

# **CHAPTER 1: INTRODUCTION**

The financial market is influenced by a combination of historical price patterns, macro events, and public sentiment. Retail investors and students often lack an integrated platform that combines price forecasting, news-based sentiment context, and a safe sandbox to test portfolio decisions. This project implements a Stock Price Prediction and Management system that integrates forecasting and sentiment analysis within a simulated trading and portfolio management workflow.

## **1.1 Background**

Stock forecasting models attempt to learn relationships from historical price series, while sentiment analysis estimates how recent news or social discussions may influence market expectations. When combined, these signals can provide a more informative decision-support view than either source alone.

## **1.2 Problem Definition**

Most stock-prediction demonstrations provide either:

- Forecasting models without user-facing workflows, or
- Trading simulators without integrated analytics.

This project bridges this gap by offering a single platform that fetches market data, generates short-horizon forecasts, computes model error, extracts news sentiment, and provides an interpretable recommendation in a simulated trading environment.

## **1.3 Project Overview**

The Stock Price Prediction and Management system provides a unified web application that supports:

- short-horizon stock forecasting using Linear Regression,
- recent financial-news sentiment analysis,
- simulated trading and portfolio tracking and an admin monitoring dashboard.

## **1.4 Scope**

### **1.4.1 In Scope**

- Full-stack Flask web application
- Role-based authentication (user / admin)
- Market data retrieval (primary: yfinance; fallback: Alpha Vantage)
- Stock price forecasting using Linear Regression (7-day horizon)
- Multi-source news sentiment engine with fallbacks
- Portfolio tracking: holdings, transactions, dividends, wallet, broker commission
- Admin dashboard: brokers, transactions, users, companies, summary statistics

### **1.4.2 Out of Scope**

- Real brokerage integration / real-money trading
- Advanced MLOps (scheduled retraining, model registry, experiment tracking)
- Production hardening (distributed caches, job queues, multi-tenant scaling)

## 1.5 Technology Stack

- Backend: Python, Flask
- Database/ORM: SQLite, SQLAlchemy
- ML & Analytics: scikit-learn (Linear Regression), pandas, numpy
- Sentiment / NLP: NLTK VADER, TextBlob, BeautifulSoup, requests, newspaper3k, aiohttp, tenacity
- Frontend: HTML/CSS, Bootstrap, JavaScript, D3.js

## **CHAPTER 2: LITERATURE REVIEW**

### **2.1 Stock Market Forecasting**

Stock market forecasting aims to estimate future prices or trends using historical market data and auxiliary signals such as financial news sentiment. Linear regression models have gained significant attention in recent research due to their simplicity, interpretability, and effectiveness for short-term price prediction. Lavanya and Gnanasskaran [1] demonstrate the application of linear regression for stock exchange price prediction, highlighting its computational efficiency and practical applicability in real-time systems. Their work emphasizes that linear models can provide reliable forecasts when properly configured with appropriate feature engineering.

### **2.2 Linear Regression for Stock Price Prediction**

Linear regression has proven to be an effective approach for financial forecasting, particularly for short-horizon predictions. Sangeetha and Alfia [2] present an evaluated linear regression-based machine learning technique for financial stock market forecasting, demonstrating that linear models can achieve competitive accuracy while maintaining low computational overhead. This makes them particularly suitable for web-based applications where response time is critical. Li [3] further validates the effectiveness of linear regression for stock price prediction through significance analysis, showing that properly selected features can lead to robust predictive performance. The interpretability of linear regression coefficients also provides valuable insights into which factors most influence price movements.

### **2.3 Multiple Linear Regression and Feature Selection**

Recent studies have explored multiple linear regression approaches that incorporate various market indicators and technical features. Lin [4] investigates predicting stock returns using linear regression with multiple predictors, demonstrating that combining historical price patterns with volume data can

improve forecast accuracy. Hu [5] extends this work by developing a multiple linear regression model that integrates diverse market features, showing that feature selection and preprocessing significantly impact model performance. These studies reinforce the value of linear regression as a practical and interpretable approach for stock price forecasting in production systems.

## 2.4 Gaps Identified

While existing research demonstrates the effectiveness of linear regression for stock price prediction [1], [2], [3], [4], [5], most studies focus solely on the forecasting component without addressing the broader context of portfolio management and decision support. A key gap addressed by this project is the end-to-end integration of:

- stock price forecasting using Linear Regression with proper feature engineering,
- multi-source sentiment analysis for contextual market interpretation,
- and a complete simulated portfolio management workflow with user and admin dashboards.

This integrated approach provides a practical platform for learning and experimentation that goes beyond isolated prediction models.

## **CHAPTER 3: OBJECTIVES**

### **3.1 Project Objectives**

- Stock Forecasting: Predict near-term prices and produce a 7-day forecast using Linear Regression.
- Sentiment Analysis: Fetch recent market news and compute polarity (positive/negative/neutral).
- Decision Support: Combine price-direction signal and sentiment polarity into a recommendation.
- Portfolio Simulation: Virtual wallet, holdings, buy/sell operations, commissions, dividends, transaction history.
- Administration: Admin monitoring dashboard and broker configuration.

### **3.2 Scope Summary**

The system focuses on prediction + sentiment + simulation for learning and experimentation. It is not designed for real-money execution.

## CHAPTER 4: METHODOLOGY

### 4.1 System Requirements

#### 4.1.1 Hardware Requirements (Minimum)

- Processor: Dual-core CPU or better
- RAM: 4 GB (8 GB recommended)
- Storage: 1 GB free space
- Internet: Required (market data + news sentiment)

#### 4.1.2 Software Requirements

- Operating System: Windows / Linux / macOS
- Python: 3.7+
- Libraries: as listed in requirements.txt

### 4.2 Data Collection and Preprocessing

#### 4.2.1 Market Data Acquisition

For a selected ticker symbol, the system:

- checks local {TICKER}.csv cache and reuses it if up-to-date,
- otherwise downloads ~2 years of historical price data using yfinance,
- falls back to Alpha Vantage if the primary source fails.

#### **4.2.2 Data Cleaning**

- Missing values are removed (dropna).
- Relevant fields such as Close are used as the primary signal.

#### **4.3 Linear Regression Forecasting Method**

The system implements a Linear Regression model following established methodologies in the literature [1], [2], [3]. The forecasting approach includes:

- A forecast horizon of 7 days is selected for short-term prediction.
- A target column is created using shift operation: Close after n days.
- Feature engineering extracts relevant price patterns from historical data.
- Standard scaling (StandardScaler) is applied to normalize features [2].
- Linear Regression model is trained on 80% of historical data.
- The model forecasts the next 7 days of closing prices.
- Model performance is evaluated using RMSE on a held-out test split (20% of data).
- A 4% adjustment factor is applied to predictions to account for market trends.

This approach balances prediction accuracy with computational efficiency, making it suitable for real-time web applications [1], [4].

#### **4.4 Sentiment Analysis Method**

- Recent headlines/news are gathered for the given ticker.
- Sentiment scores are computed and aggregated.
- Final outputs include polarity and distribution counts (positive/negative/neutral).

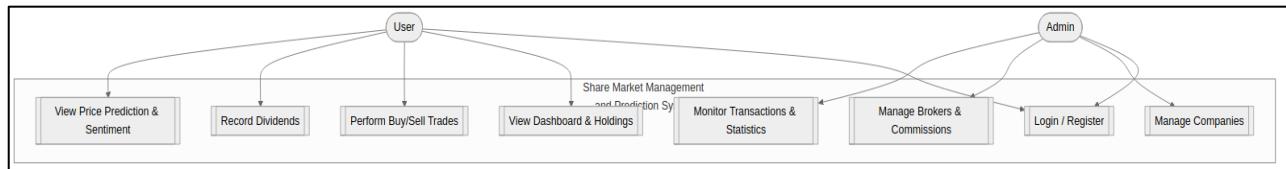
## 4.5 System Architecture and Design

### 4.5.1 Architectural Overview

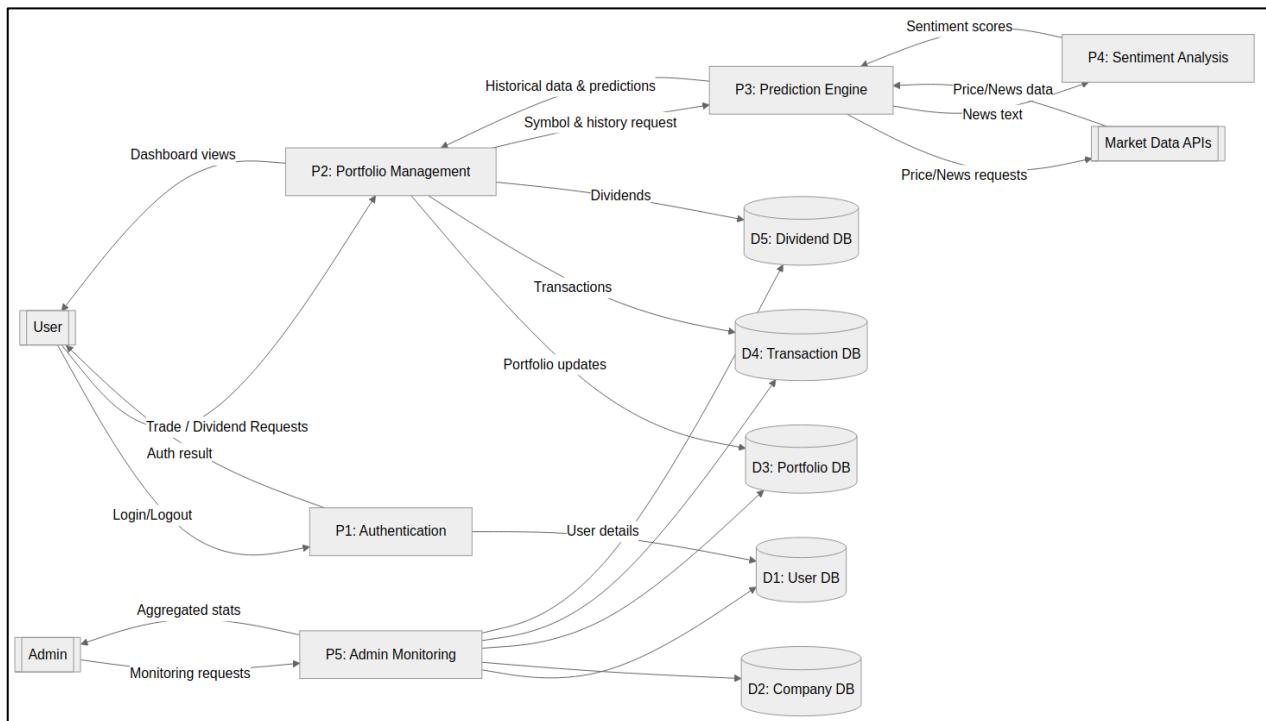
The application follows a three-tier architecture:

- Presentation Layer: HTML templates and dashboards
- Application Layer: Flask routes and business logic
- Data Layer: SQLite database via SQLAlchemy

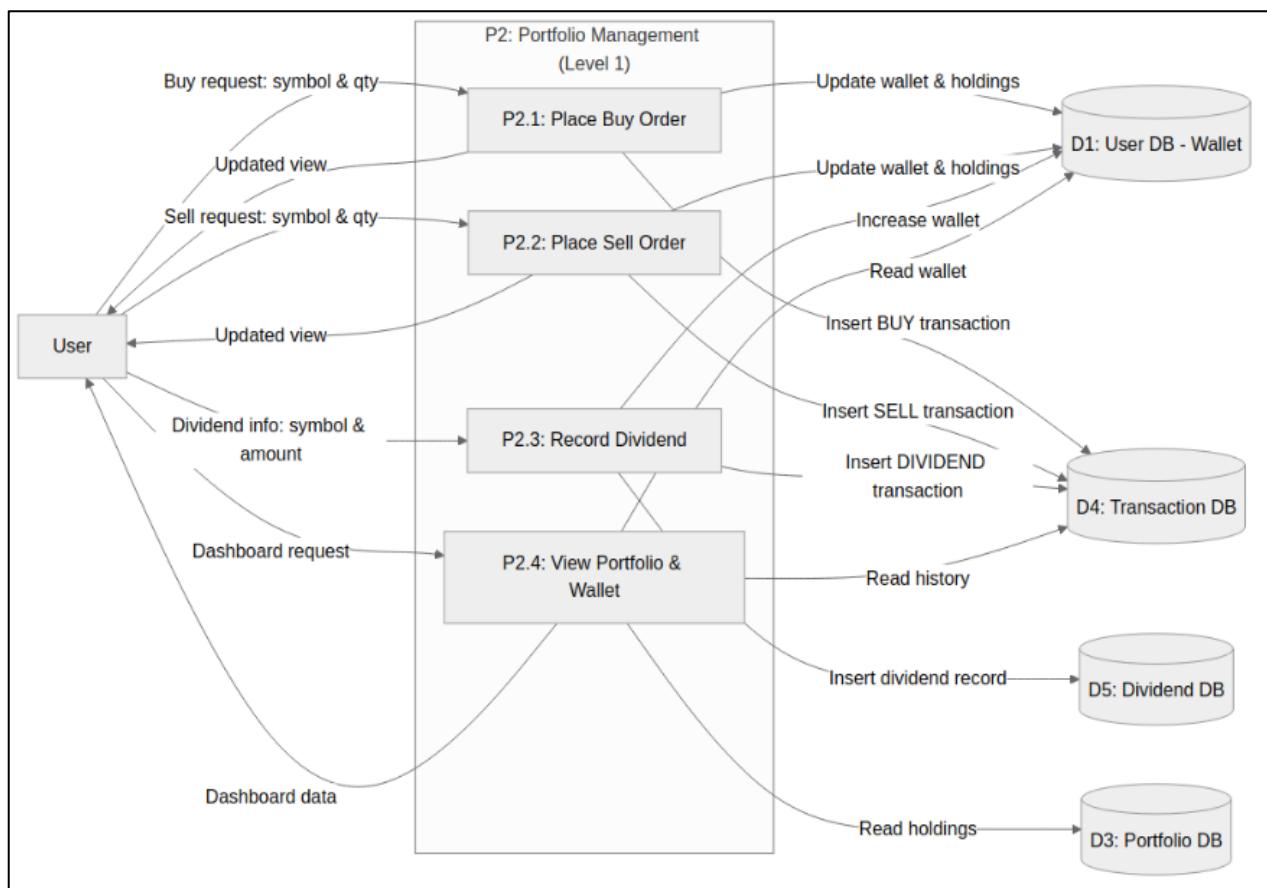
### 4.5.2 Use Case Diagram



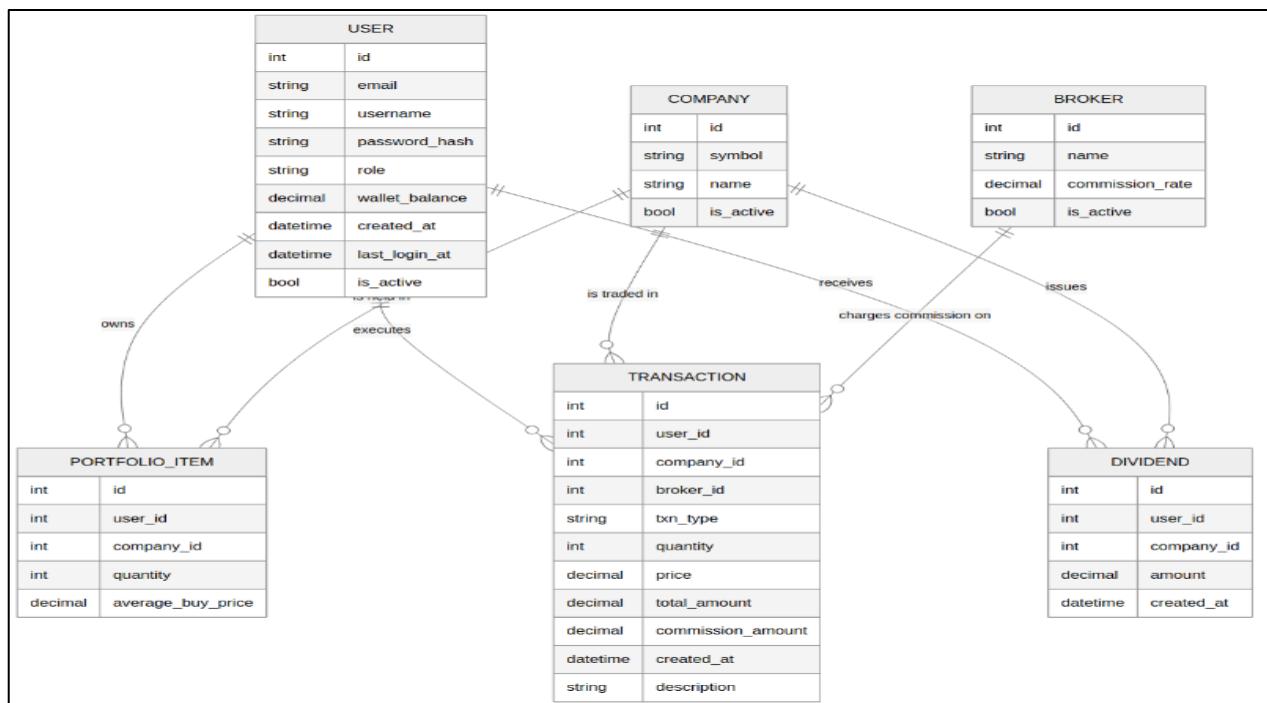
### 4.5.3 Data Flow Diagram (Level 0)



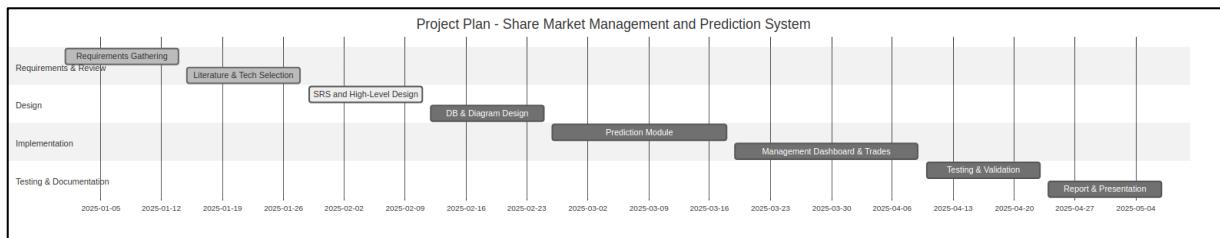
#### 4.5.4 Portfolio DFD (Level 1)



#### 4.5.5 Database Design (ER Diagram)



#### 4.5.6 Project Plan (Gantt Chart)



#### 4.6 Implementation Overview

- main.py: Flask app entry point (routes, DB models, prediction orchestration)
- news\_sentiment.py: sentiment analysis module
- templates folder: prediction results, user dashboard, admin dashboard
- static folder: styling and client-side scripts (including charts)

## CHAPTER 5: RESULTS AND DISCUSSIONS

### 5.1 Evaluation Metrics

RMSE (Root Mean Squared Error) is computed for the active forecasting model (Linear Regression) and displayed on the results page.

### 5.2 Output Screens

SAMS

HOME DASHBOARD

CALM, DATA-DRIVEN STOCK FORECASTS

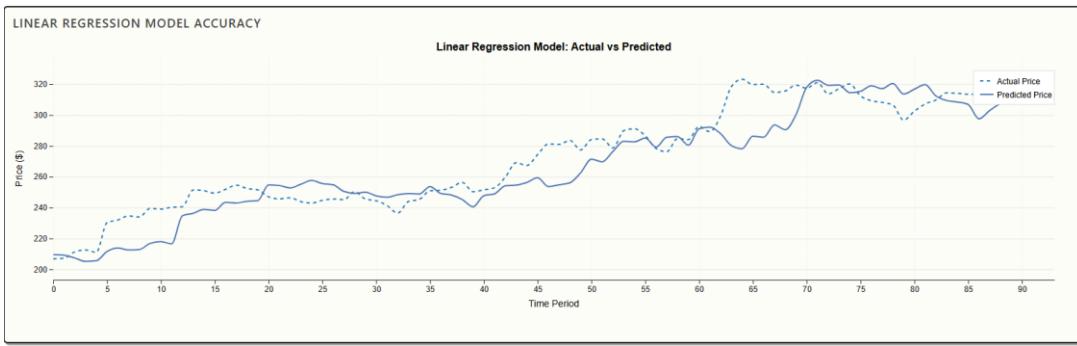
Enter a stock symbol below to generate tomorrow's price forecasts and a sentiment overview using ARIMA, LSTM, linear regression and recent news.

PLEASE ENTER A STOCK SYMBOL

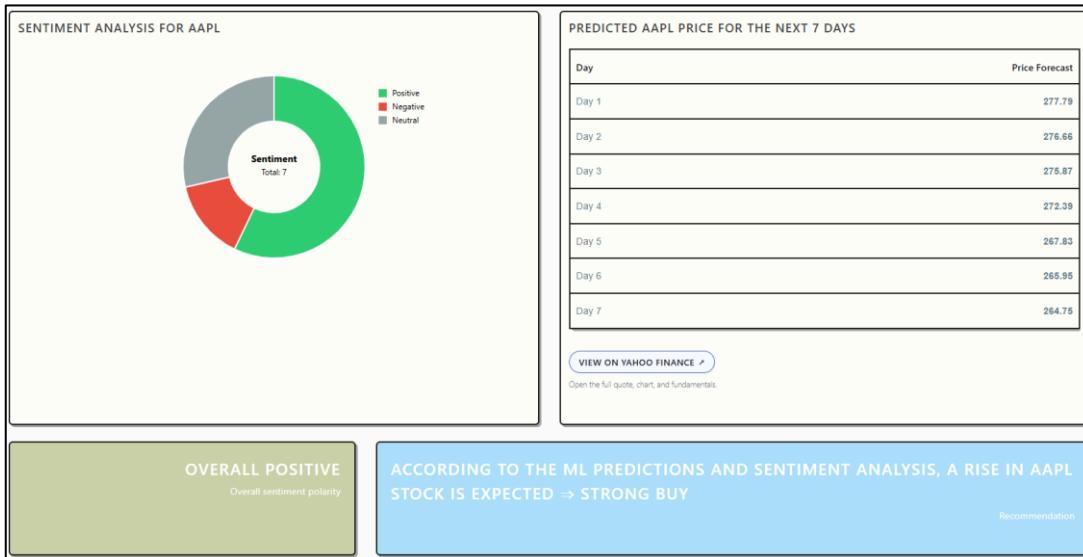
Company Stock Symbol (e.g. AAPL, GOOGL)

VIEW FORECAST

Home / Prediction Studio page



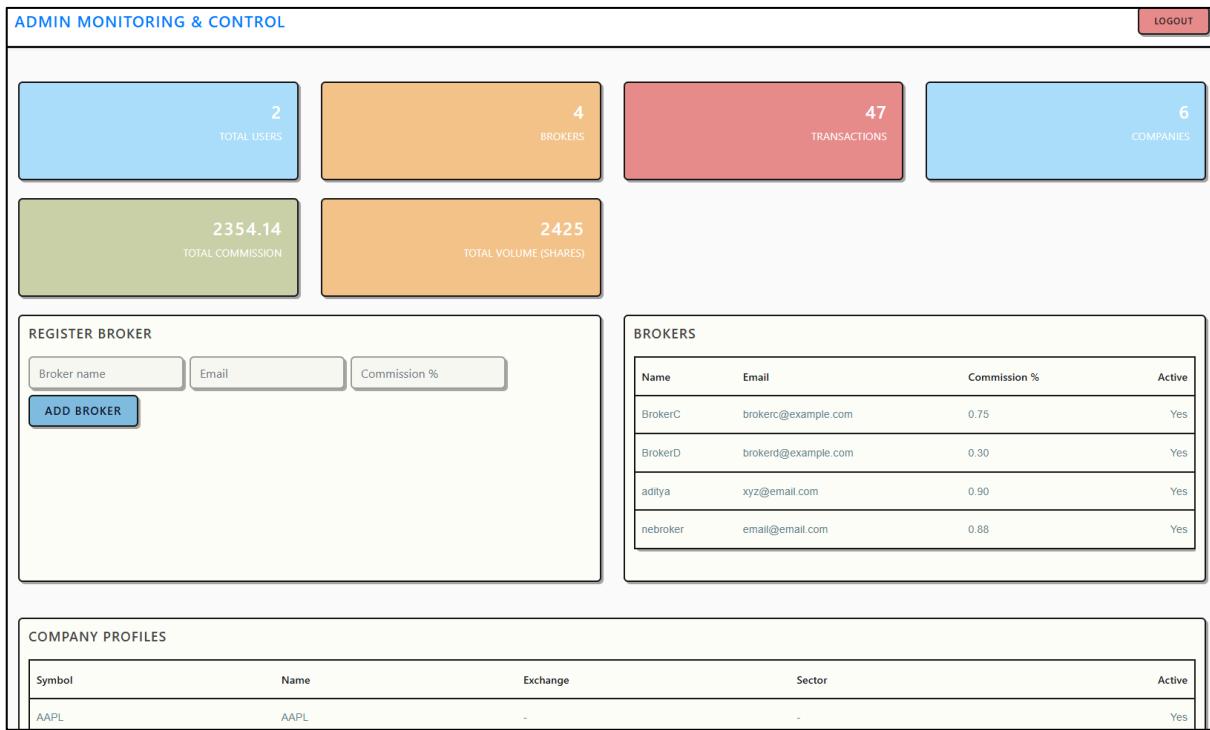
LR model test graph



7-day forecast table and Sentiment Chart



Stocks trading and dividends management



Admin dashboard and Broker management with Company management

### 5.3 Discussion

- The system provides a unified view combining technical forecasting and sentiment context, addressing the integration gap identified in the literature [1], [5].
- The Linear Regression model delivers fast response times suitable for web applications while maintaining interpretable predictions [2], [3].
- Feature scaling and proper train-test splitting ensure robust model performance as recommended by recent studies [2], [4].
- The 7-day forecast horizon provides actionable short-term predictions that align with practical trading strategies.
- The sentiment engine is designed with multiple fallback sources to remain operational even when primary data sources are unavailable.
- The combination of quantitative forecasting and qualitative sentiment analysis provides a more comprehensive decision-support framework than either approach alone.

## **CHAPTER 6: FUTURE WORK**

### **6.1 Conclusion**

This project delivers an end-to-end application for stock trend exploration in a risk-free simulated environment. It successfully integrates Linear Regression-based forecasting [1], [2], [3] with multi-source sentiment analysis and comprehensive portfolio management features. The system demonstrates that linear models can provide effective short-term predictions [4], [5] while maintaining the computational efficiency required for real-time web applications. By combining quantitative forecasting with qualitative sentiment signals, the platform addresses the integration gap identified in current research and provides a practical tool for learning and experimentation in stock market analysis.

### **6.2 Future Enhancements**

1. Notification System
2. Reporting System
3. Logging and Monitoring
4. Testing Implementation
5. Deployment Preparation
6. Final Integration and Testing
7. Documentation and Cleanup
8. Comprehensive Performance Testing
9. Comprehensive Security Testing
10. Usability Testing

11. Compatibility Testing
12. Accessibility Testing
13. Regression Testing Suite
14. Recovery and Resilience Testing
15. Acceptance Testing
16. Test Automation and CI/CD Integration
17. Testing Documentation and Knowledge Transfer
18. Enhanced Invoice Generation System
19. Volume-Based Prediction Modeling per instructions from the jury.
20. Administrative Dataset Management GUI per instructions from the jury.

## REFERENCES

- [1] M. Lavanya and P. Gnanasskaran, "Stock Exchange Price Prediction Using Linear Regression Model," in Proc. 5th Int. Conf. Inventive Res. Comput. Appl. (ICIRCA), Coimbatore, India, 2023, pp. 1-6. DOI: 10.1109/ICIRCA57980.2023.10220627.
- [2] J. M. Sangeetha and K. J. Alfia, "Financial stock market forecast using evaluated linear regression based machine learning technique," Meas.: Sensors, vol. 31, no. Supplement C, pp. 100950, Feb. 2024. DOI: 10.1016/j.measen.2023.100950.
- [3] S. Li, "Stock Price Prediction Based on Linear Regression and Significance Analysis," in Proc. 2nd Int. Conf. Data Sci. Eng. (ICDSE), 2025, pp. 596-603. DOI: 10.5220/0013702600004670.
- [4] H. Lin, "Predicting Stock Returns Using Linear Regression," Frontiers Bus., Econ. Manage., vol. 18, no. 3, pp. 139-141, 2025. DOI: 10.54097/06ck4c50.
- [5] R. Hu, "Stock Price Prediction Based on Multiple Linear Regression Model," in Proc. Int. Conf. Finance, Trade Bus. Manage. (FTBM), vol. 264, 2023, pp. 440-447. DOI: 10.2991/978-94-6463-298-9\_48.