Stock Market Tracking App

Dipali Singh - RA1911030010138

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

The roles of the different stakeholders are as follows:

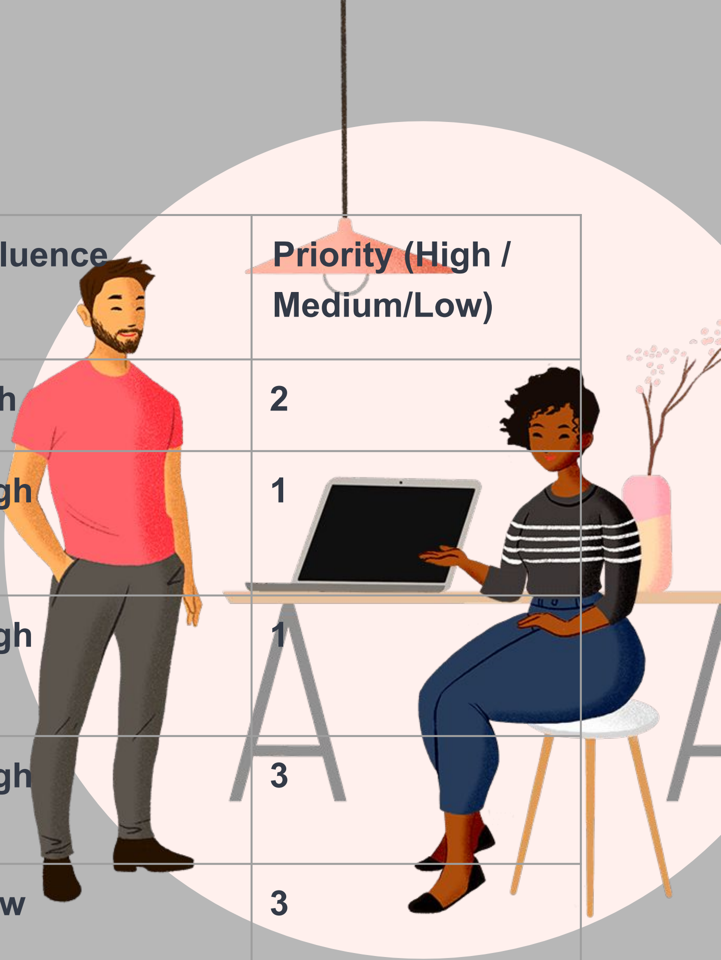
Manage and development of the project.
Manage the finances and development.
Manage the marketing and development.

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



Methodology:

Agile software development

In software development, agile practices involve discovering requirements and developing solutions through the collaborative effort of self-organizing and cross-functional teams and their customer/end user.

Communication Plans for Stakeholders

The different stakeholders involved must have a proper channel of communication to ensure the proper flow of development. Communication is extremely crucial to maintain the quality standard.

A few ways to ensure proper communication are listed below:

● Frequent meetings over online platforms like g-meet.
● Providing a single-page dashboard and communicating via meetings.
● Weekly report for High Interest and Low Influence.

Project Management Plan

Focus Area	Details
Integration Management	Stock market tracking app is a web app that will allow users to track stock market data. Our project team will hold little to no hierarchy with all out teammates working together in collaboration with outside parties. The different parts of the project will be integrated and the change will be managed using agile project methodology.
Scope Management	The scope of our project involves the tracking and transaction of stocks. The different stakeholders are: >Investors >Developers >Stock broker >Customers

Project Management Plan

Schedule Management	**Web app with detailed stock market analysis followed by a full stack app. Steps involved are:** ● **Planning Road map.** ● **Requirement Analysis** ● **Backend Software** ● **Backend Development** ● **Frontend Development** ● **Database Implementation** ● **Testing**
Quality Management	**The web app will include the necessary data encryption and will be continuously improved to maintain quality**

Project Management Plan

Resource Management	**Estimate and Manage the need** **People: Dipali Singh** **Finance: 65k** **Physical: Facilities, IT Infrastructure**

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:

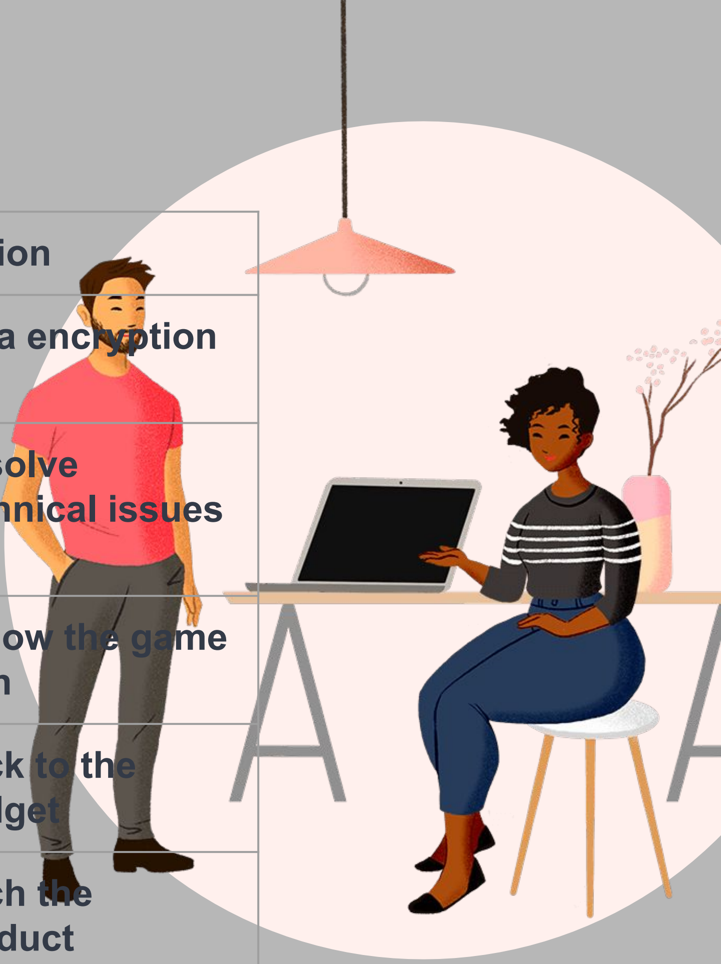
- **The predictions can be wrong due to inaccuracy in the app.**
- **Sometimes wrong stocks can be shown which may lead to misinformation.**
- **Out of date information can be shown.**
- **Sometimes personalised stocks may be incorrect due to inaccurate predictions.**

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

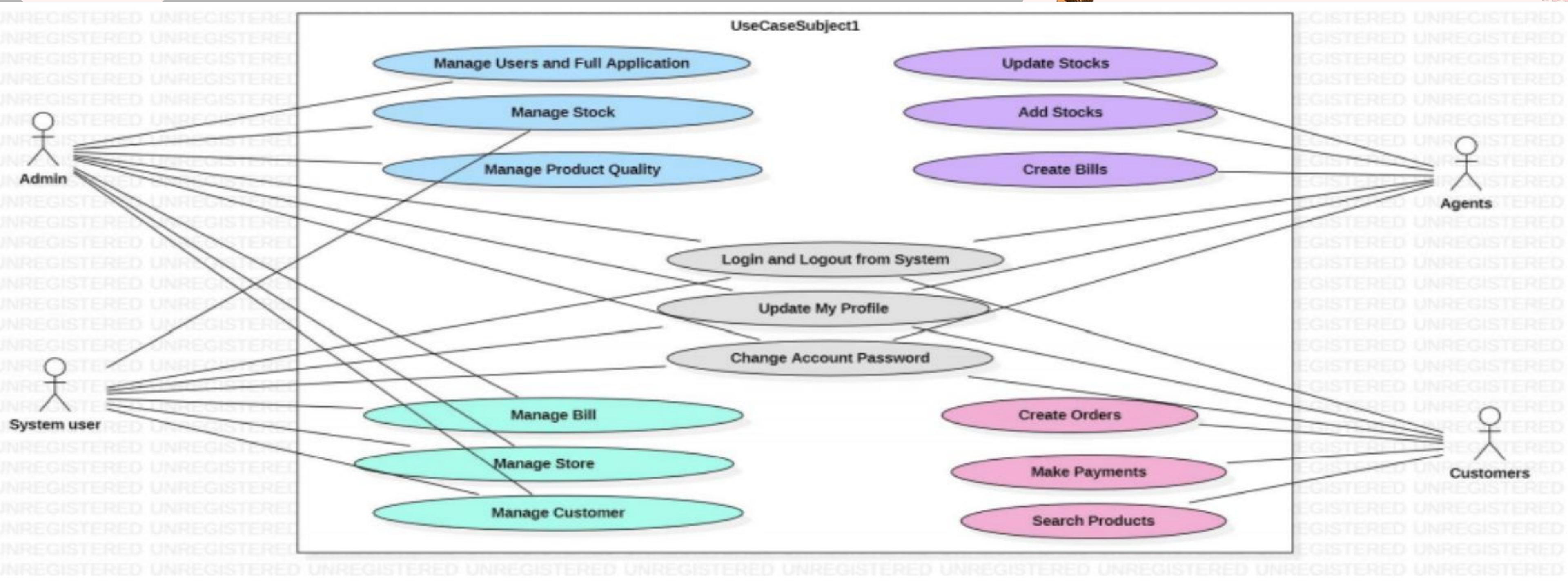
Managing Risk

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

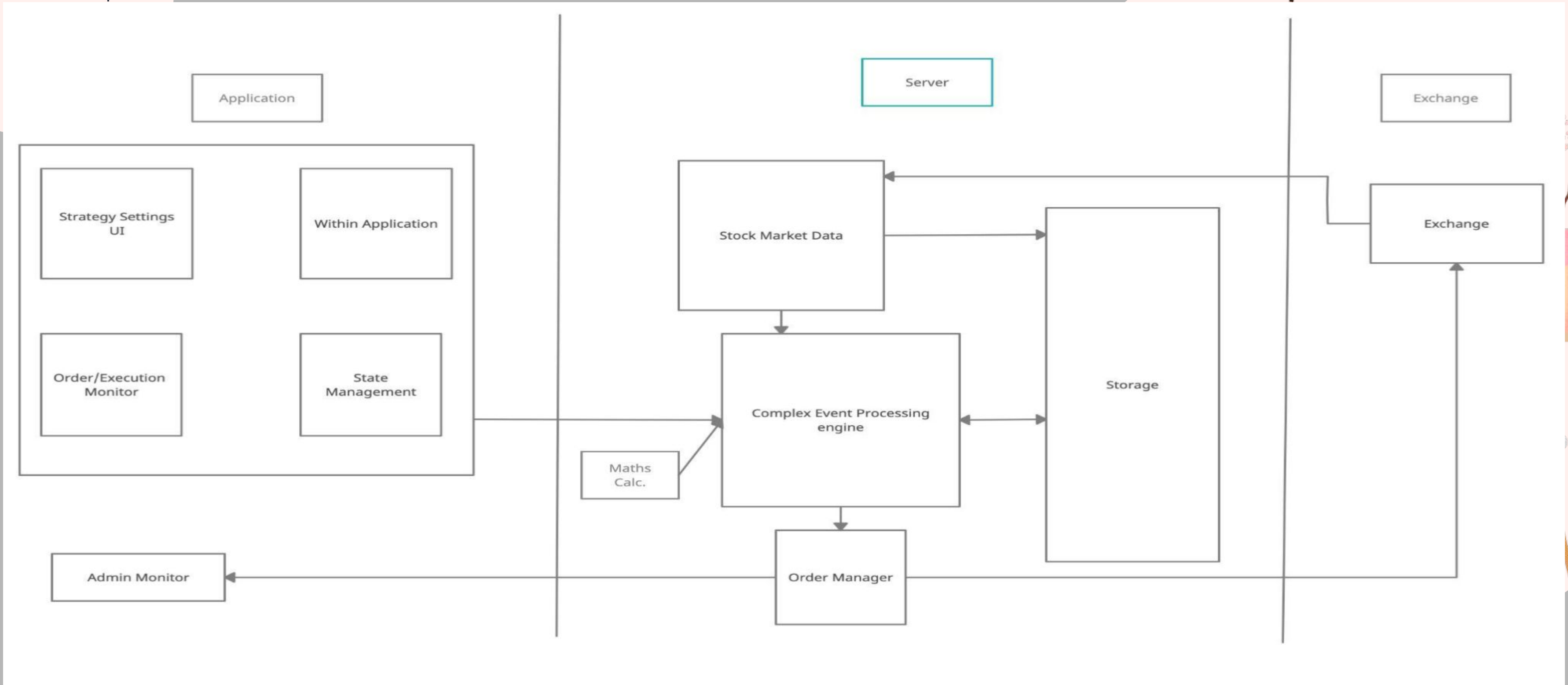


Architecture and Design of the System

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.

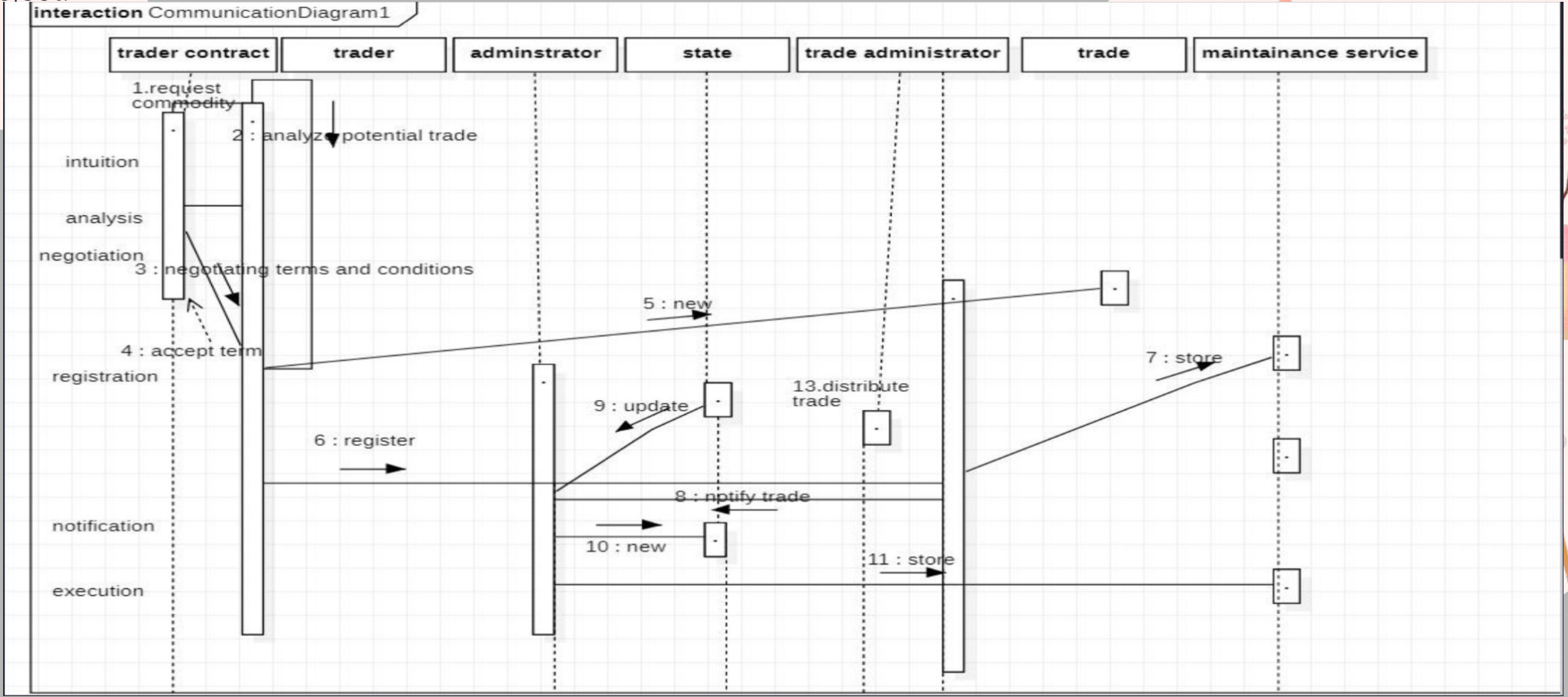


Architecture Diagram – An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the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

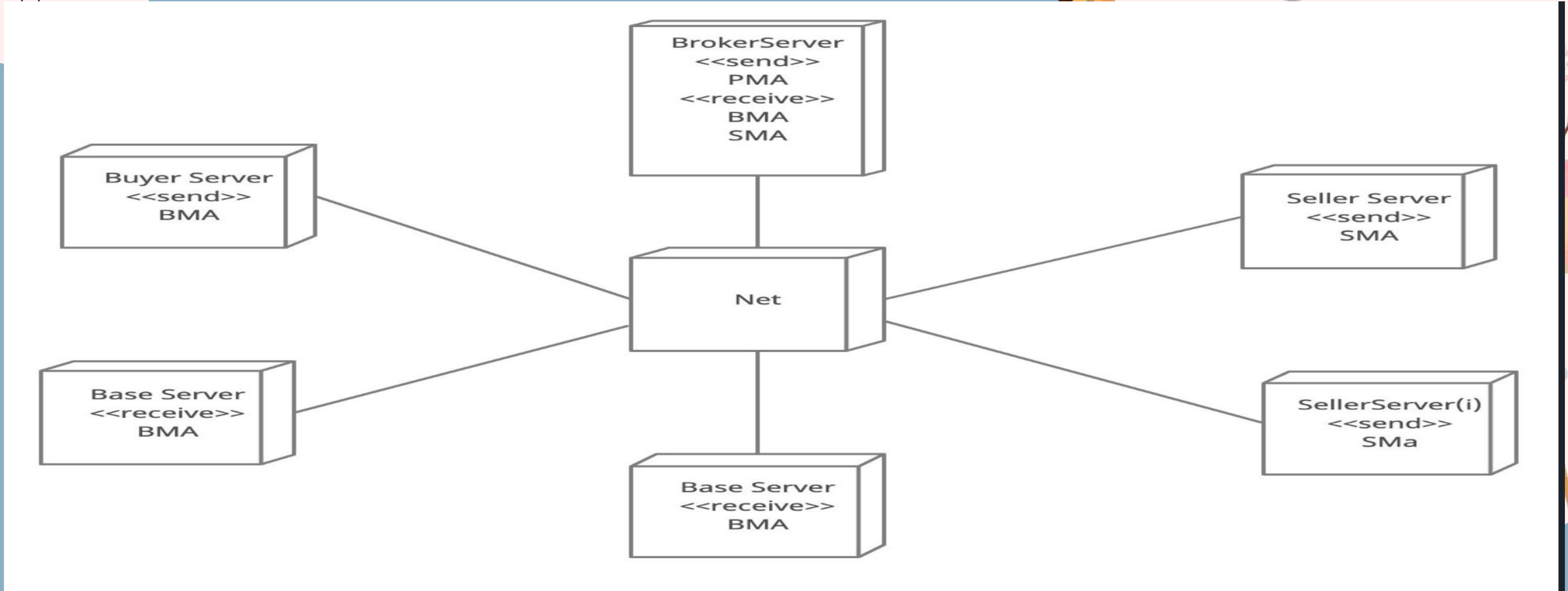


Architecture Diagram Of App

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.

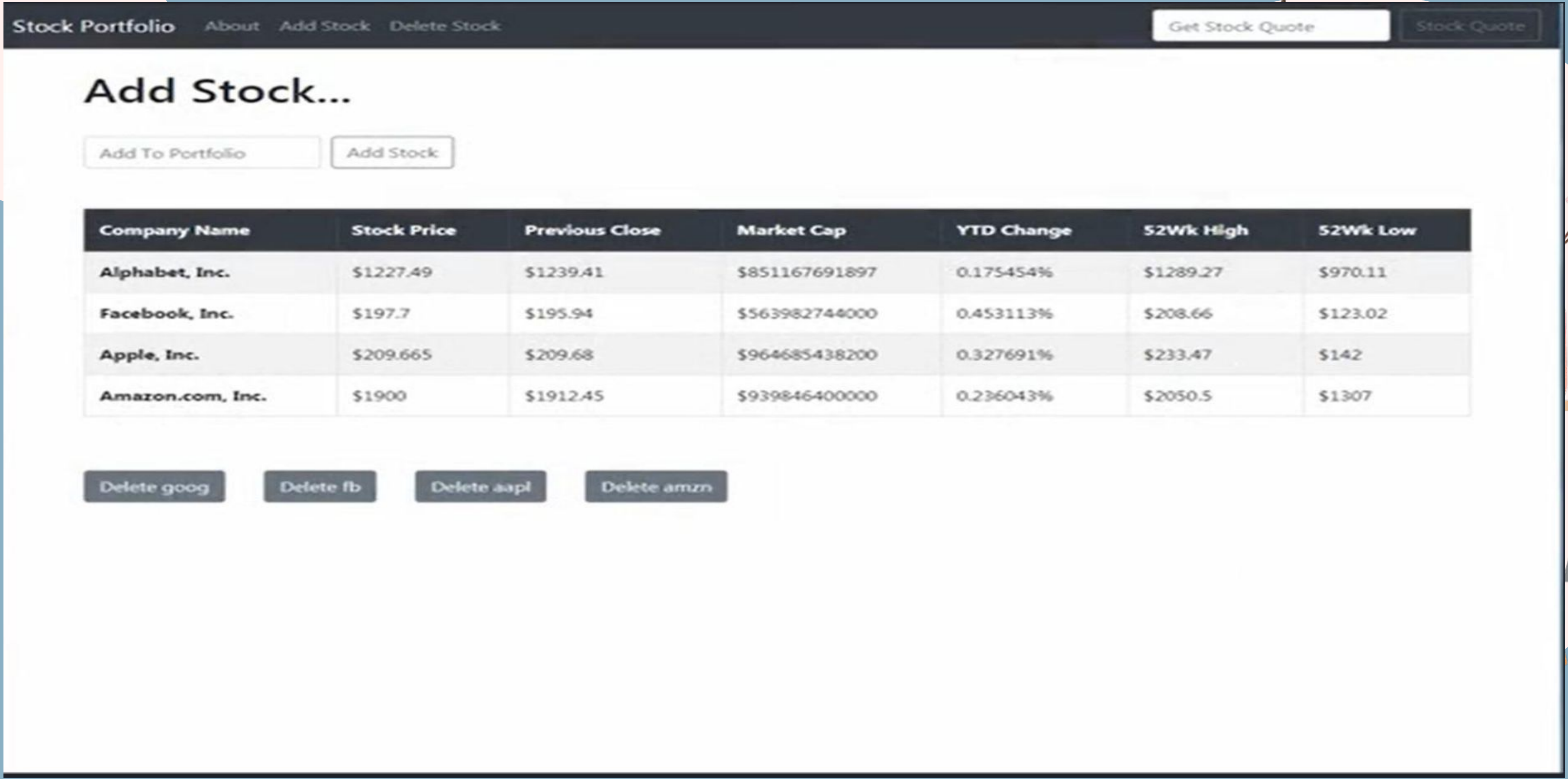


Deployment Diagram – A UML deployment diagram is a diagram that shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams are a kind of structure diagram used in modeling the physical aspects of an object-oriented system. The deployment diagram of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



Deployment Diagram for Stock Marketing App

Sample Frontend Design –



Add Stock...

Company Name	Stock Price	Previous Close	Market Cap	YTD Change	52Wk High	52Wk Low
Alphabet, Inc.	$1227.49	$1239.41	$851167691897	0.175454%	$1289.27	$970.11
Facebook, Inc.	$197.7	$195.94	$563982744000	0.453113%	$208.66	$123.02
Apple, Inc.	$209.665	$209.68	$964685438200	0.327691%	$233.47	$142
Amazon.com, Inc.	$1900	$1912.45	$939846400000	0.236043%	$2050.5	$1307

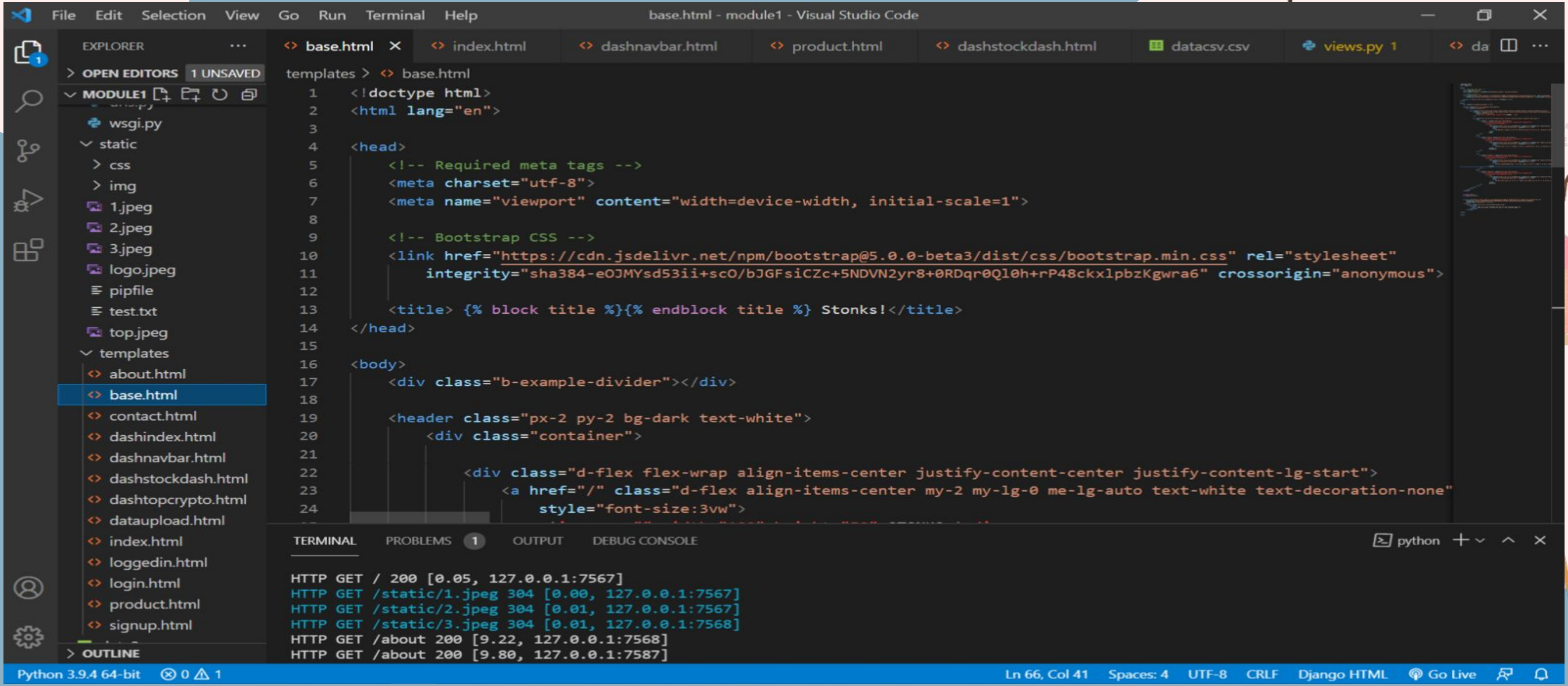
Implementation of Module 1

Code of Module 1

The code for the first module is can be found in the following link:

https://drive.google.com/drive/u/0/folders/1d114nrIDAKcCiGxYD5I6gC7bBieDIIsB

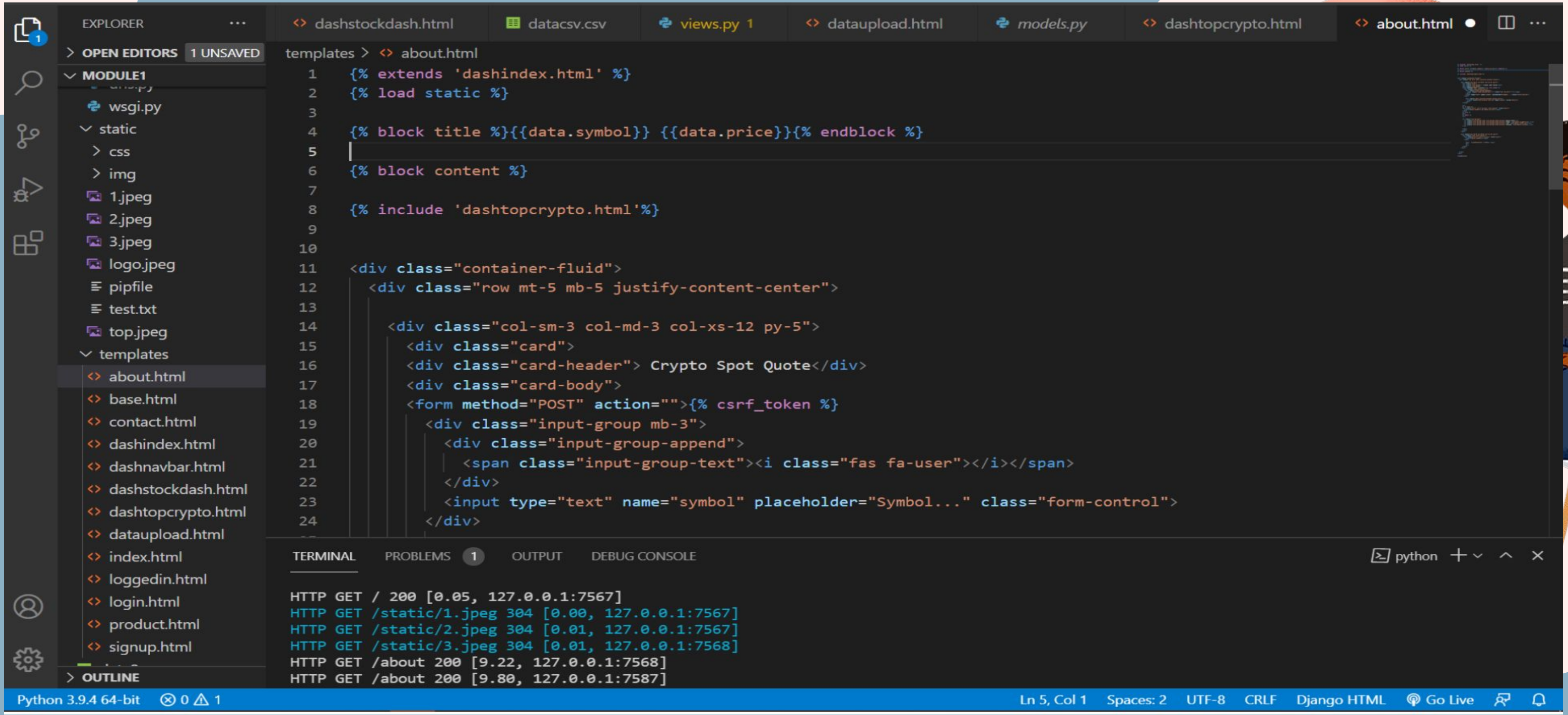
Code:



```html
<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

    <title> {% block title %}{% endblock title %} Stonks!</title>
</head>

<body>
    <div class="b-example-divider"></div>

    <header class="px-2 py-2 bg-dark text-white">
        <div class="container">

            <div class="d-flex flex-wrap align-items-center justify-content-center justify-content-lg-start">
                <a href="/" class="d-flex align-items-center my-2 my-lg-0 me-lg-auto text-white text-decoration-none"
                    style="font-size:3vw">
```

```
HTTP GET / 200 [0.05, 127.0.0.1:7567]
HTTP GET /static/1.jpeg 304 [0.00, 127.0.0.1:7567]
HTTP GET /static/2.jpeg 304 [0.01, 127.0.0.1:7567]
HTTP GET /static/3.jpeg 304 [0.01, 127.0.0.1:7568]
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

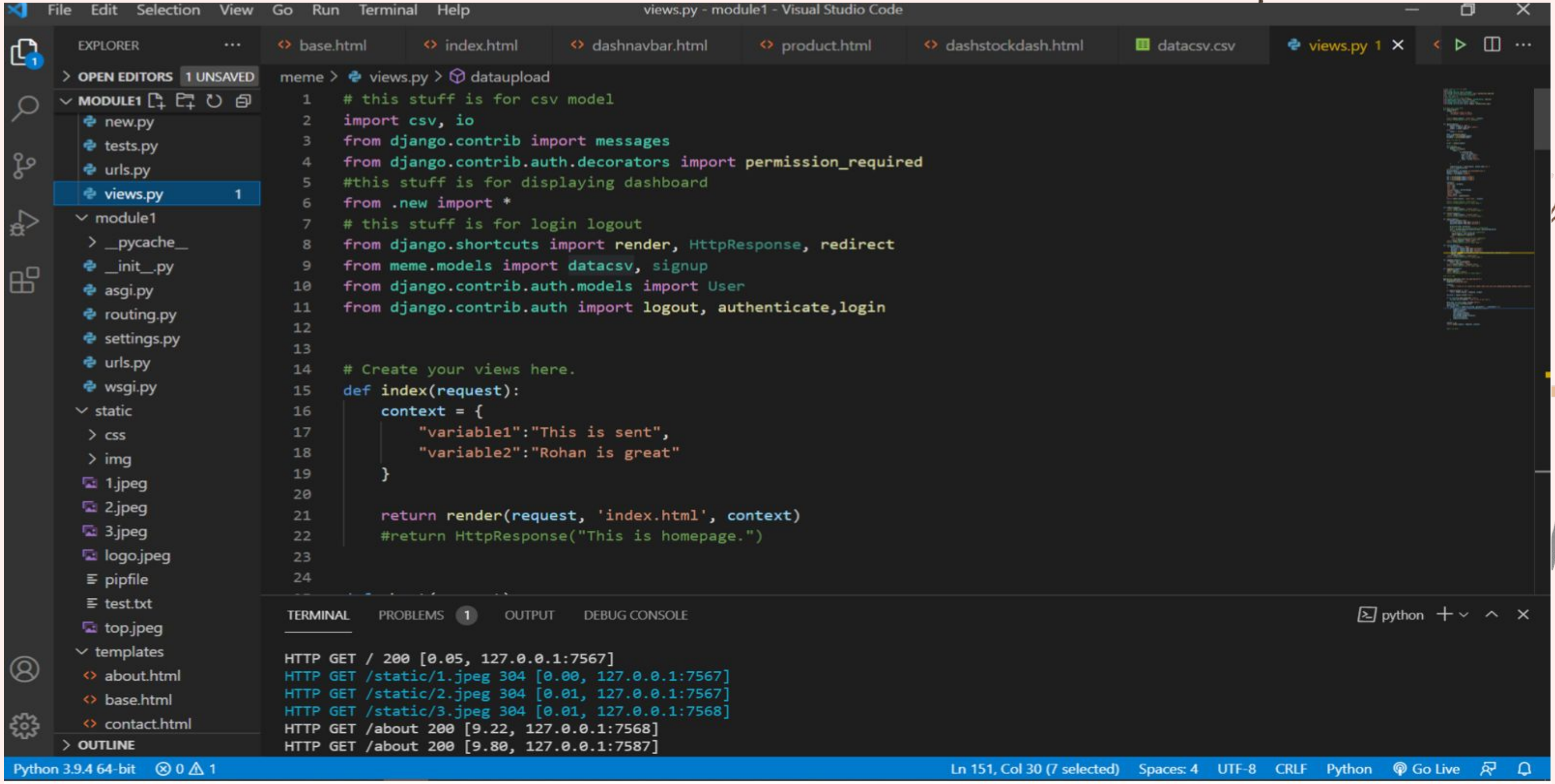
Code:



```
{% extends 'dashindex.html' %}
{% load static %}

{% block title %}{{data.symbol}} {{data.price}}{% endblock %}

{% block content %}

{% include 'dashtopcrypto.html'%}


<div class="container-fluid">
  <div class="row mt-5 mb-5 justify-content-center">

    <div class="col-sm-3 col-md-3 col-xs-12 py-5">
      <div class="card">
      <div class="card-header"> Crypto Spot Quote</div>
      <div class="card-body">
      <form method="POST" action="">{% csrf_token %}
        <div class="input-group mb-3">
          <div class="input-group-append">
            <span class="input-group-text"><i class="fas fa-user"></i></span>
          </div>
          <input type="text" name="symbol" placeholder="Symbol..." class="form-control">
        </div>
```

```
HTTP GET / 200 [0.05, 127.0.0.1:7567]
HTTP GET /static/1.jpeg 304 [0.00, 127.0.0.1:7567]
HTTP GET /static/2.jpeg 304 [0.01, 127.0.0.1:7567]
HTTP GET /static/3.jpeg 304 [0.01, 127.0.0.1:7568]
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

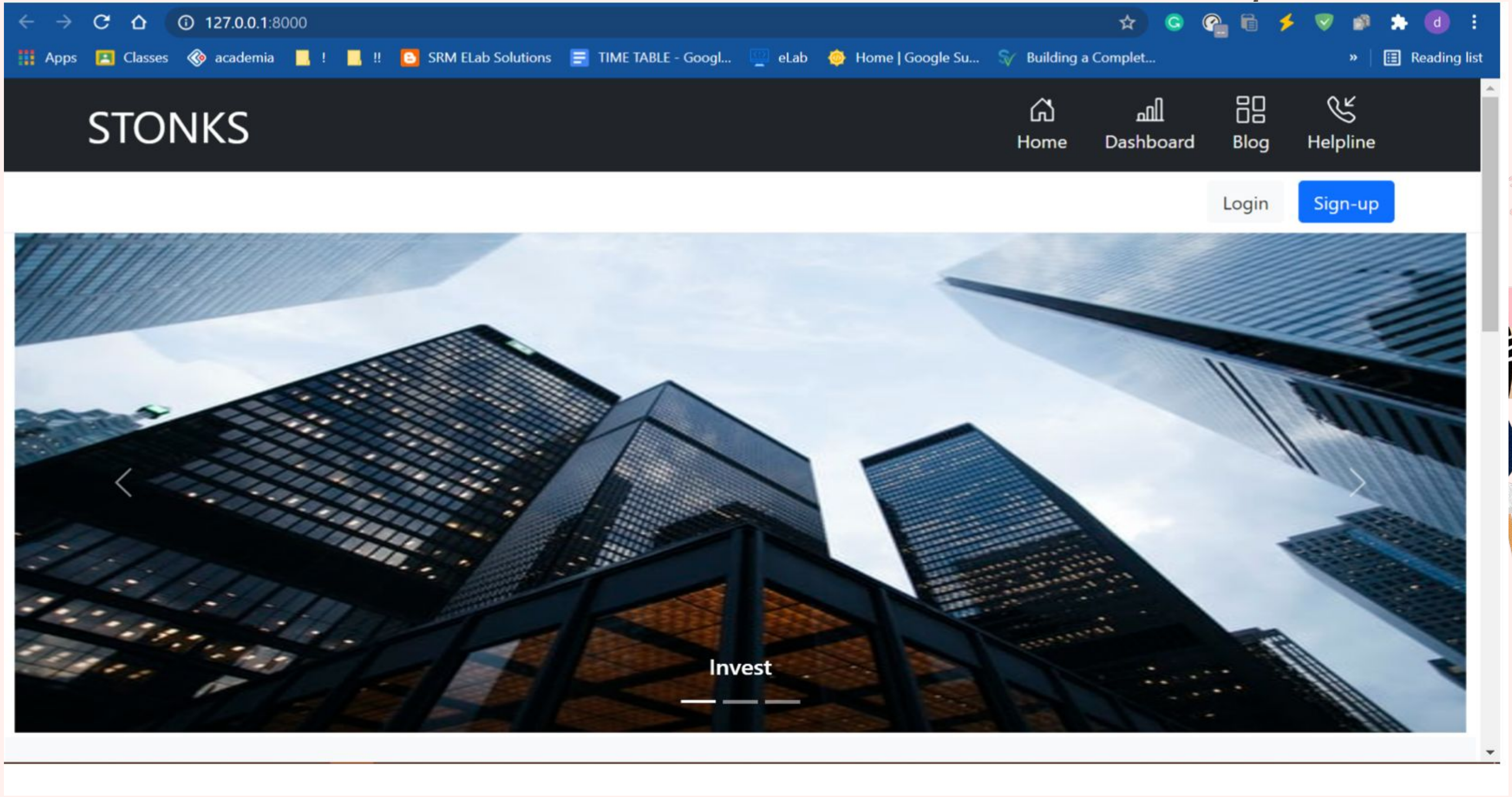
Code:



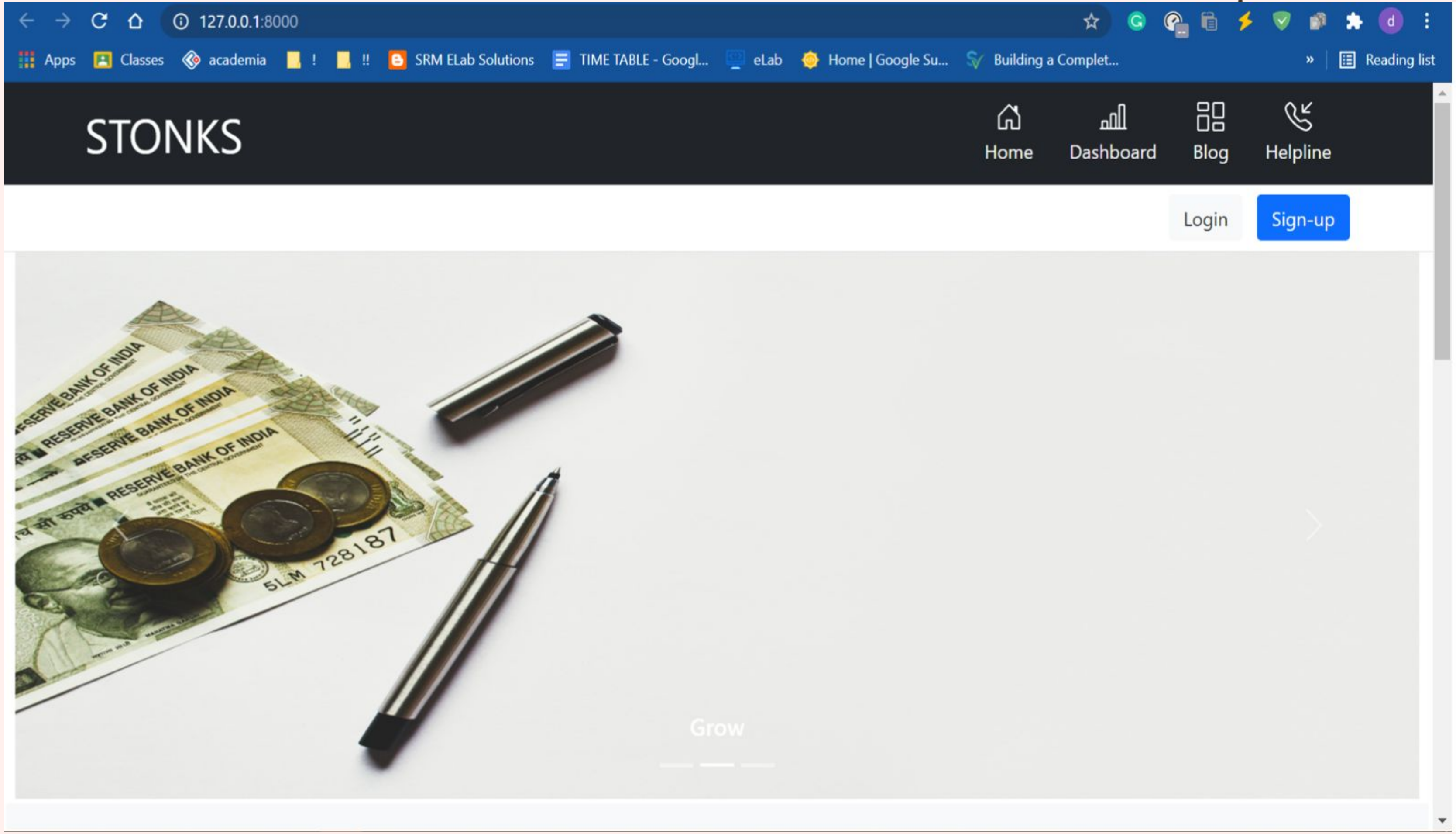
```python
# this stuff is for csv model
import csv, io
from django.contrib import messages
from django.contrib.auth.decorators import permission_required
#this stuff is for displaying dashboard
from .new import *
# this stuff is for login logout
from django.shortcuts import render, HttpResponse, redirect
from meme.models import datacsv, signup
from django.contrib.auth.models import User
from django.contrib.auth import logout, authenticate,login


# Create your views here.
def index(request):
    context = {
        "variable1":"This is sent",
        "variable2":"Rohan is great"
    }

    return render(request, 'index.html', context)
    #return HttpResponse("This is homepage.")

```

```
HTTP GET / 200 [0.05, 127.0.0.1:7567]
HTTP GET /static/1.jpeg 304 [0.00, 127.0.0.1:7567]
HTTP GET /static/2.jpeg 304 [0.01, 127.0.0.1:7567]
HTTP GET /static/3.jpeg 304 [0.01, 127.0.0.1:7568]
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

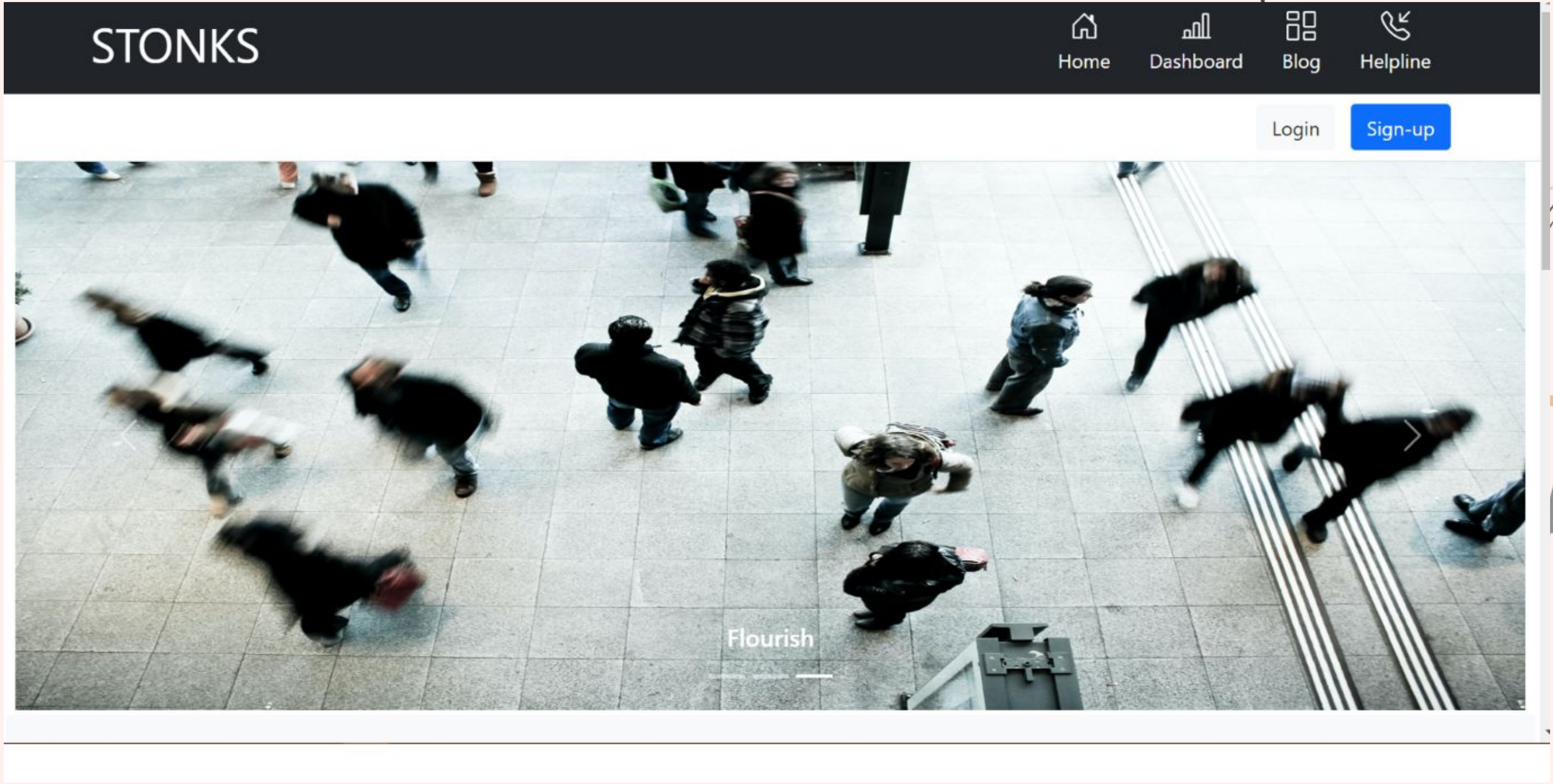
Frontend


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

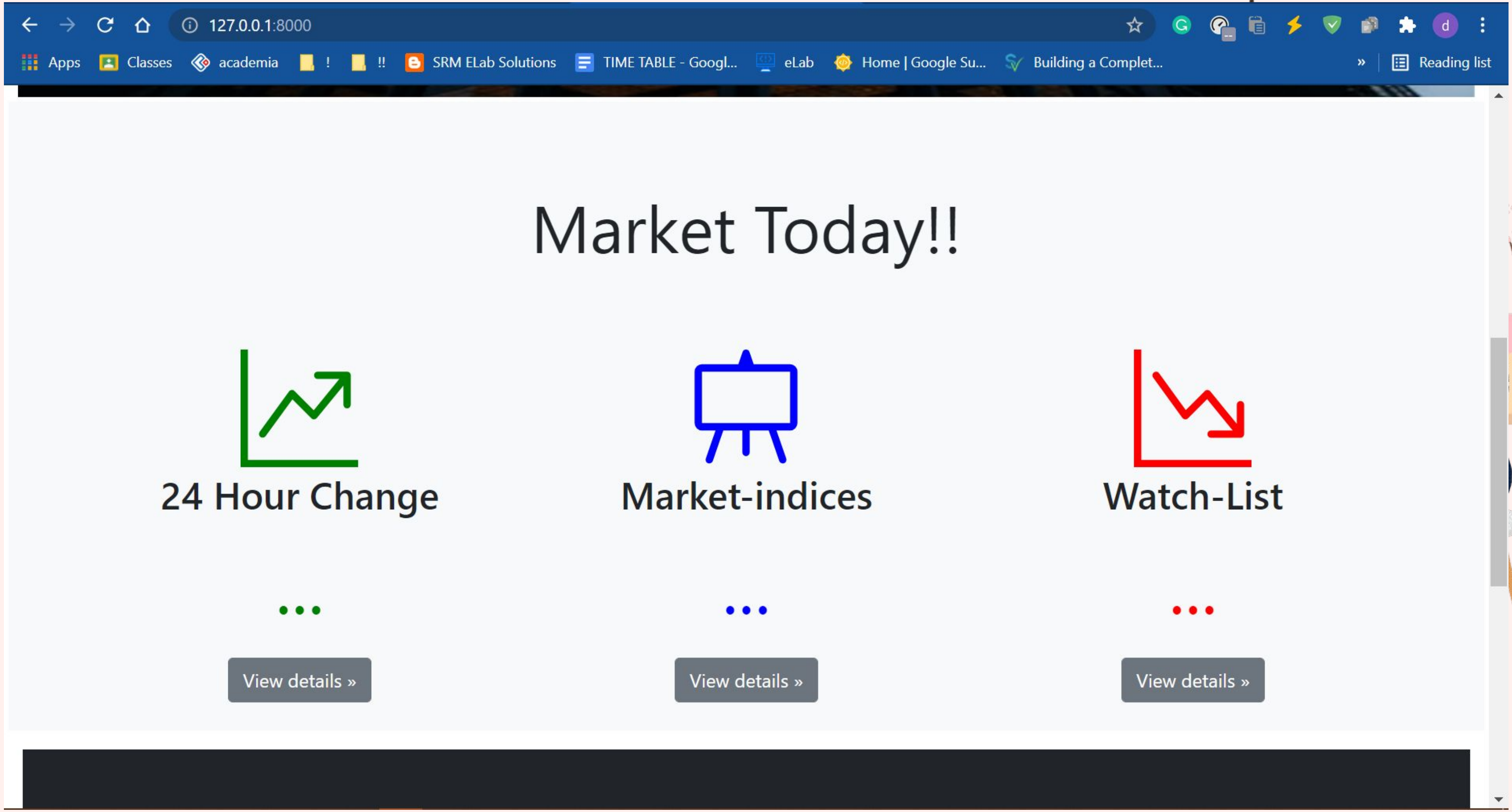
Frontend


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


127.0.0.1:8000

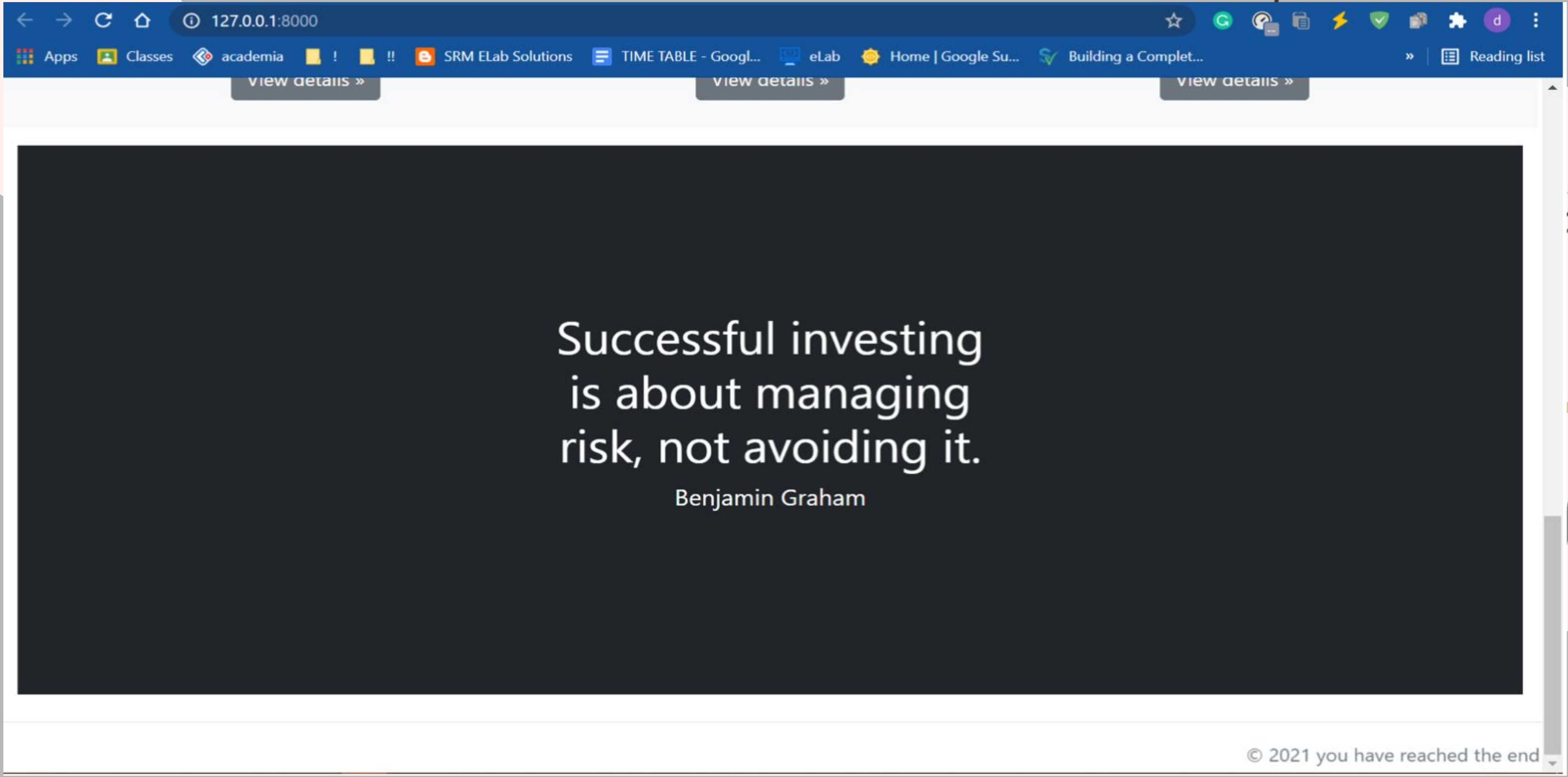
Market Today!!

24 Hour Change

Market-indices

Watch-List

Frontend


127.0.0.1:8000

Apps · Classes · academia · ! · !! · SRM ELab Solutions · TIME TABLE - Googl... · eLab · Home | Google Su... · Building a Complet... · Reading list

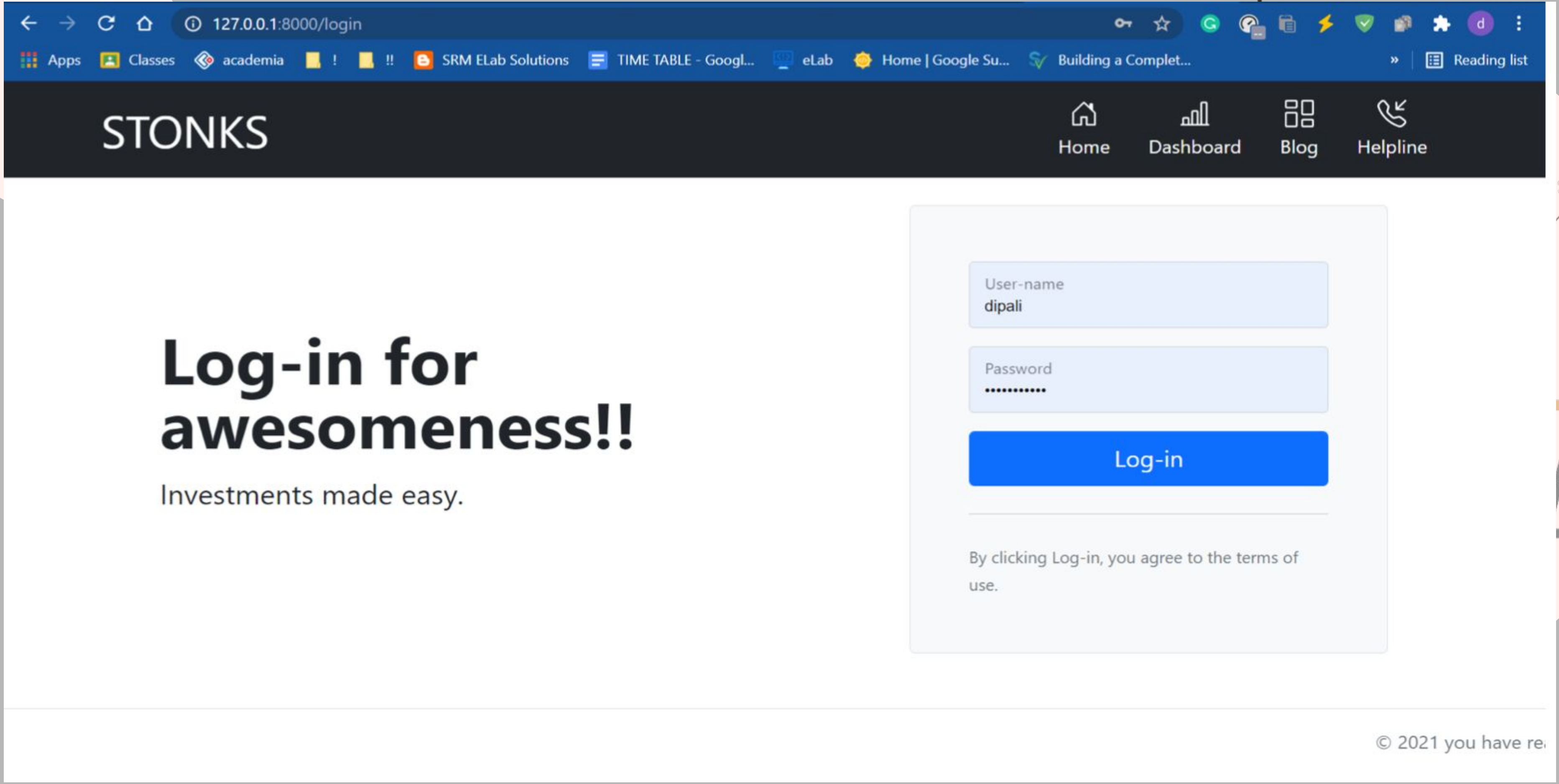
View details » · View details » · View details »

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

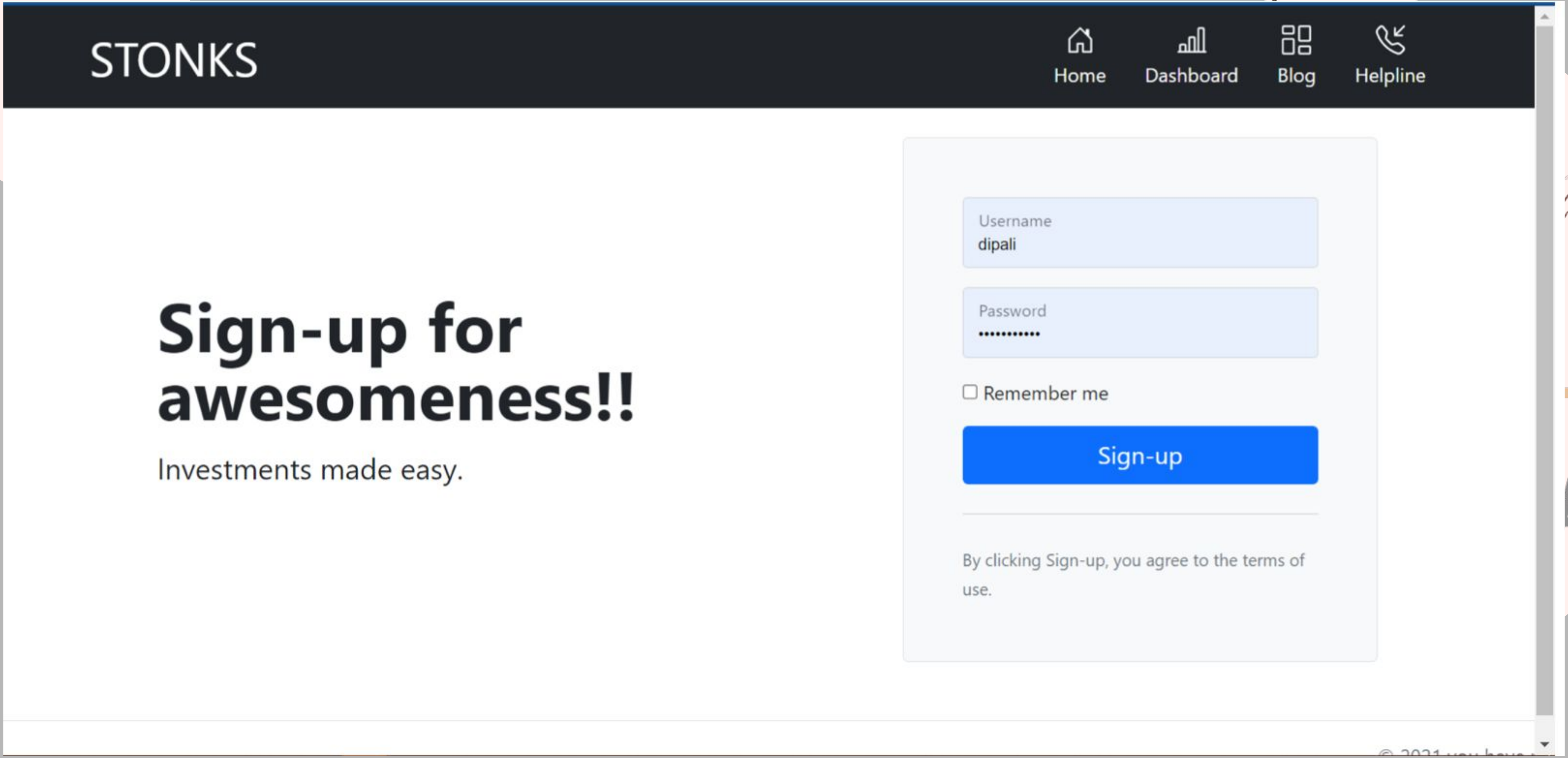
Password
••••••••••••

Log-in

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

© 2021 you have re

Frontend



STONKS

Home Dashboard Blog Helpline

Sign-up for awesomeness!!

Investments made easy.

Username: dipali

Password: •••••••••••

☐ Remember me

Sign-up

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

Implementation of Module 2

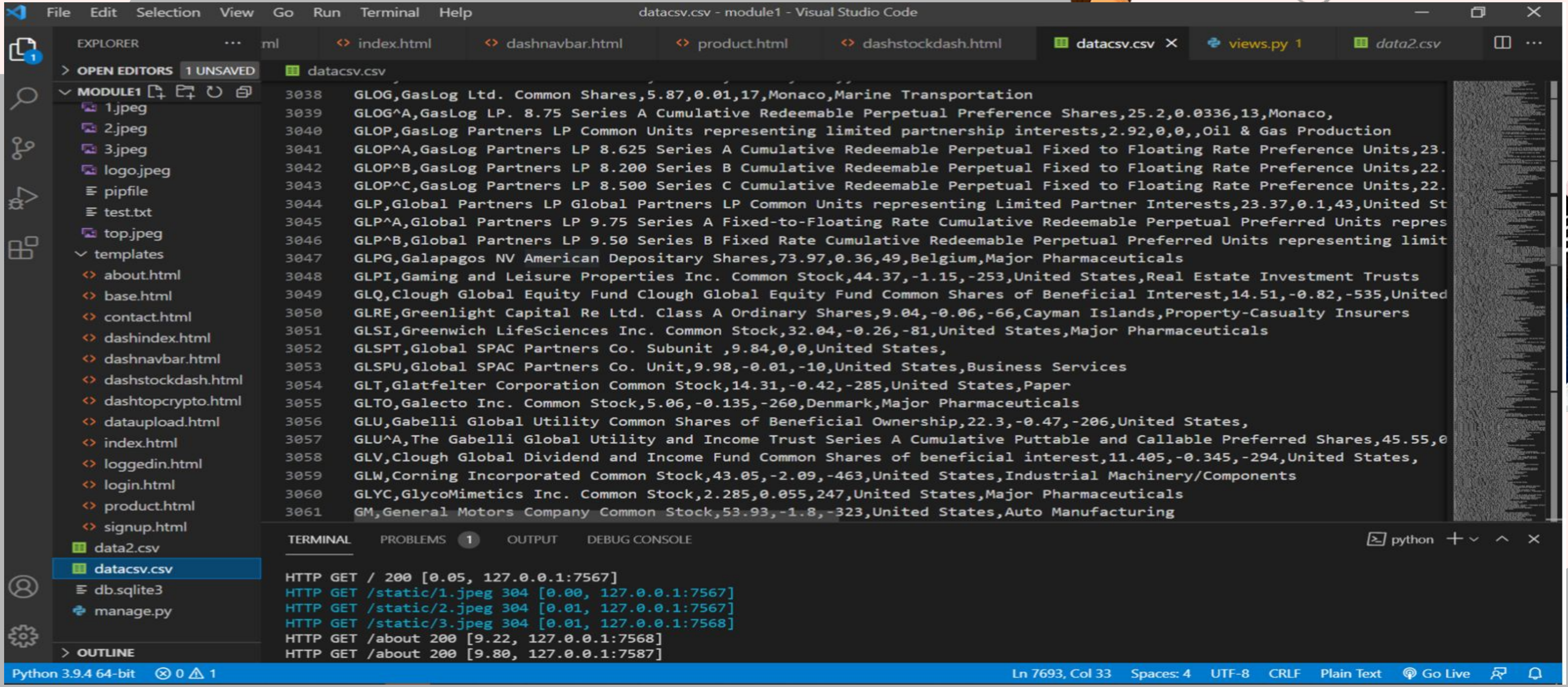
Code of Module 2

The code and the data for back-end can be found in the following link:

https://drive.google.com/drive/u/0/folders/1d114nrIDAKcCiGxYD5l6gC7bBieDIIsB

The data of around 9000 companies in stock-market:

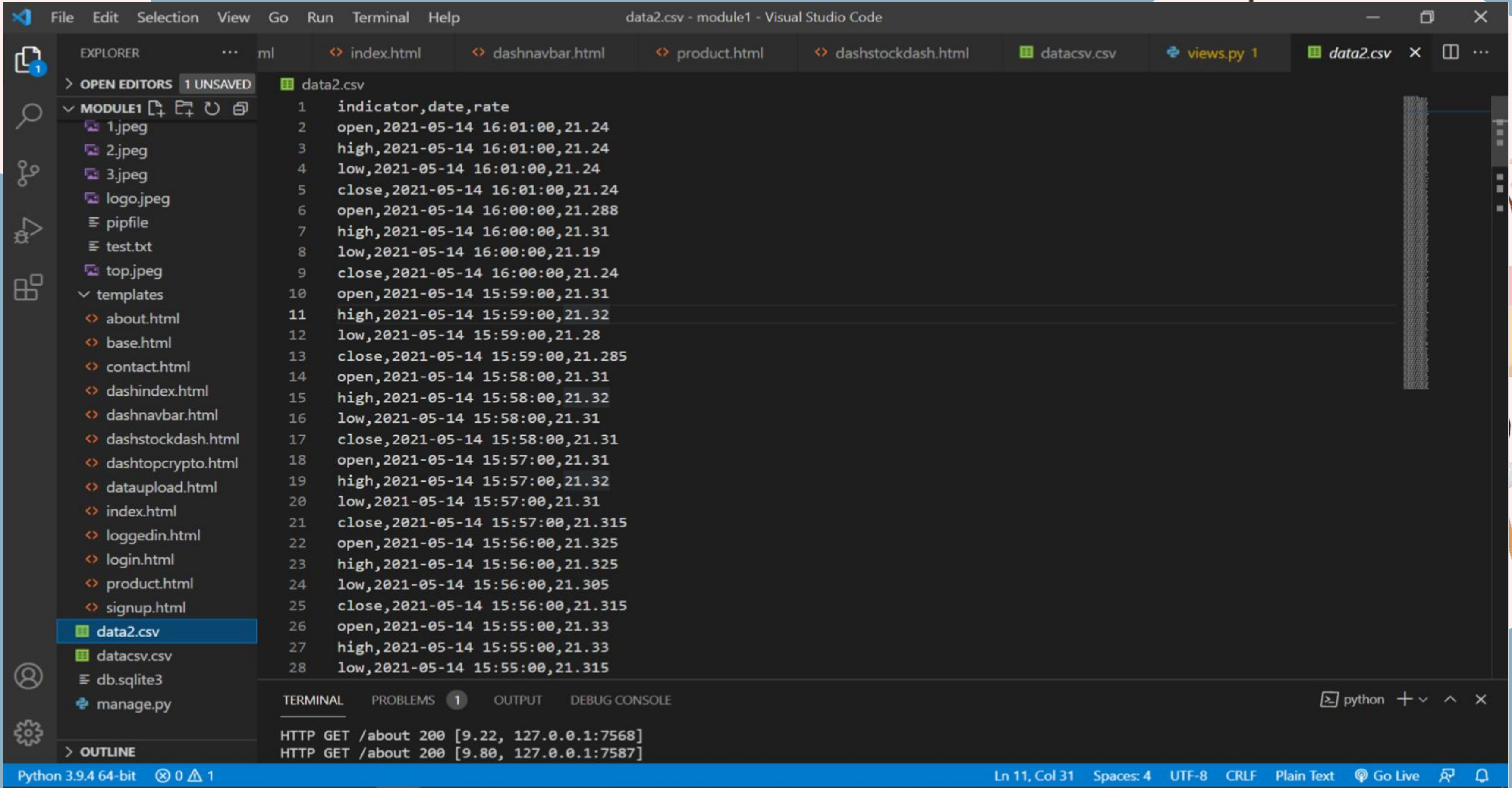
The Data of around 7000 companies in Stock Market



```
GLOG,GasLog Ltd. Common Shares,5.87,0.01,17,Monaco,Marine Transportation
GLOG^A,GasLog LP. 8.75 Series A Cumulative Redeemable Perpetual Preference Shares,25.2,0.0336,13,Monaco,
GLOP,GasLog Partners LP Common Units representing limited partnership interests,2.92,0,0,,Oil & Gas Production
GLOP^A,GasLog Partners LP 8.625 Series A Cumulative Redeemable Perpetual Fixed to Floating Rate Preference Units,23.
GLOP^B,GasLog Partners LP 8.200 Series B Cumulative Redeemable Perpetual Fixed to Floating Rate Preference Units,22.
GLOP^C,GasLog Partners LP 8.500 Series C Cumulative Redeemable Perpetual Fixed to Floating Rate Preference Units,22.
GLP,Global Partners LP Global Partners LP Common Units representing Limited Partner Interests,23.37,0.1,43,United St
GLP^A,Global Partners LP 9.75 Series A Fixed-to-Floating Rate Cumulative Redeemable Perpetual Preferred Units repres
GLP^B,Global Partners LP 9.50 Series B Fixed Rate Cumulative Redeemable Perpetual Preferred Units representing limit
GLPG,Galapagos NV American Depositary Shares,73.97,0.36,49,Belgium,Major Pharmaceuticals
GLPI,Gaming and Leisure Properties Inc. Common Stock,44.37,-1.15,-253,United States,Real Estate Investment Trusts
GLQ,Clough Global Equity Fund Clough Global Equity Fund Common Shares of Beneficial Interest,14.51,-0.82,-535,United
GLRE,Greenlight Capital Re Ltd. Class A Ordinary Shares,9.04,-0.06,-66,Cayman Islands,Property-Casualty Insurers
GLSI,Greenwich LifeSciences Inc. Common Stock,32.04,-0.26,-81,United States,Major Pharmaceuticals
GLSPT,Global SPAC Partners Co. Subunit ,9.84,0,0,United States,
GLSPU,Global SPAC Partners Co. Unit,9.98,-0.01,-10,United States,Business Services
GLT,Glatfelter Corporation Common Stock,14.31,-0.42,-285,United States,Paper
GLTO,Galecto Inc. Common Stock,5.06,-0.135,-260,Denmark,Major Pharmaceuticals
GLU,Gabelli Global Utility Common Shares of Beneficial Ownership,22.3,-0.47,-206,United States,
GLU^A,The Gabelli Global Utility and Income Trust Series A Cumulative Puttable and Callable Preferred Shares,45.55,0
GLV,Clough Global Dividend and Income Fund Common Shares of beneficial interest,11.405,-0.345,-294,United States,
GLW,Corning Incorporated Common Stock,43.05,-2.09,-463,United States,Industrial Machinery/Components
GLYC,GlycoMimetics Inc. Common Stock,2.285,0.055,247,United States,Major Pharmaceuticals
GM,General Motors Company Common Stock,53.93,-1.8,-323,United States,Auto Manufacturing
```

```
HTTP GET / 200 [0.05, 127.0.0.1:7567]
HTTP GET /static/1.jpeg 304 [0.00, 127.0.0.1:7567]
HTTP GET /static/2.jpeg 304 [0.01, 127.0.0.1:7567]
HTTP GET /static/3.jpeg 304 [0.01, 127.0.0.1:7568]
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

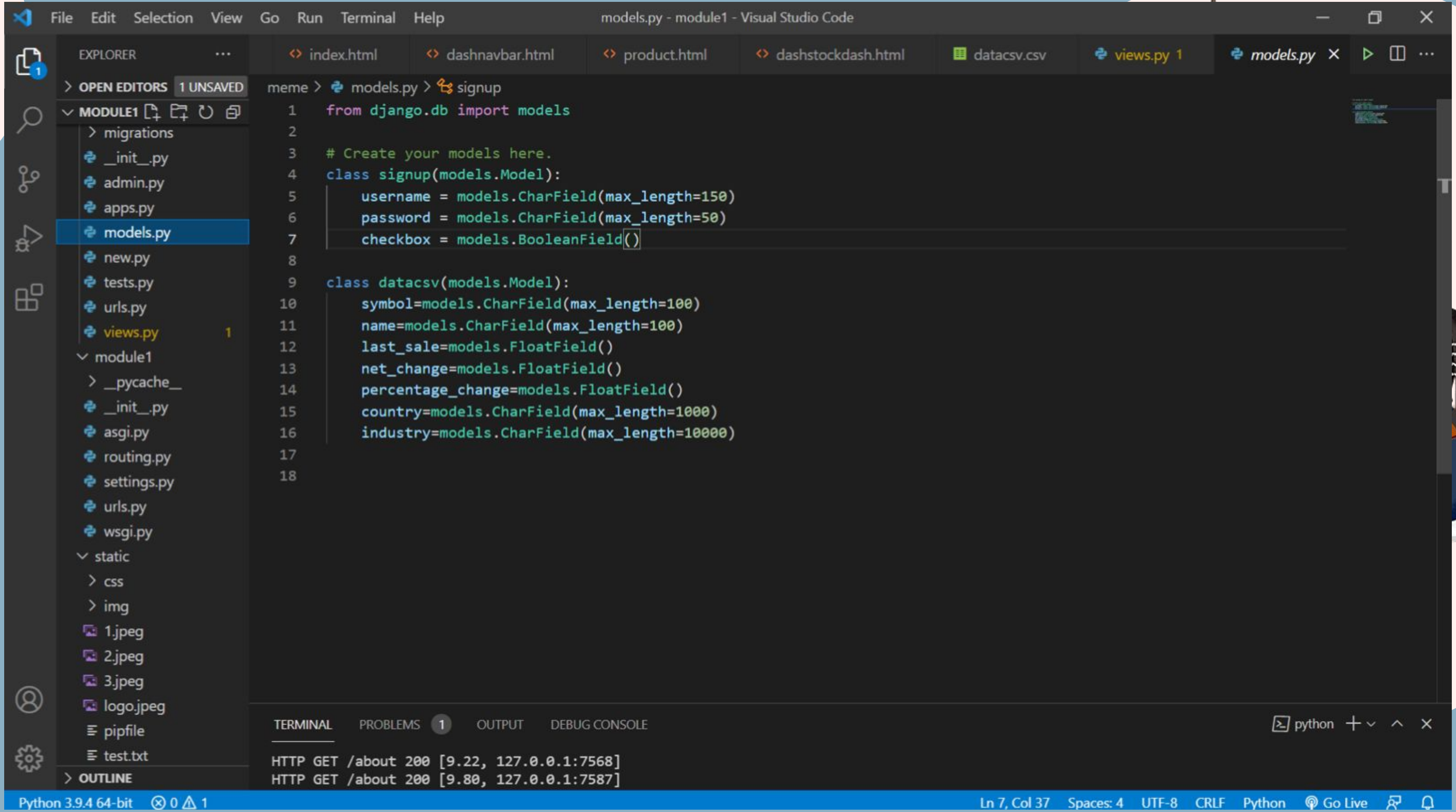
The Live data of highs and lows of each stock:



```
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]
```

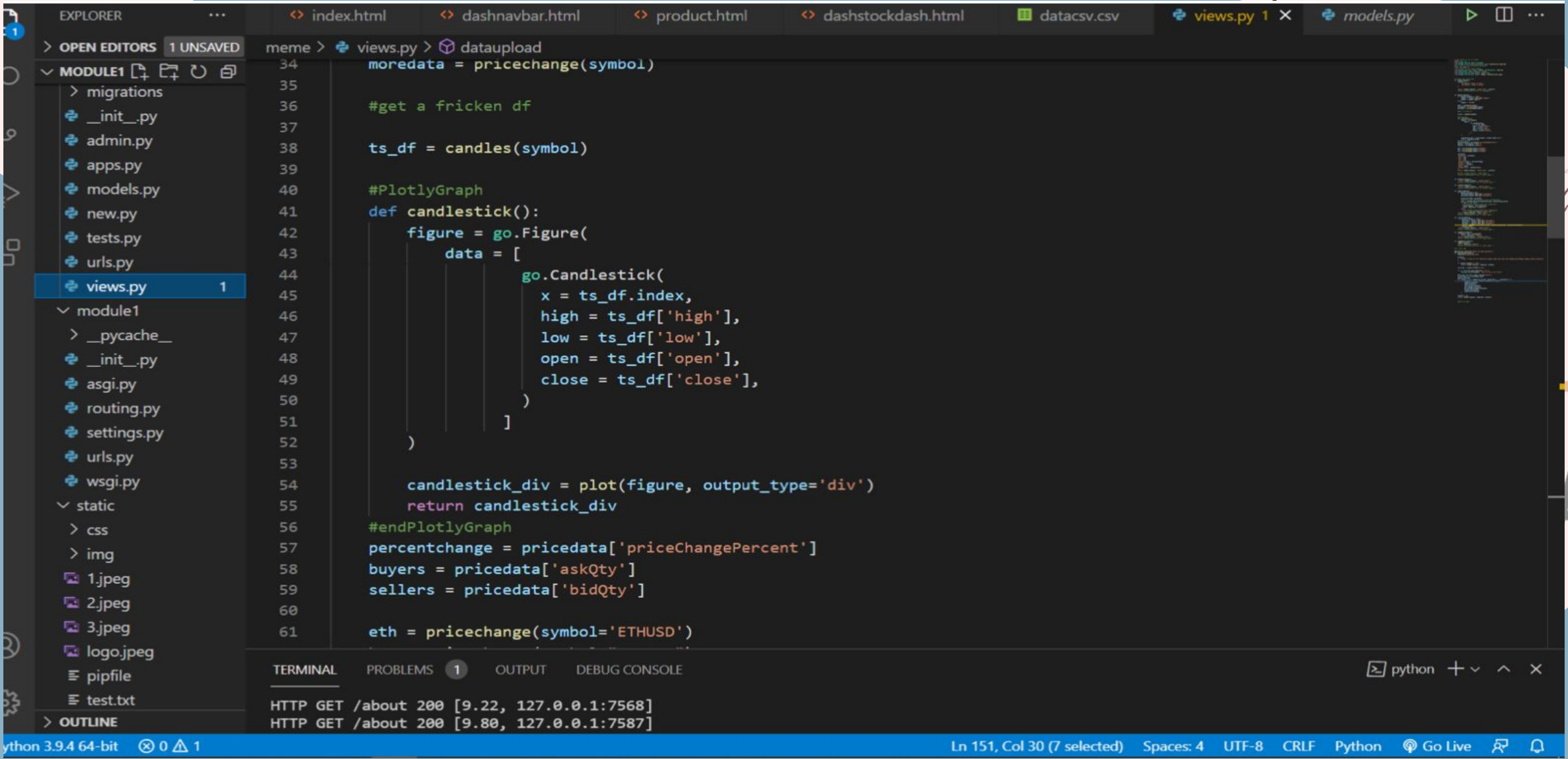
Code for backend -



```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]
```

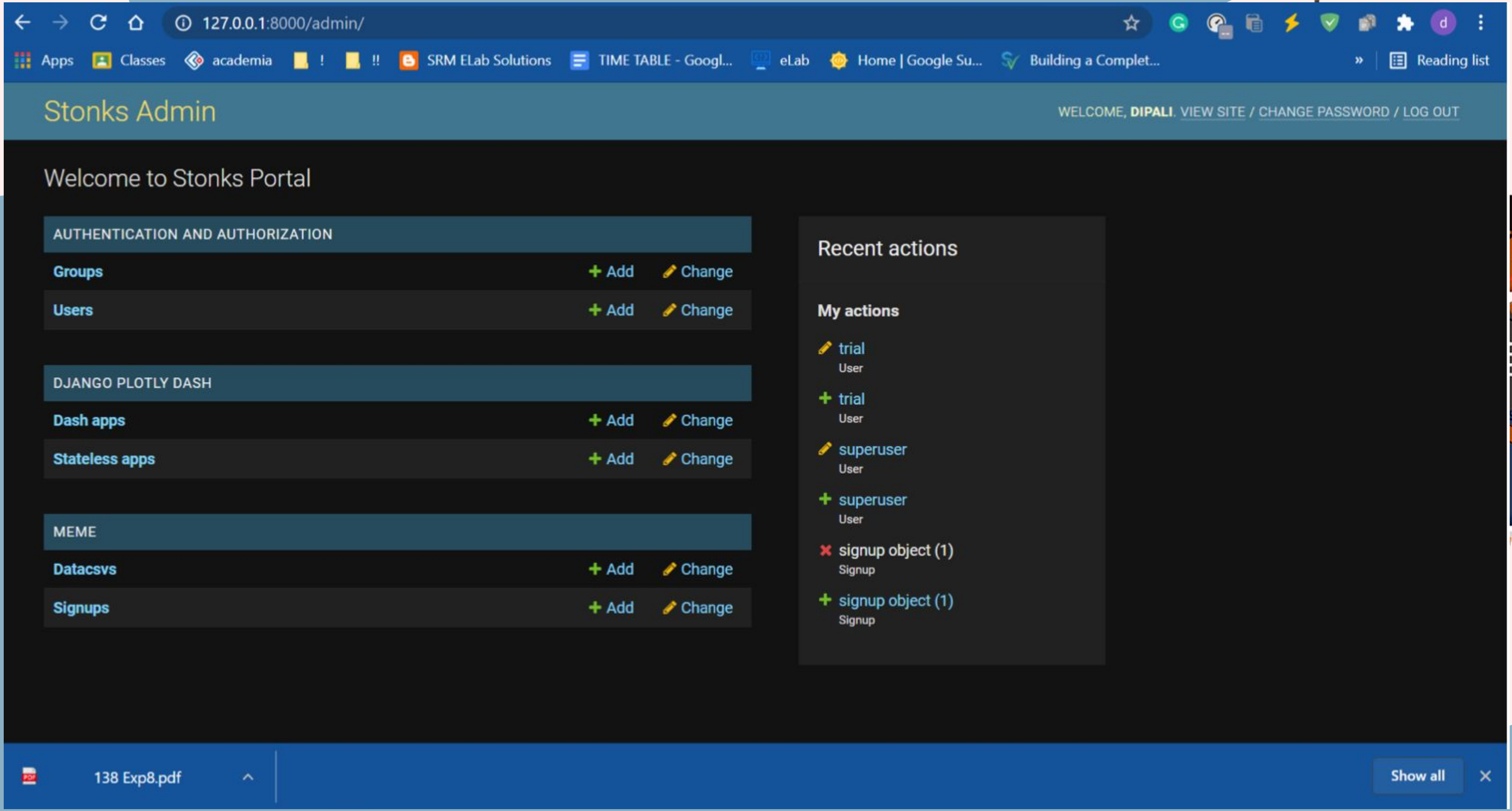
Code for backend -



```python
    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



127.0.0.1:8000/admin/

Apps Classes academia ! !! SRM ELab Solutions TIME TABLE - Googl... eLab Home | Google Su... Building a Complet... Reading list

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

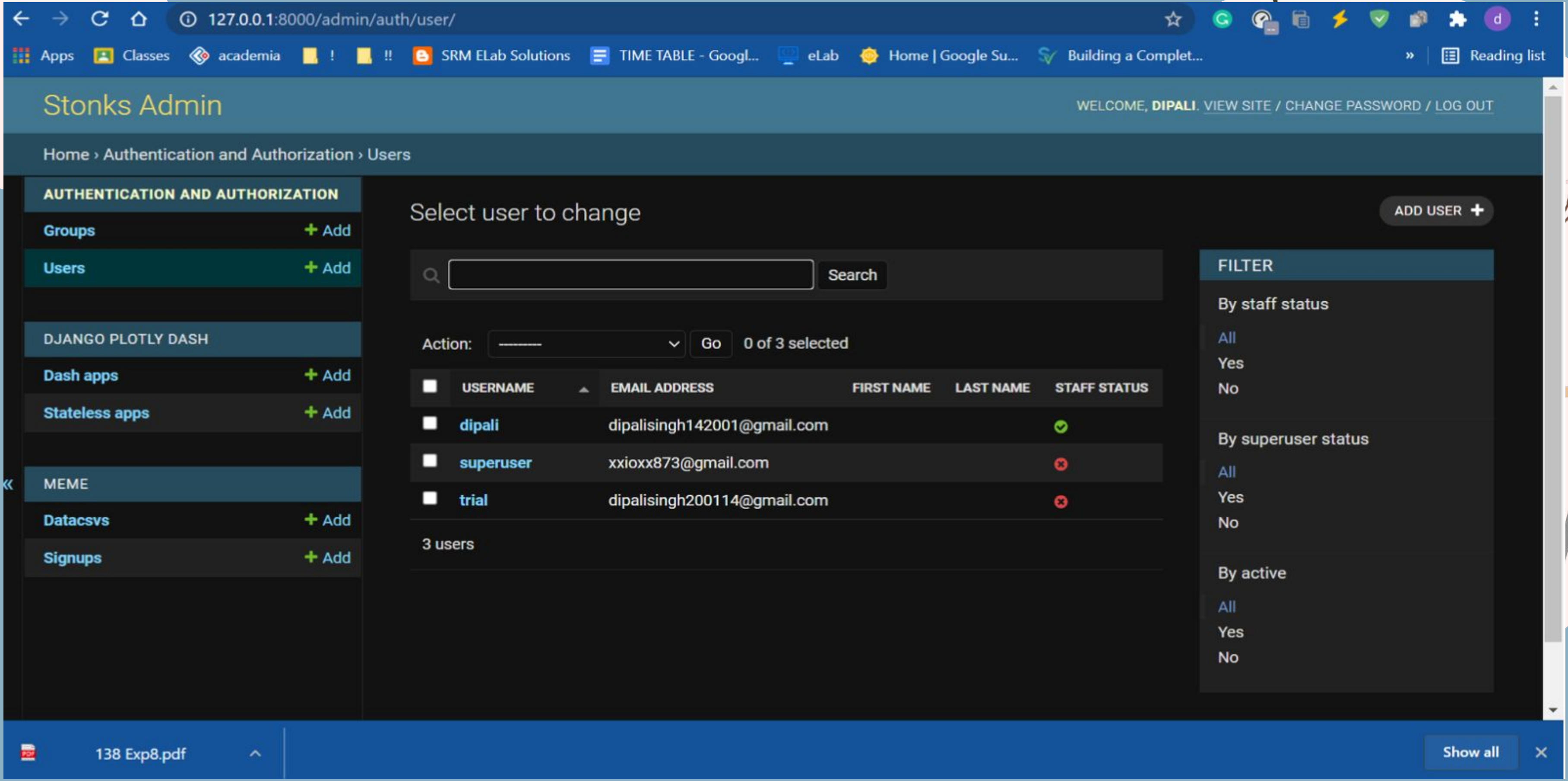
My actions

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

138 Exp8.pdf

Show all

Result of Module 2



127.0.0.1:8000/admin/auth/user/

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

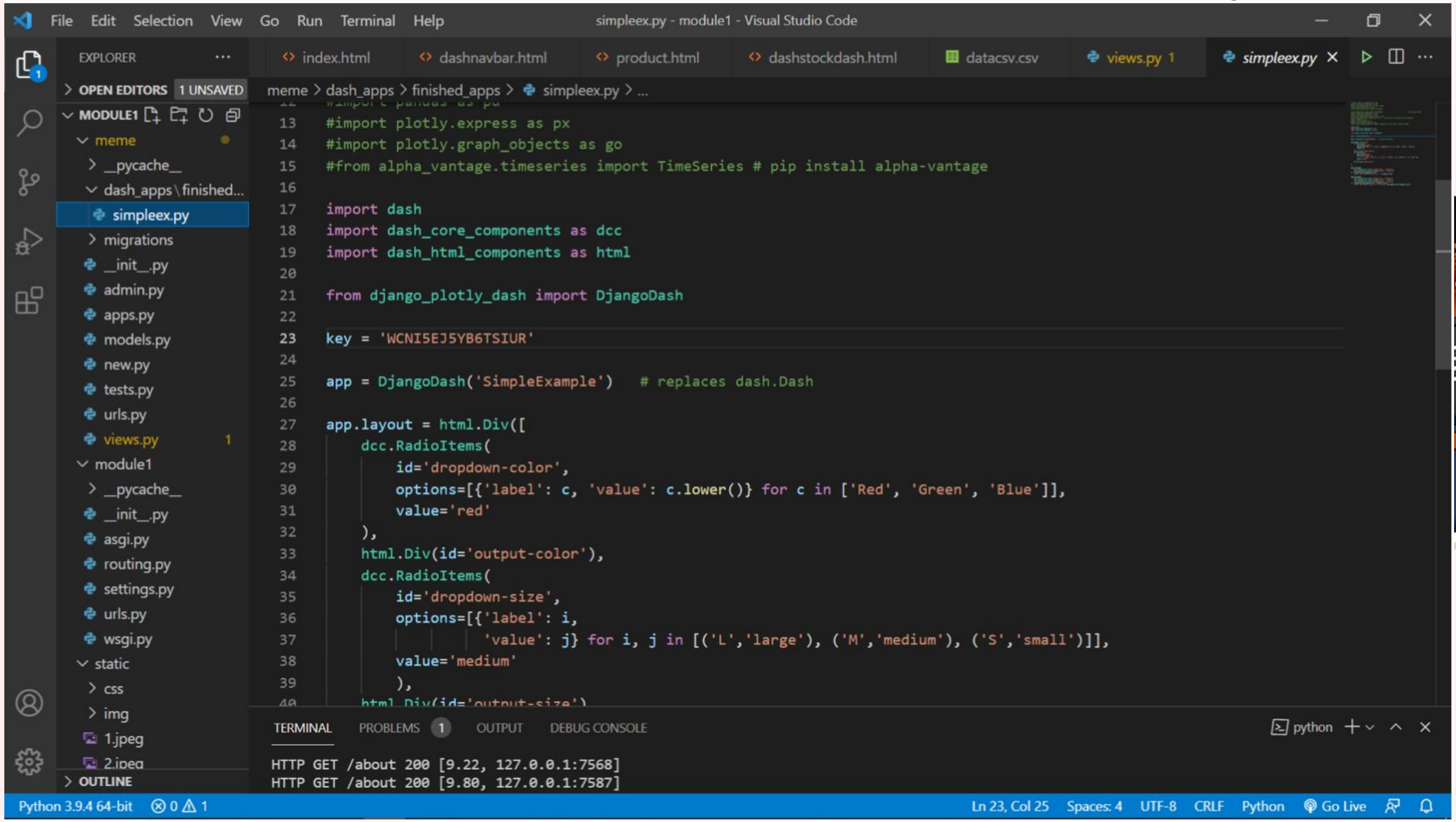
By active
- All
- Yes
- No

Implementation of Module 3

Code of Module 3:

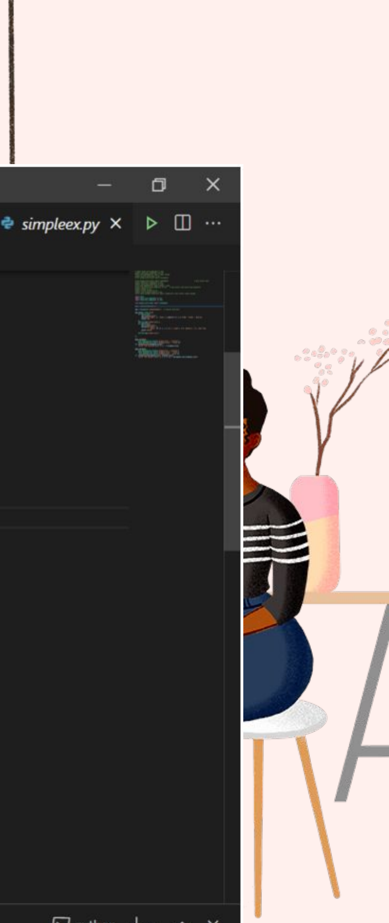
Module 3 (Live API parsing to show live graphs) was implemented using the following code: Link for detailed code: https://drive.google.com/drive/u/0/folders/1d114nrIDAKcCiGxYD5I6gC7bBieDIIsB

Code -

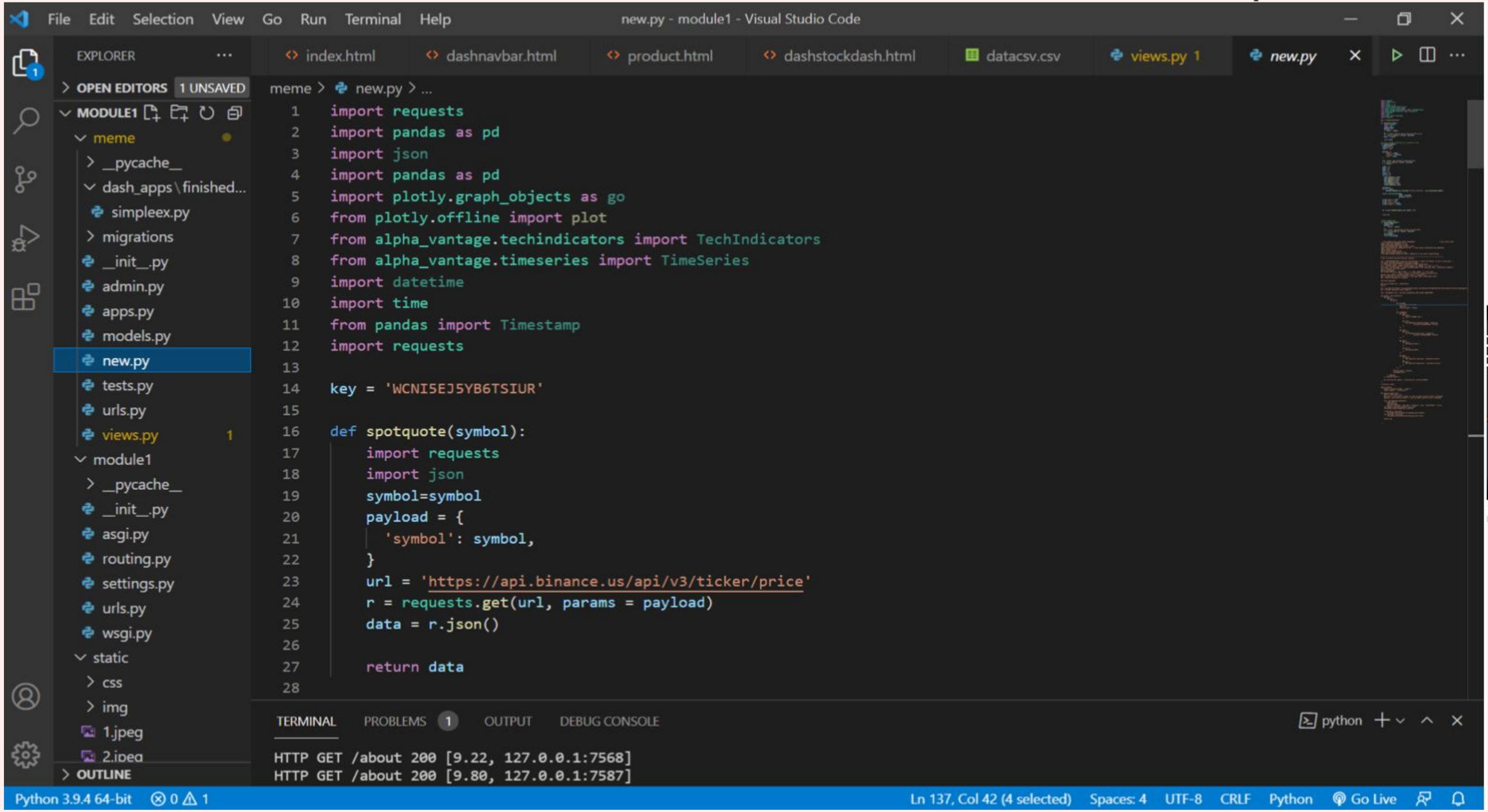


```python
#import pandas as pd
#import plotly.express as px
#import plotly.graph_objects as go
#from alpha_vantage.timeseries import TimeSeries # pip install alpha-vantage

import dash
import dash_core_components as dcc
import dash_html_components as html

from django_plotly_dash import DjangoDash

key = 'WCNI5EJ5YB6TSIUR'

app = DjangoDash('SimpleExample')   # replaces dash.Dash

app.layout = html.Div([
    dcc.RadioItems(
        id='dropdown-color',
        options=[{'label': c, 'value': c.lower()} for c in ['Red', 'Green', 'Blue']],
        value='red'
    ),
    html.Div(id='output-color'),
    dcc.RadioItems(
        id='dropdown-size',
        options=[{'label': i,
                  'value': j} for i, j in [('L','large'), ('M','medium'), ('S','small')]],
        value='medium'
    ),
    html.Div(id='output-size')
```

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



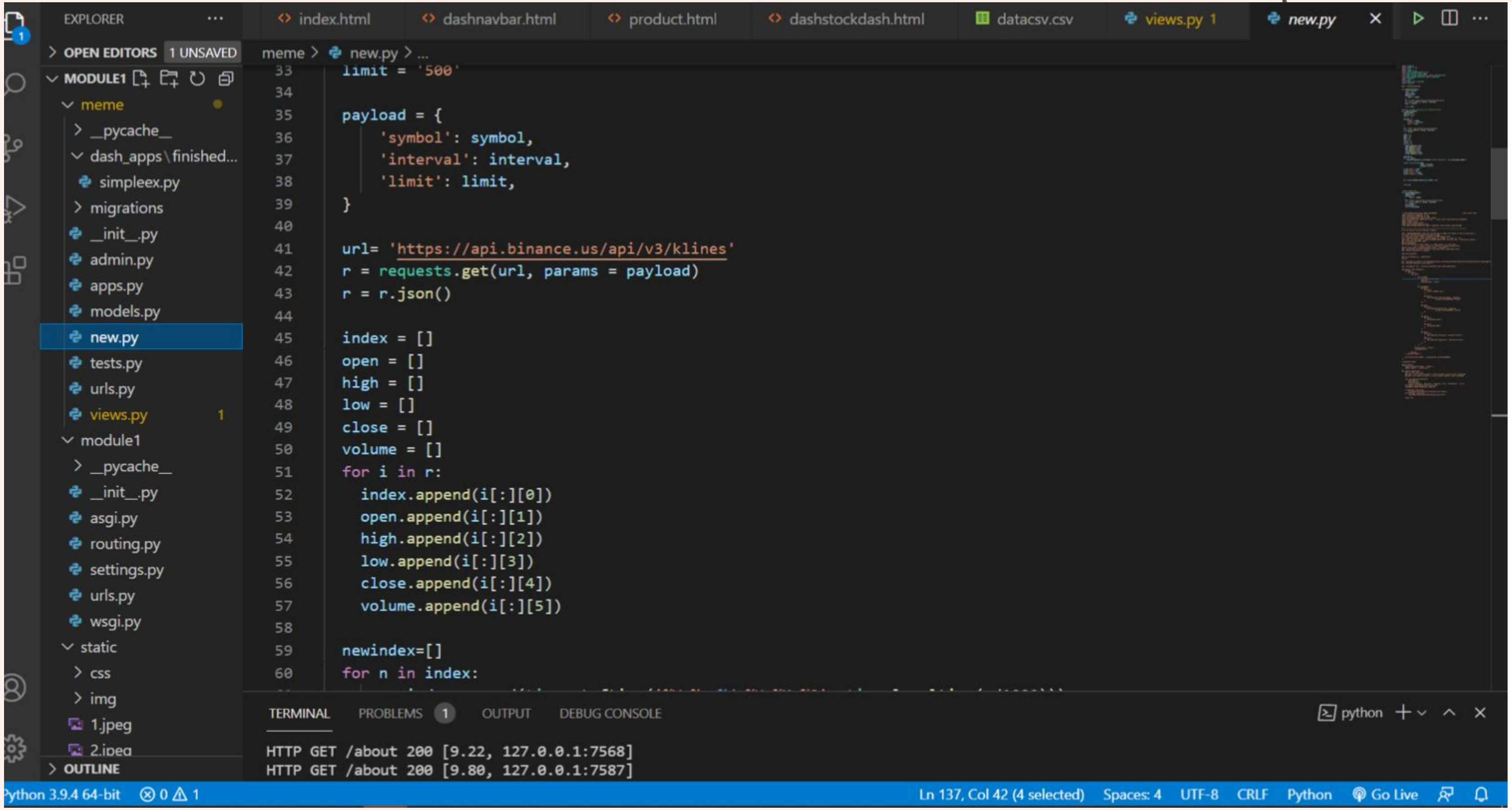
Code -



```python
import requests
import pandas as pd
import json
import pandas as pd
import plotly.graph_objects as go
from plotly.offline import plot
from alpha_vantage.techindicators import TechIndicators
from alpha_vantage.timeseries import TimeSeries
import datetime
import time
from pandas import Timestamp
import requests

key = 'WCNI5EJ5YB6TSIUR'

def spotquote(symbol):
    import requests
    import json
    symbol=symbol
    payload = {
      'symbol': symbol,
    }
    url = 'https://api.binance.us/api/v3/ticker/price'
    r = requests.get(url, params = payload)
    data = r.json()

    return data

```

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

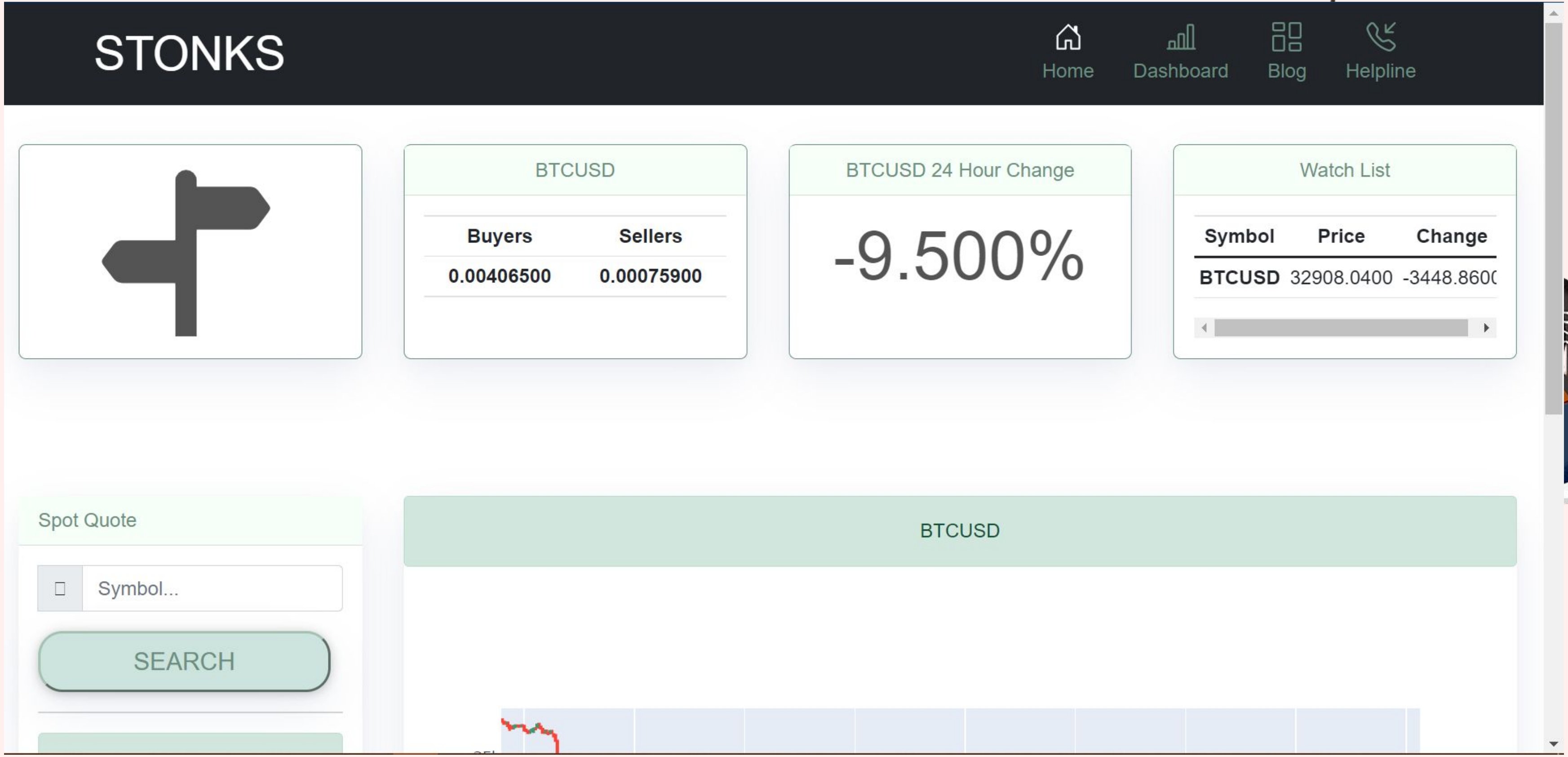
Code -



```python
limit = '500'

payload = {
    'symbol': symbol,
    'interval': interval,
    'limit': limit,
}

url= 'https://api.binance.us/api/v3/klines'
r = requests.get(url, params = payload)
r = r.json()

index = []
open = []
high = []
low = []
close = []
volume = []
for i in r:
  index.append(i[:][0])
  open.append(i[:][1])
  high.append(i[:][2])
  low.append(i[:][3])
  close.append(i[:][4])
  volume.append(i[:][5])

newindex=[]
for n in index:
```

```
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 3:



BTCUSD

Buyers	Sellers
0.00406500	0.00075900

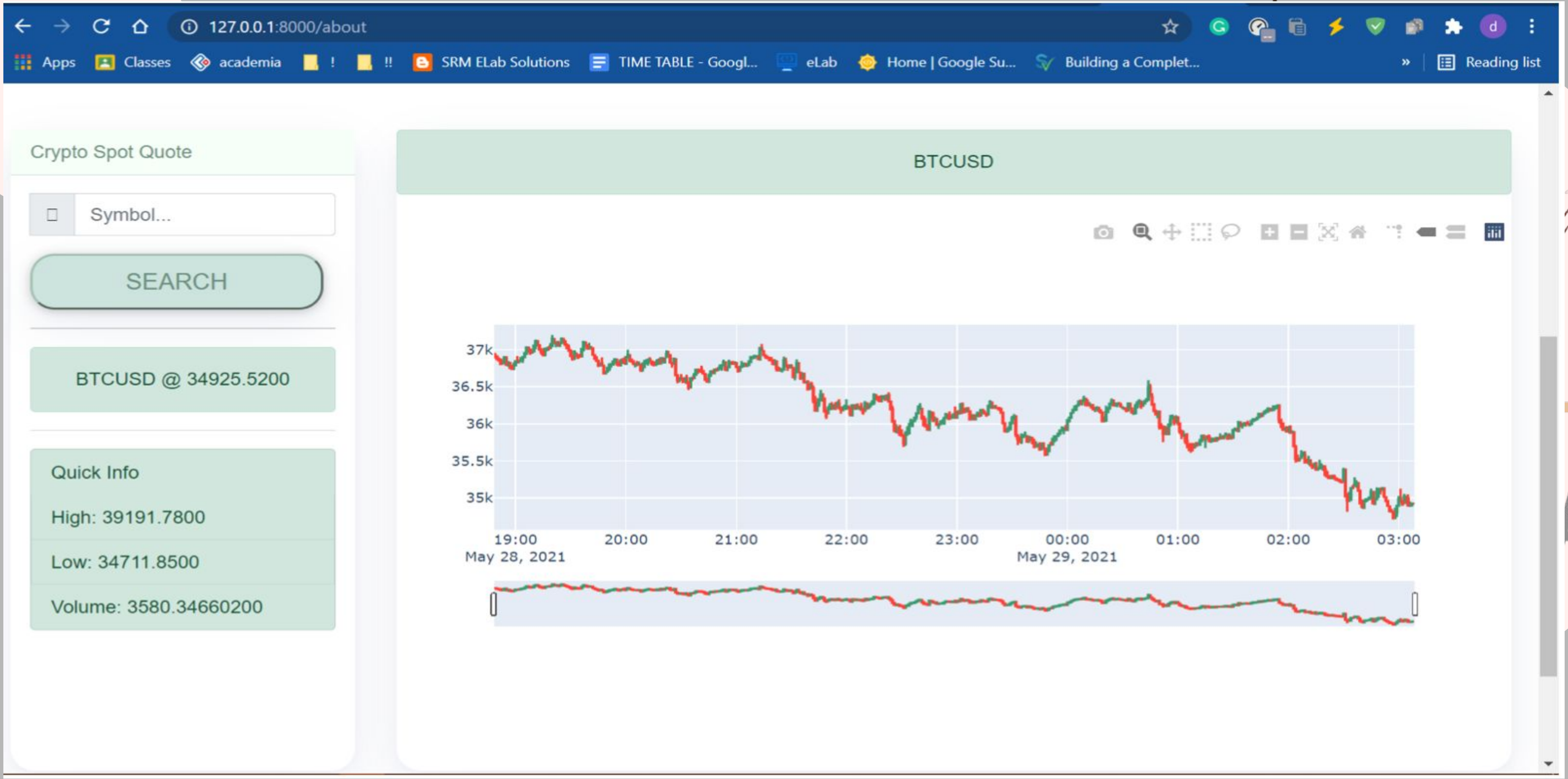
BTCUSD 24 Hour Change

-9.500%

Watch List

Symbol	Price	Change
BTCUSD	32908.0400	-3448.860(

Result of Module 3:



BTCUSD @ 34925.5200

Quick Info

High: 39191.7800

Low: 34711.8500

Volume: 3580.34660200

Master Test Plan and Test cases

Scope of Testing

Functional: The following modules need to be tested:

1. Front-end
2. Back-end(migrations)
3. Live API parsing.

Non-Functional: The non-functional modules are:

1. Blog page
2. Helpline page

Types of Testing, Methodology and Tools

Category	Methodology	Tools Required
Functional Requirements	Manual	Django Frame

Functional Test Cases

Test ID#	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome
1.	Verify user registration	Accept Valid usernam e on page#1	1. User clicks on user registration link 2. Enter the username on the box	User should be taken to login page	successful
2.	Verify live data parsi	Parse API using Alpha-va ntage API.	View data, and display it into a candlestick graph.	Live data	successful

Non-Functional Test cases

Test ID(#)	Test Scenario	Execution Steps	Expected outcome	Test cases	Actual outcome
1	Blog Page	Execution using the following command: -python manage.py runserver	Working blog page	nil	Working blog page
2	Helpline Page	Execution using command: Python manage.py runserver	Working helpline page	nil	Working helpline pag

Defect Log

Requirement #	Defect ID #	Defect Description	Assigned
M1R1	1	Favourites button for signed in users	Dipali

Test Report

Category	Progress Against Plan	Status:
Functional Testing	Green	In-Progress
Non-Functional Testing	Green	In-Progress

Functional	Test Case Coverage	Status
Module ID 1	80%	in-Progress
Module ID 2	70%	In-Progress
Module ID 3	85%	In-Progress

Test Report Reference

1. https://www.pmi.org/

Thankyou