**1. Introduction** This application is a web-based financial analysis and stock prediction platform that leverages AI-powered insights to provide users with valuable information on stocks, trends, and market sentiments. Built using Flask for the backend and Streamlit for an interactive interface, the application integrates stock data retrieval, AI-generated insights, and chatbot functionalities. Users can analyze stock trends, receive stock price predictions using machine learning models, and access AI-generated market insights based on real-time data from Yahoo Finance and news sources.

**2. Objective** The primary objective of this application is to assist investors and traders in making informed decisions by providing real-time stock data, AI-driven market analysis, and stock price predictions. The application also includes a chatbot that answers financial queries concisely, making stock market information accessible and actionable for users. By integrating AI and machine learning, the system aims to bridge the gap between raw financial data and meaningful investment insights.

**3. Software and Hardware Requirement Specifications**

* **Software Requirements:**
  + Python 3.x
  + Flask (Backend Web Framework)
  + Streamlit (Interactive UI)
  + yFinance (Stock Market Data API)
  + Google Generative AI (Gemini API)
  + AWS Bedrock (Claude AI Model)
  + NumPy, Scikit-learn (ML for stock prediction)
  + Requests, BeautifulSoup (Web scraping for news data)
* **Hardware Requirements:**
  + Minimum 8GB RAM
  + Quad-core Processor
  + High-speed Internet Connection
  + 10GB Free Disk Space

**4. Problem Definition/Methodology** The problem addressed by this application is the lack of a unified platform that offers real-time stock data analysis, AI-generated insights, and stock price predictions. The methodology involves:

1. Collecting stock data from Yahoo Finance.
2. Scraping news articles to assess market sentiment.
3. Using Linear Regression for stock price prediction.
4. Employing AI models (Gemini and AWS Bedrock) for market insights and chatbot responses.
5. Displaying results in a user-friendly web interface using Flask and Streamlit.

**5. Implementation Plan with Modules**

1. **Stock Data Retrieval:** Fetches stock market data using yFinance.
2. **News Aggregation:** Scrapes financial news to assess market sentiment.
3. **Stock Price Prediction:** Uses Linear Regression to predict future stock prices.
4. **AI-Generated Insights:** Utilizes Google Gemini and AWS Claude AI for financial insights.
5. **Chatbot Module:** Answers financial queries using AI models.
6. **Frontend (index.html & chatbot.html):** Implements an interactive UI using HTML, CSS, and Streamlit.

**6. Expected Outcome**

* Users will receive accurate real-time stock data and AI-generated insights.
* AI-driven market sentiment analysis will assist in decision-making.
* The chatbot will provide quick and concise financial advice.
* Stock price prediction will help users understand potential future trends.
* A seamless user experience integrating Flask, Streamlit, and AI models.

**7. Conclusion** This application effectively integrates AI, machine learning, and financial data analytics to assist investors and traders. By leveraging APIs from Yahoo Finance, AWS Bedrock, and Google Generative AI, it provides users with an intuitive and intelligent tool to make informed investment decisions. The combination of stock analysis, AI-powered insights, and a chatbot creates an efficient and robust financial assistant.

**8. References**

1. Yahoo Finance API - <https://pypi.org/project/yfinance/>
2. Flask Documentation - <https://flask.palletsprojects.com/>
3. Google Gemini AI - <https://ai.google.dev/>
4. AWS Bedrock AI - <https://aws.amazon.com/bedrock/>
5. BeautifulSoup - <https://www.crummy.com/software/BeautifulSoup/>
6. Streamlit - <https://docs.streamlit.io/>
7. NumPy & Scikit-learn - <https://scikit-learn.org/>