Project Documentation: FinHub - AI-Powered Stock and Sentiment Analysis Dashboard

1. Introduction

This document provides a formal overview of FinHub, an advanced end-to-end Fintech tool designed for stock and sentiment analysis. The project is designed to fetch real-time stock market data, extract financial news, conduct sentiment analysis, forecast future stock prices, and visualize insights through an interactive dashboard.

The key objectives of the system are:

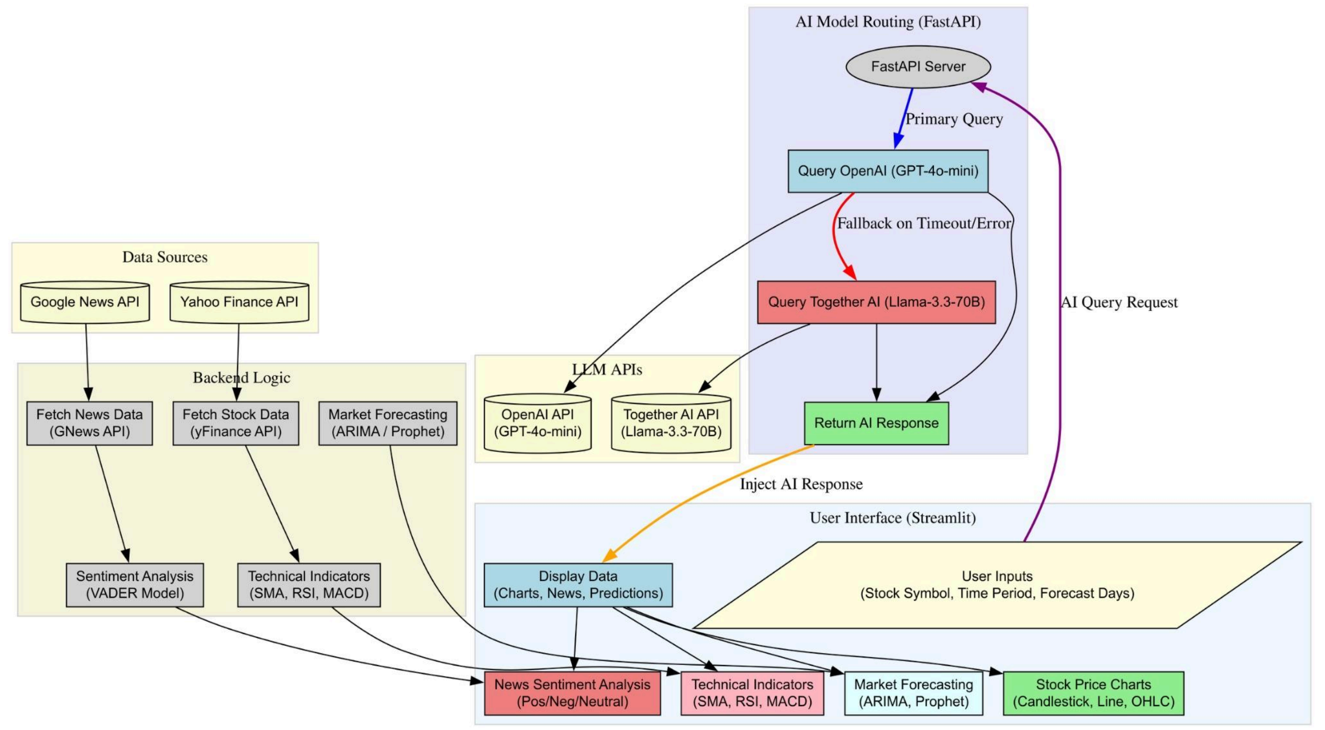
- To monitor live stock price movements using interactive candlestick charts
- To analyze sentiment trends from financial news headlines
- To display sentiment insights using bar charts and heatmaps
- To provide AI-powered forecasting for future stock price movements
- To deliver advanced technical indicators for comprehensive market analysis

The project is developed using a modern technology stack with Streamlit and Next.js for frontend development, FastAPI for backend services, and advanced AI models for sentiment analysis and forecasting.

2. System Architecture

The system follows a modular architecture with four main components:

2.1 Architecture Diagram


AI Model Routing (FastAPI)
FastAPI Server
Primary Query
Query OpenAI (GPT-4o-mini)
Fallback on Timeout/Error
Query Together AI (Llama-3.3-70B)
Return AI Response
AI Query Request
Data Sources
Google News API
Yahoo Finance API
Backend Logic
Fetch News Data (GNews API)
Fetch Stock Data (yFinance API)
Market Forecasting (ARIMA / Prophet)
Sentiment Analysis (VADER Model)
Technical Indicators (SMA, RSI, MACD)
LLM APIs
OpenAI API (GPT-4o-mini)
Together AI API (Llama-3.3-70B)
Inject AI Response
User Interface (Streamlit)
Display Data (Charts, News, Predictions)
User Inputs (Stock Symbol, Time Period, Forecast Days)
News Sentiment Analysis (Pos/Neg/Neutral)
Technical Indicators (SMA, RSI, MACD)
Market Forecasting (ARIMA, Prophet)
Stock Price Charts (Candlestick, Line, OHLC)

2.2 Technology Stack

Frontend and UI Development

- **Streamlit**: Used for developing the interactive dashboard
- **Plotly**: Employed for visualizing stock price movements and sentiment data

Backend and Data Processing

- **FastAPI**: Powers the backend API services and AI model routing
- **Alpha Vantage API**: Alternative source for financial data
- **Yahoo Finance API (yFinance)**: Fetches real-time stock market data
- **Google News API (feedparser)**: Scrapes financial news headlines related to a given stock symbol
- **VaderSentiment**: Conducts advanced sentiment analysis on extracted news headlines
- **TextBlob**: Alternative NLP library for sentiment analysis
- **Pandas**: Handles data manipulation and structuring for visualization

AI and Forecasting Models

- **Prophet**: Meta's forecasting model for time-series predictions
- **ARIMA**: AutoRegressive Integrated Moving Average for stock price forecasting
- **Together AI**: Leveraging Llama-3.3-70B for AI-powered insights
- **OpenAI**: Using GPT-4o-mini for market analysis and insights

3. System Implementation and Features

3.1 Repository Structure

The project is organized into four main components:

1. **FinTech API (Stock & Market Data Processing)**
    - ○ Fetches live stock market data
    - ○ Retrieves historical data for trend analysis
    - ○ Provides JSON-formatted API responses
    - ○ Computes technical indicators (SMA, RSI, MACD)
2. **Market Sentiment Analysis** Fetches latest financial news from Google News API
    - ○ Applies VaderSentiment Analysis for polarity detection
    - ○ Provides structured sentiment scores (Positive, Negative, Neutral)
3. **XNL - AI-Based Forecasting (Stock Price Predictions)**
    - ○ Predicts future stock prices based on historical trends
    - ○ Uses ARIMA for time-series forecasting
    - ○ Implements Prophet for better accuracy
    - ○ Provides AI-powered market trend insights
4. **Frontend**
    - ○ Backend:
        - ■ Developed using FastAPI
        - ■ Routes AI queries via OpenAI (GPT-4o-mini) & Together AI (Llama-3.3-70B)
        - ■ Hosts forecasting models (ARIMA, Prophet)
        - ■ Fetches stock data & news sentiment
        - ■ Hosted on Railway.app
    - ○ Frontend:
        - ■ Built using Next.js & Tailwind CSS
        - ■ Provides an interactive financial dashboard
        - ■ Uses Streamlit Cloud for AI model visualization

3.2 Stock Price Monitoring

The system retrieves real-time stock price data from the Yahoo Finance API and Alpha Vantage API at one-minute intervals for the current trading day. The data includes:

- Open price (Price at the start of the interval)
- High price (Highest price within the interval)
- Low price (Lowest price within the interval)
- Close price (Final price at the end of the interval)

The stock price data is visualized using a candlestick chart implemented with Plotly, allowing users to track market trends effectively.

3.3 Technical Indicators

The system computes several technical indicators to aid in stock analysis:

- Simple Moving Average (SMA)
- Relative Strength Index (RSI)
- Moving Average Convergence Divergence (MACD)

These indicators are displayed alongside the stock price charts to provide additional insights for trading decisions.

3.4 Financial News and Sentiment Analysis

News Data Extraction

- The system fetches Google News RSS feeds to retrieve the latest stock-related headlines
- It extracts the top news articles relevant to the given stock symbol

Sentiment Analysis

- Each news headline is processed using VaderSentiment and TextBlob NLP to determine its sentiment polarity:
    - Positive sentiment: Indicates favorable market trends
    - Negative sentiment: Suggests market concerns or bearish movement
    - Neutral sentiment: No significant impact detected

3.5 Visualization of Sentiment Data

Sentiment Bar Chart

- A bar chart is generated using Plotly Express to display sentiment scores
- Headlines are positioned above the bars instead of on the side to improve readability
- Sentiment scores are represented on the x-axis, while news headlines are arranged accordingly

Sentiment Heatmap

- A heatmap is implemented to represent sentiment scores in a structured format
- The issue of overlapping sentiment scores has been resolved to ensure better visibility
- The color scale ranges from red (negative sentiment) to green (positive sentiment) to provide a clear indication of market sentiment

3.6 AI-Based Forecasting

The system incorporates advanced AI models for predictive analysis:

ARIMA Model

- Implements AutoRegressive Integrated Moving Average for time-series forecasting
- Analyzes historical data patterns to predict future stock movements

Prophet Model

- Utilizes Meta's Prophet forecasting model for enhanced accuracy
- Handles seasonal trends and market patterns for more reliable predictions

AI-Powered Insights

- Integrates with Together AI (Llama-3.3-70B) and OpenAI (GPT-4o-mini) for advanced market analysis
- Provides natural language insights on market trends and investment opportunities

4. User Interface Adjustments and Improvements

Several modifications were made to optimize the display and layout of charts:

- Reorganized the order of visualization:
    - First, the stock price chart is displayed
    - Second, the sentiment bar chart is presented
    - Third, the sentiment heatmap is shown
    - Fourth, the AI forecasting visualization is presented
- Fixed positioning of headlines in the sentiment bar chart by placing them directly above the corresponding bars
- Resolved overlapping sentiment scores in the heatmap, ensuring clear representation
- Enhanced UI with Next.js and Tailwind CSS for a more professional financial dashboard experience

5. Local Development Setup

Clone the Repository

Unset

```
git clone
https://github.com/Architgarg2003/XNL-21BCE11235-LLM-1.git
cd XNL-21BCE11235-LLM-1
```

Set Up the Backend (FastAPI & AI Models)

Unset

```
cd backend
python -m venv venv
source venv/bin/activate  # On macOS/Linux
venv\Scripts\activate    # On Windows
pip install -r requirements.txt
```

Set Up & Run the Frontend (Streamlit)

Unset

```
cd frontend
streamlit run streamlit_app.py
```

6. Deployment and Hosting

The project is deployed on multiple cloud platforms:

- **Streamlit Cloud**: Hosts the Streamlit frontend application
- **Railway.app**: Powers the FastAPI backend services
- **Frontend URL**: https://xnl-21bce11235-llm.streamlit.app
- **Backend URL**: https://stock-dashboard-backend-production.up.railway.app/
- **GitHub URL (XNL-21BCE2223-LLM-1)**: https://github.com/Mudit1404/XNL-21BCE2223-LLM-1
- **GitHub URL (XNL-21BCE11235-LLM-1)**: https://github.com/Architgarg2003/XNL-21BCE11235-LLM-1.git

7. Future Enhancements

Potential improvements to enhance the system include:

- **Integration of Additional AI Models**: Exploring more sophisticated forecasting models like LSTM and transformers
- **Expanded Data Sources**: Incorporating data from Bloomberg, Reuters, and SEC filings for comprehensive market insights
- **Advanced User Interface**: Further refining the Next.js and Tailwind CSS frontend for enhanced user experience
- **Portfolio Management Features**: Adding personal portfolio tracking and optimization recommendations

- **Real-time Alerts System**: Implementing push notifications for significant market movements and sentiment shifts

8. Conclusion

This project successfully integrates real-time stock market tracking with financial news sentiment analysis and AI-powered forecasting into a user-friendly dashboard. The implemented improvements ensure accurate sentiment representation, enhanced readability of visualizations, and reliable future price predictions. The system lays a strong foundation for future expansion, including deeper AI-driven market predictions, more comprehensive financial insights, and improved trading decision support.

I would like to express my sincere gratitude to XNL Innovations for providing me with this incredible opportunity to work on this project as part of the placement process. It has been a challenging yet rewarding experience, allowing me to d