Assignment: Backend Development Challenge

Problem Statement:
Create a backend application using **Node.js**, **Rust (or any other)** that simulates a basic trading bot for a hypothetical stock market. The purpose of this bot is to execute trades based on predefined rules and conditions, while tracking its profit/loss and performance metrics.

Requirements:

1. **Backend Application**:
   - Build a backend service that continuously monitors stock price changes using mock data.
   - Define a basic trading strategy that decides when to "buy" or "sell" based on market movements.
2. **Data Handling**:
   - Use a mock API endpoint to get real-time stock prices.
   - Parse and handle API responses effectively.
3. **Trading Logic**:
   - Implement simple trading strategies (e.g., moving average crossover, momentum trading) based on the stock prices.
   - For example, buy when the stock price drops by 2% and sell when it rises by 3%.
4. **Profit/Loss Tracking**:
   - Track the bot's positions, balance, and overall profit/loss.
   - Provide a summary report showing the trades made and the final profit/loss statement.
5. **Documentation**:
   - Include brief documentation explaining the trading logic, API usage, and how to run the application.

Submission:

- Upload the project to a GitHub repository and share the link.
- Ensure that the code is well-structured, commented, and follows best practices.
- Use environment variables for sensitive configurations.
- Ideal time to complete all the features should not exceed 4 hours. If you feel that the complete feature list is time taking, feel free to select your own set of features with proper explanation.

Evaluation Criteria:

1. **Code Quality**:
   - Clean, maintainable, and well-documented code.
   - Proper use of version control and meaningful commit messages.
2. **Algorithmic Implementation**:
   - Correctness of trading logic.
   - Efficient handling of API calls and data processing.
3. **Backend Skills**:
   - Effective use of language features.
   - Error handling.
4. **Creativity**:
   - Use of additional features like logging, analytics, or alternative strategies.