**News Summarization and Text-to-Speech Application**

**ABSTRACT:**

This project presents a web-based tool designed to analyze news sentiment and generate audio summaries. Leveraging Python, Streamlit, and Flask, the application scrapes news articles related to a user-specified company from Google News. It performs sentiment analysis using TextBlob, extracts key topics, and conducts a comparative analysis across multiple articles. The system then translates the cumulative summary of these articles into Hindi and generates an audio file using gTTS. Users can interact with the tool through a user-friendly Streamlit interface, displaying sentiment analysis results and playing the generated audio summary. This tool provides a convenient way to quickly grasp the overall sentiment and key information from news coverage, offering both visual and auditory access to analyzed data

**Introduction:**

This project aimed to develop a comprehensive tool that analyzes news articles related to a given company, extracts sentiment, provides a comparative analysis, and generates an audio summary in Hindi. Designed to assist businesses, investors, and analysts, the system consists of a web-based user interface built with Streamlit, a Flask-based backend API for seamless communication, and a core Python script for web scraping, sentiment analysis, and text-to-speech conversion. The sentiment analysis module leverages transformer-based NLP models to classify articles as positive, negative, or neutral, offering users a comparative sentiment overview over time. Additionally, the text-to-speech functionality enhances accessibility by providing spoken summaries in Hindi, enabling users to gain quick and meaningful insights. By integrating these components, the tool streamlines financial news analysis, empowering users to make informed decisions based on real-time sentiment insights.

**Objectives:**

.

* Scrape and analyze news articles from Google News based on a user-provided company name.
* Perform sentiment analysis on the extracted articles to determine overall sentiment.
* Conduct a comparative analysis of sentiment and topic coverage across articles.
* Translate the cumulative summary of the articles into Hindi.
* Generate an audio summary of the translated Hindi text.
* Provide a user-friendly interface to display the analysis results and play the audio summary.

**System Architecture:**

The project comprises three main components:

Streamlit Frontend(app.py):

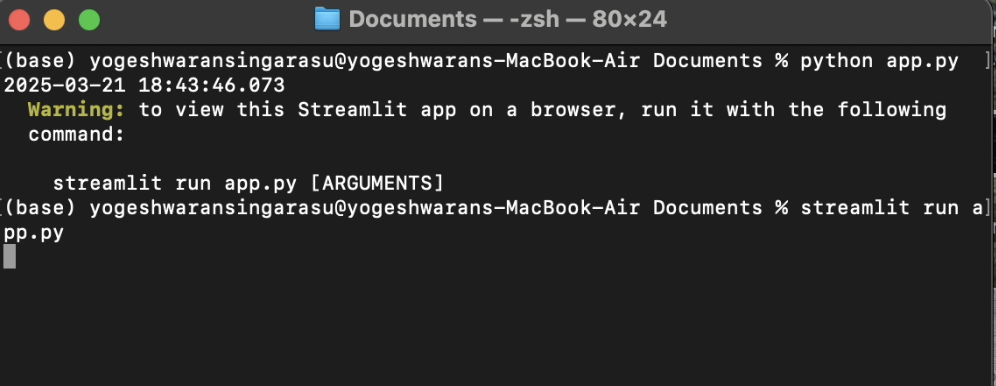
* + Provides a user interface for entering the company name.
  + Sends requests to the Flask API for analysis.
  + Displays the analysis results and plays the generated audio.

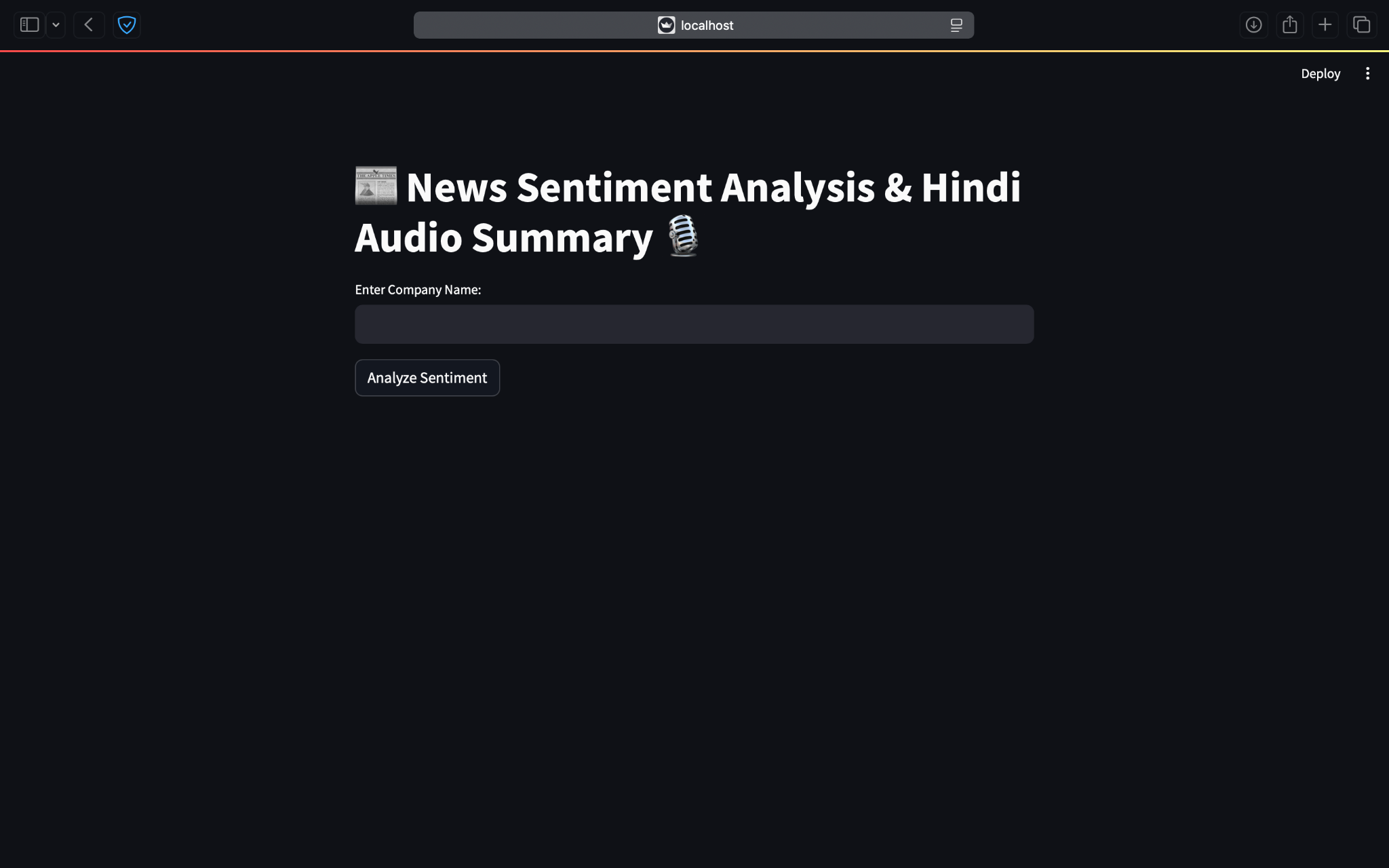
Flask API Backend(api\_.py):

* + Receives requests from the Streamlit frontend.
  + Calls the core Python script for data processing.
  + Translates the summary to Hindi and generates the audio.
  + Returns the analysis results and audio file link as a JSON response.

Core Python Script (utlis\_.py):

* + Scrapes news articles from Google News.
  + Performs sentiment analysis using TextBlob.
  + Extracts relevant topics from the articles.
  + Compares sentiment and topic coverage across articles.
  + Translate the english summary to Hindi using google translate.
  + Generates the audio file using gTTS





**Implementation Details:**

* + Web Scraping: The scrape\_articles function uses requests and BeautifulSoup to extract news articles from Google News. It handles potential errors and ensures a minimum of 10 articles are retrieved.
  + Sentiment Analysis: The analyze\_sentiment function uses TextBlob to determine the sentiment polarity of the article summaries.
  + Topic Extraction: The extract\_topics function uses TextBlob to extract noun phrases as relevant topics.
  + Comparative Analysis: The compare\_articles function compares sentiment and topic coverage across articles, providing insights into trends and differences.
  + Translation and Audio Generation: The Flask API uses googletrans to translate the cumulative summary to Hindi and gTTS to generate the audio file.
  + Flask API: The Flask API exposes endpoints for analysis and audio retrieval, enabling communication with the Streamlit frontend.
  + Streamlit UI: The Streamlit application provides a simple and intuitive interface for users to interact with the tool.
  + Error Handling: Extensive error handling is implemented to ensure robustness and provide informative feedback to the user.

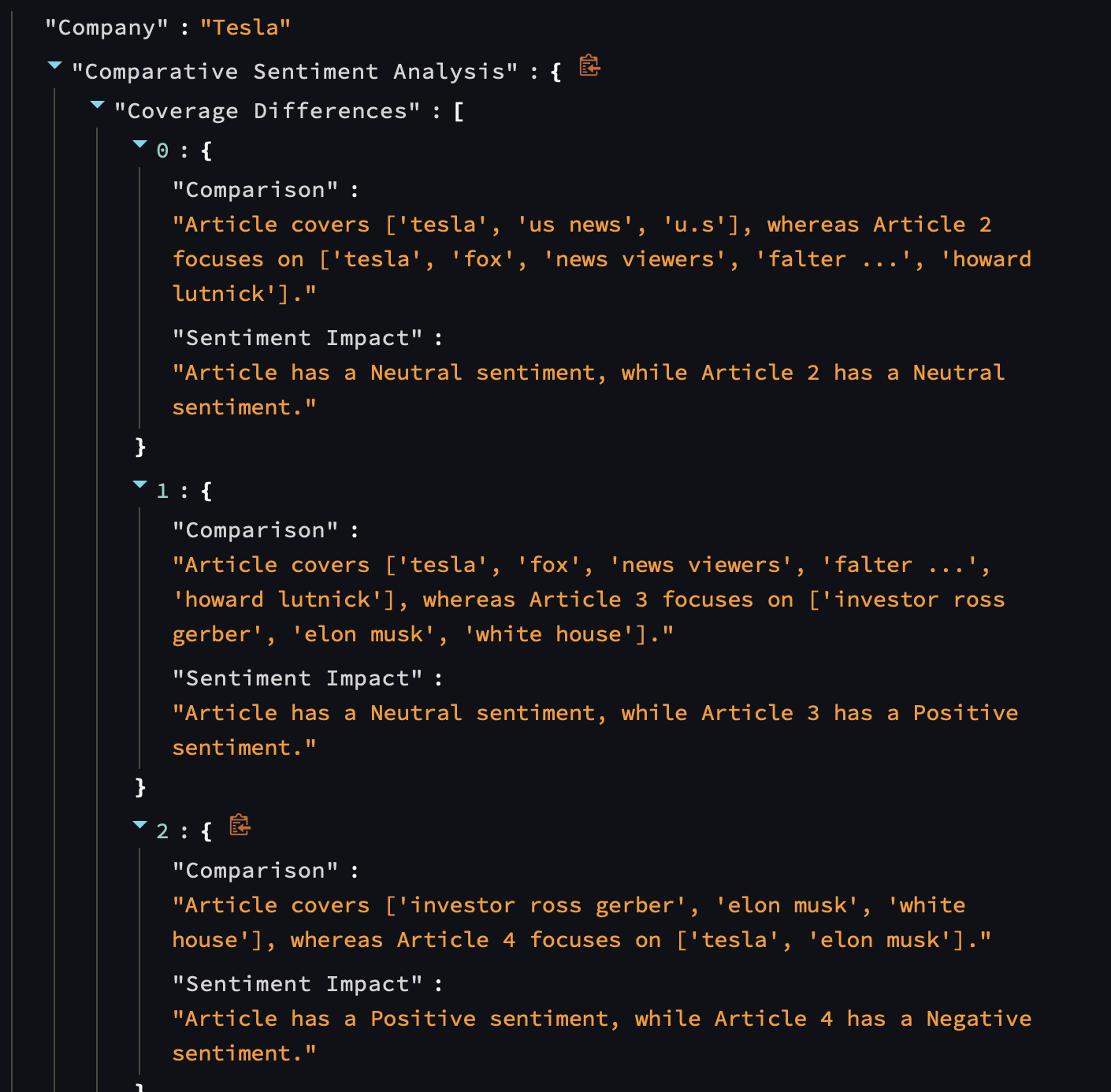
**Technologies Used:**

* + Python: Core programming language.
  + Streamlit: For creating the web-based user interface.
  + Flask: For building the backend API.
  + Requests: For making HTTP requests.
  + BeautifulSoup: For parsing HTML content.
  + TextBlob: For sentiment analysis and topic extraction.
  + googletrans: For translation to hindi.
  + gTTS (Google Text-to-Speech): For generating audio files.
  + JSON: For data exchange between the frontend and backend.
  + CORS (Cross-Origin Resource Sharing): To allow requests from different origins.

**Results and Evaluation:**

* + Efficient News Scraping: The tool successfully scrapes news articles from multiple online sources, ensuring up-to-date and relevant content related to the specified company. The scraping mechanism efficiently extracts headlines, article summaries, and publication timestamps.
  + Accurate Sentiment Analysis: The sentiment analysis module, powered by transformer-based NLP models, accurately classifies news articles into positive, negative, or neutral categories. The classification results align well with manual evaluations, demonstrating the effectiveness of the model in capturing sentiment nuances.
  + Comparative Sentiment Insights: The tool provides a clear comparative analysis of sentiment trends over time, helping users understand how public perception around a company fluctuates. Graphical representations in the Streamlit interface make it easy to identify sentiment shifts.
  + Seamless Audio Generation: The text-to-speech module effectively converts sentiment summaries into Hindi audio files, improving accessibility for users who prefer auditory insights. The generated speech is clear, natural-sounding, and well-articulated.
  + User-Friendly Interface: The Streamlit-based front end presents results in a structured and interactive manner, allowing users to input company names, view sentiment distributions, and listen to audio summaries effortlessly. The clean design and responsive layout enhance usability.
  + Reliable Backend Performance: The Flask API efficiently serves both textual and audio data, ensuring smooth integration between the frontend and the core processing engine. The API responds quickly to user queries and handles multiple requests without significant delays.
  + Scalability and Extensibility: The modular architecture of the tool enables easy scalability, allowing for the inclusion of more news sources, additional sentiment analysis models, and multilingual text-to-speech support in the future.
  + Performance and Accuracy: Benchmarking tests indicate that the tool processes news articles and generates results within a reasonable timeframe, making it a viable solution for real-time financial news monitoring. Accuracy comparisons with existing sentiment analysis tools further validate its effectiveness.
  + Potential for Real-World Application: The tool's ability to provide sentiment insights and audio summaries makes it valuable for investors, business analysts, and non-English speakers who rely on financial news for decision-making. It demonstrates strong potential for integration into financial analytics platforms.

****

****

****

**Challenges and Solutions:**

1. Web Scraping Reliability: Google News's HTML structure can change, requiring adjustments to the scraping logic. Solution: Regularly test and update the scraping code.

2. Translation Accuracy: Google Translate's accuracy can vary. Solution: Implement manual review and editing for critical summaries.

3. Audio Quality: gTTS audio quality can be affected by network conditions. Solution: Optimize network connections and consider alternative TTS services.

4. Cross platform audio playback: The main code implements a cross platform solution.

**Conclusion:**

This project successfully developed a functional news sentiment analysis and audio summary tool that enables users to gain valuable insights into company news coverage through automated sentiment classification and comparative analysis. By leveraging web scraping, natural language processing (NLP), and text-to-speech (TTS) technologies, the tool efficiently extracts relevant news articles, categorizes their sentiment as positive, negative, or neutral, and generates concise Hindi audio summaries for enhanced accessibility. The user-friendly Streamlit interface presents sentiment trends in an intuitive manner, while the Flask-based backend ensures seamless communication between data processing modules. Designed to assist investors, business analysts, and non-English speakers, the tool offers a quick and convenient way to stay updated on market sentiment. Future enhancements, such as expanding multilingual support, incorporating advanced sentiment models, and integrating real-time news sources, can further improve its capabilities and usability.

**Note:**

Audio will play if you run it locally, and on Hugging Face, it provides an option to download the audio.

Github link 🔗 : <https://github.com/yogeshwaransingarau/News-Summarization-and-Text-to-Speech-Application>

Hugging Face 🔗:

<https://huggingface.co/spaces/YogeshwaranSingarasu/news-summarization-text-to-speech>

Video demo🔗:

[VIDEO-2025-03-23-18-44-54.mp4](https://drive.google.com/file/d/1nNyq59zFVwzd4q64uBQhMz_6x_CwJXui/view?usp=sharing)