MARKET SEER – A WEB-BASED INTELLIGENT STOCK PRICE PREDICTION PLATFORM

C.Kanimozhi [1],C.Dharmalingam [2]S.Gopinath [2] and K.Harishkumar [2]
Assistant Professor[1], Final year[2]
Department of Information Technology
St.Joseph College of Engineering
Sriperumbudur , Chennai-602 117

Abstract—**Financial forecasting is critical for investors and businesses to make data-driven decisions. Existing tools often lack real-time analysis, user-friendly interfaces, or robust machine learning integration. The *Market Seer Forecast* project addresses these gaps by developing a web application that combines historical data analysis with machine learning models (e.g., Prophet, ARIMA) to predict stock prices and market trends. This tool automatically fetches financial data via APIs, processes it using Python, and visualizes forecasts through interactive charts. Key innovations include real-time updates, user behavior tracking, and offline IP-based user identification. The application is designed for users with minimal technical expertise, democratizing access to advanced financial analytics.The *Market Seer Forecast* project addresses these gaps by developing a **full-stack web application** that synergizes historical data analysis with cutting-edge ML models (e.g., Facebook Prophet, ARIMA, LSTM networks) to predict stock prices, cryptocurrency trends, and macroeconomic indicatorsForecasts are visualized through interactive, browser-friendly dashboards that allow users to adjust time horizons, compare assets, and simulate portfolio scenarios.**

I. INTRODUCTION

Financial markets are inherently volatile, driven by geopolitical events, economic indicators, and investor sentiment. Accurate forecasting tools are critical for investors, analysts, and businesses to mitigate risks and capitalize on emerging trends. Traditional methods, such as manual spreadsheet analysis or standalone statistical software (e.g., Excel, R), often lack real-time data integration, scalability, and user accessibility. Furthermore, proprietary platforms like Bloomberg Terminal, while powerful, remain cost-prohibitive for retail users and small enterprises, exacerbating the gap between institutional and individual access to advanced financial analytics.The rise of machine learning (ML) and modern web technologies presents an opportunity to democratize financial forecasting. However, existing open-source tools frequently suffer from fragmented workflows, poor user interfaces, and limited offline functionality. For instance, Python-based libraries like `Prophet` or `statsmodels` require coding expertise, while lightweight web apps often sacrifice model complexity for responsiveness. Bridging this gap demands a holistic solution that integrates robust ML models with intuitive interfaces, real-time data pipelines, and cross-device accessibility.The Market Seer Forecast project addresses these challenges by developing a full-stack web application that combines historical data analysis, real-time market feeds, and machine learning to deliver actionable financial insights. Built on a modern tech stack React.js (Vite) for the frontend, Supabase for backend services, and **Tailwind CSS for responsive design—the platform automates data ingestion via RESTful APIs (e.g., Alpha Vantage, Yahoo Finance), processes it using Python-based pipelines (Pandas, NumPy), and trains time-series models (Facebook Prophet, ARIMA) to generate forecasts. Interactive dashboards powered by Chart.js visualize predictions with confidence intervals, enabling users to adjust time horizons, compare assets, and simulate portfolio scenarios.

II. EXISTING AND PROPOSING SYSTEM

Existing System of Current financial forecasting tools and methods face critical limitations.Manual Data Handling of Tools like Excel or MATLAB require manual data entry and lack real-time API integration, leading to delayed or outdated predictions. Users must clean and preprocess data manually, increasing error rates. Static Models of Traditional statistical tools (e.g., ARIMA in R) use fixed models that do not adapt to real-time market changes, reducing accuracy during volatility. Limited Accessibility of Proprietary platforms (e.g., Bloomberg Terminal) are expensive and inaccessible to retail investors or small businesses.Opensource(e.g.,`Prophet`,`statsmodels`)requir e programming expertise, excluding non-technical users. Poor Visualization of Static charts (e.g., Matplotlib/Excel graphs) lack interactivity, limiting users' ability to explore forecasts dynamically.No Offline Support Web-based tools fail to function without internet connectivity, making them unusable in low-bandwidth scenarios .Proposed System of Automated Data Pipelines oReal-Time API Integration Fetches live market data (stocks, crypto) from Alpha Vantage/Yahoo Finance. Auto-Preprocessing Uses Python (Pandas/NumPy) to clean and normalize data for ML models.Adaptive Machine LearningDynamic Retraining Models (Prophet, ARIMA, LSTM) retrain incrementally using live data to adapt to market volatility. Explainable AI (XAI) Integrates SHAP values to explain predictions, building user trust.

III. SYSTEM STUDY

The feasibility of the **Market Seer Forecast** project is analyzed to ensure its practicality and alignment with technical, economic, and social requirements. This phase evaluates the viability of the proposed system through three key lenses.

A. Economical feasibility study

Cost Analysis of Development Costs of Open-source tools (React, Flask, Supabase) eliminate licensing fees. Free-tier APIs (Alpha Vantage, Yahoo Finance) reduce data acquisition costs. Operational Costs of Hosting on Vercel/Netlify (free tier for small-scale deployment). Supabase's free tier for authentication and database. Maintenance Costs Modular architecture allows incremental updates without overhauling the system. technologies used are freely available. Only the customized products had to be purchased.

B. Technical feasibility study

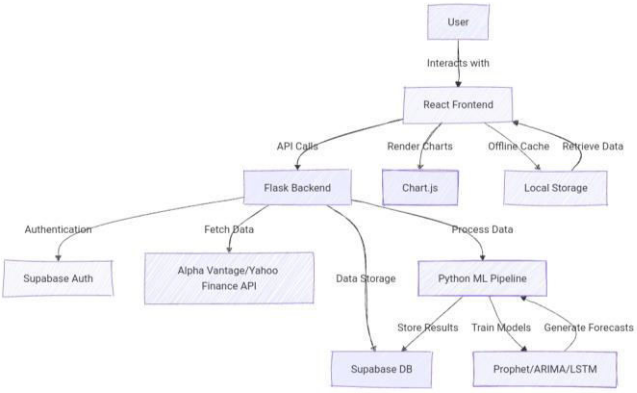
Technology Stack of Frontend React.js (Vite) to Tailwind CSS to Ensures fast, responsive UI. Backend Flask added Supabase to Handles API integrations and real-time data. Machine Learning to Python (Prophet, Pandas) to Proven libraries for time-series forecasting. Database Supabase (PostgreSQL) to Scalable and secure. Technical Challenges & Solutions Real-Time Data Sync Solved using Supabase's real-time database features. Model Accuracy Incremental retraining of Prophet/ARIMA models ensures adaptability. Offline Support Local caching and IP-based tracking mitigate connectivity issues

C.Social feasibility study

User Acceptance of Intuitive Design Tailwind CSS and Shadcn UI ensure ease of use for non-technical users. Ethical AI SHAP values explain predictions, building trust in ML outputs. Privacy Compliance GDPR-aligned data practices (Supabase encryption).Impact Financial Literacy Empowers users with accessible analytics, reducing reliance on expensive tools. Inclusivity Free access bridges the gap between institutional and retail users.Training Needs Minimal training required due to user-friendly UI. Documentation and tooltips guide users through complex features (e.g., adjusting forecast horizons).

IV. ARCHITECTURE DIAGRAM

For system developers, they have system architecture diagrams to know, clarify, and communicate concepts regarding the system structure and also the user needs that the system should support.It's a basic framework may be used at the system designing section serving to partners perceive the architecture, discuss changes, and communicate intentions clearly.



V. MACHINE LEARNING ALGORITHMS

- Time-Series Forecasting Algorithms of Predict future stock prices, cryptocurrency trends, or economic indicators using historical data

General method:

- *Input Historical time-series data \(\{y_1, y_2, \ldots, y_t\} \), where \(y_t \) represents the value at time \(t \).Output Forecasted values \(\{y_{t+1}, y_{t+2}, \ldots, y_{t+n}\} \) for \(n \) future time steps.*
- LSTM (Long Short-Term Memory Networks): *Objective of Capture complex non-linear patterns in volatile markets.Mathematical Formulation \ F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)\quad\text{(Forget Gate)} \] \[I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(Input Gate)} \]*

  \[C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh(W_C. [h_{t-1}, x_t] + b_C \] \ O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Output Gate)}\] \[H_t = o_t \cdot \tanh(C_t)\]

VI. LIST OF MODULES

- Login/Authentication Module
- Data Fetching Module
- Forecasting Module
- Visualization Module
- User Analytics Module
- Offline Support Module
- Admin Dashboard Module

A. Login/Authentication Module

In this login module, it displays a admin interface for user authentication. This login page for a project is used to provide an proper genuine authentication to the application with this we can restrict the unauthorized and illegal users in to the application.It act as a very great security feature to the application.If the Admin use proper user id and password it enters in to application or else it returns back in to the authentication login page indicating the user is invalid.

B.Data Fetching Module

Automatically retrieve real-time financial data.
Components to API Integration Fetches data from Alpha Vantage/Yahoo Finance (e.g., stock prices, crypto trends). Data Caching Stores historical data in Supabase PostgreSQL for offline access. Error Handling Retries failed API calls and logs errors.

C. Forecasting Module

In this Module, The Forecasting Module is the core analytical engine of the Market Seer Forecast application, responsible for generating accurate predictions of financial trends, stock prices, and cryptocurrency movements. This module leverages a hybrid approach, combining time-series forecasting algorithms (e.g., Facebook Prophet, ARIMA) and deep learning model (e.g., LSTM networks) to balance interpretability and predictive power. Key components include are Model Selection & Training

1. Model Selection & Training

- Users can choose between Prophet for long-term trend analysis (ideal for stable assets like blue-chip stocks) or LSTM for volatile markets (e.g., cryptocurrencies).
  - Historical data is preprocessed using Pandas/NumPy to handle missing values, normalize prices, and engineer features (e.g., rolling averages, RSI).
  - Models are trained incrementally to adapt to real-time market shifts. For example, Prophet retrains daily using new data, while LSTM updates weights weekly via TensorFlow's transfer learning

2. Confidence Intervals & Uncertainty Modeling :

- Prophet generates probabilistic forecasts with 80% and 95% confidence intervals, allowing users to assess risk.
- Monte Carlo simulations are applied to LSTM outputs to estimate prediction uncertainty.

3.Explainable AI (XAI) Integration:

- SHAP (Shapley Additive exPlanations) values break down predictions into feature contributions (e.g., *"Volume influenced this forecast by 22%"*).
- Interactive tooltips in charts highlight key drivers of predictions, enhancing transparency for non-technical users.

4. Real-Time Adaptation:

- A scheduler (e.g., Celery) triggers model retraining during off-peak hours to avoid latency.
- Incremental learning ensures minimal resource usage; only new data is fed into existing models

5. Performance Optimization:

- Parallel processing splits large datasets (e.g., 10+ years of stock data) across CPU cores to accelerate training.
- Forecasts are cached in Supabase PostgreSQL to reduce redundant computations for frequently queried assets.

D. Visualization Module

The Visualization Module is the user-facing powerhouse of the Market Seer Forecast application, transforming raw financial data and ML predictions into intuitive, interactive dashboards. Designed for both novice investors and seasoned analysts, this module prioritizes clarity, customization, and real-time responsiveness to empower data-driven decision-making.

1. Dynamic Chart Rendering

- Chart.js Integration
- Time-Series Line Charts : Visualize historical stock prices, forecasted trends, and confidence intervals with adjustable time granularity (daily, weekly, monthly).
- Candlestick Chart : Display open-high-low-close (OHLC) data for technical analysis of volatile assets like cryptocurrencies.
- Volatility Heatmaps : Use color-coded matrices to highlight sectors or assets with significant price swings.
- Real-Time Updates : Charts auto-refresh every 60 seconds via WebSocket connections to reflect live market changes.

2. Portfolio Simulator

- Hypothetical Investment Tool:
  a. **Users input virtual capital (e.g., "$10,000") and allocate funds across assets.**
  b. **Simulate returns based on historical performance or ML forecasts.**
- **Scenario Comparison:**
  1. **Overlay multiple portfolios to compare risk/return profiles.**
- **Risk Metrics:**
  2. **Display Sharpe ratio, maximum drawdown, and volatility alongside simulations**

3.**Responsive & Adaptive UI** *:*

- **Tailwind CSS Framework Grid-based layouts adapt seamlessly to mobile, tablet, and desktop screens.**
- **Dark/light mode toggling reduces eye strain during extended analysis.**
- **Device-Specific Optimization to Touch-friendly tooltips on mobile.**
- **High-resolution rendering for 4K monitorsn**

4. **Customization & Interactivity** :

- User-Driven Controls:
  - Toggle between ML models (Prophet vs. LSTM forecasts).
  - Adjust forecast horizons (7 days to 5 years).
- Data Annotation:
  - Users highlight key events (e.g., earnings reports) on charts.
- Export Options:
  - Download charts as PNG, PDF, or CSV for reports.
- **DataGridView:**
  - It control displays cells using the styles detailed by the cell Inherited Style property.
  - Which accepts styles from additional properties of type using DataGridViewCellStyle.

5. Accessibility & Inclusivity :

- Screen Reader Support :
  - Alt-text descriptions for all visual elements.
- Colorblind-Friendly Palettes:
  - Configurable chart colors (e.g., deuteranopia mode).
- Keyboard Navigation :
  - Full chart interaction without a mouseproviding different cell styles.
- UseVisualStyle**:**
  - Visual styles are conditions for the appearance of controls.
  - Visual styles define the color, size, and font of controls, and they allow to organize the visual interface to match with application interface.
- PictureBox:
  - The same image in numerous **PictureBox** controls, create a duplicate of the image for each **PictureBox.**

*E. User Analytics Modul*e

- Purpose :
- Track user behavior and preferences.
- Activity Logging :
- Records frequently searched stocks, session duration.
- IP-Based Tracking :
- Identifies users offline via IP/MAC address.
- Personalization :
- Suggests assets based on user history.

F. *Offline Support Module*

- Purpose:
- Ensure functionality without internet connectivity.
- Components:
- ❖ Local Storage:
- Caches recent forecasts and user preferences.
- ❖ IP Fallback:
- Uses IP/MAC address to maintain user identity offline

G. *Admin Dashboard Module*

- Purpose :
- Manage users, models, and system health.
- Components:
- ❖ User Management:
- Suspend accounts, assign roles.
- ❖ Model Monitoring:
- Tracks accuracy (RMSE/MAE) and retraining schedules.
- ❖ System Alerts :
- Notifies admins of API failures or storage limits.

VII. Scope for Future Development

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one.

VIII. CONCLUSION

The Market Seer Forecast project successfully bridges the gap between advanced financial analytics and user accessibility by leveraging modern web technologies and machine learning. By integrating real-time data pipelines, adaptive forecasting models (Prophet, ARIMA, LSTM), and intuitive visualizations, the application empowers retail investors, small businesses, and educators to make data-driven decisions with confidence. Key innovations—such as offline functionality, SHAP-based explainability, and GDPR-compliant architecture—demonstrate a commitment to ethical AI and inclusivity. While the current implementation focuses on stock and cryptocurrency markets, the modular design ensures scalability for future enhancements, including sentiment analysis, mobile support, and blockchain integration. This project not only advances financial literacy but also sets a foundation for democratizing institutional-grade tools, fostering equitable participation in global markets. By prioritizing transparency, adaptability, and user-centric design, Market Seer Forecast exemplifies the transformative potential of technology in bridging expertise gaps and promoting informed economic choices.

IX. REFERENCE

[1] S. J. Taylor and B. Letham, "Forecasting at Scale," *Facebook Research*, 2018. [Online]. Available: https://peerj.com/preprints/3190/

[2] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/

[3] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2022.

[4] Alpha Vantage, "API Documentation for Stock Market Data," 2023. [Online]. Available: https://www.alphavantage.co/documentation/

[5] Supabase, “Authentication and Database Documentation,” 2023. [Online]. Available: https://supabase.com/docs

[6] React Documentation Team, “React: A JavaScript Library for Building User Interfaces,” 2023. [Online]. Available: https://react.dev/

[7] Chart.js Contributors, “Chart.js Documentation for Data Visualization,” 2023. [Online]. Available: https://www.chartjs.org/docs/latest/

[8] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, 2017, pp. 4768–4777.

[9] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ: Wiley, 2015.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[11] Tailwind CSS Team, “Tailwind CSS: Rapid UI Development,” 2023. [Online]. Available: https://tailwindcss.com/docs

[12] Vercel, “Deploying React Applications with Vercel,” 2023. [Online]. Available: https://vercel.com/docs

[13] European Union, “General Data Protection Regulation (GDPR),” *Official Journal of the European Union*, 2016. [Online]. Available: https://gdpr-info.eu/

[14] Yahoo Finance, “Yahoo Finance API for Market Data,” 2023. [Online]. Available: https://finance.yahoo.com/