Final Project Proposal Deliverable:

Team 8: Sathvik Vadavatha : 002443505
Rutuja Patil : 002827468
Sakshi Aade : 002813258

Title: FAANG Stock Analytics and Recommendation Platform

1. Introduction:

Background: Making smart stock market investment decisions often requires looking at historical data, staying updated with real-time market trends, and understanding the overall sentiment. Yet, retail investors and analysts frequently struggle to bring all these pieces together into a single, easy-to-use system. With the rapid advancements in machine learning and AI in the financial world, there's a great opportunity to build an intuitive and intelligent platform tailored for stock analysis, predictions, and recommendations, on FAANG companies. Such a platform could simplify complex data, making it more accessible and actionable for everyday investors. Additionally, it would bridge the gap between retail investors and institutional-grade tools, empowering users with data-driven insights and smarter decision-making capabilities.

Objective: The aim is to create an interactive stock analytics platform that brings together historical data visualization, stock price predictions, sentiment analysis, live updates, and high-quality investment recommendations. To make the platform even more engaging and user-friendly, it will include an LLM-powered chatbot that allows users to ask questions and interact naturally, offering a seamless and enhanced experience.

2. Project Overview:

Scope:

The project integrates diverse data sources, including historical stock data for FAANG companies to identify long-term trends, real-time stock prices via financial **APIs** for up-to-date insights, and market narratives from **NewsAPI** or **Google News**. Sentiment data from Reddit (PRAW) and Twitter (Tweepy) provide insights into public and investor sentiment, forming a robust data ecosystem.

Utilizing cutting-edge technologies, the platform features **Streamlit** for interactive dashboards and **FastAPI** for efficient backend processing. Data storage incorporates **MS SQL** for user authentication and **Pinecone** for embedding vector storage. **ARIMA, Prophet,** and **LSTM** models forecast stock prices, while **HuggingFace** transformers power sentiment analysis. An **OpenAI GPT-4** chatbot enhances user engagement with natural language queries. Deliverables include dashboards for stock price analysis, sentiment aggregation, real-time updates with news summaries, and actionable investment recommendations, tailored for retail investors, financial analysts, and data science enthusiasts.

3. Problem Statement:

Current Challenges

- Existing tools often fail to integrate key elements such as historical data analysis, real-time updates, and sentiment trends into a single platform, leaving users to rely on multiple disconnected resources.
- Retail investors, who may lack in-depth financial expertise, find it challenging to interpret complex technical indicators and sentiment data, which limits their ability to make informed decisions.
- Institutional-grade recommendations and insights are typically locked behind expensive platforms or exclusive services, making them inaccessible to everyday investors.

Opportunities

- By leveraging predictive models, the platform can empower users to make data-driven investment decisions, offering projections that simplify complex analyses.
- Integrating an AI-driven assistant allows for personalized queries, enabling users to interact with the platform in a conversational manner and access tailored insights more intuitively.
- Combining historical data trends, real-time market updates, and sentiment analysis creates a comprehensive and holistic tool, providing users with a 360-degree view of the market and streamlining their decision-making process.

4. Methodology:

Data Sources:

Dataset: https://www.kaggle.com/datasets/aayushmishra1512/faang-complete-stock-data
News articles: NewsAPI
Reddit stock posts: Reddit API

Tools and Technologies:

- ETL Pipeline: Python, Pandas, Airflow, AWS S3.

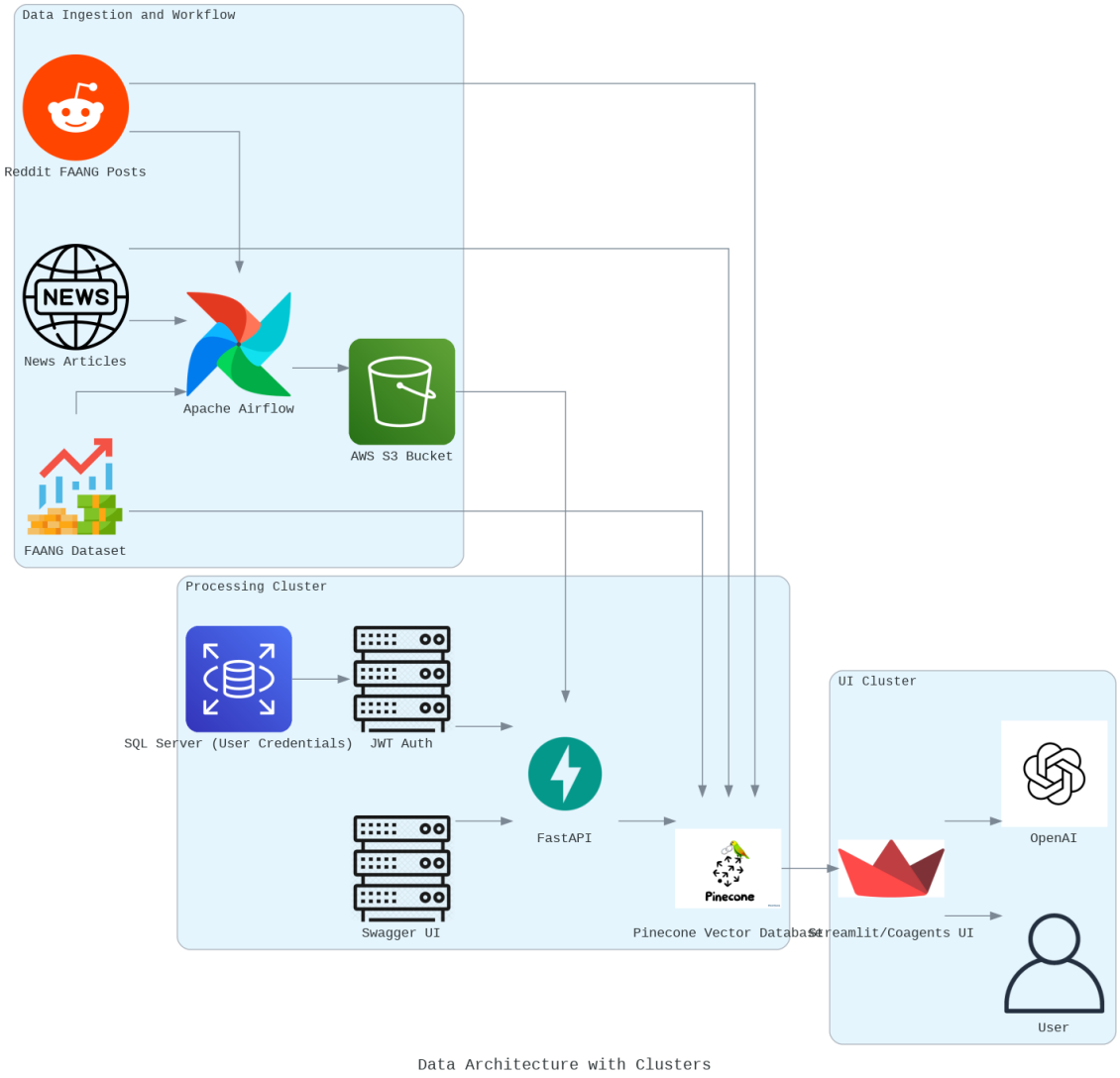
- Storage: AWS S3
- Database: SQL Server hosted on RDS database for user credentials
- Vector Database: Pinecone for processed data.
- UI: Streamlit or Coagents
- API: OpenAI API, NewsAPI, Vantage API, TavilyAPI, Reddit/Twitter API.
- FastAPI for endpoints and LangGraph for agents:
    - Stock prediction results.
    - News summarization.
    - Chatbot responses.
- Cloud: Docker and Amazon EC2

Data Processing and Transformation:
EDA: http://localhost:8888/notebooks/EDA.ipynb#
POC: http://localhost:8889/notebooks/Downloads/PoC.ipynb#

Data Architecture Design:

The below data architecture diagram outlines a scalable pipeline with three clusters:

1. **Data and Workflow Cluster:** Data from Reddit, news articles, and FAANG stock datasets is ingested, orchestrated by Apache Airflow, and stored in AWS S3 for further use.
2. **Processing Cluster**: Includes FastAPI for backend API handling, JWT Auth for secure authentication, SQL Server for credential storage, and Pinecone for vectorized data retrieval.
3. **UI Cluster**: Features a Streamlit UI for visualization, integrated with OpenAI for generative AI insights, enabling users to interact with processed data.



Data Architecture with Clusters

5. Project Plan and Timeline:

Date	Task	Assigned To	Deliverable
Nov 24	Set up Airflow DAGs for fetching historical data and live news	Sathvik	Airflow DAGs to automate data ingestion tasks.
Nov 24	Fetch and clean historical stock data from the dataset	Sakshi	Cleaned historical stock data for modeling.
Nov 24	Fetch live news articles using NewsAPI and validate data	Rutuja	Preprocessed live news data in JSON format.
Nov 25	Preprocess stock data for model training (ARIMA, Prophet, LSTM)	Sathvik	Preprocessed dataset ready for modeling.
Nov 25	Integrate NewsAPI with OpenAI GPT for sentiment analysis	Sakshi	Functional sentiment analysis pipeline.
Nov 25	Test Airflow DAGs for initial tasks	Rutuja	Validated task execution and logs in Airflow.
Nov 26	Train ARIMA model for stock price prediction	Sakshi	ARIMA model predictions for the next 4 years.
Nov 26	Perform sentiment analysis on fetched articles using OpenAI GPT	Rutuja	Sentiment scores and summaries for the articles.
Nov 27	Train Prophet model for stock price prediction	Rutuja	Prophet model predictions for the next 4 years.
Nov 27	Integrate ARIMA predictions with the backend API	Sathvik	Backend API updated with ARIMA predictions.
Nov 28	Train LSTM model for stock price prediction	Sathvik	LSTM model predictions for the next 4 years.
Nov 28	Visualize predictions from ARIMA and Prophet models	Sakshi	Comparison plots for ARIMA and Prophet predictions.

Nov 29	Integrate Prophet predictions into the backend API	Sakshi	Backend API updated with Prophet predictions.
Nov 29	Visualize predictions from LSTM and integrate into the API	Rutuja	LSTM prediction plots and API integration.
Nov 30	Test the API endpoints for predictions and sentiment analysis	Rutuja	Functional API with validated endpoints for predictions and sentiment analysis.
Nov 30	Develop the chatbot backend using OpenAI GPT	Sathvik	Chatbot backend capable of responding to user queries.
Dec 1	Implement live news updates pipeline in Airflow	Sakshi	Airflow DAGs to dynamically fetch live news articles.
Dec 1	Perform integration testing for backend components	Rutuja	Fully functional backend with working API endpoints.
Dec 2	Develop the front-end interface using Streamlit	Sathvik	Interactive front-end interface with prediction visualization and sentiment analysis results.
Dec 2	Connect chatbot to the front-end	Sakshi	Chatbot integrated into the application interface.
Dec 3	Perform unit testing for all individual components	Rutuja	Unit test results for each module (models, API, chatbot, sentiment analysis).
Dec 3	Create documentation for data ingestion and preprocessing pipeline	Sakshi	Documentation on data pipeline and API integrations.
Dec 4	Conduct integration testing for the full application	Sathvik	End-to-end testing results ensuring all components work seamlessly.

Dec 5	Prepare the final presentation and deployment setup	Sakshi	Final presentation slides and app deployed on a public or local cloud.
Dec 5	Write user manual and technical documentation	Rutuja	User manual for app usage and technical documentation for backend components.
Dec 6	Submit the project	All Members	Fully completed project submitted with all deliverables (app, code, presentation, documentation).

6. Resources and Team:

Team Member	Role	Responsibilities
Sathvik	Backend & Integrator	Develop backend components (API, Airflow, LSTM model), integrate predictions, and perform end-to-end testing.
Sakshi	Data & Visualization	Clean historical data, Prophet models, create documentation, and prepare final presentation.
Rutuja	Sentiment Analysis & QA	Perform sentiment analysis, validate data pipelines, test APIs, and write user manual and technical documentation.

7. Risks and Mitigation Strategies:

Risks

- **Reddit API**: Limited data availability and noisy, unstructured content may reduce the quality of sentiment analysis.
- **NewsAPI**: Biased or inconsistent financial news coverage and limited article details can affect sentiment insights.
- **Alpha Vantage API**: API call limits and potential data latency may impact real-time stock updates.
- **Kaggle Dataset**: Outdated or incomplete historical data and format inconsistencies may affect prediction accuracy.
- **Application**: Users may misinterpret recommendations or be overwhelmed by complex data without personalized options.

Mitigation Strategies

- **Reddit API**: Diversify content sources, preprocess comments for relevance, and cache frequently accessed data locally.
- **NewsAPI**: Use alternative APIs or trusted sources and aggregate sentiment across multiple articles to reduce bias.
- **Alpha Vantage API**: Employ fallback APIs (e.g., Yahoo Finance) and cache data to minimize reliance on real-time calls.
- **Kaggle Dataset**: Conduct thorough EDA and complement with additional datasets to address gaps and ensure data quality.
- **Application**: Simplify the interface with educational tooltips, customizable recommendations, and disclaimers for informed decision-making.

8. Expected Outcomes and Benefits:

Expected Outcomes:

Accuracy:

- Achieve >85% accuracy in stock price predictions using Prophet/ARIMA models.

Improved Data Coverage:

- Extract relevant and high-quality data such as News articles, Buy/Sell recommendations, and answer retrieval for >95% of user queries by integrating multiple data sources and preprocessing pipelines.

Sentiment Insights Quality:

- Ensure >90% accuracy in sentiment classification for Reddit and news data by aggregating multiple models' outputs and validating results against real-world events.

Expected Benefits

- **Actionable Insights:**
  Provide retail investors and analysts with precise, real-time recommendations based on integrated insights from historical trends, live updates, and sentiment data.

- **High Data Reliability:**
  Leverage redundant data sources and robust pipelines to ensure minimal disruption in providing stock price, news, and sentiment data.

- **Comprehensive Stock Knowledge Platform:**
  Build a unified platform that integrates advanced predictive models, real-time updates, sentiment analysis, and LLM-based agents to serve as a one-stop solution for both institutional users and public investors, enhancing decision-making and knowledge accessibility.

9. Conclusion:

The FAANG Stock Analytics and Recommendation Platform combines cutting-edge technologies like Apache Airflow for data orchestration, AWS S3 for scalable storage, FastAPI for backend APIs, Pinecone for vectorized data retrieval, and Streamlit for visualization. It integrates datasets from Kaggle, NewsAPI, and Reddit API, leveraging models such as ARIMA,

Prophet, and LSTM for stock prediction. Sentiment analysis using OpenAI and dynamic API integration ensures reliable insights. The platform offers a streamlined pipeline for data ingestion, processing, and visualization, ensuring high data quality and accessibility.

Potential Impact of the Project:

This platform allows retail investors and analysts with real-time, actionable insights by combining historical trends, live data, and sentiment analysis. It enhances decision-making through accurate stock predictions, ensures reliability via robust data pipelines, and provides a unified interface for comprehensive stock analytics. By integrating advanced technologies and addressing data quality challenges, the project serves as a one-stop solution for informed investment decisions and market analysis.

References:

- Kaggle - FAANG Complete Stock Data
- NewsAPI
- Alpha Vantage API
- Pinecone Vector Database
- OpenAI API Documentation
- FastAPI Documentation
- LangGraph Documentation