Stock Portfolio Management and Analysis System

Project Overview:

This project aims to manage and analyze a portfolio of stocks using the "Stock Market Dataset" from Kaggle. It involves downloading, processing and calculating stock-data. Visualizing the results through a web interface. The project will include both backend and frontend tasks.

Back-End

Setup and Data Collection:

- Download the dataset from Kaggle.(https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset?resource=download)
- Load the ***symbols_valid_meta.csv*** and ***a.csv*** file into your Python environment using `pandas`.

Data Preprocessing:

- Clean and preprocess the data to remove any missing values or irrelevant columns.
- Filter the dataset to include only the relevant columns for analysis.

Select relevant columns from *symbols_valid_meta.csv*

```
data = data[['Symbol', 'Security Name']]
```

Select relevant columns from A.csv

```
data = data[['Date', 'Open', 'High']]
```

Portfolio Performance Analysis:

- Calculate portfolio metrics such as the number of unique stocks, distribution across different exchanges, and categories.

Number of unique stocks

```
num_unique_stocks = data['Symbol'].nunique()
```

Distribution across exchanges

```
exchange_distribution = data['Listing Exchange'].value_counts()
```

Distribution across market categories

```
category_distribution = data['Market Category'].value_counts()
```

- Calculate average daily return, volatility, Sharpe ratio, and Value at Risk (VaR).

Calculate average daily return and volatility

```
average_daily_return = data['Daily_Return'].mean()
```

```
                volatility = data['Daily_Return'].std()
```

Data Analysis:

- Calculate necessary metrics for plotting using Matplotlib.

  Number of unique stocks

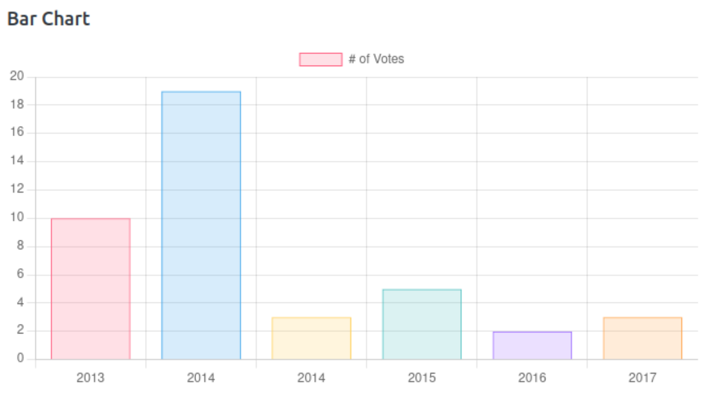
  ```
      num_unique_stocks = data['Symbol'].nunique()
  ```

  Distribution across exchanges

```
exchange_distribution = data['Listing Exchange'].value_counts()
```

  Plotting with Matplotlib

  ```
      plt.figure(figsize=(10, 6))
      plt.bar(exchange_distribution.index,color='skyblue')
      plt.xlabel('Listing Exchange')
      plt.ylabel('Number of Stocks')
      plt.title('Distribution of Stocks Across Exchanges')
      plt.show()
  ```



API Development:

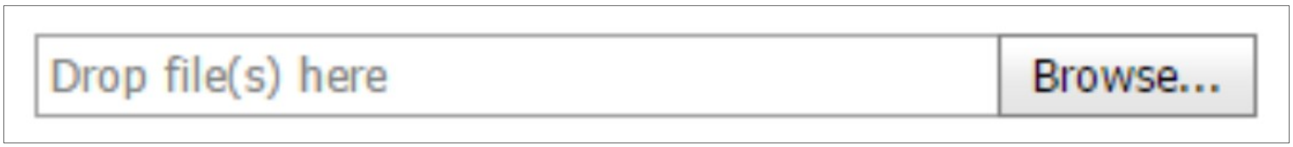
- Create a Flask API to show the portfolio performance metrics and stock metadata to dashboard.

  ```
  @app.route('/stocks')
  def get_stocks():
      stocks = data.to_dict(orient='records')
      return jsonify(stocks)
  ```

Front-End

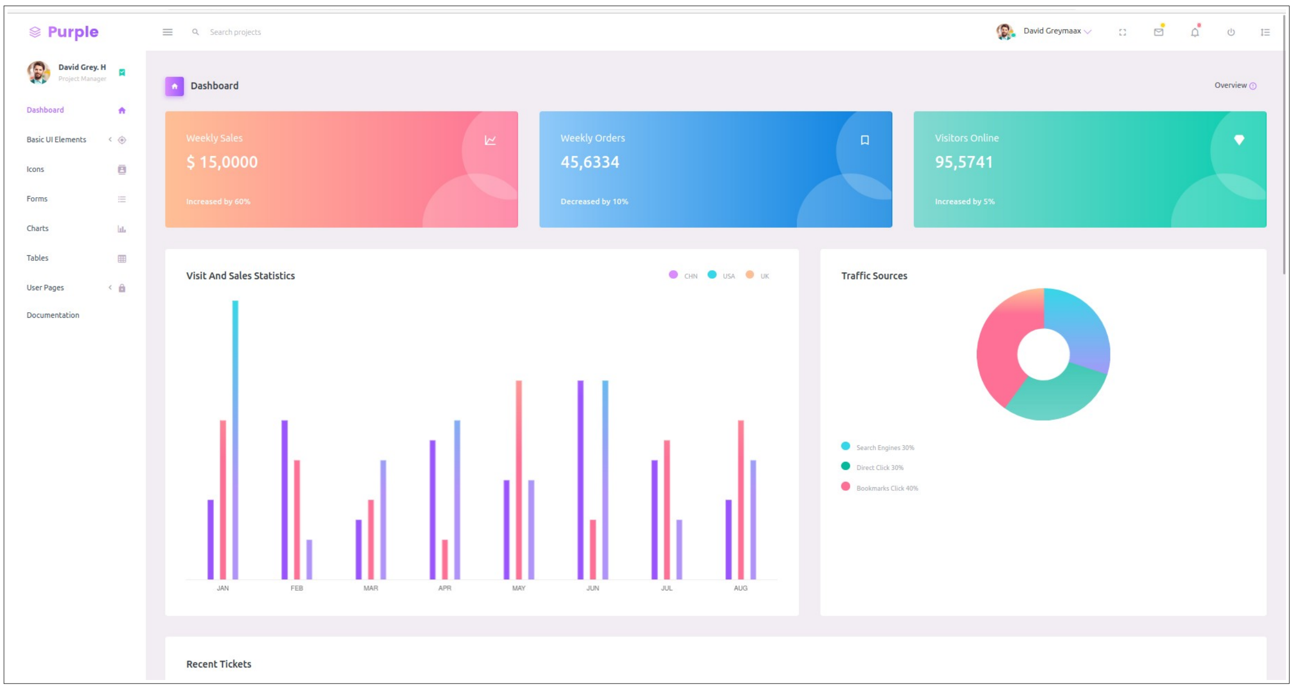
- ***Setup HTML and CSS:***
  - Create an HTML template to display the portfolio performance metrics and stock metadata.
  - Use CSS for styling the frontend.

Upload file to get the stock data.



- **Fetching Data from the API:**
  - Use JavaScript to fetch the portfolio performance metrics and stock metadata from the Flask API.
  -

- **Styling with CSS:**
  - Add styles to enhance the look and feel of the dashboard.



- Detail info of each share using dataframes and list.

Striped Table

Add class .table-striped

User	First name	Progress	Amount	Deadline
	Herman Beck		$ 77.99	May 15, 2015
	Messsy Adam		$245.30	July 1, 2015
	John Richards		$138.00	Apr 12, 2015
	Peter Meggik		$ 77.99	May 15, 2015
	Edward		$ 160.25	May 03, 2015
	John Doe		$ 123.21	April 05, 2015
	Henry Tom		$ 150.00	June 16, 2015