Department of Computer Science and Engineering

BENGALURU, KARNATAKA, INDIA.

B. TECH. (CSE)
V SEMESTER
Aug. – Dec. 2023

UE21CS341A
SOFTWARE ENGINEERING PROJECT REPORT
ON

STOCK PRICE PREDICTION

SUBMITTED BY

PES1UG21CS102 - ANUSHKA GHEI
PES1UG21CS103 - ANVESHA NAYAK
PES1UG21CS104 - APOORVA RAJ
PES1UG21CS112 - ARVIN NOOLI

TABLE OF CONTENTS

1. PROJECT PROPOSAL

We propose to undertake a project titled "Reinforcement Learning for Stock Price Prediction". Departing from traditional supervised learning, our project leverages reinforcement learning, enabling the model to interact with the stock market environment and learn optimal strategies for maximising cumulative rewards. The project encompasses rigorous stages, including requirements gathering, system design, Python implementation, thorough testing, evaluation, and optimization. We will also create comprehensive documentation to facilitate ease of understanding and maintenance. With the successful deployment of the system for real-time stock price prediction, we anticipate providing valuable insights and contributing to the evolution of data-efficient, adaptable models in the financial sector.

2. SOFTWARE REQUIREMENT SPECIFICATION

1. Introduction

1.1 Purpose

The primary objective of this project is to develop a system capable of making automated and data-driven stock price predictions. This eliminates reliance on human intuition, which may result in more informed investment decisions. The system uses reinforcement learning to provide quick and accurate forecasts that can keep up with the fast-paced,dynamic nature of financial markets.

This is version 1.0 of the SRS document, marking the initial release.This SRS covers the entire system rather than just a particular subsystem, from data collection and preprocessing to deploying the trained model for real-time forecasts.

1.2 Intended Audience

The document is intended to meet the needs of the following :

1. Developers:contains comprehensive technical requirements, algorithms and architectural insights for developers working on the design, coding, and implementation of the software system.
2. Project Managers in charge of managing the system's development and delivery will obtain a grasp of the project's scope, objectives, and timescales. This document explains the main milestones and needs for the project, which will aid in project planning and management.
3. Data Scientists and Machine Learning Engineers working on the project's reinforcement learning parts will find details on algorithms, data sources, and model training needs.
4. Testers: Information about test cases, anticipated results, and acceptance criteria. This article serves as a guide for the unit, integration, and system testing processes.
5. Writers of documentation: contains references for those responsible for producing user manuals, technical manuals, and code documentation. It offers information on how to use and maintain the system as well as insights into its features.
6. Users: A high-level summary of the project's goals and scope is available for end users or stakeholders who want to know the capabilities and restrictions of the Stock Price Prediction system.

1.3 Product Scope

The main goal of this software system is to improve the decision-making process in the financial markets by providing precise and automated stock price predictions.

1. Make informed decisions while buying and selling.
2. Manage risks by estimating possible losses.
3. Optimising portfolio diversification.
4. Automated trading techniques.
5. Discover fresh insights by analysing quantitatively previous stock data.

The development and deployment of this software system align with corporate goals and business strategies in several ways:

1. Rise in profits for both individual investors and financial organisations as a result of better trading selections.
2. Competitive edge in the financial markets
3. Risk Mitigation and reducing potential monetary losses.
4. Regulatory Compliance: It is essential that financial markets abide by ethical and legal standards, and this software may be designed to do so.

1.4 References

- Title: Stock Market Prediction and Investment using Deep Reinforcement Learning- a Continuous Training Pipeline

  Authors: Amritha Sharma R, Debjyoti Guha, Hitesh Agarwal, Kothiya Meetkumar Harshadbhai

  Source: Stock Market Prediction and Investment using Deep Reinforcement Learning- a Continuous Training Pipeline
- Title: Predicting Stock Prices using Reinforcement Learning

  Authors:

  This article was published as a part of the Data Science Blogathon.

  Ekta Shah — Updated On December 23rd, 2020

  Source: Predicting Stock Prices using Reinforcement Learning (with Python Code!)
- Title: Ralph Lauren Corp - Class A Stock Forecast, "RL" Share Price Prediction Charts

  Source: Ralph Lauren Corp - Class A Stock Forecast: up to 119.031 USD! - RL Stock Price Prediction, Long-Term & Short-Term Share Revenue Prognosis with Smart Technical Analysis
- Title: How To Automate The Stock Market Using FinRL (Deep Reinforcement Learning Library)?

  Version: Published on January 5, 2021 In Mystery Vault

  Source: How To Automate The Stock Market Using FinRL (Deep Reinforcement Learning Library)?

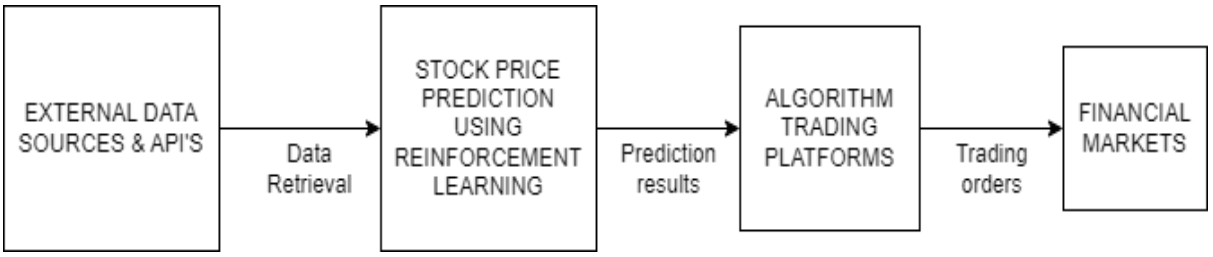
2. Overall Description

2.1 Product Perspective

It is not intended to replace any existing systems directly but rather to supplement and improve the skills of investors, traders, and financial experts in the area of stock trading and investment.

Although the system is intended to be used independently, it may have the following interfaces with bigger systems:

1. To obtain historical and real-time stock price data, the system may link with other data sources, such as financial market data providers.
2. There might be interfaces with algorithmic trading platforms or order execution systems for users who want to automate their trading strategies based on the system's predictions.
3. To visualise prediction outcomes, historical data analysis, and portfolio performance, system users may need interfaces to reporting and analytics tools.



2.2 Product Functions

2.2.1 Data Retrieval and Preprocessing:
Obtain historical and real-time stock price information from external sources and APIs. Preprocess and clean the data to guarantee accuracy and consistency.
2.2.2. Feature engineering,
Includes the use of technical tools like the Relative Strength Index (RSI) and moving averages.
2.2.3. Reinforcement Learning Model:
Implement Markov Decision Processes (MDPs) and Q-learning. Using historical data, train and fine-tune the reinforcement learning model.
2.2.4. Generating Predictions:
Create real-time stock price forecasts using the taught reinforcement learning model. Calculate probability and confidence intervals.
2.2.5. User Interface:
Offer a simple way for people to communicate with the system. Showcase the outcomes of previous data analysis and pertinent insights.
2.2.6. Algorithmic Trading Integration:
automated trading based on forecasts made by the system. Produce buy and sell signals, as well as carry out trading orders.
2.2.7. Reporting and Analytics:
Enables users to analyse data in-depth, track portfolio performance, and visualise prediction outcomes.

2.2.8. Documentation:
User manuals and technical guides. Provide tools for support and assistance for users to troubleshoot problems.
2.2.9. Testing and Evaluation: -
Confirm the system's accuracy and dependability.
2.2.10. Deployment and Maintenance:
Support deployment and enable ongoing maintenance and updates to better adapt to shifting market conditions.

2.3 User Classes and Characteristics

The system is expected to be utilised by a variety of user classes, each of which will have varied roles, levels of technical expertise, usage patterns, and goals, as well as distinct characteristics and requirements.

2.3.1 Investors and traders:
These users may have a high usage frequency and a thorough knowledge of the financial markets. They depend on the system's precise stock price predictions to guide their trading choices. They need the capacity to connect predictions with their trading platforms, real-time prediction outcomes, and historical data analysis.

2.3.2 Data Analysts:
Data analysts utilise a system for quantitative analysis of historical stock data to find trends and patterns. They need access to historical data, tools for data visualisation and the ability to export data for further analysis.

2.3.3 Machine Learning Engineers:
These users are in charge of configuring and training the reinforcement learning model. They are well-versed in its algorithms.They must have access to training data, model parameters, and choices for fine-tuning the model.

2.3.4 Algorithmic traders :
Traders utilise the system to build and execute automatic trading strategies based on the system's predictions. They could have advanced technical abilities. They must be able to integrate with algorithmic trading systems, generate trading signals, and execute orders.

2.3.5 Financial Consultants:
The method provides financial analysts with information regarding stock price changes and associated threats. They may utilise the system less frequently, but they require comprehensive analytic tools. Financial analysts must have access to thorough historical data analysis, risk assessment, and reporting capabilities.

2.3.6 Administrators:
Administrators are in charge of system maintenance, updates, and user administration. They require administrative access as well as technical knowledge. They must have tools for system configuration, security management, and user role assignment.

2.3.7 Novice Users:
Inexperienced users may have minimal trading or machine learning experience. They want an easy-to-use interface and guidance on how to use the system efficiently. Clear and straightforward user interfaces, tooltips, and user documentation are essential.

2.3.8 Compliance officers:

Compliance officers guarantee that the system adheres to financial market ethical and regulatory requirements. To meet regulatory requirements, features such as transparency, audit trails, and compliance reporting may be required.

Traders, investors, machine learning engineers, and algorithmic traders are the most crucial user classes for this product because they rely heavily on the system's forecasts and have certain technical needs.

2.4 Operating Environment

2.4.1 Hardware Platform:

It can be used to perform data analysis and machine learning activities on popular hardware platforms.

2.4.1.1. Processor (CPU): A multi-core processor with enough processing power to effectively handle feature extraction, feature preprocessing, and model training.

2.4.1.2. Large datasets and model training can be accommodated by having enough memory (RAM). Although more memory can be required to manage large amounts of data, 8 GB is the minimum suggested.

2.4.1.3. Storage: Enough space to store both historical and current stock price information. For quicker data retrieval and preprocessing, SSDs are desirable.

2.4.2 Operating System:

The program should work with widely used operating systems. The following versions and operating systems are supported:

Windows 10 or a later version.

macOS: 10.15 (Catalina) or a later version.

Linux: Ubuntu 18.04 LTS or later, as well as other well-known Linux releases.

2.4.3 Software Components:

1. Programming Language: Python for machine learning and system development. Python 3.x should be set up for users.
2. Machine learning frameworks: Reinforcement learning techniques and model training can be implemented using libraries like TensorFlow, PyTorch, or scikit-learn.
3. Data processing tools include Pandas and NumPy for preparing and manipulating data.
4. Web framework like Flask or Django.
5. Storage: Historical data can be stored in a database like PostgreSQL or MySQL.
6. Algorithmic Trading Platforms: For algorithmic trading functionality, integration with trading platforms like MetaTrader, Interactive Brokers, or custom trading APIs may be required.

2.5 Design and Implementation Constraints

The system operates within strict design and implementation constraints to ensure ethical, legal, and secure functionality. Adherence to corporate and regulatory policies is paramount, requiring compliance with data privacy regulations, transparency, ethical use, risk disclosure, and financial market regulations. Hardware limitations demand efficient utilisation of processing power, memory, and storage space, optimising the system's performance. Moreover, robust security measures are implemented, encompassing data encryption, access control, vulnerability management, incident response, data backup, and adherence to industry security standards. These constraints collectively guarantee the system's integrity, protect user data, and establish trustworthiness in the financial markets, essential for its successful operation and user confidence.

2.6 Assumptions and Dependencies

The system relies on several assumptions to function effectively. It assumes access to reliable external data sources for accurate stock price information and availability of necessary hardware and software resources for system operation. Dependencies include external interfaces, particularly reliance on external data providers for real-time and historical stock price data. Any discrepancies in these assumptions or dependencies could impact the system's functionality, requiring alignment with these factors for seamless operation.

3. External Interface Requirements

3.1 User Interfaces

The system provides users with an intuitive interface, featuring clear navigation, interactive data visualisations, and tooltips for novice users. It supports data export for further analysis and ensures a seamless user experience.

3.2 Software Interfaces

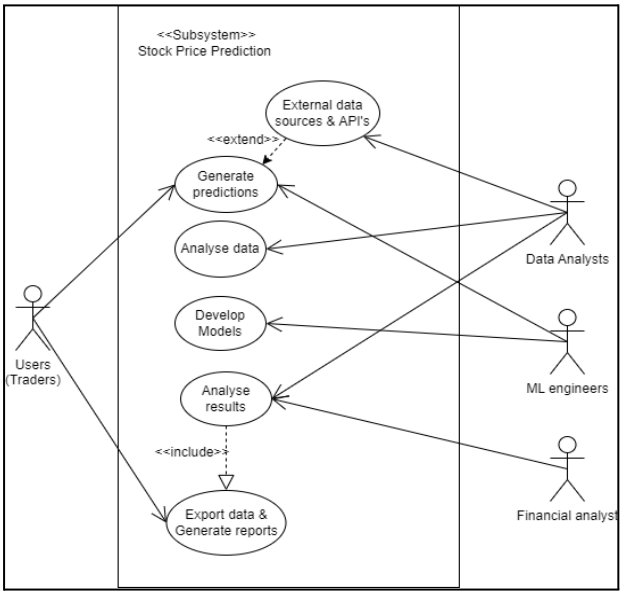
The system interfaces with external data sources for stock price data, ensuring accurate and up-to-date information. Additionally, it may integrate with algorithmic trading platforms and reporting tools for enhanced functionality, enabling users to implement trading strategies based on predictions and visualise prediction outcomes efficiently.

3.3 Communications Interfaces

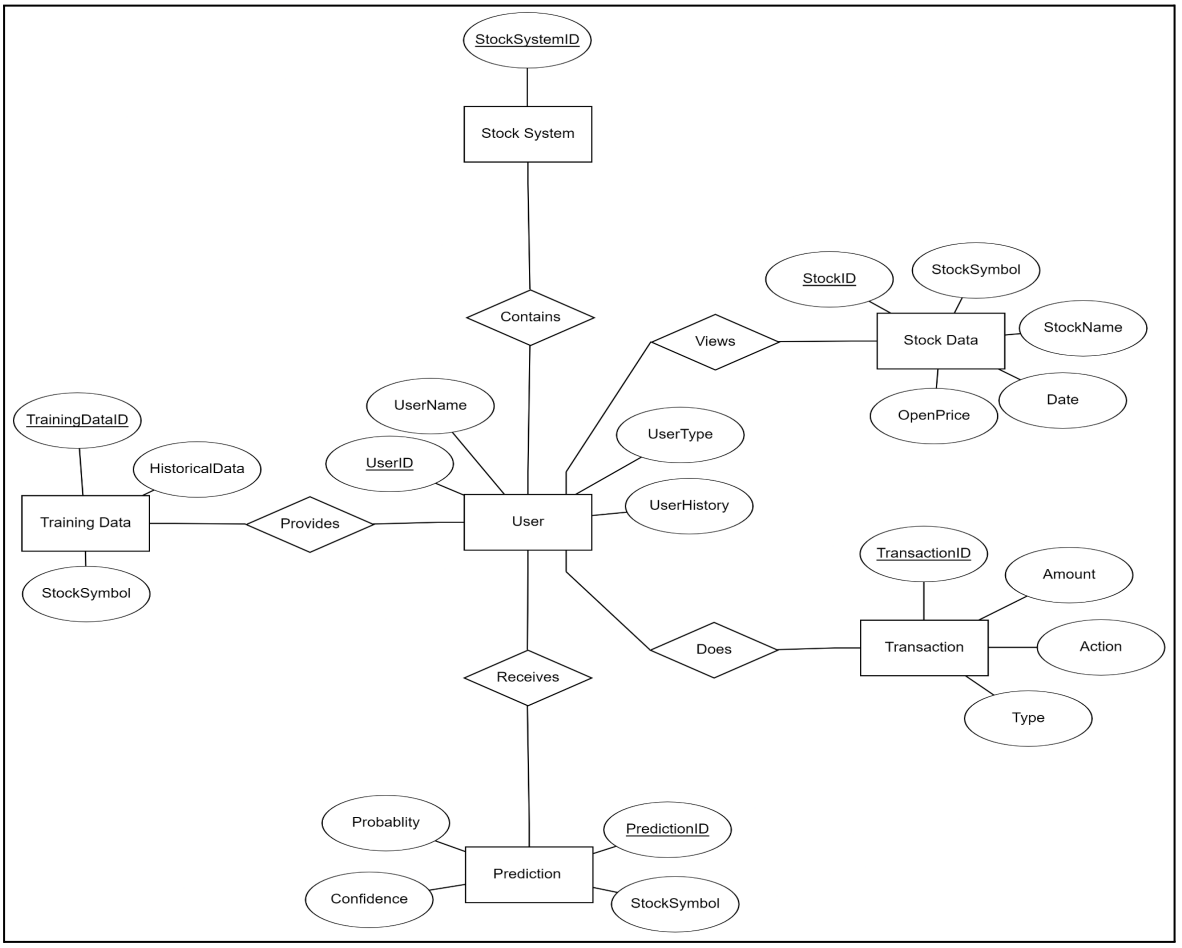
The system relies on standard internet communication protocols such as HTTP and FTP for data retrieval. It prioritises secure communication, implementing encryption to protect user data during transmission. The system adheres to stringent communication standards, ensuring data transfer integrity and user security.These interfaces collectively enable smooth communication between the 'Stock Price Prediction' system and its users, external data sources, and supplementary trading platforms, ensuring accurate predictions and user satisfaction.

4. Analysis Models

4.1 USE CASE DIAGRAM:



4.2 Entity Relationship Model:



5. System Features

5.1 Training Reinforcement Learning Model

5.1.1 Description and Priority

This feature involves training a reinforcement learning model on historical stock data to predict future stock prices.

Priority: High.

5.1.2 Stimulus/Response Sequences

- User provides historical stock data and model training parameters.
- System processes the data and trains the reinforcement learning model.
- System responds with a trained model or training success/failure notification.

5.1.3 Functional Requirements

- REQ-1: The system must preprocess historical stock data to be suitable for reinforcement learning.
- REQ-2: The system must implement the Q-learning algorithm for reinforcement learning.
- REQ-3: The system must allow users to specify and adjust training parameters.
- REQ-4: The system must provide feedback on the training process and convergence.
- REQ-5: The system must store the trained model for future use.

5.2 Predicting Stock Prices

5.2.1 Description and Priority

This feature involves using the trained reinforcement learning model to predict stock prices based on current market conditions.

Priority: High.

5.2.2 Stimulus/Response Sequences

- User requests stock price prediction for a specific stock and provides current market data.
- System processes the data using the trained model and predicts the stock price.
- System responds with the predicted stock price and confidence level.

5.2.3 Functional Requirements

- REQ-6: The system must integrate with real-time market data sources.
- REQ-7: The system must apply the trained model to predict stock prices.
- REQ-8: The system must provide a confidence level for each prediction.
- REQ-9: The system must allow users to request predictions for specific stocks.

5.3 User Interface for Prediction Visualization

5.3.1 Description and Priority

This feature involves providing a user interface for users to visualise the predictions made by the system.
Priority: Medium.

5.3.2 Stimulus/Response Sequences

User navigates to the prediction visualisation section.
User selects a specific stock or stocks for prediction visualisation.
System retrieves and displays the predicted stock prices along with confidence intervals.

5.3.3 Functional Requirements

REQ-10: The system must have an intuitive and easy-to-use interface for prediction visualisation.
REQ-11: The system must allow users to select and view predictions for specific stocks.
REQ-12: The system must display predicted stock prices with associated confidence intervals.
REQ-13: The system must provide visual aids like graphs or charts for better prediction understanding.

5.4 Algorithmic Trading Integration

5.4.1 Description and Priority

This feature involves integrating the system with algorithmic trading platforms to enable automated trading based on predictions.
Priority: High.

5.4.2 Stimulus/Response Sequences

User configures automated trading parameters based on predictions.
Users enable automated trading for selected stocks.
System communicates with the algorithmic trading platform to execute trading orders.

5.4.3 Functional Requirements

REQ-14: The system must integrate with algorithmic trading platforms or APIs.
REQ-15: The system must generate buy and sell signals based on predictions.
REQ-16: The system must communicate with the trading platform to execute trading orders.
REQ-17: The system must provide feedback on the outcome of trading orders.

5.5 Reporting and Analytics

5.5.1 Description and Priority

This feature involves enabling users to analyse data in-depth, track portfolio performance, and visualise prediction outcomes.
Priority: Medium.

5.5.2 Stimulus/Response Sequences

User navigates to the reporting and analytics section.
User selects the type of analytics or reporting they want to view.
System processes the data and displays the requested analytics or reports.

5.5.3 Functional Requirements

REQ-18: The system must provide options for different types of analytics and reporting.
REQ-19: The system must allow users to customise the parameters for analytics and reporting.
REQ-20: The system must display analytics and reports in a clear and understandable format.

5.6 Documentation

5.6.1 Description and Priority

This feature involves providing comprehensive user manuals and technical guides to assist users in using and maintaining the system effectively.
Priority: Medium.

5.6.2 Stimulus/Response Sequences

User accesses the documentation section.
User selects the specific documentation they need (user manual, technical guide, etc.).
System displays the selected documentation.

5.6.3 Functional Requirements

REQ-21: The system must provide user manuals and technical guides in a user-friendly format.
REQ-22: The system must cover all aspects of using and maintaining the system effectively.
REQ-23: The system must allow for easy navigation within the documentation.

5.7 Testing and Evaluation

5.7.1 Description and Priority

This feature involves confirming the system's accuracy and dependability through testing and evaluation processes.
Priority: High.

5.7.2 Stimulus/Response Sequences

Testing team initiates various test cases to evaluate system performance.
System processes the test cases and generates test results.
System provides feedback on passed and failed tests.

5.7.3 Functional Requirements

REQ-24: The system must support unit, integration, and system testing processes.
REQ-25: The system must have predefined test cases covering all functionalities.
REQ-26: The system must provide detailed test results and logs for analysis.

5.8 Deployment and Maintenance

5.8.1 Description and Priority

This feature involves supporting deployment and enabling ongoing maintenance and updates to better adapt to shifting market conditions.
Priority: High.

5.8.2 Stimulus/Response Sequences

Administrator initiates the deployment process.
System processes the deployment and verifies successful deployment.
System provides tools and processes for ongoing maintenance and updates.

5.8.3 Functional Requirements

REQ-27: The system must support easy and efficient deployment processes.
REQ-28: The system must provide tools and mechanisms for ongoing maintenance and updates.
REQ-29: The system must ensure minimal downtime during deployment and updates.

6. Other Non Functional Requirements

6.1 Performance Requirements

6.1.1 NON Real-Time Processing:

The system will provide non real-time predictions for stock market movements based upon a pre-trained dataset. The primary objective is to outline the functionality and parameters of the predictive system.It will utilise historical stock data, technical indicators, and potentially fundamental factors to generate predictions regarding future stock prices.In further versions of this model we will implement Real-time processing as it is essential to ensure that users receive timely and accurate predictions for making informed decisions in the volatile market.

6.1.2 Latency:

Latency refers to the time delay between the occurrence of a market event and the system's response in generating predictions. Minimising latency is critical for real-time decision-making in the dynamic and fast-paced environment of stock trading. The system must be designed to process and analyse market data swiftly, ensuring that predictions are delivered to users in near real-time. Low-latency performance is essential to enable traders to make timely and informed decisions based on the latest market trends.

6.1.3 Scalability:

To accommodate the dynamic nature and growing volume of financial data inherent in stock market environments, the solution must be scalable. This entails creating an architecture and employing algorithms that can handle an increasing number of financial instruments, diverse data sources, and a growing user base with ease. The system should be capable of processing large datasets efficiently, adapting to changing market conditions, and seamlessly integrating new features or modules.

6.1.4 Throughput:

The system should be able to process a minimum of 50 stock market transactions per minute.Throughput refers to how quickly the predictive model processes and analyses financial data to produce actionable insights.It highlights the system's efficiency in handling large volumes of real-time market data and should be optimised to process historical, live data quickly, allowing investors and traders to make timely decisions. A high throughput ensures that the predictive algorithms can quickly adapt to market fluctuations, improving the system's responsiveness and providing users with accurate and timely predictions to inform their investment strategies.

6.1.5 Accuracy:

The prediction accuracy of the system should be at least 70%.The algorithm's precision is defined by its ability to forecast stock movements precisely based on historical data and relevant indicators. The criteria and parameters influencing the accuracy metrics, which include factors such as data sources, machine learning models used, and the frequency of updates, should all be present in adequate amounts.

6.1.6 Resource Utilisation:

Resource Utilisation refers to the efficient use of computational resources in order to improve the accuracy and speed of predictive models. The system will analyse historical stock data, market trends, and relevant financial indicators using cutting-edge algorithms, machine learning techniques, and statistical analysis. Cloud computing resources will be optimised for large datasets, ensuring scalability and responsiveness. Furthermore, the system will include parallel processing capabilities to speed up model training and prediction, reducing latency. The allocation of resources will be dynamic, adapting to changing workloads and taking into account factors such as market volatility and data volume. This strategic resource allocation aims to enable the stock market prediction system to provide timely and reliable forecasts, thereby improving overall performance and user satisfaction.

6.1.7 Availability:

Mechanisms must be in place to address potential downtimes, data outages, or other disruptions in order to ensure a robust and reliable stock market prediction system. Implementing failover solutions, regular backups of critical data, and quick recovery protocols are all part of this. By prioritising resource availability, the stock market prediction system aims to maintain uninterrupted functionality, instilling confidence in users and stakeholders in the predictive analytics platform's reliability.

6.2. Safety Requirements

6.2.1 Data Integrity and Confidentiality:

Maintaining the system's safety and reliability requires ensuring the integrity and confidentiality of data. Data integrity measures will be put in place to ensure the accuracy and consistency of the information used in stock market forecasts. This includes using cryptographic techniques and checksums to detect and prevent unauthorised data changes. Confidentiality safeguards will be put in place to protect sensitive financial data from unauthorised access, with only authorised personnel having access. To protect the confidentiality of stock market data, robust encryption protocols and access controls will be implemented, preventing any compromise that could impact prediction reliability and overall system security.

6.2.2 Disaster Recovery and Backup:

Disaster recovery and backup mechanisms are essential for ensuring the integrity and availability of critical financial data. Implementing a solid Disaster Recovery plan entails taking preventative measures to lessen the impact of unforeseen events such as system failures, natural disasters, or cyber-attacks. Regular and automated backups of critical datasets, code repositories, and system configurations are included. Backups act as a safety net, allowing for quick recovery in the event of data corruption or loss. Furthermore, disaster recovery protocols are designed to minimise downtime, ensuring that stock market prediction systems can quickly resume normal operations, protecting against potential financial losses and maintaining the predictive models' overall reliability.

6.2.3 Compliance with Financial Regulations:

The Stock Market Prediction System must incorporate robust safety measures in order to comply with stringent financial regulations, as outlined in this document. Compliance with regulatory standards is critical to ensuring the financial operations' integrity, security, and transparency. The system will use advanced encryption protocols to protect sensitive financial data while also ensuring confidentiality and preventing unauthorised access. Furthermore, stringent authentication mechanisms will be used to control user access and prevent potential breaches. Comprehensive audit trails will be set up to track and monitor system activity, making regulatory compliance audits easier.

6.2.4 Market Data Accuracy:
The reliability and precision of market data serve as the foundation for the system's predictive capabilities. To minimise risks and improve the robustness of stock market forecasts, the SRS requires stringent measures for market data verification, validation, and real-time updating. To foster the system's trustworthiness and effectiveness, accurate and up-to-date information on stock prices, trading volumes, and relevant financial indicators is required. Adherence to data accuracy standards not only protects against incorrect predictions but also strengthens the system's integrity, instilling trust among users and stakeholders in the volatile world of stock market analysis.

6.2.5 Error Handling:
Since financial markets are dynamic and unpredictable, the SRS must include comprehensive error detection and recovery strategies. This entails putting in place real-time data validation protocols to detect anomalies or inconsistencies in the incoming stock data. Furthermore, the SRS should include error-handling routines that not only log and report errors but also initiate corrective actions. These safeguards are critical for ensuring system stability, preventing erroneous predictions, and ensuring the overall safety and credibility of the stock market forecasting platform.

6.2.6 Ethical Use:
Transparency, fairness, and accountability must be prioritised in the development and deployment of predictive models. Safety measures must include mechanisms to prevent financial market manipulation and ensure that the system's predictions are based on reliable and unbiased data. The SRS should explicitly define safeguards against insider trading, market manipulation, and any other unethical use of predictive capabilities. Furthermore, the document should emphasise the importance of ongoing monitoring and auditing in order to detect and correct any unintended consequences or biases that may arise during system operation. The ethical foundation of stock market prediction systems is critical not only for regulatory compliance but also for fostering trust among financial ecosystem users and stakeholders.

6.3 Security Requirements

6.3.1 User Authentication:
The system will employ a multi-layered authentication process that will necessitate users providing unique credentials such as username and password. Furthermore, two-factor authentication (2FA) will be required to add an extra layer of security and reduce the risk of unauthorised access. To protect sensitive information, user accounts will be securely stored in a hashed format using encryption protocols. User activities, particularly those related to stock market predictions, will be logged and monitored to improve accountability.

6.3.2 Access Controls:
Given that financial data is dynamic and sensitive, robust access control mechanisms are required to regulate and restrict user interactions. Role-based access controls should be implemented in the system, with specific privileges defined for different user roles such as administrators, analysts, and viewers. Authentication protocols, such as strong password policies and multi-factor authentication, must also be in place to protect the system from unauthorised access. This strict access control framework protects not only the confidentiality of proprietary algorithms and market data, but also the credibility of the predictions generated, fostering a secure and reliable environment for stock market analysis.

6.3.3 Data Encryption:

Data encryption is critical in ensuring the integrity of stock market prediction models and the data they contain. The Stock Market Prediction System reduces the risk of unauthorised access and tampering with critical data by utilising strong encryption algorithms such as Advanced Encryption Standard (AES) or RSA. Encryption converts raw stock market data into a secure format that can be deciphered only with the appropriate decryption key, providing a strong defence against potential threats and unauthorised interventions.

6.3.4 Secure Data Storage:

In the context of stock market prediction safety requirements outlined in a Software Requirements Specification (SRS) document, secure data storage is critical. Robust security measures must be implemented to protect sensitive financial data and the integrity of market forecasting models. This includes employing encryption techniques, access controls, and secure backup protocols to safeguard data against unauthorised access, data breaches, and loss. The SRS document establishes a critical foundation for trust and reliability in stock market prediction systems by ensuring secure data storage, reinforcing their resilience in the face of potential threats and vulnerabilities.rmation.

6.3.5 Audit Trails:

Audit trails serve as a comprehensive record-keeping mechanism for the predictive model system, meticulously documenting every transaction, decision, or modification. These trails not only improve transparency, but also add a layer of security by allowing for traceability and accountability in the event of discrepancies or errors. Audit trails serve as a safeguard against unauthorised access, manipulation, or malfeasance in the volatile world of stock market predictions, where accuracy and reliability are critical. Audit trails contribute to the overall integrity of the system by systematically logging all activities related to predictive algorithms and data inputs, ensuring compliance with regulatory standards, and fostering trust among users and stakeholders.

6.3.6 Security Patching:

Continuous and timely security patching is required to protect the system from potential vulnerabilities and threats. In the volatile world of stock market forecasting, where accuracy and reliability are critical, ensuring the timely application of security patches becomes a critical aspect of risk mitigation. These patches are preventative measures, addressing identified software vulnerabilities and protecting sensitive financial data from unauthorised access or manipulation. Integrating a robust security patching mechanism into the system's design not only improves its resilience, but also aligns it with industry best practises for maintaining the integrity and confidentiality of financial data in the realm of stock market predictions.

6.3.7 Data Backup and Recovery:

Data Backup and Recovery describes the mechanisms in place to protect critical financial data. Robust backup protocols are defined to duplicate and store historical stock market data on a regular basis, reducing the risk of data loss due to unforeseen circumstances such as hardware failures, cyber threats, or system errors. Furthermore, the document outlines the recovery procedures that will be used in the event of data corruption or loss, emphasising the importance of a quick and dependable restoration process. The stock market prediction system aims to maintain the highest level of data reliability by incorporating these measures into the SRS, ensuring that accurate and timely information is available for effective market analysis and decision-making.

6.3.8 Incident Response Plan(IRP):

The IRP outlines a systematic approach to dealing with potential security incidents in the dynamic landscape of stock market prediction, where accuracy and reliability are critical. It specifies predefined procedures for detecting anomalies, mitigating threats, and recovering from disruptions in order to ensure continuous functionality. To protect against cyber threats and data breaches, this plan includes real-time monitoring, threat intelligence integration, and rapid response mechanisms. By incorporating the IRP into the SRS, the stock market prediction system not only anticipates potential risks, but also creates a structured framework for adaptive and proactive security measures, thereby strengthening the predictive analytics platform's reliability and safety.

6.3.9 Privacy Controls:

The system must include strict safeguards to ensure the confidentiality and integrity of user information, as well as to prevent unauthorised access and data breaches. User authentication protocols, encryption mechanisms for data transmission and storage, and access management functions should all be included in privacy controls. Furthermore, the SRS should establish clear guidelines for user consent, allowing individuals to make informed decisions about the collection and use of their personal data for predictive analytics. By incorporating comprehensive privacy controls, the stock market prediction system can not only adhere to regulatory compliance but also foster user trust, reinforcing the overall application's security foundation.

6.4 Software Quality Attributes

6.4.1. Accuracy:

The system's primary goal is to achieve a prediction accuracy of 85%, allowing users to confidently rely on its insights for informed investment decisions. This characteristic is critical for establishing user trust and credibility in the volatile landscape of stock market forecasting.

6.4.2. Reliability:

The system prioritises reliability, offering users consistent and dependable predictions while aiming for a mean time between failures (MTBF) of 1000 hours. Users who rely on timely and accurate information to navigate the complexities of financial markets require reliability.

6.4.3. Availability:

The system ensures uninterrupted access for users by ensuring 90% availability during market hours, especially during critical trading periods. High availability is critical for building user trust and facilitating seamless interaction with the prediction platform.

6.4.4. Maintainability:

The system prioritises ease of maintenance, allowing for quick updates and modifications and emphasising a mean time to repair (MTTR) of 4 hours. This trait is essential for adapting to changing market conditions and quickly addressing any system issues.

6.4.5. Scalability:

The system prioritises scalability by being designed to handle a 50% increase in users and data volume without significant performance decline. This ensures that the platform can expand with the user base and data demands as they grow.

6.4.6. Usability:

The system prioritises a user-friendly interface, with a target usability score of 80% in user satisfaction surveys. This emphasis on usability improves user interactions, resulting in a more positive overall experience and fostering long-term engagement with the prediction system.

6.4.7. Portability:

The system ensures portability by prioritising compatibility with major operating systems (Windows, macOS, and Linux). This feature enables users to access the prediction system from a variety of devices and operating systems, increasing flexibility and user accessibility.

6.4.8. Robustness:

The system emphasises robustness, gracefully handling unexpected inputs and adverse conditions, with a failure rate of less than 1%. This characteristic ensures the system's resilience in the face of unexpected challenges, contributing to its overall effectiveness and dependability.

6.7.9. Testability:

The system prioritises high testability by aiming for at least 90% test coverage for critical functionalities. This feature allows for the early detection and resolution of issues during both the development and maintenance phases, ensuring the system's ongoing quality and performance.

6.7.10. Adaptability:

The system positions itself for long-term success by prioritising adaptability to changes in market conditions and regulatory requirements with minimal disruption. This characteristic recognises the volatile nature of financial markets, allowing the system to evolve in tandem with industry shifts.

6.4.11. Reusability:

The system promotes efficiency by emphasising the design of code components and algorithms for reusability in different modules or future projects. This property reduces development time, improves consistency across various system elements, and facilitates future endeavours with streamlined efforts.

6.5 Business Rules

6.5.1. Conceptualise the Prototype:

When developing the prototype for our stock market prediction service, we focus on creating a Minimum Viable Product (MVP) that encapsulates key features. The MVP includes basic prediction algorithms that allow users to make educated investment decisions. User registration ensures a personalised experience, while an intuitive user interface allows for quick access to predictions. We can deliver a functional prototype that addresses key business aspects, laying the groundwork for a robust and scalable stock market prediction platform, thanks to this strategic approach.

6.5.2.Technical Development:

In terms of technical development, our stock market prediction system prioritises a resilient data infrastructure for efficient financial data collection, storage, and processing. A key focus is algorithm development, which employs machine learning and data science techniques to constantly refine and improve prediction accuracy. At the same time, user interface (UI) design is critical for ensuring a user-friendly experience for accessing predictions across both web and mobile platforms, aligning our business with modern usability standards, and broadening accessibility for a diverse user base.

6.5.3.Business Features:

The Subscription Model, which incorporates dynamic business features into the stock market prediction service, provides users with various tiers, each of which provides a different level of service. The Freemium Option introduces a free version, enticing users to upgrade to unlock premium features. Customization allows users to tailor predictions based on their personal risk tolerance and investment goals. The platform provides timely non-real-time market updates, with the option of upgrading to real-time data, enhancing user decision-making capabilities. These features not only cater to a wide range of user requirements, but also create potential revenue streams, fostering a sustainable and adaptable business model for the stock market prediction service.

6.5.4.Regulatory Compliance:

Meticulous attention to regulatory compliance is required for successful business operations in the stock market prediction domain. Legal counsel is required to navigate the complex landscape of financial regulations and data protection laws. This ensures that the stock market prediction service operates within legal parameters, thereby mitigating potential risks and liabilities. Parallel to this, it is critical to develop clear and comprehensive Terms of Service and a Privacy Policy. These documents articulate the platform's rules and provide transparency into how user data is handled, fostering trust and accountability between the service provider and its users.

6.5.6.Iterative Development:

The Iterative Development tenet is critical for long-term success in stock market forecasting. Establishing a robust Feedback Loop is critical, soliciting user feedback on a regular basis to refine features and align the platform with changing user needs. Simultaneously, Scalability Planning is critical—anticipating and preparing for an increase in user base growth. Ensure that the infrastructure scales seamlessly with increased demand to avoid performance bottlenecks. This strategic approach not only improves the platform's agility and responsiveness, but also strengthens its ability to accommodate the volatile stock market landscape and meet the increasing demands of a growing user community.

6.6 Domain Requirements

6.6.1. Real-Time Market Data Integration:

The system's vitality stems from its seamless integration with real-time market data feeds, which provide users with up-to-the-minute insights into stock prices, trading volumes, and vital indicators, fostering informed decision-making in a volatile financial landscape.

6.6.2. Historical Market Data Analysis: By providing users with a retrospective lens, the system enables users to gain access to and analyse historical market data, revealing intricate patterns, trends, and correlations that are critical for making strategic and informed investment decisions.

6.6.3. Algorithm Integration: The system's versatility is highlighted by its support for various prediction algorithms, which allows users to integrate and customise algorithms to align predictions with their specific preferences and strategic objectives.

6.6.4. Risk Assessment and Management: In order to reduce uncertainty, the system provides users with powerful tools for assessing and managing risks associated with various investment strategies. Volatility and historical performance considerations ensure a comprehensive risk management approach.

6.6.5. Integration with Trading Platforms: The system defines itself through seamless integration with popular trading platforms, allowing users to execute trades directly from the prediction system, streamlining and enhancing the trading experience with real-time predictive insights.

6.6.6. News and Social Media Sentiment Analysis: In addition to traditional metrics, the system incorporates news and social media sentiment analysis, providing users with a comprehensive view of external influences that may affect stock prices and enriching predictive models with broader market dynamics.

6.6.7. Regulatory Compliance Monitoring: To stay ahead of regulatory changes, the system diligently monitors and reports on changes that may have an impact on financial markets. This keeps users informed about changing compliance requirements, encouraging responsible and compliant investment practices.

6.6.8. Machine Learning Model Training: At the cutting edge of innovation, the system provides users with tools for training machine learning models on historical data, allowing for a dynamic and personalised approach to predictive model development.

6.6.9. Sector and Industry Analysis: Users benefit from the ability to conduct in-depth analyses of specific sectors and industries, which helps them make better decisions by taking into account nuanced performance factors and trends within targeted market segments.

6.6.10. Portfolio Management Tools: The system provides robust portfolio management features to optimise investment strategies. Users can track, analyse, and fine-tune their investment portfolios with ease, ensuring alignment with long-term financial goals.

6.6.11. Dividend and Earnings Analysis: For those interested in delving into the complexities of financial health, the system provides tools for analysing dividend payments and earnings reports. This enables users to take corporate financial performance into account when making strategic investment decisions.

6.6.12. Market Sentiment Analysis: In order to decipher market sentiment nuances, the system analyses social media and news data. This useful information helps users anticipate potential market movements and make timely and informed investment decisions.

6.6.13. Automated Trading Strategies: The system, which is at the cutting edge of innovation, supports the implementation of automated trading strategies. Users can use predictive models and define criteria to automate trading, improving efficiency and precision.

6.6.14. Alerts and Notifications: The system allows users to customise alerts and notifications to meet their specific needs. This feature keeps users up to date on specific market conditions or events that may affect their investments, allowing for more proactive decision-making.

6.6.15. Tax Implications Reporting: To round out the financial picture, the system gives users the tools they need to generate reports on the tax implications of their trades and investments. This ensures compliance and transparency in the management of their financial portfolio's tax-related aspects.

7. Other Requirements

1. Database Requirements: The system should use a relational database management system (RDBMS) to store and manage historical market data, user profiles, and system logs. A structured database is essential for efficient data storage, retrieval, and analysis.

2. Internationalisation and Localization: The system should be designed to support internationalisation and localization, allowing users to access the platform in multiple languages and adapt to regional market conventions.Facilitates a broader user base and ensures usability across different regions.

3. Legal and Compliance Requirements: The system must comply with all relevant laws and regulations governing financial systems, data privacy, and electronic trading. Ensures legal compliance and reduces the risk of regulatory penalties.

4. Data Backup and Recovery: The system should regularly back up critical data and provide a robust data recovery mechanism in the event of system failures or data corruption. Mitigates the risk of data loss and ensures business continuity.

5. Third-Party Integrations: The system should allow for seamless integration with third-party tools, services, and APIs that are commonly used in the financial industry. Enhances the system's functionality and extends its capabilities through external integrations.

6. Documentation Standards: The development team must adhere to specified documentation standards for code, system architecture, and user manuals. Ensures maintainability and supports knowledge transfer within the development team.

7. User Support and Training: The system should provide comprehensive user support through documentation, FAQs, and a help desk. Additionally, training materials and sessions should be available for users. Enhances user experience and reduces the learning curve for new users.

8. Performance Monitoring and Reporting: The system should have built-in performance monitoring tools to track system health, resource utilisation, and response times. It should also generate performance reports. Facilitates proactive maintenance, optimization, and system performance analysis.

9. User Feedback Mechanism: The system should incorporate a mechanism for users to provide feedback and report issues, and there should be a process for addressing and resolving user feedback. Enables continuous improvement and addresses user concerns promptly.

10. Environmental Sustainability: The development process should consider environmental sustainability, and the system should be designed to minimise energy consumption and resource usage. Aligns with corporate social responsibility and sustainability goals.

11. Integration with Analytics Tools: The system should integrate with popular data analytics and visualisation tools to allow users to perform in-depth analysis on prediction results. Enhances the analytical capabilities of users and supports a more comprehensive understanding of market trends.

12. Open API for Customization: The system should provide an open API that allows users or third-party developers to customise and extend functionalities. Encourages innovation and allows for the development of custom solutions tailored to specific user needs.

13. Comprehensive Logging: The system should maintain detailed logs of user activities, system events, and errors for auditing and troubleshooting purposes. Supports forensic analysis and helps in identifying and resolving issues efficiently.

14. Comprehensive Reporting: The system should provide comprehensive reports on prediction accuracy, user engagement, and system performance.Facilitates decision-making, system optimization, and performance evaluation.

Appendix : Requirement Traceability Matrix

Sl No	Req ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	6.1	Performance requirements considering the dynamic and sensitive nature of financial markets	-	DFD level 0	-	-	-
2	6.2	Safety requirements ensure security,reliability, ethical use and maintains an audit trail	-	DFD level 0	-	-	-
3	6.6	Domain requirements for specific functionalities	-	DFD level 1	-	-	-
4	5.1	Training Reinforcement Learning Model	-	DFD level 2	CODE-3	UT-01	-
5	5.2	Predicting Stock Prices	-	DFD level 2	CODE-4	UT-02	-
6	5.3	User Interface for Prediction Visualization	-	DFD level 1	-	UT-03	-
7	5.4	Algorithmic Trading Integration	-	DFD level 2	CODE-3	UT-04	-
8	5.5	Reporting and Analytics	-	-	-	UT-05	-
9	5.6	Documentation	-	-	CODE	UT-06	-
10	5.7	Testing and Evaluation	-	DFD level 2	CODE-4	UT-07	-
11	5.8	Deployment and Maintenance	-	DFD level 2	CODE-4	UT-08, UT-09	-

3. PROJECT PLAN

1. Lifecycle to be followed for the execution

For this project, we have opted to adopt the Agile methodology as the primary lifecycle model. The Agile approach's iterative and adaptable framework is well-suited for the intricate task of developing predictive software for the stock market. This model facilitates the timely delivery of incremental updates in short development cycles, allowing us to promptly respond to the ever-changing dynamics of financial markets and user demands.

The choice of the Agile model is grounded in the inherent volatility of the stock market. Its constant fluctuations and the evolving requirements of users necessitate a development methodology that is flexible and responsive. The Agile approach enables continuous enhancement and fine-tuning of our predictive algorithms, ensuring our software remains agile and swiftly adaptable to real-time market changes. Furthermore, it ensures the user interface remains dynamic, aligning with evolving user expectations and industry standards.

Embracing the Agile methodology empowers us to deliver essential functionalities within compressed timeframes, maintaining a high standard of software quality. This approach allows us to promptly address shifts in market trends, swiftly cater to user requirements, and uphold a high level of adaptability throughout the software development lifecycle.

2. Tools used throughout the lifecycle

- **Planning Tool**: We employed Trello, a user-friendly agile project management tool, to efficiently organise and oversee our project's sprints and tasks. Trello offered the same essential functionalities as Jira, allowing us to effectively manage the project's progress by utilising boards and lists.
- **Design Tool**: Figma will be employed for the comprehensive design of the user interface and user experience (UI/UX), while UML (Unified Modeling Language) diagrams will delineate the system architecture, aiding in a clear visual representation of the software structure.
- **Version Control**: Git and GitHub will serve as our chosen version control system, facilitating collaborative development, version management, and code repository organisation.
- **Development Tool**: Visual Studio Code, a free and versatile Integrated Development Environment (IDE), supports various programming languages and streamlines coding efforts, providing a robust platform for efficient and streamlined software development.
- **Bug Tracking**: For issue tracking and resolution, we harnessed the capabilities of OpenProject, an open-source and free platform. OpenProject provided us with effective bug management and robust collaboration features, akin to what Bugzilla

offers, enabling us to address reported bugs and system issues efficiently.

- **Testing Tool**: Selenium, an automated testing tool, and JUnit for unit testing will be employed to ensure the accuracy and reliability of our software through automated and systematic testing procedures.

3. Deliverables

Database Management System (RDBMS):

- Category: Reusable
- Justification: RDBMS software like MySQL, PostgreSQL, or Oracle is a reusable component. These systems are widely used and can be configured and customized according to specific project needs. The choice of the specific RDBMS and its configuration will depend on project requirements, but the software itself is a reusable component.

Third-Party Integrations (APIs and Tools):

- Category: Reusable
- Justification: Third-party APIs and tools commonly used in the financial industry, such as market data APIs, payment gateways, and analytics tools like Tableau or Power BI, are reusable components. These APIs and tools are designed to be integrated into various systems and applications, making them reusable across different projects.

Internationalisation and Localization Libraries:

- Category: Reusable
- Justification: Libraries and frameworks for internationalisation and localization are reusable components. These libraries provide functionalities to translate and adapt the user interface and content to multiple languages and regional conventions. Developers can use these libraries to support internationalisation and localization in various projects.

Open API (Application Programming Interface):

- Category: Reusable
- Justification: An open API that allows users or third-party developers to customise and extend functionalities is a reusable component. Once developed, this API can be used across different projects, enabling innovation and the development of custom solutions tailored to specific user needs.

Prediction Algorithms and Machine Learning Models:

- Category: Build
- Justification: Prediction algorithms and machine learning models need to be built specifically for the stock market prediction system. These algorithms and models are tailored to the unique requirements of the system, including historical market data analysis, trend prediction, and risk assessment. They are custom-built components designed to address the specific needs of this project.

User Interface (UI) and User Experience (UX) Design:

- Category: Build
- Justification: The user interface and user experience design elements, including interactive charts, dashboards, and user interaction workflows, need to be custom-built for the stock market prediction system. These components are designed based on the system's requirements and user feedback, making them unique to this project.

User Authentication and Access Control System:

- Category: Build
- Justification: The user authentication and access control system, including multi-factor authentication, role-based access control, and secure storage of user credentials, needs to be custom-built for the system. These components are designed to ensure the security and integrity of user accounts and are specific to this project's requirements.

System Monitoring and Performance Tracking Tools:

- Category: Build
- Justification: Tools for system monitoring, performance tracking, and generating reports need to be custom-built for the stock market prediction system. These tools are designed to track system health, resource utilization, user engagement, and prediction accuracy, providing insights specific to this project's performance and user interactions.

Feedback Mechanism and Issue Resolution Process:

- Category: Build
- Justification: The mechanism for users to provide feedback and report issues, as well as the process for addressing and resolving user feedback, needs to be custom-built for the system. This component is designed to facilitate continuous improvement and user satisfaction, making it specific to this project's feedback requirements.

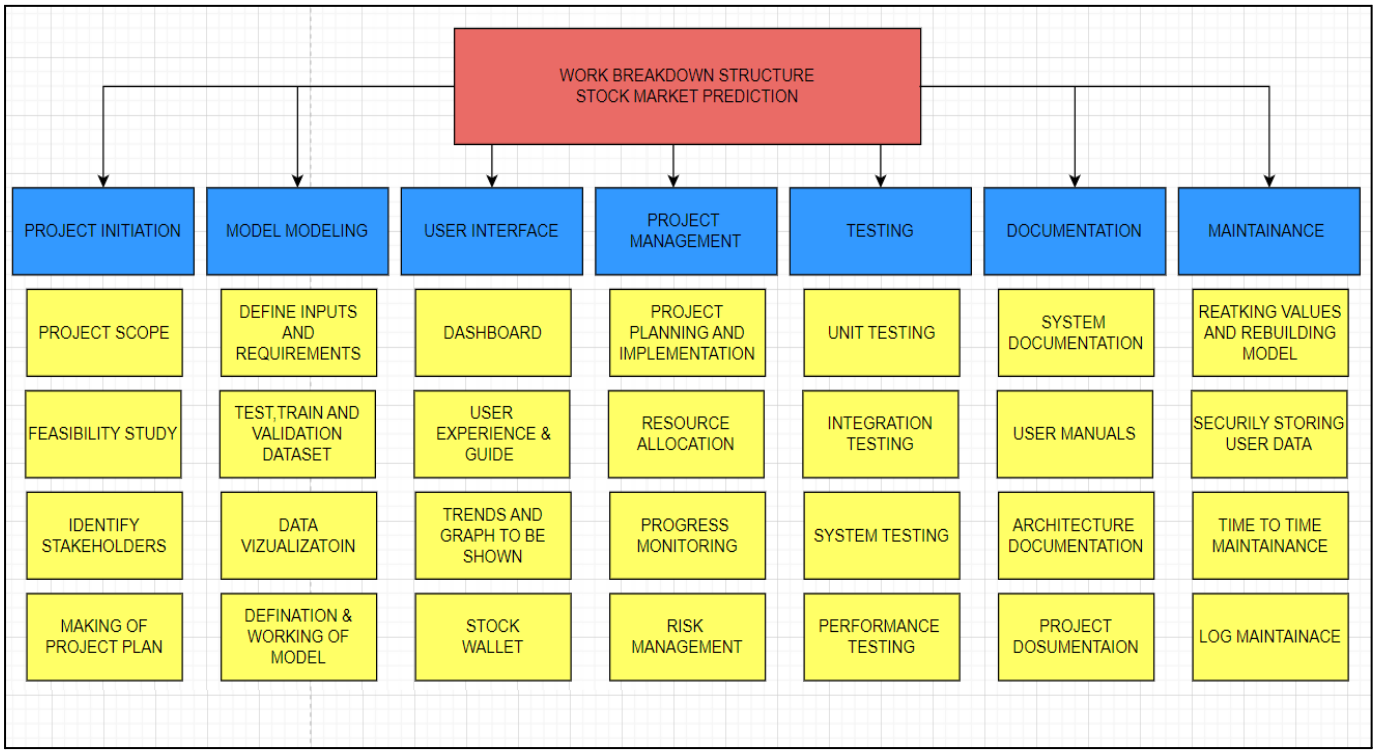
Environmental Sustainability Considerations:

- Category: Build
- Justification: Environmental sustainability considerations, including minimising energy consumption and resource usage, need to be integrated into the system's design and development process. These considerations are specific to this project and align with the project's corporate social responsibility and sustainability goals.

Comprehensive Logging and Reporting System:

- Category: Build
- Justification: The logging system for user activities, system events, errors, and comprehensive reporting on prediction accuracy, user engagement, and system performance needs to be custom-built for the stock market prediction system. These components are designed to meet the specific reporting and auditing requirements of this project.

4. Work Breakdown Structure



5. Rough estimate of effort required to accomplish each task in terms of person months.

1. Training Reinforcement Learning Model (Priority: High) :
- Preprocessing Historical Data: 1-2 person-months
- Q-learning Algorithm Implementation: 2-3 person-months
- Training Feedback and Storage: 1-2 person-months

2. Predicting Stock Prices (Priority: High)
- Real-time Data Integration: 1-2 person-months
- Prediction Implementation: 1-2 person-months
- Confidence Level Integration: 1-2 person-months

3. User Interface for Prediction Visualization (Priority: Medium)
- Interface Design and Development: 2-3 person-months
- Data Visualization Tools: 1-2 person-months

4. Algorithmic Trading Integration (Priority: High)
- Integration with Trading Platforms: 2-3 person-months
- Signal Generation and Order Execution: 1-2 person-months

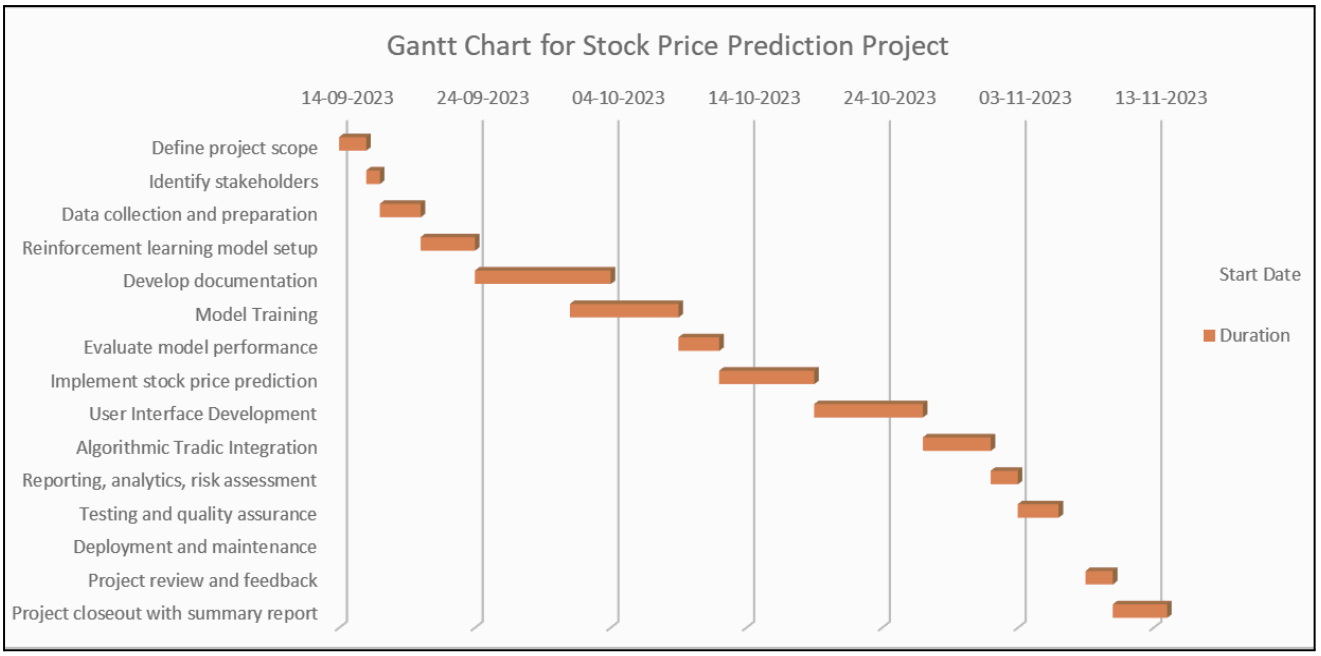
5. Reporting and Analytics (Priority: Medium)
- Analytics Tools: 2-3 person-months
- Customization and Reporting: 1-2 person-months

6. Documentation (Priority: Medium)
- User Manuals and Technical Guides: 2-3 person-months

7. Testing and Evaluation (Priority: High)
- Test Case Design: 3-4 person-months
- Testing and Reporting: 2-3 person-months

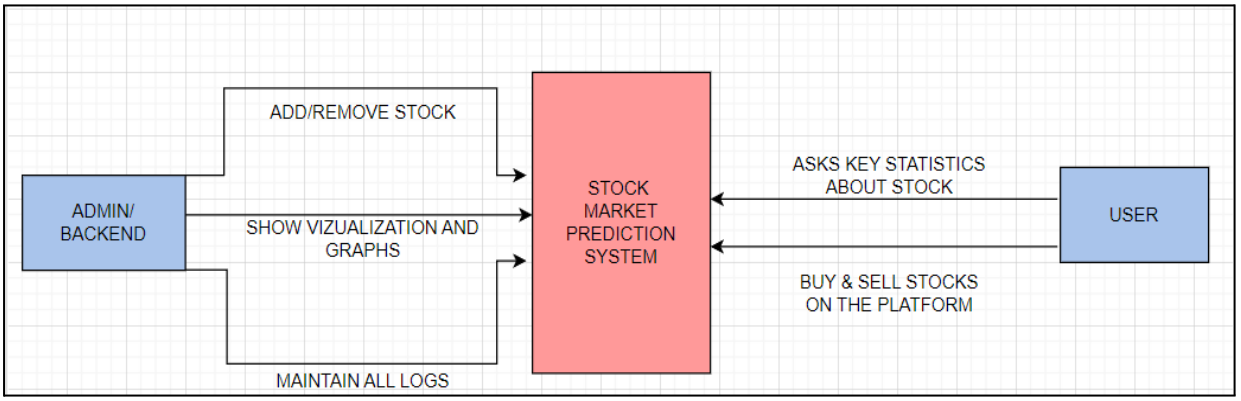
8. Deployment and Maintenance (Priority: High)
- Deployment Process: 1-2 person-months
- Maintenance and Updates: Ongoing effort

6. Gantt Chart

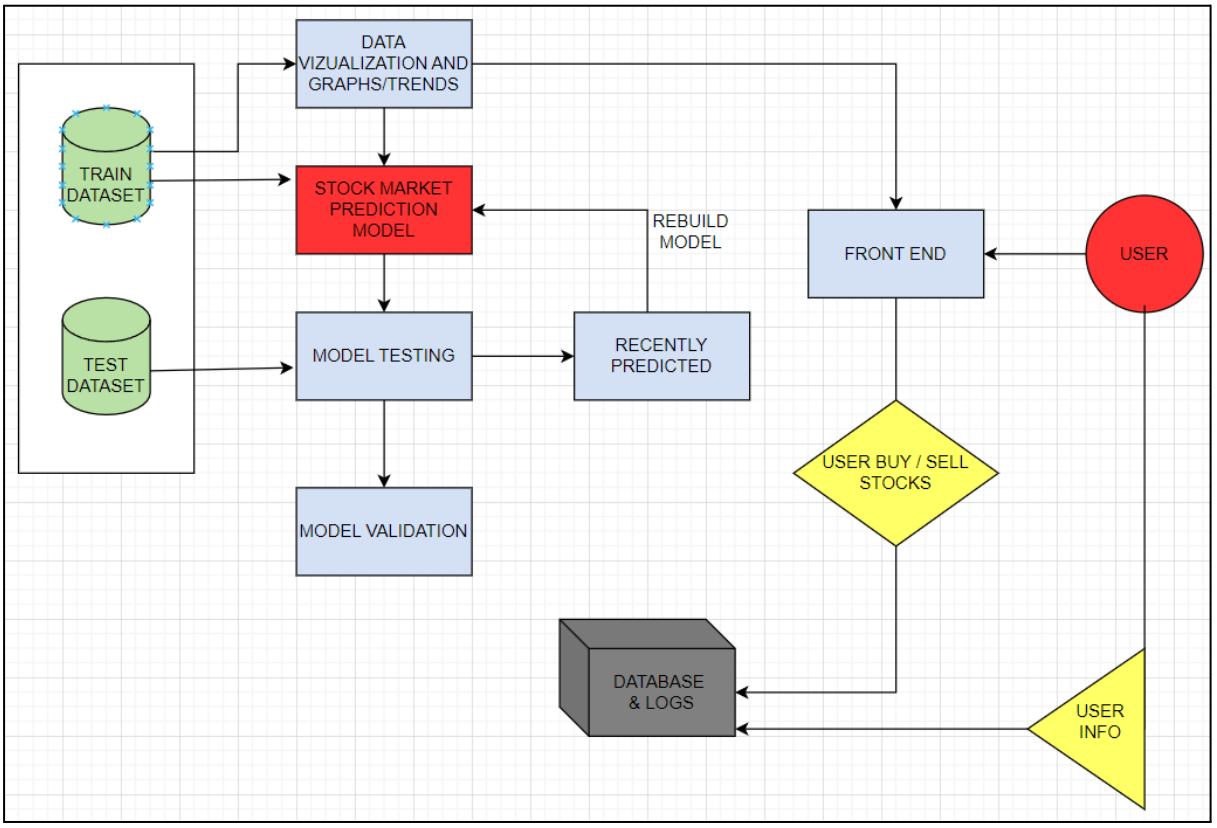


4.DESIGN DIAGRAMS

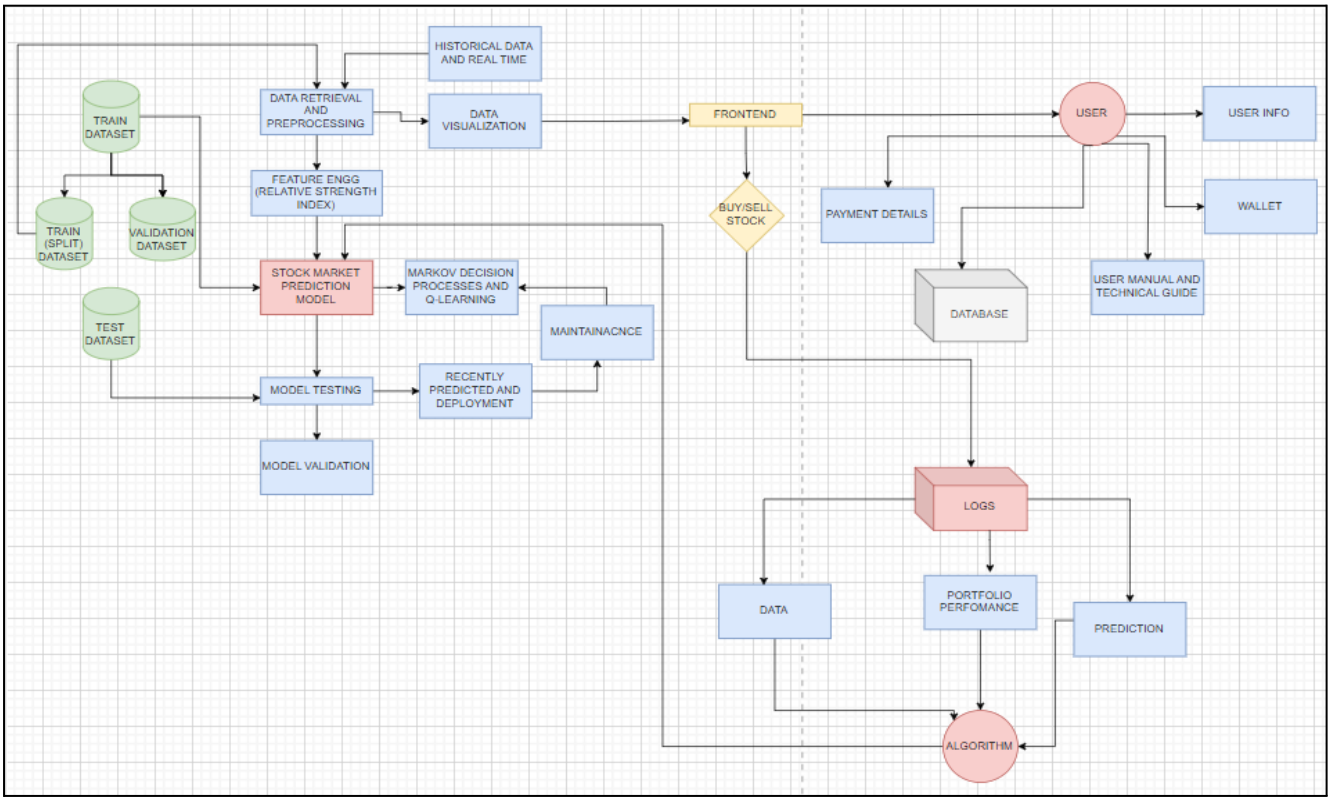
LEVEL 0 DFD DIAGRAM:



LEVEL 1 DFD DIAGRAM:



LEVEL 2 DFD DIAGRAM:



5. TEST CASES

Test Case ID	Name of Module	Test Case Description	Pre -conditions	Test Steps	Test Data	Expected Results	Actual Result	Test Result
UT-01	Training Reinforcement Learning model	Verify successful training of the RL model on historical stock data	Historical stock data is available	1. Open the RL model training module. 2. Provide historical stock data and model training parameters. 3. Initiate the training process.	Sample historical stock data, specified model training parameters	The system processes the data, trains the RL model without errors, and provides a trained model or a success notification.	Exactly as expected results	PASS
UT-02	Predicting stock prices	Verify accurate prediction of stock prices using the trained RL model	RL model is trained successfully, and current market data is available	1. Open the stock price prediction module. 2. Select a specific stock. 3. Provide current market data. 4. Initiate the prediction process.	Trained RL model, current market data for the selected stock	The system processes the data, predicts the stock price, and provides a confidence level. Prediction aligns with expected market trends and historical accuracy.	Exactly as expected results	PASS
UT-03	User Interface for prediction visualisation	Verify the functionality of the prediction visualization user interface.	Prediction module is accessible, and predictions are available.	1. Navigate to the prediction visualization section. 2. Select a specific stock. 3. Initiate the prediction visualisation process.	Selected stock for prediction	The system displays the predicted stock prices along with confidence intervals. The visual aids (graphs or charts) enhance prediction understanding.	Exactly as expected results	PASS

UT-04	Algorithmic Trading Integration	This test case verifies the integration of the system with algorithmic trading platforms and the execution of trading orders based on system predictions.	1.The system should be properly configured and deployed. 2.Algorithmic trading platforms are accessible & functional.	1. Log in to the system. 2. Navigate to the "Algorithmic Trading" section. 3. Monitor the system for the generation of buy/sell signals.	1.User credentials for logging in. 2.Selected stock for algorithmic trading. 3.Trading parameters (e.g., buy/sell threshold, stop-loss, take-profit).	1. The system successfully logs in 2. The "Algorithmic Trading" section is available and functional. 3. The system allows the configuration of trading parameters. 4. Automated trading is enabled without errors. 5. The system generates buy/sell signals based on predictions without errors.	Exactly as expected results	PASS
UT-05	Reporting Analytics	Verify that the reporting analytics module generates accurate and meaningful reports based on the stock market prediction data.	1. Sufficient historical stock market data is available for analysis. 2.Prediction models have been trained and applied to generate prediction results.	1.Launch the stock market prediction system. 2.Navigate to the "Reporting Analytics" module. 3.Select the desired time frame for the report (e.g., daily, weekly, monthly). 4.Choose specific stocks or indices for analysis. 5.Trigger the report generation process	Set time frame	The system should generate a comprehensive report.	Exactly as expected results	PASS

UT-06	Documentation	Verify the accessibility and content of system documentation.	Verify the accessibility and content of system documentation.	1. Access the documentation section. 2. Select the user manual. 3. Navigate through different sections of the documentation.	User manual document	The system displays user manuals and technical guides in a user-friendly format. The documentation covers all aspects of using and maintaining the system effectively.	Exactly as expected results	PASS
UT-07	Testing and evaluation	Verify the test cases and evaluate all the parameters used in the code and check if the project follows the project plan and meets the customer needs.	The code is working with everything as per the project plan and customer needs	1.meticulouslyExecute all codes and test them with different datasets. 2.If all requirements are met, cross-reference with the project plan.	Data Set: 1.Historical stock market data for the specified test period. 2.Simulated data for scenario testing.	1.Successful acquisition and integration of historical data. Model trained and updated without errors. Configuration Testing: 2.Proper configuration of market indicators. Real-time synchronization without delays or errors. Prediction Accuracy: 3.Accurate predictions aligned with actual market outcomes. Acceptable margin of error based on	Exactly as expected results	PASS

						historical performance. Scenario Testing: 4.System responds appropriately to different market scenarios. Predictions exhibit consistency and reliability. Model Update Testing: 5.Seamless integration of new data. Updated model improves accuracy or maintains performance.		
UT-08	Deployment	Ensures that the system can be deployed efficiently without issues	The system and all its components are ready for deployment.	1.Initiate the deployment process. 2.Monitor the deployment process for any errors or issues. 3.Verify that the system is operational and accessible after deployment 4.Confirm that user data and settings are preserved during deployment	1.Deployment tools and mechanism 2.System components and user data.	1.The deployment process is initiated without errors. 2.All system components are successfully deployed. 3.The system is operational and accessible after deployment. 4.User data and settings are preserved during deployment.	Exactly as expected results	PASS
UT-09	Maintenance	verifies the maintenanc	Maintenance tools and	1.Initiate a system update or	1.Maintenance	1.The update /maintenance	Exactly as expected	PASS

		e and update processes of the system	mechanisms are in place.	maintenance process. 2.Monitor the update/maintenance process for any errors or issues 3.Confirm that the update/maintenance does not cause data loss or corruption. 4.Validate that the system is updated to the latest version. 5.Ensure minimal downtime during the update/maintenance process.	tools and mechanisms. 2.System components and user data.	process is initiated without errors. 2.The system remains operational during it 3.No data loss or corruption 4.The system is updated to the latest version. 5.Downtime is minimal and within an acceptable range.	results	

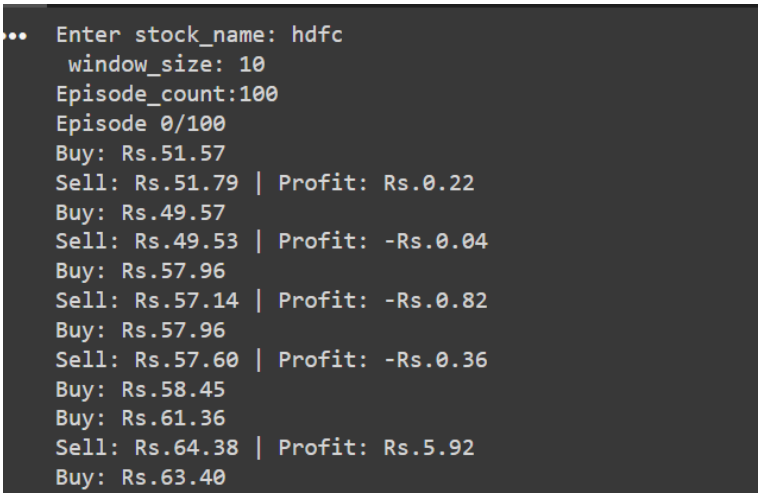
6. CODE AND OUTPUT SCREENSHOTS

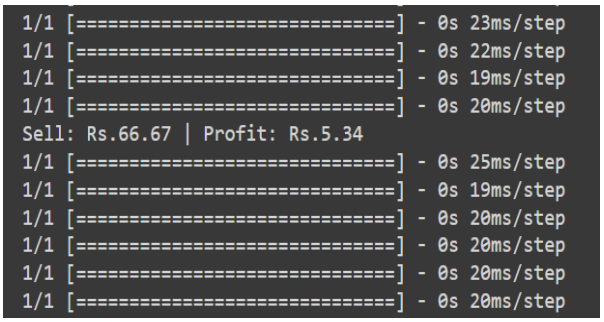
Please find the github link for the code files :

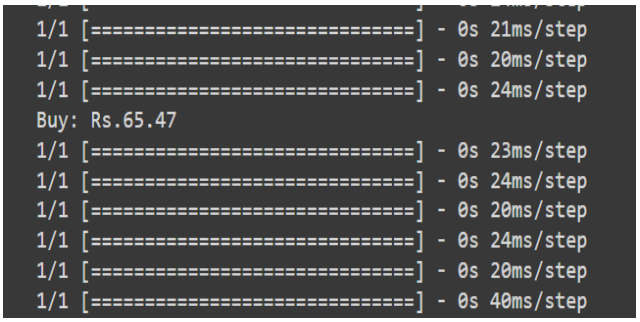
[Stock-price-prediction-(CODE)]

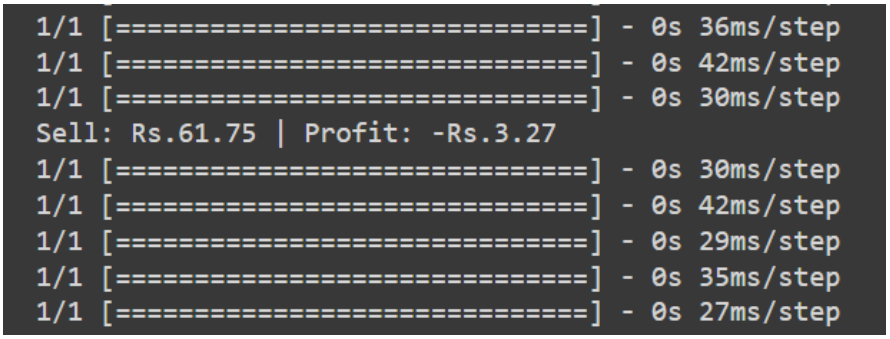
OUTPUT SS:

Training output screenshots:-









```
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 25ms/step
--------------------------------
Total Profit: Rs.38.34
--------------------------------
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
```

Evaluation output screenshot:-