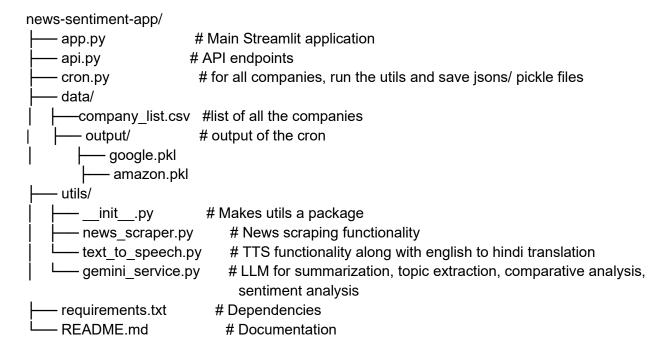
#### **Problem Statement:**

Design a web-based application that extracts news articles about a company, performs sentiment analysis, and generates Hindi TTS output.

#### Folder structure:



## Flow of execution:

- 1. Run the cron which:
  - a. reads the list of companies from the company\_list.csv file
  - searches google news for each company in the list and extracts the title, article from 10 unique news articles related to the given company. Consider only non-JS weblinks that can be scraped using BeautifulSoup (bs4).
  - c. for each company, the cron calls gemini with the news articles and generates a json response that contains summary of each article, topics in each article, comparative analysis, sentiment analysis, and an overall sentiment of the articles.
  - d. These jsons are dumped as pickle files in the data folder.
- app: Streamlit application where users input a company name (via dropdown) to fetch news articles and generate the sentiment report along with an audio output in hindi of just the overall sentiment
- 3. api: interacts with the app

# Sample JSON output of the cron:

```
"Company": "Tesla",
"Articles": [
"Title": "Tesla's New Model Breaks Sales Records",
"Summary": "Tesla's latest EV sees record sales in Q3...",
"Sentiment": "Positive",
"Topics": ["Electric Vehicles", "Stock Market", "Innovation"]
},
"Title": "Regulatory Scrutiny on Tesla's Self-Driving Tech",
"Summary": "Regulators have raised concerns over Tesla's self-driving
software...",
"Sentiment": "Negative",
"Topics": ["Regulations", "Autonomous Vehicles"]
],
"Comparative Sentiment Score": {
"Sentiment Distribution": {
"Positive": 1,
"Negative": 1,
"Neutral": 0
},
"Coverage Differences": [
"Comparison": "Article 1 highlights Tesla's strong sales, while Article 2
discusses regulatory issues.",
"Impact": "The first article boosts confidence in Tesla's market growth,
while the second raises concerns about future regulatory hurdles."
},
"Comparison": "Article 1 is focused on financial success and innovation,
whereas Article 2 is about legal challenges and risks.",
"Impact": "Investors may react positively to growth news but stay cautious
due to regulatory scrutiny."
}
"Topic Overlap": {
"Common Topics": ["Electric Vehicles"],
"Unique Topics in Article 1": ["Stock Market", "Innovation"],
"Unique Topics in Article 2": ["Regulations", "Autonomous Vehicles"]
}
```

```
},
,
"Final Sentiment Analysis": "Tesla's latest news coverage is mostly positive. Potential stock
growth expected."
}
```

#### **External Libraries and Tools:**

## Web Scraping and HTTP

- Requests HTTP library for making API calls and web requests
- BeautifulSoup4 (bs4) Library for parsing HTML and XML documents

# Al and Machine Learning

• Google Generative AI - Library for accessing Google's Gemini AI model for text summarization, topic extraction, comparative analysis, sentiment analysis

## **Text-to-Speech and Translation**

- gTTS (Google Text-to-Speech) Python library and CLI tool to interface with Google Translate's text-to-speech API
- Googletrans Python library that implements Google Translate API for text translation

## Web Framework and API

- FastAPI Modern, high-performance web framework for building APIs
- Uvicorn ASGI server for running FastAPI applications
- Streamlit Web application framework for creating data apps

#### **Deployment Platform**

• Hugging Face Spaces - Platform for hosting and sharing machine learning applications