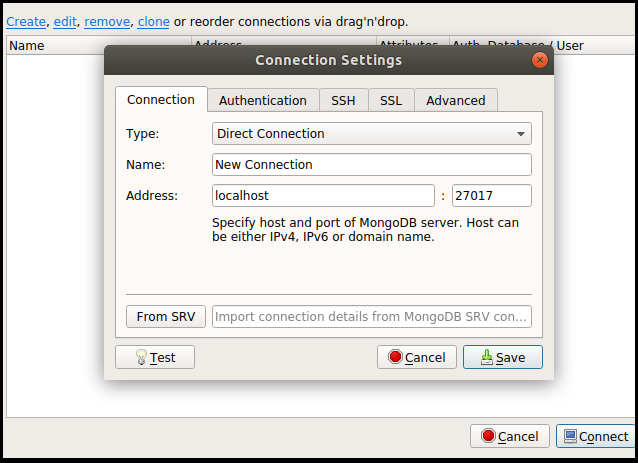
## **Integrating Robo3T in MongoDB:**

Ensure that the MongoDB service is running in the background.

Once the robo3t application is running for the first time, a MongoDB Connections configuration window will prompt the user to Create, edit, remove, clone or “reorder connections via drag’n’drop”.



To create a new MongoDB Connection, click Create menu in upper left corner of the window.



Click Test button at the lower to see if the connection settings is correct. After that, click Save to save the connection settings and click Connect to connect to a MongoDB deployment.

Below is the result of the above procedure.

Graphical user interface, application, Word

Description automatically generated

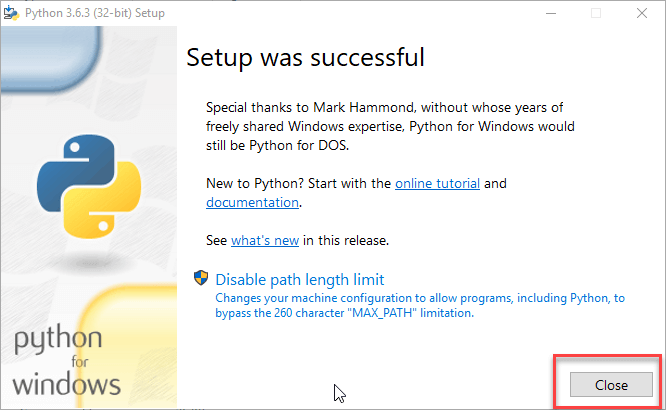
**Python**:

* Download and install Python from the official website of Python <https://www.python.org/downloads/> and choose your version.

Graphical user interface, application, website

Description automatically generated

* Once the download is completed, run the .exe file to install Python. Now click on Install Now.
* When it finishes, you can see a screen that says the Setup was successful. Now click on “Close”.



### **PyCharm:**

PyCharm – a Python IDE that provides a lot of assistance with Python code is used in this project. One of its most impressive aspects is its intelligent code completion. On the back-end side of things, PyCharm’s database integration made it really easy to manage my objects while I was programming the web app initially.

**PyCharm Installation:**

* Download PyCharm from the website  <https://www.jetbrains.com/pycharm/download/> and Click the “DOWNLOAD” link under the Community Section.

Graphical user interface, text, application, chat or text message

Description automatically generated

* Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.
* Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.
* After you click on “Finish,” the Following screen will appear.

Graphical user interface, application, Teams

Description automatically generated

**Flask:**

Flask is a web framework for Python. It provides functionality to building web applications, including managing [HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP#:~:text=Hypertext%20Transfer%20Protocol%20(HTTP)%20is,be%20used%20for%20other%20purposes.) requests. Flask is a simple and easy to start with. By default, flask runs a local server at port 5000.

Within the activated environment, use the following command to install Flask:

$ pip install Flask

**PyMongo:**

There are several *Flask* extensions for integrating *MongoDB with python*, but here we'll be using the *Flask-PyMongo* extension. MongoDB maps well with Object Oriented Programming so the syntax is rather logical, at least for basic query operations. Connecting to a MongoDB is done simply by passing the connection string to the pymongo.MongoClient() constructor. The syntax for PyMongo is very similar to the original MongoDB syntax. To begin using Flask-PyMongo we need to install it with the following command.

$ pip install Flask-PyMongo

The StockAnalysis is a web application that comprises the following components:Diagram

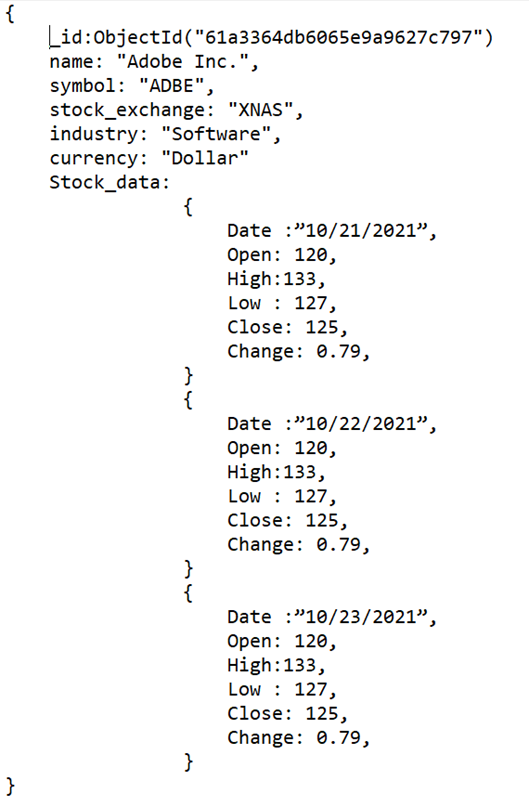
Description automatically generated

**Data Model:**

The StocksDB database has one collection which includes two different types of documents. One that stores company information and the other that stores everyday stock prices. One of the important decisions in the data model design of MongoDB is the document design structure. Documents are similar to rows in the SQL database. Effective data models support the application’s needs. The key consideration for the structure of your documents is the decision to embed or to use references.

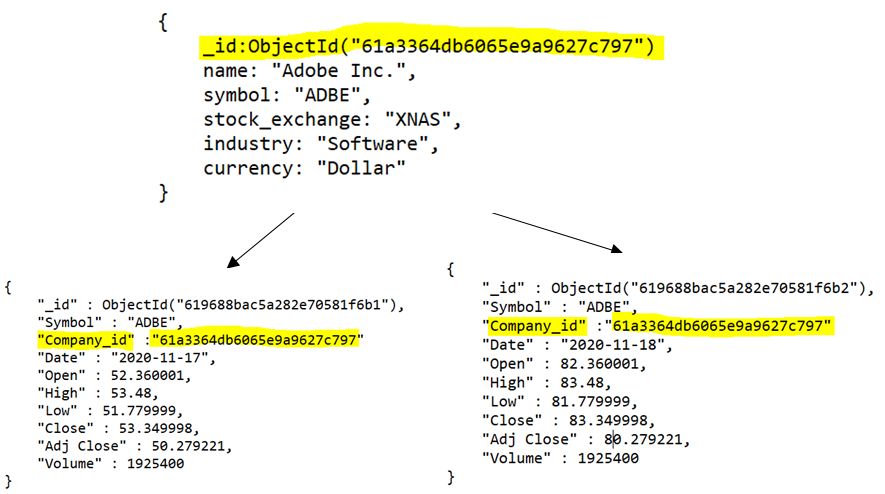
**Embedded Data Models:**

With MongoDB, you may embed related data in a single structure or document. These schema are generally known as "denormalized" models, and take advantage of MongoDB's rich documents. Embedded data models allow applications to store related pieces of information in the same database record. Documents in MongoDB must be smaller than 16MB size. Due to this size limitation and considering the exponential growth nature of data in stocks analysis application, this embedded data models has limitations and may not be suitable for this app. An example of embedded data model for stock data would be as follows:

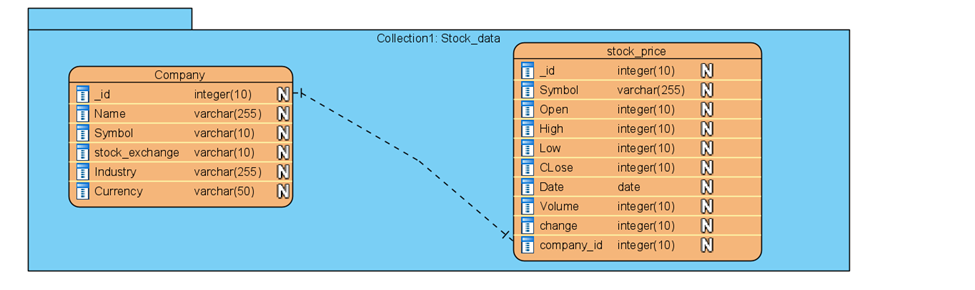


**Normalized Data Model:**

Normalized data models describe relationships using references between documents. An example of normalized data model for stock data would be as follows:



Below is the data model design of stock market analysis app.

****\_id is the autogenerated field by mongoDB that is unique for each document. This field is used as the reference in this data model. Each company has a unique id that is used as reference by the stock price data of that company. The reference id is stored in the company\_id field.

# Document creation scripts:

**Company document creation scripts:**

{ name: "Adobe Inc.",

symbol: "ADBE",

stock\_exchange: "XNAS",

industry: "Software",

currency: "Dollar"

}

{

name: "Dick's Sporting Goods Inc.",

symbol: "DKS",

stock\_exchange: "XNYS",

industry: "Specialty Retail",

currency: "Dollar"

}

{

name: "Tesla Inc.",

symbol: "TSLA",

stock\_exchange: "XNAS",

industry: "Automobile",

currency: "Dollar"

}

{

name: "iRobot Corp.",

symbol: "IRBT",

stock\_exchange: "XNAS",

industry: "Housewares",

currency: "Dollar"

}

{

name: "North Dallas Bank & Trust Co.",

symbol: "NODB",

stock\_exchange: "OOTC",

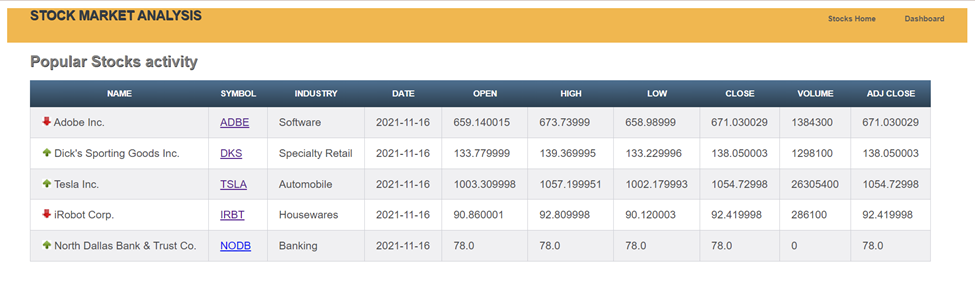
industry: "Banking",

currency: "Dollar"

}

# Chapter 4: Results and Discussion

The StockMarketAnalysis app has 3 main screens – Home page, Dashboard and Stocks trend. The home page displays the latest stock prices for all the stocks that are used in the application. A snapshot of the home screen is shown below.



1. Name: Company name of the stock data. The icon beside the name field, indicates whether the stock price has increased or decreased in comparison over previous day.

2. Symbol : The symbol to represent the stock.

3. Industry: The domain/industry of the company.

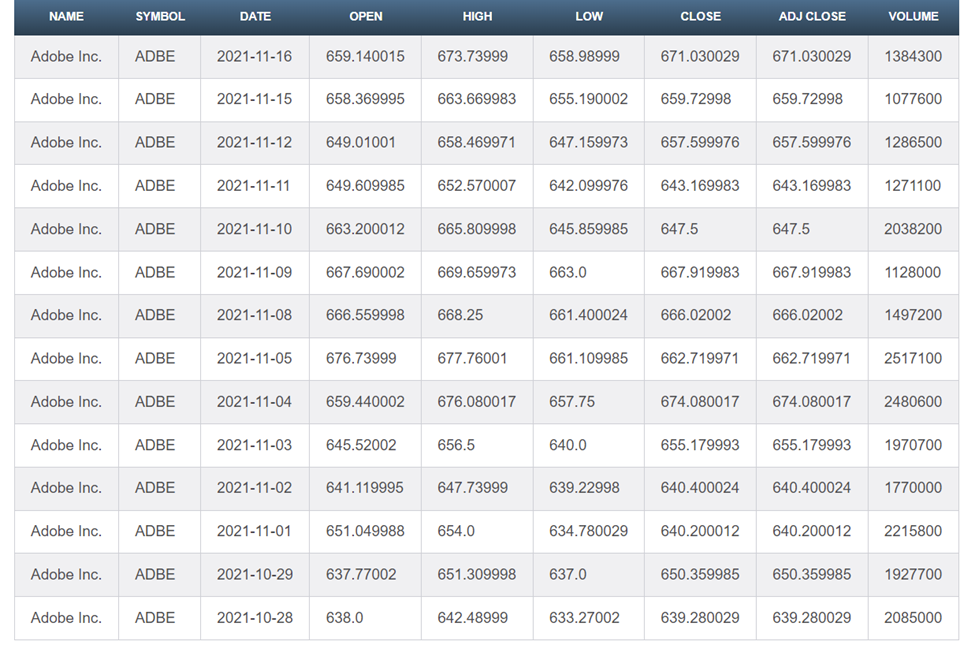
4. Open,High,Low,Close : These represent the stock prices throughout the day.

5. Volume: It is the total number of shares that are actually traded (bought and sold) during the trading day.

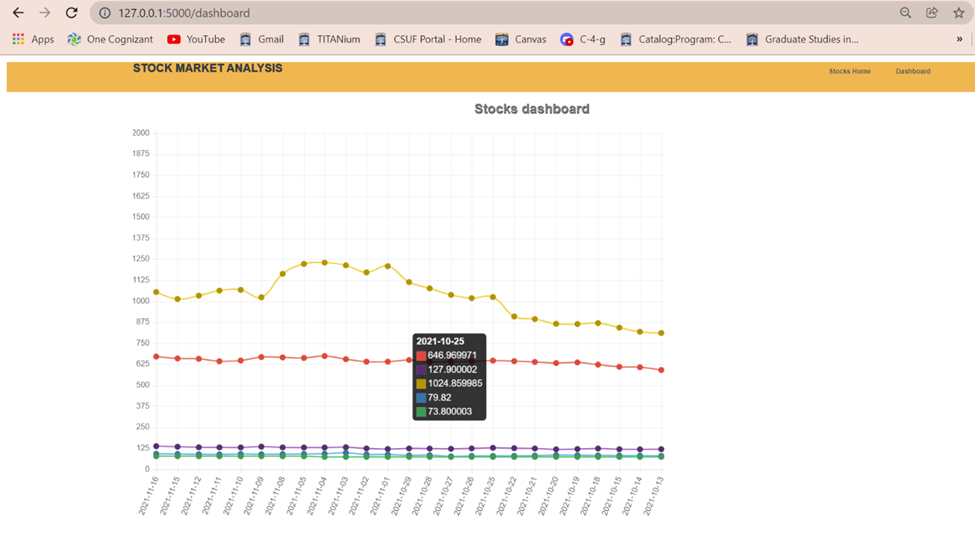
On click on any symbol, the history and trend of that stock data will be shown in the trend chart and in tabular format. The below snapshot shows the trend of the stock data over the period of 2 weeks. On hovering over the data points, it shows the date and exact stock price on that day.



History data of stocks over the period of 2 weeks is also shown in the tabular format as below.



The dashboard page shows a comparison between all the company stocks over the period of 2 weeks. It shows a line trend chart for all 5 stocks in different colors to differentiate them. On hovering over the data points, it shows the exact stock price of all 5 stocks on that particular day.



# Conclusions

In this chapter, interpret, examine, and qualify the results of the investigation and draw inferences from them.

The following topics are intended to serve as a guide:

- Significance of this project

- What have you learned from this project?

- Suggest any improvements for the future design.

- The potential can be used for other applications, where applicable

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# References

1. Stock Market Analysis :Meaning Importance and More <https://cleartax.in/s/stock-market-analysis#:~:text=invest%20in%20Equities%3F-,What%20is%20Stock%20Market%20Analysis%3F,%2Fsector%2Fmarket%20in%20future>.
2. Building a Seccure stock trading app with MongoDB Atlas <https://www.mongodb.com/blog/post/building-a-secure-stock-trading-app-with-mongodb-atlas>
3. MongoDB Real-Time Analytics :A Comprehsive Guide <https://hevodata.com/learn/mongodb-real-time-analytics/>

# Appendices

To show your project originality, Programs source codes must be documented here (Not Program libraries, software tools), test results and reports. Installation procedures and user guides etc., must be

# Files (Source programs codes, Not Software Libraries)