Stock Market Tracking App

Team BUFFER
Dipali Singh
Sankalp Jain

Project Description

In the present times when investors need to be two steps ahead at all time, we need a web-means that can be as fast and efficient. Investors hold back from dabbling in the stock market due to lack of information. Our aim while building this website is to facilitate and assist investors to keep track of the stock market. We want to make a user-friendly website that will allow investors to favourite the stocks they have invested in and track their stocks by analysing the percentage increase or percentage decrease alongside with graphs over a given timeline. The app will be complete with blogs and helpline that will give more insights into the current market status.

Roles and Models

Stakeholder Name	Activity / Area / Phase	Interest	Influence	Priority (High / Medium/Low)
Regional Head of Sales & Marketing	Subscription using mobile App	High	High	1
Finance Account Receivable consultant	Multiple Currency Payment	High	Low	3
Developer	Development of the app	High	High	1
Stock broker	User	High	Low	3

Roles and Models

Stakeholder Name	Activity / Area / Phase	Interest	Influence	Priority (High / Medium/Low)
Customer	User	High	High	2
Supplier	Make resources available	High	High	1
Investors	Provide financial assistance	High	High	1
Parent companies	Monitor the functioning	Low	High	3
Stock broker	User	High	Low	3

Estimation –

Activity Description	Sub - Task Desc.	Efforts in hours	Costs in INR
Design the web app	To create a webapp using python and django	6-12	Frontend- 30k Backend- 30k Integration- 5k
Research and development	Evaluating of software tech trends and incorporating them with updates and patches.	4-8	For Research - 15K For Rolling out patches/ updates - 5K
Data analytics	Administration Data, Database, Management, and evaluating stock market trends.	4-5	Data Analyst- 15k Data scientist- 20k
Identify Data Source for displaying units of Energy Consumption	Go through Interface contract (Application Data Exchange) documents	5	15K
Marketing	Advertisement	8-10	1L

Maintenance and Support Cost

Category	Details	Qty	Cost per annum	Cost per item
People	Network, System, Middleware and DB admin	5	2,000,000	6,000,000
License	Operating System, Database, Middleware, IDE	4	10000	100,000
Infrastructure	Server, Storage and Network	3	20000	400,000

Risk

Risk Identification

Risk identification and categorization:

- **The predictions can be wrong due to inaccuracy in the app.**

- **Sometimes wrong stocks can be shown which may lead to misinformation.**

- **Out of date information can be shown.**

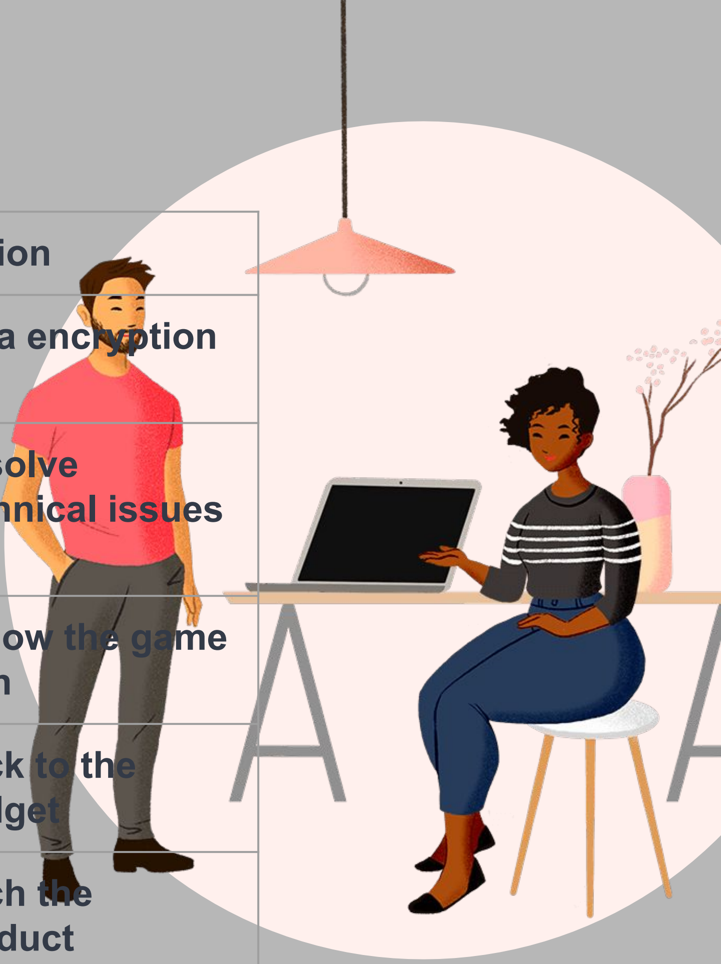
- **Sometimes personalised stocks may be incorrect due to inaccurate predictions.**

List (Describe) Register

Risk ID#	Risk Description	Impact Description
R01	Hacking of application	Has a high impact as it can leak information
R02	Technical difficulties like Data Security.	High impact technical risk
R03	Unable to meet deadlines.	Low impact risk
R04	Insufficiency of the budget	Medium impact risk as it slows down development
R05	Lack of stakeholder engagement	Medium impact risk

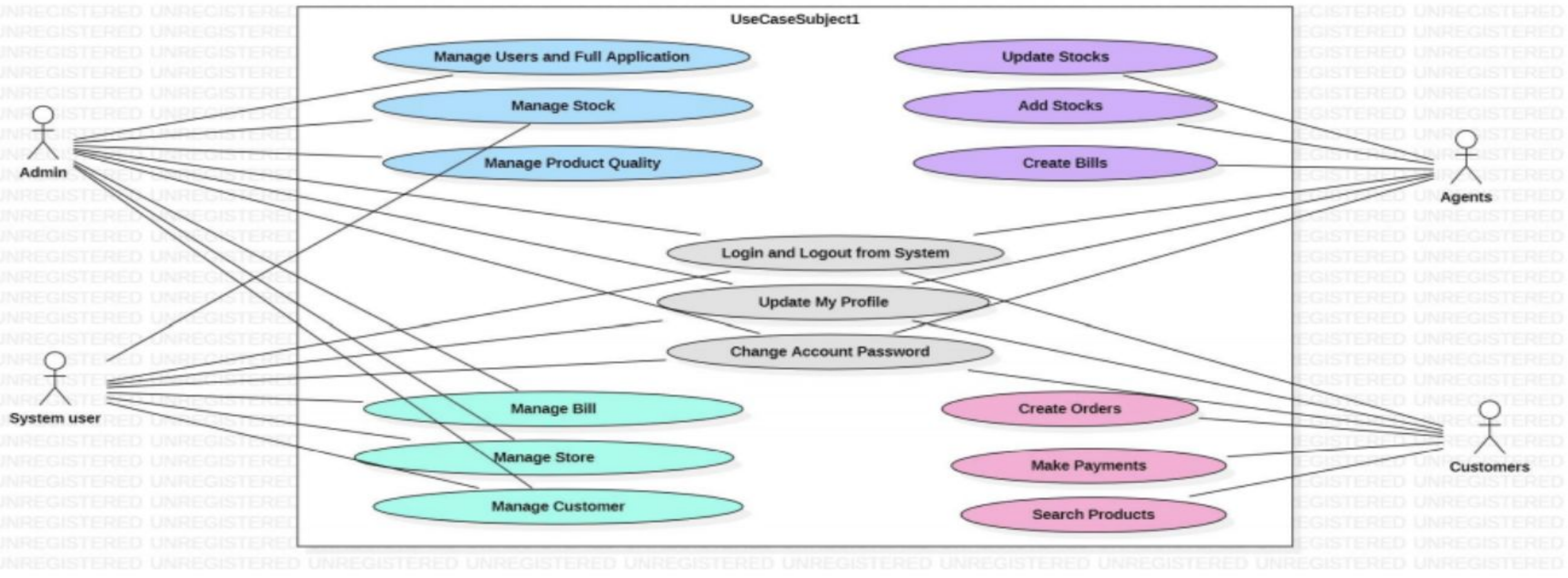
Managing Risk

Risk ID#	Status	Risk Appetite	Action
R01	Open	Avoid	Data encryption
R02	Open	Mitigate	Resolve technical issues
R03	Open	Mitigate	Follow the game plan
R04	Open	Avoid	Stick to the budget
R05	Open	Accept	Pitch the product

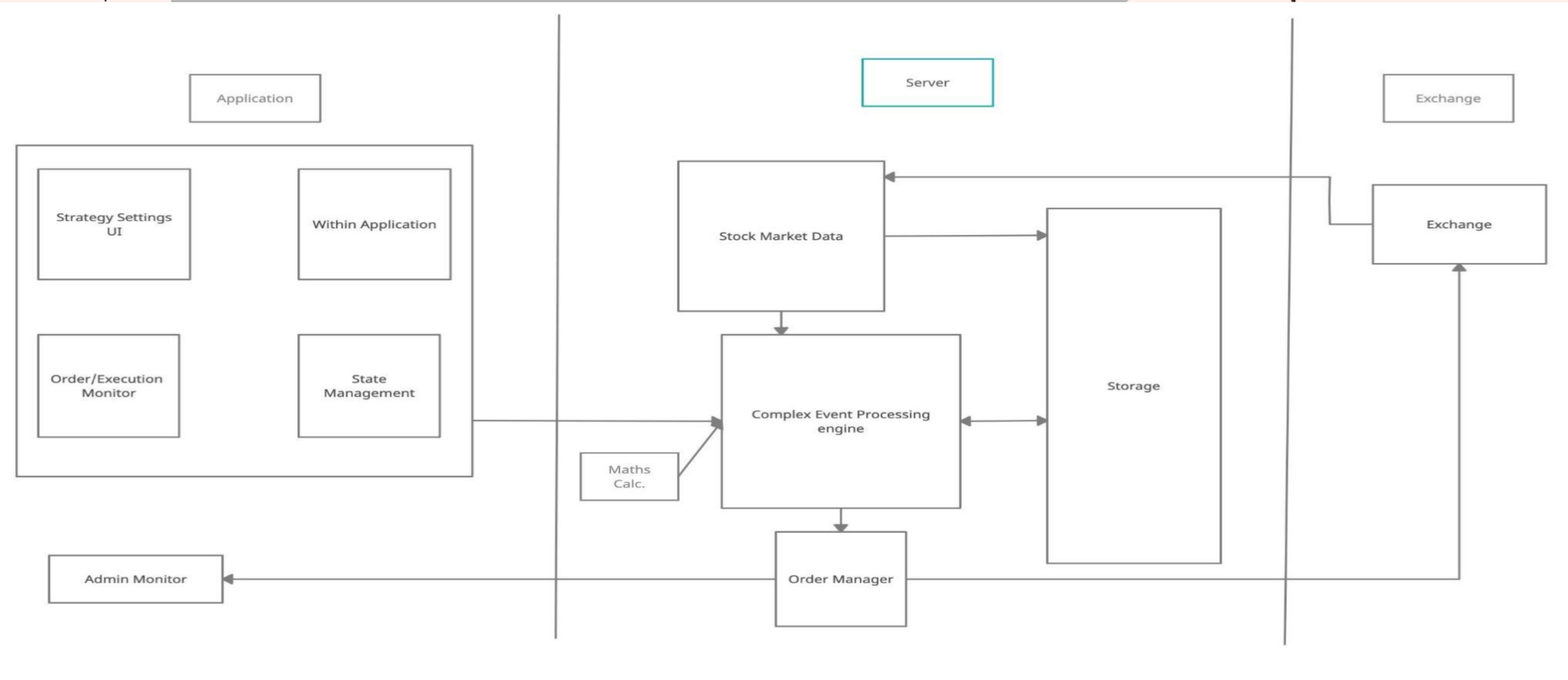


Architecture and Design of the System

Use case Diagram - This use case diagram is a graphic depiction of the interactions among the elements of the Stock Management System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Stock Management System.

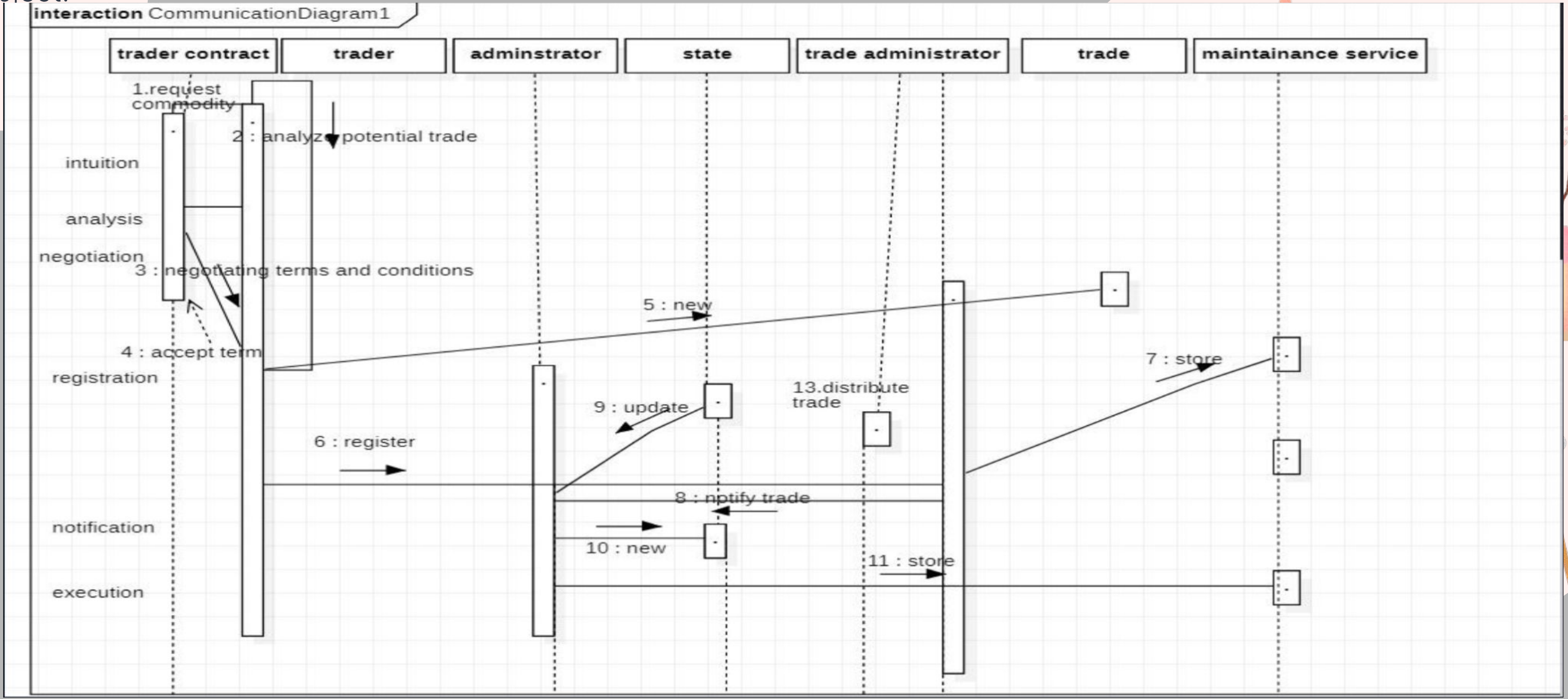


Architecture Diagram – An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

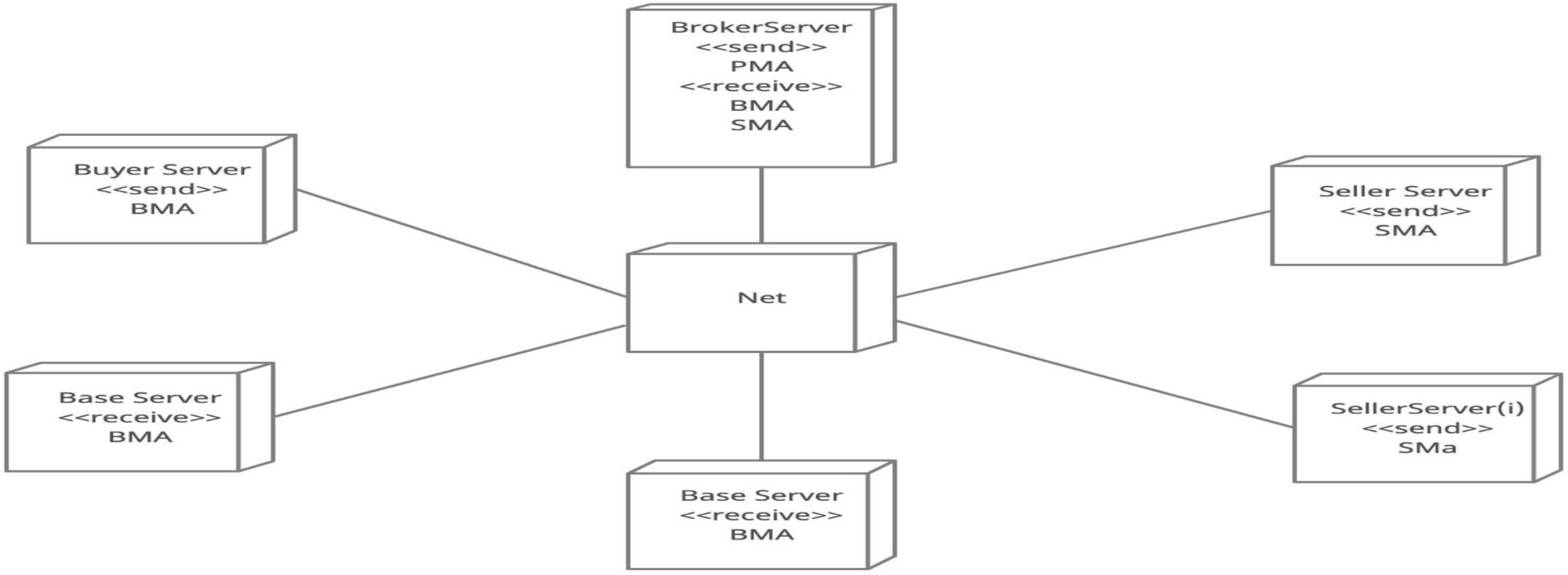


Architecture Diagram Of App

Collaboration Diagram – A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.



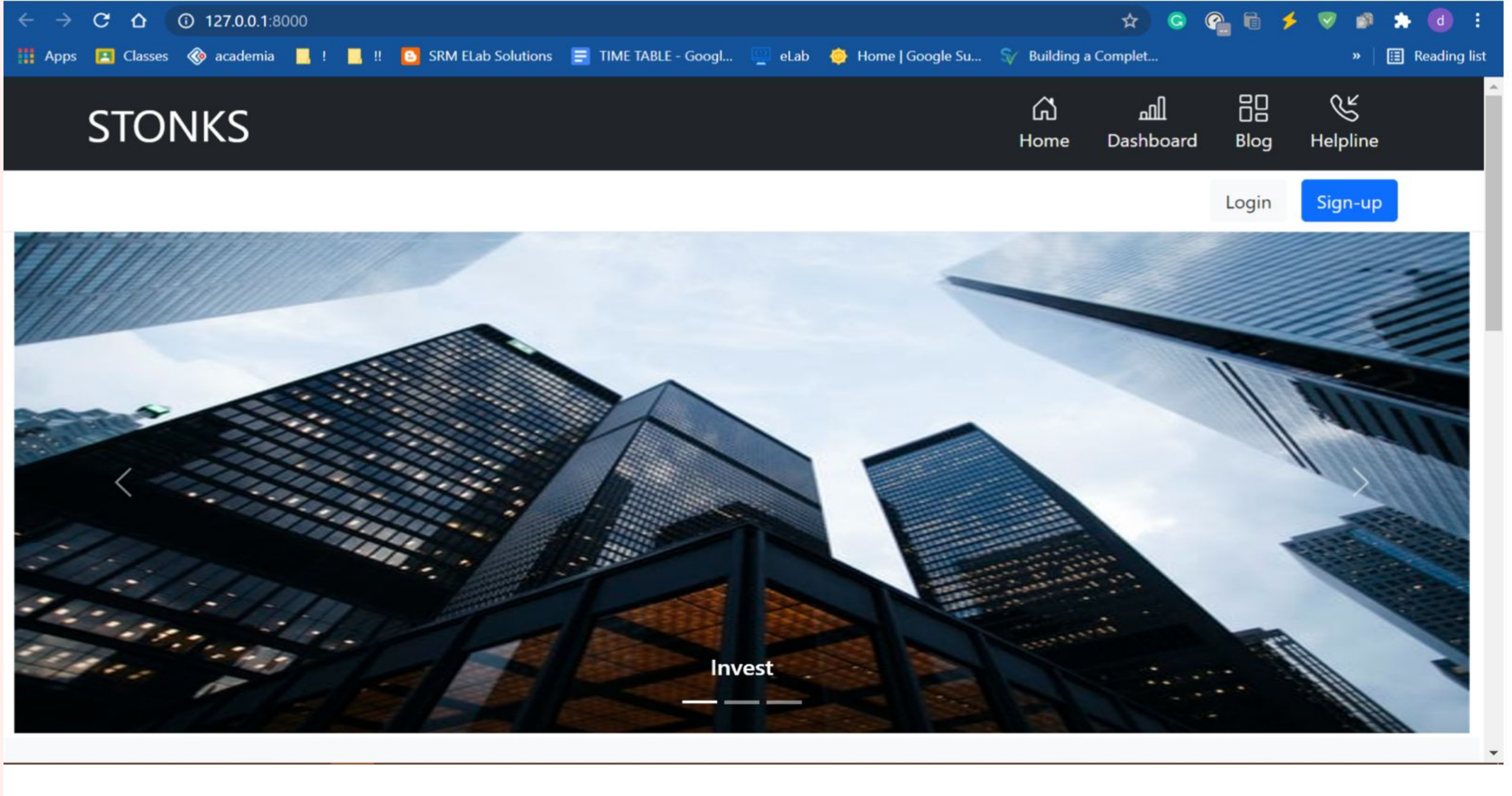
Deployment Diagram – A UML deployment diagram is a diagram that shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams are a kind of structure diagram used in modeling the physical aspects of an object-oriented system. The deployment diagram of our application is shown below:



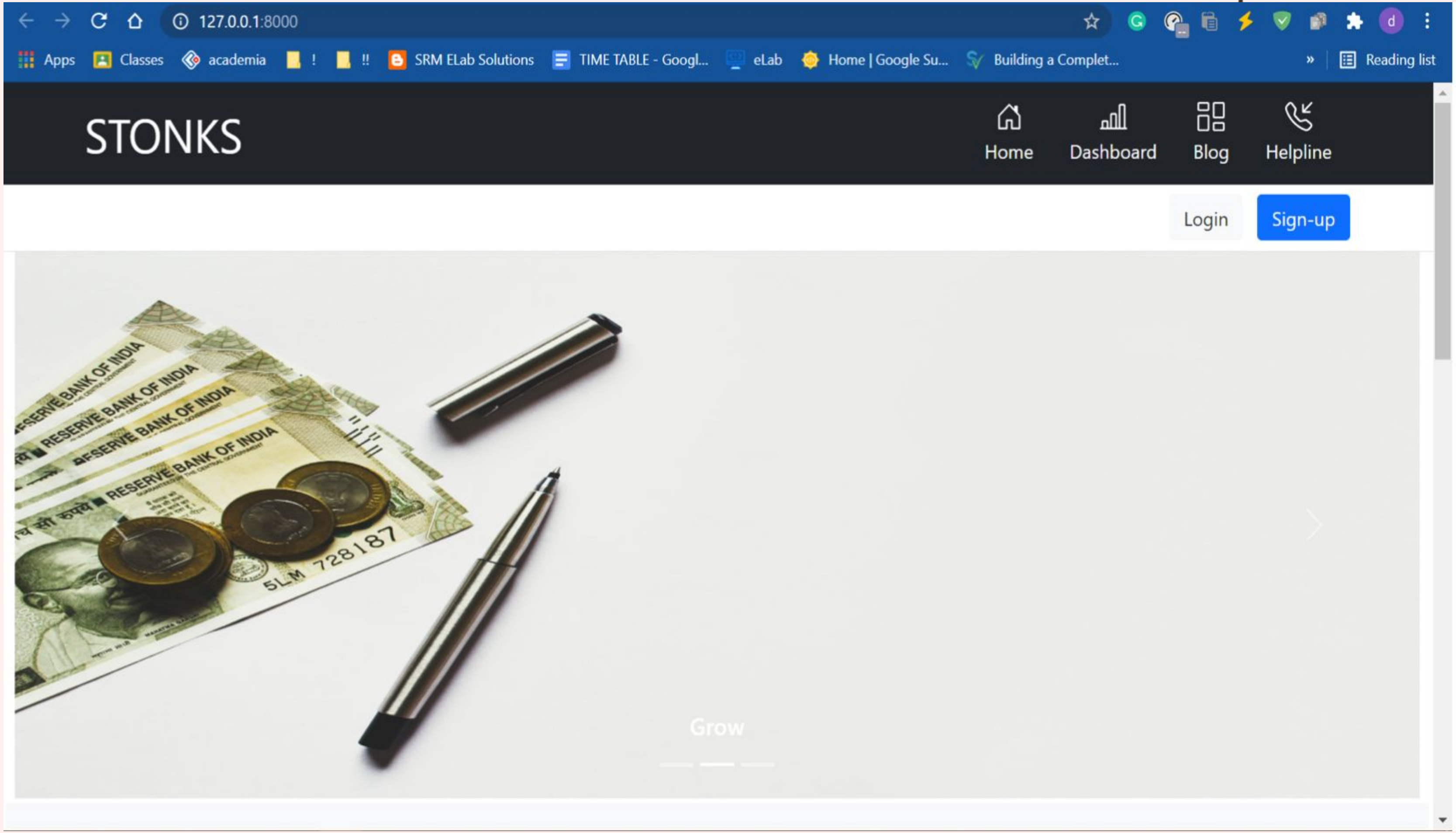
Deployment Diagram for Stock Marketing App

Front-end

Frontend

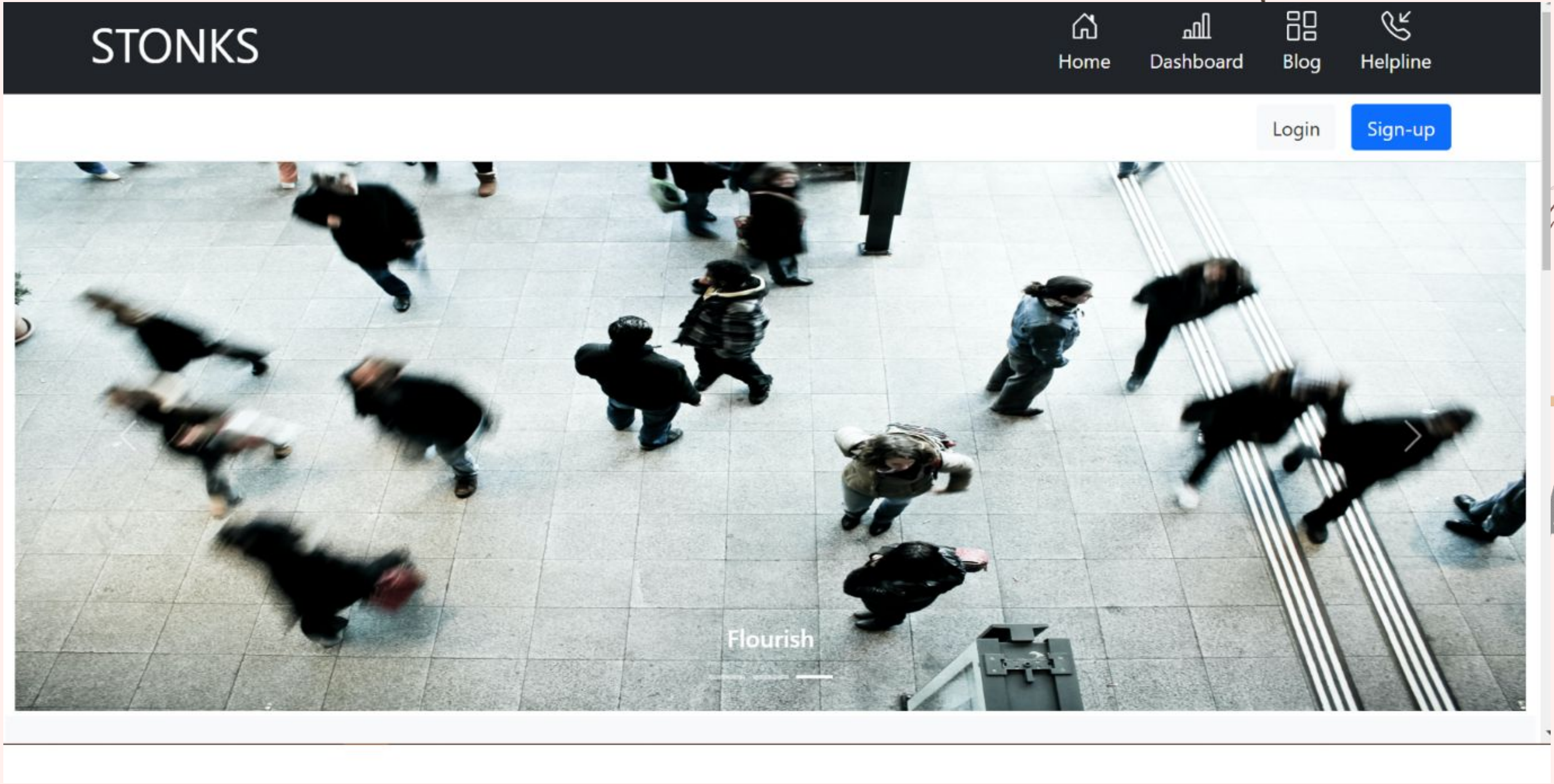

127.0.0.1:8000
Apps
Classes
academia
!
!!
SRM ELab Solutions
TIME TABLE - Googl...
eLab
Home | Google Su...
Building a Complet...
Reading list
STONKS
Home
Dashboard
Blog
Helpline
Login
Sign-up
Invest

Frontend



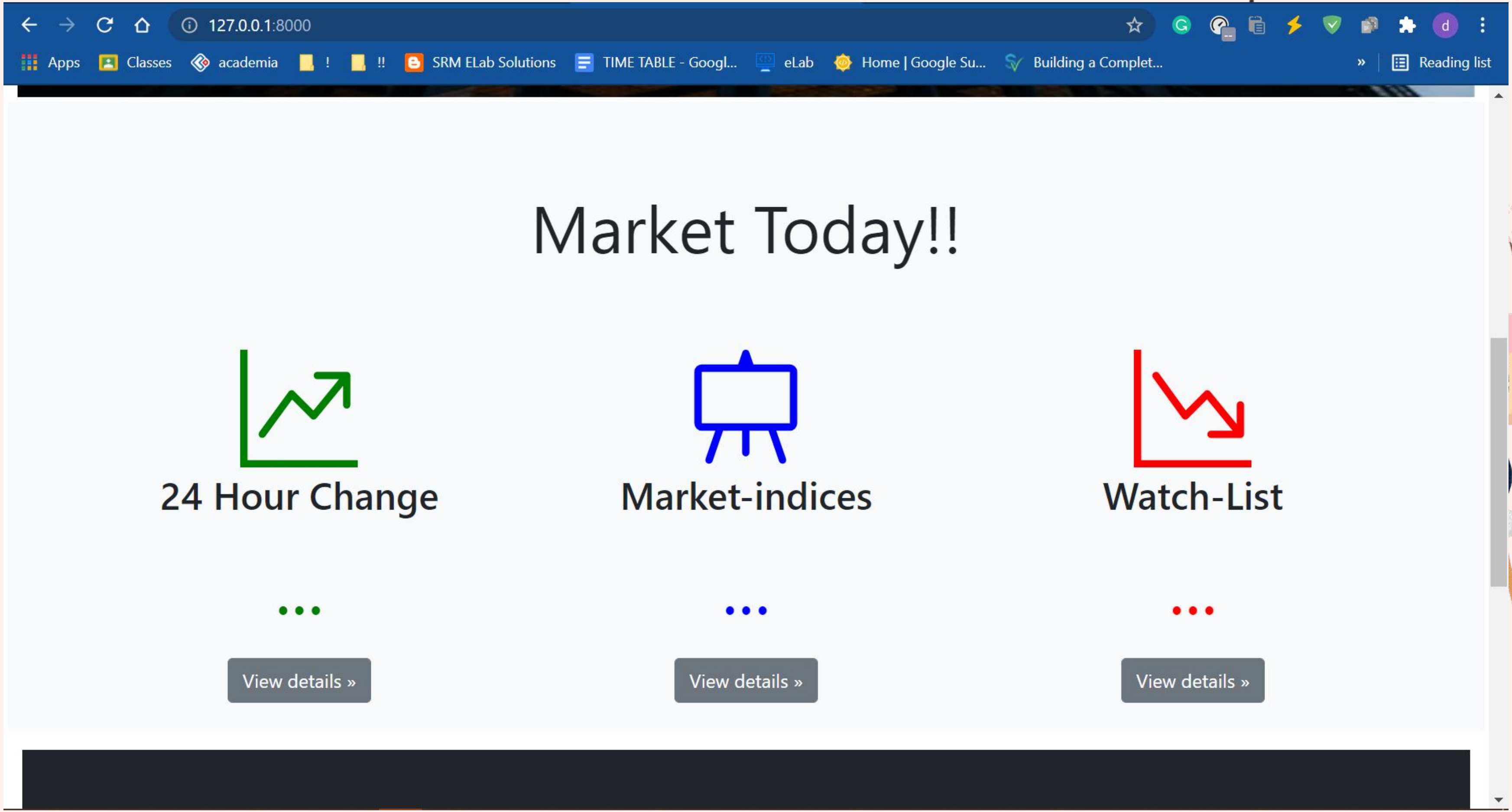
127.0.0.1:8000
Apps
Classes
academia
!
!!
SRM ELab Solutions
TIME TABLE - Googl...
eLab
Home | Google Su...
Building a Complet...
»
Reading list
STONKS
Home
Dashboard
Blog
Helpline
Login
Sign-up
Grow

Frontend



STONKS
Home
Dashboard
Blog
Helpline
Login
Sign-up
Flourish

Frontend



Market Today!!

24 Hour Change

...

View details »

Market-indices

...

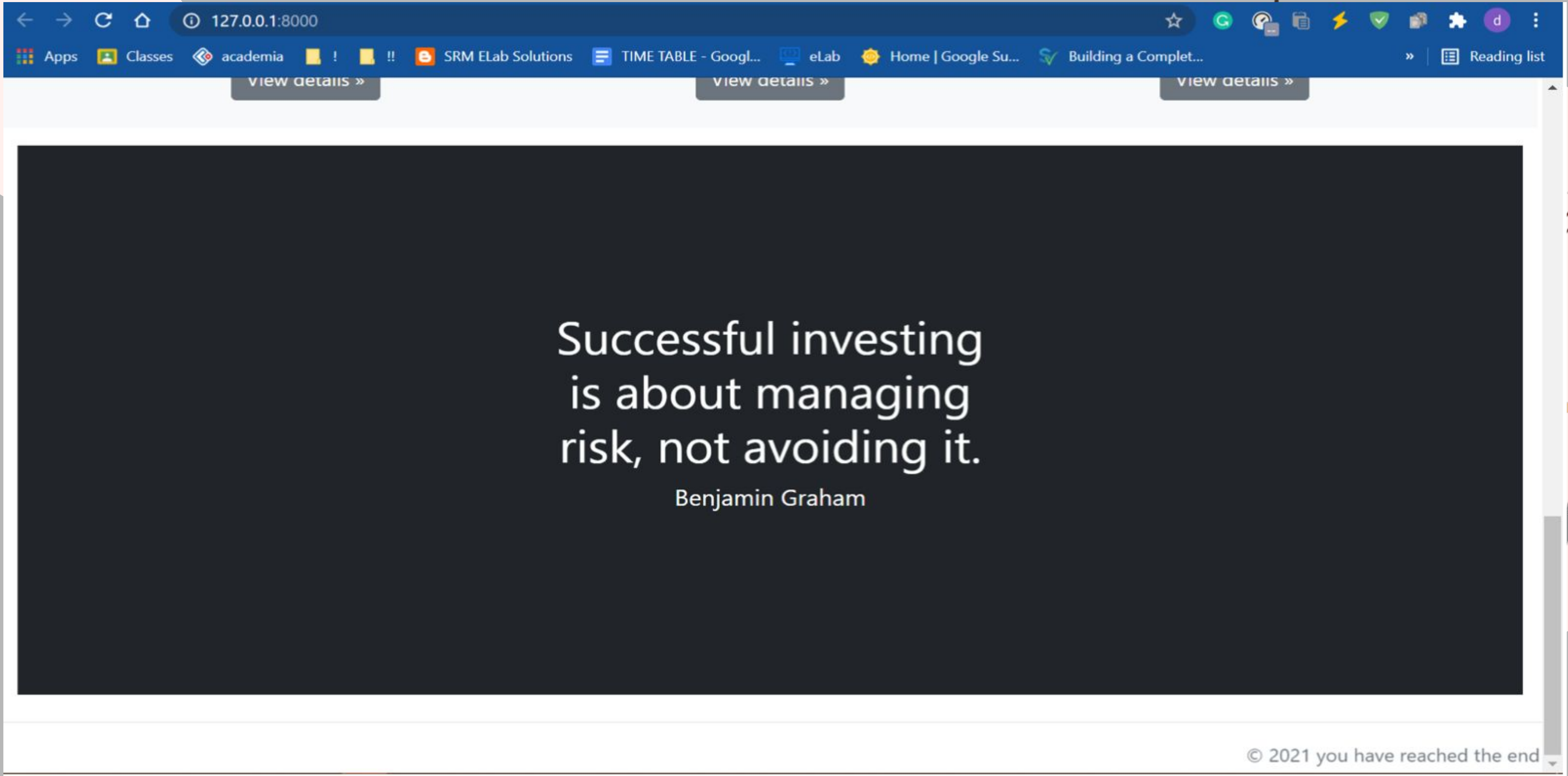
View details »

Watch-List

...

View details »

Frontend

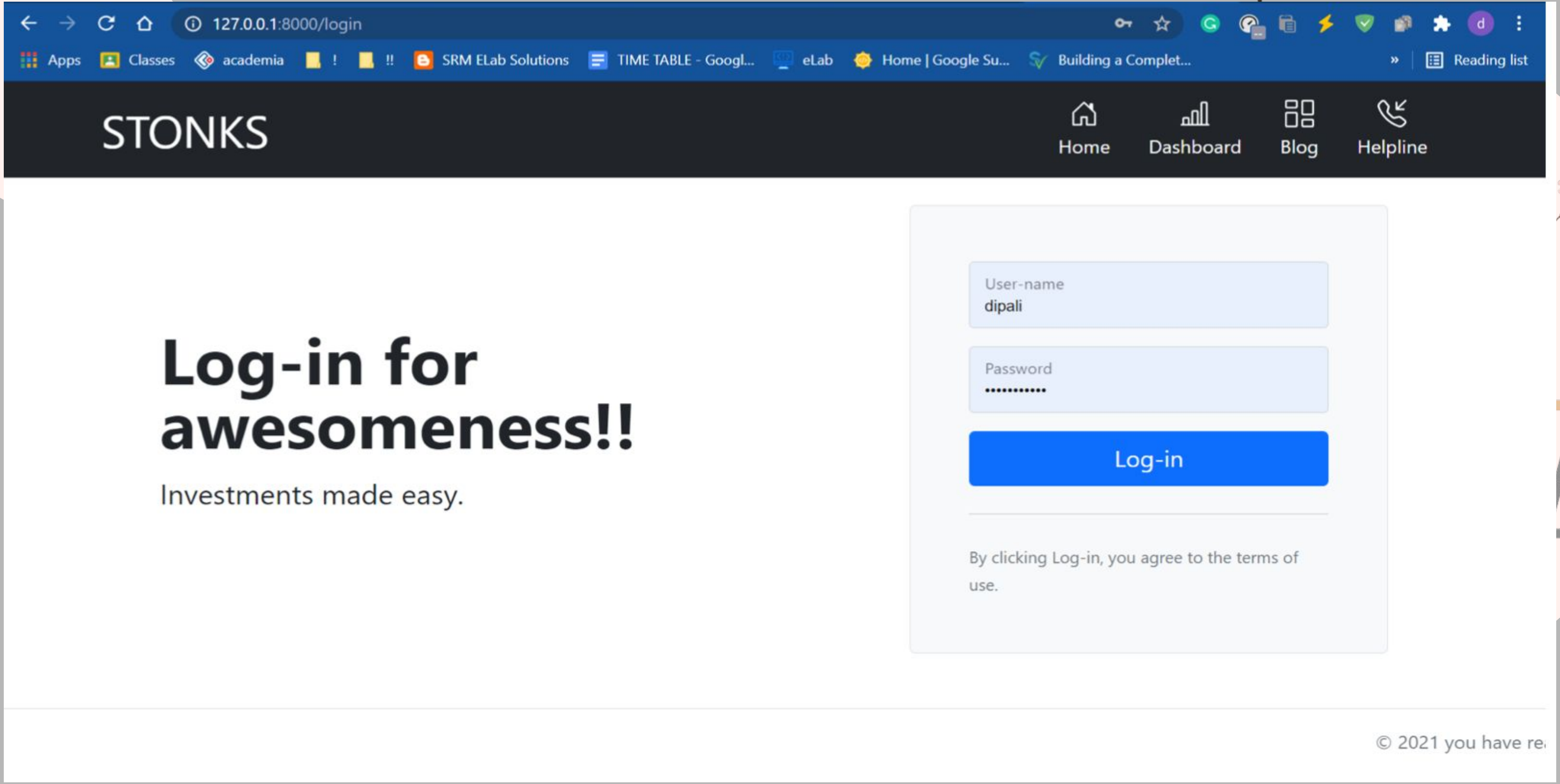


Successful investing is about managing risk, not avoiding it.

Benjamin Graham

© 2021 you have reached the end

Frontend



Log-in for awesomeness!!

Investments made easy.

User-name
dipali

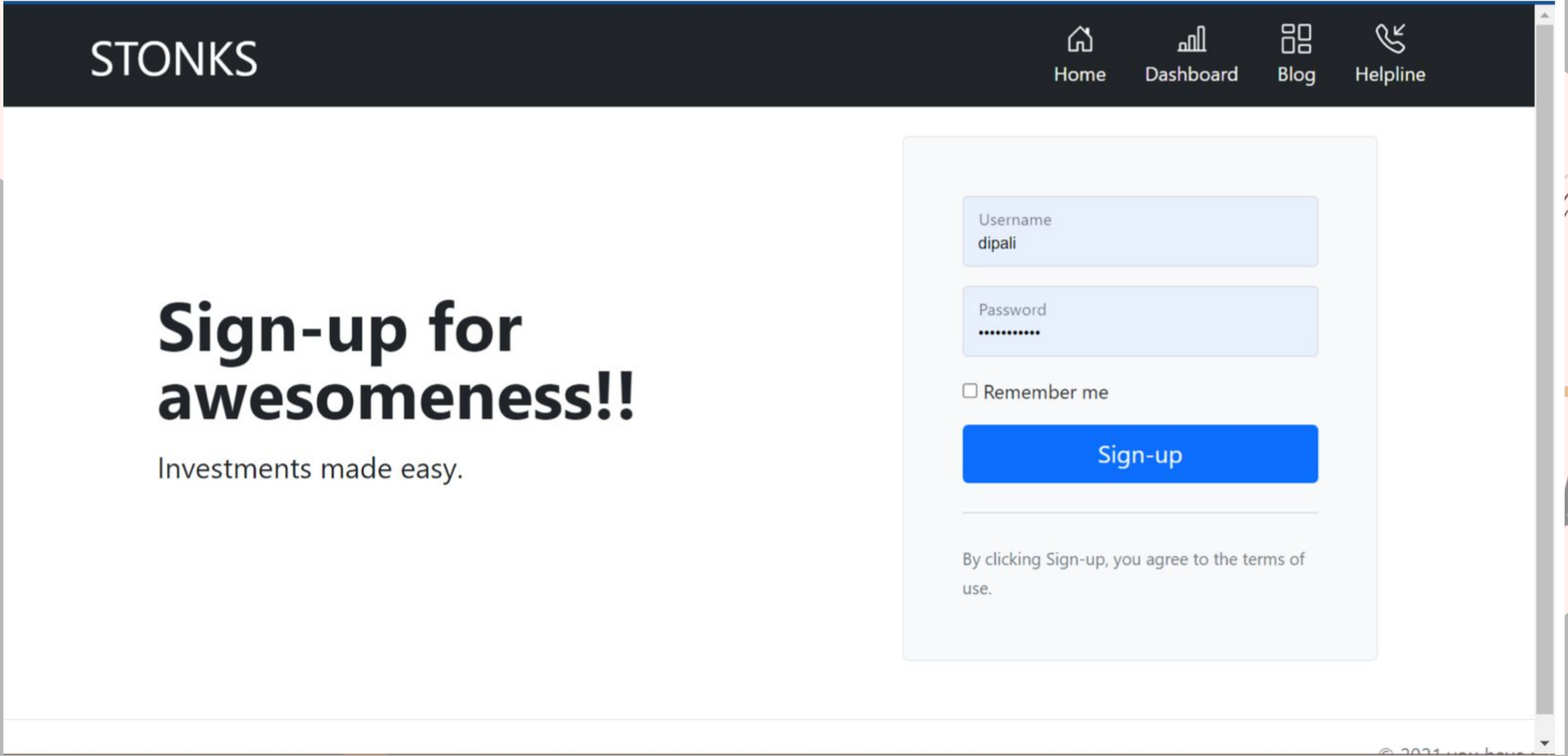
Password
••••••••••••

Log-in

By clicking Log-in, you agree to the terms of use.

© 2021 you have re

Frontend



Sign-up for awesomeness!!

Investments made easy.

Username: dipali

Password: •••••••••••

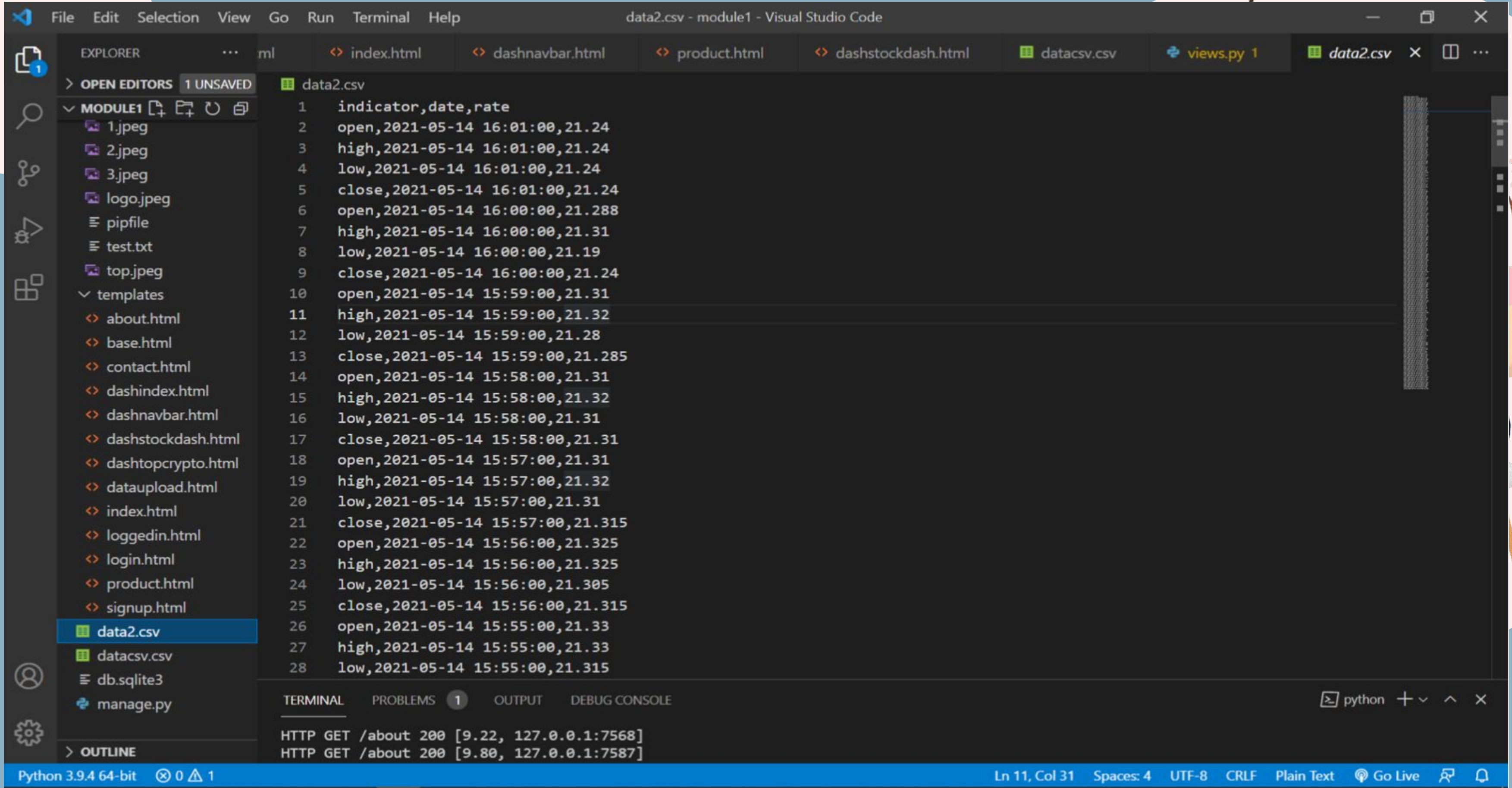
□ Remember me

By clicking Sign-up, you agree to the terms of use.

© 2021 you have

Back-end

The Live data of highs and lows of each stock:

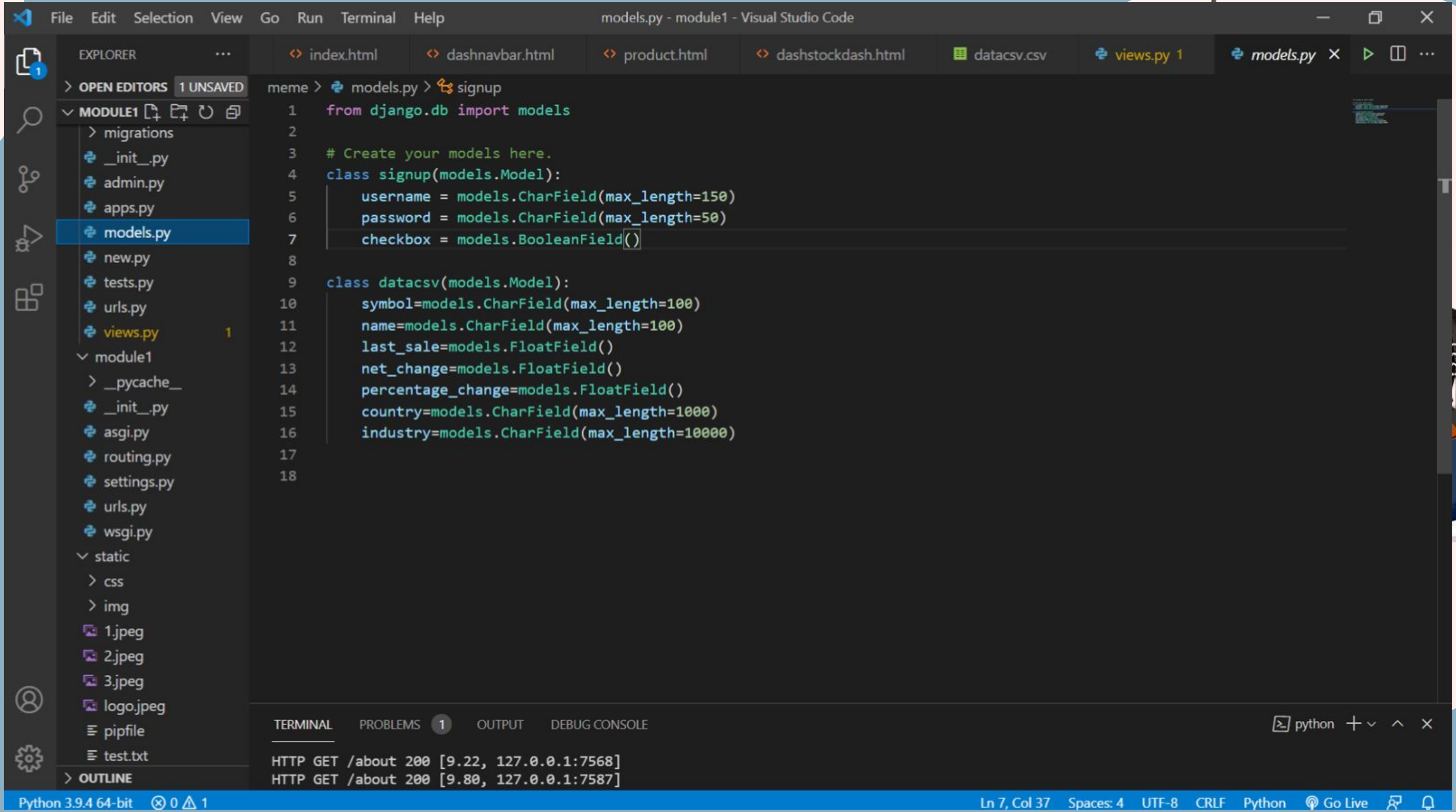


data2.csv

```
indicator,date,rate
open,2021-05-14 16:01:00,21.24
high,2021-05-14 16:01:00,21.24
low,2021-05-14 16:01:00,21.24
close,2021-05-14 16:01:00,21.24
open,2021-05-14 16:00:00,21.288
high,2021-05-14 16:00:00,21.31
low,2021-05-14 16:00:00,21.19
close,2021-05-14 16:00:00,21.24
open,2021-05-14 15:59:00,21.31
high,2021-05-14 15:59:00,21.32
low,2021-05-14 15:59:00,21.28
close,2021-05-14 15:59:00,21.285
open,2021-05-14 15:58:00,21.31
high,2021-05-14 15:58:00,21.32
low,2021-05-14 15:58:00,21.31
close,2021-05-14 15:58:00,21.31
open,2021-05-14 15:57:00,21.31
high,2021-05-14 15:57:00,21.32
low,2021-05-14 15:57:00,21.31
close,2021-05-14 15:57:00,21.315
open,2021-05-14 15:56:00,21.325
high,2021-05-14 15:56:00,21.325
low,2021-05-14 15:56:00,21.305
close,2021-05-14 15:56:00,21.315
open,2021-05-14 15:55:00,21.33
high,2021-05-14 15:55:00,21.33
low,2021-05-14 15:55:00,21.315
```

```
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

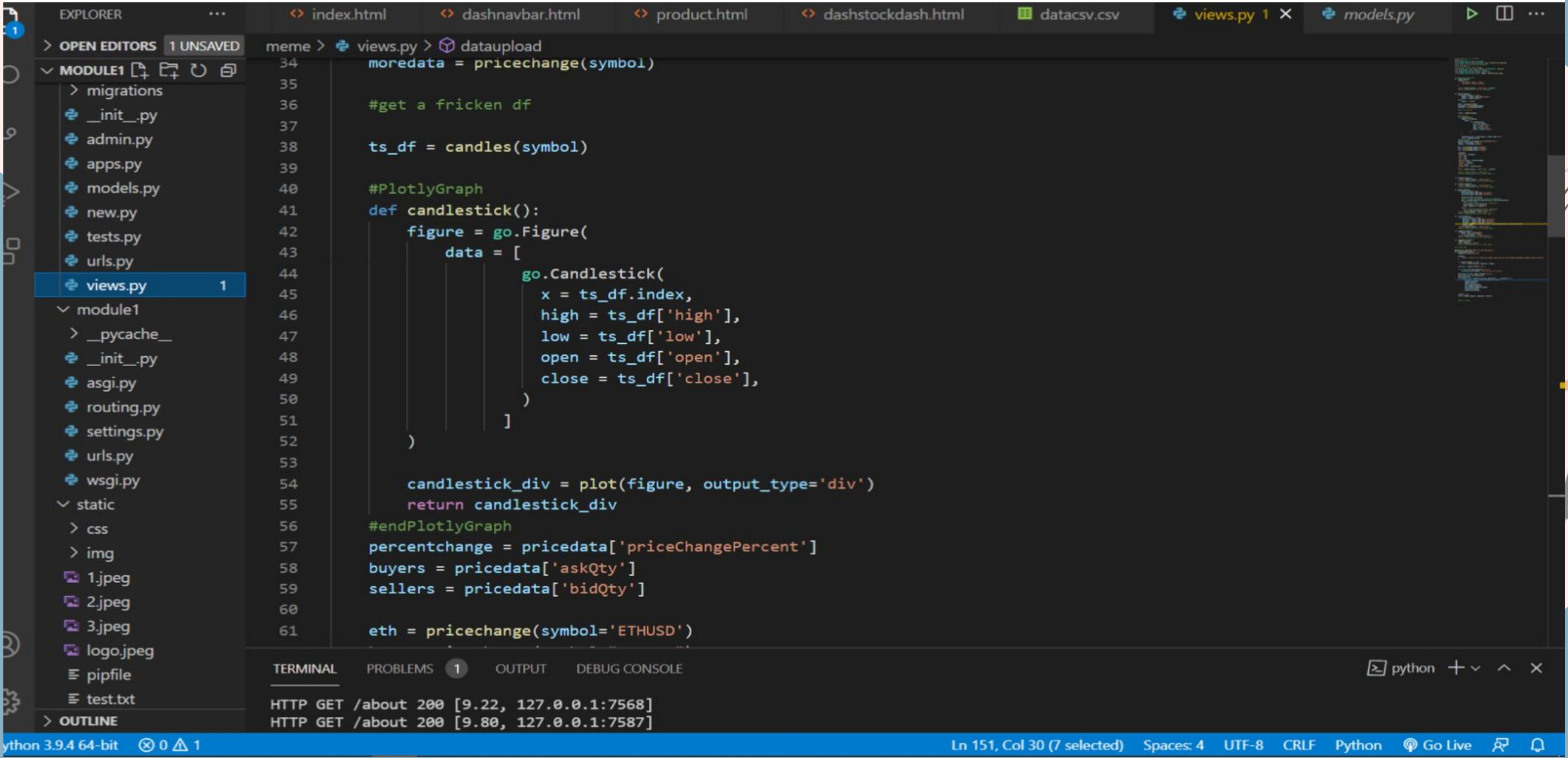
Code for backend -



```python
from django.db import models

# Create your models here.
class signup(models.Model):
    username = models.CharField(max_length=150)
    password = models.CharField(max_length=50)
    checkbox = models.BooleanField()

class datacsv(models.Model):
    symbol=models.CharField(max_length=100)
    name=models.CharField(max_length=100)
    last_sale=models.FloatField()
    net_change=models.FloatField()
    percentage_change=models.FloatField()
    country=models.CharField(max_length=1000)
    industry=models.CharField(max_length=10000)
```

```
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

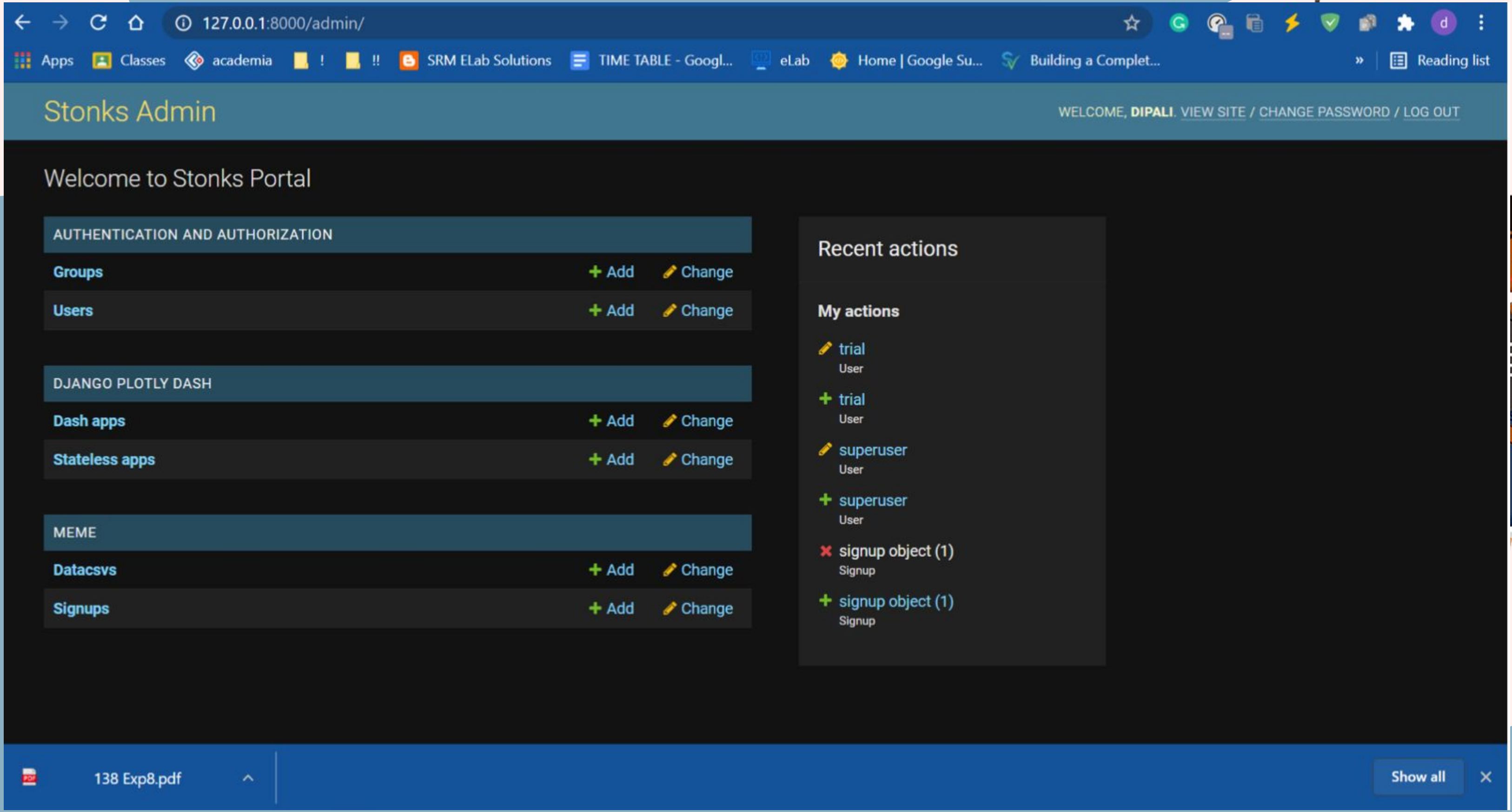
Code for backend -



```
    moredata = pricechange(symbol)

    #get a fricken df

    ts_df = candles(symbol)

    #PlotlyGraph
    def candlestick():
        figure = go.Figure(
            data = [
                    go.Candlestick(
                      x = ts_df.index,
                      high = ts_df['high'],
                      low = ts_df['low'],
                      open = ts_df['open'],
                      close = ts_df['close'],
                    )
                  ]
        )

        candlestick_div = plot(figure, output_type='div')
        return candlestick_div
    #endPlotlyGraph
    percentchange = pricedata['priceChangePercent']
    buyers = pricedata['askQty']
    sellers = pricedata['bidQty']

    eth = pricechange(symbol='ETHUSD')
```

```
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

Result of Module 2



Stonks Admin

WELCOME, **DIPALI**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Welcome to Stonks Portal

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

DJANGO PLOTLY DASH		
Dash apps	+ Add	Change
Stateless apps	+ Add	Change

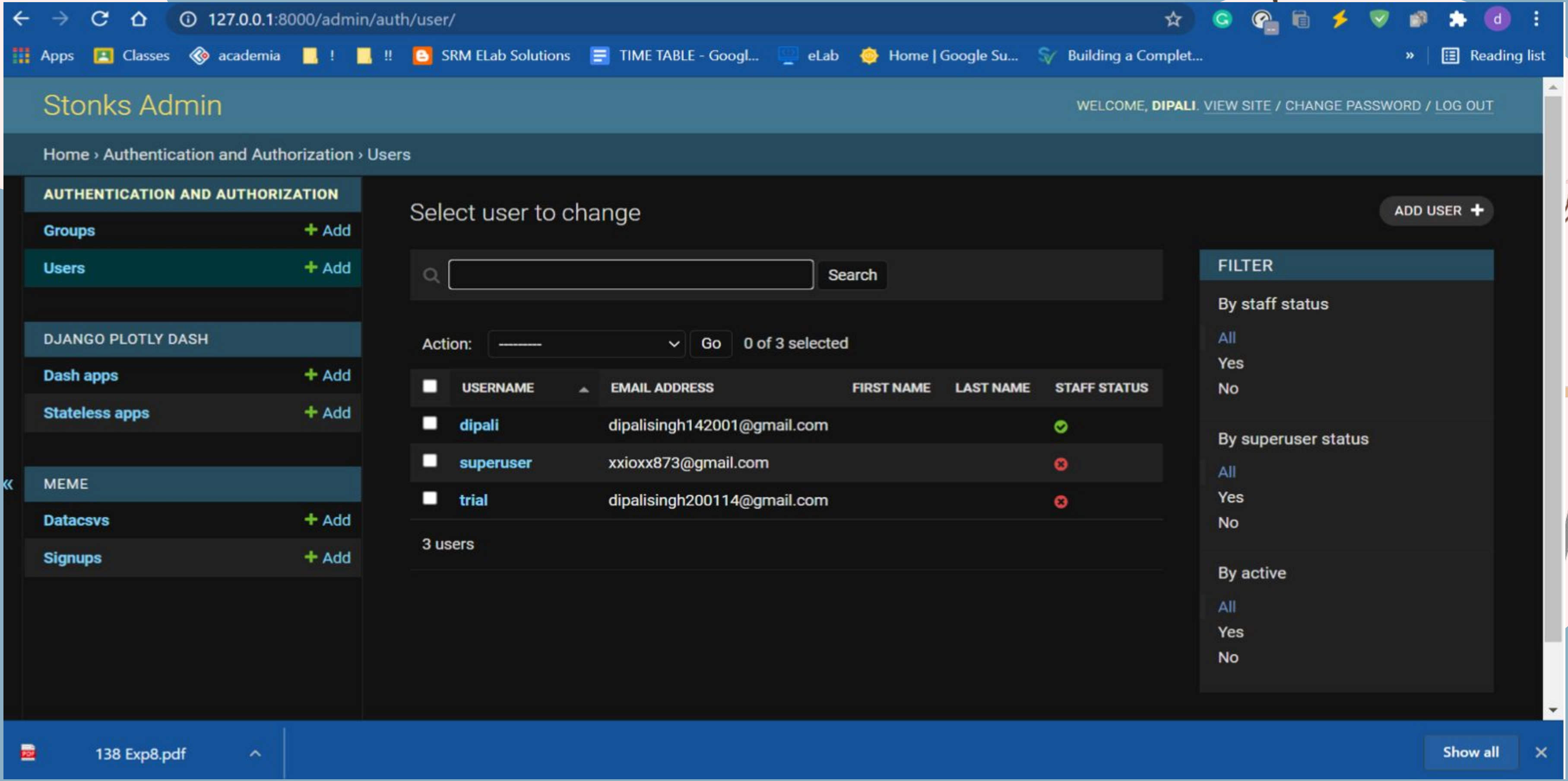
MEME		
Datacsvs	+ Add	Change
Signups	+ Add	Change

Recent actions

My actions

- trial
User
- + trial
User
- superuser
User
- + superuser
User
- ✗ signup object (1)
Signup
- + signup object (1)
Signup

Result of Module 2



Stonks Admin

WELCOME, **DIPALI**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Authentication and Authorization › Users

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

DJANGO PLOTLY DASH

- Dash apps + Add
- Stateless apps + Add

MEME

- Datacsvs + Add
- Signups + Add

Select user to change

ADD USER +

Search

Action: --------- Go 0 of 3 selected

□	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
□	dipali	dipalisingh142001@gmail.com			✓
□	superuser	xxioxx873@gmail.com			✗
□	trial	dipalisingh200114@gmail.com			✗

3 users

FILTER

By staff status
- All
- Yes
- No

By superuser status
- All
- Yes
- No

By active
- All
- Yes
- No

Prediction

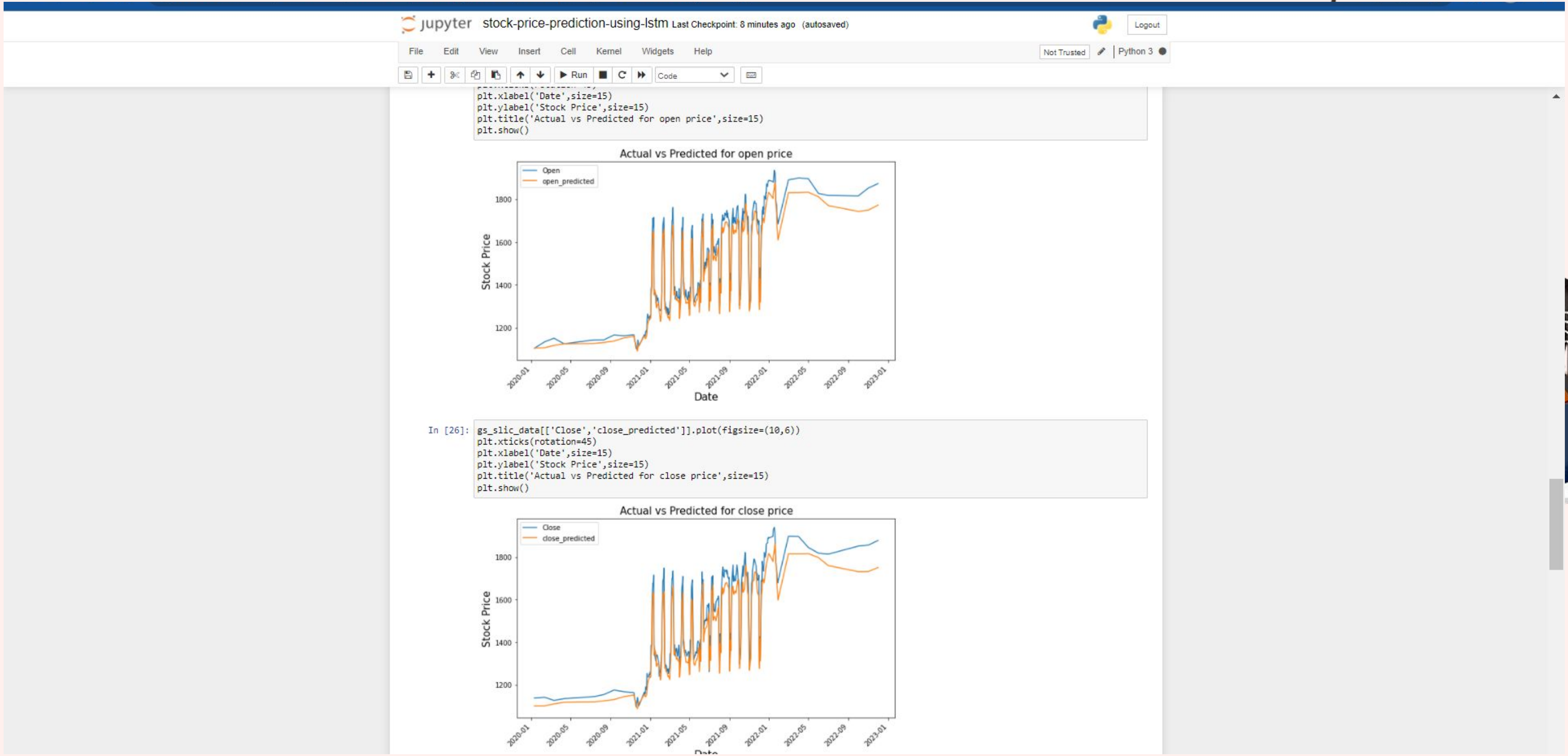
LSTM Model

- We will use the Long Short-Term Memory(LSTM) method to create a Machine Learning model to forecast stock values. They are used to make minor changes to the information by multiplying and adding. Long-term memory (LSTM) is a deep learning artificial recurrent neural network (RNN) architecture.
- 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 Opening Value of the stock, the Highest and Lowest values of that stock on the same days, as well as the Closing Value at the end of the day, are all indicated for each date.

Dataset

The dataset includes the data of the stocks infosys, facebook and apple from 29 Jan 2015 to 29 Jan 2022. This dataset was acquired from the yahoo finance website.

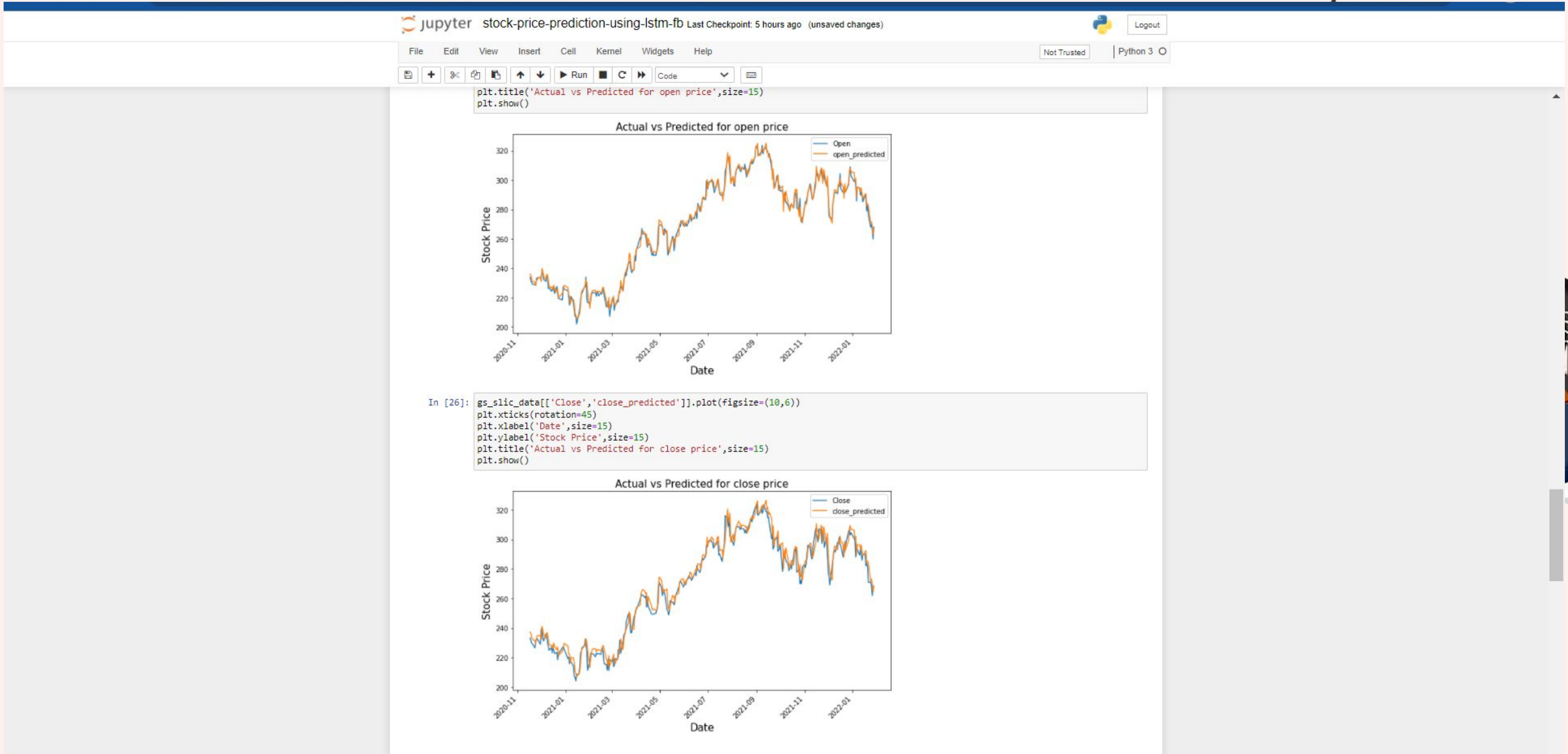
Result of Module 3:



```
plt.xlabel('Date',size=15)
plt.ylabel('Stock Price',size=15)
plt.title('Actual vs Predicted for open price',size=15)
plt.show()
```

```
In [26]: gs_slic_data[['Close','close_predicted']].plot(figsize=(10,6))
         plt.xticks(rotation=45)
         plt.xlabel('Date',size=15)
         plt.ylabel('Stock Price',size=15)
         plt.title('Actual vs Predicted for close price',size=15)
         plt.show()
```

Result of Module 3:

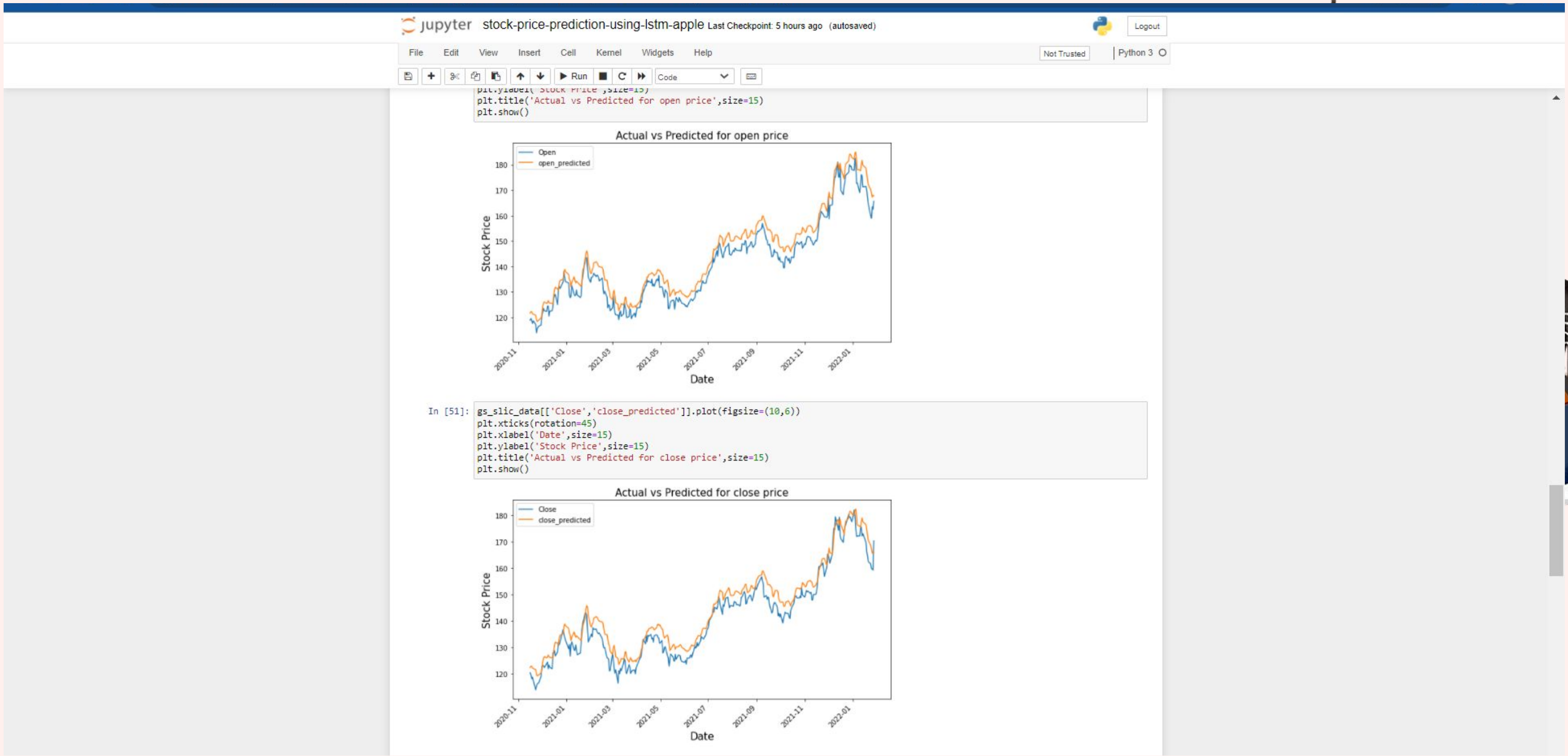


```
plt.title('Actual vs Predicted for open price',size=15)
plt.show()
```

In [26]:

```
gs_slic_data[['Close','close_predicted']].plot(figsize=(10,6))
plt.xticks(rotation=45)
plt.xlabel('Date',size=15)
plt.ylabel('Stock Price',size=15)
plt.title('Actual vs Predicted for close price',size=15)
plt.show()
```

Result of Module 3:



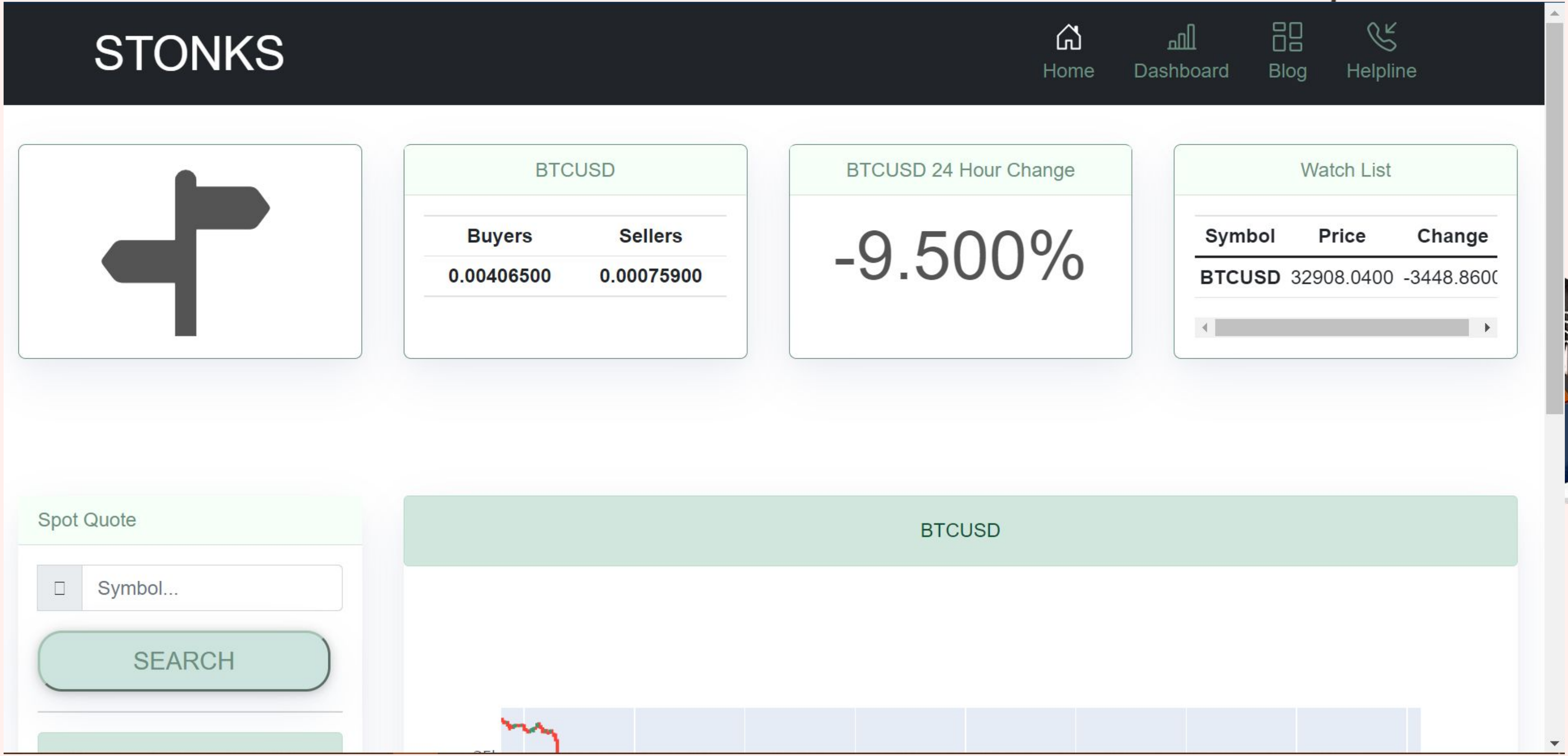
jupyter
stock-price-prediction-using-lstm-apple Last Checkpoint: 5 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code

```
plt.ylabel('Stock Price',size=15)
plt.title('Actual vs Predicted for open price',size=15)
plt.show()
```

In [51]:

```
gs_slic_data[['Close','close_predicted']].plot(figsize=(10,6))
plt.xticks(rotation=45)
plt.xlabel('Date',size=15)
plt.ylabel('Stock Price',size=15)
plt.title('Actual vs Predicted for close price',size=15)
plt.show()
```

Result of Module 3:



BTCUSD

Buyers	Sellers
0.00406500	0.00075900

BTCUSD 24 Hour Change

-9.500%

Watch List

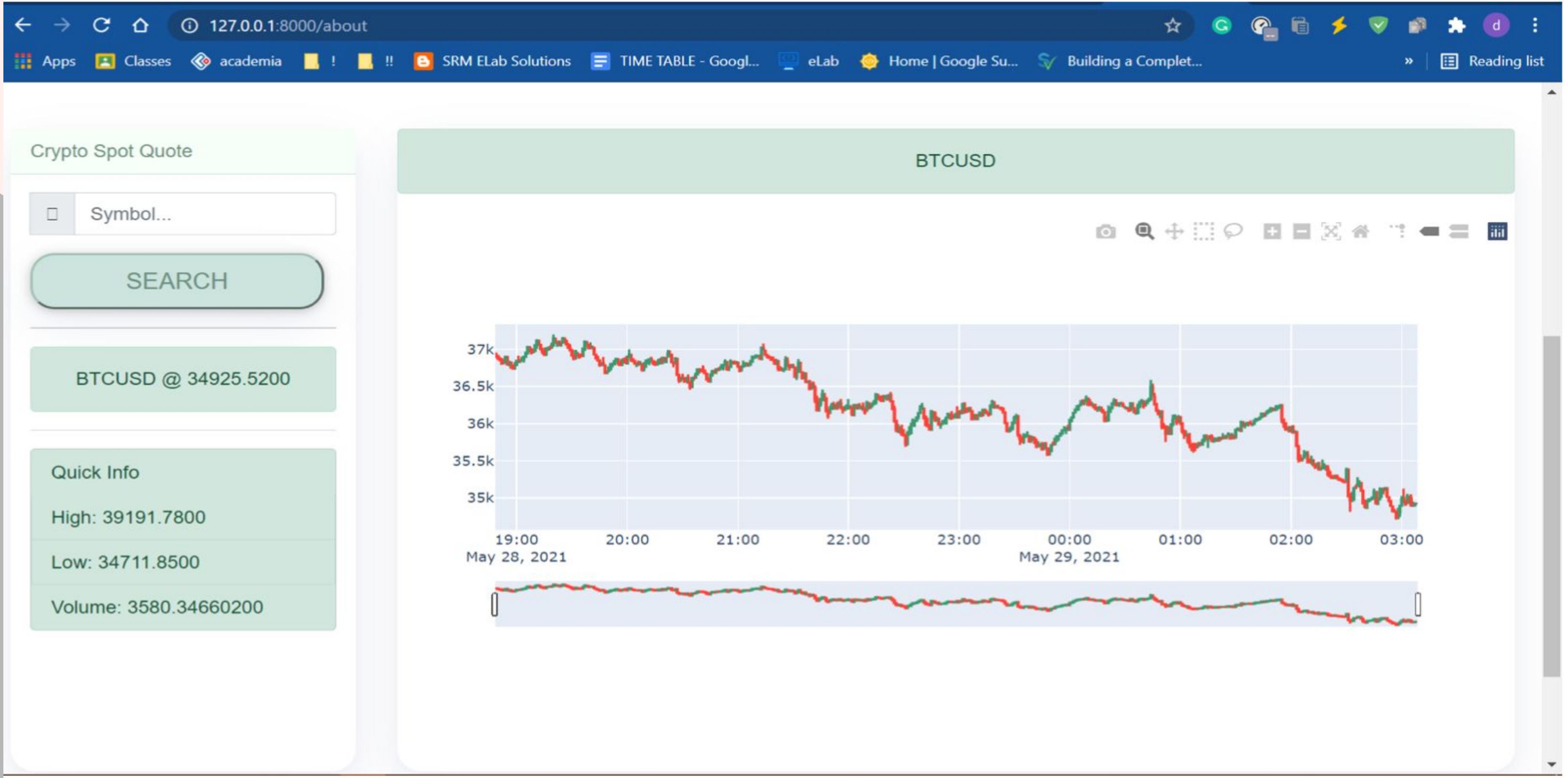
Symbol	Price	Change
BTCUSD	32908.0400	-3448.860(

Spot Quote

Symbol...

BTCUSD

Result of Module 3:



Crypto Spot Quote

BTCUSD

BTCUSD @ 34925.5200

Quick Info

High: 39191.7800

Low: 34711.8500

Volume: 3580.34660200

Thankyou