PROJECT PLAN

TECHNOLOGY INDUSTRY STOCK MARKETS OVER THE 10 YEARS

1. Project Overview

Objective:

The objective of this project is to develop an **interactive stock analysis and investment tracking tool** that allows users to:

Analyze Historical Stock Data – Retrieve and visualize stock performance over a specified period.

Assess Investment Growth – Calculate the potential returns of an investment made over the past 10 years based on historical stock prices.

Enable User-Driven Interaction – Provide an intuitive web-based interface where users can input stock tickers, investment years, and amounts to receive insights on potential gains.

Store and Retrieve Data Efficiently – Use **PostgreSQL** to manage stock data and ensure efficient access to historical trends.

Enhance Visualization – Display stock trends, price movements, and investment growth using **interactive charts** powered by JavaScript (Plotly.js) and Python (Matplotlib).

Build a Scalable Flask API – Serve real-time stock analysis data and investment calculations via RESTful API endpoints.

This project aims to help users **make informed investment decisions** by understanding stock performance over time and visualizing potential returns based on past investments.

How has a stock performed over a specific time period?

- What trends can be observed in stock prices (open, close, high, low)?
- Are there seasonal patterns in stock performance?

Research questions How has a stock performed over a specific time period?

- What trends can be observed in stock prices (open, close, high, low)?
- Are there seasonal patterns in stock performance?

Which stocks have provided the highest returns over time? How much would an investment in a stock have grown over a given period? How do different stocks compare in performance over the same period? How much would an investment in a stock have grown over a given period?

Dataset: https://www.nasdaq.com/ , https://finance.yahoo.com/sectors/technology/

Audience: Small/medium investors

PHASE	TASKS	DEADLINE
Planning	Define scope, objectives, and research idea.	2025-02-19
Identify Stakeholders	Small/medium investors, project team members.	2025-02-19
Data Collection	Fetch historical stock data using yfinance from NASDAQ and Yahoo Finance.	2025-02-20
Data Visualization	Develop interactive charts using Plotly.js and Matplotlib. Create visual elements like line charts, candlestick charts, and investment growth bar charts.	2025-02-22
API Development	Implement a Flask API to serve stock data and calculate potential investment returns.	2025-02-23
Frontend Development	Build a web-based interface using HTML, CSS, and JavaScript.	2025-02-26
Testing & Validation	Perform unit testing on individual components. Conduct integration testing for end-to-end functionality. Gather feedback through user testing.	2025-02-28
Deployment	Host the frontend on GitHub Pages.	2025-03-01
Reporting	Create a presentation to showcase the tool's features and potential investor insights.	2025-03-03

2. Methodology

Tools:

- Backend: Python, Flask, PostgreSQL
- Frontend: HTML, CSS, JavaScript (Plotly.js)
- Data Handling: Pandas, SQL Alchemy
- Visualization: Matplotlib, Plotly.js
- Data Source: Yahoo Finance, NASDAQ

3.Steps:

1. Project Initialization

🎯 Objective:

- Define the project scope, goals, and technical requirements.

- Define project goals: Analyze historical stock data, assess investment growth, provide user interaction.
- Identify stakeholders: Small/medium investors, project team members.
- Outline deliverables: Interactive web app, RESTful API, interactive charts, PostgreSQL integration.
- Set up project management tools: (e.g., Trello, Asana, or GitHub Projects) to track progress.

2. Environment Setup

- 💻 Tools & Technologies:
- Backend: Python, Flask, PostgreSQL
- Frontend: HTML, CSS, JavaScript (Plotly.js)
- Data Handling: Pandas, SQLAlchemy
- Visualization: Matplotlib, Plotly.js
- Data Source: Yahoo Finance, NASDAQ

3. Database Design & Setup

🗃️ Database: PostgreSQL

- Schema Design: Create a table to store historical stock data.
- Set up a local PostgreSQL database.
- Write a Python script (database_setup.py) to automate schema creation.
- Test database connection using SQLAlchemy.

4. Data Collection & Storage

📈 Data Source: Yahoo Finance, NASDAQ

- Use yfinance library to fetch historical stock data.
- Write a script (fetch_data.py) to:
- Download stock data.
- Clean and transform the data.
- Store data into the PostgreSQL database.

5. Backend Development

🚀 Flask API Development

- Build a RESTful API with Flask to:
- Serve stock data (/stocks/<symbol> endpoint).

- Calculate investment growth (/investment endpoint).
- Allow user-driven interaction.

6. Frontend Development

🎨 Building the User Interface

- Create an index.html page with:
- Dropdown menu to select stock symbols.
- Input fields for investment amount and start year.
- Interactive charts to display stock trends and investment growth.
- Use Plotly.js for dynamic visualizations.

7. Data Visualization & Analytics

📊 Visualization with Plotly & Matplotlib

Create:

- Line Chart: Stock price trends over time.
- Candlestick Chart: Daily price movements.
- Bar Chart: Investment growth over the years.
- Add interactive elements:
- Select different stocks.
- Change timeframes.
- Toggle investment scenarios.

8. Testing & Validation

🧪 Testing Phases:

- Unit Testing: Test individual functions (data fetching, investment calculations).
- Integration Testing: Ensure frontend and backend work together.
- User Testing: Gather feedback from potential users.

9. Deployment

🌐 Deploying the Application:

- Frontend: Host HTML & static files on GitHub Pages
- Backend: Deploy Flask API
- Database: Use PostgreSQL on a cloud service

🚦 Deployment Steps:

Push code to GitHub.

Connect repository to deployment platform.

Test the live application and resolve deployment issues.

10. Documentation & Finalization

📚 Prepare Documentation:

- README.md: Explain how to set up, run, and use the application.
- Technical Documentation: API endpoints, database schema, project structure.
- User Guide: How to use the stock analysis tool effectively.

🎯 Final Tasks:

Create a presentation/demo video.

Share the GitHub repository link.

4. Deliverables

- Dataset Preparation Report: Document on data collection, cleaning, and transformation processes for historical stock data.
- Analysis Outputs: Interactive charts, investment growth visualizations, and performance comparison reports.
- API Documentation: Clear instructions on how to use the Flask API endpoints for retrieving stock data and calculating investment returns.
- Final Report: Comprehensive analysis of stock performance, investment insights, and potential returns over selected periods.
- Presentation Deck: Visual summary of key findings and a demo of the interactive tool for stakeholders.

5. Timeline

Duration: 14 days
Start Date: 19.02.2025
End Date: 03.03.2025

6. Stakeholder Communication Plan

Audience	**Meeting Time**	**Communication**
Project Team	3 classes	Slack/Meetings
Students/Research	3 classes + 2 off-class meetings	Report/Presentation
Development Initiatives	3 off-class meetings	Report/Presentation