**APIs and Controller for WordSmith**

**Overview**

The WordSmith app follows a **client-server architecture**, where the **Swift App Layer (front-end)** communicates with the **Swift Function Layer (engine)** to process user inputs, retrieve course content, and store AI-generated feedback. The **engine** acts as a middleware, handling requests to third-party APIs (ChatGPT, ElevenLabs) and managing interactions with the **database**.

**Front-End to Engine Communication**

The Swift app will use **HTTP requests** or **Swift functions** to communicate with the engine. The primary interactions include:

1. **Fetching courses and learning modules**
2. **Submitting user responses**
3. **Receiving AI-generated feedback**
4. **Retrieving and storing user progress**

**API Endpoints & Function Descriptions**

**1. getCourse(course\_id)**

* **Request:**
  + GET /api/course/{course\_id}
* **Response:**
  + Returns **course content, rubrics, and learning modules**.
* **Use Case:**
  + Fetches structured lessons when users navigate to a course.

**2. onNodeSubmission(node\_id, submission\_data)**

* **Request:**
  + POST /api/submit/{node\_id}
  + Body: { "submission": "User's response text" }
* **Response:**
  + Stores submission in the **database**.
* **Use Case:**
  + Saves user answers and progresses them through the course.

**3. getGPTFeedback(node\_id, submission\_id)**

* **Request:**
  + GET /api/feedback/{node\_id}/{submission\_id}
* **Response:**
  + Returns **AI-generated feedback** from ChatGPT API.
* **Use Case:**
  + Fetches evaluation and suggestions after user submission.

**4. saveGPTFeedback(submission\_id, feedback\_text)**

* **Request:**
  + POST /api/feedback/save
  + Body: { "submission\_id": "id", "feedback": "GPT response text" }
* **Response:**
  + Confirms that feedback is stored in the **database**.
* **Use Case:**
  + Ensures AI feedback is logged for future reference.

**5. convertTextToAudio(feedback\_text)**

* **Request:**
  + POST /api/audio/convert
  + Body: { "text": "Feedback text to convert" }
* **Response:**
  + Returns **MP3 file URL** generated by ElevenLabs API.
* **Use Case:**
  + Converts text feedback into voice for accessibility.

**6. progressSave(user\_id, node\_id, progress\_data)**

* **Request:**
  + POST /api/progress/save
  + Body: { "user\_id": "id", "node\_id": "id", "progress": "data" }
* **Response:**
  + Confirms that progress is saved.
* **Use Case:**
  + Saves the user's position within a course.

**Integration with Third-Party SDKs and APIs**

1. **ChatGPT API (gpt-4o, JSON mode)**
   * Used in getGPTFeedback for AI-driven analysis of submissions.
   * Data is sent in JSON format, processed, and returned as structured feedback.
2. **ElevenLabs API (Voice AI MP3)**
   * Used in convertTextToAudio to generate voice-based feedback.
   * Receives text and returns an MP3 file for playback.
3. **Database (Firebase/Supabase)**
   * Stores user submissions, feedback, and progress.
   * Accessed by getCourse, onNodeSubmission, saveGPTFeedback, and progressSave.

**Conclusion**

The **front-end communicates with the engine via API calls**, enabling seamless interaction between user submissions, AI-powered feedback, and database storage. The system ensures **structured learning, real-time feedback, and accessibility features** through **ChatGPT and ElevenLabs integrations**.