Introduction to Software Engineering

Requirements Analysis

The student team is required to complete the Software Requirements Specification (SRS) document for the assigned course project, following the attached template.

Table of Contents

Software Requirements Specification

Objectives

This document focus on the following topics:

- ✔ Complete the Software Requirements Specification (SRS) document with the following contents:
  - Elaborate on the Problem Statement
  - Overview of Requirements (Functional and Non-Functional), Stakeholders
  - Use Case Model
  - Use Case Specifications
  - Create Prototype and Mockup Diagrams of the System Interface
- ✔ Read and understand the requirements analysis document.

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127241	Đoàn Thành Phát	25%	
23127089	Nguyễn Quang Minh	25%	
23127085	Phạm Phát Lộc	25%	
23127102	Lê Quang Phúc	25%	

2 Problem Statement

The persistent global demand for practical English proficiency, particularly spoken fluency and accuracy, faces a bottleneck in accessible, quality training. In many markets, traditional learning methods and even modern mobile applications fall short in providing the precise, corrective feedback necessary for mastering challenging English phonology.

The proposed system, online English learning system focusing on the Speaking skill, aims to help learners practice pronunciation, speaking fluency, and conversational skills with automatic feedback, similar to how the ELSA application operates. All learning activities - including lessons, speech recording, pronunciation analysis, and performance tracking - are conducted entirely online through a web interface.

When a user first accesses the system, they must register and log in to create a personal learning account. Each learner has a profile that stores their name, current level, completed lessons, pronunciation scores, and progress data. After logging in, learners can choose from a list of speaking topics, such as *Greetings*, *Self-introduction*, *Work and Career*, *Travel*, or *Shopping*.

Each topic contains multiple lessons. In each lesson, the system displays an English sentence and provides an audio sample for the learner to listen to. The learner then clicks a record button to speak the same sentence aloud using their microphone. Once finished, the system uses speech recognition technology to convert the recorded voice into text, compare it with the target sentence, and calculate a pronunciation score based on similarity.

In addition to the overall score, the system performs detailed phoneme-level analysis, highlighting mispronounced sounds and giving suggestions for improvement. Results are presented visually so learners can easily identify which parts of the sentence were pronounced incorrectly.

After each practice session, learners can view a performance report that includes their score, pronunciation accuracy, list of errors, and specific feedback for each mistake. The system also maintains a learning history, allowing users to monitor their progress and improvement over time.

The system provides an optional random practice mode, where learners can be given random sentences at their level to improve their speaking reflexes.

To increase engagement, learners can earn badges or rankings based on their speaking performance and consistency. This gamification element motivates users to practice regularly.

From the administration side, system administrators can add, delete, or update lessons, manage user accounts, monitor learner statistics, and adjust lesson difficulty.

All voice data, learning records, and user information are securely encrypted to ensure privacy and data protection.

Operating Environment

The English Speaking Learning System will be developed as a web-based application. The system operates under the following environment:

- **Client-side (Frontend):**
  - Web browser supporting HTML5, CSS3, and JavaScript (e.g., Google Chrome, Microsoft Edge).
  - Audio recording support through Web Speech API or compatible browser audio APIs.
  - Responsive design compatible with desktop and mobile devices.
- **Server-side (Backend):**
  - Web server: Apache or Node.js-based server (Express framework).
  - Operating System: Windows
  - Database server: MySQL or MongoDB.
  - Speech processing services: Google Speech-to-Text API or OpenAI Whisper API for speech recognition.

The system requires a stable Internet connection to handle real-time voice data and cloud-based speech processing.

Design and Implementation Constraints

- **Programming Languages:**
  - Frontend: HTML5, CSS3, JavaScript, Vite, Tailwind, ReactJS,.
  - Backend: Node.js (Express).
- **Database:**
  - MySQL (relational model) or MongoDB (NoSQL document model).
- **Speech Recognition Engine:**
  - Web Speech API, Google Speech-to-Text, or OpenAI Whisper for pronunciation scoring and text conversion.
- **Development Tools:**
  - Visual Studio Code (IDE), Postman for API testing, GitHub for version control.
- **Documentation Standards:**
  - Use of IEEE standard documentation format for software requirement and design specifications.

- All code should follow clean coding standards and include inline comments for maintainability.

- **Security Constraints:**
  - All user credentials and learning data must be encrypted using HTTPS/TLS.
  - Input validation and sanitization must be applied to prevent injection attacks.
- **Performance Constraints:**
  - Voice analysis must be processed and returned within 3–5 seconds for smooth user experience.
  - Server must handle multiple concurrent users with scalable architecture.

English Speaking Learning System provides an interactive, AI-assisted platform for English learners to practice and improve their speaking skills. By combining modern web technologies, speech recognition, and real-time feedback, the system aims to enhance pronunciation accuracy, communication confidence, and learning motivation for users worldwide.

3 Requirements Overview

3.1 Stakeholders

STT	Stakeholder	Description
1	*Learner (User)*	The primary end-user of the system. Students, professionals, or ESL learners who use the application to practice English speaking, receive pronunciation feedback, and track their learning progress.
2	Administrator	The user responsible for managing the system's content and community. They have permissions to create/edit lessons, manage user accounts, and monitor system statistics.
3	Project Team (Dev Team)	The group of students responsible for analyzing, designing, developing, testing, and deploying the software.

4	Sponsor / Instructors	The Faculty of Information Technology (University of Science) and the lecturers who provide the project requirements, guidance, and evaluation.

3.2 Requirements

3.2.1. Functional Requirements Specification

1. Authentication & User Management Module

- The system must allow users to register a new personal account using their email and password.
- The system must allow users to log in to access their personal learning data.
- The system must maintain a user profile that stores the learner's name, current level, and progress data.
- The system must ensure user authentication is secured via JWT (JSON Web Tokens) or session management.

2. Learning & Practice Module (Core Features)

- The system must provide a list of speaking topics (e.g., Greetings, Travel, Work) containing multiple lessons.
- In each lesson, the system must display a target English sentence and provide a sample audio file for the learner to listen to.
- The system must allow learners to record their voice directly through the web interface using the "Record" button.
- The system must convert the recorded speech to text and compare it with the target sentence using speech recognition technology.
- The system must calculate and display an overall pronunciation score based on similarity.
- The system must perform detailed phoneme-level analysis to highlight specific mispronounced sounds and provide suggestions for improvement.
- The system must provide a "Random Practice Mode" to serve random sentences suitable for the learner's level.

3. Progress Tracking & Gamification Module

- The system must generate a performance report after each session, including score, accuracy, and a list of errors.
- The system must save learning history so users can review their progress over time.
- The system should implement gamification elements, such as awarding badges or rankings based on performance and consistency.

4. Administration Module

- The system must allow administrators to add, delete, or update topics and lesson content (sentences, audio).
- The system must allow administrators to manage user accounts (e.g., view list, disable accounts).
- The system must allow administrators to monitor learner statistics.

3.2.2. Non-Functional Requirements Specification

1. Performance Requirements

- The voice analysis process (recording upload, processing, and feedback return) must be completed within 3–5 seconds to ensure a smooth user experience.
- The speech recognition accuracy should be maintained above 85%.
- The system must utilize a scalable architecture capable of supporting at least 1,000 concurrent users.

2. Security Requirements

- All user credentials and sensitive learning data must be encrypted in the database and during transmission using HTTPS/TLS.
- The system must implement input validation and sanitization to prevent injection attacks (e.g., SQL Injection, XSS).

3. Interface & Usability Requirements

- The application must be developed as a web-based application.
- The user interface must utilize Responsive Design, ensuring compatibility with both desktop computers and mobile devices.

- The frontend must support modern web browsers (Chrome, Edge) with HTML5, CSS3, and JavaScript support.

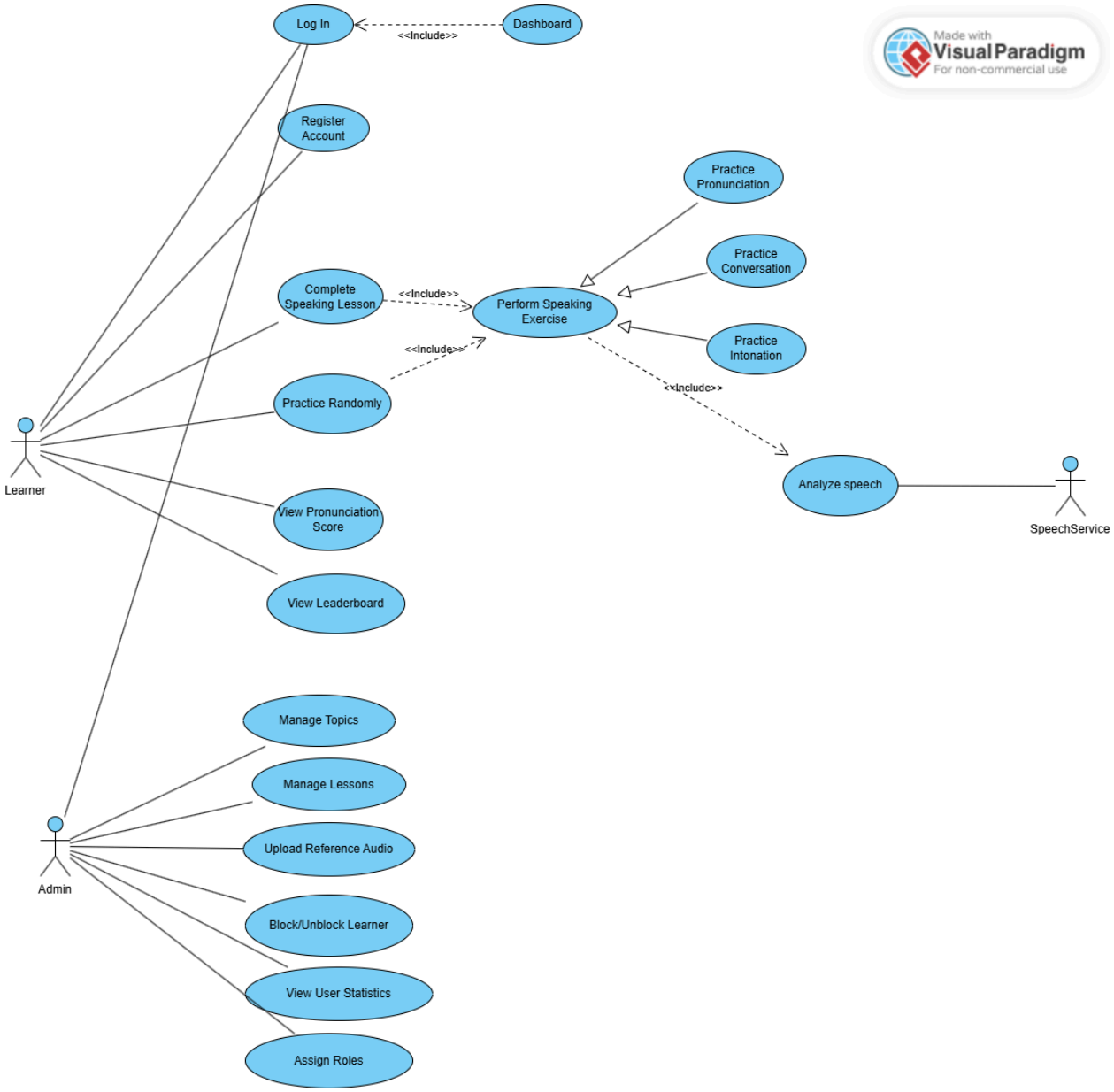
4. Reliability & Availability

- The system requires a stable Internet connection to operate, as speech processing is cloud-based.

- The system backend must be deployed on a reliable cloud platform to ensure high availability.

4 Requirements Analysis

4.1 Use Case model



4.2 Use Case Specification

4.2.1. Use Case Register Account

Use case ID	**U001**
Use Case	Register Account
Brief Description	A new user (Learner) creates a personal learning account to access the system and track their progress.
Actor	Learner
Pre-Condition	The user is not logged in.
Result	A new Learner account and personal profile are created in the database. The user is logged in.

Main Scenario	1. The Learner selects the "Register" option. 2. The system displays a form for required information (e.g., Name, Email, Password). 3. The Learner fills in the information and submits. 4. The system validates the data (e.g., valid email format, strong password) and checks that the email does not already exist. 5. The system creates a new user account and an associated personal profile in the database (e.g., MySQL). 6. The system automatically logs the Learner in.
Alternative Scenarios	**A1.** Email already exists. The system displays an error: "This email is already in use." **A2.** Invalid data. The system displays an error message (e.g., "Invalid email format," "Password is too weak").
Non-Functional Constraints	**Security:** All user credentials must be encrypted . **Security:** Input validation and sanitization must be applied to prevent injection attacks. **Usability:** Responsive design compatible with desktop and mobile devices.

4.2.2. *Use Case Log in*

Use case ID	**U002**
Use Case	Log In
Brief Description	An existing user (Learner or Administrator) provides credentials to gain secure access to the system.
Actor	Learner, Administrator
Pre-Condition	The user has an existing, registered account. The user is currently on the login page.
Result	(Success) The system authenticates the user, creates a session, and redirects them to the appropriate dashboard. (Failure) The user remains on the login page and receives an error message.

Main Scenario	1. The system displays a form for "Email" and "Password". 2. The user enters their credentials. 3. The user clicks the "Log In" button. 4. The system sends the credentials to the backend (e.g., Node.js) for verification. 5. The backend validates the credentials against the database (e.g., MySQL). 6. Credentials are correct. The system creates a user session 7. The system redirects the user to their main dashboard based on their role (Learner or Admin).
Alternative Scenarios	**A1.** Invalid credentials (wrong email or password). The system displays an error: "Invalid email or password." **A2.** Empty fields. The system displays an error: "Please fill in all fields."
Non-Functional Constraints	**Security:** The login process must use HTTPS/TLS **Security:** User credentials and information must be securely encrypted.

4.2.3. *Use Case Complete Speaking Lesson*

Use case ID	**U003**
Use Case	**Complete Speaking Lesson**
Brief Description	The learner selects a specific lesson from a topic and sequentially practices all sentences or exercises contained within it.
Actor	Learner
Pre-Condition	The learner is logged in. The lesson content exists in the database.
Result	All exercises in the lesson are attempted. The lesson is marked as completed in the learning history.

Main Scenario	1. The learner selects a Lesson from the Topic list. 2. The system retrieves the list of exercises for this lesson. 3. **Loop:** For each exercise item in the list: a. The system determines the type of exercise (Pronunciation, Conversation, or Intonation). b. **The system executes the "Perform Speaking Exercise" (U018) use case.** c. The system saves the score for that item. 4. Once all items are finished, the system calculates the average score. 5. The system displays the Lesson Summary Report.
Alternative Scenarios	**A1. Empty Lesson:** If the lesson has no content, the system displays "This lesson is under construction" and returns to the topic list. **A2. Learner Exits Early:** The system saves progress of completed items only.
Non-Functional Constraints	**Performance:** Lesson content must load within 2 seconds.

4.2.4. *Use Case Practice Randomly*

Use case ID	**U004**
Use Case	**Practice Randomly**
Brief Description	The system randomly selects practice materials for the learner to practice endlessly or until they choose to stop.
Actor	Learner
Pre-Condition	The learner is logged in.
Result	Practice history is saved per sentence attempted.

Main Scenario	1. The learner clicks the "Quick Practice" button. 2. The system randomly fetches a sentence or word from the database (based on user level). 3. **The system executes the "Perform Speaking Exercise" (U018) use case.** 4. After the feedback is shown, the system displays a "Next" button. 5. Steps 2-4 repeat until the learner clicks "End Session". 6. The system displays a session summary.
Alternative Scenarios	**A1. Network Error:** If the system cannot fetch a random question, display "Connection lost. Please check your internet."
Non-Functional Constraints	**Randomness:** The system should not repeat the same sentence within a session of 10 items.

4.2.5. Use Case Perform Speaking Exercise

Use case ID	U005
Use Case	**Perform Speaking Exercise**
Brief Description	An abstract use case defining the core workflow for any speaking activity: displaying prompts, recording audio, and triggering analysis.
Actor	Learner
Pre-Condition	Microphone permission is granted.
Result	Audio is recorded, processed, and feedback is displayed.

Main Scenario	1. The system displays the content prompt (Word, Sentence, or Dialogue context). 2. The system activates the microphone. 3. The learner speaks and the system captures audio. 4. The learner clicks "Stop" (or silence is detected). 5. **The system includes "Analyze Speech" to process the audio.** 6. The system receives the analysis result (Score, Errors) from the Speech Service. 7. The system displays the feedback UI.
Alternative Scenarios	**A1. Microphone Access Denied:** The system displays a prompt asking the user to enable permissions. **A2. Audio Too Short:** The system warns "Recording too short" and asks the learner to try again.
Non-Functional Constraints	**Reliability:** Audio recording must not crash the browser even if the session is long.

4.2.6. Use Case Practice Pronunciation

Use case ID	U006
Use Case	**Practice Pronunciation**
Parent Use Case	Inherits from U018 (Perform Speaking Exercise)
Brief Description	The learner practices pronouncing specific words or single sentences. Feedback focuses on phoneme-level accuracy.
Actor	Learner
Main Scenario	1. **(Inherits Step 1 of U018):** System displays Target Word + IPA (e.g., "Hello" /həˈloʊ/). 2. **(Inherits Steps 2-4 of U018):** Learner records audio.

	3. **(Inherits Step 5-6 of U018):** System analyzes speech. 4. **Specific Feedback:** The system highlights incorrect letters in Red (e.g., the 'l' sound) and provides a "Listen to your error" feature.
Alternative Scenarios	**A1. Skip Word:** Learner chooses to skip the difficult word. System moves to the next word without saving a score.
Non-Functional Constraints	**Accuracy:** Feedback must pinpoint the exact syllable that was mispronounced.

4.2.7. Use Case Practice Conversation

Use case ID	**U007**
Use Case	**Practice Conversation**
Parent Use Case	Inherits from U018 (Perform Speaking Exercise)

Brief Description	The learner engages in a role-play dialogue where the system acts as a conversation partner.
Actor	Learner
Main Scenario	1. The system displays context and assigns Role B to the learner. 2. The system plays audio for Role A. 3. **(Inherits Steps 2-4 of U018):** The system prompts the learner to speak Role B's line and captures audio. 4. **Specific Logic:** - If Score > 70%: System accepts and moves to the next turn. - If Score < 70%: System prompts "Try again". 5. The process repeats until the dialogue ends.
Alternative Scenarios	**A1. Use Hint:** Learner presses "Hint" to hear the sample audio for Role B before recording.

Non-Functional Constraints	**Latency:** The delay between the learner finishing speaking and Role A responding must be under 1 second.

4.2.8.Use Case Practice Intonation

Use case ID	**U008**
Use Case	**Practice Intonation**
Parent Use Case	Inherits from U018 (Perform Speaking Exercise)
Brief Description	The learner practices the rhythm, stress, and pitch of sentences.
Actor	Learner
Main Scenario	1. **(Inherits Step 1 of U018):** System displays the sentence with visual stress markers (e.g., "Where are you **GO**ing?").

	2. **(Inherits Steps 2-4 of U018):** Learner records audio. 3. **Specific Feedback:** The system displays a **Pitch Contour Graph** (Waveform) overlaying the learner's voice on the native speaker's voice.
Alternative Scenarios	**A1. Playback Comparison:** User clicks "Compare" to hear their recording played simultaneously with the native audio.
Non-Functional Constraints	**Visualization:** The waveform graph must be responsive and smooth on mobile devices.

4.2.9.Use Case Analyze Speech

Use case ID	**U009**
Use Case	**Analyze Speech**

Brief Description	The system processes the raw audio file using an external Speech Service to generate transcripts, alignment data, and proficiency scores.
Actor	**SpeechService** (External System)
Pre-Condition	A valid audio file is captured in the temporary buffer. Internet connection is active.
Result	A structured data object (JSON) containing Transcript, Phoneme list, Scores (Fluency, Intonation, Pronunciation), and Timestamps is returned.

Main Scenario	1. The system constructs an API request containing the **User Audio File** and the **Target Text** (the sentence the user was supposed to say). 2. The system sends the request to the **SpeechService** API. 3. The **SpeechService** performs Speech-to-Text (STT) conversion to generate a transcript. 4. The **SpeechService** performs **Phoneme Alignment** to map user sounds to the target phonemes. 5. The **SpeechService** calculates metrics: Word Error Rate (WER), Intonation match, and Fluency score. 6. The **SpeechService** returns the analysis results to the system. 7. The system parses the result and formats it for display.
Alternative Scenarios	**A1. API Timeout:** The system does not receive a response within 5 seconds. The system performs one retry. If it fails again, it displays "Analysis Failed. Please check your network." **A2. No Speech Detected:** The SpeechService returns an empty transcript (user recorded silence). The system prompts the user: "We couldn't hear you. Please speak louder."

Non-Functional Constraints	**Performance:** Total processing time (round-trip) must be under **3 seconds** to maintain a smooth user experience. **Security:** Audio data sent to the API must be encrypted via HTTPS/TLS.

4.2.10.Use Case View Pronunciation Score

Use case ID	**U010**
Use Case	**View Pronunciation Score**
Brief Description	The learner views a comprehensive analysis of their speaking skills, including an overall proficiency score (e.g., ELSA Score) and a breakdown of specific skills such as Intonation, Fluency, and Pronunciation of specific phonemes.
Actor	Learner
Pre-Condition	The learner is logged in and has completed enough lessons for the system to generate an analysis.
Result	A detailed skill dashboard (often visualized as a radar chart) is displayed to the learner.

Main Scenario	1. The learner navigates to the "Your Skills" or "Profile" tab. 2. The system calculates the aggregate score based on recent performance. 3. The system displays the Overall Score (e.g., "85% Native-like"). 4. The system displays a breakdown chart (Radar Chart) covering: Pronunciation, Stress, Intonation, Fluency, and Listening. 5. The system lists specific "Top Phoneme Challenges" (e.g., /th/, /r/) that the user struggles with. 6. The learner clicks on a specific skill to see detailed advice.
Alternative Scenarios	**A1. Not enough data:** If the learner is new, the system displays: "Complete at least 5 lessons to unlock your Skill Score." **A2. Outdated data:** If the learner hasn't practiced in a while, the system displays a "Score Decay" warning and encourages practice to refresh the score.

Non-Functional Constraints	**Visualization:** The charts must be rendered clearly and be responsive on mobile screens. **Performance:** Calculations for aggregated scores must happen on the backend and load within 2 seconds.

4.2.11. Use Case View Leaderboard

Use case ID	**U011**
Use Case	**View Leaderboard**
Brief Description	The learner views a ranking list of users based on Experience Points (XP) or lessons completed to compare their progress with friends or the global community.
Actor	Learner
Pre-Condition	The learner is logged in.
Result	A list of users sorted by rank is displayed. The learner sees their own position on the list.

Main Scenario	1. The learner selects the "Community" or "Leaderboard" menu. 2. The system retrieves the current rankings from the database. 3. The system displays the top learners (e.g., Top 10) and the learner's current rank. 4. The learner toggles the filter between "Global", "Friends", or "This Week/All Time". 5. The system refreshes the list based on the selected filter. 6. The learner taps on another user's name to view their public profile summary.

Alternative Scenarios	**A1. No Friends:** If the learner selects "Friends" filter but has no connections, the system displays: "Add friends to compete with them!" **A2. Privacy Mode:** If a user has set their profile to 'Private', they will not appear on the Global Leaderboard, and the system notifies them of this visibility setting.
Non-Functional Constraints	**Scalability:** The leaderboard query must be optimized to handle thousands of concurrent users without slowing down the database. **Real-time:** Rankings should be updated in near real-time (or at least cached and updated every 5-10 minutes).

4.2.12. *Use Case Manage Topics*

Use case ID	**U012**
Use Case	**Manage Topics**

Brief Description	The administrator creates, updates, or deletes learning topics (categories) to organize lessons logically.
Actor	Administrator
Pre-Condition	The administrator is authenticated and has content management privileges.
Result	Topic list is updated in the database. Changes are reflected in the Learner's topic selection screen.
Main Scenario	1. The administrator navigates to the "Content Management" section. 2. The system displays a list of existing topics. 3. The administrator selects "Add New Topic" or selects an existing topic to edit. 4. The administrator enters/updates topic details (Name, Description, Difficulty Level, Thumbnail Image). 5. The administrator clicks "Save".

	6. The system validates the input data and saves changes to the database.
Alternative Scenarios	**A1. Duplicate Topic Name:** The system detects an existing topic with the same name. It displays an error and asks for a different name. **A2. Delete Topic with Lessons:** If the admin deletes a topic containing lessons, the system warns: "Cannot delete. Please remove lessons first."
Non-Functional Constraints	**Performance:** List of topics must load within 2 seconds.

4.2.13. Use Case Manage Lessons

Use case ID	**U013**
Use Case	**Manage Lessons**
Brief Description	The administrator adds, edits, or removes specific lessons within a topic, including defining the text script for learners to practice.

Actor	Administrator
Pre-Condition	The administrator is authenticated. At least one Topic must exist.
Result	Lesson content (text script) is stored in the database, linked to a specific Topic.
Main Scenario	1. The administrator selects a specific Topic from the list. 2. The system displays lessons within that Topic. 3. The administrator chooses to "Create Lesson". 4. The administrator inputs the Lesson Title and adds a list of sentences (text script) to be practiced. 5. The administrator clicks "Save Draft" or "Publish". 6. The system validates the input and saves the lesson structure.

Alternative Scenarios	**A1. Empty Lesson:** The admin tries to save a lesson without any sentences. The system shows an error. **A2. Character Limit:** If a sentence is too long, the system warns the admin to split it.
Non-Functional Constraints	**Usability:** The interface should support bulk-importing sentences via text file.

4.2.14.Use Case Upload Reference Audio

Use case ID	U014
Use Case	**Upload Reference Audio**
Brief Description	The administrator uploads native speaker audio files corresponding to the sentences in a lesson for the system to use as a benchmark.
Actor	Administrator

Pre-Condition	The administrator is editing a Lesson. The target sentence text is defined.
Result	Audio file is stored in cloud storage and linked to the sentence record.
Main Scenario	1. The administrator selects a specific sentence within the Lesson Editor. 2. The administrator clicks the "Upload Audio" button. 3. The system opens a file explorer dialog. 4. The administrator selects an audio file (MP3/WAV) and confirms. 5. The system validates the file format and size. 6. The system uploads the file to the media server and links it to the sentence.
Alternative Scenarios	**A1. Invalid Format:** The admin uploads a non-audio file. System displays: "Invalid format. Use MP3/WAV." **A2. File Too Large:** The file exceeds the limit (e.g., 5MB). System rejects the upload.

Non-Functional Constraints	**Performance:** Upload processing should not take more than 5 seconds per file.

4.2.15.Use Case Block/Unblock Learner

Use case ID	**U015**
Use Case	**Block/Unblock Learner**
Brief Description	The administrator restricts or restores a learner's access to the system, typically used for moderating spam or policy violations.
Actor	Administrator
Pre-Condition	The administrator is authenticated. The target user exists.
Result	The user's status is updated to 'Blocked' or 'Active'.

Main Scenario	1. The administrator searches for a user in the User Management interface. 2. The system displays the user's profile and current status. 3. The administrator clicks the "Block" (or "Unblock") button. 4. The system prompts for confirmation and an optional reason. 5. The administrator confirms. 6. The system updates the status and sends an email notification to the user.
Alternative Scenarios	**A1. Self-Block:** If the admin tries to block themselves, the system denies the action. **A2. User already in state:** System notifies that the user is already blocked/active.
Non-Functional Constraints	**Security:** Only Admins with specific user-management privileges can perform this.

4.2.16.Use Case View User Statistics

Use case ID	U016
Use Case	**View User Statistics**
Brief Description	The administrator views aggregated data about user activities, including new registrations, active users, and completion rates.
Actor	Administrator
Pre-Condition	The administrator is authenticated.
Result	Statistical charts and tables are displayed on the Admin Dashboard.

Main Scenario	1. The administrator navigates to the "Analytics" tab. 2. The system fetches usage data from the database. 3. The system renders charts (e.g., Daily Active Users, Retention Rate). 4. The administrator applies filters (e.g., "Last 7 days"). 5. The system refreshes the view based on the filter. 6. The administrator can export data to CSV.
Alternative Scenarios	**A1. No Data:** If the date range has no activity, system displays "No data found."
Non-Functional Constraints	**Accuracy:** Data should be updated in near real-time (max 1 hour delay).

4.2.17.Use Case Assign Roles

Use case ID	**U017**

Use Case	Assign Roles
Brief Description	The administrator assigns or modifies the system role of a user (e.g., promoting a Learner to Content Editor or Admin).
Actor	Administrator (Root/Super Admin)
Pre-Condition	The actor must be logged in as a Super Administrator.
Result	The user's permission level is updated in the database.
Main Scenario	1. The Super Admin opens the profile of a specific user. 2. Selects the "Edit Role" option. 3. The system displays a dropdown of roles (Learner, Editor, Admin). 4. The Super Admin selects the new role. 5. The system requests password re-entry for security. 6. The system saves the new role and logs the event.

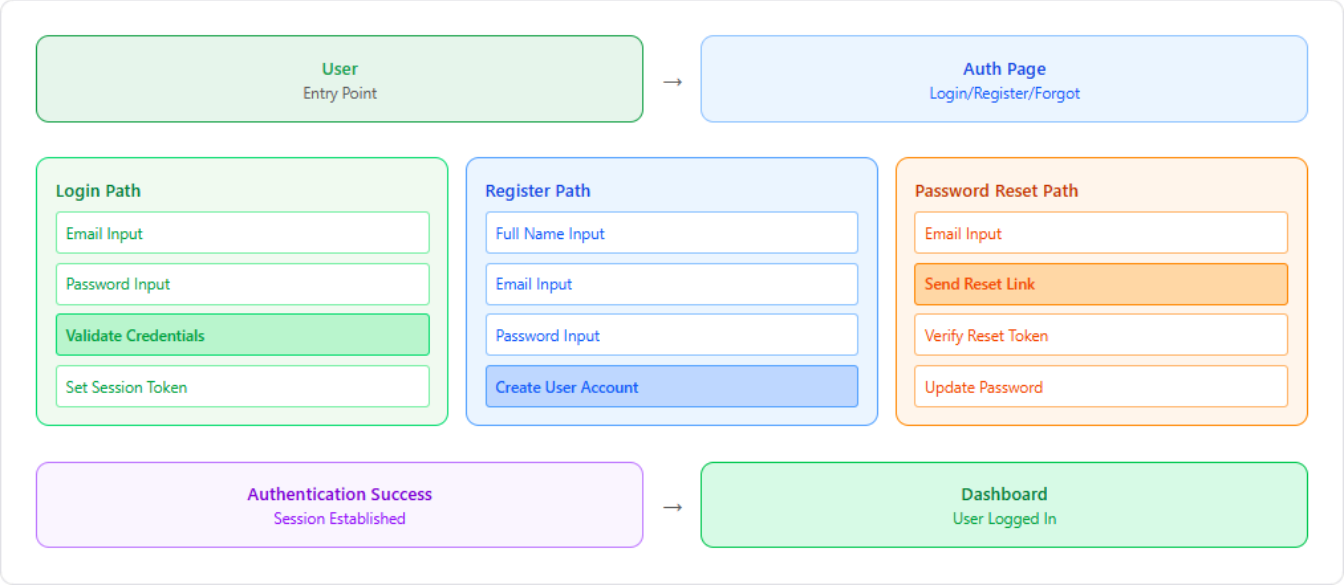
Alternative Scenarios	**A1. Downgrade Self:** System prevents removing one's own admin rights. **A2. Auth Fail:** Incorrect password prevents the role change.
Non-Functional Constraints	**Security:** Role changes must force a token refresh for the target user if logged in.

5 Prototype/Mockup

5.1.Wireframe diagrams

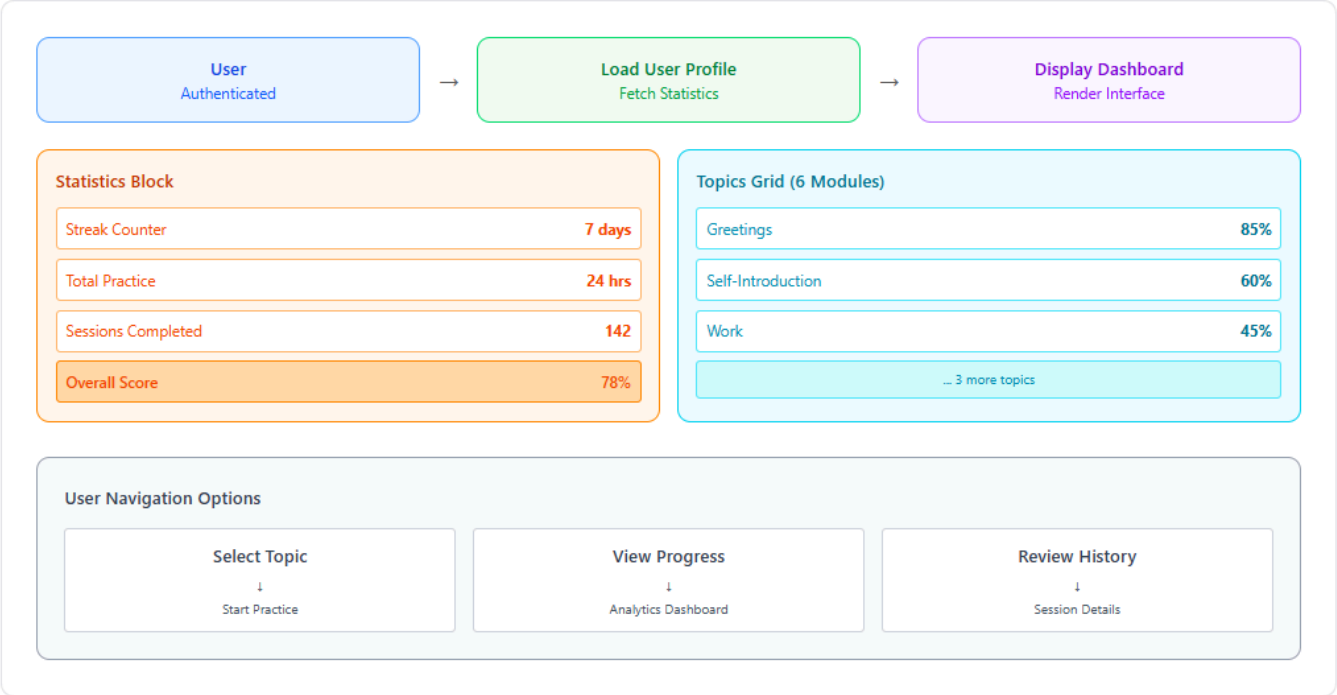
5.1.1. Authentication Flow

Login, register, and forgot password screens with form fields and social login options



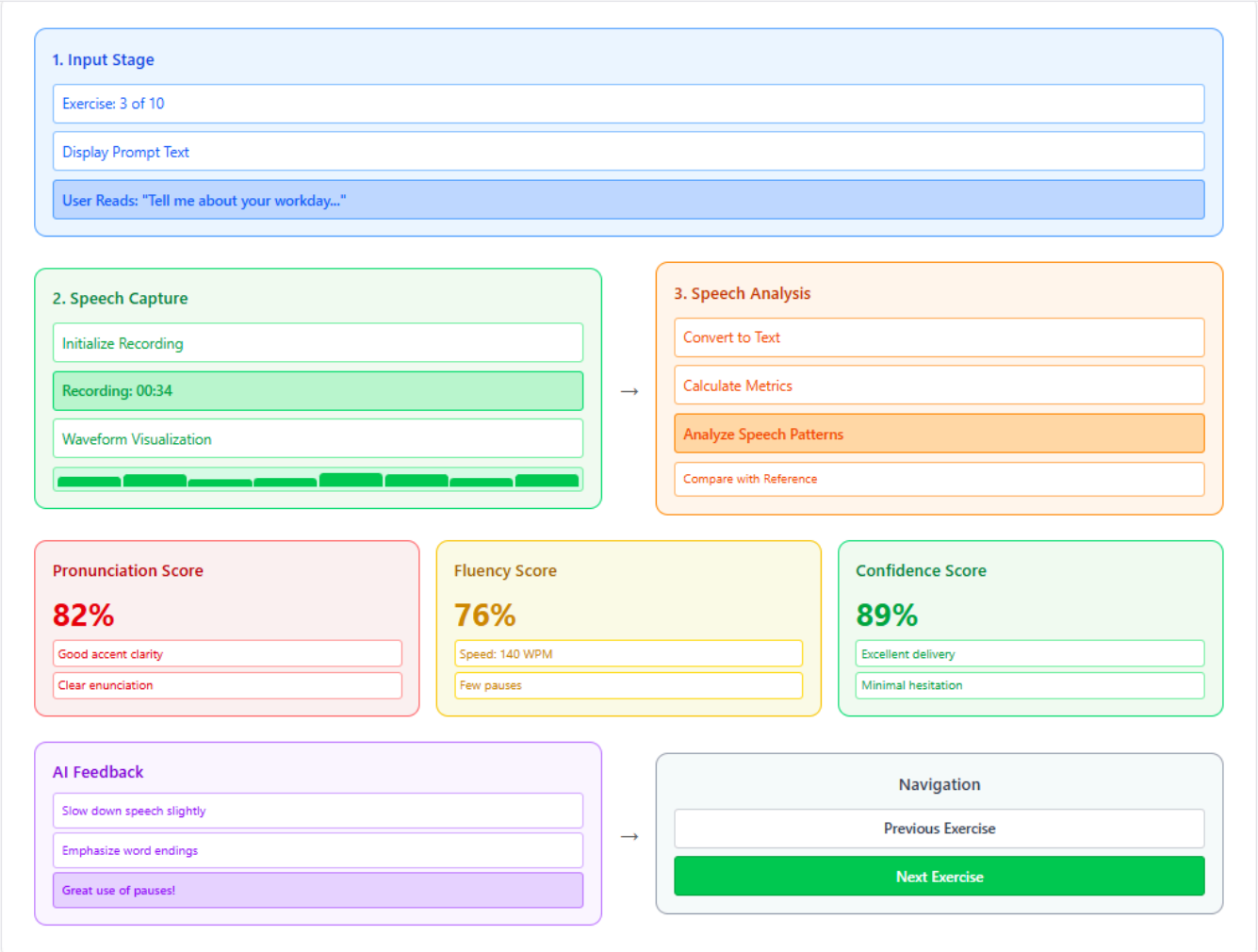
5.1.2. Dashboard Interface

Welcome message, streak counter, progress statistics, and topic grid with 6 learning topics



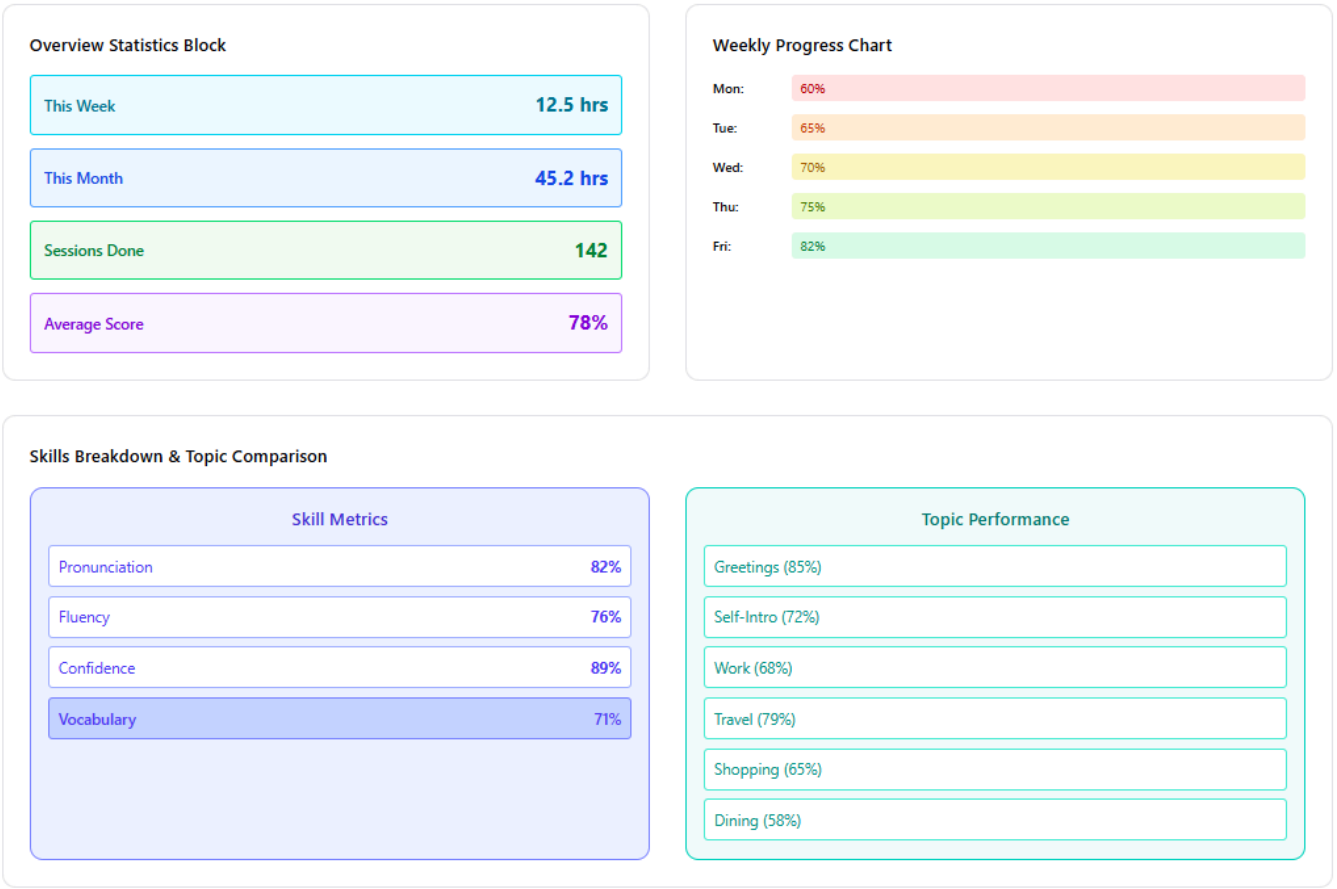
5.1.3. Practice Session Interface

Exercise counter, timer controls, text prompts, speech recording with waveform, and real-time feedback



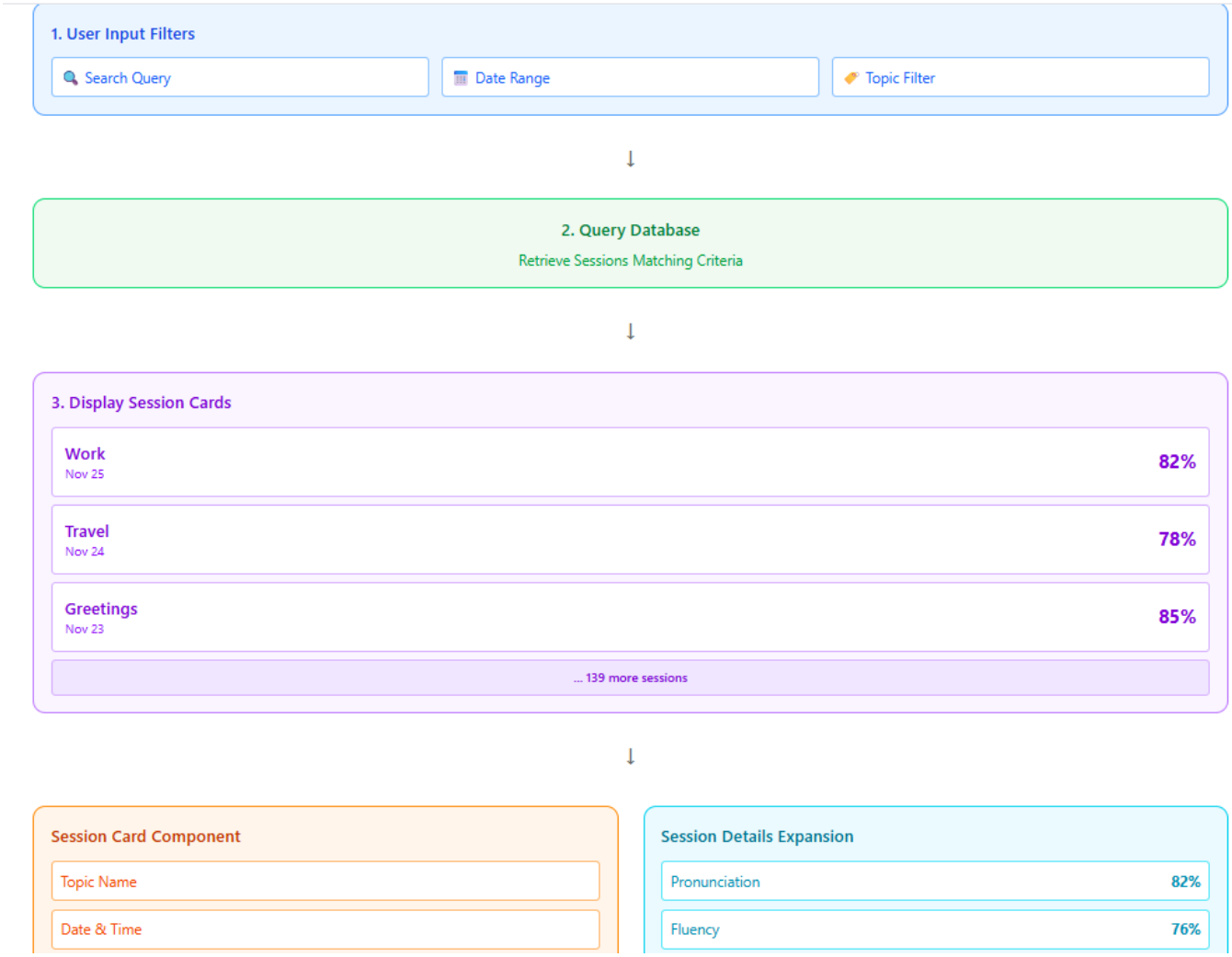
5.1.4. Progress Analytics Dashboard

Overview statistics, progress line charts, skills radar, and topic performance comparisons



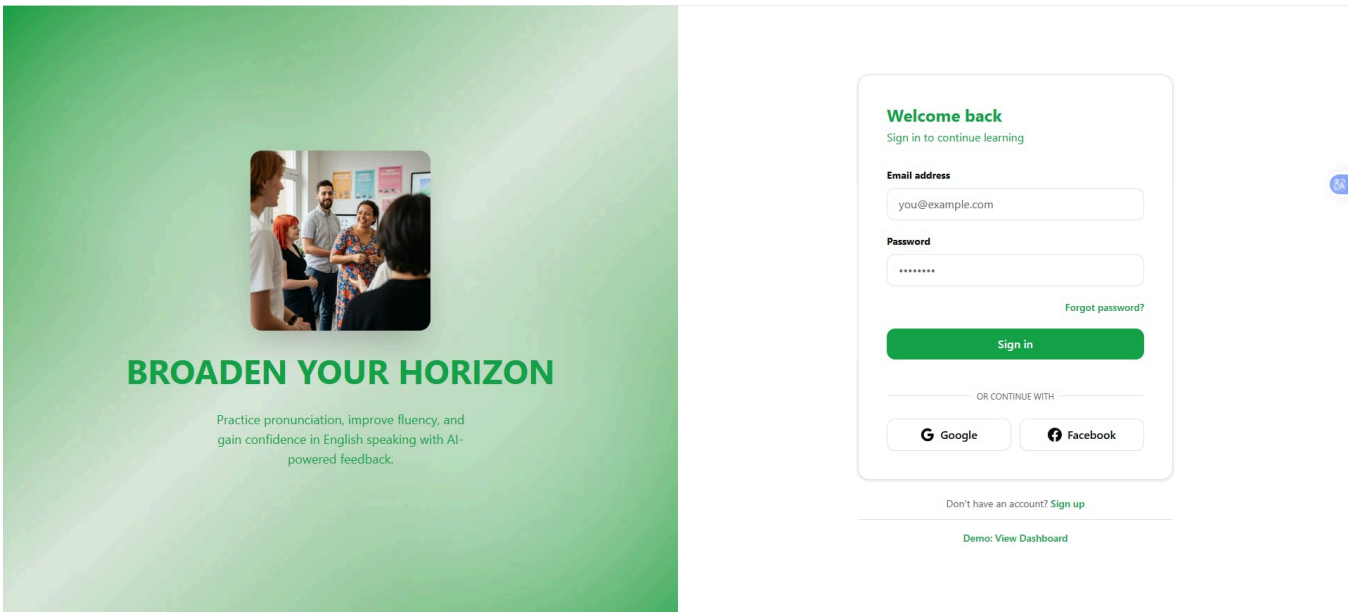
5.1.5. Session History Page

Search filters, session cards with timestamps, topics, scores, and expandable detailed metrics

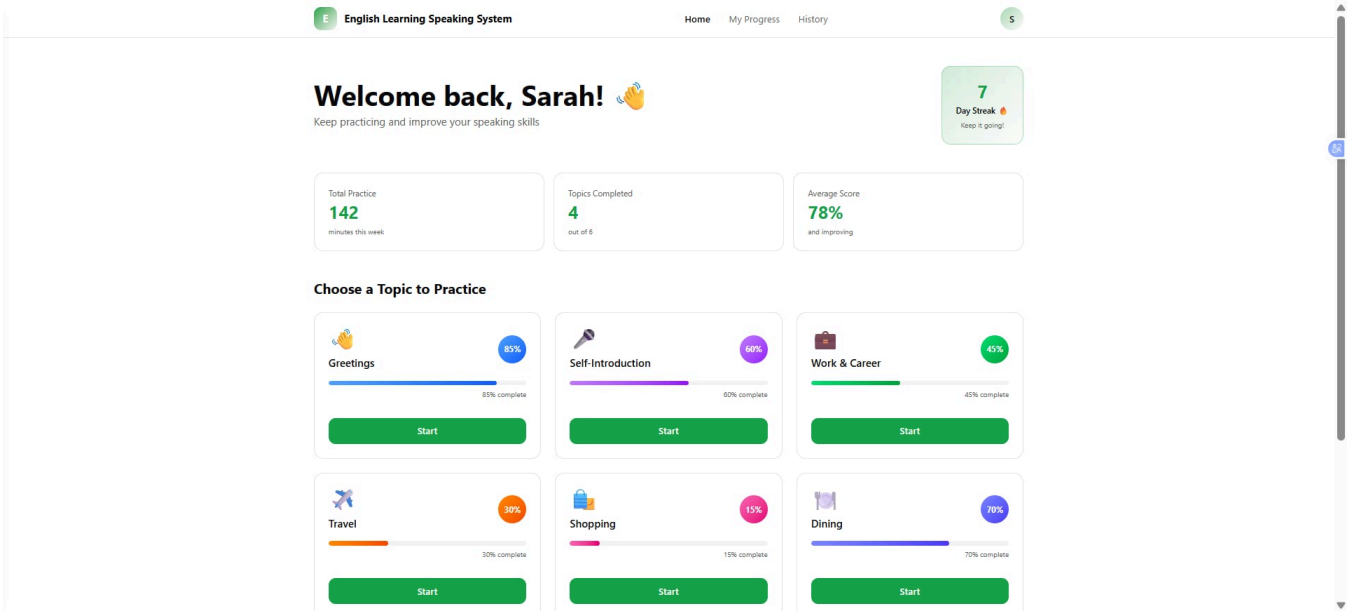


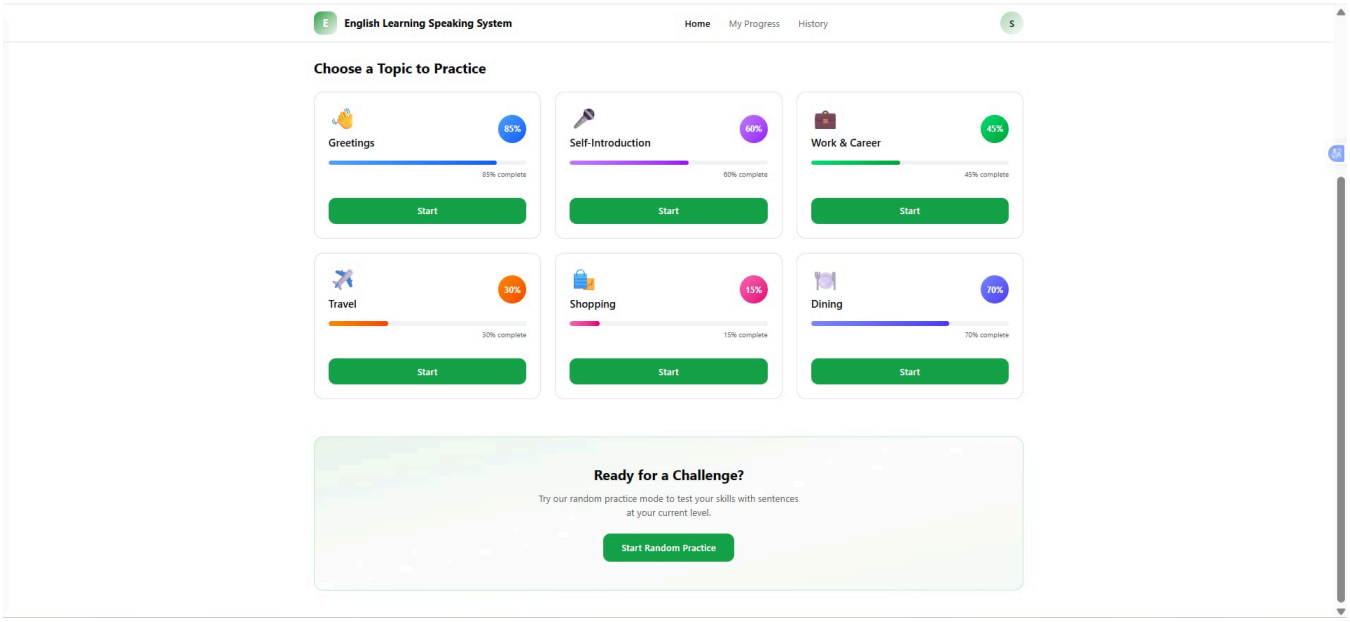
5.2. Prototypes

5.2.1. Authentication/Login Interface prototype

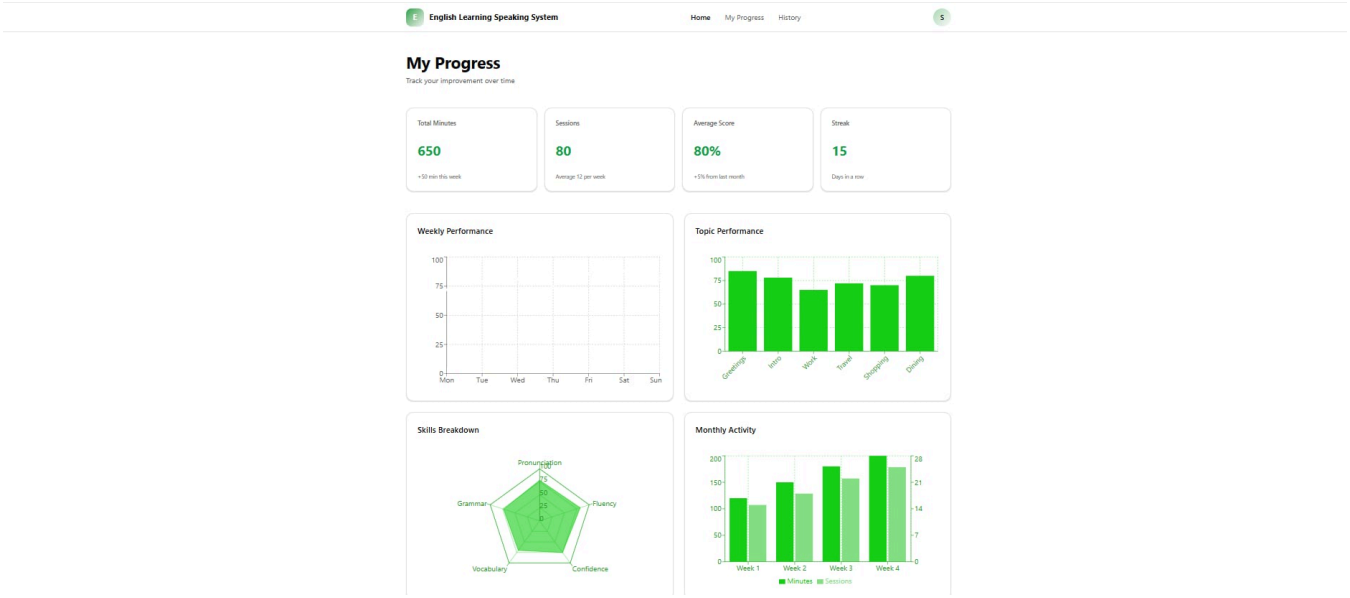


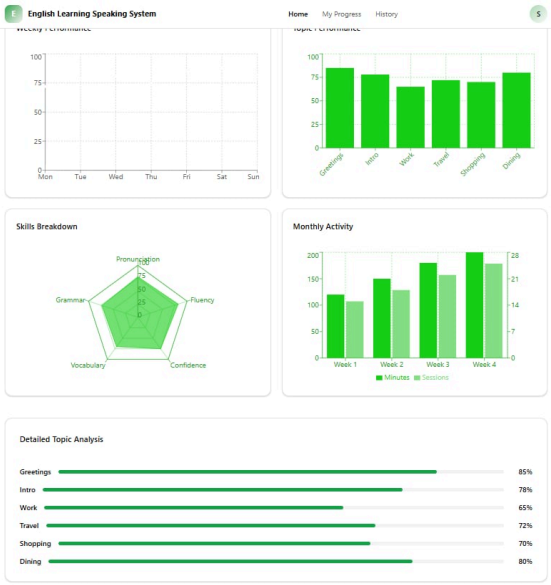
5.2.2. Dashboard Interface prototype



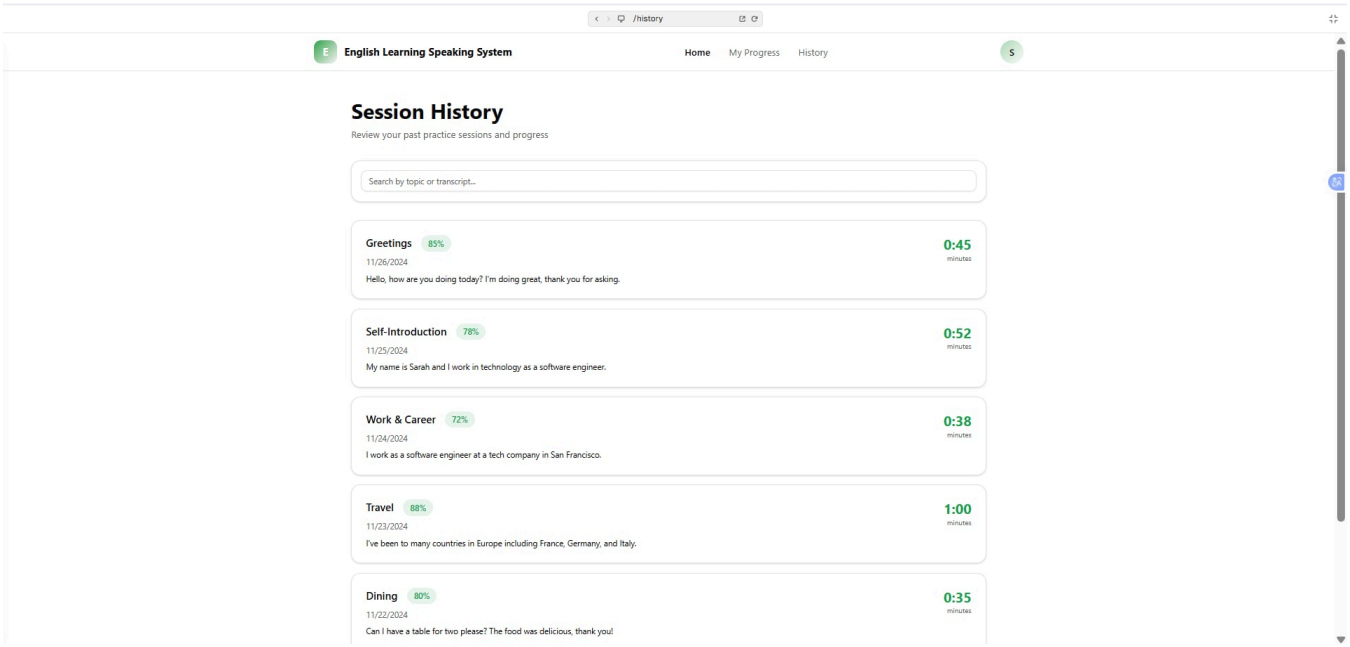


5.2.3. Progress Tracking prototype

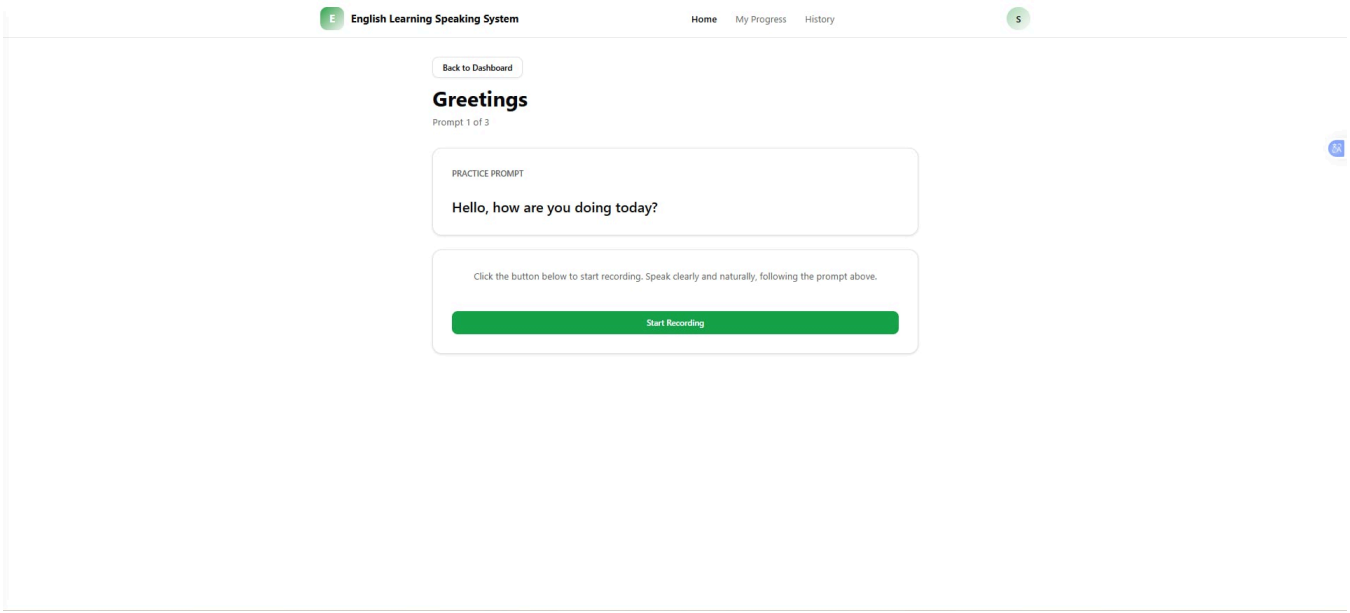


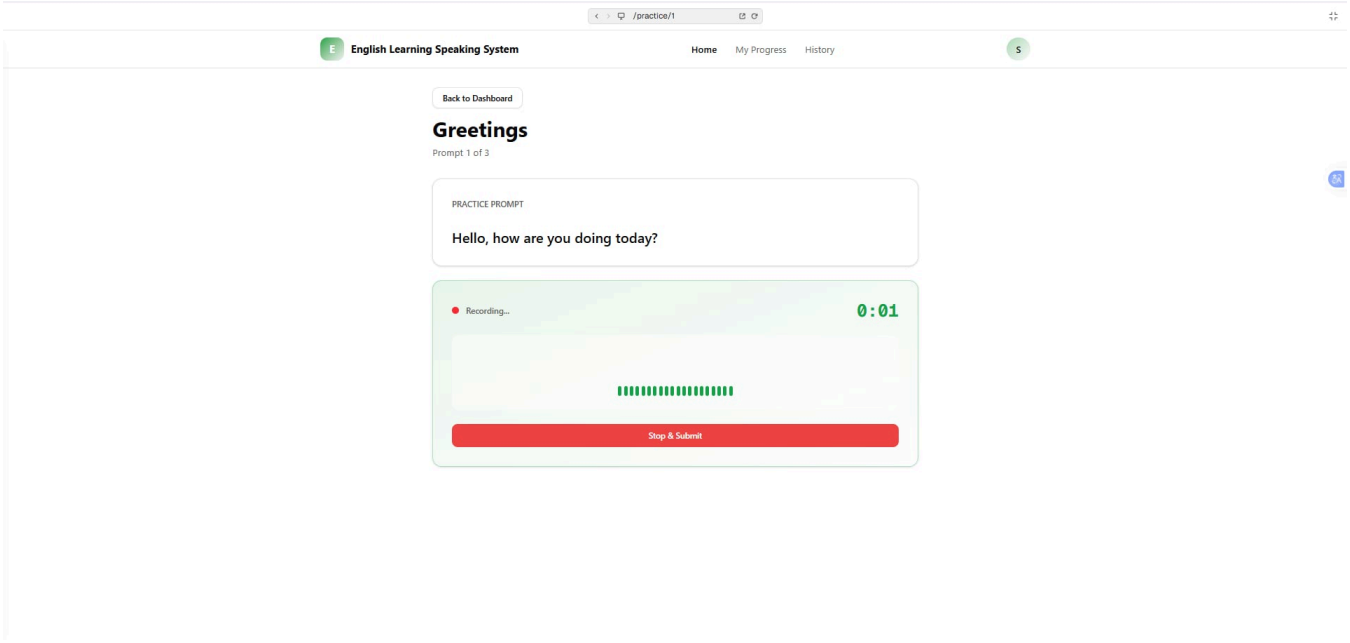


5.2.4. History Session prototype

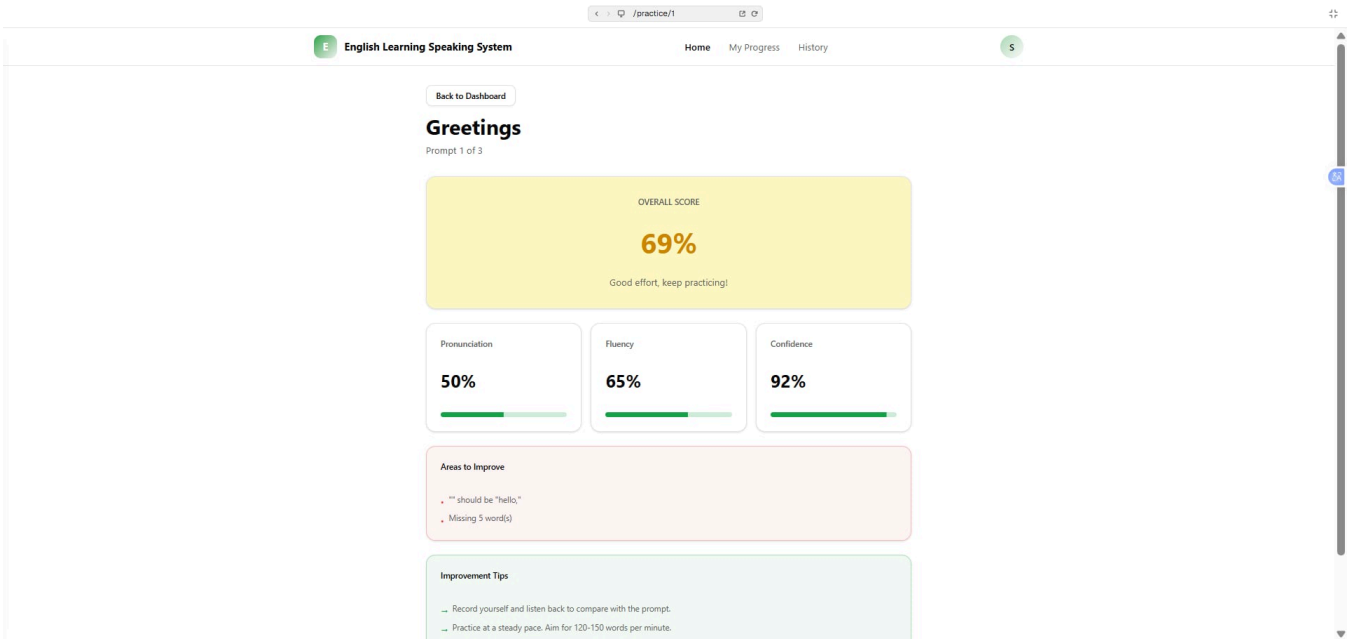


5.2.5. Practice Session prototype





5.2.6. AI Feedback prototype


/practice/1
English Learning Speaking System
Home
My Progress
History
S
Back to Dashboard
Greetings
Prompt 1 of 3
OVERALL SCORE
69%
Good effort, keep practicing!
Pronunciation
50%
Fluency
65%
Confidence
92%
Areas to Improve
" should be "hello,"
Missing 5 word(s)
Improvement Tips
Record yourself and listen back to compare with the prompt.
Practice at a steady pace. Aim for 120-150 words per minute.