Introduction to Software Engineering

Test Plan

Software Engineering Department
Faculty of Information and Technology
University of Science

1.1 Objectives

The primary objective of this testing plan is to verify that the **English Speaking Learning System** functions correctly according to the requirements specified in Section 3, ensures data integrity, and meets the performance constraints (response time < 5 seconds). The testing process aims to identify and fix defects before the final deployment.

1.2 Testing Levels and Techniques

1.2.1 Unit Testing

- **Technique:** White-box testing.
- **Objects:**
  - **Backend Functions (Node.js):** Specifically the algorithms for calculating pronunciation scores and processing JSON feedback from the AI engine.
  - **Frontend Components (ReactJS):** Individual UI components like the *VoiceRecorder*, *AudioPlayer*, and *ResultChart*.
- **Goal:** Ensure that each specific module of code performs its logic correctly in isolation.

1.2.2 Integration Testing

- **Technique:** Black-box testing (API Testing).
- **Objects:**
  - **RESTful APIs:** Testing endpoints for User Authentication (Login/Register), Lesson Retrieval, and Audio Upload.
  - **External Service Integration:** Verifying the communication between the Backend and **Google Cloud Speech-to-Text API** to ensure audio is sent and transcripts are received correctly.
  - **Database Interaction:** Verifying that User, PracticeSession, and ExerciseAttempt data is correctly created, read, updated, and deleted in MySQL.
- **Goal:** Verify that different modules and services work together seamlessly.

1.2.3 System & Performance Testing

- **Technique:** Load testing and Functional testing.
- **Objects:** The entire web application deployed on the staging environment.

- **Goal:**
  + Validate the Non-Functional Requirement that voice analysis returns results within 3–5 seconds
  + Ensure the system handles concurrent users without crashing.
  + Verify the application is responsive on both Desktop and Mobile devices.

1.2.4 User Acceptance Testing (UAT)

- **Technique:** Beta testing with real users.
- **Objects:** End-to-end user flows (e.g., A user logs in -> Selects a Topic -> Records Audio -> Receives Feedback -> Checks History).
- **Goal:** Confirm the system meets the "Definition of Done" and satisfies the needs of real English learners.

1.3 Tools and Environment

- **Unit Testing Framework:** Jest (for React and Node.js logic).
- **API Testing:** Postman (for testing backend endpoints).
- **Performance Testing:** Google Lighthouse (for Frontend performance) and Apache JMeter (for load testing).
- **Browser Environment:** Google Chrome, Microsoft Edge (latest versions).

1.4 Success Criteria

The testing phase will be considered successful when:

- All critical and high-priority bugs are resolved.
- The speech recognition and scoring accuracy reaches >= **85%**.
- The average system response time for feedback is under **5 seconds**.