InvestIQ – Smart Investment Assistant

Ruchitha Reddy Kuthuru[1], Afrida Mehanaz Shaik[2], Sharan Jagini[3], Mahidhar Reddy Kandula[4],
Nadeem Hussain Shaik[5], Shrinidhi Daheechi[6]

Seidenberg School of Computer Science and Information Systems
Pace University, New York, NY, USA
[1]rk04332n@pace.edu [2]as36597n@pace.edu [3]sj56091n@pace.edu [4]mk64394n@pace.edu
[5]ns38406n@pace.edu [6]sd69205n@pace.edu

Abstract— **InvestIQ is a smart investment assistant designed to simplify financial decision-making through personalized recommendations. It combines real-time financial news analysis with historical stock data using machine learning techniques such as NLP and Time Series forecasting. By aligning recommendations with users' individual risk tolerance and budget, InvestIQ empowers users to make more informed and confident investment choices**

Keywords— **Investment, Personal Finance, Recommendation System, Natural Language Processing, Time Series Analysis, React, Python, Stock Market, API Integration**

I. Introduction

In today's dynamic economic landscape, financial markets are more volatile and data-rich than ever before. While access to market data has improved significantly with the advent of digital finance platforms, the abundance of information can overwhelm individual investors. Many struggle to interpret trends, understand market sentiment, and align decisions with personal financial goals. Consequently, users often make suboptimal investment choices or rely on generalized advice that may not suit their unique needs.

InvestIQ is a smart investment assistant developed to address these challenges by offering personalized and data-driven investment guidance. Unlike traditional platforms that merely present raw financial data or static charts, InvestIQ delivers tailored insights by considering individual user preferences such as risk tolerance, investment goals, and budget constraints.

The platform uses a combination of Natural Language Processing (NLP) to interpret real-time market sentiment from financial news and Time Series Analysis to understand historical stock performance. These capabilities are integrated into a rule-based recommendation engine that delivers actionable investment suggestions.

InvestIQ is developed using a modern and scalable technology stack. The frontend, built using React and styled with Tailwind CSS, ensures a responsive and user-friendly interface across devices. The backend, developed in Python using frameworks like Flask or FastAPI, supports API-based communication and integrates machine learning services. SQLite is used for lightweight yet efficient data storage, making the system ideal for rapid development and deployment.

By bridging the gap between data overload and actionable insights, InvestIQ empowers users to invest smarter and with greater confidence.

II. Literature Review

The development of recommendation systems has seen significant adoption in domains such as e-commerce and entertainment. In recent years, these systems have made their way into the financial sector, offering users personalized investment advice and trading strategies. As financial decision-making increasingly depends on timely data and behavioral insights, leveraging machine learning models has become a promising approach to improving investment outcomes.

One key area of innovation lies in **sentiment analysis**. Research in financial NLP shows that market sentiment derived from news articles, analyst opinions, and social media signals has a measurable impact on stock prices. By analyzing the tone and content of real-time news using sentiment classifiers, models can anticipate short-term market reactions. Tools like the **phi model** and **FinBERT** have been particularly effective in capturing nuances in financial language.

In parallel, **Time Series Forecasting** plays a critical role in understanding historical trends and predicting future movements in asset prices. Traditional statistical models like ARIMA have been surpassed by deep learning architectures such as **Long Short-Term Memory (LSTM)** networks and **Temporal Fusion Transformers (TFT)**. These models are capable of learning complex temporal dependencies and seasonality in financial data, making them suitable for forecasting stock prices and identifying trend reversals.

Several studies have validated the combination of sentiment and time series data in improving the quality of

recommendations. For example, merging short-term sentiment shifts with long-term price trends has been shown to yield more robust and adaptive financial models.

The design of InvestIQ draws heavily from these research findings. Its architecture integrates NLP techniques for real-time news sentiment extraction with Time Series models to analyze stock performance over recent weeks. The fusion of these two data streams allows InvestIQ to deliver nuanced, context-aware investment suggestions that are both timely and relevant.

By staying informed by the latest research and proven methodologies, InvestIQ aims to provide a state-of-the-art solution in the growing field of AI-driven financial advisory platforms.

III. PROJECT REQUIREMENTS

The InvestIQ platform is designed to provide a seamless and personalized investment experience, and its project requirements reflect a balance between user-centric functionality and robust technical architecture. One of the core functional requirements is **user authentication**, which allows individuals to securely register and log into the platform using their email and password. The authentication system ensures proper validation of inputs, including email format and password length, and provides informative error messages for failed login or registration attempts.

Once authenticated, users are encouraged to **set up their investment profile**. This involves submitting key personal preferences such as risk tolerance (categorized as Low, Medium, or High) and their available investment budget. This profile setup is handled through an intuitive form with validation mechanisms to catch invalid inputs (e.g., negative budgets) and provide confirmation once details are saved.

The dashboard, which is the central interface of InvestIQ, features multiple sections including **recommendations, historical trends, and market news**. Initially, mock recommendations are displayed as placeholders to help users visualize how the system will deliver insights. As the system evolves, these placeholders are replaced with dynamic, **personalized investment suggestions** based on both the user's profile and historical stock performance data. Each recommendation includes concise labels (e.g., "Stock: AAPL, Suggestion: Buy") and optional interaction buttons for future features such as deeper analysis or simulation.

Another major feature is the ability to **persist and update user data**, which involves storing profile information in a backend database and retrieving it when needed. This ensures that users can return to the platform and see previously configured preferences, maintaining continuity in their experience. On the backend, the application also **integrates external APIs**, such as Yahoo Finance for historical stock data and NewsAPI for up-to-date financial news. These integrations are supported by caching mechanisms and error handling to ensure reliable and efficient data retrieval.

Finally, the system is supported by a robust backend architecture with a structured database schema, including tables for users, profiles, and historical stock data. The backend follows REST principles and exposes various endpoints for frontend interaction, including user management and data fetching. These technical foundations allow InvestIQ to deliver a scalable, secure, and insightful user experience that bridges financial data with actionable recommendations.

IV. SYSTEM DESIGN

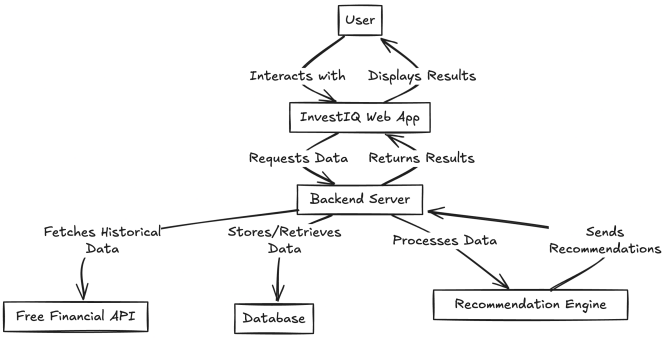

Fig 1. Conceptual Architecture Diagram

A. *Architecture*

InvestIQ follows a modular, decoupled architecture that separates the frontend user interface from backend business logic and machine learning services. This design promotes scalability, maintainability, and parallel development among teams. The application is built as a single-page application (SPA) on the front, which interacts with the backend via RESTful APIs. These APIs act as a communication bridge, handling everything from user authentication and data storage to machine learning recommendations and external data fetching. The architecture also includes integration with third-party APIs like Yahoo Finance and NewsAPI for real-time financial data. To further support the machine learning pipeline, a separate ML API service is hosted independently, ensuring a clear separation of responsibilities and allowing independent development and deployment of recommendation logic.

B. *Frontend*

The frontend of InvestIQ is developed using React, a modern JavaScript library known for building responsive, component-based user interfaces. It leverages Tailwind CSS for streamlined styling, ensuring consistency across the application and responsiveness across both desktop and mobile devices. The interface is organized into key views such as the Dashboard and Profile Setup pages. The dashboard serves as the main user interface, displaying investment recommendations, historical stock trends, and curated financial news articles. Routing is handled using React Router, allowing users to navigate between pages without full page reloads, enhancing

the user experience. Each section of the UI is designed with usability in mind, featuring form validation, feedback messages, and interactive components to simulate a real-world financial platform.

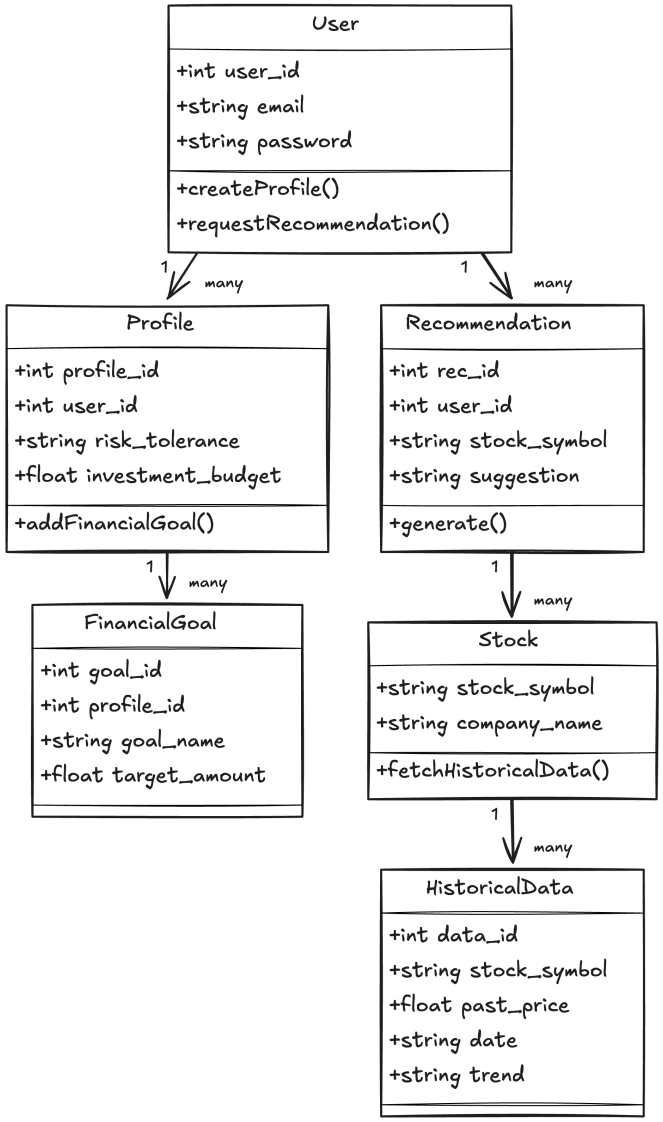

Fig 2. Class Diagram

C. *Backend*

The backend is developed in Python, utilizing either Flask or FastAPI for lightweight, high-performance RESTful API services. It serves as the backbone of the platform, handling operations such as user registration, login, profile data management, and the retrieval of financial data. The backend communicates with an SQLite database, which stores user profiles, authentication credentials, and historical stock data. The system also includes logic for caching data and managing API rate limits to ensure smooth interaction with external services such as Yahoo Finance for stock trends and NewsAPI for market news. Additionally, the backend includes error handling and logging mechanisms to ensure reliability. For recommendations, the backend interacts with a dedicated machine learning service, which processes user profiles and financial data to return personalized investment suggestions. This interaction is secured and abstracted behind internal endpoints, supporting a clean separation between core business logic and AI-driven functionalities.

D. *Backend API Details*

The backend of InvestIQ is structured around a RESTful API architecture, enabling modular and scalable communication between the frontend and various backend services. The API is organized into key functional domains: User Management, Profile Management, Data Integration, and Recommendation Handling. Each endpoint follows standard HTTP methods (GET, POST, PUT, DELETE) for resource manipulation, adhering to clean and predictable URI structures. For example, endpoints like /api/register, /api/login, and /api/profile handle user authentication and preferences, while others like /api/stocks and /api/news retrieve external financial data.

Security and data integrity are critical aspects of the backend. User data is sanitized and validated at multiple points, and sensitive information such as passwords is encrypted and stored securely. For performance and efficiency, the backend implements caching strategies when accessing third-party APIs, reducing latency and dependency on external services. Furthermore, error handling is robust, with structured error responses and retry mechanisms in place to gracefully handle API failures or missing data.

The backend also plays an intermediary role in processing and routing user inputs to the Machine Learning API. For instance, when a user requests recommendations, the backend extracts relevant user data, formats it appropriately, and sends it to the ML service, then forwards the response to the frontend in a user-friendly format. This separation of concerns allows the backend to remain lightweight and focused, while delegating computational tasks to specialized modules.

E. *Machine Learning API*

The Machine Learning (ML) API within InvestIQ serves as a dedicated microservice responsible for delivering advanced analytical capabilities that support investment decision-making. Developed using Python and Flask, this microservice is containerized using Docker and operates independently from the core backend. This architectural choice promotes modularity, simplifies deployment and scaling, and allows the ML models to be updated or replaced without interrupting the main application's flow or affecting other services. By isolating the computational and inference workloads, InvestIQ ensures high availability and efficient resource utilization, especially for compute-intensive operations like model inference.

The ML API exposes two principal endpoints, each tailored to a specific aspect of financial analysis: **stock price prediction** and **market sentiment analysis**. These endpoints support POST requests and are designed to be consumed by the backend service, which acts as a proxy for the frontend application. This intermediary design helps avoid Cross-

Origin Resource Sharing (CORS) issues and maintains secure and streamlined data flow throughout the platform.

The first endpoint, **`POST /api/predict/price`**, provides short-term stock price forecasting using Long Short-Term Memory (LSTM) models—an advanced form of recurrent neural networks well-suited for time series data. The API expects a JSON-formatted request containing recent historical stock data, including the open, high, low, close prices, and volume for a given stock symbol. Depending on the forecast horizon specified by the client (e.g., 1-day, 5-day, or 30-day), the API selects the appropriate pre-trained model. The input data is preprocessed and normalized using a Min-Max scaler, as configured in the `lstm_config.json` file. After processing, the model returns a predicted closing price or price trend for the specified horizon. These forecasts can help investors make more informed decisions based on anticipated short-term price movements.

The second endpoint, **`POST /api/analyze/sentiment`**, performs market sentiment analysis using a fine-tuned GPT-2 language model. This endpoint accepts financial news text or headlines as input and returns sentiment scores categorized as positive, neutral, or negative. In detailed mode, the response also includes extracted keywords, named entities, and contextual indicators such as economic terms, company mentions, or market actions. This level of analysis enables InvestIQ to contextualize news articles and headlines in real-time, providing users with sentiment-driven investment insights. Sentiment signals generated by this endpoint can be directly integrated into the investment recommendation engine to adjust risk levels or prioritize trending opportunities.

Integration between the ML microservice and the InvestIQ platform is facilitated through the backend API. Environment variables such as `ML_API_URL` are configured in the backend's `.env` file to route requests dynamically and securely. This integration strategy allows the frontend, built in Next.js, to remain agnostic of ML infrastructure details while still benefiting from its predictive power.

From a deployment perspective, the ML API is designed to be containerized and supports both CPU and GPU environments. The service can be launched using Docker, and GPU support is enabled for environments with CUDA-compatible hardware to accelerate model inference, particularly beneficial for larger models like GPT-2. The deployment directory includes all necessary scripts, configuration files, and model checkpoints, ensuring consistent and reproducible setups across development, staging, and production environments.

Looking ahead, the architecture supports the addition of more sophisticated ML capabilities, such as Transformer-based sequence prediction for long-term forecasting, deep reinforcement learning for dynamic portfolio optimization, or collaborative filtering for peer-driven investment suggestions. The ML API may also be exposed as a standalone SaaS (Software as a Service) product, allowing third-party platforms to leverage InvestIQ's AI capabilities through standardized API access.

V. MACHINE LEARNING

The machine learning component of InvestIQ serves as the core intelligence layer that empowers the platform's ability to deliver real-time, personalized investment insights. It acts as the analytical backbone for both stock recommendation generation and market sentiment interpretation. Designed with modularity in mind, the ML stack is composed of two key models: a Time Series Analysis model and a Natural Language Processing (NLP) model. These models are deployed as independent yet collaborative microservices, allowing for flexibility, scalability, and continuous evolution as market dynamics and model capabilities advance.

The Time Series Analysis model is built on state-of-the-art deep learning architectures such as Long Short-Term Memory (LSTM) networks and Temporal Fusion Transformers (TFT). These models are optimized to capture non-linear temporal dependencies, seasonal behavior, and irregular volatility patterns within stock price data. InvestIQ processes 30 days of historical stock data for each asset under consideration. During this process, technical indicators such as moving averages and trend vectors are calculated. The model identifies whether an asset is in a bullish (upward momentum), bearish (downward trend), or neutral state. This classification directly informs the rule-based logic behind the platform's investment recommendations, especially when paired with user-specific risk profiles and portfolio preferences.

Running in parallel, the Natural Language Processing (NLP) model augments the data-driven signals from the time series module by integrating real-time sentiment extracted from financial news articles. This module employs fine-tuned transformer models—such as GPT-2 derivatives or domain-specific phi models—trained on financial corpora to recognize the tone and context of market-relevant language. By analyzing headlines and article bodies, the NLP engine assigns sentiment scores ranging from positive to neutral to negative. Additionally, it performs entity recognition and term extraction to highlight key subjects such as company names, industries, or financial actions. This enables a deeper semantic understanding of how market narratives may influence price movement or investor sentiment.

The integration of these two models—temporal trend forecasting and contextual sentiment analysis—produces a more holistic investment recommendation system. By combining quantifiable market trends with qualitative sentiment cues, InvestIQ delivers insights that are not only

statistically grounded but also contextually aware of current events and market psychology. This dual-layered intelligence ensures that each recommendation takes into account both historical performance data and emerging market narratives.

From an operational perspective, both the Time Series and NLP modules are deployed as standalone, containerized APIs. This microservices-based architecture allows for independent model updates, version control, and horizontal scaling without disrupting the core application. Furthermore, the separation of these services enables specialized teams to enhance and experiment with new model versions, improving accuracy and responsiveness over time. This architecture also facilitates potential external integrations, where InvestIQ's ML capabilities could be offered as a licensed API to third-party fintech platforms or institutional partners.

In essence, the machine learning infrastructure in InvestIQ transforms static market data and textual information into actionable intelligence. Through its combined use of temporal pattern recognition and sentiment analysis, the platform provides investors with a dynamic, data-informed, and context-sensitive decision-making experience.

VI. Conclusion

InvestIQ is a smart investment assistant platform designed to address the challenges modern investors face in navigating a fast-paced, information-rich financial environment. Through its personalized and data-driven approach, the platform helps users make more informed investment decisions by combining historical trend analysis with real-time market sentiment.

Built using a scalable and modern technology stack, InvestIQ delivers a seamless user experience with features such as customizable risk profiles, tailored recommendations, trend visualizations, and curated financial news. Its architecture supports modular development, enabling the machine learning services, backend logic, and frontend interface to evolve independently while maintaining system integrity and performance.

The initial version of InvestIQ lays the groundwork for future enhancements, including automated portfolio tracking, real-time simulation tools, and social investing features. With the ability to adapt to new financial data, refine its machine learning models, and integrate with additional services, InvestIQ is well-positioned to grow into a comprehensive personal finance and investment platform.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] A. Vaswani et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[3] B. Lim et al., "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, 2021.

[4] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] D. Araci, "FinBERT: Financial sentiment analysis with pre-trained language models," *arXiv preprint arXiv:2006.08097*, 2019.

[6] Yahoo Finance API, 2024. [Online]. Available: https://www.yahoofinanceapi.com

[7] NewsAPI, 2024. [Online]. Available: https://newsapi.org

[8] J. Brownlee, *Deep Learning for Time Series Forecasting*. Machine Learning Mastery, 2021.

[9] C. C. Aggarwal, *Recommender Systems: The Textbook*. Springer, 2016.

[10] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, 2008.

[11] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[12] A. Gensler et al., "Deep learning for financial time series forecasting using stacked autoencoders," in *ESANN*, 2016.

[13] A. Tsantekidis et al., "Forecasting stock prices from the limit order book using convolutional neural networks," in *IEEE*, 2017.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[16] I. Sutskever et al., "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014.

[17] C. Olah, "Understanding LSTM networks," *colah.github.io*, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[18] F. Provost and T. Fawcett, *Data Science for Business*. O'Reilly Media, 2013.

[19] S. Rasp et al., "WeatherBench: A benchmark dataset for data-driven weather forecasting," *JAMES*, vol. 12, no. 11, pp. 1461–1479, 2020.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training," *arXiv preprint arXiv:1502.03167*, 2015.

[21] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[22] A. Shleifer and R. W. Vishny, "The limits of arbitrage," *The Journal of Finance*, vol. 52, no. 1, pp. 35–55, 1997.

[23] N. Barberis and R. Thaler, "A survey of behavioral finance," in *Handbook of the Economics of Finance*, Elsevier, 2003, pp. 1053–1128.

[24] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *The Journal of Finance*, vol. 62, no. 3, pp. 1139–1168, 2007.

[25] W. S. Chan, "Stock price reaction to news and no-news: Drift and reversal after headlines," *Journal of Financial Economics*, vol. 70, no. 2, pp. 223–260, 2003.

[26] F. Li, "The information content of forward-looking

statements in corporate filings," *Journal of Accounting Research*, vol. 48, no. 5, pp. 1049–1102, 2010.
[27] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.
[28] A. W. Lo, "The adaptive markets hypothesis," *The Journal of Portfolio Management*, vol. 30, no. 5, pp. 15–29, 2004.
[29] D. Kahneman, *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.
[30] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *KDD '16*, pp. 785–794.
[31] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[32] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *OSDI*, 2016.
[33] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.
[34] React.js Documentation, 2024. [Online]. Available: https://reactjs.org
[35] Flask Documentation, 2024. [Online]. Available: https://flask.palletsprojects.com
[36] FastAPI Documentation, 2024. [Online]. Available: https://fastapi.tiangolo.com
[37] Tailwind CSS Documentation, 2024. [Online]. Available: https://tailwindcss.com
[38] SQLite Documentation, 2024. [Online]. Available: https://sqlite.org/docs.html
[39] GitHub Repositories – LSTM Stock Prediction, 2023. [Online]. Available: https://github.com
[40] Hugging Face Transformers, 2024. [Online]. Available: https://huggingface.co/transformers
[41] Alpha Vantage API, 2024. [Online]. Available: https://www.alphavantage.co
[42] Quandl API, 2024. [Online]. Available: https://www.quandl.com
[43] Google Finance Documentation, 2023. [Online]. Available: https://www.google.com/finance
[44] AWS Cloud Machine Learning Services, 2024. [Online]. Available: https://aws.amazon.com/machine-learning
[45] Azure Cognitive Services – Text Analytics, 2024. [Online]. Available: https://azure.microsoft.com/services/cognitive-services/text-analytics