

# Predictiv: A Real-Time Stock Market Platform with Multi-Source Data Integration and Portfolio Management

L.Rajan Roop, K.Sri Charan, V.V.Kavya Sree


Under the Guidance of Ms.T.Aruna Jyothi

Department of Information Technology Matrusri Engineering College INDIA



<https://doi.org/10.55041/ijst.v2i4.379>

**Cite this Article:** Roop., L., Charan., K. & Sree, V. (2026). Predictiv: A Real-Time Stock Market Platform with Multi-Source Data Integration and Portfolio Management. International Journal of Science, Strategic Management and Technology, 02(04). <https://doi.org/10.55041/ijst.v2i4.379>

**License:**  This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

**Abstract:** This paper presents Predictiv, a full-stack real-time stock market platform designed to provide investors with comprehensive market data, financial analytics, and portfolio management capabilities. The system integrates multiple financial data sources including Yahoo Finance and Alpha Vantage APIs to deliver live price quotes, OHLCV candlestick charts, fundamental financial statements, and news sentiment analysis. The frontend is built using React.js with Redux state management, D3.js-powered charting via react-stockcharts, and a context-based dark/light theme system. The backend is implemented with Django REST Framework, providing secure JWT-based authentication and persistent portfolio and funds management. The platform features a three-tier data fallback architecture ensuring high availability, pre-loaded support for major Indian NSE stocks and global equities, and real-time market hours awareness for the National Stock Exchange of India. Results demonstrate that the system successfully delivers a responsive, data-rich trading interface suitable for both retail and research-oriented users.

**Key-Words:** Stock Market, React.js, Django, Redux, Yahoo Finance API, Alpha Vantage, Portfolio Management, Real-Time Data, Candlestick Chart, JWT Authentication

## 1 Introduction

The modern financial ecosystem demands real-time, accessible, and comprehensive market data platforms that cater to a wide spectrum of users — from casual retail investors to professional analysts. Traditional desktop-based trading terminals, while powerful, often lack the accessibility and cross-platform flexibility expected by today's users. Web-based stock market platforms bridge this gap by leveraging modern browser technologies, REST APIs, and reactive UI frameworks to deliver live financial data in an intuitive interface. Predictiv is a full-stack web application that addresses these requirements by integrating multiple financial data providers, providing real-time price tracking, rich candlestick charting, detailed financial statement analysis, and a complete portfolio management system. The platform is specifically designed with Indian NSE market support at its core, while also covering major global equities listed on NASDAQ and NYSE.

This paper describes the architecture, technology stack, data integration strategies, and key features of the Predictiv platform. Section 2 covers the system architecture and problem formulation, Section 3 presents the implementation details and solution,

Section 4 discusses results, and Section 5 concludes the paper.

## 2 Problem Formulation

Retail investors in India and globally face several challenges when accessing financial market data. Existing platforms either require costly subscriptions, are limited to a single exchange, lack comprehensive fundamental data, or do not provide a unified interface for portfolio tracking alongside live market data. Furthermore, many platforms do not gracefully handle API rate limits or failures, leading to broken user experiences.

The key challenges addressed by this work are as follows: (1) integrating multiple heterogeneous financial data APIs in a unified interface, (2) ensuring data availability through intelligent fallback mechanisms, (3) providing a secure and persistent portfolio management system, (4) supporting Indian NSE market instruments alongside global equities, and (5) delivering a responsive and themeable user interface.

### 2.1 Data Source Challenges

Financial data APIs impose strict rate limits. Alpha Vantage's free tier permits a maximum of 25 requests

per day, while Yahoo Finance imposes soft limits via CORS restrictions that require proxy intermediaries. A naive implementation that relies on a single data source will fail frequently in production. The system must therefore implement intelligent caching, fallback hierarchies, and mock data generation to ensure continuous functionality.

## 2.2 State Management Complexity

A stock market application involves numerous asynchronous data flows — live quotes, multi-year financial statements, news feeds, portfolio positions, and user authentication state — all operating concurrently. Managing these without a structured state management solution leads to prop-drilling, race conditions, and difficult-to-debug UI inconsistencies.

## 2.3 Market Hours Awareness

The National Stock Exchange (NSE) of India operates from 09:15 to 15:30 IST on weekdays. Accurately determining market open/close status from a client-side application requires timezone-aware computation, as server timestamps may differ from IST (UTC+5:30).

## 3 Problem Solution

The Predictiv platform is architected as a decoupled full-stack application with a React.js frontend communicating with a Django REST Framework backend over HTTP. The system is structured as follows.

### 3.1 System Architecture

The application follows a three-tier architecture: (1) Presentation Tier — React.js SPA with Redux, (2) Application Tier — Django REST API with JWT authentication, and (3) Data Tier — PostgreSQL/SQLite database combined with external financial APIs. The frontend is a single-page application (SPA) bootstrapped with Create React App. The backend exposes RESTful endpoints for authentication (`/api/auth/`), portfolio management (`/api/portfolio/`), and funds management (`/api/funds/`).

### 3.2 Frontend Technology Stack

The frontend is built with React.js 16, employing class components for stateful containers and functional components for presentational elements. Redux 4 with Redux Thunk middleware manages all asynchronous side effects and global application

state. The Redux store is organized into eight reducers: `authReducer`, `iexReducer`, `userReducer`, `searchReducer`, `tradeReducer`, `fundsReducer`, `newsReducer`, and `fmpReducer`.

Navigation is handled by React Router v5, with the primary layout defined in the Platform component which wraps all routes with a persistent Navbar. The charting subsystem uses `react-stockcharts 0.7.8`, a D3.js-based library that renders interactive candlestick charts with OHLCV data. `Material-UI v3` provides foundational UI components including icons and form elements.

Theming is implemented via React's Context API. The `ThemeContext` provides `isDark` state and a `toggleTheme` callback to all child components. Theme transitions are animated using CSS transitions on `background-color` and `color` properties. The default theme is dark (`#080808` background, `#f0f0f0` text) with a light mode (`#f8f9fc` background, `#0e0e1a` text). The font family used throughout is DM Sans.

### 3.3 Data Integration Layer

The data integration layer implements a three-tier fallback strategy for company profiles and a two-tier fallback for price data:

**Tier 1 — Known Stocks Database:** A hardcoded JavaScript object contains pre-verified profile data for 15 major stocks including Reliance Industries, TCS, Infosys, HDFC Bank, SBI, Zomato, Wipro, ICICI Bank, Bajaj Finance, Tata Motors (NSE), Apple, Alphabet, Microsoft, Amazon, and Snap. This ensures instant UI rendering without any network latency for the most commonly searched instruments.

**Tier 2 — Yahoo Finance via CORS Proxy:** The application queries the Yahoo Finance v8/v10 API endpoints through `corsproxy.io` or `allorigins.win` proxy services. This provides live OHLCV chart data (60 trading days, 3-month range with 1-day interval), real-time price quotes, and detailed asset profiles including business summaries, sector, industry, CEO information, and company websites.

**Tier 3 — Alpha Vantage API:** For fundamental financial data — including P/E ratio, EPS, Beta, 52-week high/low, market capitalization, profit margin, return on equity, price-to-book ratio, moving averages, and complete income statements, balance sheets, and cash flow statements — the system uses the Alpha Vantage REST API. The API key is stored securely as a React environment variable (`REACT_APP_ALPHA_VANTAGE_KEY`).

A client-side in-memory cache is implemented for all API responses. Price data is cached for 2 minutes, fundamental data for 1 hour, and news for 10

minutes. This significantly reduces API call frequency and improves perceived performance.

Mock data fallbacks are provided for all data types. The mock chart generator creates 30 days of synthetic OHLCV data using a random walk algorithm, ensuring the candlestick chart always renders even when all API sources are unavailable.

### 3.4 Backend Architecture

The backend is a Django REST Framework application providing four main API endpoint groups. The authentication endpoints (`/api/auth/login`, `/api/auth/register`, `/api/auth/logout`, `/api/auth/user`) handle user registration, login, and profile management using JWT bearer tokens. Tokens are issued with a 1-hour expiration, with the frontend automatically logging out users upon expiry via Redux-dispatched timeout handlers.

Portfolio endpoints (`/api/portfolio/`) support full CRUD operations for trade records, enabling users to add and remove stock positions. Funds endpoints (`/api/funds/`) similarly manage user account balance records. All protected endpoints require the `Authorization: Bearer <token>` header, which is configured in a shared `tokenConfig` utility function that reads from `localStorage`.

### 3.5 Key Features

**Stock Search and Discovery:** Users can search for any NSE or global stock symbol using a React Select-powered search component. The search results are dispatched to the Redux `searchReducer` and trigger cascading data fetch actions for quote, chart, profile, statistics, and news.

**Candlestick Charting:** The charting component renders interactive OHLCV candlestick charts using `react-stockcharts` and `D3.js`. Charts display up to 60 trading days of historical data with green/red candle color coding for up/down days respectively.

**Financial Statements:** The platform presents four years of annual financial data across three statement types — Income Statement (revenue, gross profit, EBIT, net income, EPS), Balance Sheet (total assets, liabilities, shareholder equity, debt breakdown), and Cash Flow Statement (operating cash flow, capital expenditures, free cash flow, depreciation).

**News Feed:** Stock-specific news articles are fetched from Alpha Vantage's `NEWS_SENTIMENT` endpoint, providing headline, source, summary, publication datetime, and an optional banner image. A curated set of mock Indian market news (Economic Times, Business Standard, Mint) serves as fallback.

**Portfolio Management:** Authenticated users can add trade records specifying symbol, quantity, and price, and delete existing positions. The portfolio state is persisted to the backend database and reloaded on each session.

### 3.6 NSE Market Hours Detection

Client-side NSE market hours detection is implemented by converting the current UTC time to IST (Asia/Kolkata timezone) using the JavaScript Intl API, then checking whether the day is a weekday (Monday–Friday) and the time falls within the 09:15–15:30 IST window. This status is included in the quote payload as `isUSMarketOpen` (repurposed field) and displayed in the UI.

## 4 Results and Discussion

The Predictiv platform was developed and tested on a Windows environment with Node.js v24 (frontend) and Python 3.13 with Django (backend). The React frontend is bootstrapped with Create React App and requires the `NODE_OPTIONS=--openssl-legacy-provider` flag when running on Node.js versions above 17 due to OpenSSL compatibility changes introduced in Node.js v18.

The three-tier data fallback strategy proved highly effective during testing. Yahoo Finance API calls through the `corsproxy.io` proxy returned live data within 500ms–1200ms on average. When proxy failures occurred, Alpha Vantage served as a reliable fallback for fundamental data. The in-memory cache reduced redundant API calls by an estimated 80% for typical browsing sessions.

The Redux state architecture cleanly separated concerns across eight reducers, enabling predictable state transitions and straightforward debugging via Redux DevTools. The thunk middleware enabled clean async action creators for all API integrations without additional boilerplate.

The candlestick chart component successfully rendered 60-day OHLCV data with smooth interactions. The dark theme was well-received during user testing, with the light theme toggle providing adequate contrast for daytime use. Portfolio and funds management operations showed correct real-time UI updates driven by Redux state changes following successful API responses.

## 5 Conclusion

This paper presented Predictiv, a comprehensive real-time stock market platform that addresses the challenges of multi-source financial data integration, robust portfolio management, and accessible market analytics. The system's three-tier data fallback architecture, client-side caching strategy, and mock data generation ensure a resilient user experience even under API rate limiting conditions.

The combination of React.js, Redux, D3.js, and Django REST Framework provides a scalable and maintainable full-stack foundation. The platform's specific support for Indian NSE market instruments, real-time price quoting, four-year financial statement history, and news sentiment integration make it a capable tool for Indian retail investors.

Future work includes adding WebSocket-based real-time price streaming, implementing advanced portfolio analytics (P&L, returns), adding stock screener functionality, integrating options chain data, and deploying the application to cloud infrastructure for production use.

### References:

- [1] D. Abramov, Redux Documentation, Redux.js.org, 2015. Available: <https://redux.js.org>
- [2] Meta Platforms Inc., React – A JavaScript library for building user interfaces, React.js Documentation, 2023. Available: <https://react.dev>
- [3] Alpha Vantage Inc., Alpha Vantage API Documentation, 2023. Available: <https://www.alphavantage.co/documentation/>
- [4] Yahoo Finance, Yahoo Finance API, query1.finance.yahoo.com, 2023.
- [5] D. Bostock, M. Ogievetsky, J. Heer, D3: Data-Driven Documents, IEEE Transactions on Visualization and Computer Graphics, Vol.17, No.12, 2011, pp. 2301-2309.
- [6] Django Software Foundation, Django REST Framework Documentation, 2023. Available: <https://www.django-rest-framework.org>
- [7] S. Bird, S. Klein, E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009.
- [8] B. Lawson, R. Sharp, Introducing HTML5, New Riders, 2011.
- [9] NSE India, National Stock Exchange of India – MarketTimings, 2023. Available: <https://www.nseindia.com>
- [10] Clearbit Inc., Clearbit Logo API Documentation, 2023. Available: <https://clearbit.com/docs#logo-api>