International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI Based Investment Analysis

Guide - Prof. Sushmita Choure[1]
Ishika Avinash Jadhav[2]
Srushti Atul Bhanavse[3]
Rohan Shrinivas Dhangar[4]
Tanishka Kapil Deshmukh[5]
Janhavi Vitthal Ghadge[5]
Department of Artificial Intelligence and Machine Learning,
Zeal Polytechnic, Pune, Maharashtra, India

ABSTRACT

Investment decisions require the joint consideration of market behaviour, uncertainty, portfolio exposure, and an investor's tolerance for risk. Retail investors often use separate tools for price charts, indicators, risk calculations, and forecasts, making it difficult to review these factors together. This paper presents an implemented web-based AI-Based Investment Analyst that integrates a dashboard, stock analysis, risk calculation, forecasting, and a scenario-comparison feature called FIN-TWIN AI. The application uses React with Vite for the frontend and Python with FastAPI for the backend. The completed frontend and backend were tested together, and the project was observed to operate smoothly during the testing carried out. The paper describes the system architecture, main features, analytical approach, and functional testing. Since numerical forecasting metrics and detailed test measurements were not supplied, no specific prediction accuracy, investment return, or risk-reduction percentage is claimed. The application is intended for educational and analytical support; it does not guarantee future market performance or replace professional financial advice.

Index Terms—Artificial intelligence, investment analysis, stock market forecasting, portfolio risk, scenario analysis, FIN-TWIN AI, FastAPI, React, financial decision support.

scenario simulation, FastAPI, React, financial decision support.

I. INTRODUCTION

Financial markets produce large volumes of price and trading information, and it can be difficult for new investors to interpret this information consistently. A useful analytical application should present historical performance alongside measures of risk and clearly distinguish observed data from estimates. Portfolio theory provides a foundation for considering risk and return jointly [1]. However, historical returns and model forecasts are uncertain and do not guarantee future outcomes

Machine-learning techniques can identify patterns in time-series data, but financial data are noisy and market behaviour changes over time. Therefore, forecasts should be interpreted carefully and should not be treated as certain outcomes. This project brings key investment-analysis functions into one web application and includes functional testing of the integrated frontend and backend. The results reported in this paper are limited to the observed operation of the application; quantitative forecasting performance has not been established here.

The AI-Based Investment Analyst was developed as a student project at Zeal Polytechnic, Pune, under the guidance of Prof. Sushmita Choure. Its central features include a stock-analysis dashboard, risk calculation, forecasting, and FIN-TWIN AI, a scenario-comparison module. FIN-TWIN AI is designed to help users compare alternative investment assumptions through the application. The project is an educational decision-support tool and does not claim guaranteed investment outcomes.

II. PROBLEM STATEMENT

Individual users may find it difficult to bring stock-price history, quantitative risk measures, model estimates, and portfolio-allocation comparisons together in one accessible interface. The project investigates whether a small web-based decision-support prototype can combine these functions while making its assumptions and limitations visible.

III. AIM AND OBJECTIVES

The aim of this project is to develop and test a web-based investment-analysis application that presents stock information and provides analysis features to support risk-aware review of investment scenarios.

1. Develop an interactive dashboard for viewing stock information and analysis outputs.
2. Integrate the frontend and backend to support the application's stock-analysis functions.
3. Provide risk calculations to help users review risk-related information.
4. Implement a forecasting feature to present estimated stock-market trends.
5. Develop FIN-TWIN AI to compare hypothetical investment scenarios.
6. Test the integrated application and present its functions and limitations clearly.

IV. SCOPE OF THE PROJECT

The completed project includes a web dashboard, stock analysis, risk calculation, forecasting, and the FIN-TWIN AI scenario-comparison feature. The frontend is developed using React and Vite, while the backend is developed using Python and FastAPI. The frontend and backend were tested together. The application is intended for educational analysis and scenario comparison; it does not execute real stock-market trades, guarantee profitability, or provide personalised financial advice.

V. REALTED WORK

Markowitz's portfolio-selection framework formalized the trade-off between expected return and variance and remains a foundational reference for portfolio analysis [1]. In a student prototype, this motivates presenting risk alongside return rather than ranking assets by price movement alone. The practical quality of any portfolio comparison will still depend on input data, assumptions, and estimation choices.

Fischer and Krauss studied Long Short-Term Memory (LSTM) networks for financial-market prediction and compared their approach with other machine-learning methods [2]. Their work provides relevant background for forecasting in financial applications. However, results from one dataset or period cannot automatically be applied to other assets or market conditions. The forecasting feature in this project is presented as an application function; its quantitative performance would require separate measurement using a documented dataset, forecast horizon, and evaluation metrics.

Lundberg and Lee introduced SHAP, a framework for attributing a model prediction to input features [3]. This work provides background on explainable machine learning. The present project description does not claim that SHAP was implemented unless it is confirmed as part of the completed application. Explanations from a model should not be interpreted as proof of causality or guaranteed correctness.

The completed project brings together stock analysis, risk calculation, forecasting, and scenario comparison in a single educational web application. Its contribution is the integration of these functions into a working frontend and backend. Further quantitative evaluation would be required to compare forecast performance with baseline methods or to establish benefits over other investment-analysis tools.

VI. SYSTEM ARCHITECTURE

FIG. 1. SYSTEM ARCHITECTURE FLOW

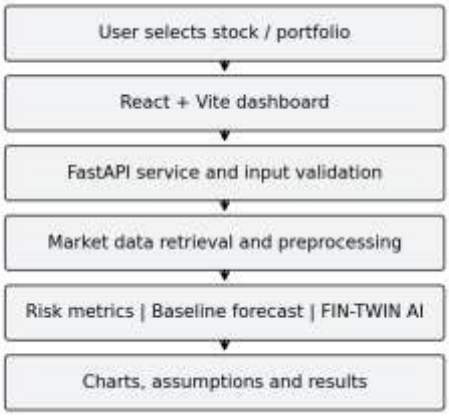


TABLE I. TECHNOLOGY STACK

Component	Technology	Implemented role
Frontend	React, Vite	User interface and dashboard
Backend	Python, FastAPI	Backend services and API communication
Analysis features	Project implementation	Stock analysis, risk calculation and forecasting
Scenario comparison	FIN-TWIN AI	Comparison of hypothetical

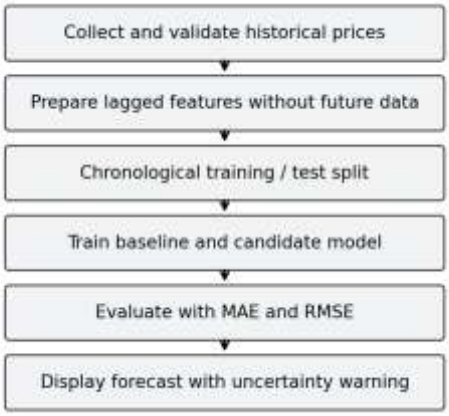
		investment assumptions
Market data	Provider to be selected	Historical stock-price records

A. The application follows a client-server architecture with the following main components:
B. Frontend: React and Vite are used to build the user interface and dashboard through which users access stock-analysis information and application features.
C. Backend API: Python and FastAPI provide the backend services and handle communication between the frontend and analysis functions.
D. Analysis functions: The application includes stock analysis, risk calculation, and forecasting functions. The exact algorithms and supporting libraries should be documented according to the final source code.
E. FIN-TWIN AI: The scenario-comparison feature presents alternative hypothetical investment assumptions to support comparison. Its outputs should be understood as estimates or simulations, not guaranteed future results.

The completed application includes the React/Vite frontend and Python/FastAPI backend. The two parts were tested together, and the integrated project operated smoothly during the testing carried out. The implementation therefore moves beyond the initial interface and backend setup described in the earlier draft.

VII. IMPLEMENTATION METHODOLOGY

FIG. 2. PROPOSED FORECASTING AND EVALUATION WORKFLOW



A. Data Collection and Validation

The application uses stock-related information as the input for its analysis features. Data used by the application should be checked for missing, duplicate, invalid, or inconsistent records wherever applicable. The specific data provider, stock symbols, date range, and retrieval method should be stated in the final manuscript based on the actual implementation.

B. Feature Preparation

The application processes stock information for its analysis functions. Where daily returns or other derived values are calculated, the formulas and input data should be consistent. The exact features and preprocessing steps should be described in accordance with the implemented code; no unverified data-processing method is assumed in this paper.

C. Forecasting and Evaluation

The application includes a forecasting feature that produces estimated market-related outputs. The forecasting function was included in the completed application and was exercised during project testing. However, the model name, training dataset, forecast horizon, train-test split, and numerical evaluation results were not provided for this revision. Therefore, this paper does not report MAE, RMSE, prediction accuracy, or a comparison with a baseline model. These measurements should be added only after they are calculated from the actual implementation.

D. Risk Measures

TABLE II. RISK AND PERFORMANCE MEASURES

Measure	**Definition / role**
Daily return	Price change relative to previous close
Historical volatility	Dispersion of daily returns; annualization to be documented
Maximum drawdown	Largest peak-to-trough decline in cumulative wealth
Portfolio return	Weighted sum of constituent-asset returns

The risk-calculation feature is part of the completed application. Risk-related outputs are intended to help users review stock or portfolio behaviour. The exact formulas, assumptions, and calculation period should be

documented from the source code or project report before publication. Any historical risk measure describes the selected data period and does not predict future losses.

E. FIN-TWIN AI Scenario Comparison

FIN-TWIN AI is the project's scenario-comparison feature. It allows users to review alternative hypothetical investment assumptions through the application. The output should be interpreted as an assumption-based comparison rather than a promise of future returns. The feature is intended to support understanding and comparison, not to make investment decisions on behalf of users.

F. User Interface and Safety

The dashboard presents the application's analysis features to users. Stock information, calculated risk values, forecasts, and hypothetical scenario outputs should be distinguished clearly wherever they appear. Users should be informed that forecasts are uncertain and that the application is an educational decision-support tool, not the sole basis for real investment decisions.

VIII. TESTING AND RESULTS

TABLE III. FUNCTIONAL TESTING SUMMARY

Test area	Testing carried out / result
Frontend-backend integration	Frontend and backend were run and tested together
Main application functions	Dashboard, stock analysis, risk calculation, forecasting and FIN-TWIN AI were exercised
Overall operation	The project team observed smooth operation during testing
Quantitative performance	No numerical response-time or forecasting metrics supplied
Scenario	Portfolio weights and response to changed assumptions

The completed project was tested by running the frontend and backend together. The integrated application was exercised to check that the main project functions could be accessed and used through the frontend-backend workflow. During the testing carried out by the project team, the application operated smoothly. This provides a qualitative functional observation of the integrated system.

The testing feedback available for this manuscript is qualitative. No test-case count, response-time measurement, error-rate measurement, dataset size, forecast accuracy, MAE, or RMSE value was supplied. Accordingly, the paper does not claim a numerical performance score or proven improvement in investment outcomes. A more detailed evaluation can report individual test cases, expected and actual outputs, screenshots, response times, and forecasting metrics if those measurements are available.

IX. CONTRIBUTION AND LIMITATIONS

The completed project provides an integrated educational web application with a dashboard, stock analysis, risk calculation, forecasting, and FIN-TWIN AI scenario comparison. It combines a React/Vite frontend with a Python/FastAPI backend and was tested as an integrated application. The observed result was smooth operation during the project team's testing. This is a functional outcome and should not be confused with evidence of forecasting accuracy or investment profitability.

The application is subject to the uncertainty of financial markets and the quality and coverage of the data used. Forecast outputs may vary with the selected asset, time period, and model assumptions. The current manuscript does not include numerical forecasting validation or measured investment returns. Any use of the application should therefore be limited to educational analysis. Future evaluation should document the data source, selected stocks, model details, test period, numerical risk calculations, and forecast metrics.

X. CONCLUSION AND FUTURE WORK

This paper presents the completed AI-Based Investment Analyst, a web application developed with React and Vite on the frontend and Python with FastAPI on the backend. The implemented features include a dashboard, stock analysis, risk calculation, forecasting, and FIN-TWIN AI scenario comparison. The frontend and backend were tested together, and the application operated smoothly during the testing carried out. The present results are qualitative; numerical forecasting accuracy and investment-performance results are not claimed because corresponding measurements were not provided. Future work can strengthen the evaluation by documenting the data source, forecasting model, test dataset, evaluation period, and metrics such as MAE and

RMSE. Additional improvements may include expanded model comparison, explainability, and more extensive scenario testing.

ACKNOWLEDGEMENT

The authors thank Prof. Sushmita Choure and Zeal Education Society for their academic guidance and support.

REFERENCES

1. H. Markowitz, "Portfolio Selection," The Journal of Finance, vol. 7, no. 1, pp. 77–91, 1952. doi: 10.1111/j.1540-6261.1952.tb01525.x.

2. T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018. doi: 10.1016/j.ejor.2017.11.054.

3. S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems 30, 2017, pp. 4765–4774.