COP 290 Subtask 2

Abhinav Shripad (2022CS11596)
Jahnabi Roy (2022CS11094)

January 2024

1 Introduction

In this report, we present the design and implementation details of the user interface for the stock analysis platform developed in Subtask 2. The goal was to create a clean, efficient, and user-friendly interface that incorporates features such as plotting stock graphs, visual comparison of multiple stocks, and applying technical filters.

2 Design Decisions

2.1 Choice of Framework

We chose Flask as the web framework for our project. While the starter code provided a foundational structure, we implemented numerous modifications and enhancements to tailor it precisely to our needs. Our development stack includes **JavaScript**, **HTML**, **CSS** for front-end design, **SQL** for database management, and both `.json` and `.csv` formats for effective data handling.

2.2 Modular Code Structure

To ensure clean and modular code, we organized our codebase into separate modules for different functionalities. These includes :-

- SQL for managing user_name, `password` and `balance`. This is because the number of users can be very high.
- json for managing each users individual profile, as each user can buy at max 50 different stocks. We provide options only for Nifty 50 Stocks
- Separate python functions at backend to pre-process data

3 Features Implementation

3.1 Account Management

Upon creating an account, each user is credited with a starting balance of 10 lakh. Users can buy and sell stocks based on current prices, and their balance updates in real time. The platform also provides a summary of owned stocks, indicating the percentage change in them with respect to the previous day's closing.

3.2 Individual Stock

This section provides comprehensive details for individual stocks, enhancing the user's understanding and decision-making process:

- **Real-time Price and P/E Ratio**
- **Buy or Sell Options**
- **Percentage Increase and Change from Last Closing Price**
- **Historical Data Graph**
- **Company Information**

3.3 Comparing Multiple Stocks

Users can visually compare the prices of multiple stocks on the same graph. We designed an intuitive interface where users can select multiple stocks, and the system dynamically generates a comparative graph.

3.4 Applying Technical Filters

To assist users in filtering stocks based on technical indicators like P/E ratio, or average closing price, volume, value, LTP and number of trades over a time period. Users can input their desired filter criteria, and the system generates a list of stocks that meet the specified conditions.

4 Libraries Used

The libraries used can be found in our submitted `requirements.txt` file. Few significant libraries are

- **beautifulsoup4 and Selenium:** For web scraping
- **requests:** For sql database management and accesing few websites.
- **yfinance:** Since jugaad-data was having conflicting version requirements with click, we used yfinance for fetching historical data.