Project Title	**Data-Driven Stock Analysis: Organizing, Cleaning, and Visualizing Market Trends**
Skills take away From This Project	**Pandas, Python, Power BI, Streamlit, SQL,Statistics ,Data Organizing, Cleaning, and Visualizing.**
Domain	**Finance / Data Analytics**

Problem Statement:

The **Stock Performance Dashboard** aims to provide a comprehensive visualization and analysis of the Nifty 50 stocks' performance over the past year. The project will analyze daily stock data, including open, close, high, low, and volume values. Clean and process the data, generate key performance insights, and visualize the top-performing stocks in terms of price changes, as well as average stock metrics. The solution will offer interactive dashboards using **Streamlit** and **Power BI** to help Investors, analysts, and enthusiasts make informed decisions based on the stock performance trends.

Business Use Cases:

1. **Stock Performance Ranking**: Identify the top 10 best-performing stocks (green stocks) and the top 10 worst-performing stocks (red stocks) over the past year.
2. **Market Overview**: Provide an overall market summary with average stock performance and insights into the percentage of green vs red stocks.
3. **Investment Insights**: Help investors quickly identify which stocks showed consistent growth and which ones had significant declines.
4. **Decision Support**: Provide insights on average prices, volatility, and overall stock behavior, useful for both retail and institutional traders.

Approach:

Data Extraction and Transformation:

- **Data is provided in YAML format, organized by months.**
- **Within each month's folder, there are date-wise data entries.**
- **The task is to extract this data from the YAML file and transform it into a CSV format, organized by symbols**
- **This will result in 50 CSV files after the extraction process, one for each symbol or data category**

Data Analysis and Visualization Requirements:

- **Python DataFrame for Key Metrics**:
  - **Top 10 Green Stocks**: Sort the stocks based on their yearly return and select the top 10.
  - **Top 10 Loss Stocks**: Sort the stocks based on their yearly return and select the bottom 10.
  - **Market Summary**:
    - Calculate the overall number of green vs. red stocks.
    - Calculate the average price across all stocks.
    - Calculate the average Volume across all stocks.

1. Volatility Analysis:

- **Objective**: Visualize the volatility of each stock over the past year by calculating the standard deviation of daily returns.
- **Reason**: Volatility gives insight into how much the price fluctuates, which is valuable for risk assessment. Higher volatility often indicates more risk, while lower volatility indicates a more stable stock.

Metrics:

- Calculate **daily returns** for each stock: `(Close Price - Previous Close Price) / Previous Close Price`.
- Compute the **standard deviation** of daily returns for each stock to measure volatility.
- Plot a **bar chart** showing the volatility of the top 10 most volatile stocks over the year.

Visualization:

- **Top 10 Most Volatile Stocks**: A bar chart with the stock ticker on the x-axis and volatility (standard deviation) on the y-axis.

2. Cumulative Return Over Time:

- **Objective**: Show the cumulative return of each stock from the beginning of the year to the end.
- **Reason**: The cumulative return is an important metric to visualize overall performance and growth over time. This helps users compare how different stocks performed relative to each other.

Metrics:

- For each stock, calculate the **cumulative return** by applying a running total of daily returns.
- Plot a **line chart** for the top 5 performing stocks (based on cumulative return) over the course of the year.

Visualization:

- **Cumulative Return for Top 5 Performing Stocks**: A line chart displaying cumulative returns for each stock over the year (increasing trend indicates positive performance).

3. Sector-wise Performance:

- **Objective**: Provide a breakdown of stock performance by sector (sector data shared as csv).
- **Reason**: Investors and analysts often look at sector performance to gauge market sentiment in specific industries (e.g., IT, Financials, Energy, etc.).

Metrics:

- Classify each stock by its **sector** (this can be done by adding a separate dataset or manually mapping sectors to stocks).
- Calculate the **average yearly return** for each sector.
- Plot a **bar chart** showing the average performance for each sector.

Visualization:

- **Average Yearly Return by Sector**: A bar chart where each bar represents a sector and its height indicates the average yearly return for stocks within that sector.

4. Stock Price Correlation:

- **Objective**: Visualize the correlation between the stock prices of different companies.

- **Reason**: This analysis is valuable to understand if certain stocks tend to move in tandem (e.g., correlated with market trends or sector performance).

Metrics:

- Calculate the **correlation coefficient** between the closing percentage of different stocks. For this, use the `pandas.DataFrame.corr()` method.
- Create a **correlation matrix** to identify how stocks are related to each other.
- Plot a **heatmap** of the correlation matrix to visualize these relationships.

Visualization:

- **Stock Price Correlation Heatmap**: A heatmap to show the correlation between the closing prices of various stocks. Darker colors represent higher correlations.

5. Top 5 Gainers and Losers (Month-wise):

- **Objective**: Provide monthly breakdowns of the top-performing and worst-performing stocks.
- **Reason**: This analysis will allow users to observe more granular trends and understand which stocks are gaining or losing momentum on a monthly basis.

Metrics:

- Group the stock data by **month** and calculate the **monthly return** for each stock.
- For each month, identify the top 5 **gainers** and top 5 **losers** based on percentage change.
- Create a **dashboard-style visualization** with 5 charts showing top gainers and losers for each month (12 months total).

Visualization:

- **Top 5 Gainers and Losers by Month**: Create a set of 12 bar charts for each month showing the top 5 gainers and losers based on percentage return.

Dataset:[Dataset]

Results:

- A fully functional dashboard showing the top-performing and worst-performing stocks over the last year.
- Insights on the overall market with clear indicators of stock performance trends.
- Interactive visualizations using **Power BI** and **Streamlit** to make the data easily accessible for users.

Technical Tags:

- **Languages**: Python
- **Database**: MySQL/PostgreSQL
- **Visualization Tools**: Streamlit, Power BI
- **Libraries**: Pandas, Matplotlib, SQLAlchemy

Project Deliverables:

1. **SQL Database**: Contains clean and processed data.
2. **Python Scripts**: For data cleaning, analysis, and database interaction.
3. **Power BI Dashboard**: Visualizations for stock performance.
4. **Streamlit Application**: Interactive dashboard for real-time analysis.

Project Guidelines:

- **Follow coding standards**: Consistent naming conventions, modular code.
- **Data validation**: Ensure all data is accurate and complete.
- **Optimized queries**: Efficient SQL queries for large datasets.
- **Documentation**: Well-documented code and a detailed project report.

Streamlit Doc	https://docs.streamlit.io/library/api-reference
Streamlit recording (Tamil)	**Special Session for STREAMLIT Tamil**
Project Live Evaluation	Project Live Evaluation
GitHub Reference	How to Use GitHub.pptx

Project Orientation (Tamil)	Project Orientation
Power BI doc	Power bi doc
Power BI Session	Power BI Session
Project Orientation (English)	Project Orientation English

Project Evaluation metrics:

- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system)
- You have to maintain your code on **GitHub**.(Mandatory)
- You have to keep your **GitHub** repo public so that anyone can check your code.(Mandatory)
- Proper readme file you have to maintain for any project development(Mandatory)
- You should include basic workflow and execution of the entire project in the readme file on **GitHub**

- Follow the coding standards: https://www.python.org/dev/peps/pep-0008/
- You need to Create a Demo video of your working model and post in **LinkedIn**(Mandatory)

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday to Saturday (4:00PM to 5:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.
Note: This form will Open on Saturday and Sunday Only on Every Week

Timing: Monday-Saturday (5:30PM to 7:00PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA