**Software Documentation for Insight Invest**

**1. Introduction**

**1.1 Purpose**

Insight Invest is an AI-powered equity research analyst that automates the manual tasks of an equity research analyst. It gathers financial data from multiple sources, analyzes stock-related news, and evaluates company annual reports to provide investment insights. The tool aims to streamline the investment decision-making process by offering predictive analytics, risk assessments, and automated reporting.

**1.2 Scope**

This software documentation describes the functionality, system architecture, and requirements of Insight Invest. The system integrates various machine learning models and APIs to automate data collection and analysis, enhancing decision-making for investors. Insight Invest will serve financial analysts, institutional investors, and researchers who require an in-depth understanding of stock market trends and company performance.

**1.3 Definitions, Acronyms, and Abbreviations**

* **AI**: Artificial Intelligence
* **LSTM**: Long Short-Term Memory (a type of recurrent neural network)
* **RAG**: Retrieval Augmented Generation
* **API**: Application Programming Interface
* **Yahoo Finance**: A financial data provider
* **Multimodal AI**: AI models that analyze multiple types of data (text, images, tables, etc.)
* **Risk Analysis**: Evaluating potential risks associated with a financial decision
* **Competitor Analysis**: Comparing financial and operational risks between industry rivals

**1.4 References**

* IEEE 1063-2001 Standard for Software User Documentation
* Yahoo Finance API Documentation
* Google Gemini API Documentation
* News API Documentation
* Financial Risk Assessment Models
* Machine Learning in Financial Forecasting

**2. Overall Description**

**2.1 Product Perspective**

Insight Invest acts as an intelligent assistant for equity research analysts, integrating financial data from various sources to provide comprehensive analysis and visualization. The system combines AI-driven insights, financial modeling, and data aggregation to deliver actionable investment recommendations. By automating manual research processes, Insight Invest enhances efficiency, reduces errors, and offers real-time stock insights.

**2.2 Product Functions**

The software provides six key services:

1. **Data Visualization** - Graphical representation of stock performance and trends.
2. **News Analysis** - Aggregation and summarization of stock-related news.
3. **RAG on Annual Financial Reports** - AI-driven chatbot analysis of financial documents.
4. **RAG on Finance Websites** - AI chatbot for financial website data analysis.
5. **Risk Analysis Report Generation** - Automated risk assessment of financial reports.
6. **Competitor Analysis** - Comparative risk assessment between rival companies.

**2.3 User Characteristics**

* **Financial Analysts**: Require deep insights into company performance.
* **Investors**: Need quick decision-making tools.
* **Researchers**: Use historical data for financial studies.
* **Portfolio Managers**: Assess risk and stock performance trends.

**2.4 Constraints**

* Real-time data extraction depends on external APIs.
* Large-scale document processing requires high computational resources.
* Financial forecasting accuracy is dependent on data quality.
* Compliance with financial regulations and data privacy laws.

**3. Specific Requirements**

**3.1 Functional Requirements**

Here's an expanded version of **3.1.1 Data Visualization** with additional details:

### ****3.1.1 Data Visualization****

#### ****Input:****

* **Stock Ticker Symbol** (e.g., AAPL for Apple Inc.)
* **Date Range** (Start Date and End Date)

#### ****Output:****

* **Graphical Representation of Stock Performance**, including various financial indicators and trend analysis.
* **Predictive Modeling Charts**, forecasting future stock trends based on historical data.

#### ****Features:****

##### **1. Historical Stock Price Trends**

* Displays historical stock prices as a **line chart** with data points for **Open, High, Low, Close (OHLC)** prices.
* Users can visualize trends over different time frames:
  + **Intraday** (1-minute to hourly data)
  + **Short-term** (daily, weekly)
  + **Long-term** (monthly, yearly)

##### **2. Technical Indicators**

To assist investors in making informed decisions, various financial indicators are provided:

* **Moving Averages (SMA & EMA)**: Helps smoothen price data to identify trends.
* **Relative Strength Index (RSI)**: Determines if a stock is **overbought or oversold** (values range from 0-100).
* **Moving Average Convergence Divergence (MACD)**: Tracks momentum shifts in stock trends.
* **Bollinger Bands**: Measures volatility and potential overbought/oversold conditions.
* **Volume Analysis**: Shows trading volume, indicating investor interest in a stock.

##### **3. Interactive Charting**

* Users can interact with **zoomable** and **customizable** charts to analyze trends in detail.
* **Multiple Stocks Comparison**: Allows users to compare multiple stock trends side by side.
* **Candlestick Charting**: Provides detailed **daily price movement**, useful for technical analysis.

##### **4. Predictive Modeling Using LSTM**

* **Long Short-Term Memory (LSTM) Neural Network** is used for forecasting stock prices.
* The model learns from past stock price data to predict future movements.
* Users can view **next-day, weekly, or monthly** price predictions based on historical patterns.
* The system provides a **confidence score** for predictions to indicate accuracy levels.

##### **5. Anomaly Detection**

* Detects unusual stock price movements based on historical trends.
* Alerts users when a significant deviation occurs (e.g., **sudden price spikes or drops**).

##### **6. Sector and Market Trends**

* Analyzes how a specific stock performs **relative to its industry sector**.
* Provides **heatmaps** of the stock market to visualize top gainers and losers.

##### **7. Export and Reporting**

* Users can **download charts and reports** in **PDF, Excel, or CSV formats**.
* Generates an **automated stock analysis report** summarizing key insights.

#### ****Tools & Technologies Used:****

* **Yahoo Finance API**: Fetches stock price and historical market data.
* **Seaborn & Matplotlib**: Used for visualizing stock trends.
* **Pandas & NumPy**: Data processing for stock analytics.
* **LSTM Neural Networks**: Deep learning model for price forecasting.

Would you like additional details, such as workflow diagrams or expanded descriptions of each analysis method? 🚀

### 3.1.2 News Analysis

#### ****Input:****

* Stock ticker symbol (e.g., AAPL for Apple Inc.).
* Date range (optional) for filtering news within a specific period.

#### ****Output:****

* A list of news articles relevant to the entered stock ticker.
* Metadata associated with each news article, including:
  + **Headline:** Title of the news article.
  + **Source:** Name of the news publisher.
  + **Published Date:** Timestamp of when the news was published.
  + **Sentiment Score:** A numerical value indicating whether the news is positive, negative, or neutral.
  + **Keywords:** Key terms extracted from the article.

#### ****Options:****

1. **Download News Data:** Users can export the retrieved news articles into an MS Excel file, including metadata and sentiment analysis.
2. **Summarized Insights:** The system provides an AI-generated summary of all articles, highlighting key events that may impact the stock.

#### ****Features:****

##### **1. Sentiment Analysis on Stock-Related News**

* The system performs **Natural Language Processing (NLP)** on news headlines and article content.
* It categorizes news sentiment as **Positive, Negative, or Neutral** based on predefined AI models.
* A **Sentiment Score** is generated on a scale of -1 to +1:
  + **-1.0 to -0.5:** Highly negative news.
  + **-0.5 to 0.0:** Slightly negative news.
  + **0.0 to 0.5:** Slightly positive news.
  + **0.5 to 1.0:** Highly positive news.

##### **2. Keyword Extraction and Event Impact Prediction**

* Extracts important terms such as **mergers, acquisitions, stock splits, SEC investigations, earnings reports**, etc.
* Identifies **high-impact events** that could influence stock prices.
* Compares recent news with historical trends to predict short-term stock movement.

##### **3. News Categorization and Filtering**

* News articles are categorized into:
  + **Earnings Reports** (e.g., quarterly financial results).
  + **Regulatory Changes** (e.g., SEC filings, legal issues).
  + **Market Trends** (e.g., economic policies, industry trends).
  + **Company-Specific Events** (e.g., product launches, leadership changes).
* Users can filter news based on **category**, **sentiment**, or **relevance**.

##### **4. Real-time Alerts and Notifications**

* Users can set alerts for critical stock-related news.
* Notifications are sent via **email or SMS** if a significant event is detected (e.g., a CEO resignation, a major lawsuit).

#### ****Tools & Technologies Used:****

* **Yahoo Finance API:** Fetches stock-related news.
* **News API:** Aggregates news articles from multiple sources.
* **Natural Language Processing (NLP) Models:** For sentiment analysis and keyword extraction.
* **Machine Learning Models:** For event impact prediction.

### ****3.1.3 RAG on Annual Reports****

#### ****Input:****

* **Annual Financial Report** (PDF, Word, or scanned document).
* Optional: **Specific sections or questions** for targeted analysis.

#### ****Output:****

* **Chatbot-assisted interactive analysis** of the financial report.
* **Detailed breakdown** of balance sheets, income statements, and cash flow statements.
* **Summarized insights** extracted from tables, graphs, and numerical data.

#### ****Features:****

##### **1. AI-Powered Financial Document Parsing**

* Uses **Optical Character Recognition (OCR)** for extracting data from scanned PDFs.
* Natural Language Processing (NLP) identifies **key financial metrics**.
* Detects **important financial ratios, trends, and anomalies** in the document.

##### **2. Automated Analysis of Financial Statements**

* **Balance Sheet Analysis**:
  + Compares assets, liabilities, and equity over different years.
  + Identifies leverage ratios and liquidity risk.
* **Income Statement Analysis**:
  + Extracts revenue, expenses, profit margins, and earnings trends.
  + Detects irregularities in net income growth.
* **Cash Flow Statement Analysis**:
  + Analyzes operating, investing, and financing cash flows.
  + Identifies cash flow consistency and dependency on external financing.

##### **3. Graph and Table Interpretation via Multimodal AI**

* AI-driven extraction of **tables, graphs, and charts** from the document.
* Converts financial trends into **interactive visualizations**.
* Detects patterns in revenue growth, expense allocation, and profitability.

##### **4. AI-Driven Q&A for Financial Insights**

* Users can ask **specific financial questions** about the report.
* AI retrieves the most relevant sections and provides summarized responses.
* Example prompts:
  + “What was the net income trend over the last five years?”
  + “How does this company’s debt-to-equity ratio compare to industry standards?”

##### **5. Red Flag Detection for Investment Risks**

* Highlights **significant financial risks**, such as:
  + Unusual debt accumulation.
  + Declining revenue trends.
  + Regulatory compliance issues.
* Provides **recommendations** based on financial health assessment.

##### **6. Export and Reporting**

* Generates **automated financial summaries** in PDF or Excel format.
* Users can **download the AI-extracted data** for further analysis.

#### ****Tools & Technologies Used:****

* **Google Gemini Multimodal AI**: For text, table, and graph interpretation.
* **OCR Technology**: Extracts text from scanned financial documents.
* **Financial NLP Models**: Identifies key metrics and trends.

### ****3.1.4 RAG on Finance Websites****

#### ****Input:****

* **Website URL**: Users can enter a specific financial website (e.g., Bloomberg, Reuters, Yahoo Finance).
* **Topic or Query**: Instead of a URL, users can enter a topic such as "latest stock market trends" or "Tesla earnings report."

#### ****Output:****

* **Chatbot-assisted interactive analysis** of extracted financial data.
* **Summarized insights** on key market trends, economic policies, and stock performance.
* **Detailed reports** based on extracted news, articles, and research papers.

#### ****Features:****

##### **1. Data Extraction from Investment-Related Articles**

* Uses **web scraping and NLP models** to extract content from financial news sources.
* Filters **relevant financial articles, research papers, and investment reports**.
* Categorizes extracted data into topics such as:
  + **Stock Market Trends** (e.g., Bull/Bear Market Indicators).
  + **Macroeconomic Factors** (e.g., Inflation, Interest Rates).
  + **Industry-Specific News** (e.g., Tech, Healthcare, Energy sectors).
  + **Company-Specific News** (e.g., Earnings Reports, Mergers & Acquisitions).

##### **2. AI-Driven Report Generation**

* Summarizes extracted content into **key takeaways, financial implications, and market sentiment**.
* Provides **trend analysis** by comparing historical news coverage with recent developments.
* Identifies **potential investment risks** based on financial news impact.
* Generates **automated PDF reports** summarizing financial insights.

##### **3. Question-Based Interaction for In-Depth Insights**

* Users can **ask questions** about specific stocks, industries, or financial events.
* AI chatbot retrieves **contextually relevant information** from extracted articles.
* Example queries:
  + "What are analysts saying about Apple’s latest earnings?"
  + "How has inflation impacted tech stocks in the last six months?"
  + "Compare the latest financial news on Tesla and Ford."

##### **4. Sentiment and Impact Analysis**

* Performs **Natural Language Processing (NLP)** to determine sentiment (positive, neutral, negative).
* Predicts **potential stock price movements** based on news sentiment and historical patterns.
* Detects **market-moving events**, such as:
  + CEO resignations.
  + Government policy changes.
  + Major acquisitions or lawsuits.

##### **5. Real-Time Alerts & Notifications**

* Users can set alerts for **specific stocks, industries, or financial events**.
* Notifications sent via **email or SMS** when significant updates occur.
* Example: "Alert me if there’s breaking news about the Federal Reserve’s interest rate decision."

##### **6. Export and Data Sharing**

* Users can download extracted data and reports in **Excel, CSV, or PDF format**.
* Allows **integration with financial tools** such as Bloomberg Terminals or trading platforms.

#### ****Tools & Technologies Used:****

* **Google Gemini Multimodal AI**: For text and financial content interpretation.
* **Web Scraping (BeautifulSoup, Scrapy)**: For extracting financial news from web sources.
* **NLP Models (BERT, GPT-based)**: For summarization and sentiment analysis.
* **Machine Learning (Stock Trend Predictions)**: For impact assessment.

**3.1.5 Risk Analysis Report**

**Input:**

* **Annual Financial Report** (PDF, Word, or structured data format).
* Optional: **User-selected focus areas** (e.g., operational risk, financial ratios).

**Output:**

* **Comprehensive Risk Assessment Report** that evaluates financial and operational risks.
* **Breakdown of income statement trends, governance structure, and key financial ratios.**
* **Actionable insights for investors on risk mitigation and financial health assessment.**

**Analysis Components:**

**1. Income Statement Analysis**

Evaluates the company’s revenue generation, expenses, and profitability trends.

* **Revenue Growth Trends**: Identifies patterns in sales and revenue over multiple years.
* **Gross & Net Profit Margins**: Assesses the company’s ability to convert revenue into profit.
* **Expense Breakdown**: Analyzes cost structure, including operational and administrative expenses.
* **Earnings Before Interest & Taxes (EBIT) Analysis**: Determines operational efficiency.
* **Cash Flow Trends**: Examines if earnings are sustainable and cash flow is stable.
* **Year-over-Year (YoY) Comparisons**: Highlights inconsistencies in financial growth.

**2. Ratio Analysis**

Provides a financial health check using key performance indicators.

* **Debt-to-Equity (D/E) Ratio**: Evaluates financial leverage and debt burden.
* **Price-to-Earnings (P/E) Ratio**: Assesses whether the stock is undervalued or overvalued.
* **Return on Equity (ROE) & Return on Assets (ROA)**: Measures profitability against shareholders' equity and assets.
* **Current Ratio & Quick Ratio**: Determines short-term liquidity and ability to cover liabilities.
* **Earnings Per Share (EPS) Growth**: Tracks profitability from an investor’s perspective.
* **Interest Coverage Ratio**: Evaluates a company’s ability to manage its debt obligations.

**3. Operational Risks**

Assesses internal and external risks affecting the company’s ability to sustain operations.

* **Supply Chain Vulnerabilities**: Identifies dependencies on key suppliers and regions.
* **Production & Manufacturing Risks**: Detects bottlenecks that may impact output.
* **Workforce & Labor Concerns**: Evaluates labor strikes, talent retention, and hiring trends.
* **Market Positioning Risks**: Examines competitive landscape and threats from emerging players.
* **R&D Investment Risks**: Analyzes whether innovation spending aligns with industry standards.

**4. Market Risks**

Evaluates external factors that could impact the company's financial performance.

* **Stock Price Volatility**: Measures historical price fluctuations and trends.
* **Macroeconomic Indicators**: Examines inflation, GDP growth, interest rates, and global economic conditions.
* **Foreign Exchange (Forex) Risks**: Assesses exposure to currency fluctuations for multinational firms.
* **Regulatory & Policy Risks**: Identifies government policies or trade restrictions that may impact operations.
* **Commodity Price Dependencies**: For industries reliant on raw materials (e.g., oil, gold, or semiconductors).

**5. Governance Risks**

Analyzes corporate leadership, regulatory compliance, and ethical business practices.

* **Board Composition & Transparency**: Reviews diversity, independence, and structure of the board.
* **Regulatory Compliance & Legal Exposure**: Checks for ongoing lawsuits, fines, or non-compliance risks.
* **Executive Compensation & Incentive Alignment**: Examines if management’s interests align with shareholders.
* **Shareholder Rights & Activism**: Identifies risks of hostile takeovers or shareholder disputes.
* **ESG (Environmental, Social, and Governance) Factors**: Evaluates corporate responsibility and sustainability initiatives.

**6. Report Customization & Export**

* Generates risk assessment reports in **PDF, Excel, and interactive dashboards.**
* Allows users to filter risk categories and prioritize areas of concern.
* Provides **AI-generated recommendations** for mitigating identified risks.

**Tools & Technologies Used:**

* **Financial Statement Parsing (OCR & NLP)**: Extracts structured data from financial reports.
* **Google Gemini AI**: Analyzes financial documents, trends, and risks.
* **Machine Learning Models**: Predicts stock volatility and operational risks.
* **Data Visualization (Seaborn, Matplotlib)**: Generates risk graphs and comparative charts.  
    
  RISK ASSESMENT REPORT
* Overview Of Company
* This is the 2023 Annual Report for Gul Ahmed. Gul Ahmed is a textile company. The report includes financial statements (consolidated and un-consolidated), a directors' report, financial highlights, and other information such as a sustainability report, details on corporate governance, and stakeholder engagement.
* Income Statement Analysis
* Revenue:
* The provided text gives sales figures for several quarters and years, but not consistently across three full years in a single table. Therefore, a precise answer to your question requires piecing together information from different sections. I will present what I can find, but it's not a complete three-year picture.
* \*Sales Data (Partial):\*
* \* \*2023:\*
* \* Q3: Rs. 29,428 million
* \* Total for the year: Implied by the sum of direct and indirect export sales and local sales. Direct export sales: Rs. 72,941 million; Indirect export sales: Rs. 34,035 million; Local sales: Rs. 14,473 million. Therefore, total sales for 2023 is approximately Rs. 121,449 million.
* \* \*2022:\*
* \* Direct export sales: Rs. 56,299 million
* \* Indirect export sales: Rs. 41,818 million
* \* Local sales: Rs. 11,357 million. Total sales for 2022 is approximately Rs. 109,474 million.
* \* \*2021:\* Sales figures are not explicitly stated for 2021.
* \*Analysis:\*
* Based on the available data, total sales in 2023 (approximately Rs. 121,449 million) were higher than in 2022 (approximately Rs. 109,474 million). However, a complete three-year trend cannot be determined due to the missing 2021 data.
* \*Impact on Investors:\*
* An increase in revenue is generally positive for investors as it suggests the company is growing and potentially more profitable. However, the analysis is incomplete without the 2021 data. Investors would also need to consider other factors such as profitability (profit margins), expenses, and overall financial health before making investment decisions. The text mentions that despite increased revenue, profit margins decreased in 2023 due to challenges like raw material price increases and higher finance costs. This is crucial information for investors to assess the true impact of the revenue increase.
* COCS (Cost of good sold ):
* The provided text only gives the Cost of Goods Sold (COGS) for the year ending June 2023 and June 2022. Data for a third year is not available.
* \*COGS:\*
* \* \*June 2023:\* Rs. 110,746,798,000
* \* \*June 2022:\* Rs. 94,800,552,000
* \*Analysis:\*
* The COGS increased from Rs. 94,800,552,000 in June 2022 to Rs. 110,746,798,000 in June 2023. This represents a substantial increase.
* \*Impact on Investors:\*
* An increase in COGS, without a corresponding increase in sales revenue at a faster rate, reduces the gross profit margin. This is negative for investors because it indicates lower profitability. Investors may react negatively, potentially leading to a decrease in the company's stock price. A detailed analysis would require examining the sales figures for both periods to determine the impact on the gross profit margin and overall profitability. The annual report mentions that the gross profit margin decreased from 17.32% to 14.89%, supporting this negative impact on profitability.
* Gross profit:
* Here's the gross profit data for the past three years, extracted from the provided annual report, followed by a short analysis:
* \*Gross Profit (Rs. in Million)\*
* | Year | Gross Profit |
* |---|---|
* | 2023 | 16,678,982 |
* | 2022 | 17,367,670 |
* | 2021 | 12,868,608 |
* \*Analysis:\*
* The gross profit shows a fluctuating trend over the past three years. There was a significant increase from 2021 to 2022 (Rs. 4,499,062 million), followed by a decrease from 2022 to 2023 (Rs. 688,688 million).
* \*Impact on Investors:\*
* \* \*Decreasing Gross Profit:\* A decrease in gross profit, as seen from 2022 to 2023, generally signals reduced profitability. This could lead to lower earnings per share (EPS), potentially causing a negative impact on investor sentiment and share price. Investors might become concerned about the company's ability to maintain its growth trajectory and dividend payouts. The reasons for the decrease (e.g., rising raw material costs) would be carefully considered by investors.
* \* \*Increasing Gross Profit:\* An increase in gross profit, as seen from 2021 to 2022, is usually viewed positively by investors. It suggests improved efficiency in operations or successful pricing strategies. This can lead to higher EPS, increased investor confidence, and potentially higher share prices. However, investors would still want to understand the sustainability of this increase.
* \*Important Note:\* A complete investment decision should not be based solely on gross profit. Investors should consider a broader range of financial metrics (net profit, operating cash flow, debt levels, etc.) and qualitative factors (market conditions, competitive landscape, management quality) before making any investment decisions.
* Total Operating Expence:
* The provided text gives a Profit & Loss account summary for only two years, not three. Therefore, I cannot calculate the total operating expenses for the past three years.
* \*Operating Expenses for June 2023 and June 2022:\*
* To calculate operating expenses, we need to sum up the distribution expenses, administrative expenses, and other expenses from the Profit & Loss account.
* \* \*June 2023:\* 2,200,949 + 3,428,329 + 432,181 = Rs. 6,061,459,000
* \* \*June 2022:\* 1,957,801 + 2,220,600 + 860,379 = Rs. 5,038,780,000
* \*Analysis:\*
* Operating expenses increased from Rs. 5,038,780,000 in June 2022 to Rs. 6,061,459,000 in June 2023. This is a significant rise.
* \*Effect on Investors:\*
* An increase in operating expenses, without a corresponding increase in revenue, reduces profitability (profit margins). This negatively impacts investors because:
* \* \*Lower Earnings:\* Reduced profits mean lower earnings per share (EPS), potentially leading to lower dividend payouts or reduced share price appreciation.
* \* \*Reduced Return on Investment (ROI):\* Investors' return on their investment diminishes if the company's profitability suffers.
* \* \*Increased Risk:\* Higher operating expenses can indicate operational inefficiencies or unsustainable business practices, increasing the risk of future losses. Investors may react by lowering their valuation of the company's stock.
* To provide a complete analysis, information on revenue and net profit for the three years would be needed. The provided data only allows for a partial analysis of the two years presented.
* Operating Profit:
* Here's the operating profit for the past three years, extracted from the provided text, along with a short analysis:
* \*Operating Profit (Rs. in Million):\*
* \* \*2023:\* 11,296.897
* \* \*2022:\* 13,137.979
* \* \*2021:\* 7,505.847
* \*Analysis:\*
* The operating profit shows a decrease from 2022 to 2023 (13,137.979 to 11,296.897 million). However, there was a significant increase from 2021 to 2022 (7,505.847 to 13,137.979 million). Overall, the trend over the three-year period is not consistently increasing or decreasing.
* \*Effect on Investors:\*
* Decreases in operating profit, like that seen between 2022 and 2023, are generally viewed negatively by investors. This suggests reduced profitability and potentially weaker financial health, which can lead to lower stock prices and reduced investor confidence. Conversely, increases in operating profit, such as the one from 2021 to 2022, are positive signals. They indicate improved efficiency and potentially higher future returns, which can attract investors and boost stock prices. The inconsistent trend shown here would likely cause some investor concern and require further investigation into the underlying causes of the fluctuations.
* Net Profit:
* Here's the net profit data for the past three years, extracted from the provided annual report, followed by a short analysis:
* \*Net Profit (Rs. in '000):\*
* \* \*2023:\* 3,986,019
* \* \*2022:\* 8,861,647
* \* \*2021:\* 4,424,540
* \*Analysis:\*
* The net profit shows a fluctuating trend over the past three years. There was a significant increase from 2021 to 2022, followed by a substantial decrease from 2022 to 2023.
* \*Effect on Investors:\*
* \* \*Decreasing Net Profit (2022-2023):\* A decrease in net profit generally negatively impacts investor confidence. It can lead to lower dividend payouts, reduced share price, and potentially decreased investor interest in the company. Investors may perceive higher risk and lower potential returns.
* \* \*Increasing Net Profit (2021-2022):\* An increase in net profit is generally positive for investors. It suggests the company is performing well, potentially leading to higher dividends, increased share price, and greater investor confidence. Investors may see higher potential returns and lower risk.
* \* \*Fluctuating Net Profit:\* The inconsistent performance over the three years introduces uncertainty for investors. They may be hesitant to invest heavily in a company with unpredictable profitability. A thorough understanding of the reasons behind the fluctuations (as explained in the report's discussion of raw material costs, interest rates, etc.) is crucial for investors to make informed decisions.
* Ratio Analysis
* Liquidity Ratio Analysis
* Current Ratio and Quick Ratio
* Here's the information on the Current Ratio and Quick Ratio for the past three years, along with a short analysis based on the provided data:
* \*Gul Ahmed Textile Mills Limited: Current and Quick Ratios (2021-2023)\*
* | Year | Current Ratio | Quick/Acid Test Ratio |
* | -------- | -------------- | --------------------- |
* | 2023 | 1.11 | 0.47 |
* | 2022 | 1.15 | 0.55 |
* | 2021 | 1.12 | 0.57 |
* \*Analysis:\*
* Both the current ratio and the quick ratio show a slightly decreasing trend over the past three years. The current ratio has fallen from 1.15 in 2022 to 1.11 in 2023, while the quick ratio has decreased from 0.57 in 2021 to 0.47 in 2023.
* \* \*Effect on Investors:\* A decreasing current ratio suggests a decline in the company's short-term liquidity. This means the company may have less ability to meet its immediate obligations. Investors might view this negatively, potentially leading to a decrease in investor confidence and share price. A lower quick ratio, which excludes inventories from current assets, further strengthens this concern as it indicates a potentially more precarious short-term liquidity position.
* \* \*Comparison to Averages:\* To make a meaningful comparison to averages, we need the average current ratio and average quick ratio for a relevant benchmark (e.g., industry average, competitor averages). This data is not provided in the given report, preventing a direct comparison. Without this benchmark, we cannot definitively say whether Gul Ahmed's ratios are above or below the average.
* \*Important Note:\* The analysis is based solely on the provided data. A comprehensive analysis would require additional information, such as industry benchmarks and a deeper understanding of the company's specific circumstances (e.g., changes in inventory management, credit policies, etc.).
* Laverage Ratio Analysis
* Debt-to-Equity Ratio and Interest Coverage Ratio:
* The provided text gives Debt-to-Equity Ratio and Interest Coverage Ratio for the past three years, but it lacks data for a meaningful comparison with industry averages. Therefore, I cannot provide a comparison of the company's ratios to industry averages.
* Here's the information available from the report:
* \*Debt-to-Equity Ratio & Interest Coverage Ratio (Past 3 Years):\*
* | Year | Debt-to-Equity Ratio (%) | Interest Coverage Ratio (Times) |
* |---|---|---|
* | 2023 | 33 | 2.11 |
* | 2022 | 36 | 4.92 |
* | 2021 | 41 | 3.86 |
* \*Short Analysis:\*
* \* \*Debt-to-Equity Ratio:\* This ratio has decreased from 41% in 2021 to 33% in 2023. A lower debt-to-equity ratio generally indicates improved financial health and reduced risk. The decrease suggests the company has become less reliant on debt financing relative to equity.
* \* \*Interest Coverage Ratio:\* This ratio has decreased from 4.92 times in 2022 to 2.11 times in 2023. A lower interest coverage ratio indicates a reduced ability to meet interest obligations from earnings. The decline is a cause for concern, suggesting increased financial risk.
* \*Effect on Investors:\*
* \* \*Decreasing Debt-to-Equity Ratio:\* A lower debt-to-equity ratio is generally viewed positively by investors as it suggests lower financial risk. It implies a stronger ability to withstand economic downturns and meet financial obligations. This could lead to increased investor confidence and potentially higher share prices.
* \* \*Decreasing Interest Coverage Ratio:\* A decreasing interest coverage ratio is a negative signal for investors. It indicates increased financial risk and a higher probability of default on debt obligations. This could lead to decreased investor confidence and potentially lower share prices. Investors may demand a higher return to compensate for the increased risk.
* \*Missing Information:\*
* To complete a thorough analysis, we need the average Debt-to-Equity Ratio and average Interest Coverage Ratio for the industry in which Gul Ahmed operates. This would allow for a comparison to see if the company's performance is better or worse than its peers. This information is not provided in the
* Investor Profitibility Ratio
* Net Profit Margin, ROE, ROA, and EPS:
* The provided annual report gives data for profitability ratios and investor information for six years (2018-2023), not just three. I will use the last three years (2021-2023) to answer your question. ROA (Return on Assets) is not explicitly provided, so I cannot calculate it.
* \*Financial Ratios Trends (2021-2023):\*
* | Ratio | 2023 | 2022 | 2021 | Trend | Analysis |
* |-----------------|-----------|-----------|-----------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
* | Net Profit Margin | 3.56% | 8.84% | 5.62% | Decreasing | A significant decrease in net profit margin, especially from 2022 to 2023, indicates declining profitability. This is likely due to increased costs that couldn't be passed on. |
* | ROE | 10.46% | 27.96% | 21.46% | Decreasing | A sharp decline in ROE reflects reduced returns on shareholder investments. This is a major concern for investors. |
* | EPS | Rs. 5.39 | Rs. 11.97 | Rs. 7.83 | Decreasing | Lower EPS means less profit distributed per share, directly impacting investor dividends and share value. |
* \*Effect on Investors:\*
* \* \*Decreasing Net Profit Margin:\* Investors will be concerned about the company's ability to generate profits. A shrinking margin suggests the company is struggling to control costs or increase prices, potentially leading to lower returns on their investments.
* \* \*Decreasing ROE:\* This is a significant negative signal. It indicates that the company is not efficiently using shareholder equity to generate profits. Investors may lose confidence and seek higher returns elsewhere.
* \* \*Decreasing EPS:\* Lower EPS translates to lower potential dividends and a potentially lower share price. This directly impacts investor returns and may lead to selling pressure, driving the share price down further.
* \*In summary:\* The trends in net profit margin, ROE, and EPS over the past
* Major Cash Flow Risks:
* The annual report identifies several major cash flow risks:
* \* \*Interest/Mark-up Rate Risk:\* Fluctuations in interest and markup rates on financial instruments (both assets and liabilities) impact the company's cash flows. A 1% increase in markup on long-term financing in 2023 would have lowered post-tax profit by Rs. 232.1 million, while a similar increase on short-term borrowings would have lowered it by Rs. 302.05 million. The report notes that the company uses prepayment and rollover options to mitigate this risk.
* \* \*Currency Risk:\* Fluctuations in foreign exchange rates affect the cash flows from trade debts, cash and bank balances, short-term borrowings, and trade and other payables. The company manages this risk through monitoring exchange rates, adjusting net exposure, and using forward covers.
* \* \*Price Risk of Raw Materials:\* Fluctuations in cotton and yarn prices, changes in import policies, utility rate shifts, and markup rate changes all impact the cost of raw materials and thus affect cash flows. The company mitigates this through stock management, working capital optimization, and exchange rate risk management techniques (forward contracts, bill discounting, and FCY credit).
* \* \*Geopolitical Risk impacting Export Sales:\* A decline in export sales due to geopolitical instability or reduced global demand poses a risk to cash flow. The company uses subsidiary offices, resource optimization, and risk managers (internal and external) to mitigate this.
* \* \*Depletion of Natural Gas Resources and Fluctuating Fuel Prices:\* Pakistan's reliance on imported fuel for thermal electric generation creates vulnerability to price fluctuations and supply disruptions. The company is mitigating this by acquiring tri-fuel generating sets and transitioning to renewable energy.
* \* \*Economic Policy Changes in Pakistan:\* Changes in duties, taxes, and tax rates directly impact material costs, finished goods pricing, divisible profits, and cash outflows. The company addresses this by closely monitoring government economic strategy and adjusting its profit and cash flow forecasts accordingly.
* Operational Risks
* The annual report mentions several key operational risks:
* \* \*Credit Risk:\* Risk of default on receivables, primarily from local sales. Mitigated through due diligence and security measures for credit extensions.
* \* \*Market Risk:\* Fluctuations in raw material prices (cotton, yarn), import policy changes, utility rate shifts, and markup rate changes. Mitigated through stock management, working capital optimization, and using financial instruments (forward contracts, bill discounting, FCY credit) for exchange rate risk. Also includes the risk of declining export sales due to geopolitical risks and reduced global demand. Mitigated through subsidiary offices, resource optimization, and engagement of risk managers.
* \* \*Liquidity Risk:\* Risk of difficulty meeting financial obligations when due. Mitigated by maintaining sufficient cash and credit facilities. The report states that management believes this risk is insignificant at the reporting date.
* \* \*Technology Risk:\* The report highlights the company's commitment to staying technologically advanced, but doesn't explicitly state this as a risk. However, the reliance on technology implies a risk of system failures or cyberattacks. The report mentions cybersecurity measures implemented to mitigate this.
* \* \*Economic Policy Risk:\* Operating in Pakistan's dynamic economic environment presents challenges due to changing economic policies, new taxes, and fluctuating tax rates. The company mitigates this by monitoring government strategies and adjusting profit and cash flow forecasts accordingly.
* \* \*Utilities Availability and Prices Risk:\* Depletion of Pakistan's natural gas resources and reliance on imported fuel for thermal electric generation lead to supply disruptions and price volatility. Mitigated by acquiring tri-fuel generating sets and transitioning to renewable energy.
* The report also mentions other threats, such as security issues, geopolitical risks, competition, employee attrition, and raw material shortages, but these are presented more broadly as threats to the business rather than specifically defined operational risks within the risk management section.
* Major Market Risks
* Based on the provided annual report, the major market risks affecting Gul Ahmed are:
* \* \*Fluctuations in raw material prices:\* Specifically, cotton and yarn prices are cited as significant concerns.
* \* \*Changes in import policies and restrictions:\* The company is vulnerable to shifts in government regulations impacting its import of raw materials.
* \* \*Shifts in utility rates:\* Changes in energy costs pose a risk to profitability.
* \* \*Fluctuations in markup rates:\* Changes in interest rates affect the company's financing costs.
* \* \*Exchange rate risk:\* Fluctuations in foreign exchange rates impact the company's export sales and transactions in foreign currencies.
* \* \*Decline in export sales due to geopolitical risk:\* Geopolitical instability and reduced global demand can negatively affect export revenue.
* The company uses various strategies to mitigate these risks, including stock management, working capital optimization, exchange rate risk management (forward contracts, bill discounting, FCY credit), maintaining subsidiary offices in strategic locations, resource optimization, and engaging risk managers.
* Governance And Regulatory Risks:
* The provided text mentions several governance and regulatory risks, although not explicitly labeled as such:
* \*Governance Risks:\*
* \* \*Compliance with Regulations:\* The report highlights the auditors' responsibility to review the company's compliance with regulations. While the auditors found no material non-compliance, the existence of this review process itself indicates a governance risk related to potential regulatory breaches. Specific regulations (3, 6, 7, 8, 27, 32, 33, and 36) are mentioned as having been complied with, implying the potential for non-compliance with others.
* \* \*Related Party Transactions:\* The report notes that the company's related party transactions are reviewed and approved by the Audit Committee and Board of Directors. This process, while in place, represents a governance risk if not properly implemented or if transactions are not fully disclosed or appropriately valued.
* \* \*Internal Controls:\* The report discusses the company's internal control system, including its outsourcing of the internal audit function. While the system is described as comprehensive, the reliance on an external entity for internal audit introduces a governance risk related to potential conflicts of interest or limitations in the scope of the external auditor's review. The report explicitly states that the auditors are not required to form an opinion on the effectiveness of internal controls.
* \* \*Risk Management Committee:\* The report notes that the company chose not to form a separate Risk Management Committee, opting instead for management-level oversight. This decision, while permissible, represents a governance risk due to the potential for conflicts of interest or insufficient independent oversight of risk management activities. Similarly, the absence of a Nomination Committee is noted.
* \*Regulatory Risks:\*
* The report doesn't explicitly list regulatory risks, but the following can be inferred:
* \* \*Financial Reporting Standards:\* The report emphasizes the management's responsibility for preparing financial statements in accordance with applicable accounting and reporting standards. Non-compliance with these standards would represent a significant regulatory risk.
* \* \*Companies Act, 2017:\* The report mentions the Companies Act, 2017 as a regulatory framework for financial reporting. Failure to comply with this act would constitute a regulatory risk.
* \* \*Unspecified Regulations:\* The report mentions compliance with specific regulations (3, 6, 7, 8, 27, 32, 33, and 36) but doesn't specify what

**3.1.6 Competitor Analysis**

**Input:**

* **Two Companies**: Users enter stock tickers or company names for comparison.
* **Time Frame Selection**: Users can specify a time period for historical analysis (e.g., past year, past five years).
* **Report Customization**: Users can choose specific metrics for comparison, such as financial ratios, market performance, or risk factors.

**Output:**

* **Comprehensive Competitive Risk Analysis Report** comparing financial stability, market positioning, and risk exposure of two selected companies.
* **Visual Comparisons** using graphs, tables, and AI-generated insights.
* **Key Strengths & Weaknesses** of each competitor, helping investors make informed decisions.

**Features:**

**1. Automated Benchmarking of Financial Health**

* Compares **key financial ratios and performance metrics** to evaluate overall company stability.
* Uses machine learning models to **rank companies based on profitability, liquidity, and leverage.**
* Benchmarks against **industry standards** and sector-specific competitors.
* **Metrics Analyzed:**
  + Revenue Growth Rate
  + Gross & Net Profit Margins
  + Earnings Per Share (EPS)
  + Return on Equity (ROE) & Return on Assets (ROA)
  + Debt-to-Equity Ratio (D/E)
  + Free Cash Flow & Liquidity Ratios

**2. Competitive Positioning Based on Market Trends**

* **Stock Performance Comparison**: Tracks historical stock price movements, volatility, and investor sentiment.
* **Market Capitalization & Valuation Analysis**: Compares Price-to-Earnings (P/E) and Price-to-Book (P/B) ratios.
* **Macroeconomic & Industry Trend Analysis**: Evaluates external factors influencing both companies.
* **Innovation & R&D Spending**: Assesses whether companies are investing in future growth.
* **Customer Sentiment & Brand Perception**: Uses AI-driven sentiment analysis on financial news and social media.

**3. Side-by-Side Comparison of Risk Categories**

* **Financial Risk**: Evaluates exposure to debt, liquidity constraints, and credit risks.
* **Operational Risk**: Compares supply chain resilience, manufacturing efficiency, and workforce stability.
* **Market Risk**: Analyzes stock price volatility, competition intensity, and global economic exposure.
* **Regulatory & Governance Risk**: Compares legal compliance, board structure, and corporate transparency.
* **ESG (Environmental, Social, Governance) Factors**: Reviews sustainability initiatives and ethical business practices.

**4. Advanced Data Visualization & Export Options**

* Generates **interactive dashboards** for financial and risk comparison.
* Provides **heatmaps, SWOT analysis (Strengths, Weaknesses, Opportunities, Threats), and AI-generated insights.**
* **Exportable Reports** in PDF, Excel, and PowerPoint formats for investor presentations.

**Tools & Technologies Used:**

* **Financial Data APIs (Yahoo Finance, Bloomberg, Alpha Vantage)**: For real-time data retrieval.
* **Google Gemini AI & NLP Models**: For automated risk assessment and sentiment analysis.
* **Machine Learning Algorithms (Random Forest, Gradient Boosting)**: For financial benchmarking.
* **Data Visualization Tools (Seaborn, Matplotlib, Tableau)**: For graphical representation.

**3.2 Non-functional Requirements**

* **Performance**: Data retrieval and visualization within 5 seconds.
* **Security**: End-to-end encryption for financial data.
* **Scalability**: Supports large datasets and multiple concurrent users.
* **Compliance**: Adheres to financial data protection laws.
* **Availability**: 99.9% uptime guarantee.

## ****Functional Description****

### ****Overview****

The **Functional Description** of **Insight Invest** provides a detailed breakdown of how the system operates, including its core services, data processing mechanisms, user interactions, and expected outputs. This section outlines the features and functionalities of the system, ensuring that users understand how it works, what it offers, and how it integrates with different financial data sources.

### ****System Workflow****

The **Insight Invest** platform functions as an AI-powered **equity research analyst**, automating the process of data collection, financial analysis, and risk assessment. The system follows these major steps:

1. **User Input Collection**
   * Users enter stock tickers, date ranges, financial reports, or company names based on the service they need.
   * The system accepts various input formats, including text, URLs, PDFs, and Excel files.
2. **Data Retrieval and Processing**
   * The system fetches real-time and historical financial data from sources like **Yahoo Finance, News API, and financial databases.**
   * Large Language Models (LLMs) and **Retrieval-Augmented Generation (RAG)** techniques analyze financial reports and web articles.
   * Machine learning algorithms process historical data for **predictive modeling and risk assessment.**
3. **Analysis & Computation**
   * AI-driven models analyze data, generate insights, and evaluate **financial risks, stock trends, and competitor benchmarks.**
   * Sentiment analysis techniques assess the **impact of financial news on stock performance.**
   * Multimodal AI models **interpret financial reports, graphs, and tables.**
4. **Output Generation & Visualization**
   * The system **generates reports, risk assessments, and visual analytics** in an easy-to-understand format.
   * Users receive outputs in **interactive dashboards, downloadable reports (PDF, Excel), or chatbot-driven summaries.**

### ****Core Functionalities****

#### ****1. Data Visualization****

* Allows users to **view graphical representations of stock data.**
* Users input a **stock ticker and date range**, and the system fetches **historical price trends, moving averages, RSI, and MACD indicators.**
* **LSTM-based predictive models** forecast future stock trends.
* **Tools Used:** Yahoo Finance API, Seaborn (Python), LSTM models.

#### ****2. News Analysis****

* Aggregates stock-related news based on **company ticker symbols.**
* Users can:
  1. **Download news reports** in an Excel format.
  2. **Get AI-generated summaries** for quick insights.
* **Sentiment Analysis** determines the **positive/negative impact of news on stock performance.**
* **Tools Used:** News API, NLP (Natural Language Processing) for text summarization.

#### ****3. RAG on Annual Financial Reports****

* Users upload **annual reports (PDFs, text, Excel)** for AI-powered analysis.
* The system **parses income statements, balance sheets, and financial graphs.**
* AI chatbot assists in answering **questions about revenue, profits, and risk factors.**
* **Tools Used:** Google Gemini multimodal AI, OCR (Optical Character Recognition), RAG models.

#### ****4. RAG on Finance Websites****

* Users provide a **finance website URL or specific topic** (e.g., "Apple stock trends").
* AI chatbot extracts and analyzes **relevant financial data from articles and reports.**
* Provides **summarized insights and trend predictions.**
* **Tools Used:** Web scraping tools, NLP models for text analysis.

#### ****5. Risk Analysis Report****

* Users upload **financial reports**, and the system generates a **detailed risk assessment.**
* Evaluates:
  + **Financial risks (debt, liquidity, profitability).**
  + **Operational risks (supply chain, workforce).**
  + **Market risks (economic factors, industry trends).**
  + **Governance risks (corporate transparency, regulatory compliance).**
* **Tools Used:** Machine learning models, Google Gemini AI, financial ratio calculators.

#### ****6. Competitor Analysis****

* Users select **two companies** for side-by-side financial comparison.
* The system generates a **benchmarking report covering stock performance, risk exposure, and market positioning.**
* **Graphical dashboards display financial ratios, stock trends, and SWOT analysis.**
* **Tools Used:** Yahoo Finance API, machine learning models for comparative analysis.

### ****System Integration & APIs****

The **Insight Invest** platform integrates with several external and internal services:

* **Financial Data APIs**: Yahoo Finance, Alpha Vantage, Bloomberg API.
* **News Aggregation APIs**: News API, Google News Scraper.
* **Machine Learning Frameworks**: TensorFlow, PyTorch for LSTM-based predictions.
* **Natural Language Processing (NLP)**: Google Gemini AI for text summarization and chatbot functionality.
* **Data Visualization Tools**: Seaborn, Matplotlib, Power BI.

### ****User Interaction & Interface****

* **Web-Based Portal:** Users can access Insight Invest via a **web dashboard with interactive reports and chatbots.**
* **Chatbot Interface:** AI-powered assistant for **answering investment-related queries.**
* **Export Options:** Users can **download reports in Excel, PDF, and PowerPoint formats.**
* **Mobile Compatibility:** Designed for use on **desktops, tablets, and smartphones.**

### ****Performance & Scalability****

* **Real-Time Processing:** Fetches stock data within **seconds** of user input.
* **Scalability:** Supports multiple concurrent users and **large datasets** for enterprise use.
* **Security:** Uses **end-to-end encryption** for financial transactions and data privacy compliance.

### ****Error Handling & Limitations****

* **API Failures:** If financial APIs are down, the system notifies the user and **retrieves cached data** if available.
* **Data Inconsistencies:** The system validates data sources and **flags discrepancies in financial reports.**
* **Processing Limits:** Large datasets might require additional **processing time or cloud resources.**

## ****Features Implemented Description****

### ****Overview****

Insight Invest is an AI-powered **equity research analyst tool** that automates stock market research, financial report analysis, and risk assessment. It integrates multiple AI-driven functionalities to provide **real-time financial insights, predictive analysis, and risk evaluation** for investors, financial analysts, and portfolio managers. This section provides a **detailed breakdown of the implemented features, their functionalities, and how they contribute to investment decision-making.**

## ****1. Data Visualization****

**Description:**  
The **Data Visualization module** provides investors and analysts with **interactive graphical insights into stock trends and financial performance.** Instead of manually analyzing large financial datasets, users can enter a **stock ticker and date range** to receive an automated **visual representation of key stock metrics.**

**Key Functionalities:**  
✔ **Historical Stock Data:** Fetches **open, close, high, and low stock prices** over the selected date range.  
✔ **Technical Indicators:**

* **Moving Averages (SMA & EMA):** Helps identify market trends.
* **Relative Strength Index (RSI):** Determines stock momentum and overbought/oversold conditions.
* **MACD (Moving Average Convergence Divergence):** Signals potential trend reversals.  
  ✔ **LSTM-Based Predictive Modeling:** Uses AI to **forecast future stock movements** based on historical trends.  
  ✔ **Data Export Options:** Users can download **graphs and stock trend reports in PDF/Excel format.**

**Tools & Technologies Used:**  
📌 Yahoo Finance API, Seaborn, Matplotlib, LSTM neural networks, Python.

## ****2. News Analysis****

**Description:**  
Stock prices are highly sensitive to **news events, earnings reports, mergers, and global economic updates.** The **News Analysis feature** scans financial news articles, applies **AI-driven sentiment analysis**, and provides users with **summarized insights** to gauge a company’s stock volatility.

**Key Functionalities:**  
✔ **Automated News Fetching:** Users enter a **stock ticker**, and the system retrieves **relevant financial news articles.**  
✔ **Sentiment Analysis:** AI classifies news as **positive, negative, or neutral** to assess its impact on stock movement.  
✔ **Keyword & Event Extraction:** Highlights **critical events** such as **lawsuits, earnings reports, acquisitions, and CEO resignations.**  
✔ **Download Options:** Users can:

* **Download news articles in Excel format.**
* **Get AI-generated summaries of important news events.**

**Tools & Technologies Used:**  
📌 News API, Natural Language Processing (NLP), Sentiment Analysis Models.

## ****3. Retrieval-Augmented Generation (RAG) on Annual Reports****

**Description:**  
Annual financial reports contain **detailed company performance metrics**, but analyzing them manually is time-consuming. The **RAG-based chatbot** provides **an AI-driven interactive way to analyze annual reports** by extracting key insights from **text, tables, and financial graphs.**

**Key Functionalities:**  
✔ **Automated Report Parsing:** Users upload an **Annual Report (PDF/Excel)**, and the system extracts **key financial metrics, earnings summaries, and risk factors.**  
✔ **Multimodal AI Analysis:**

* **Text Processing:** Analyzes **income statements, balance sheets, and cash flow reports.**
* **Graph & Table Interpretation:** Uses **AI to read and interpret tables, bar charts, and financial figures.**  
  ✔ **Interactive Chatbot Assistance:** Users can **ask questions about specific sections of the report**, and the chatbot provides **detailed explanations and insights.**  
  ✔ **Financial Metric Comparisons:** The system automatically **compares multiple years of financial data** for trend analysis.

**Tools & Technologies Used:**  
📌 Google Gemini AI, RAG models, Optical Character Recognition (OCR), NLP, Large Language Models (LLMs).

## ****4. Retrieval-Augmented Generation (RAG) on Finance Websites****

**Description:**  
Financial articles, reports, and investment blogs contain valuable insights, but **manually reading and analyzing them is inefficient.** This feature allows users to **input a website URL or topic**, and the AI-powered chatbot **extracts key financial insights from the content.**

**Key Functionalities:**  
✔ **Automated Web Scraping:** The system scans finance websites for **relevant stock analysis, expert opinions, and company performance updates.**  
✔ **AI-Driven Summary Generation:**

* Extracts **investment recommendations, risk factors, and market outlooks.**
* Highlights **key financial indicators mentioned in the articles.**  
  ✔ **Chatbot for Quick Analysis:** Users can ask **"What is the revenue forecast for Apple?"** and receive **AI-generated responses** from multiple sources.  
  ✔ **Data Export Options:** Download reports in **Excel, PDF, or text format.**

**Tools & Technologies Used:**  
📌 Google Gemini AI, Web Scraping Libraries (BeautifulSoup, Scrapy), NLP Models.

## ****5. Risk Analysis Report****

**Description:**  
The **Risk Analysis Report** evaluates the financial stability of a company based on its **annual reports, stock trends, and macroeconomic factors.** Investors can use this report to identify potential **investment risks before making financial decisions.**

**Key Functionalities:**  
✔ **Comprehensive Risk Assessment** of a company based on **five major factors:**

1. **Income Statement Analysis:** Evaluates revenue growth, net profit margins, and cost structures.
2. **Ratio Analysis:**
   * Debt-to-Equity Ratio (D/E)
   * Price-to-Earnings Ratio (P/E)
   * Liquidity Ratios (Current Ratio, Quick Ratio)
3. **Operational Risks:** Analyzes supply chain vulnerabilities, business continuity, and industry competition.
4. **Market Risks:** Tracks **stock volatility, economic downturn risks, and sector-specific risks.**
5. **Governance & Compliance Risks:** Reviews **board transparency, executive decisions, and regulatory adherence.**  
   ✔ **AI-Powered Risk Scoring:** Generates a **risk score for the company** to help investors quickly understand **potential vulnerabilities.**  
   ✔ **Interactive Graphs & Heatmaps:** Visually represents **high-risk areas in a company’s financials.**

**Tools & Technologies Used:**  
📌 Machine Learning Risk Models, Financial APIs, Google Gemini AI, Power BI.

## ****6. Competitor Analysis****

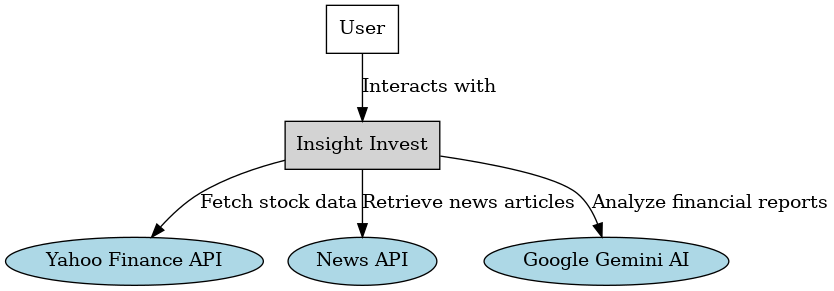
**Description:**  
Understanding how a company compares to its **industry rivals** is crucial for investors. The **Competitor Analysis feature** generates a **side-by-side comparison** of two companies' financial health, risks, and stock performance.

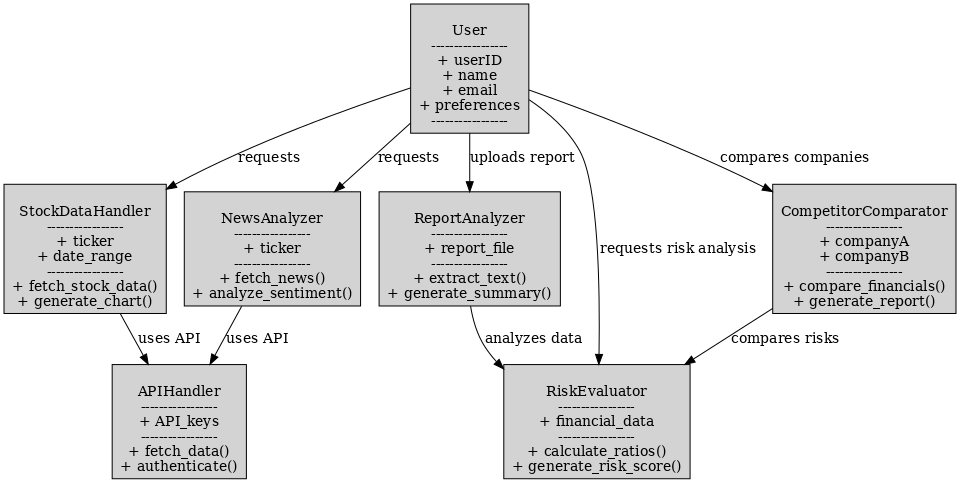
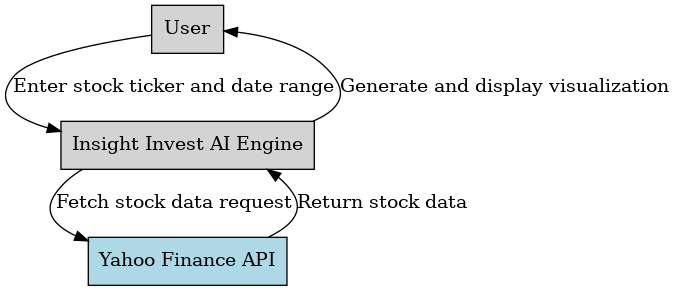
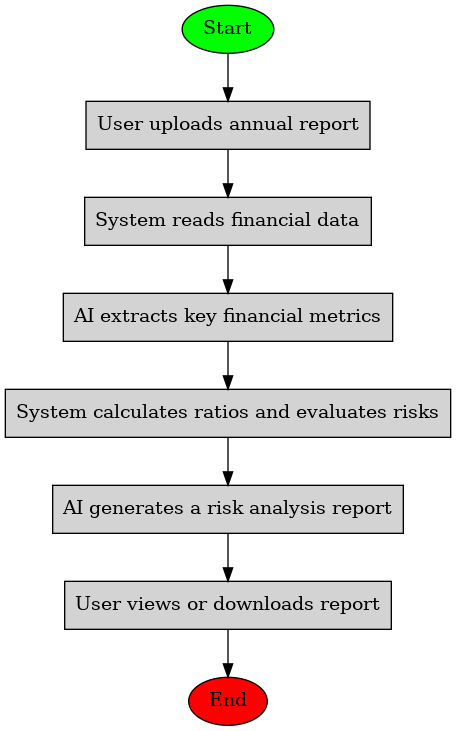
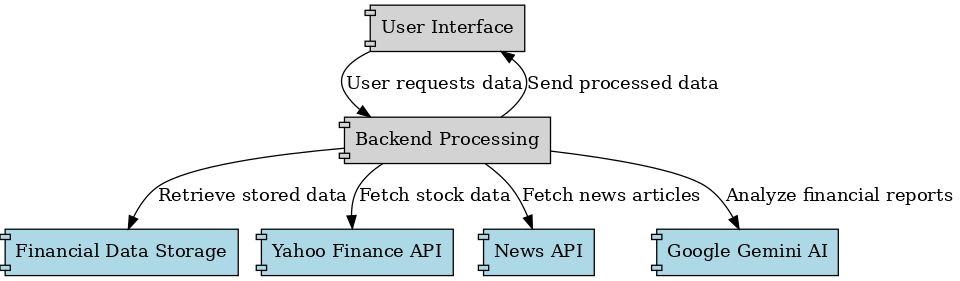
**Key Functionalities:**  
✔ **Benchmarking Financial Health:**

* Compares revenue, profitability, and financial stability between competitors.  
  ✔ **Stock Price & Market Trends Comparison:**
* Tracks **historical stock performance** to determine **which company has stronger market momentum.**  
  ✔ **Risk Profile Comparison:**
* Evaluates **operational, financial, and market risks** for both companies.  
  ✔ **SWOT Analysis (Strengths, Weaknesses, Opportunities, Threats):**
* AI-powered analysis of **which company holds a competitive edge.**  
  ✔ **Graphical Visualizations:**
* Heatmaps, charts, and tables for **quick insights into competitor performance.**

**Tools & Technologies Used:**  
📌 Yahoo Finance API, Machine Learning for Financial Analysis, Seaborn, Google Gemini AI.

**UMLs**Use Case Diagram

****

Class Diagram   
  
  
  
  
  
  
  
  
  
  
  
Sequence Diagram   
  
  
  
Activity Diagram   
  
  
  
Component Diagram   
  
  
  
  
  
  
**Technical Description Of Code  
  
  
Service #1**  
This AI-powered Equity Research Analyst is a web-based Flask application that allows users to upload financial reports (PDFs), retrieve insights using a Retrieval-Augmented Generation (RAG) pipeline powered by Google Generative AI, and analyze stock market trends using historical stock data and deep learning (LSTM). The application integrates financial document processing, real-time stock analysis, and AI-driven question-answering capabilities.

**2. Technology Stack**

* **Web Framework**: Flask
* **Vector Database**: FAISS (Facebook AI Similarity Search)
* **Large Language Model (LLM)**: Google Generative AI (Gemini)
* **Document Processing**: LangChain PyPDFLoader, Semantic Chunker
* **Stock Market Data**: Yahoo Finance (yfinance)
* **Financial News Retrieval**: News API
* **Deep Learning for Stock Prediction**: TensorFlow/Keras (LSTM model)

**3. Key Functionalities**

**3.1. Web Interface & User Interaction**

* **Homepage (/)**: Renders the main page where users can navigate to different features.
* **File Upload (/pdfRAG)**: Allows users to upload PDF financial reports for AI-powered analysis.
* **Question-Answering System (/response)**: Provides a chatbot-like interaction where users can ask questions related to uploaded reports.

**4. PDF Processing & Retrieval-Augmented Generation (RAG)**

**4.1. File Handling**

* The application stores uploaded PDF files in the uploads directory.
* Uses secure\_filename() to ensure safe file names.

**4.2. PDF Text Extraction & Chunking**

* **PyPDFLoader**: Extracts text from the uploaded PDF.
* **Semantic Chunker**: Splits documents into meaningful chunks using Google Generative AI embeddings.

python

CopyEdit

def process\_pdf(file\_path):

loader = PyPDFLoader(file\_path)

docs = loader.load()

embedding\_model = GoogleGenerativeAIEmbeddings(

api\_key=os.environ["GOOGLE\_API\_KEY"],

model="models/embedding-001"

)

semantic\_splitter = SemanticChunker(embedding\_model)

chunks = semantic\_splitter.split\_documents(docs)

db = FAISS.from\_documents(chunks, embedding\_model)

retriever2 = db.as\_retriever(search\_type="similarity", search\_kwargs={"k": 10})

return retriever2

**4.3. Prompt Engineering for LLM Querying**

* A system prompt ensures structured and concise responses based on the retrieved financial document context.

python

CopyEdit

def create\_prompt():

system\_prompt = (

"You are an assistant for question-answering with Financial Annual Reports. "

"Use the following pieces of retrieved context to answer "

"the question. If you don't know the answer, say that you "

"don't know. Use three sentences maximum and keep the answer concise.\n\n{context}"

)

return ChatPromptTemplate.from\_messages([("system", system\_prompt), ("human", "{input}")])

**4.4. Retrieval-Augmented Generation (RAG) Pipeline**

* **Retriever**: Searches for relevant financial document snippets.
* **LLM**: Processes user queries and generates responses using Google Generative AI.

python

CopyEdit

retriever2 = process\_pdf(file\_path)

prompt = create\_prompt()

llm = LLm\_config()

question\_answer\_chain = create\_stuff\_documents\_chain(llm, prompt)

rag\_chain = create\_retrieval\_chain(retriever2, question\_answer\_chain)

**5. Stock Market Analysis & Prediction**

**5.1. Data Fetching from Yahoo Finance**

* The system retrieves historical stock data using yfinance.

python

CopyEdit

def fetch\_stock\_data(ticker, start\_date, end\_date):

stock\_data = yf.download(ticker, start=start\_date, end=end\_date)

return stock\_data

**5.2. LSTM-Based Stock Price Prediction**

* **Preprocessing**: Normalizes stock data using MinMaxScaler.
* **Model Architecture**:
  + **LSTM Layers**: Captures sequential patterns.
  + **Dropout Layers**: Prevents overfitting.
  + **Batch Normalization**: Improves stability.
  + **Early Stopping**: Prevents unnecessary training.

python

CopyEdit

def create\_lstm\_model():

model = Sequential([

LSTM(50, return\_sequences=True, input\_shape=(60, 1)),

Dropout(0.2),

BatchNormalization(),

LSTM(50, return\_sequences=False),

Dropout(0.2),

BatchNormalization(),

Dense(25),

Dense(1)

])

model.compile(optimizer='adam', loss='mean\_squared\_error')

return model

**6. Flask Routes & User Interaction**

**6.1. File Upload & Processing (/pdfRAG)**

* Handles file uploads, processes PDFs, and redirects to the response page.

python

CopyEdit

@app.route("/pdfRAG", methods=["GET", "POST"])

def pdfRAG():

if request.method == "POST":

if "file" not in request.files:

flash("No file part", "error")

return redirect(request.url)

file = request.files["file"]

if file.filename == "":

flash("No selected file", "error")

return redirect(request.url)

if file and allowed\_file(file.filename):

filename = secure\_filename(file.filename)

file\_path = os.path.join(app.config["UPLOAD\_FOLDER"], filename)

file.save(file\_path)

session["chat\_history"] = []

session["filename"] = filename

return redirect(url\_for("response"))

return render\_template("pdfRAG.html")

**6.2. AI-Powered Response Generation (/response)**

* Loads the processed PDF retriever, generates answers using the RAG pipeline, and maintains chat history.

python

CopyEdit

@app.route("/response", methods=["GET", "POST"])

def response():

filename = session.get("filename")

if not filename:

return redirect(url\_for("pdfRAG"))

file\_path = os.path.join(app.config["UPLOAD\_FOLDER"], filename)

retriever2 = process\_pdf(file\_path)

prompt = create\_prompt()

llm = LLm\_config()

question\_answer\_chain = create\_stuff\_documents\_chain(llm, prompt)

rag\_chain = create\_retrieval\_chain(retriever2, question\_answer\_chain)

if request.method == "POST":

question = request.form.get("question")

if question:

result = rag\_chain.invoke({"input": question})

answer = result["answer"]

if "chat\_history" not in session:

session["chat\_history"] = []

session["chat\_history"].append({"question": question, "answer": answer})

session.modified = True

return render\_template("response.html", chat\_history=session.get("chat\_history", []))

**7. Security & Configuration**

* **Session Management**: Uses Flask sessions to track uploaded files and chat history.
* **Environment Variables**: Stores API keys securely.
* **Input Validation**: Ensures only allowed files (PDFs) are processed.

python

CopyEdit

app.secret\_key = "GOOGLE\_API\_KEY" # Replace with a secure key

app.config["UPLOAD\_FOLDER"] = UPLOAD\_FOLDER

### Service#2 Stock Data Visualization (continued)

* **create\_stock\_graph()**: Generates line charts for different stock attributes (e.g., opening price, closing price, high price, and volume). It uses Plotly to display the stock trends over time with customizable colors and titles.

python

CopyEdit

closing\_graph = create\_stock\_graph(df, 'Close', 'Closing Price Trend', '#FF5733')

opening\_graph = create\_stock\_graph(df, 'Open', 'Opening Price Trend', '#33FFCE')

high\_graph = create\_stock\_graph(df, 'High', 'High Price Trend', '#FFA500')

volume\_graph = create\_stock\_graph(df, 'Volume', 'Volume Trend', '#FF00FF')

* **plot\_moving\_averages()**: This function plots two moving averages—short-term (50-day) and long-term (200-day)—on top of the stock’s closing price. The moving averages help identify trends and crossovers, which are key indicators in technical analysis.

python

CopyEdit

moving\_avg\_graph = plot\_moving\_averages(df, ticker)

**3. Preparing Data for LSTM Model**

* **prepare\_lstm\_data()**: Prepares the stock price data for use in an LSTM model. The function scales the closing price using MinMaxScaler to normalize the data between 0 and 1. It then creates sequences of data (60 days of stock prices) as input and the next day's price as the target output.

python

CopyEdit

X, y, scaler = prepare\_lstm\_data(df)

* **Input (X)**: Sequences of historical closing prices (60 days).
* **Target (y)**: The stock price for the day after the sequence.

**4. LSTM Model for Stock Price Prediction**

* **build\_lstm\_model()**: Defines a sequential LSTM model using TensorFlow/Keras. The model has two LSTM layers, each followed by batch normalization and dropout layers to prevent overfitting. The final output layer predicts a single value (next day's closing price).

python

CopyEdit

model = build\_lstm\_model(X.shape[1])

* **LSTM Layers**: Capture sequential dependencies in stock price data.
* **Dropout**: Reduces overfitting.
* **Batch Normalization**: Speeds up training and stabilizes the learning process.
* **Output Layer**: Outputs the predicted closing price.

**5. Training the LSTM Model**

* The model is trained on 80% of the data, while the remaining 20% is used for validation. The training is done using the fit() method with early stopping to avoid overfitting (by monitoring validation loss and stopping if performance plateaus).

python

CopyEdit

early\_stopping = EarlyStopping(monitor='val\_loss', patience=10, restore\_best\_weights=True)

model.fit(X\_train, y\_train, epochs=50, batch\_size=16, validation\_data=(X\_test, y\_test), callbacks=[early\_stopping])

* **Early Stopping**: Stops training if the model's performance on validation data doesn’t improve for 10 epochs.

**6. Forecasting Future Stock Prices**

* **predict\_future\_prices()**: Once the model is trained, it generates future stock price predictions based on the last sequence of closing prices. This function predicts the stock prices for the specified number of days (prediction\_days) and uses the LSTM model iteratively to generate each future day’s price based on the previous predictions.

python

CopyEdit

future\_predictions = predict\_future\_prices(model, X\_test, scaler, prediction\_days)

* **Prediction Loop**: The last 60 days' input is used to predict one future day, then the result is fed back into the input for predicting the next day.

**7. Rendering Results in Flask**

* **Visualization**: The prediction results are rendered in the web interface using Plotly graphs. The predicted prices for future days are displayed as a line graph, and the historical data (closing, opening, high, and volume trends) are shown as interactive charts.

python

CopyEdit

fig = go.Figure()

fig.add\_trace(go.Scatter(x=dates, y=future\_predictions.flatten(), mode='lines', name='Forecasted Price', line=dict(color='#FF5733')))

forecast\_graph = fig.to\_html(full\_html=False)

* **Forecast Table**: A table showing the predicted stock prices for future days is created and rendered in the template.

python

CopyEdit

forecast\_df = pd.DataFrame({'Date': dates, 'Predicted Price': future\_predictions.flatten()})

* **HTML Template**: The LSTM.html template is populated with the graphs and forecast table and returned to the user via Flask's render\_template() function.

python

CopyEdit

return render\_template('LSTM.html', closing\_graph=closing\_graph, opening\_graph=opening\_graph, high\_graph=high\_graph,

volume\_graph=volume\_graph, moving\_avg\_graph=moving\_avg\_graph, forecast\_graph=forecast\_graph,

forecast\_table=forecast\_df.to\_html(classes='table table-dark'))

**8. Flask Route (/lstm)**

The Flask route /lstm processes the user’s input (stock ticker and prediction days), retrieves the stock data, builds and trains the LSTM model, and renders the results (stock trends, moving averages, and predictions) on the page.

python

CopyEdit

@app.route("/lstm", methods=['GET', 'POST'])

def lstm():

if request.method == 'POST':

ticker = request.form['ticker']

prediction\_days = int(request.form['prediction\_days'])

# Fetch historical stock data, train LSTM, predict, and render results

...

return render\_template('LSTM.html', ...)

return render\_template('LSTM.html', ...)

**Service#4**

**1. API Key Configuration**

The script uses two API keys for external services:

* NEWSAPI\_KEY: Used to fetch news articles from **NewsAPI**.
* GOOGLE\_API\_KEY: Used to access **Google Gemini AI** for text embedding and summarization.

python

CopyEdit

NEWSAPI\_KEY = "c0d8b16dbbd14701aea9abc230289c7d"

GOOGLE\_API\_KEY = "AIzaSyBKDVPPmYIvTH1SiNdvGxHs\_jUFpxAQKKA"

newsapi = NewsApiClient(api\_key=NEWSAPI\_KEY)

**2. News Retrieval (fetch\_news())**

This function gathers financial news articles related to a given stock ticker from **NewsAPI** and **Yahoo Finance**.

**Steps:**

1. **Retrieve the company’s short name** using Yahoo Finance (yfinance), which improves search accuracy.
2. **Fetch news from NewsAPI** using the company name, filtering for English articles sorted by recent publication date.
3. **Fetch news from Yahoo Finance**, which provides stock-specific news articles.
4. **Structure the news data** into a Pandas DataFrame with essential fields (source, title, description, URL, timestamp).
5. **Save the data as a CSV file (news.csv)** for further processing.

python

CopyEdit

def fetch\_news(ticker):

company\_name = yf.Ticker(ticker).info.get("shortName", ticker)

newsapi\_news = newsapi.get\_everything(q=company\_name, language="en", sort\_by="publishedAt", page\_size=20)

yahoo\_news = yf.Ticker(ticker).news

articles = []

# Processing NewsAPI articles

for article in newsapi\_news.get("articles", []):

articles.append({

"source": article["source"]["name"],

"title": article["title"],

"description": article["description"],

"url": article["url"],

"published\_at": article["publishedAt"],

"source\_type": "NewsAPI"

})

# Processing Yahoo Finance news

for article in yahoo\_news:

articles.append({

"source": article.get("provider", "Unknown"),

"title": article.get("title", "No Title Available"),

"description": article.get("summary", "No Description Available"),

"url": article.get("link", "#"),

"published\_at": pd.to\_datetime(article.get("providerPublishTime", 0), unit="s"),

"source\_type": "Yahoo Finance"

})

df = pd.DataFrame(articles)

df.to\_csv("news.csv", index=False)

return df

**Error Handling Considerations:**

✅ Avoids KeyError by using .get() for dictionary lookups.  
✅ Converts timestamps safely using pd.to\_datetime().  
✅ Uses "Unknown" as a fallback source name to prevent missing data.

**3. News Summarization (generate\_summary())**

This function extracts key insights from the collected news using **retrieval-augmented generation (RAG)**.

**Steps:**

1. **Load the raw news data** from news.csv.
2. **Chunk the text** using **LangChain's RecursiveCharacterTextSplitter**, ensuring the LLM processes relevant sections efficiently.
3. **Embed the text** using **Google Generative AI embeddings** and store it in a **FAISS vector database**.
4. **Retrieve relevant news chunks** based on semantic similarity.
5. **Use Google Gemini AI** to generate a structured summary.

python

CopyEdit

def generate\_summary():

with open("news.csv", "r", encoding="utf-8") as file:

text\_data = file.read()

# Splitting the text into manageable chunks

documents2 = [Document(page\_content=text\_data)]

text\_splitter = RecursiveCharacterTextSplitter(chunk\_size=1500, chunk\_overlap=200)

chunked\_docs = text\_splitter.split\_documents(documents2)

# Embedding and storing in FAISS

embedding\_model = GoogleGenerativeAIEmbeddings(api\_key=GOOGLE\_API\_KEY, model="models/embedding-001")

db = FAISS.from\_documents(chunked\_docs, embedding\_model)

db.save\_local("faiss\_index")

retriever = db.as\_retriever(search\_type="similarity", search\_kwargs={"k": 10})

# Creating a structured prompt

system\_prompt = "Summarize financial news in a structured format with key indicators."

prompt = ChatPromptTemplate.from\_messages([

("system", system\_prompt),

("human", "{context}")

])

# Using Google Gemini AI to generate the summary

llm = ChatGoogleGenerativeAI(model="gemini-2.0-flash", temperature=0.3, max\_tokens=500)

question\_answer\_chain = create\_stuff\_documents\_chain(llm, prompt)

rag\_chain = create\_retrieval\_chain(retriever, question\_answer\_chain)

# Running the RAG pipeline

response = rag\_chain.invoke({"input": "Summarize all news with detailed analysis."})

return response["answer"]

**Key Components Used:**

🔹 **FAISS** – Efficient nearest-neighbor search for retrieving relevant text chunks.  
🔹 **Google Gemini AI** – Generates a structured summary from retrieved information.  
🔹 **LangChain RAG** – Enhances summarization by combining retrieval and generation.

**4. Web Interface (/news Route)**

This **Flask web route** provides two main features:

* **Download raw news data** as a CSV file.
* **Generate a financial news summary** using AI.

**Steps:**

1. User submits a stock ticker via an HTML form.
2. If the user selects **"Download CSV"**, fetch\_news() is called, and the file is sent as a download.
3. If the user selects **"Generate Summary"**, both fetch\_news() and generate\_summary() are executed.
4. The generated summary is displayed on the news.html page.

python

CopyEdit

@app.route("/news", methods=["GET", "POST"])

def news():

summary = None

if request.method == "POST":

ticker = request.form.get("ticker").upper()

action = request.form.get("action")

if action == "download\_csv":

fetch\_news(ticker)

return send\_file("news.csv", as\_attachment=True)

elif action == "generate\_summary":

fetch\_news(ticker)

summary = generate\_summary()

return render\_template("news.html", summary=summary)

## service#5 **1. Processing a Web URL (**process\_url**)**

This function extracts text from a given URL, converts it into vector embeddings, and stores it in a FAISS index for efficient retrieval.

### **Steps:**

1. **Load web content** using WebBaseLoader(url).
2. **Create embeddings** using **Google Generative AI embeddings**.
3. **Store document vectors** in a **FAISS** database.
4. **Enable semantic search** by creating a retriever that fetches the most relevant chunks for answering questions.

python

CopyEdit

def process\_url(url):

loader = WebBaseLoader(url) # Extracts text from the URL

docs = loader.load() # Loads document content

# Creating embeddings using Google Generative AI

embedding\_model = GoogleGenerativeAIEmbeddings(

api\_key=os.environ["GOOGLE\_API\_KEY"],

model="models/embedding-001"

)

# Indexing document embeddings in FAISS

db = FAISS.from\_documents(docs, embedding\_model)

# Creating a retriever for similarity-based search

retriever = db.as\_retriever(search\_type="similarity", search\_kwargs={"k": 10})

return retriever

### **Key Components:**

✅ **WebBaseLoader** – Extracts text from the web page.  
✅ **Google Generative AI Embeddings** – Converts text into vector representations.  
✅ **FAISS Vector Index** – Stores embeddings and enables similarity-based retrieval.  
✅ **Retriever** – Retrieves the top k=10 most relevant text chunks.

## **2. Generating Prompts for AI (**create\_prompt**)**

This function constructs a **system prompt** that instructs the AI on how to answer questions concisely.

### **Steps:**

1. **Define assistant behavior** – Answers should be based only on retrieved context.
2. **Ensure response quality** – Answers should be limited to three sentences and concise.
3. **Handle unknown cases** – If there’s no relevant information, the AI should say, **"I don’t know."**

python

CopyEdit

def create\_prompt():

system\_prompt = (

"You are an assistant for question-answering based on web articles. "

"Use the retrieved context to answer the question. "

"If you don't know the answer, say that you don't know. "

"Use three sentences maximum and keep the answer concise.\n\n{context}"

)

return ChatPromptTemplate.from\_messages([

("system", system\_prompt),

("human", "{input}") # User input will replace `{input}`

])

### **Key Features:**

✅ **Ensures response accuracy** – AI must rely on retrieved content.  
✅ **Conciseness enforcement** – Limits answers to three sentences.  
✅ **Handles uncertainty gracefully** – AI acknowledges when it lacks sufficient information.

## **3. Web Interface (**/web **&** /response2 **Routes)**

The Flask application provides a user interface for submitting URLs, asking questions, and displaying AI-generated responses.

### **3.1 URL Submission (**/web**)**

This route handles user input for web page URLs and initializes the session.

#### **Steps:**

1. **Accepts user input (URL).**
2. **Validates the URL.** If missing, flashes an error and redirects.
3. **Stores the URL in a session variable** to maintain continuity.
4. **Clears previous chat history** when a new URL is submitted.
5. **Redirects to** /response2 for question-answering.

python

CopyEdit

@app.route("/web", methods=["GET", "POST"])

def web():

if request.method == "POST":

url = request.form.get("url")

if not url:

flash("No URL provided", "error") # Display error message

return redirect(request.url)

session["chat\_history"] = [] # Reset chat history

session["url"] = url # Store URL in session

return redirect(url\_for("response2")) # Redirect to Q&A page

return render\_template("webRAG.html")

### **Key Features:**

✅ **Session-based URL handling** – Allows users to ask multiple questions about the same article.  
✅ **Flash messages for validation** – Ensures a valid URL is entered.

### **3.2 Question Answering (**/response2**)**

This route retrieves web content, processes questions, and generates AI responses.

#### **Steps:**

1. **Retrieve the stored URL** from the session.
2. **Process the URL** to create a retriever for similarity search.
3. **Generate a prompt for structured question-answering.**
4. **Configure the LLM (Google Gemini) for text generation.**
5. **Create a RAG pipeline** that retrieves relevant content and generates answers.
6. **Handle user-submitted questions**, retrieve answers, and store chat history.
7. **Display the chat history** in response.html.

python

CopyEdit

@app.route("/response2", methods=["GET", "POST"])

def response2():

url = session.get("url")

if not url:

return redirect(url\_for("index")) # Redirect if no URL is stored

retriever = process\_url(url) # Process URL for retrieval

prompt = create\_prompt() # Create system prompt

llm = LLm\_config() # Initialize LLM (Google Gemini)

# Create the RAG pipeline

question\_answer\_chain = create\_stuff\_documents\_chain(llm, prompt)

rag\_chain = create\_retrieval\_chain(retriever, question\_answer\_chain)

if request.method == "POST":

question = request.form.get("question")

if question:

result = rag\_chain.invoke({"input": question}) # Retrieve and generate answer

answer = result["answer"]

# Store chat history in session

if "chat\_history" not in session:

session["chat\_history"] = []

session["chat\_history"].append({"question": question, "answer": answer})

session.modified = True # Ensure session updates

return render\_template("response.html", chat\_history=session.get("chat\_history", []))

### **Key Features:**

✅ **Session-based Q&A** – Maintains chat history across interactions.  
✅ **RAG Pipeline** – Combines retrieval and generative AI for context-aware responses.  
✅ **Dynamic Answering** – Retrieves the most relevant text snippets before generating a response.

## **Potential Improvements**

✅ **Enhance Performance:**

* Implement **caching** to prevent redundant web scraping.
* Optimize FAISS search by tuning indexing parameters.

✅ **Improve User Experience:**

* Add **a progress indicator** while retrieving and summarizing content.
* Display **source text snippets** with AI-generated answers.

✅ **Security Considerations:**

* Use **URL validation** to prevent **web scraping of malicious sites**.
* **Sanitize user input** to prevent **injection attacks**.

## Service # 6 1. Environment Setup

Before execution, the system sets up the required API key for **Google Generative AI** to access embeddings and LLM functionalities.

python

CopyEdit

os.environ["GOOGLE\_API\_KEY"] = "AIzaSyBKDVPPmYIvTH1SiNdvGxHs\_jUFpxAQKKA"

🔹 **Ensures secure access** to Google's AI services for text embedding and response generation.

**2. Predefined Financial Queries (query\_headings)**

The application includes a structured **dictionary of financial analysis queries**, where each query is mapped to a **specific financial metric**.

**Example Queries:**

* **Revenue Comparison** – Matches fiscal years and analyzes revenue trends.
* **Cost of Goods Sold (COGS)** – Compares trends and evaluates profitability.
* **Operating Profit & Net Profit** – Assesses financial efficiency.
* **Liquidity Ratios (Current & Quick Ratio)** – Evaluates financial stability.
* **Debt-to-Equity & Interest Coverage Ratio** – Analyzes financial leverage.
* **Net Profit Margin & EPS Trends** – Assesses investor profitability.
* **Cash Flow & Market Risks** – Identifies financial vulnerabilities.

🔹 **Purpose:** These queries guide the **LLM** to generate structured and insightful financial analysis instead of free-form responses.

**3. Processing the Financial TXT File (process\_txt)**

This function **extracts text** from a **user-uploaded financial report**, splits it into chunks, creates **vector embeddings**, and enables retrieval for AI-based question answering.

**Steps:**

1. **Reads** the uploaded TXT file.
2. **Splits text** into 1000-character chunks (with 100-character overlap) for effective retrieval.
3. **Creates vector embeddings** using **Google Generative AI embeddings**.
4. **Stores** document vectors in **FAISS** for efficient search.
5. **Loads an LLM** (**Gemini 1.5 Flash**) for financial Q&A.
6. **Defines a system prompt** to instruct the AI on answering financial queries.
7. **Creates a RAG pipeline** to process user questions using **retrieved text**.

python

CopyEdit

def process\_txt(txt\_path):

try:

with open(txt\_path, 'r', encoding='utf-8') as file:

text = file.read()

# Splitting large text into chunks for retrieval

splitter = RecursiveCharacterTextSplitter(chunk\_size=1000, chunk\_overlap=100)

chunks = splitter.split\_text(text)

# Creating vector embeddings

embedding\_model = GoogleGenerativeAIEmbeddings(api\_key=os.environ["GOOGLE\_API\_KEY"], model="models/embedding-001")

db = FAISS.from\_texts(chunks, embedding\_model)

db.save\_local("faiss\_index") # Storing embeddings

# Creating a retriever for semantic search

retriever = db.as\_retriever(search\_type="similarity", search\_kwargs={"k": 5})

# Configuring Google Gemini LLM

llm = ChatGoogleGenerativeAI(model="gemini-1.5-flash", temperature=0.3, max\_tokens=1000)

# Defining system behavior

system\_prompt = """

Act as a Finance Assistant. Answer questions based on the uploaded text file.

If data is missing, reply with 'not present'.

{context}

"""

# Creating a structured prompt for the LLM

prompt = ChatPromptTemplate.from\_messages([

("system", system\_prompt),

("human", "{input}"),

])

# Creating the RAG pipeline (retriever + LLM)

question\_answer\_chain = create\_stuff\_documents\_chain(llm, prompt)

rag\_chain = create\_retrieval\_chain(retriever, question\_answer\_chain)

return rag\_chain

except Exception as e:

print(f"Error processing TXT: {e}")

return None

**Key Features:**

✅ **Text Chunking & Overlapping** – Ensures better semantic retrieval.  
✅ **FAISS Vector Indexing** – Speeds up financial query processing.  
✅ **Google Gemini AI** – Provides insightful financial analysis.  
✅ **Structured Retrieval-Augmented Generation (RAG)** – Enhances answer quality.

**4. Formatting the AI Response (format\_answer)**

This function **formats AI-generated responses** to enhance readability by converting Markdown-style headings into HTML.

python

CopyEdit

def format\_answer(answer):

return re.sub(r'\\(.?)\\\*', r'<h3>\1</h3>', answer)

✅ **Improves UI readability** by formatting responses into structured headings.

**5. Handling User Uploads & Processing Financial Queries (/compi Route)**

This Flask route:

1. **Accepts a user-uploaded TXT file** and validates its format.
2. **Processes the financial document** using process\_txt().
3. **Executes predefined financial queries** on the processed document.
4. **Formats responses** and displays them in the UI.

python

CopyEdit

@app.route('/compi', methods=['GET', 'POST'])

def compi():

answers = {}

if request.method == 'POST':

if 'txt' not in request.files:

return render\_template('Compi.html', error='No TXT file provided.')

txt\_file = request.files['txt']

if txt\_file.filename == '' or not txt\_file.filename.endswith('.txt'):

return render\_template('Compi.html', error='Invalid file type. Please upload a TXT file.')

# Save the uploaded file temporarily

with tempfile.NamedTemporaryFile(delete=False, suffix='.txt') as temp\_txt:

txt\_file.save(temp\_txt.name)

temp\_txt\_path = temp\_txt.name

# Process the uploaded file

rag\_chain = process\_txt(temp\_txt\_path)

if rag\_chain:

for query, heading in query\_headings.items():

response = rag\_chain.invoke({"input": query})

answers[heading] = format\_answer(response["answer"])

else:

return render\_template('Compi.html', error='Failed to process TXT file.')

os.unlink(temp\_txt\_path) # Remove temporary file

return render\_template('Compi.html', answers=answers)

**Key Features:**

✅ **Validates File Uploads** – Ensures the user provides a valid TXT file.  
✅ **Session-Based Processing** – Supports multiple file uploads dynamically.  
✅ **Automated Financial Analysis** – Executes structured financial queries.  
✅ **Displays Results in UI** – Formats and presents the AI-generated insights.

## Conclusion

Insight Invest is a cutting-edge AI-powered equity research analyst designed to automate and enhance the financial analysis process. By integrating state-of-the-art machine learning models, Retrieval Augmented Generation (RAG) techniques, and API-based data extraction, the system streamlines the traditionally manual and time-consuming tasks of an equity research analyst. The tool provides investors, financial analysts, and researchers with accurate, real-time insights into stock trends, financial news, risk factors, and company reports.

Through its six core services—data visualization, news analysis, financial report retrieval, finance website analysis, risk assessment, and competitor benchmarking—Insight Invest offers a holistic approach to financial decision-making. The system eliminates the need for manual data aggregation by leveraging APIs from Yahoo Finance, News API, and Google Gemini Multimodal AI. Additionally, advanced AI-driven analysis enables investors to make well-informed decisions based on historical trends, predictive modeling, and sentiment analysis.

One of the key advantages of Insight Invest is its ability to process large-scale financial data efficiently while maintaining high accuracy. The automation of stock research and risk assessment significantly reduces the likelihood of human error, providing more reliable insights. Furthermore, the system's chatbot-driven approach makes it highly interactive, allowing users to engage with financial data in a more intuitive and user-friendly manner.

The project adheres to IEEE documentation standards, ensuring that all functional and non-functional requirements are clearly defined, and that the architecture is well-structured for future enhancements. The implementation of robust security measures, compliance with financial regulations, and support for scalability make Insight Invest a powerful and versatile tool in the financial technology sector.