StockSync: Python Final Project Documentation

Group 1

Team Members
Aman Joshi
Parin Sudhirkumar Mandavia
Ayesha Savaliya
Dhruvi Raval
Kathan Suthar

Project Overview

StockSync is a comprehensive stock trading simulation platform developed as a Python final project. It aims to provide an immersive experience in stock market trading through a web-based application that incorporates real-time data for educational purposes.

Objectives
- To simulate stock trading activities with real-time market data.
- To educate users on effective trading strategies and market trends.
- To develop a user-friendly platform with robust backend support for trading simulations.

Development Steps

Review of TTD and ERD
- The project development began with a thorough review of the Test-Driven Development (TTD) approach and the Entity-Relationship Diagram (ERD) to ensure a solid foundation and clear understanding of the project requirements.

Database Selection and Integration
- **MySQL**: MySQL was chosen for its simplicity and ease of integration with Flask, the selected web framework.
- **Operations**: CRUD operations were implemented to interact with the database effectively, supporting functionalities like user registration, stock trading, and portfolio management.

Framework Selection
- Flask was selected for its lightweight nature and flexibility, facilitating rapid development and ease of deployment.

Task Breakdown and Collaboration
- Tasks were distributed among team members based on their expertise, with a focus on collaborative coding and peer reviews to maintain high-quality code standards.
- Version control was managed through GitHub, with a branching strategy that encouraged feature-based development and continuous integration.

Code Quality and Organization

Code Structure
- Adherence to Pythonic best practices ensured readable and maintainable code.
- The project's modular design facilitated separation of concerns and reusability across components.

Documentation
- Comprehensive code documentation and clear comments were maintained throughout the development process to describe the functionality of different sections.

Modularity
- The application's architecture promoted modularity, with distinct layers for the database model, business logic, and presentation.

Testing and Quality Assurance

Unit Tests
- Extensive unit tests covered a significant portion of the functionalities, ensuring the application's robustness and reliability.
- **Test Framework**: pytest was utilized for its simplicity and powerful testing capabilities.

Error Handling
- The application implements graceful error handling, offering clear feedback to users and ensuring a smooth user experience.

Database Integration

Data Manipulation
- StockSync performs sophisticated data manipulation, including real-time updates of stock prices and user portfolios.

Data Integrity
- Comprehensive data validation and integrity checks were implemented to prevent erroneous data entries and maintain consistency.

Deployment and Documentation

Deployment Platform
- The application was deployed on PythonAnywhere, a choice that offered simplicity in setting up and accessing the project.

Deployment Documentation
- Detailed instructions were provided to guide the setup and deployment process on PythonAnywhere, ensuring ease of access for users and evaluators.

Live Demo Preparation

Demo Content
- The live demo is prepared to showcase key features of StockSync, including user registration, stock trading simulations, portfolio management, and market trends analysis.

Team Participation

- Each team member will actively participate in the demo, highlighting their contributions and explaining the functionalities they were responsible for.

Conclusion

StockSync represents a collective effort to bridge the gap between theoretical knowledge and practical experience in stock trading. Through a user-centric design, real-time data integration, and a solid technical foundation, the project aims to educate and empower users to navigate the complexities of the stock market confidently.