

Introduction to Software Engineering

Software Design



Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

1 Member Contribution Assessment	2
2 Conceptual Model	3
3 Architectural Design	4
3.1 Architecture Diagram	4
3.1.1 System Decomposition Tree Diagram	4
3.1.2 Overall System Architecture Diagram	5
3.1.3 Architectural Design Aspects	6
3.2 Class Diagram	8
3.3 Class Specifications	9
3.3.1 Class PracticeService	9
3.3.3 Class Topic	12
3.3.4 Class Exercise	13
3.3.5 Class ExerciseAttempt	14
3.3.6 Class SpeechRecognitionService	15
3.3.7 Class UserController	16
3.3.8 Class UserStats	17
3.3.9 Class UserSkills	18
3.3.10 Class PracticeController	18
4 Data Design	20
4.1 Data Diagram	20
4.2 Data Specification	22
1. Table users	22
2. Table topics	23
3. Table lessons	24
4. Table exercises	25
5. Table practicesessions	26
6. Table exerciseattempts	27
7. Table userstats	28
8. Table userskills	29
5 User Interface and User Experience Design	31
5.1 Screen Diagram	31
5.2 Screen Specifications	33
5.2.1 Landing Page	33
5.2.2 Login Screen	35
5.2.3 Dashboard Screen:	36

5.2.4 Practice Screen	38
5.2.5 Progress Report Screen	40
5.2.6 Exercise Screen:	42
5.2.7 Register Screen:	42
5.2.8 Forgot Password Screen:	43

Software Design

Objectives

This document focus on the following topics:

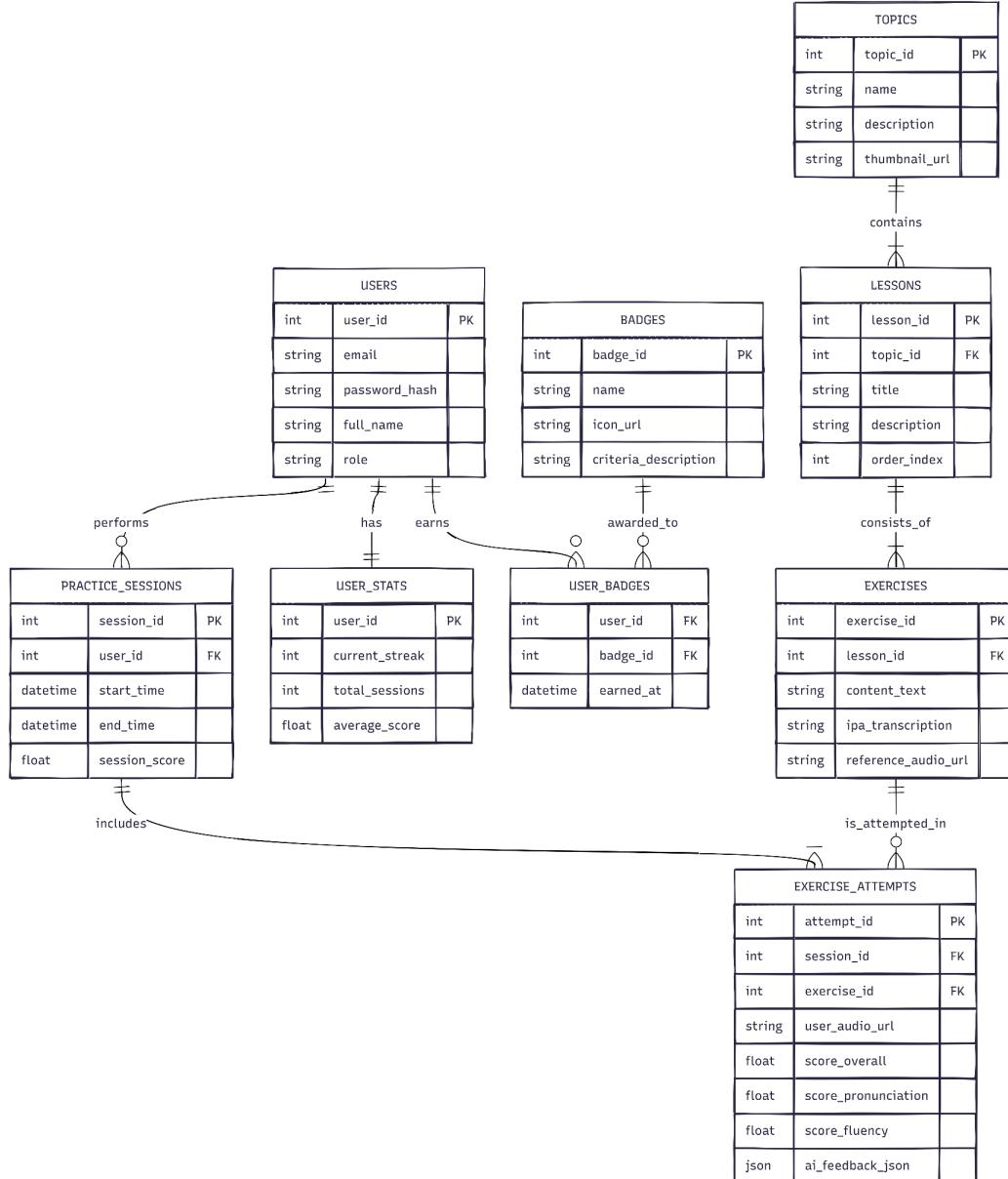
- ✓ Complete the Software Design Document with the following contents:
 - Conceptual Model
 - Architectural Design
 - Data Design
 - User Interface Design
- ✓ Understanding the Software Design Document.

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 Conceptual Model

The Conceptual Model is designed using the **Extended Entity-Relationship (EER)** diagram to represent the semantic entities within the English Speaking Learning System.



* Entity Description

The model comprises the following key entities:

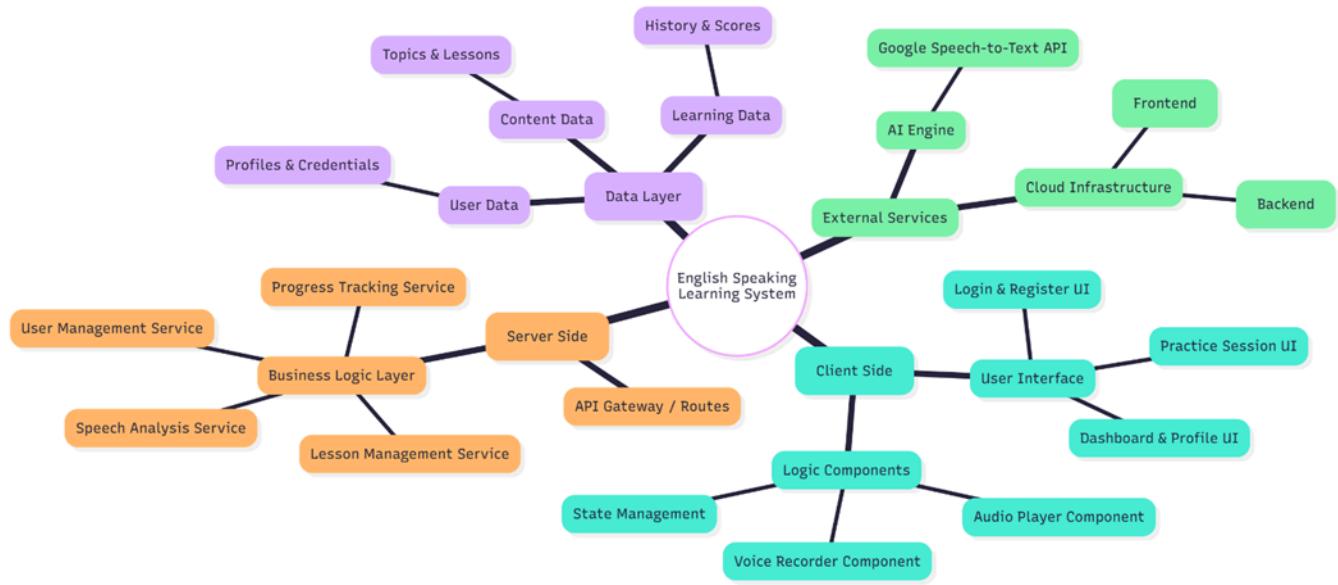
- **Users & Stats:** The central entity USERS stores account credentials and profile information. It is linked one-to-one with USER_STATS to optimize the retrieval of performance metrics like learning streaks and average scores.
- **Content Management (Topics, Lessons, Exercises):** The learning content is structured hierarchically. TOPICS (e.g., Travel, Work) contain multiple LESSONS, and each lesson consists of various EXERCISES. The EXERCISES entity stores the ipa_transcription and reference_audio_url to support pronunciation practice.
- **Practice & Evaluation:** When a learner starts a lesson, a PRACTICE_SESSION is created. Each specific recording is stored as an EXERCISE_ATTEMPT. Crucially, this entity stores the ai_feedback_json, which contains the detailed phoneme-level analysis and error data returned by the speech recognition engine.
- **Gamification:** To support the requirement of motivating users, the BADGES and USER_BADGES entities track the achievements unlocked by learners based on their performance.

3 Architectural Design

3.1 Architecture Diagram

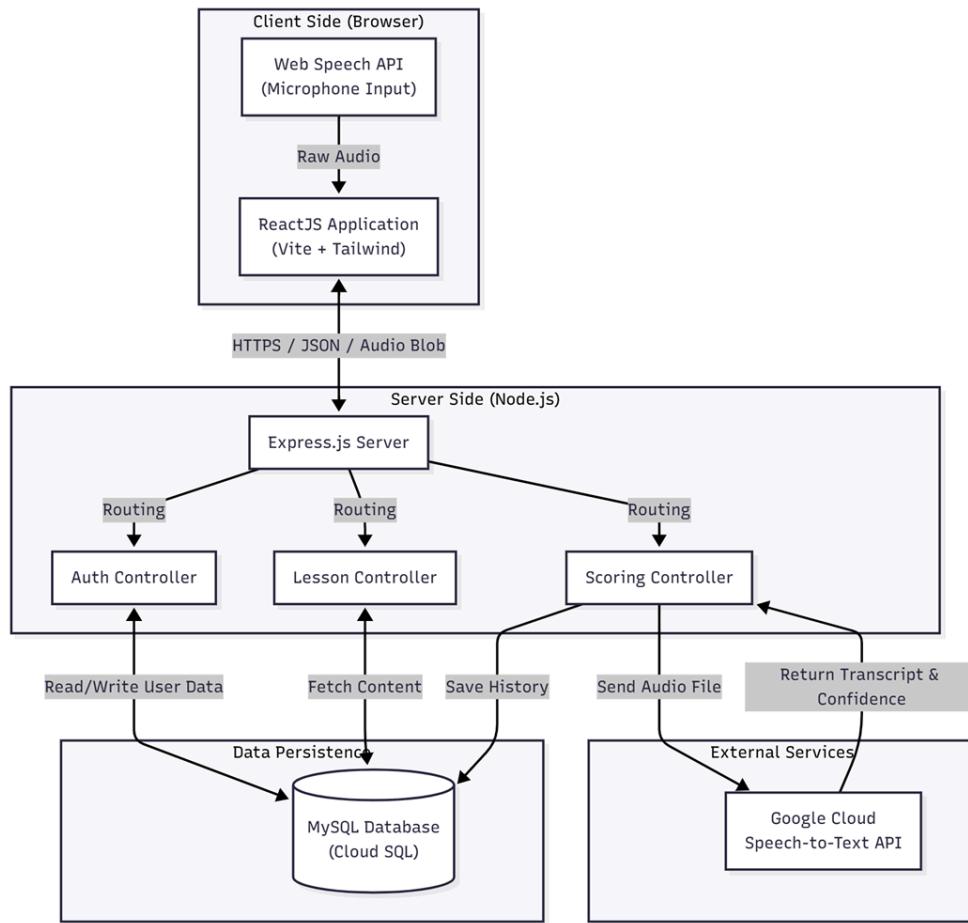
3.1.1 System Decomposition Tree Diagram

The system is decomposed into four main subsystems: Frontend, Backend, Database, and External Services. The decomposition tree below illustrates the hierarchical components of the English Speaking Learning System based on the functional modules defined in the proposal.



3.1.2 Overall System Architecture Diagram

The following diagram illustrates the high-level architecture of the system, demonstrating the relationships and data flow between the main components. The system follows a standard web application flow where the ReactJS frontend communicates with the Node.js backend via RESTful APIs, and the backend delegates voice processing to the Google Speech-to-Text API.



3.1.3 Architectural Design Aspects

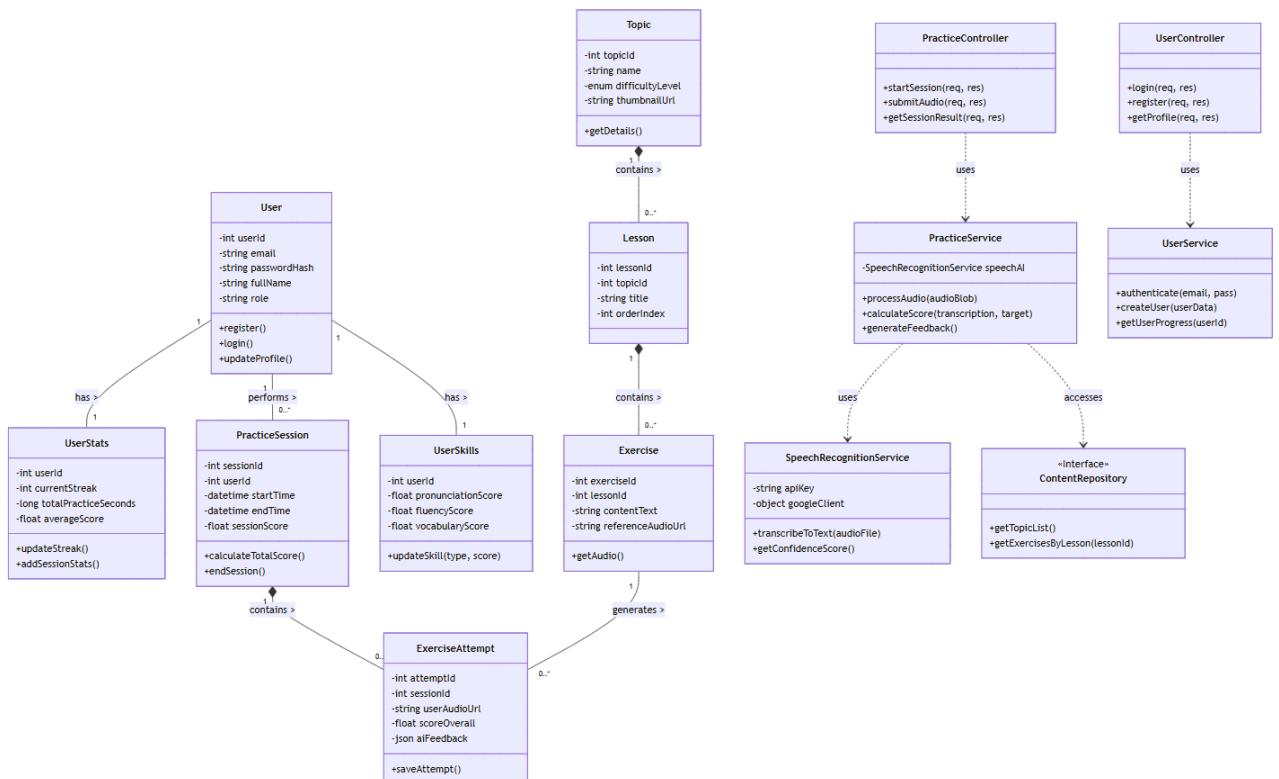
The English Speaking Learning System employs a robust and scalable architecture designed to handle real-time audio processing and user management. The key architectural styles and patterns applied are:

- **Client-Server Architecture**
 - The system adopts a clear separation between the Client and the Server.
 - **The Client (Frontend):** Built with **ReactJS**, responsible for the presentation layer, capturing user voice input via the browser's Web Speech API, and rendering feedback .

- **The Server (Backend):** Built with **Node.js (Express)**, responsible for business logic, authentication, and orchestrating calls to the database and external AI services .
- **Layered Architecture (Backend): The backend is structured into distinct layers to ensure maintainability and separation of concerns:**
 - Controller Layer: Handles incoming HTTP requests and responses
 - Service Layer: Contains the core business logic (e.g., the algorithm to compare user transcript vs. target sentence).
 - Data Access Layer: Manages direct interactions with the MySQL database.
- **Integration of Third-Party AI Services (Service-Oriented Approach)**
 - Instead of building a speech recognition engine from scratch, the system integrates the Google Speech-to-Text API. This acts as a specialized external service component. The backend acts as a secure proxy, receiving audio from the client, authenticating with Google Cloud, and processing the AI response before sending the final score back to the user.
- **Secure Data Handling**
 - Encryption: All communications between the client and server are secured over HTTPS/TLS.
 - Stateless Authentication: The system uses a stateless authentication mechanism (likely JWT - JSON Web Tokens) for user sessions, which improves scalability for the Node.js server.

3.2 Class Diagram

The class diagram illustrates the object-oriented structure of the English Speaking Learning System, emphasizing the separation between data entities and business logic. The domain model centers on the User class, which maintains one-to-one relationships with UserStats and UserSkills for tracking progress, and a one-to-many relationship with PracticeSessions. The learning content follows a hierarchical composition pattern where a Topic contains multiple Lessons, which in turn contain Exercises. On the architectural side, the system employs a layered approach: Controllers (e.g., PracticeController, UserController) manage incoming requests and depend on Services (e.g., PracticeService, UserService) to execute core business logic. Notably, the PracticeService integrates with the SpeechRecognitionService to handle audio processing and interacts with the ContentRepository for data access.



3.3 Class Specifications

3.3.1 Class PracticeService

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	speechAI	private	SpeechRecognitionService	Dependency on the AI wrapper class to handle audio transcription and scoring.
2	sessionRepository	private	Repository	Component used for interacting with the PracticeSession and ExerciseAttempt data storage.
3	contentRepository	private	ContentRepository	Component used to fetch the target text and reference audio for the exercise.

Seq	Operation	Modifier	Constraint	Description
1	processAudio(audioBlob)	public	Return ExerciseAttempt object	Takes raw user audio, orchestrates transcription and scoring via speechAI , saves the

				attempt record, and returns the result.
2	calculateScore(transcription, target)	private	Return float (0-100)	Compares the user's transcribed text against the exercise's target text to calculate an overall accuracy score.
3	generateFeedback()	private	Return JSON	Generates structured feedback (e.g., phoneme errors, suggested improvements) based on the detailed AI analysis.
4	startSession(userId, topicId)	public	Return sessionId	Creates a new record in the practicesessions table and returns the unique ID to the frontend.

3.3.2 Class User

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userId	private	int, Primary Key, Auto Increment	Unique identifier for the user.
2	email	private	string, Unique, Not Null	User's login email address.

3	passwordHash	private	string, Not Null	Encrypted (hashed) password.
4	fullName	private	string	User's full display name.
5	role	private	Enum('USER', 'ADMIN')	Determines access privileges within the system.

Seq	Operation	Modifier	Constraint	Description
1	register()	public	return boolean	Creates a new user record in the database.
2	verifyPassword(inputPass)	public	return boolean	Compares the input password with the stored hash.
3	updateProfile(data)	public		Updates user details like name or avatar URL.

3.3.3 Class Topic

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	topicId	private	int, PK, Auto Increment	Unique identifier for the topic.
2	name	private	string, Not Null	Name of the topic (e.g., "Travel").
3	difficultyLevel	private	Enum('Easy', 'Medium', 'Hard')	Difficulty level of the topic.
4	thumbnailUrl	private	string, Nullable	URL to the topic's cover image.

Seq	Operation	Modifier	Constraint	Description
1	getDetails()	public	return JSON	Returns topic details along with associated lessons.
2	getLessonOrder()	public	return List<Lesson>	Retrieves lessons sorted by their orderIndex .

3.3.4 Class Exercise

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	exerciseId	private	int, PK, Auto Increment	Unique identifier for the exercise.
2	lessonId	private	int, Foreign Key	Links to the parent lesson.
3	contentText	private	string, Not Null	The target sentence/text for the user to read/speak.
4	referenceAudioUrl	private	string, Nullable	URL for the native speaker reference audio.
5	type	private	Enum('Speaking', 'Listening')	Type of exercise.

Seq	Operation	Modifier	Constraint	Description
1	getAudio()	public	return AudioFile	Fetches the reference audio file for playback.
2	getTranscript()	public	return string	Returns the target text (contentText) for comparison.

3.3.5 Class ExerciseAttempt

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	attemptId	private	int, PK, Auto Increment	Unique identifier for the attempt.
2	sessionId	private	int, Foreign Key, Not Null	Links to the high-level practice session.
3	userAudioUrl	private	string, Nullable	URL to the user's recorded audio file (cloud storage).
4	scoreOverall	private	float (0-100), Not Null	Overall score generated by AI for this attempt.
5	aiFeedback	private	JSON, Nullable	Detailed AI feedback (phoneme errors, word-level confidence).

Seq	Operation	Modifier	Constraint	Description
1	saveAttempt()	public	return attemptId	Persists the attempt data (scores, audio URL, feedback) to the database.

2	getDetailedFeedback()	public	return JSON	Retrieves and formats the AI feedback for display.
---	-----------------------	--------	-------------	--

3.3.6 Class *SpeechRecognitionService*

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	apiKey	private	string	Credentials for Google Cloud API access.
2	googleClient	private	object	Instance of the Google Speech SDK client.

Seq	Operation	Modifier	Constraint	Description
1	transcribeToText(audioFile)	public	return string	Sends the audio file buffer to the external service and returns the transcribed text.
2	getConfidenceScore()	public	return float	Retrieves the AI's confidence level associated with the transcription.
3	getPhonemeData()	public	return JSON	Fetches detailed speech metrics used

				for generating specific feedback.
--	--	--	--	-----------------------------------

3.3.7 Class UserController

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userService	private	UserService	Dependency on the user business logic handler.

Seq	Operation	Modifier	Constraint	Description
1	login(req, res)	public	POST /api/login	Endpoint to handle user login request. Calls UserService.authenticate() .
2	register(req, res)	public	POST /api/register	Endpoint to handle new user registration.
3	getProfile(req, res)	public	GET /api/profile	Endpoint to retrieve the currently logged-in user's profile information.

3.3.8 Class UserStats

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userId	private	int, PK, Foreign Key	Links 1-to-1 with the User.
2	currentStreak	private	int, Default 0	Current count of consecutive practice days.
3	totalPracticeSeconds	private	long, Default 0	Total time spent practicing.
4	averageScore	private	float (0-100)	Average score across all exercises.

Seq	Operation	Modifier	Constraint	Description
1	updateStreak()	public		Checks activity and increments or resets the practice streak.
2	addSessionStats(duration, score)	private		Updates total time and recalculates the running average score.

3.3.9 Class UserSkills

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userId	private	int, PK, Foreign Key	Links to the User.
2	skillName	private	string, PK	The specific skill being tracked (e.g., 'Pronunciation').
3	skillScore	private	float (0-100)	Current score/rating for this skill.

Seq	Operation	Modifier	Constraint	Description
1	updateSkill(type, newScore)	public		Updates the rating for a specific skill based on new attempt results.
2	getRadarData()	public	return JSON	Retrieves all skill scores formatted for the Radar Chart on the Progress Report Screen.

3.3.10 Class PracticeController

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	practiceService	private	PracticeService	Dependency on the core practice business logic handler.

Seq	Operation	Modifier	Constraint	Description
1	startSession(req, res)	public	POST /api/session/start	Initializes a new practice session for the user/topic.
2	submitAudio(req, res)	public	POST /api/session/submit	Accepts the audio file blob from the client and triggers PracticeService.processAudio() .
3	getSessionResult(req, res)	public	GET /api/session/{id}/result	Retrieves the final results and statistics for a completed session.

4 Data Design

4.1 Data Diagram

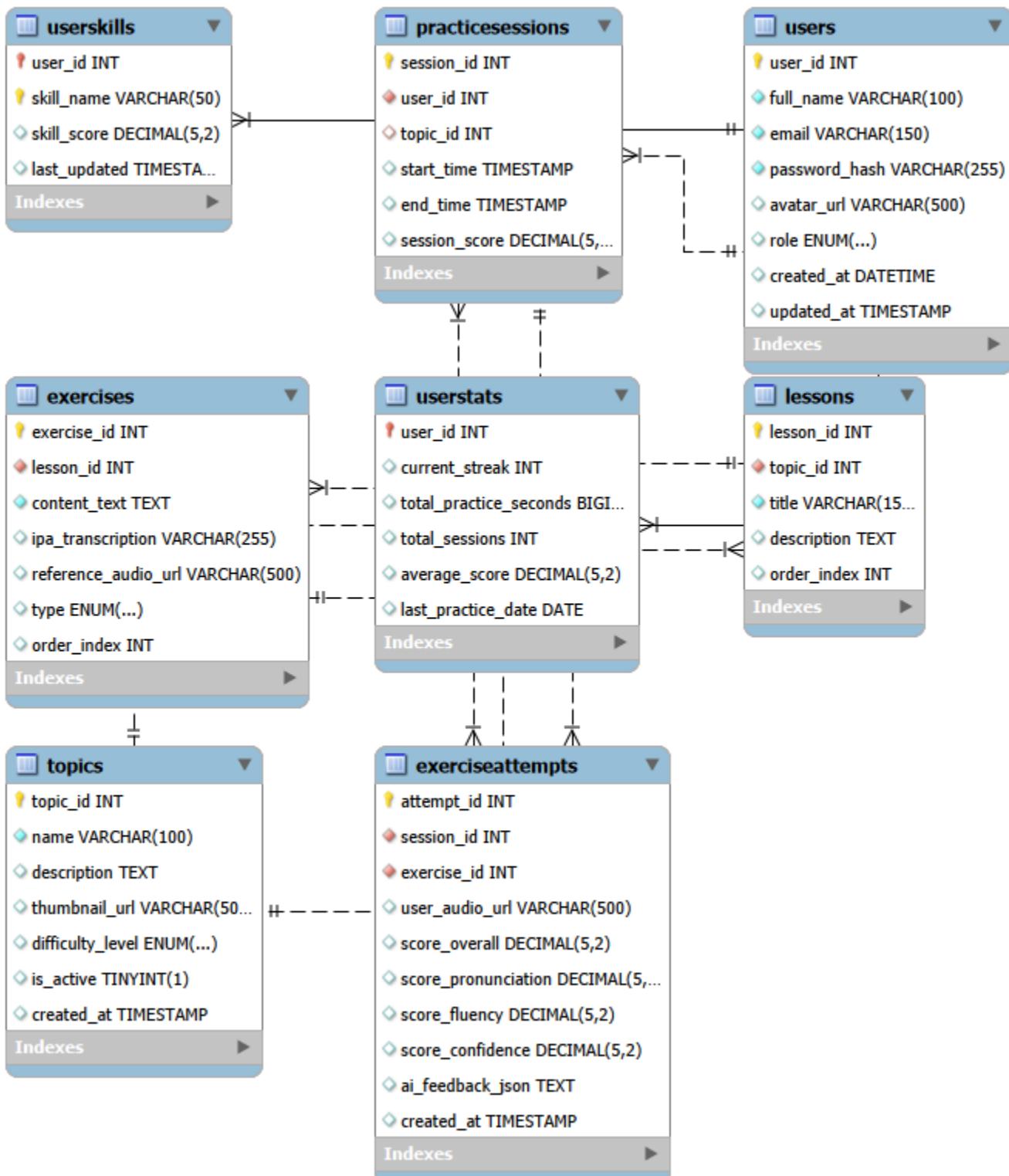


Diagram Description:

The data system is designed following the Relational Database model, focusing on managing users, the learning curriculum (Topics/Lessons/Exercises), and practice results.

- **Core:** The `users` table manages user identity and authentication.
- **Learning Content:** Organized hierarchically: `topics` contain multiple `lessons`, and `lessons` contain multiple `exercises`.
- **Practice Activity:** When users practice, high-level data is recorded in `practicesessions`, while specific details for each attempt are stored in `exerciseattempts`.
- **Statistics:** The `userstats` and `userskills` tables aggregate performance data to generate reports and track user progress

4.2 Data Specification

1. Table `users`

Stores user account information and personal profiles.

Attribute Name	Data Type	Constraints	Key Type	Description
<code>user_id</code>	INT	Auto Increment, Not Null	PK	Unique identifier for the user.
<code>full_name</code>	VARCHAR(100)	Not Null		User's full display name.
<code>email</code>	VARCHAR(150)	Unique, Not Null		User's login email address.

password_hash	VARCHAR(255)	Not Null		Encrypted (hashed) password.
avatar_url	VARCHAR(500)	Nullable		URL to the user's avatar image.
role	ENUM(...)	Values: 'USER', 'ADMIN'		User role within the system.
created_at	DATETIME	Default NOW()		Account creation timestamp
updated_at	TIMESTAMP	On Update CURRENT_TIMESTAMP		Last profile update timestamp

2. Table topics

Stores broad categories or themes in the curriculum.

Attribute Name	Data Type	Constraints	Key Type	Description
topic_id	INT	Auto Increment, Not Null	PK	Unique identifier for the topic.

<code>name</code>	VARCHAR(100)	Not Null		Name of the topic (e.g., Travel, Business).
<code>description</code>	TEXT	Nullable		Detailed description of the topic.
<code>thumbnail_url</code>	VARCHAR(500)	Nullable		URL for the topic's cover image.
<code>difficulty_level</code>	ENUM(...)	Values: 'Easy', 'Medium', 'Hard'		Difficulty level of the topic.
<code>is_active</code>	TINYINT(1)	Default 1 (True)		Topic activation status.
<code>created_at</code>	TIMESTAMP	Default NOW()		Record creation timestamp.

3. Table lessons

Stores individual lessons belonging to a specific topic.

Attribute Name	Data Type	Constraints	Key Type	Description

lesson_id	INT	Auto Increment, Not Null	PK	Unique identifier for the lesson.
topic_id	INT	Not Null	FK	Foreign key linking to the topics table.
title	VARCHAR(1 50)	Not Null		Title of the lesson.
description	TEXT	Nullable		Introduction or summary of the lesson.
order_index	INT	Not Null		Display order of the lesson within the topic.

4. Table exercises

Stores specific exercises within a lesson.

Attribute Name	Data Type	Constraint s	Key Typ e	Description
exercise_id	INT	Auto Increment, Not Null	PK	Unique identifier for the exercise.
lesson_id	INT	Not Null	FK	Foreign key linking to the lessons table.

content_text	TEXT	Not Null		The text content for reading/practice.
ipa_transcription	VARCHAR(255)	Nullable		Standard IPA transcription of the text.
reference_audio_url	VARCHAR(500)	Nullable		URL for native speaker reference audio.
type	ENUM(...)	Values: 'Speaking', 'Listening'		Type of exercise.
order_index	INT	Not Null		Order of the exercise within the lesson.

5. Table practicesessions

Stores high-level data about a user's practice session.

Attribute Name	Data Type	Constraints	Key Type	Description
session_id	INT	Auto Increment, Not Null	PK	Unique identifier for the session.

<code>user_id</code>	INT	Not Null	FK	The user who performed the session.
<code>topic_id</code>	INT	Not Null	FK	The topic practiced during the session.
<code>start_time</code>	TIMESTAMP	Not Null		Session start time.
<code>end_time</code>	TIMESTAMP	Nullable		Session end time.
<code>session_score</code>	DECIMAL(5, 2)	Min 0, Max 100		Aggregate score for the entire session.

6. Table `exerciseattempts`

Stores detailed results of every attempt (recording/answer) for an exercise.

Attribute Name	Data Type	Constraints	Key Type	Description
<code>attempt_id</code>	INT	Auto Increment, Not Null	PK	Unique identifier for the attempt.
<code>session_id</code>	INT	Not Null	FK	Links to the <code>practicesessions</code> table.

exercise_id	INT	Not Null	FK	Links to the exercises table.
user_audio_url	VARCHAR(500)	Nullable		URL to the user's recorded audio file.
score_overall	DECIMAL(5,2)	Min 0, Max 100		Overall score generated by AI.
score_pronunciation	DECIMAL(5,2)	Min 0, Max 100		Component score for pronunciation.
score_fluency	DECIMAL(5,2)	Min 0, Max 100		Component score for fluency.
score_confidence	DECIMAL(5,2)	Min 0, Max 100		Component score for confidence.
ai_feedback_json	TEXT	JSON format		Detailed AI feedback (phoneme errors, etc.).
created_at	TIMESTAMP	Default NOW()		Timestamp when the attempt was submitted.

7. Table userstats

Stores aggregated statistics for a user (1-to-1 relationship with users).

Attribute Name	Data Type	Constraints	Key Type	Description

<code>user_id</code>	INT	Not Null	PK, FK	Links 1-to-1 with the <code>users</code> table.
<code>current_streak</code>	INT	Default 0		Current count of consecutive practice days.
<code>total_practice_second</code> s	BIGINT	Default 0		Total time spent practicing (in seconds).
<code>total_sessions</code>	INT	Default 0		Total number of completed sessions.
<code>average_score</code>	DECIMAL(5, 2)	Min 0, Max 100		Average score across all exercises.
<code>last_practice_date</code>	DATE	Nullable		The date of the most recent practice.

8. Table userskills

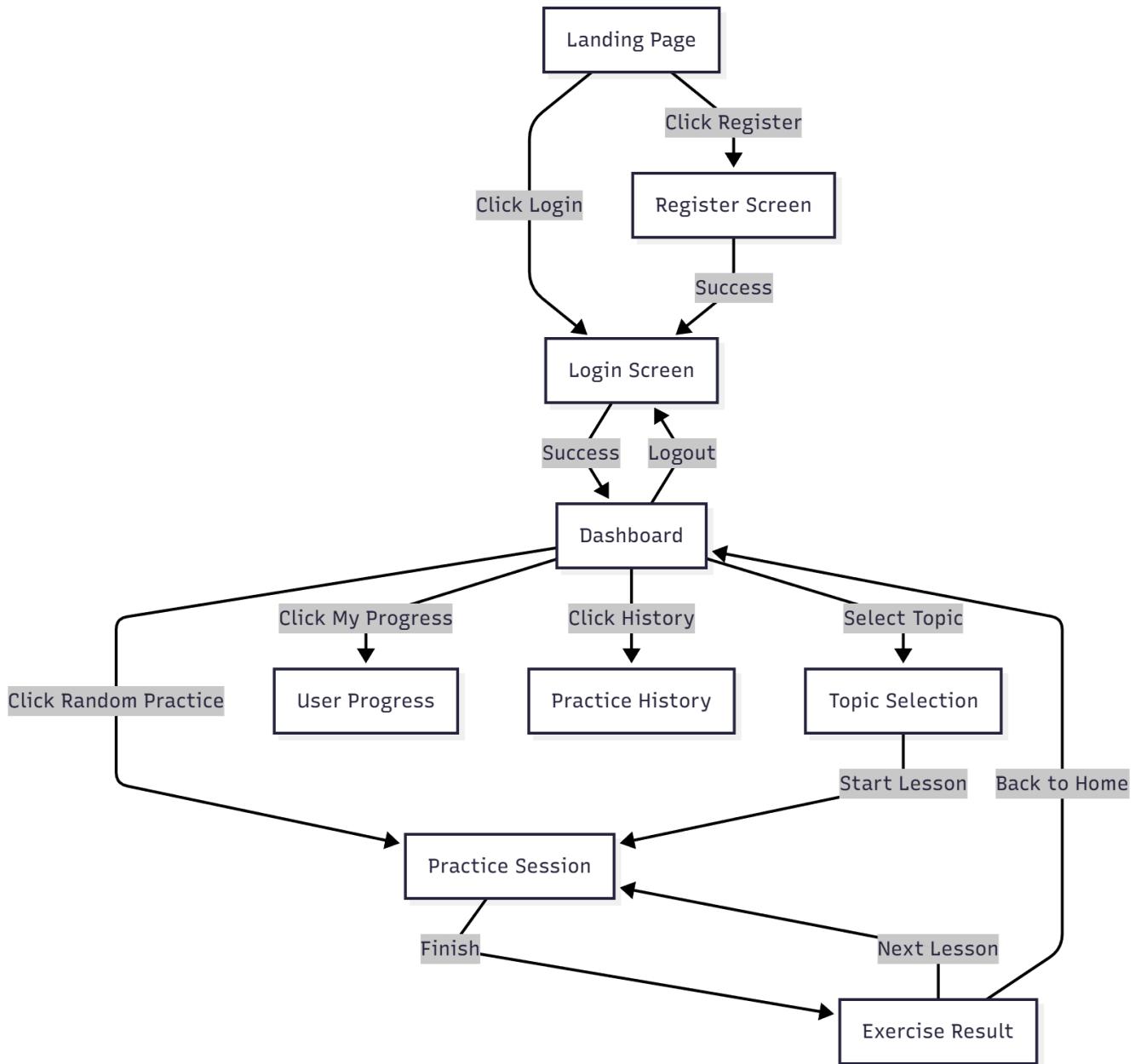
Stores specific skill ratings for a user (e.g., Intonation, Pronunciation).

Attribute Name	Data Type	Constraints	Key Type	Description

user_id	INT	Not Null	PK, FK	Links to the users table.
skill_name	VARCHAR(50)	Not Null	PK	Name of the skill (Part of Composite PK).
skill_score	DECIMAL(5, 2)	Min 0, Max 100		Current score/rating for this skill.
last_updated	TIMESTAMP	Default NOW()		Timestamp of the last score update.

5 User Interface and User Experience Design

5.1 Screen Diagram



Seq	Screen	Description
1	Landing Page	Introduces the system, highlighting key features and guiding users to Login/Register.
2	Login Screen	Allows users to authenticate and access the system.
3	Register Screen	Allows new users to create an account
4	Forgot Password Screen	Allows users to recovery account if they forgot password
5	Dashboard	The main hub after login, displaying progress overview, statistics, and topic suggestions.
6	Topic Selection	List of available topics allowing users to choose a specific area to practice.
7	Practice Session	The core screen where users perform speaking exercises and record their voice.
8	Exercise Result	Displays detailed results after completing an exercise (score, feedback).
9	User Progress	Detailed view of user skills, learning history, and improvement charts.

5.2 Screen Specifications

5.2.1 Landing Page

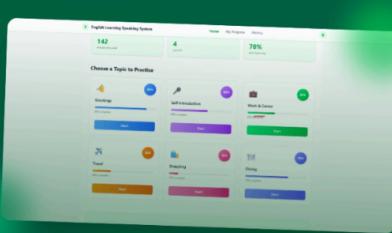
The first screen users see when accessing the application, designed to attract and convert visitors into registered users.

Presentation Format:

- Header (Sticky): "English Learning" Logo (left), Navigation Menu (Features, Testimonials), Login/Register buttons (right).
- Hero Section: Large Headline emphasizing core value, High-quality illustration or vector art about learning, prominent "Get Started" button.
- Feature Grid: List of key features (Voice Recognition, AI Feedback, Progress Tracking) with illustrative icons.
- Footer: Contact information, social media links, and copyright.

Event Handling:

- Click "Get Started" / "Register": Redirects to the Register Screen.
- Click "Login": Redirects to the Login Screen.
- Scroll: Header remains fixed when scrolling to facilitate navigation.



Speak English Like a Native

Your personal AI English coach. Get instant feedback on your pronunciation, intonation, and fluency. Practice anytime, anywhere.

[Get Started Free](#)

Already have an account? Let's start learning!

★ 4.8/5 Rating • 10M+ Downloads

WHY CHOOSE US

The World's Best English Pronunciation Coach

Powered by advanced Artificial Intelligence to help you speak confidently.

- AI Voice Recognition**
Our proprietary AI technology analyzes your speech with 95% accuracy, detecting even the smallest pronunciation errors.
- Instant Feedback**
Get real-time feedback on your pronunciation, intonation, and stress. Correct your mistakes instantly.
- Personalized Learning**
Tailored lesson plans based on your proficiency level and learning goals. Improve faster with a customized path.
- 200+ Topics**
Practice English for travel, business, daily life, and more. Learn vocabulary that is relevant to you.

Loved by English Learners Worldwide

Join thousands of users who have transformed their English speaking skills.



Sarah Jenkins
Marketing Director

"I've tried many apps, but this one is the best. My pronunciation has improved significantly in just 3 weeks!"

★★★★★



Michael Chen
Software Engineer

"The AI feedback is incredible. It's like having a personal tutor available 24/7. Highly recommended!"

★★★★★



Elena Rodriguez
Student

"I used to be afraid of speaking English. Now I feel confident and can hold conversations easily."

★★★★★

Ready to Master English Speaking?

Start your journey today with our AI-powered coach. Join for free and see the difference.

[Get Started Now](#)

No credit card required for free trial.

ELS System
The #1 AI-powered English speaking coach. Improve your pronunciation and fluency with instant feedback.

Product
Features
Pricing
For Business
For Schools

Resources
Blog
Community
Help Center
Contact Us

Legal
Privacy Policy
Terms of Service
Cookie Policy

© 2025 ELS System. All rights reserved.

Facebook Twitter

5.2.2 Login Screen

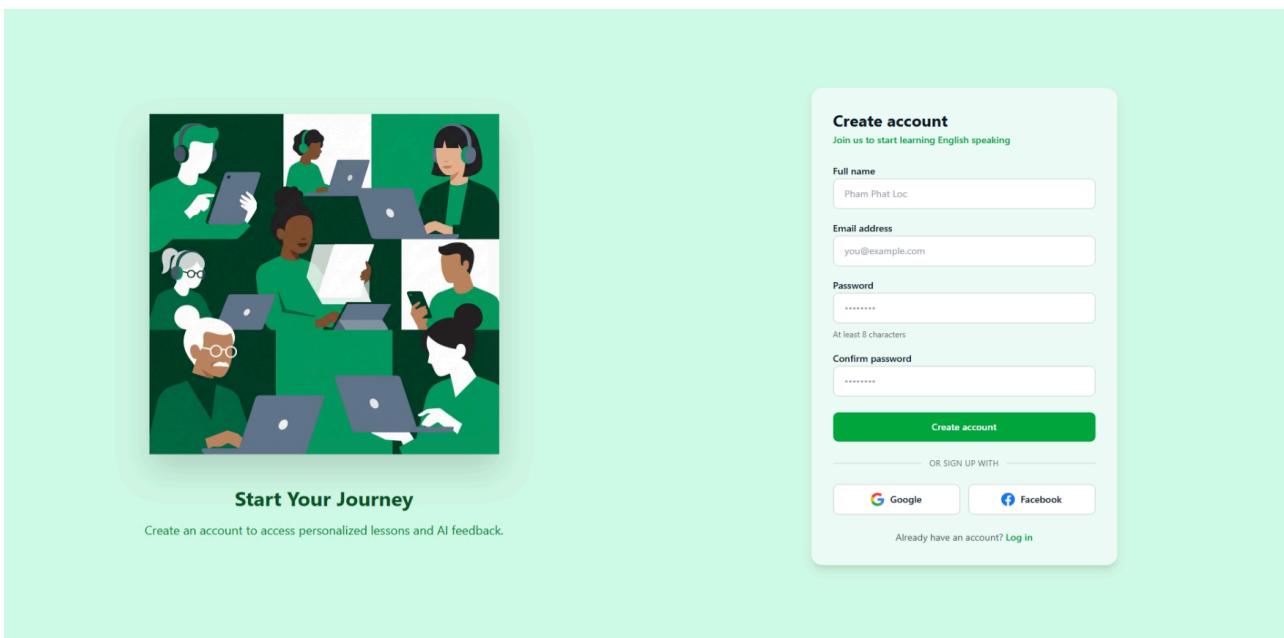
Allows existing users to access the system.

Presentation Format:

- Layout: Split into 2 columns
 - Left Side: Learning illustration image background and welcome message.
 - Right Side: Login form on a white/lighter green background.
- Form Elements: Email input, Password input, "Remember me" checkbox, "Forgot Password" link.
- Social Login: Quick login buttons via Google, Facebook.

Event Handling:

- Click "Sign in":
 - Validates input (Empty email, short password).
 - Sends login request to the server.
 - If successful: Redirects to Dashboard.
 - If failed: Displays red error message below the input field.
- Click "Forgot password?": Redirects to the password recovery screen.
- Click "Sign up": Redirects to the Register Screen.



5.2.3 Dashboard Screen:

The user's central control hub, where all learning activities begin.

Presentation Format:

- Welcome Section: Personalized greeting "Welcome back, [Name]" with an emoji.
- Day Streak Badge: Displays the number of consecutive learning days with a flame icon to motivate users.
- Stats Cards: 3 quick statistic cards (Total Practice, Topics Completed, Average Score) with a dominant green color theme.
- Topics Grid: Grid list of Topic Cards, each with an Emoji, Topic Name, and Progress Bar.
- Random Practice CTA: A prominent block encouraging users to take a random practice test.

Event Handling:

- Scroll: Header remains fixed.
- Click Topic Card: Navigates to Topic details or starts the first lesson of that Topic (Practice Session).
- Click "Start Random Practice": Immediately navigates to Practice Session with random content.
- Click Navigation (Progress): Directs to the User Progress screen.

The screenshot shows the homepage of the English Learning Speaking System. At the top, it displays "Welcome back, Sarah! 🎉" and encourages users to "Keep practicing and improve your speaking skills". A "DAY STREAK" badge indicates 7 consecutive days with the message "Keep it going!". Below this, there are three summary boxes: "TOTAL PRACTICE" (142 minutes this week), "TOPICS COMPLETED" (4 out of 6), and "AVERAGE SCORE" (78% and improving). The main section, "Choose a Topic to Practice", features six categories arranged in a 2x3 grid: Greetings (85% complete, Start button), Self-introduction (60% complete, Start button), Work & Career (45% complete, Start button); Travel (30% complete, Start button), Shopping (15% complete, Start button), and Dining (70% complete, Start button). A "Ready for a Challenge?" section at the bottom suggests random practice mode with a "Start Random Practice" button.

E English Learning Speaking System

Welcome back, Sarah! 🎉

Keep practicing and improve your speaking skills

Home My Progress History S

TOTAL PRACTICE
142 minutes this week

TOPICS COMPLETED
4 out of 6

AVERAGE SCORE
78% and improving

Choose a Topic to Practice

Greetings 85%
85% complete
Start

Self-introduction 60%
60% complete
Start

Work & Career 45%
45% complete
Start

Travel 30%
30% complete
Start

Shopping 15%
15% complete
Start

Dining 70%
70% complete
Start

Ready for a Challenge?

Try our random practice mode to test your skills with sentences at your current level.

Start Random Practice

5.2.4 Practice Screen

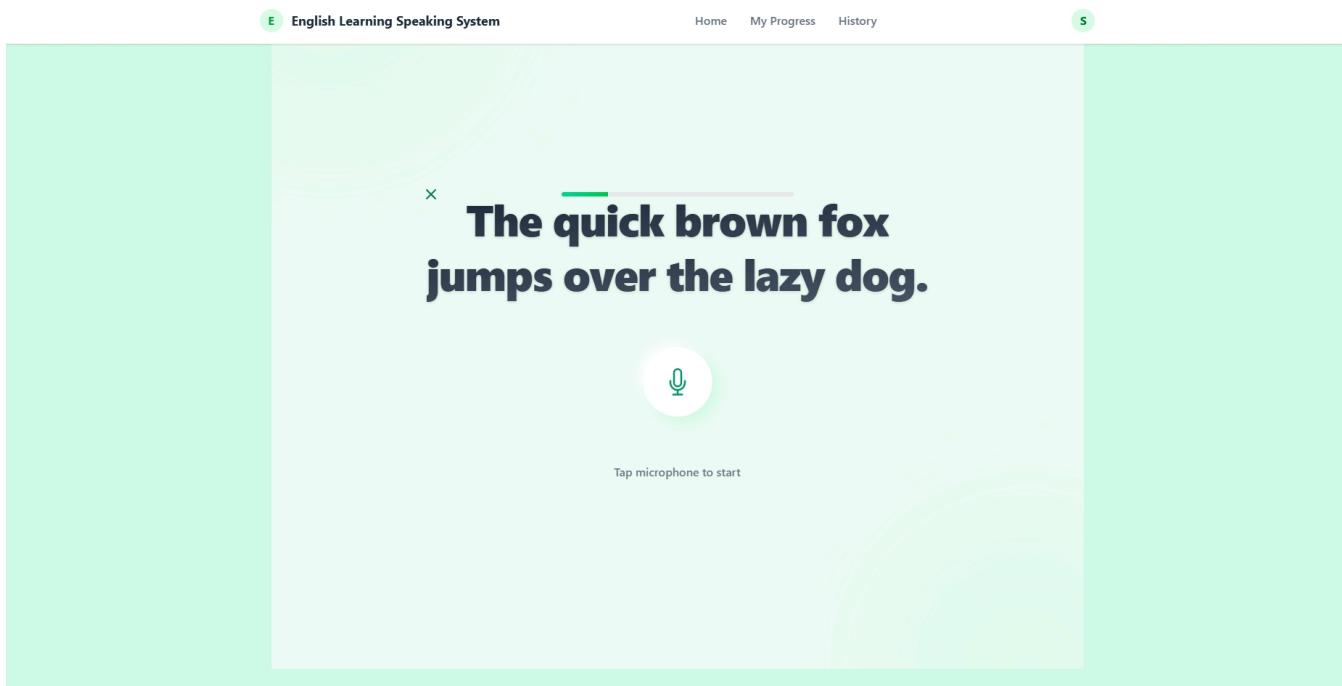
The core screen where users interact with AI to practice pronunciation.

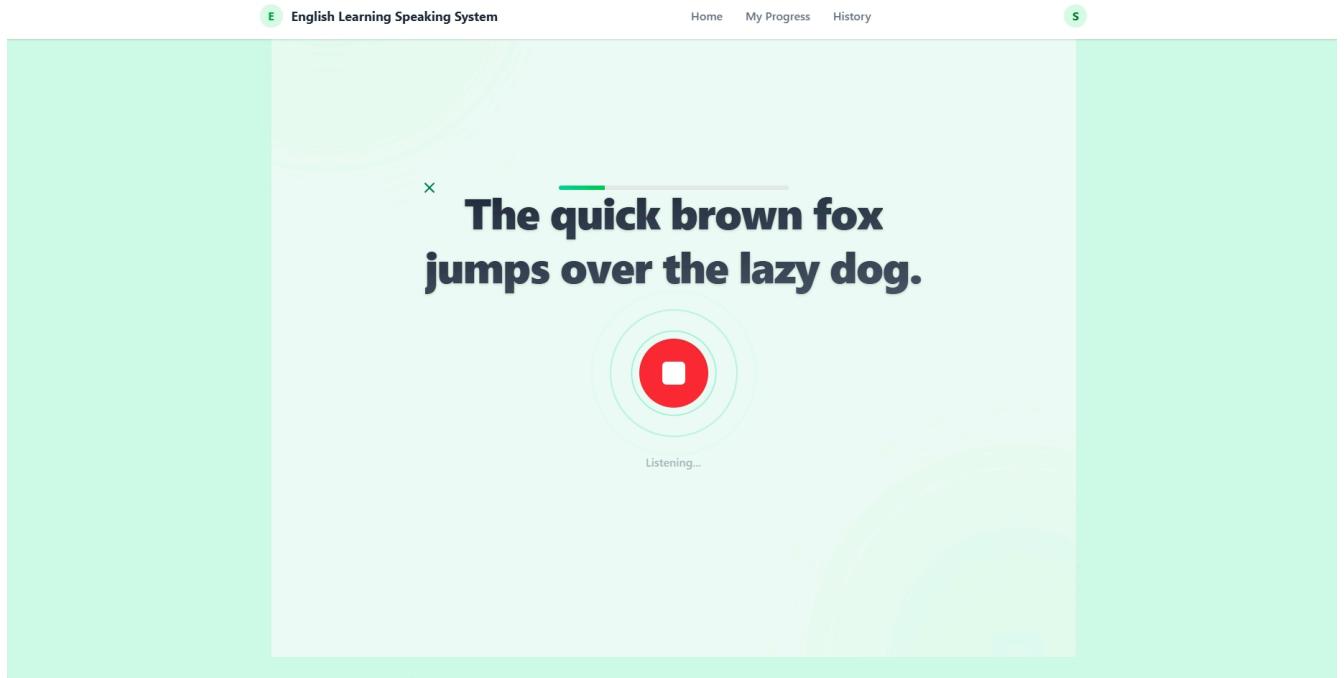
Presentation Format:

- Minimalist Interface: Distractions removed to focus entirely on the lesson content.
- Instruction Text: Status instruction text (Listening, Analyzing...).
- Hero Text: Sample sentence to be read displayed in large font at the center.
- Record Button: Large circular button at the bottom with a "Pulse" animation when recording.
- Progress Indicator: Bar showing current lesson progress (e.g., Sentence 2/5).

Event Handling:

- Click "Record": Activates Microphone, UI state changes to "Recording", visual sound wave effect appears.
- Click "Stop": Stops recording, sends audio file for processing. Displays Loading spinner.
- Completion: Upon finishing a sentence, automatically displays the Feedback Sheet.





5.2.5 Progress Report Screen

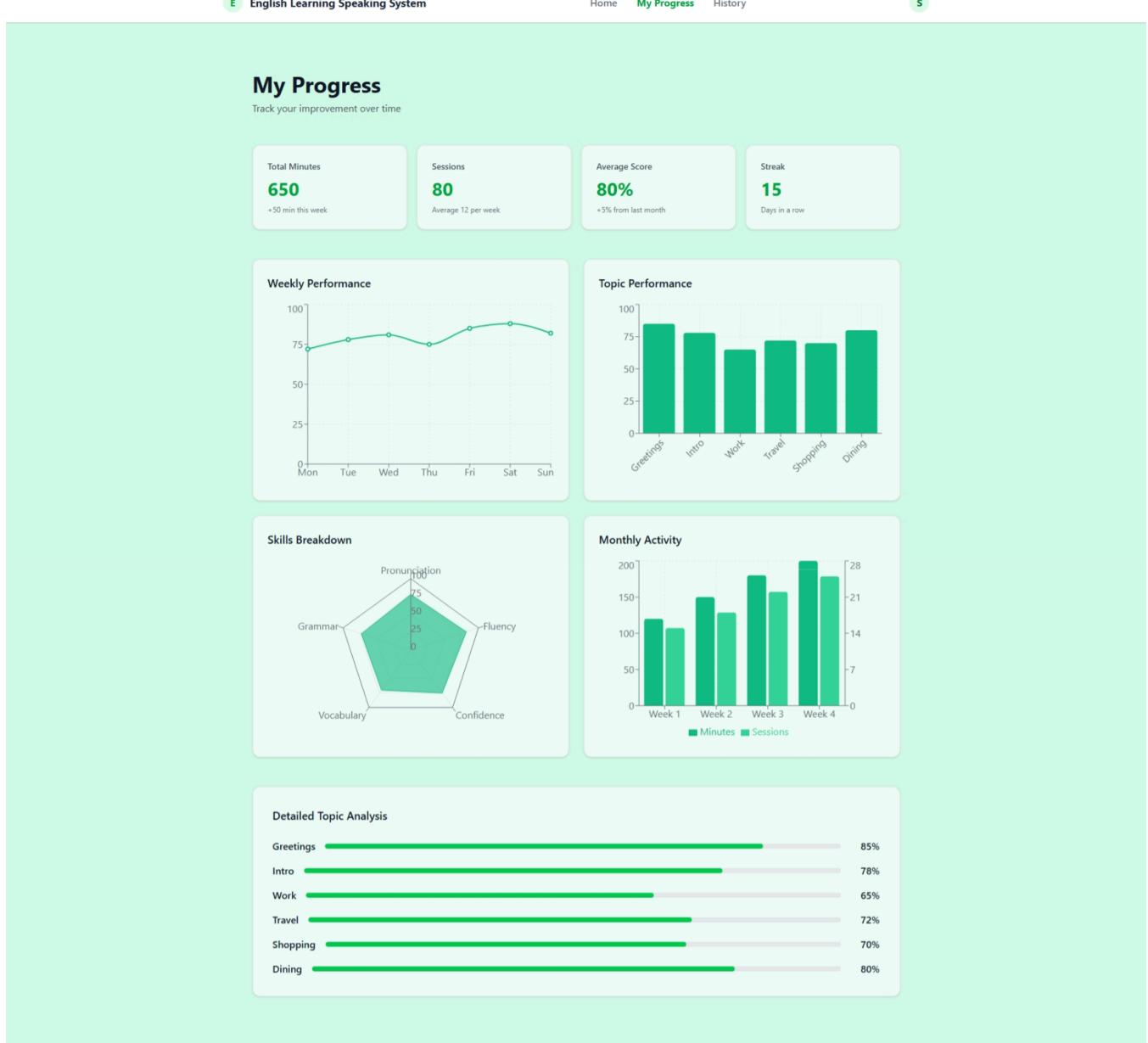
Helps users track their improvement via visual charts.

Presentation Format:

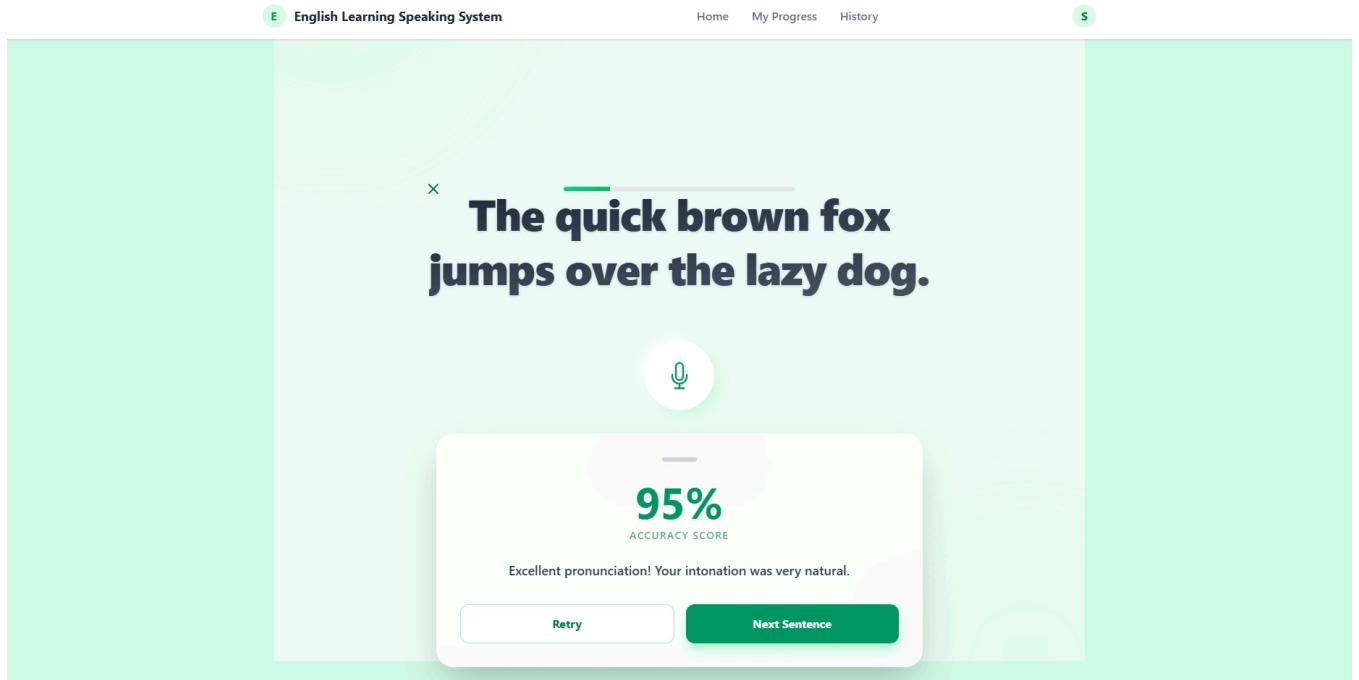
- Charts:
 - Weekly Performance (Line Chart): Score trends over the week.
 - Skills Radar (Radar Chart): Multi-dimensional assessment (Pronunciation, Fluency, Vocabulary...).
 - Topic Performance (Bar Chart): Score comparison across topics.
 - Detailed List: Detailed list of topics and corresponding scores.

Event Handling:

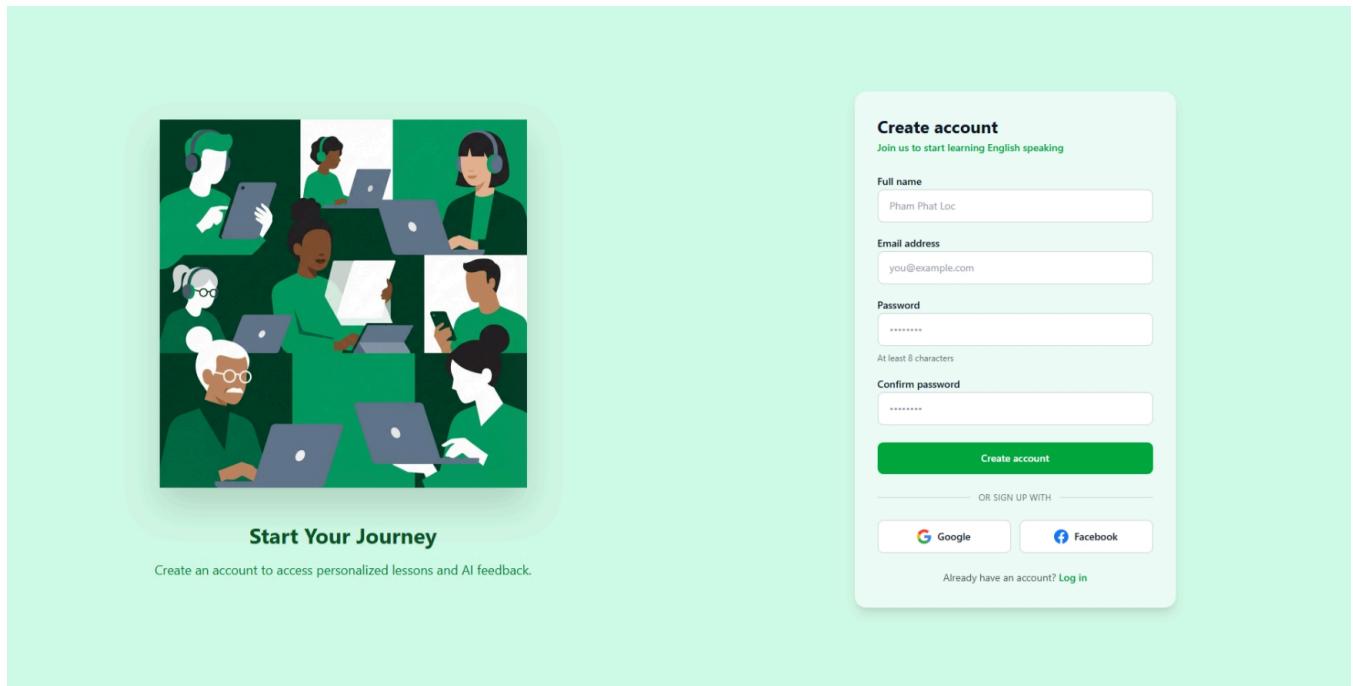
- Hover Chart: Displays Tooltip with exact figures at data points.



5.2.6 Exercise Screen:



5.2.7 Register Screen:



5.2.8 Forgot Password Screen: