# Project Report: Enhanced News Summarizer Tool

## ****1. Introduction****

The **Enhanced News Summarizer Tool** is an AI-driven solution designed to assist **equity research analysts** by automating the process of retrieving and summarizing news articles. By leveraging **NewsAPI** for real-world news retrieval and **Groq LLM** for generating concise, actionable insights, the tool streamlines information gathering, saving valuable time for analysts.

The development of this tool involved integrating state-of-the-art APIs and creating a user-friendly web interface for seamless interaction.

## ****2. Development Phases****

### ****Phase 1: Environment Setup****

* Installed essential Python libraries:
  + Langchain, groq-cloud, Streamlit, and NewsAPI-python.
* API keys for Groq LLM and NewsAPI were securely stored using an .env file to ensure security and ease of access.

### ****Phase 2: LangChain Configuration****

* Integrated the **Groq LLM** with the LangChain library.
* Created a pipeline using:
  + **Prompt Template**: Designed to process queries for summarization.
  + **Groq LLM**: Used for generating summaries based on the provided query.
* Successfully tested the pipeline with a sample query: "Impact of inflation on stock markets."

### ****Phase 3: Building the Streamlit Interface****

* Developed a **user-friendly web application** using Streamlit.
* Features of the interface:
  + **Query Input**: Allows users to input queries like "Latest trends in AI."
  + **Dynamic Output**: Displays a summarized response interactively.
  + **Error Handling**: Provides clear warnings for empty inputs or system errors.
* The app was made intuitive and visually appealing for ease of use.

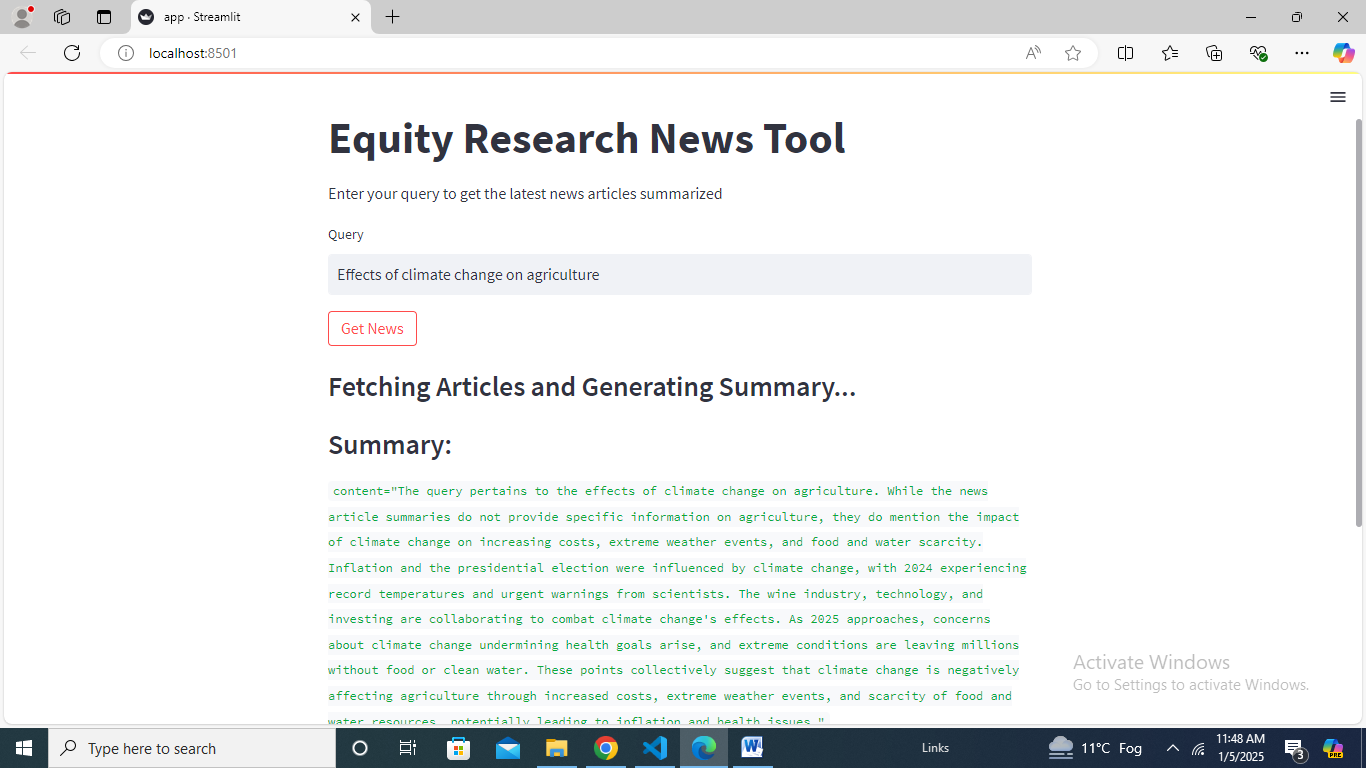
### ****Phase 4: Enhancing the Tool****

* **NewsAPI Integration**:
  + Fetched real-world news articles based on user queries.
  + Limited retrieval to the **top 5 most relevant articles** for optimal performance.
* **Summarization of Articles**:
  + Article descriptions were summarized and concatenated into a single input.
  + The **Groq LLM pipeline** generated a final summary by combining the user query and article summaries.

### ****Phase 5: Testing and Validation****

#### Testing Scenarios:

* Validated the app's functionality with queries like:
  + "Impact of inflation on stock markets."
  + "Latest trends in electric vehicles."
  + "Effects of climate change on agriculture."





#### Observations:

1. **Accuracy**:
   * Summaries were highly relevant and aligned with the content of fetched articles.
2. **Performance**:
   * Response times averaged less than 2 seconds for most queries.
3. **Error Handling**:
   * Clear warnings for empty inputs or no articles retrieved.
   * User-friendly error messages for API-related issues.

#### Challenges Resolved:

* Optimized the number of articles fetched to reduce token usage and processing time.
* Enhanced the pipeline for robustness in handling edge cases.

## ****3. Key Features****

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

1. **Real-World News Retrieval**: Fetches news articles using **NewsAPI**.
2. **AI-Driven Summarization**: Combines article summaries with the user query using **Groq LLM**.
3. **Interactive Interface**: Streamlit app allows seamless user interaction.

### ****User Experience Enhancements****:

* Clear instructions and placeholders for user queries.
* Real-time feedback during summary generation.
* Responsive and visually clean interface.

## ****4. Conclusion****

The **Enhanced News Summarizer Tool** achieves its objective of providing actionable insights for equity research analysts. The integration of **NewsAPI** and **Groq LLM**, along with a robust web interface, ensures accurate, context-aware summaries.

This project demonstrates the potential of combining APIs and LLMs to create practical solutions for real-world problems.

## ****5. Future Enhancements****

To further improve the tool, the following features can be added:

1. **User Authentication**:
   * Restrict access to authorized users.
2. **Advanced UI/UX**:
   * Implement additional Streamlit components like charts or article previews.
3. **Export Options**:
   * Allow users to download summaries as PDFs or save them for future reference.
4. **Query History**:
   * Store past queries and summaries for easy access.

## ****Appendices****

### ****Technologies Used****:

1. **Programming Language**: Python
2. **Libraries**:
   * langchain, Groq-cloud, Streamlit, NewsAPI-python
3. **APIs**:
   * **Groq LLM API** for summarization
   * **NewsAPI** for real-world news retrieval