

Project04: Makers Makin' It, Act II -- The Seequel

TNPG: PretendingGreatness

Roster: Michelle Zhu (PM), Victor Casado, Mark Ma, Jessica Yu

TARGET SHIP DATE: 2025-04-10

P04 DESIGN DOCUMENT



Description

This project implements a web application that incorporates data visualization to identify interesting patterns and correlations between Elon Musk's tweets and Tesla stock market prices.

- Users can register to use the site
- A logged-in user will be able to
 - View their name at the top of every web page
 - Access all functionalities
- A non-logged-in user will be able to:
 - Access all functionalities

Datasets

A. Elon Musk's Tweets from 2010 to 2025 (Kaggle)

<https://www.kaggle.com/dadalyndell/elon-musk-tweets-2010-to-2025-march>

B. Tesla Stocks Dataset (Kaggle)

<https://www.kaggle.com/datasets/iamtanmayshukla/tesla-stocks-dataset>

API

A. Google Gemini API

We are using the Google Gemini API to assist in providing tweets analysis.

Functionalities

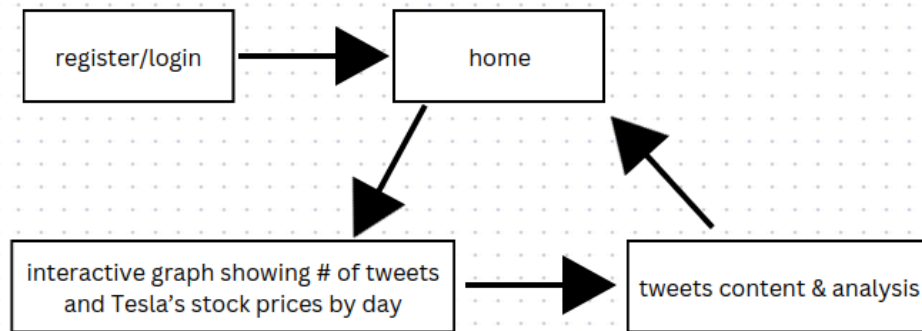
- ☐ Scatterplots of tweets (number of tweets made/number of likes or comments) and \$TSLA stock prices by day

- ☐ Clicking on a specific data point redirects user to another page showing the tweet content
- ☐ A heatmap of tweets
- ☐ AI tweet analysis
 - An AI (ChatGPT) will predict whether it is likely that Tesla stocks will go up/down given the tweet. We will compare this to the actual stock direction.
- ☐ Other AI features?

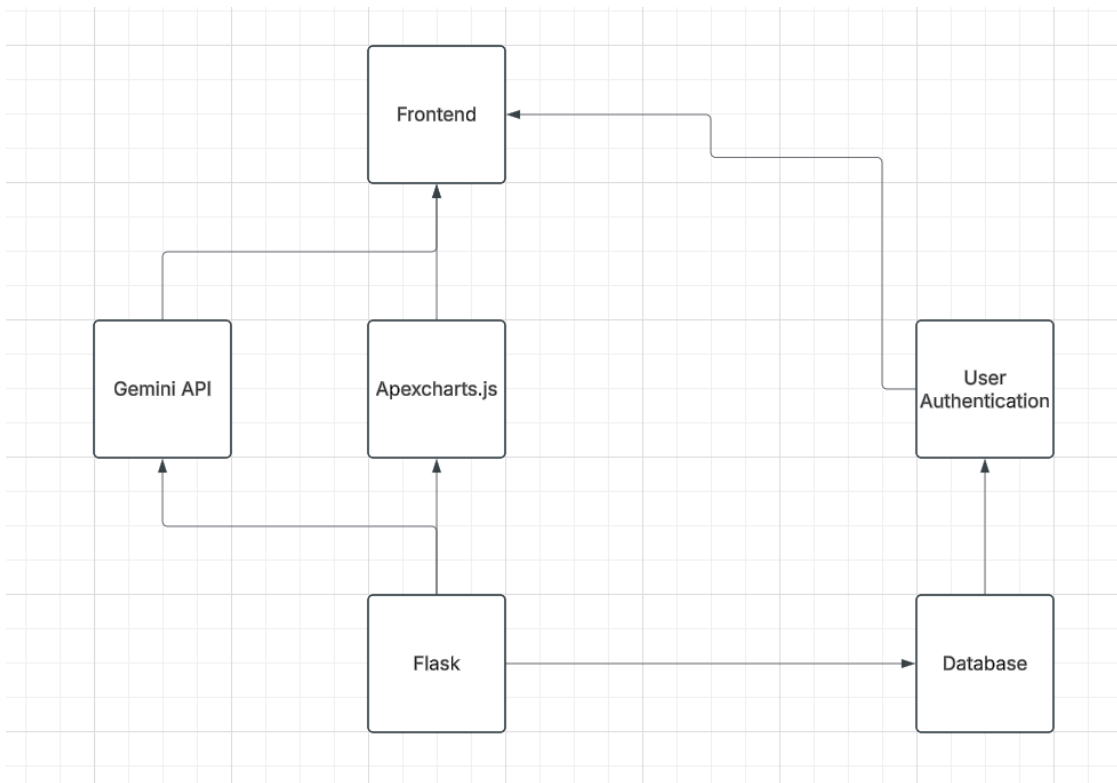
Program Components

1. Flask/Python:
 - a. Serves the websites, handles HTTP requests, authentication (login), controls logic surrounding permissions
 - b. Interacts with tables within the database file to fetch and store user data
 - c. Coordinates Flask routes (between HTML pages) and middleware functions (ex. module for storing data into tables) to control application flow
2. SQLite3:
 - a. Stores user information (usernames, passwords, permissions, etc)
 - b. Data used by middleware functions and HTML templates
3. HTML:
 - a. Maintains frontend structure
 - b. Renders content passed by Flask routes/database
 - c. Displays form for user interactions and works with Python to create the various actions
4. CSS (Bootstrap):
 - a. Adds styling
5. ApexCharts.js:
 - a. Presents visually appealing data visualizations
 - b. Adds interactivity for users

Site Map



Component Map



Database Organization

1. Register/login Information
 - a. Username
 - b. Password

SQL Database:

Username	Password
Topher	Mykloyk
abc	def

Task Breakdown

1. Michelle Zhu: Full-stack/Project Lead
 - a. Implement user authentication system and sessions
 - b. Work with backend and frontend to make sure everything works smoothly
 - c. Pick up any work that isn't explicitly delegated/falling behind schedule
 - d. Assist with roadblocks in any area
2. Victor Casado: Backend
 - a. Create SQLite3 database schema
 - b. Implement charts with apexcharts.js
 - c. Implement AI to integrate sentiment analysis with dataset analysis (i.e. use AI to predict the trend of the Tesla Stock given Elon Musk's tweets)
 - d. Create and organize python modules to handle logic
3. Mark Ma: APIs
 - a. Incorporate Gemini API
 - b. Implement graphing with apexcharts.js and dataset relationship analysis
 - c. Integrate Flask with JS
4. Jessica Yu: Frontend
 - a. Create HTML pages with basic structure
 - b. Implement charts with apexcharts.js
 - c. Use Bootstrap to make things look nice
 - d. Collaborate with the backend to ensure everything works with the frontend