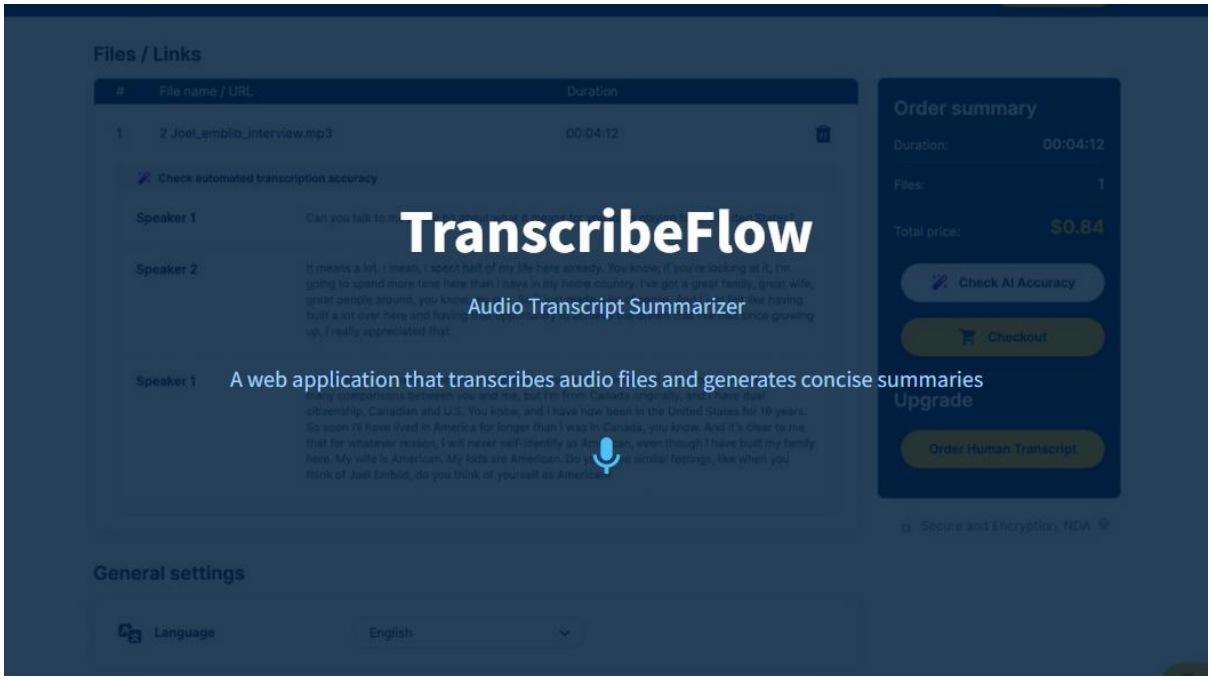
1. **Title:** TranscribeFlow: Audio Transcript Summarizer



2. **Project Statement:** Today's knowledge workers and students often need to quickly review long audio content, such as lectures, podcasts, or meeting recordings. Listening to an entire recording is inefficient when all that's needed is a summary of the key points. Manually transcribing and then summarizing is a time-consuming two-step process that is a significant barrier to productivity. "TranscribeFlow" aims to solve this by building a web application that takes an audio file, automatically transcribes it using an AI model, and then generates a concise summary of the transcript. This two-stage AI pipeline will allow users to rapidly extract the most important information from audio content, saving valuable time and making information more accessible.

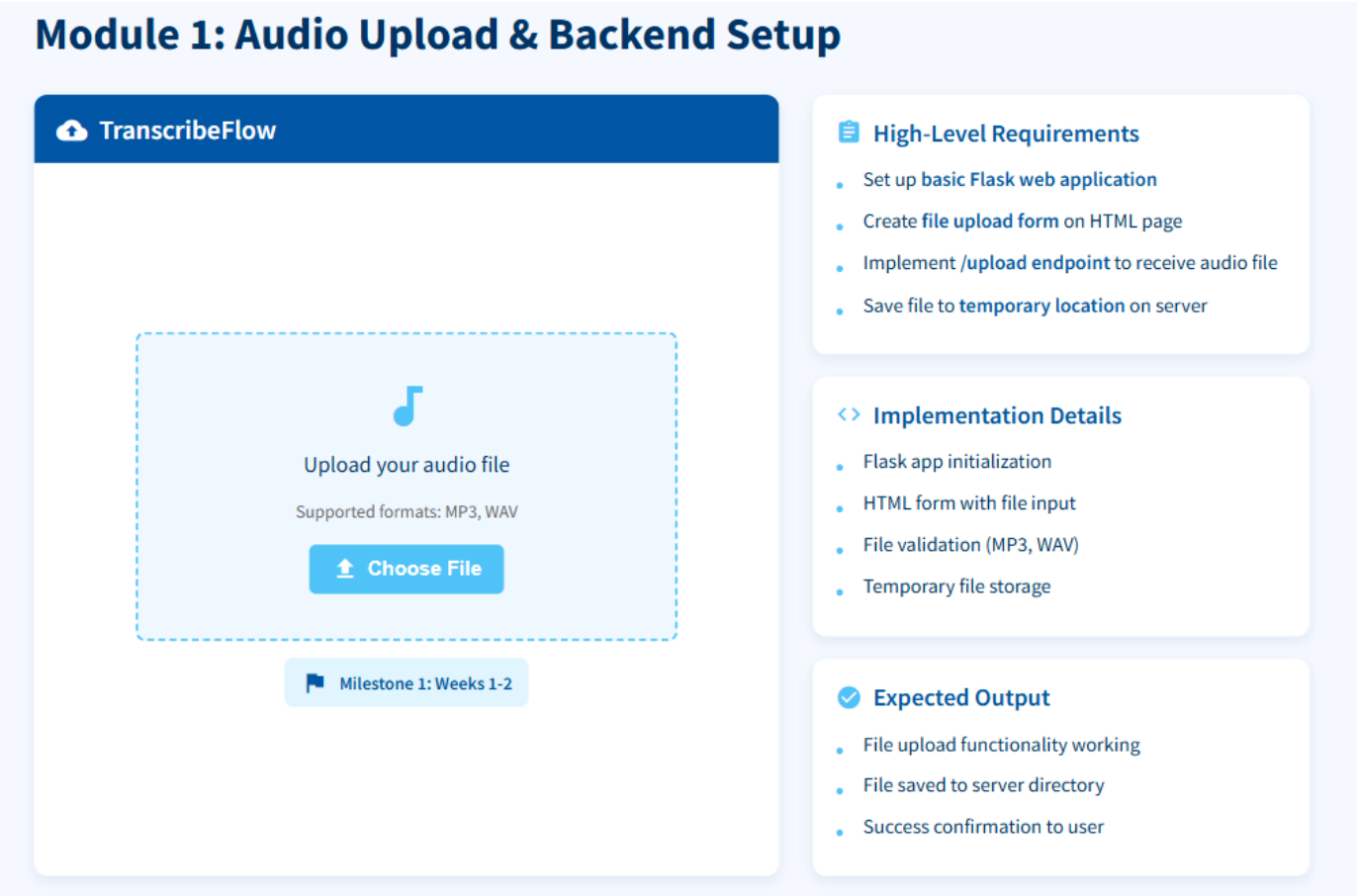
3. **Outcomes:**
    - **Audio-to-Text Transcription:** Accurately transcribe spoken audio from a file.
    - **Text Summarization:** Generate a concise summary of the transcribed text.
    - **User-Friendly Interface:** A simple web page for uploading an audio file and viewing the transcript and summary.
    - **Input Flexibility:** Capable of processing common audio formats (e.g., MP3, WAV).

4. **Modules to be Implemented:**
    - **Audio Upload & Processing Module:**
        - A web form on the front end for users to upload an audio file.
        - A backend API endpoint to receive and temporarily store the audio file.
    - **Core AI Inference Module (ASR & NLP):**
        - A pre-trained **Automatic Speech Recognition (ASR)** model (e.g., `Wav2Vec2` or `Whisper` from Hugging Face).

        - A pre-trained **text summarization model** (e.g., `T5` or `BART`).
        - Logic to run the ASR model to get the transcript, then run the summarization model on the transcript.
    - **Web Application Server:**
        - A Python web framework (e.g., Flask) to serve the front end and API endpoints.
        - A front end to display the transcript and the summary.
5. **Week-wise module implementation and high-level requirements with output screenshots:**

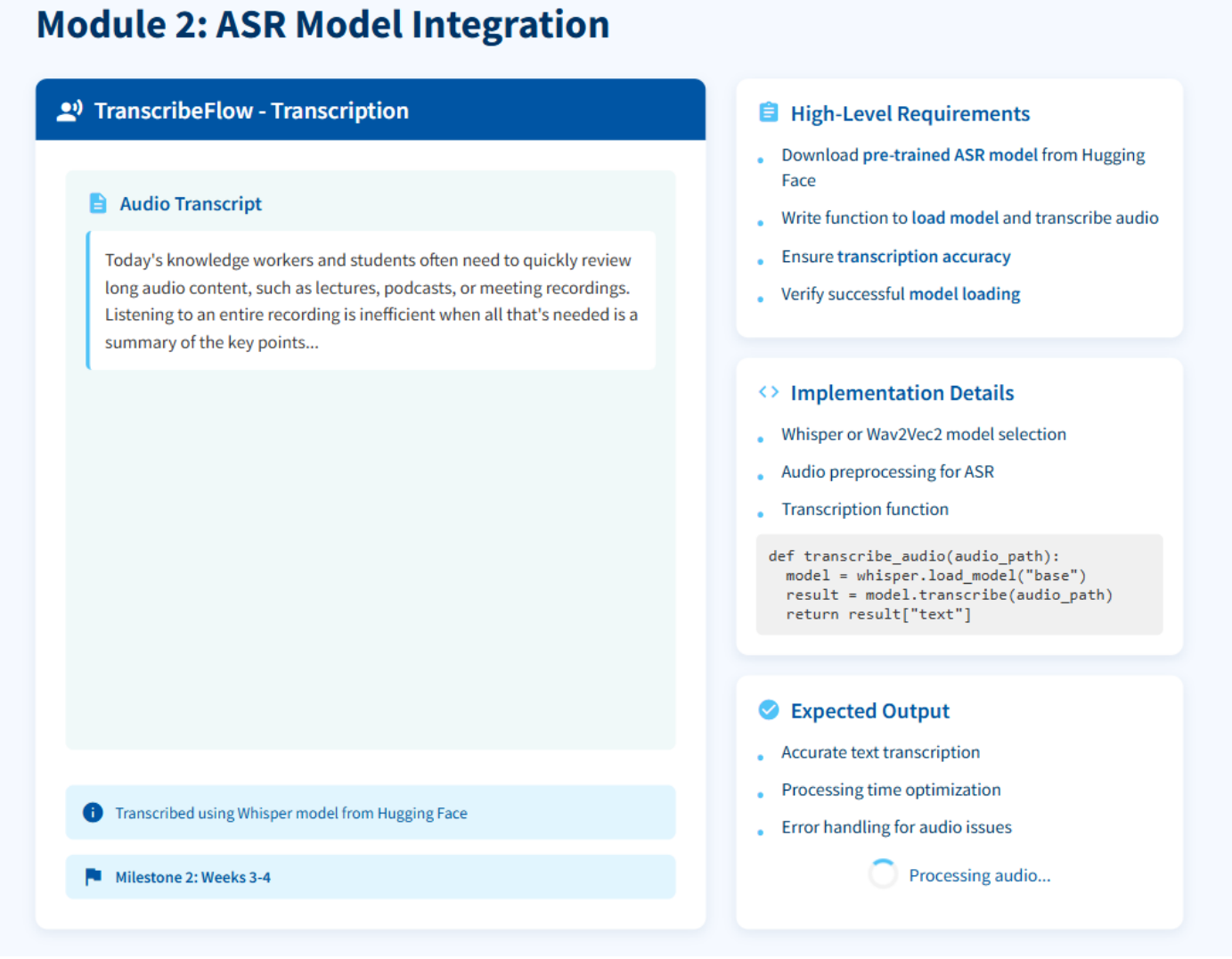
Milestone 1: Weeks 1-2

- **Module 1: Audio Upload & Backend Setup**
- **High-Level Requirements:**
    - Set up a basic Flask web application.
    - Create a file upload form on a single HTML page.
    - Implement a backend route (`/upload`) to receive the audio file and save it to a temporary location.



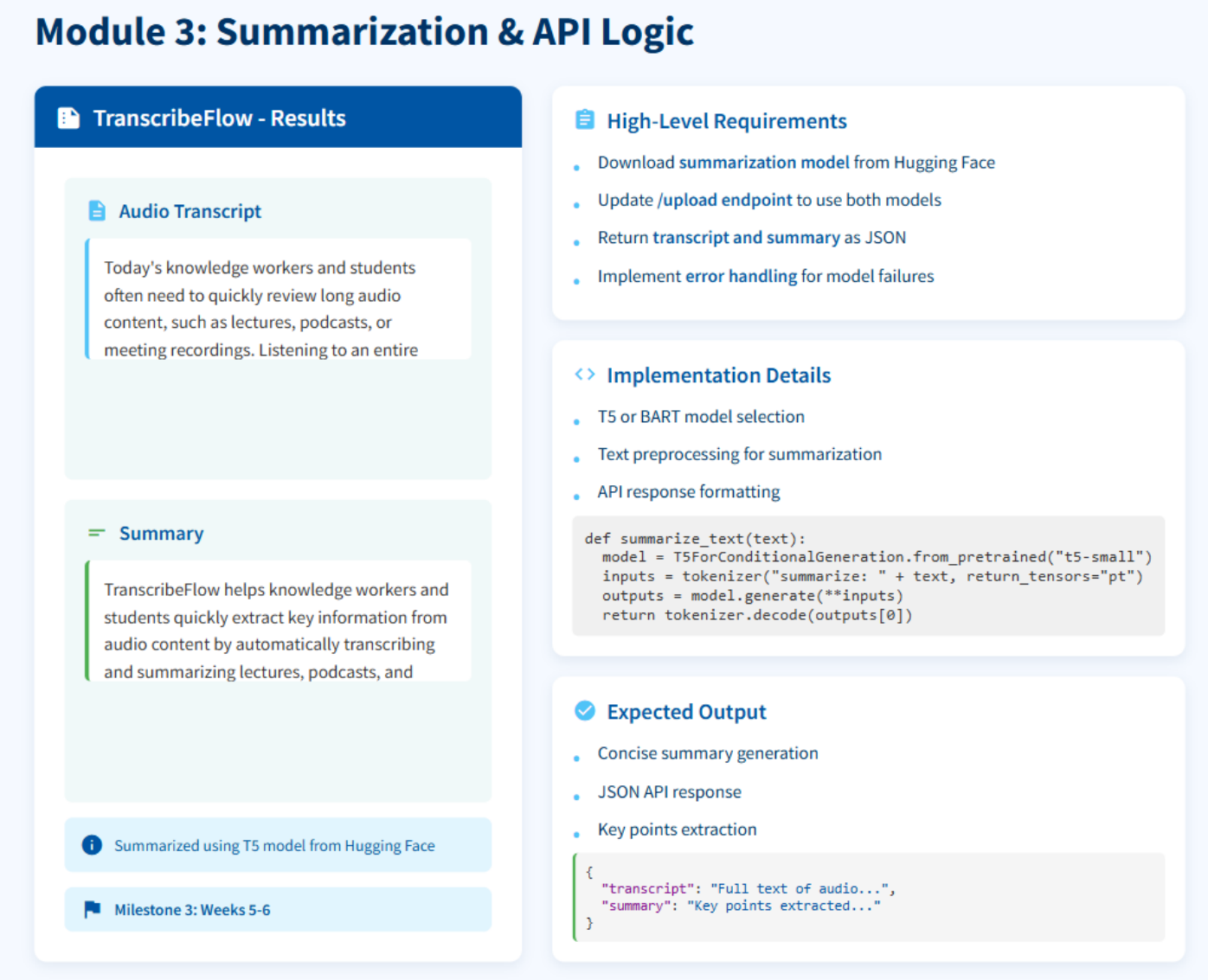
Milestone 2: Weeks 3-4

- **Module 2: ASR Model Integration**
- **High-Level Requirements:**
  - Download a pre-trained ASR model from Hugging Face.
  - Write a Python function to load the model and use it to transcribe a sample audio file.
  - Ensure the transcription is accurate and the model can be successfully loaded.



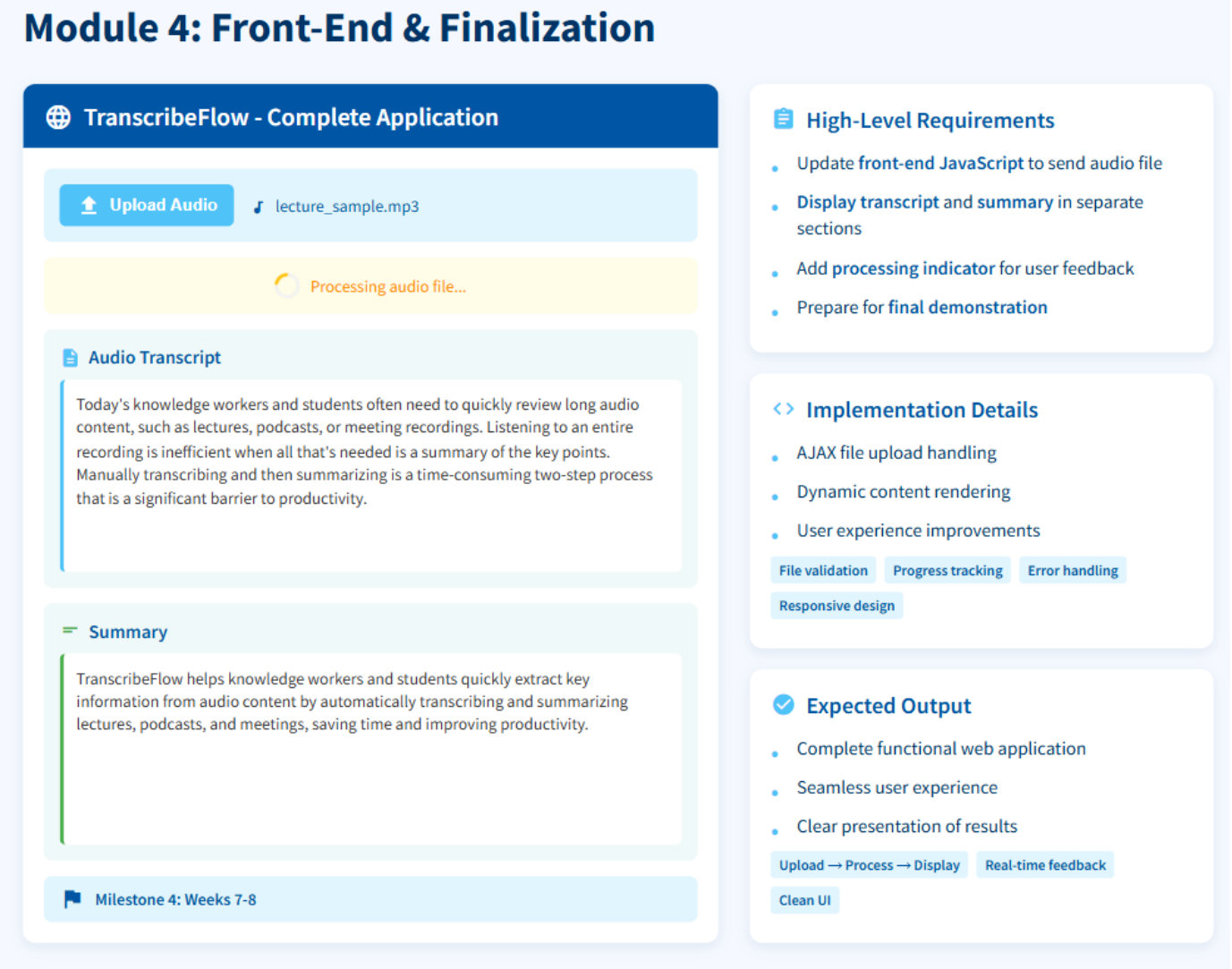
Milestone 3: Weeks 5-6

- **Module 3: Summarization & API Logic**
- **High-Level Requirements:**
  - Download a pre-trained summarization model from Hugging Face.
  - Update the `/upload` endpoint. After transcribing the audio, pass the transcript to the summarization model.
  - Return both the full transcript and the summary as a JSON response.



Milestone 4: Weeks 7-8

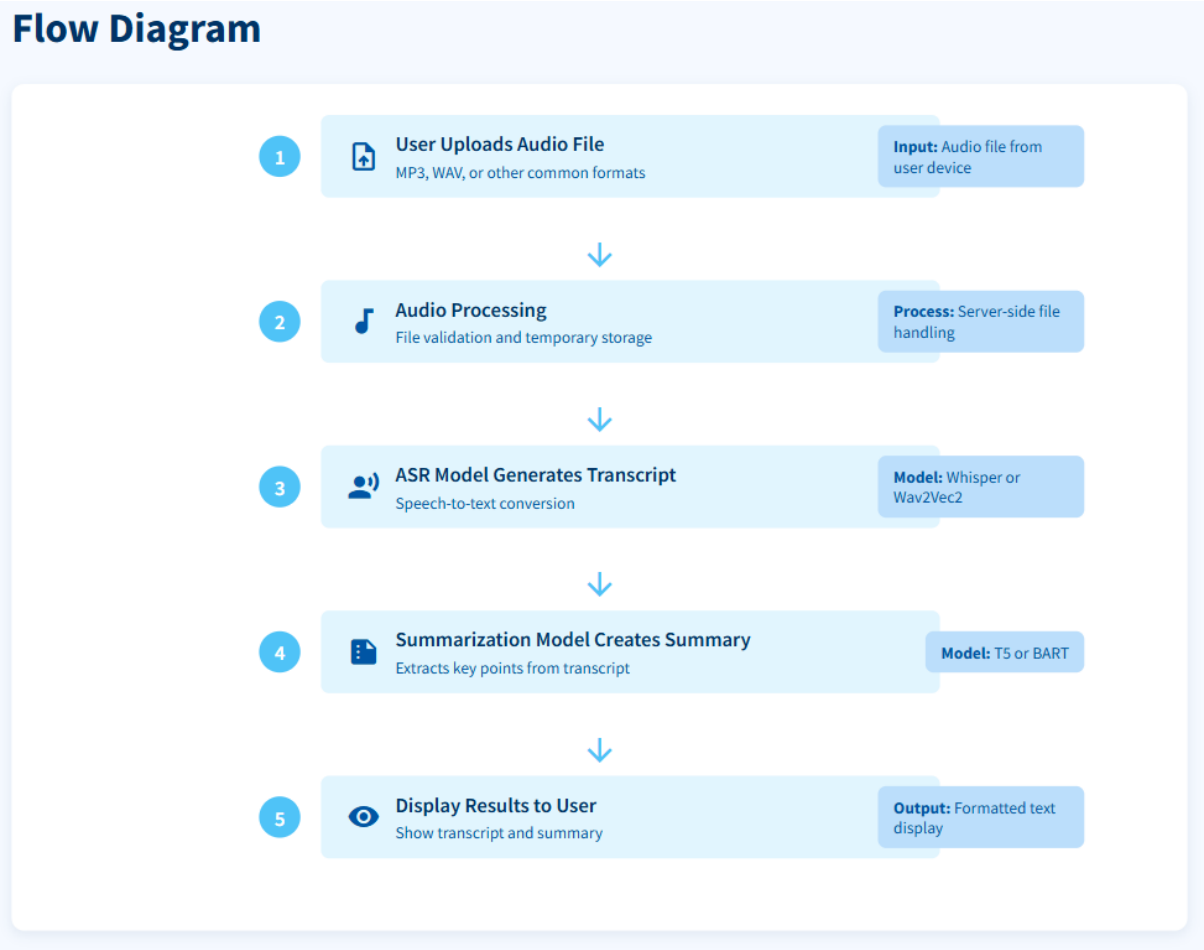
- **Module 4: Front-End & Finalization**
- **High-Level Requirements:**
  - Update the front-end JavaScript to send the audio file to the `/upload` endpoint.
  - Display the received transcript and summary on the web page in separate sections.
  - Add a "processing" indicator to provide user feedback.
  - Prepare the project for a final demonstration.



6. **Evaluation Criteria:**
   - **Milestone 1 Evaluation (End of Week 2):**
     - Audio file upload functionality is working and the file is saved to the server.
   - **Milestone 2 Evaluation (End of Week 4):**
     - The ASR model is integrated and can successfully transcribe a sample audio file.

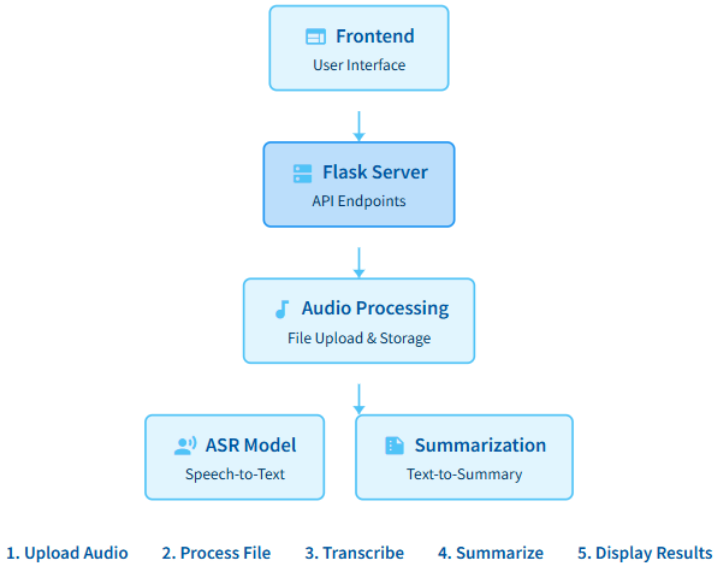
- **Milestone 3 Evaluation (End of Week 6):**
  - The summarization model works correctly and the API returns both the transcript and summary.
- **Milestone 4 Evaluation (End of Week 8):**
  - The full web application is functional, displaying the transcript and summary from an uploaded audio file.

7. **Workflow Diagram:**



8. **Architecture Diagram:**

Architecture Diagram



9. **Database Schema:**