STOCK ANALYSIS (USING YAHOO FINANCE DATA)- FLASK FRAMEWORK

By Nithin.V (22PD24)

Anuvarsana.K(22PD06)

Chaithanya N(22PD09)

Title: Stock Market Data Analysis Web Application using Flask

Abstract:

This project introduces a web application created with Flask, a Python-based web framework, designed to analyse and visualize historical stock market data. By utilizing the Yahoo Finance API, users can input a specific company and date range to fetch relevant stock information. The application's key functionalities include generating various financial metrics such as volume, market capitalization, moving averages, scatter matrix, and volatility. These insights are displayed on a straightforward web interface, aiding investors, traders, and financial analysts in making well-informed decisions. Matplotlib is employed to produce interactive plots, and base64 encoding ensures smooth integration of visuals into HTML pages. With its array of analysis tools and user-friendly interface, this web

application acts as a valuable asset for navigating the intricacies of the stock market.

The application consists of the following main components:

1.Flask Routes: The application defines routes for the homepage ("/"), a page to display a list of companies ("/companylist"), and a route to handle form submission for data analysis ("/analyze").

2.DataFetching and Processing: The fetch_data function retrieves historical stock data for a specified company within a given date range using the Yahoo Finance API.

3. **Plotting Functions:** There are several functions to generate plots of different f inancial metrics such as volume, market capitalization, moving averages, scatter matrix, and volatility. These functions utilize Matplotlib for plotting.

4. **Rendering Templates**: The Flask application renders HTML templates using Jinja2 templating engine to display the analysis results.

5. **Form Submission Handling:** The /analyze route handles form submissions containing start date, end date, and company name. It then fetches the data, performs analysis, generates plots, and renders the analysis results on a dedicated page.

6**. Image Encoding**: The generated plots are encoded as base64 strings to embed them directly into HTML pages.

CODE:

analyze(html-1) for output page-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Stock Analysis Result</title>
</head>
<body>
```

```
    <h2>Analysis Results</h2>

    <h3>Volume of Stock Traded</h3>

    <img src="{{ volume_plot }}" alt="Volume of Stock Traded"><br><br>


    <h3>Market Capitalisation</h3>

    <img src="{{ market_cap_plot }}" alt="Market Capitalisation"><br><br>


    <h3>Moving Averages</h3>

    <img src="{{ moving_average_plot }}" alt="Moving Averages"><br><br>


    <h3>Scatter Matrix</h3>

    <img src="{{ scatter_matrix_plot }}" alt="Scatter Matrix"><br><br>


    <h3>Volatility</h3>

    <img src="{{ volatility_plot }}" alt="Volatility"><br><br>

</body>

</html>
```

Companylist(html-2)- redirect to company available from main webpage-

```
<!DOCTYPE html>

<html lang="en">

<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Company List</title>
</head>
<body>
    <h1>Company List</h1>
    <p>Here is a list of TOP 10 companies and their symbols:</p>

    <ul>
        <li><strong>Apple Inc.:</strong> AAPL</li>
        <li><strong>Microsoft Corporation:</strong> MSFT</li>
        <li><strong>Amazon.com, Inc.:</strong> AMZN</li>
        <li><strong>Infosys Limited:</strong> INFY</li>
        <li><strong>Tata Consultancy Services:</strong> TCS</li>
        <li><strong>Wipro Limited:</strong> WIPRO</li>
        <li><strong>HCL Technologies Limited:</strong> HCLTECH</li>
        <li><strong>ITC Limited:</strong> ITC</li>
        <li><strong>ICICI Bank Limited:</strong> ICICIBANK</li>
        <li><strong>Reliance Industries Limited:</strong> RELIANCE</li>
    </ul>
    <p>Refer to Yahoo Finance website if company symbol not here</p>
    <a href="/">Go Back</a>

</body>
</html>
```

index(html-3)- main output page_

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Stock Analysis</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        background-image: url("/static/stockmarket.jpg");
            background-size: cover;
            background-position: center;
        }
        .container {
            max-width: 600px;
            margin: 20px auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        h1 {
            text-align: center;
            color: #333;
        }
        form {
            margin-top: 20px;
```

```
        }
        label {
            display: block;
            font-weight: bold;
            margin-bottom: 5px;
        }
        input[type="text"] {
            width: calc(100% - 12px);
            padding: 8px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        input[type="submit"] {
            width: 100%;
            padding: 10px;
            border: none;
            border-radius: 4px;
            background-color: #007bff;
            color: #fff;
            cursor: pointer;
            transition: background-color 0.3s ease;
        }
        input[type="submit"]:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
```

```
  <div class="container">
    <h1>Stock Analysis</h1>
    <p>Welcome to stock analysis tool. Please fill out the form below to get started.</p>
    <p>By Nithin.V (22PD24), Anuvarsana.K (22PD06), Chaithanya N (22PD09)</p>

    <form action="/analyze" method="post">
      <label for="start_date">Start Date (YYYY-MM-DD):</label>
      <input type="text" id="start_date" name="start_date" required pattern="\d{4}-\d{2}-\d{2}">
      <label for="end_date">End Date (YYYY-MM-DD):</label>
      <input type="text" id="end_date" name="end_date" required pattern="\d{4}-\d{2}-\d{2}">
      <label for="company">Company Symbol:</label>
      <input type="text" id="company" name="company" required>
      <input type="submit" value="Submit">
    </form>
    <p>Not sure about the company symbol? <a href="/companylist" target="_blank">Find it here</a>.</p>
  </div>
</body>
</html>
```

MAIN CODE- FLASK ROUTING WITH ALL HTML FILES-

```
from flask import Flask, request, render_template, url_for
import pandas as pd
import datetime
import numpy as np
import matplotlib.pyplot as plt
```

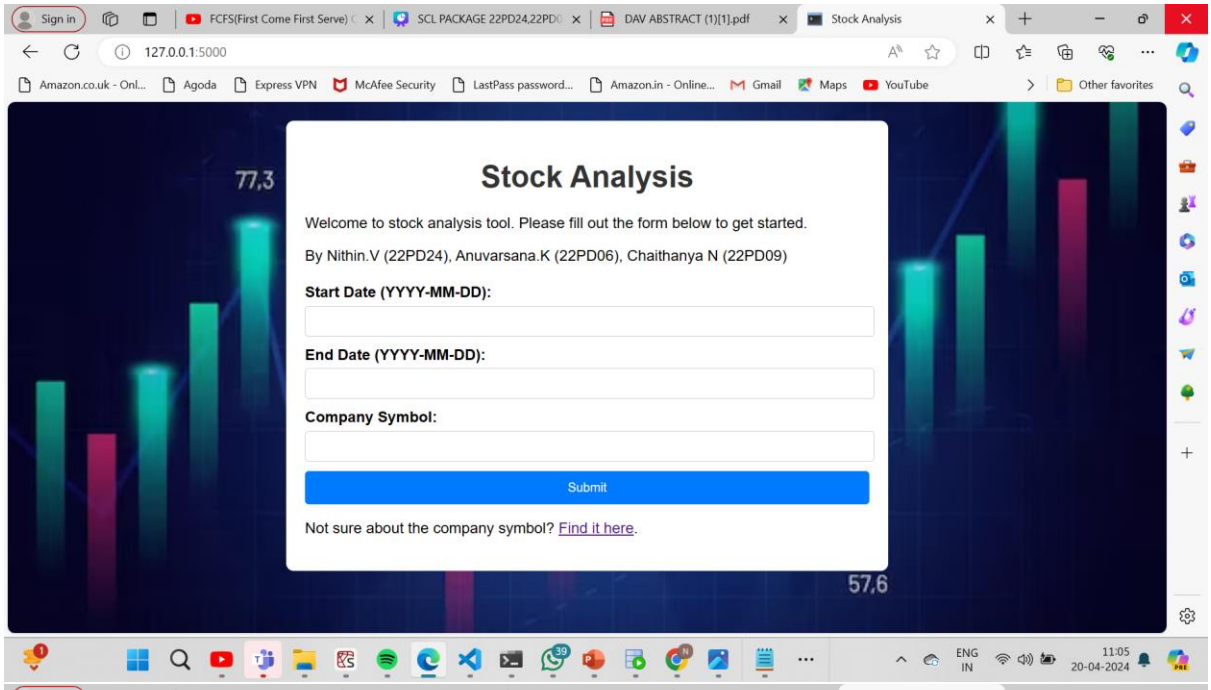
```
from pandas.plotting import scatter_matrix
import yfinance as yf
from io import BytesIO
import base64

app = Flask(__name__)

def fetch_data(company, start, end):
    return yf.download(company, start, end)

def plot_volume(traded_data):
    plt.figure(figsize=(10, 5))
    traded_data['Volume'].plot()
    plt.title('Volume of Stock Traded')
    plt.xlabel('Date')
    plt.ylabel('Volume')
    plt.grid(True)
    plt.tight_layout()
    plot_img = get_img_data()
    return plot_img
1
def plot_market_cap(data):
    data['MarketCap'] = data['Open'] * data['Volume']
    plt.figure(figsize=(10, 5))
    data['MarketCap'].plot()
    plt.title('Market Cap')
    plt.xlabel('Date')
    plt.ylabel('Market Cap')
    plt.grid(True)
```

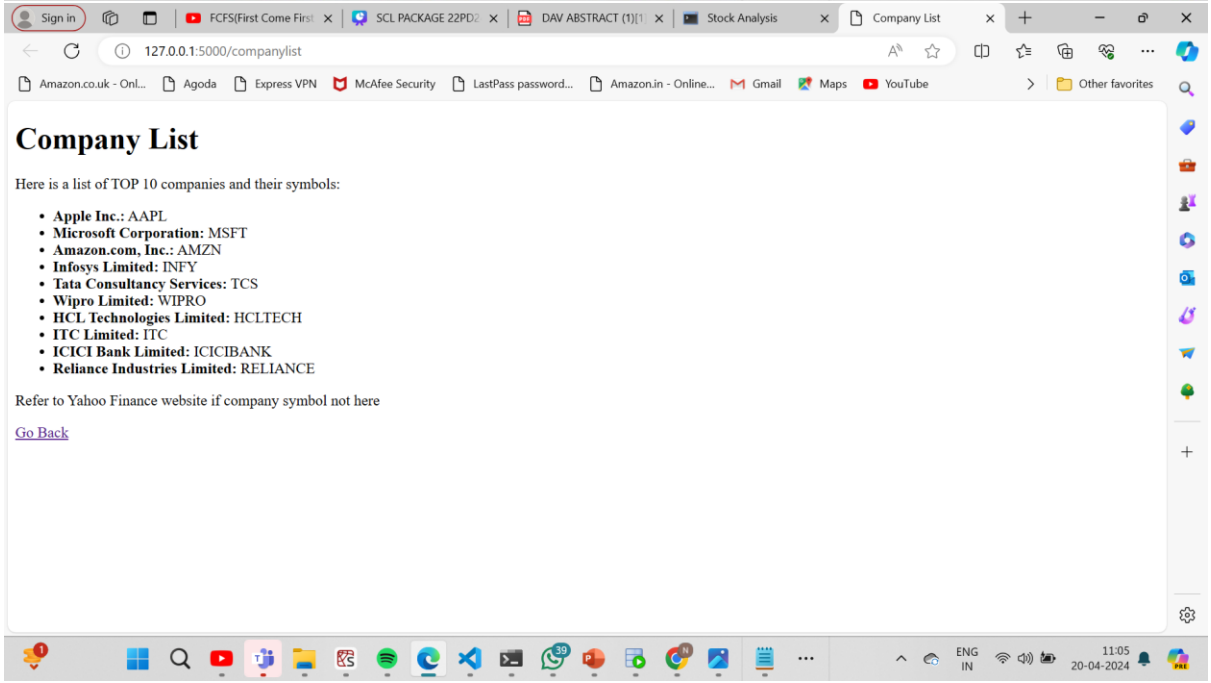
```
    plt.tight_layout()
    plot_img = get_img_data()
    return plot_img

def plot_moving_average(data):
    data['MA50'] = data['Open'].rolling(50).mean()
    data['MA200'] = data['Open'].rolling(200).mean()
    plt.figure(figsize=(10, 5))
    data['Open'].plot(label='Open Price')
    data['MA50'].plot(label='MA50')
    data['MA200'].plot(label='MA200')
    plt.title('Moving Averages')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.grid(True)
    plt.legend()
    plt.tight_layout()
    plot_img = get_img_data()
    return plot_img

def plot_scatter_matrix(data):
    plt.figure(figsize=(10, 10))
    scatter_matrix(data, alpha=0.2, figsize=(10, 10), diagonal='kde')
    plt.title('Scatter Matrix')
    plt.tight_layout()
    plot_img = get_img_data()
    return plot_img

def plot_volatility(data):
```

```
    data['returns'] = (data['Close'] / data['Close'].shift(1)) - 1
    plt.figure(figsize=(10, 5))
    data['returns'].hist(bins=100, alpha=0.5)
    plt.title('Volatility')
    plt.xlabel('Returns')
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.tight_layout()
    plot_img = get_img_data()
    return plot_img

def get_img_data():
    img = BytesIO()
    plt.savefig(img, format='png')
    img.seek(0)
    plot_img = base64.b64encode(img.getvalue()).decode()
    plt.close()  # Close the plot to avoid memory leaks
    return plot_img

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/companylist')
def companylist():
    return render_template('companylist.html')



@app.route('/analyze', methods=['POST'])
```

```
def analyze():
    start_date = request.form['start_date']
    end_date = request.form['end_date']
    company = request.form['company']

    data = fetch_data(company, start_date, end_date)

    if data is not None:
        volume_plot = plot_volume(data)
        market_cap_plot = plot_market_cap(data)
        moving_average_plot = plot_moving_average(data)
        scatter_matrix_plot = plot_scatter_matrix(data)
        volatility_plot = plot_volatility(data)

        return render_template('analyze.html',
                               volume_plot=volume_plot,
                               market_cap_plot=market_cap_plot,
                               moving_average_plot=moving_average_plot,
                               scatter_matrix_plot=scatter_matrix_plot,
                               volatility_plot=volatility_plot)
    else:
        return "Error fetching data. Please try again."

if __name__ == "__main__":
    app.run(debug=True)
```

OUTPUT WEBPAGES-





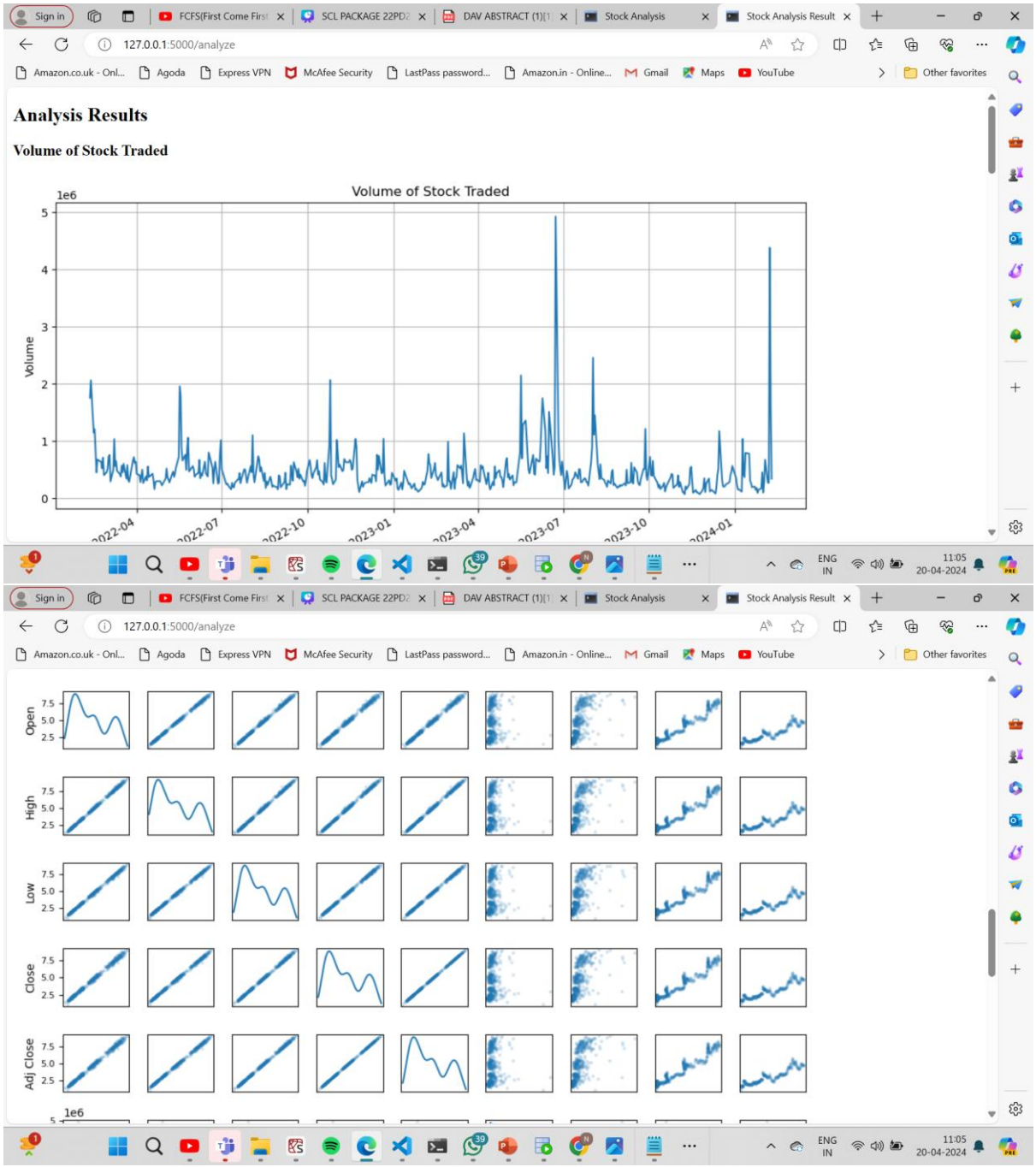
Company List

Here is a list of TOP 10 companies and their symbols:

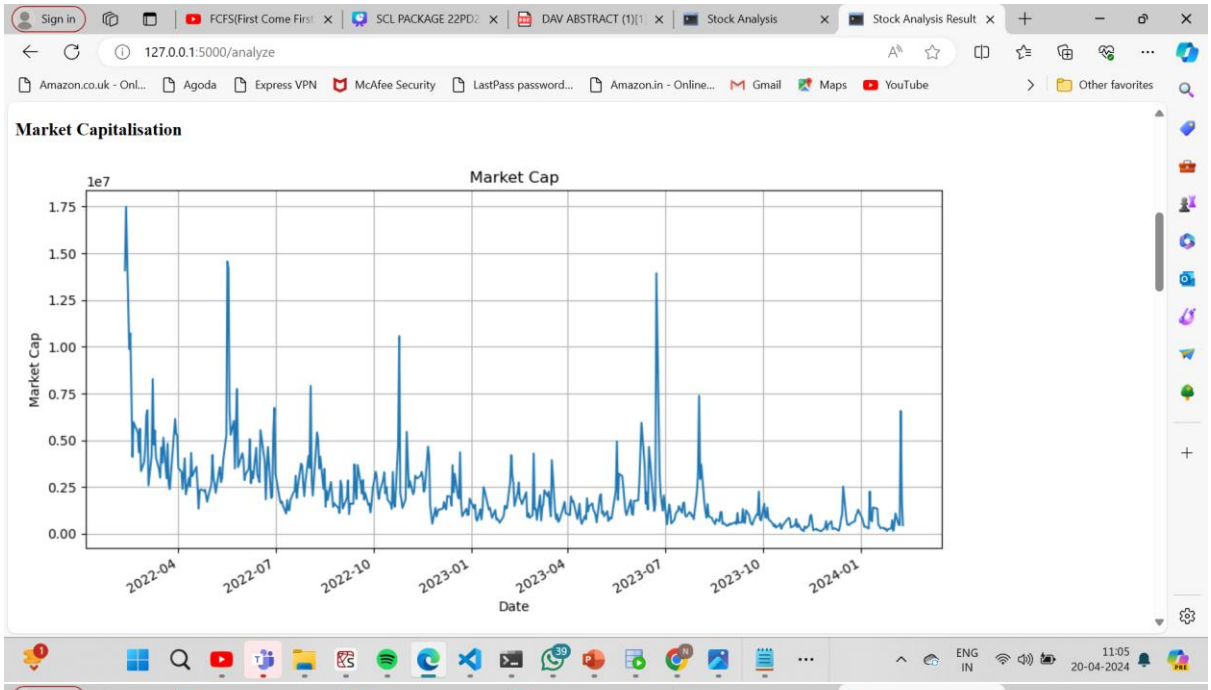
- **Apple Inc.:** AAPL
- **Microsoft Corporation:** MSFT
- **Amazon.com, Inc.:** AMZN
- **Infosys Limited:** INFY
- **Tata Consultancy Services:** TCS
- **Wipro Limited:** WIPRO
- **HCL Technologies Limited:** HCLTECH
- **ITC Limited:** ITC
- **ICICI Bank Limited:** ICICIBANK
- **Reliance Industries Limited:** RELIANCE

Refer to Yahoo Finance website if company symbol not here

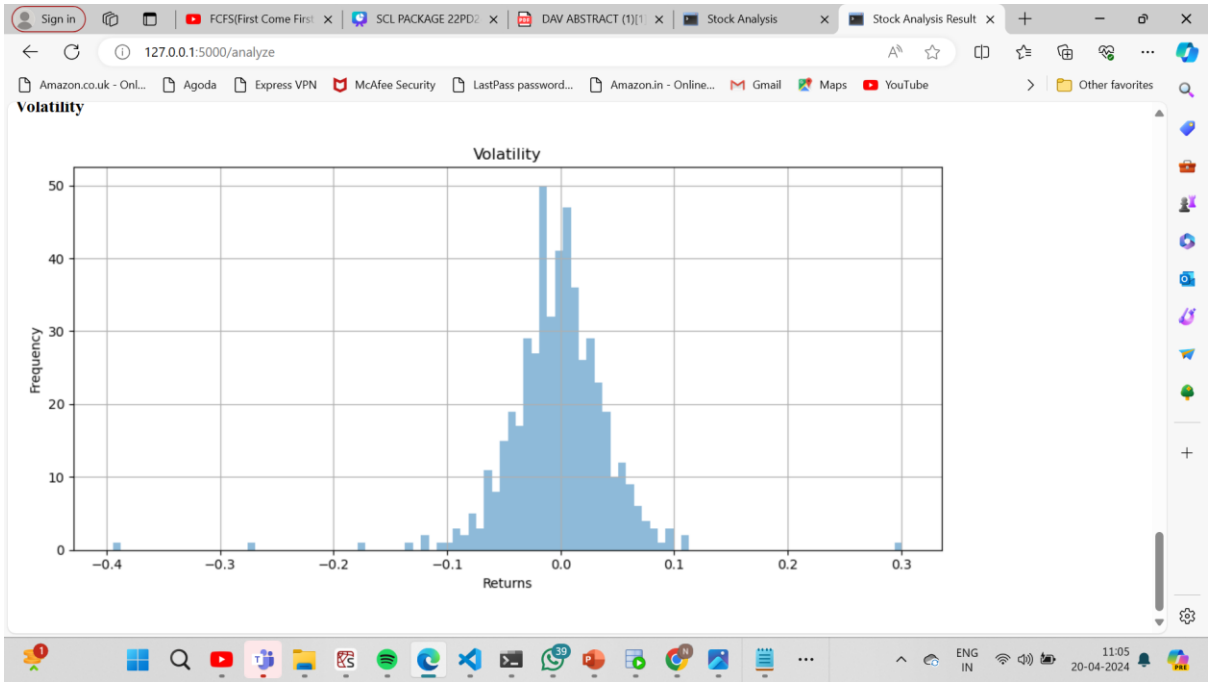
Go Back


Sign in
FCFS(First Come First
SCL PACKAGE 22PD2
DAV ABSTRACT (1)[1
Stock Analysis
Stock Analysis Result
127.0.0.1:5000/analyze
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Amazon.in - Online...
Gmail
Maps
YouTube
Other favorites
Analysis Results
Volume of Stock Traded
Volume of Stock Traded
1e6
Volume
2022-04
2022-07
2022-10
2023-01
2023-04
2023-07
2023-10
2024-01
ENG
IN
11:05
20-04-2024

Sign in
FCFS(First Come First
SCL PACKAGE 22PD2
DAV ABSTRACT (1)[1
Stock Analysis
Stock Analysis Result
127.0.0.1:5000/analyze
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Amazon.in - Online...
Gmail
Maps
YouTube
Other favorites
Open
High
Low
Close
Adj Close
7.5
5.0
2.5
1e6
ENG
IN
11:05
20-04-2024


Sign in
FCFS(First Come First
SCL PACKAGE 22PD2
DAV ABSTRACT (1)[1
Stock Analysis
Stock Analysis Result
127.0.0.1:5000/analyze
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Amazon.in - Online...
Gmail
Maps
YouTube
Other favorites
Market Capitalisation
Market Cap
1e7
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Market Cap
2022-04
2022-07
2022-10
2023-01
2023-04
2023-07
2023-10
2024-01
Date
ENG
IN
11:05
20-04-2024


Sign in
FCFS(First Come First
SCL PACKAGE 22PD2
DAV ABSTRACT (1)[1
Stock Analysis
Stock Analysis Result
127.0.0.1:5000/analyze
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Amazon.in - Online...
Gmail
Maps
YouTube
Other favorites
Moving Averages
Moving Averages
Open Price
MA50
MA200
9
8
7
6
5
4
3
2
1
Price
2022-04
2022-07
2022-10
2023-01
2023-04
2023-07
2023-10
2024-01
Date
ENG
IN
11:05
20-04-2024


Volatility
Volatility
Frequency
Returns
0
10
20
30
40
50
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3