Developer Home Assignment: Implementing a LiveKit Voice Pipeline with Audio Length Validation

Objective: Develop a voice assistant using LiveKit's Voice Pipeline Agent. Implement a mechanism that sends the length of the generated audio to a dedicated server before Text-to-Speech (TTS) processing. If the audio length exceeds 60 seconds, the server should return the audio trimmed to its middle segment.☐

Tasks:

1. **Set Up the LiveKit Voice Pipeline Agent:**
    - Clone and set up the LiveKit Voice Pipeline Agent demo from the LiveKit GitHub repository.☐
    - Ensure all dependencies are installed and the agent is operational.☐
2. **Implement the `before_tts_cb` Callback:**
    - Utilize the `before_tts_cb` function to intercept the text before it is sent to the TTS engine.☐
    - Within this callback, calculate the estimated length of the audio that the TTS will generate based on the input text.☐
    - Send this length information to a REST API endpoint on a dedicated server for validation.☐
3. **Develop a Backend Server with Flask:**
    - Create a separate backend server using Python's Flask framework.☐
    - Implement a REST API endpoint that receives the estimated audio length.☐
    - If the length exceeds 60 seconds, process the audio to trim it to its middle segment and return the modified audio.☐
    - If the length is 60 seconds or less, return the audio unchanged.☐
4. **Establish Communication Between the Agent and the Server:**
    - Ensure the LiveKit agent can communicate with the Flask server.☐
    - Use `ngrok` to expose the Flask server's local endpoint to the internet, facilitating communication during development and testing.☐
5. **Expose the Agent's Voice User Interface (UI):**
    - Set up the agent's voice UI to allow interaction and testing.☐
    - Ensure the UI is accessible for demonstration purposes.☐
6. **Document the Process and Create a Presentation Video:**
    - Document the setup process, code implementations, and any challenges encountered.☐
    - Create a short video (5-10 minutes) presenting the assignment, demonstrating the working solution, and explaining the code and architecture.☐

Guidelines:

- **Time Allocation:** The assignment is designed to be completed within 5 hours.☐
- **Use of Resources:** Utilizing Large Language Models (LLMs) and other online resources is permitted to assist with coding and problem-solving.☐

- **Submission:** Provide a link to a repository containing the source code and documentation. Include the presentation video link within the repository or as a separate attachment.☐

Evaluation Criteria:

- **Functionality:** The solution meets all specified requirements and functions correctly.☐
- **Code Quality:** Code is clean, well-documented, and follows best practices.☐
- **Problem-Solving:** Demonstrates effective problem-solving skills and the ability to overcome challenges.☐
- **Presentation:** The video is clear, concise, and effectively communicates the implementation and functionality of the solution.☐

Resources:

- LiveKit Agents Documentation
- Flask RESTful API Guide
- Ngrok Documentation

We look forward to your innovative solution and presentation.