**Intelligent Language Tutor System Documentation**

### Overview

The Intelligent Language Tutor is an interactive, AI-driven system designed to teach any language through a structured, modular course. The system dynamically adapts lesson content and difficulty based on the learner’s proficiency, tracks progress through modules and submodules, and provides personalized revision by logging and reviewing common mistakes.

### System Architecture

#### 1. User Interface and Web Layer (further implementation)

**Frontend:** The system employs a web-based interface (using technologies such as React, HTML/CSS, and JavaScript) to interact with users. The interface supports user authentication (signup and login), displays lessons, captures responses, and presents feedback.

**Backend (Flask or FastApi):** The Flask backend exposes RESTful endpoints that manage:

**User Authentication:** Signup and login functionalities.

**Session Management:** Creating and retrieving user sessions based on language preferences and proficiency.

**Chat Interaction:** Managing the conversation flow (teaching, testing, evaluation) and progress tracking.

**Data Persistence:** Saving conversations and mistakes for later review and adaptive learning.

#### 2. Application Logic Layer

**Session Management:**  
When a user logs in, the system checks for an existing learning session. New users provide their learning language, native language, and proficiency level. This information initializes a session that tracks the current module and submodule.

**Conversation Flow:**  
The tutor’s interactions are divided into three phases:

**Teaching Phase:** The tutor presents the lesson content with examples and, for beginners, explicit key points and pronunciation guides.

**Testing Phase:** The tutor asks a text-based question directly related to the teaching content.

**Evaluation Phase:** The tutor evaluates the student’s response, provides corrective feedback, and logs any mistakes.

A state machine controls these phases. After a fixed number of cycles (e.g., five cycles per submodule), if mistakes have been logged, the system presents a revision summary; if not, the session advances to the next submodule (and ultimately to the next module).

**Module Progression:**

Each module comprises three submodules. A submodule is considered mastered when the student answers correctly in five evaluation cycles.Upon completion of a submodule, the system either provides revision feedback (if mistakes were made) or automatically advances to the next submodule or module.

#### 3. AI and Language Processing Layer

**Language Tutor (LLM Integration):**  
OpenAI’s GPT-3.5-turbo, integrated via LangChain, generates dynamic teaching content, testing questions, and evaluations. Different prompt templates are employed:

**Beginners:** Templates include explicit instructions, key points, and simplified language.

**Intermediate/Advanced:** Standard prompts are used without additional simplification.

**Context Management:**  
The conversation history is maintained via LangChain’s ConversationBufferMemory. If the history grows too long, a summarization function condenses previous interactions to ensure the LLM operates within token limits.

#### 4. Data Storage Layer

**Database (SQLite):**  
A SQLite database is used to persist all user-related data. The key tables include:

**Users:** Stores user credentials (username and hashed password).

**Sessions:** Tracks each user’s learning session, including language preferences, proficiency level, and current module/submodule.

**Conversations:** Logs each interaction between the tutor and the student.

**Mistakes:** Records details of mistakes (error type, incorrect response, correct response) and optionally includes module and submodule information for targeted revision.

### Conclusion

The Intelligent Language Tutor System provides an adaptive, personalized learning experience by integrating AI-powered language processing with robust session management and detailed progress tracking. Its modular architecture supports continuous evaluation and targeted revision based on logged mistakes. Future enhancements could include richer analytics, additional interactive features, and further adaptive learning strategies.