**LinguaEase - AI-Powered Language Learning**

**App**

**A Project Report**

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (SOFTWARE DEVELOPMENT)**

**BY**

**Name: Shubham Jayram Parab**

**UID: UID No:21BSD015**

**Under the esteemed guidance of**

**Prof Mr. Wilson Rao**

**Co-ordinator**

**and**

**Ms.** **Sunita Jena**

**Assistant Professor**

**DEPARTMENT OF SOFTWARE DEVELPOMENT**

**JAI HIND COLLEGE**

****

**(*Autonomous)***

**MUMBAI, 400020**

**MAHARASHTRA**2024-25

**JAI HIND COLLEGE**

***(Autonomous)***

**MUMBAI, 400020 MAHARASHTRA**

**DEPARTMENT OF SOFTWARE DEVELOPMENT**



**CERTIFICATE**

This is to certify that the project entitled, “LinguaEase - AI-Powered Language LearningApp”, is Bonaﬁed work of Shubham Jayram Parab bearing UID (21BSD015) submitted in partial Fulﬁllment of the requirements for the award of degree of BACHELOR OF SCIENCE in Software Development from Jai Hind College Autonomous (University of Mumbai).

**Internal Guide: \_\_\_\_\_\_\_\_\_\_\_\_                                Coordinator: \_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_ External Examiner: \_\_\_\_\_\_\_\_\_\_\_**

**College Seal**

**DECLARATION**

I hereby declare that the project entitled,“LinguaEase - AI-Powered Language Learning App” done at Jai Hind College, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulﬁllment of the requirements for the award of degree of BACHELOR OF SCIENCE (Software Development) to be submitted as ﬁnal semester project as part of our curriculum.

**Name and Signature of the Student**

**ACKNOWLEDGEMENT**

I would like to express my thanks to the people who have helped me most throughout my project. I am grateful to my guide Ms. **Sunita Jena** for nonstop support for the project. I can’t say thank you enough for her tremendous support and help.

I owe my deep gratitude to our HOD of Software Development Department Mr. Wilson Rao who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

At last but not the least I want to thank all of my friends who helped/treasured me out in completing the project, where they all exchanged their own interesting ideas, thoughts and made this possible to complete my project with all accurate information. I wish to thank my parents for their personal support or attention who inspired/encouraged me to go my own way.

**INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr No** |  |  | **PARTICULARS** | **PAGE**  **NO.** | **DATE** |
| **1** |  |  | **Introduction** | **8-25** |  |
|  | 1.1 |  | Background | **8** |  |
|  | 1.2 |  | Objectives | **9** |  |
|  | 1.3 |  | Purpose, Scope, and Applicability | **10-** |  |
|  | 1.3.1 |  | Purpose | **10** |  |
|  | 1.3.2 |  | Scope | **11** |  |
|  | 1.3.3 |  | Applicability | **12** |  |
|  | 1.4 |  | Achievements |  |  |
|  | 1.5 |  | Organization of Report | **13-18** |  |
| **2** |  |  | **Survey Of Technologies** | **19** |  |
| **3** |  |  | **Requirement And Analysis** | **20-23** |  |
|  | 3.1 |  | Problem Definition | **20** |  |
|  | 3.2 |  | Requirement Specification | **21** |  |
|  | 3.3 |  | Planning and Scheduling |  |  |
|  | 3.4 |  | Software and Hardware Requirements | **22** |  |
|  | 3.5 |  | Preliminary Product Description | **23** |  |
|  | 3.6 |  | Conceptual Models |  |  |
|  |  |  |  |  |  |
|  | 3.6.2 |  | Use Case Diagram | **24** |  |
|  | 3.6.3 |  | Entity Relationship Diagram | **25** |  |
|  | 3.6.4 |  | Class Diagram | **26** |  |
|  | 3.6.5 |  | Object Diagram | **27** |  |
|  | 3.6.6 |  | Activity Diagram | **28** |  |
|  | 3.6.7 |  | Sequence Diagram | **29** |  |
|  | 3.6.8 |  | State Flow Diagram | **30** |  |
|  | 3.6.9 |  | Context Diagram | **31** |  |
|  | 3.6.10 |  | Data Flow Diagram | **32** |  |
|  | 3.6.11 |  | Component Diagram | **33** |  |
|  | 3.6.12 |  | Package Diagram | **34** |  |
|  | 3.6.13 |  | Deployment Diagram | **35** |  |
| **4** |  |  | **System Coding** | **37-** |  |
|  | 4.1 |  | Code | **37-88** |  |
|  | 4.2 |  | Data Dictionary |  |  |
|  | 4.3 |  | Program Description | **89** |  |
|  | 4.4 |  | Naming Conventions | **90** |  |
|  | 4.5 |  | Validations | **91** |  |
| **5** |  |  | **Program Listing** | **92-** |  |
|  | 5.1 |  | Cost Estimation | **92** |  |
|  | 5.2 |  | Schema Design |  |  |
|  | 5.3 |  | Game Play ScreenShots | **93-94** |  |
|  | 5.4 |  | Test Cases Design | **95-96** |  |
| **6** |  |  | **Conclusion** | **97-101** |  |
|  | 6.1 |  | Conclusion | **97** |  |
|  | 6.2 |  | Limitations of the System | **98-99** |  |
|  | 6.3 |  | Future Scope of the Project | **100** |  |
| **7** |  |  | **Bibliography** | **101** |  |

# **1. INTRODUCTION**

# **1.1 Background**

# The increasing need for language skills in education, business, and travel has fueled the demand for accessible and effective learning solutions.

# Mobile apps, being ubiquitous, provide an ideal platform for language learning.

# LinguaEase combines AI-powered personalization, gamification, and an offline mode to offer an engaging and effective learning experience.

# **1.2 Objectives**

# To create a mobile app for learning Japanese, German, and French, catering to beginners and intermediate learners. -

# To provide AI-driven features like speech recognition and personalized learning paths. -

# To gamify the language-learning process with progress tracking and rewards. -

# To enable offline access for selected modules to ensure learning without internet dependency.

# **1.3 Purpose, Scope, and Applicability**

# **1.3.1 Purpose**

# LinguaEase is designed to simplify and enhance the language-learning process for users through interactive modules, speech analysis, and real-time feedback. Its purpose includes:

# **Personalized Learning Experience** – Adapts to individual learning styles and progress to provide a customized experience.

# **AI-Driven Language Assistance** – Utilizes AI to offer grammar corrections, pronunciation evaluation, and vocabulary recommendations.

# **Offline Learning Capabilities** – Allows users to access lessons and practice without an internet connection.

# **Gamified Learning Approach** – Engages users through interactive quizzes, crosswords, and challenges to improve retention.

# **Speech Recognition & Feedback** – Helps users refine pronunciation through real-time AI-driven speech analysis.

# **Multilingual Support** – Enables users to learn multiple languages, including French, German, Spanish, and Japanese, within a single platform.

# **Progress Tracking & Analytics** – Monitors user performance and suggests areas of improvement.

# **Interactive Chatbot for Practice** – Provides an AI-based conversational partner for language practice without relying on external APIs.

# **Error Detection & Correction** – Implements Levenshtein Distance for spelling correction and text comparison to enhance learning accuracy.

# **Adaptive Quiz System** – Adjusts question difficulty based on user performance to ensure an optimal learning curve.

# **User-Friendly Interface** – Designed for ease of use, making language learning accessible to beginners and advanced learners alike.

### 1.3.2 Scope

- Multilingual support for Japanese, German, and French.

- AI-driven speech recognition and grammar improvement tools.

- Interactive quizzes, vocabulary builders, and grammar lessons.

- Offline access for core learning features.

# 1.3.3 Applicability

# - Students seeking to expand language skills.

# - Professionals and travelers needing conversational proficiency.

# - Language enthusiasts looking for an engaging learning platform.

# 1.5 Organization of Report

# This report includes:

# 1. Introduction to the app and its features.

# 2. Overview of technologies used in development.

# 3. Requirement analysis, design, and implementation strategies.

# 4. Testing, results, and conclusions.

**2. SURVEY OF TECHNOLOGIES**

- Android Studio: Development environment for mobile app creation.

- Firebase: Backend for user authentication and data management.

- TensorFlow Lite: Integration of AI for speech recognition and analysis.

- Kotlin/Java: Programming languages for developing the app.

- SQLite: Database for offline data storage.

- Figma: Design tool for prototyping UI/UX.

**3. REQUIREMENT AND ANALYSIS**

**3.1 Problem Definition**

Language learners face the following challenges:  
- Lack of personalized content.  
- Difficulty in practicing pronunciation.  
- Limited options for offline learning.  
- Tedious and monotonous learning experiences.  
  
LinguaEase tackles these issues by incorporating AI, gamification, and offline capabilities into a mobile platform.

**3.2 Requirement Specification**

**Functional Requirements:**

- User authentication and profile management.  
- AI-based speech recognition for pronunciation analysis.  
- Vocabulary and grammar modules with progress tracking.  
- Gamified quizzes and rewards system.  
- Notifications and reminders for learning consistency.

**Non-Functional Requirements:**

- High responsiveness and ease of use.  
- Secure data storage and privacy compliance.  
- Optimized for low resource consumption.  
- Support for Android devices running version 6.0 and above.

**3.3 Planning and Scheduling**

The development of the language-learning application follows a structured timeline to ensure systematic progress. Below is the breakdown of different phases along with their respective timelines:

| **Phase** | **Timeline** | **Description** |
| --- | --- | --- |
| **Requirement Gathering** | Week 1 - Week 2 | Identify project scope, define objectives, and list required features. |
| **Design Prototyping** | Week 3 - Week 4 | Create UI/UX wireframes, design system architecture, and finalize database schema. |
| **Development** | Week 5 - Week 10 | Implement core features, integrate AI modules, and develop quizzes & speech recognition. |
| **Testing and Debugging** | Week 11 - Week 12 | Perform unit testing, integration testing, and user testing to ensure functionality. |
| **Deployment** | Week 13 | Launch the application, deploy AI models, and prepare for user onboarding. |

This timeline provides a structured roadmap to manage the project efficiently. Adjustments may be made as necessary based on development progress and testing feedback.

**3.4 Software and Hardware Requirements**

**Software Requirements:**

- Android Studio (Latest Version).  
- Firebase SDK for backend integration.  
- TensorFlow Lite for AI features.  
- SQLite for local database storage.

**Hardware Requirements:**

- Development machine with 8GB RAM, 256GB SSD.  
- Android device with version 6.0 or higher for testing.

**3.5 Preliminary Product Description**

LinguaEase features an intuitive interface, interactive learning modules, and AI-powered feedback for language learners. The app includes a user-friendly dashboard, progress tracking, and gamified elements like daily streaks and achievement badges.

**3.6 Concept Models**

**User Interaction Flow:**

1. Onboarding: User registers/logs in → Chooses language and level → Sets daily goals.  
2. Learning Modules:  
- Vocabulary builder.  
- Grammar lessons with exercises.  
- Pronunciation practice using AI.  
- Interactive quizzes.  
3. Rewards and Progress Tracking:  
- Displays progress charts and awards badges for milestones.

**System Architecture:**

1. Frontend: Designed in Android Studio using Kotlin/Java.  
2. Backend: Firebase for real-time data management and analytics.  
3. AI Module: TensorFlow Lite for speech and language processing.  
4. Offline Mode: SQLite for local storage of vocabulary and grammar content.

Use Cases Diagram

1. Register/Login

2. Select Course

3. Access Lessons

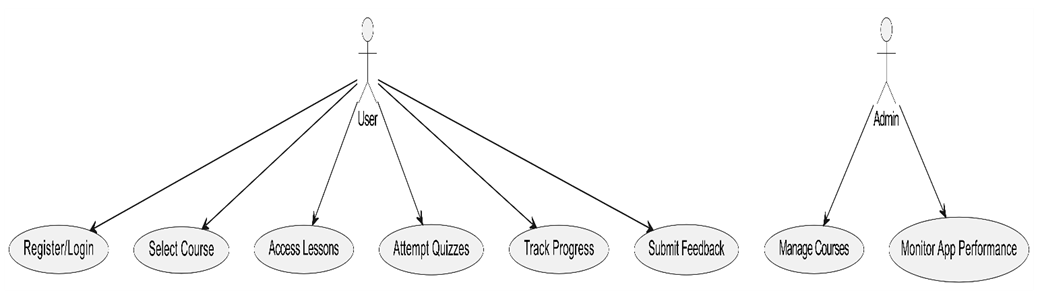
4. Attempt Quizzes

5. Track Progress

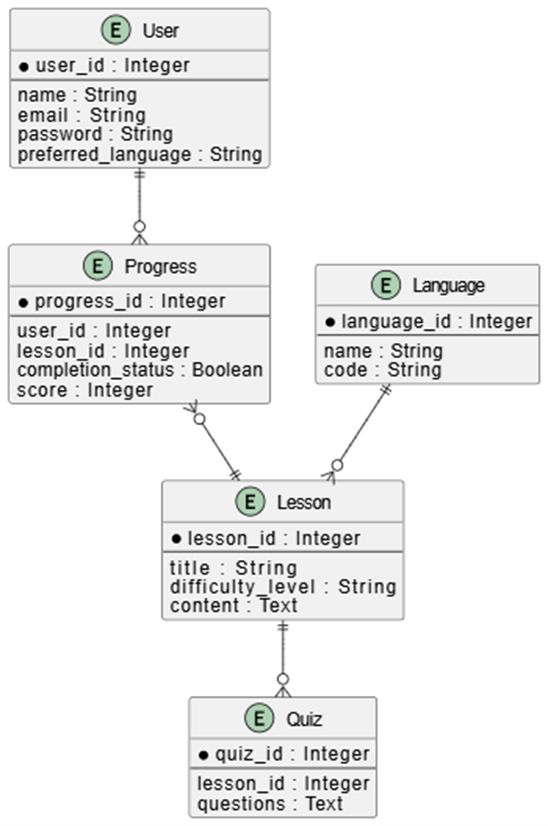
6. Submit Feedback

7. Manage Courses (Admin)

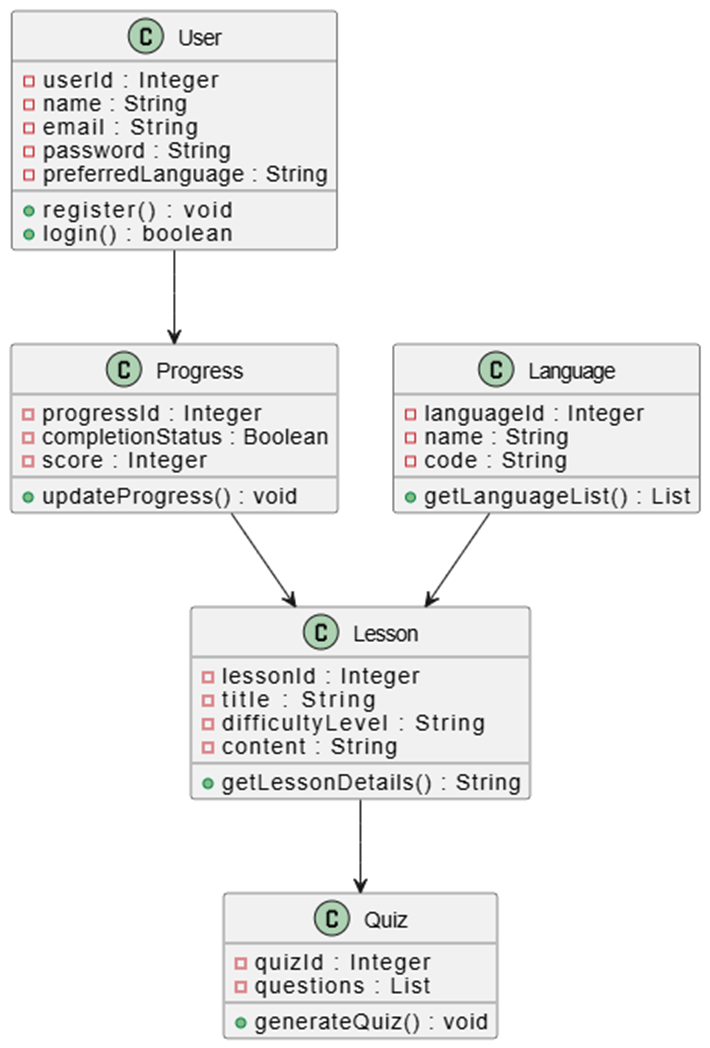
8. Monitor App Performance (Admin)



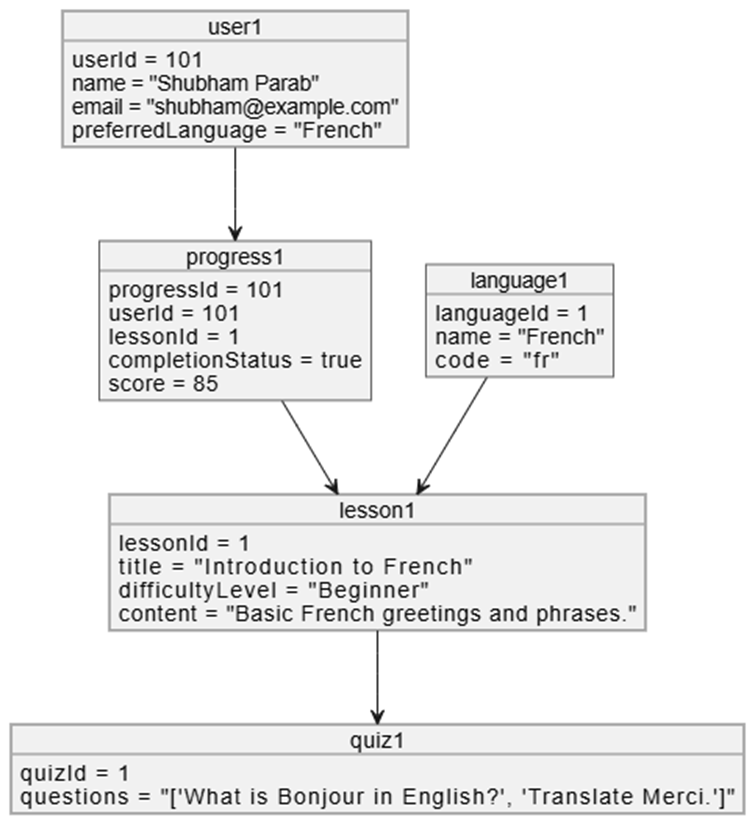
ER Entity Relationship Diagram :



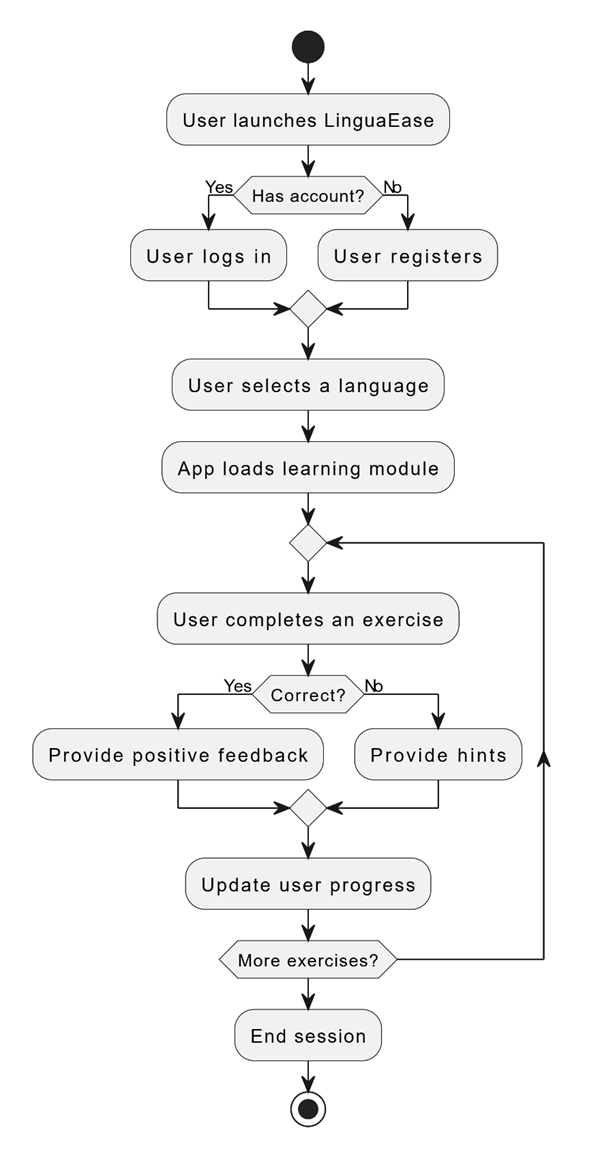
Class Diagram:



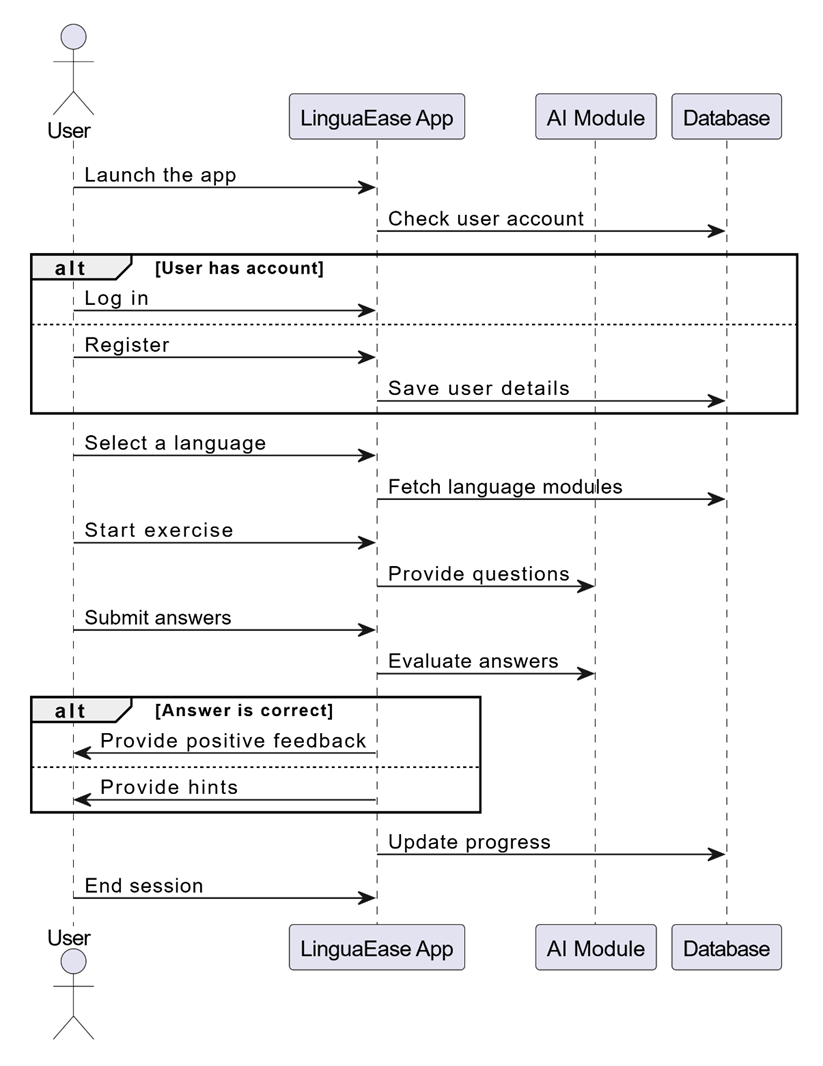
Object Diagram:



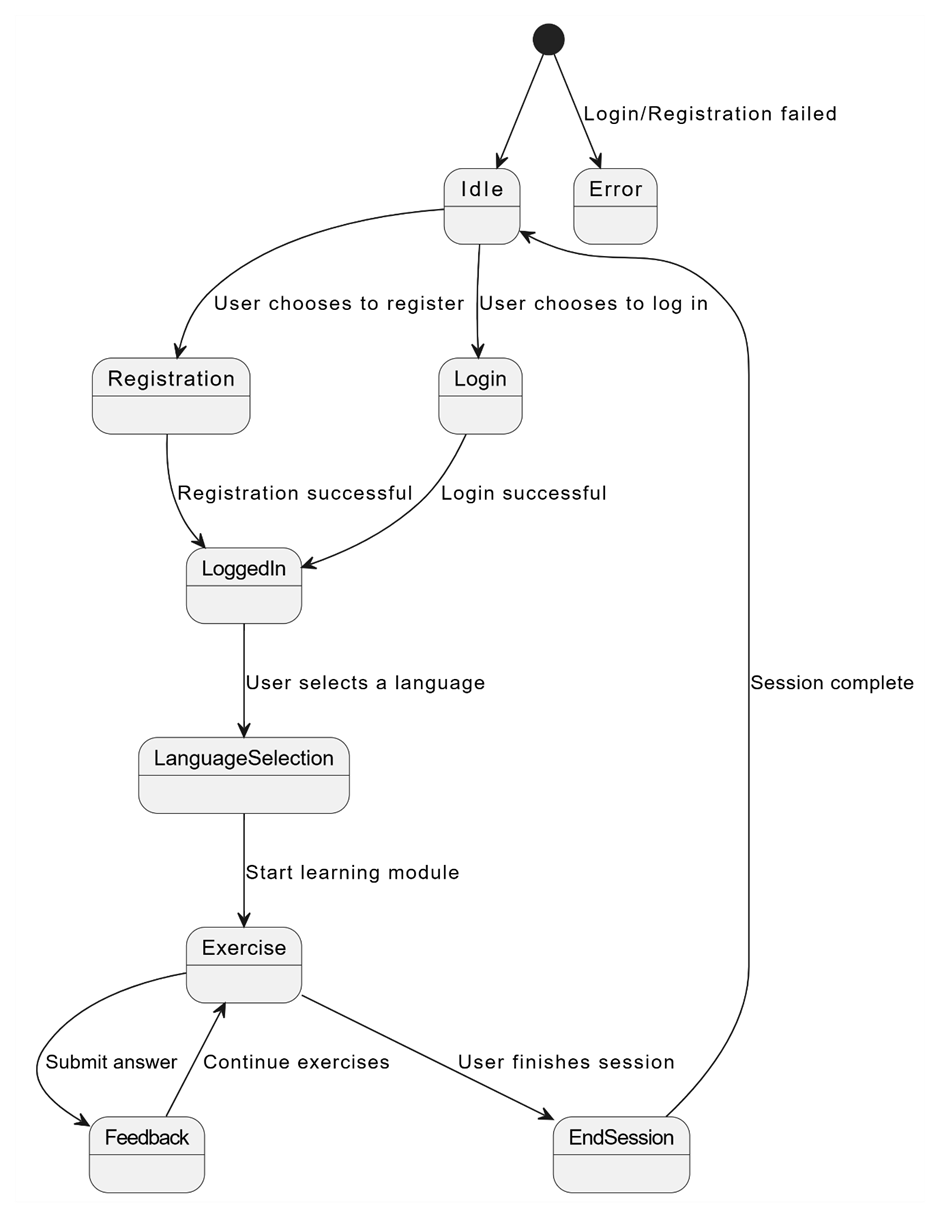
Activity Diagram:



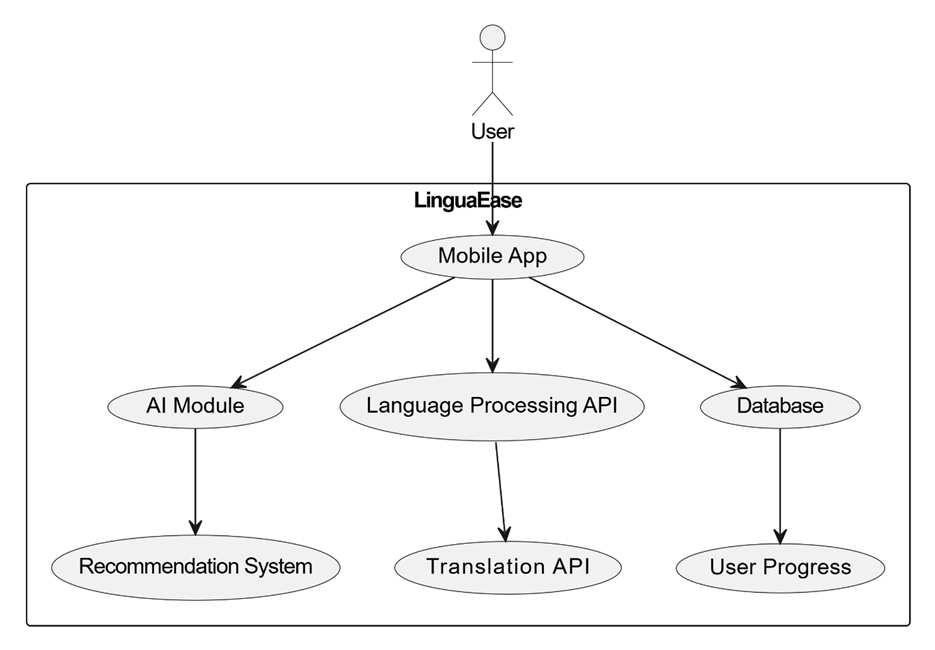
Sequence Diagram:



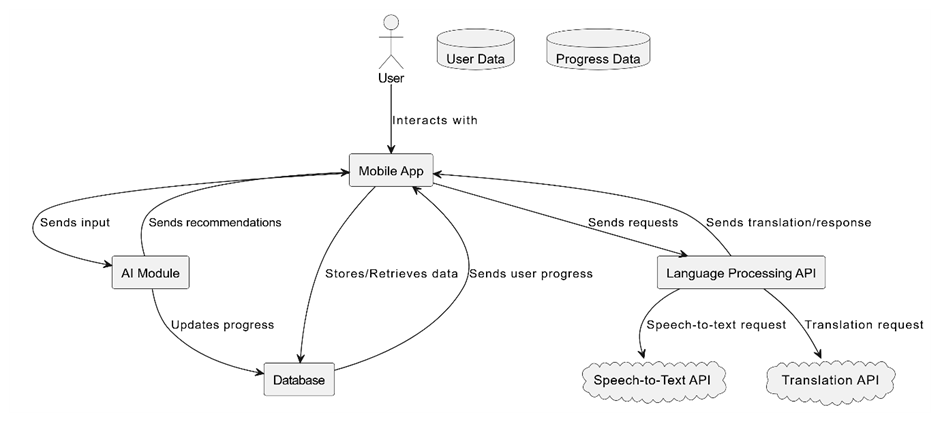
State Flow Diagram:



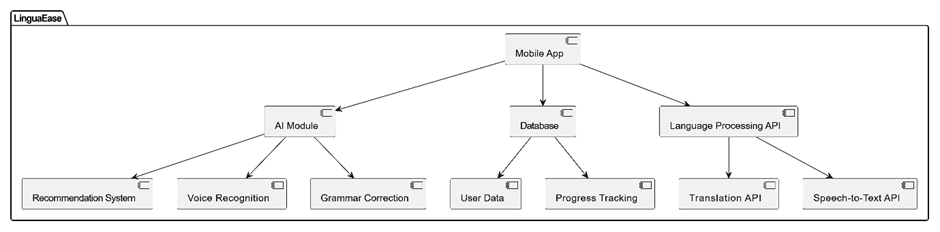
Context Diagram:



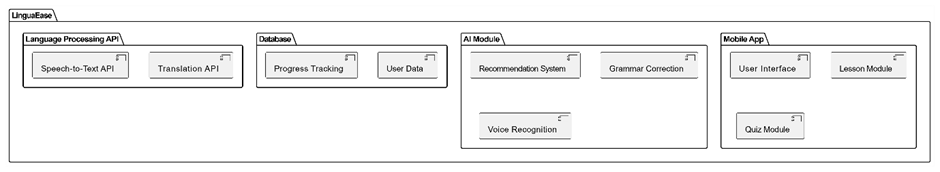
Data Flow Diagram:



Component Diagram:



Package Diagram:



System Coding:

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools">

<!-- Permissions -->

<uses-permission android:name="android.permission.RECORD\_AUDIO"/>

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.VIBRATE" />

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@drawable/ic\_launcher\_foreground"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.MyApplicationApp"

tools:targetApi="31">

<!-- Main Activity (Launcher) -->

<activity

android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<!-- Authentication Activities -->

<activity android:name=".LoginActivity" android:exported="false" />

<activity android:name=".SignupActivity" android:exported="false" />

<!-- Home & More Activities -->

<activity android:name=".HomeActivity" android:exported="true" />

<activity android:name=".MoreActivity" android:exported="true" />

<!-- Language-Specific Activities -->

<activity android:name=".FrenchActivity" android:exported="false" />

<activity android:name=".SpanishActivity" android:exported="false" />

<activity android:name=".JapaneseActivity" android:exported="false" />

<activity android:name=".GermanActivity" android:exported="false" />

<activity android:name=".ProfileActivity" android:exported="false" />

<activity android:name=".HelpActivity" android:exported="false" />

<!-- Picture Recognition Activity -->

<activity android:name=".FrenchPictureRecognitionActivity" android:exported="false" />

<activity android:name=".GermanPictureRecognitionActivity" android:exported="false"/>

<activity android:name=".JapanesePictureRecognitionActivity" android:exported="false"/>

<activity android:name=".SpanishPictureRecognitionActivity" android:exported="false"/>

<!-- Quiz Activities -->

<activity android:name=".FrenchQuizActivity" android:exported="false"/>

<activity android:name=".JapaneseQuizActivity" android:exported="false"/>

<activity android:name=".SpanishQuizActivity" android:exported="false"/>

<activity android:name=".GermanQuizActivity" android:exported="false"/>

<!-- Speech Recognition Activities (Fixed Naming) -->

<activity android:name=".FrenchSpeechRecognitionActivity" android:exported="false"/>

<activity android:name=".JapaneseSpeechRecognitionActivity" android:exported="false"/>

<activity android:name=".GermanSpeechRecognitionActivity" android:exported="false"/>

<activity android:name=".SpanishSpeechRecognitionActivity" android:exported="false"/>

<!-- Crossword Activities -->

<activity android:name=".FrenchCrosswordActivity" android:exported="false"/>

<activity android:name=".SpanishCrosswordActivity" android:exported="false"/>

<activity android:name=".GermanCrosswordActivity" android:exported="false"/>

<activity android:name=".JapaneseCrosswordActivity" android:exported="false"/>

<!-- Learning Options Activities -->

<activity android:name=".LearningOptionsActivity\_French" android:exported="false" />

<activity android:name=".LearningOptionsActivity\_German" android:exported="false" />

<activity android:name=".LearningOptionsActivity\_Japanese" android:exported="false" />

<activity android:name=".LearningOptionsActivity\_Spanish" android:exported="false" />

</application>

</manifest>

FrenchActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class FrenchActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_french*);  
  
 Button btnStartLearning = findViewById(R.id.*btn\_start\_learning*);  
  
 // Navigate to French Quiz Activity  
  
 btnStartLearning.setOnClickListener(v -> {  
 Intent intent = new Intent(FrenchActivity.this, LearningOptionsActivity\_French.class);  
 startActivity(intent);  
 });  
  
 }  
}

FrenchCrosswordActivity.java

package com.example.myapplicationapp;  
  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class FrenchCrosswordActivity extends AppCompatActivity {  
  
 private FrenchCrosswordGridAdapter adapter;  
 private SharedPreferences prefs;  
 private static final String *PREFS\_NAME* = "CrosswordPrefs";  
  
 // Updated 10x10 crossword grid with correct word structure  
 private String[][] crosswordGrid = {  
 {"C", "", "A", "",},  
 {"", "H", "", "E", "",},  
 {"U","", "A", "", "D", ""},  
 {"M", "", "I", "", "O", "", "N"},  
 {"", "O", "", "T", "", "R", "",},  
 { "É", "", "O", "", "E",},  
 {"P", "O", "", "M", "", },  
 {"", "O", "U", "", "E",},  
 {"M", "", "R", "", "O", "", "E"}  
 };  
  
 private String[] words = {"CHAT", "CHIEN", "MAISON", "VOITURE", "ÉCOLE", "JARDIN", "POMME", "ROUGE", "MER", "SOLEIL"};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*french\_crossword*);  
  
 GridView gridView = findViewById(R.id.*crosswordGrid*);  
 Button submitButton = findViewById(R.id.*btn\_submit*);  
 Button saveButton = findViewById(R.id.*btn\_save*);  
  
 prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 adapter = new FrenchCrosswordGridAdapter(this, crosswordGrid, prefs);  
  
 gridView.setAdapter(adapter);  
  
 submitButton.setOnClickListener(view -> checkAnswers());  
 saveButton.setOnClickListener(view -> saveProgress());  
 }  
  
 private void checkAnswers() {  
 if (adapter.isCrosswordCorrect(words)) {  
 Toast.*makeText*(this, "Bravo! You completed the crossword!", Toast.*LENGTH\_LONG*).show();  
 } else {  
 Toast.*makeText*(this, "Try again! Some words are incorrect.", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 private void saveProgress() {  
 adapter.saveUserProgress();  
 Toast.*makeText*(this, "Progress saved!", Toast.*LENGTH\_SHORT*).show();  
 }  
}

FrenchCrosswordGridAdapter.java

package com.example.myapplicationapp;  
  
import android.content.Context;  
import android.content.SharedPreferences;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.EditText;  
import java.util.HashSet;  
  
public class FrenchCrosswordGridAdapter extends BaseAdapter {  
 private Context context;  
 private String[][] grid;  
 private EditText[][] inputCells;  
 private SharedPreferences prefs;  
  
 public FrenchCrosswordGridAdapter(Context context, String[][] grid, SharedPreferences prefs) {  
 this.context = context;  
 this.grid = grid;  
 this.prefs = prefs;  
 this.inputCells = new EditText[grid.length][grid[0].length];  
 }  
  
 @Override  
 public int getCount() {  
 return grid.length \* grid[0].length;  
 }  
  
 @Override  
 public Object getItem(int position) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
 return grid[row][col];  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
  
 // Prevent index out of bounds error  
 if (row >= grid.length || col >= grid[row].length) {  
 return new View(context);  
 }  
  
 EditText cell;  
 if (convertView == null) {  
 cell = new EditText(context);  
 } else {  
 cell = (EditText) convertView;  
 }  
  
 if (!grid[row][col].equals("")) {  
 cell.setText(grid[row][col]);  
 cell.setEnabled(false);  
 } else {  
 String savedValue = prefs.getString(row + "\_" + col, "");  
 cell.setText(savedValue);  
 cell.setHint("\_");  
 }  
  
 return cell;  
 }  
  
 public boolean isCrosswordCorrect(String[] correctWords) {  
 HashSet<String> enteredWords = new HashSet<>();  
 StringBuilder wordBuilder = new StringBuilder();  
  
 for (int i = 0; i < grid.length; i++) {  
 wordBuilder.setLength(0);  
 for (int j = 0; j < grid[i].length; j++) {  
 String letter = inputCells[i][j].getText().toString().toUpperCase();  
 if (!letter.isEmpty()) {  
 wordBuilder.append(letter);  
 }  
 }  
 if (wordBuilder.length() > 1) {  
 enteredWords.add(wordBuilder.toString());  
 }  
 }  
  
 for (String word : correctWords) {  
 if (!enteredWords.contains(word)) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
 public void saveUserProgress() {  
 SharedPreferences.Editor editor = prefs.edit();  
 for (int i = 0; i < grid.length; i++) {  
 for (int j = 0; j < grid[i].length; j++) {  
 editor.putString(i + "\_" + j, inputCells[i][j].getText().toString());  
 }  
 }  
 editor.apply();  
 }  
}

FrenchPictureRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Collections;  
import java.util.List;  
  
public class FrenchPictureRecognitionActivity extends AppCompatActivity {  
  
 private ImageView imageView;  
 private RadioGroup optionsGroup;  
 private Button nextButton, submitButton;  
  
 private final Question[] questions = {  
 // Cuisine  
 new Question(R.drawable.*baguette*, "Baguette", new String[]{"Croissant", "Baguette", "Fromage", "Vin", "Macaron"}),  
  
 // Art  
 new Question(R.drawable.*monalisa*, "Mona Lisa", new String[]{"Guernica", "Mona Lisa", "Le Cri", "Les Tournesols", "La Nuit étoilée"}),  
  
 // City  
 new Question(R.drawable.*eiffeltower*, "Paris", new String[]{"Lyon", "Marseille", "Paris", "Bordeaux", "Toulouse"}),  
  
 // History  
 new Question(R.drawable.*napoleon*, "Napoléon", new String[]{"Louis XIV", "Napoléon", "Charlemagne", "Clovis", "De Gaulle"}),  
  
 // Famous Landmark  
 new Question(R.drawable.*notredame*, "Notre-Dame", new String[]{"Mont Saint-Michel", "Château de Versailles", "Notre-Dame", "Arc de Triomphe", "Tour Eiffel"})  
 };  
  
  
 private int currentQuestionIndex = 0;  
 private boolean answered = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_french\_picture\_recongnition*);  
  
 imageView = findViewById(R.id.*imageView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 nextButton = findViewById(R.id.*btnNext*);  
 submitButton = findViewById(R.id.*btnSubmit*);  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 nextButton.setOnClickListener(v -> loadNextQuestion());  
 }  
  
 private void loadQuestion() {  
 if (currentQuestionIndex >= questions.length) {  
 Toast.*makeText*(this, "Quiz terminé!", Toast.*LENGTH\_LONG*).show();  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(false);  
 return;  
 }  
  
 Question question = questions[currentQuestionIndex];  
  
 imageView.setImageResource(question.imageResId);  
  
 // Shuffle options to randomize order  
 List<String> shuffledOptions = new ArrayList<>(Arrays.*asList*(question.options));  
 Collections.*shuffle*(shuffledOptions);  
  
 optionsGroup.removeAllViews(); // Clear previous options  
  
 // Ensure exactly 5 options  
 for (String option : shuffledOptions) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 radioButton.setTextSize(18);  
 radioButton.setPadding(10, 10, 10, 10);  
 optionsGroup.addView(radioButton);  
 }  
  
 optionsGroup.clearCheck(); // Reset selection  
 answered = false;  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(true);  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId != -1) {  
 RadioButton selectedButton = findViewById(selectedId);  
 String selectedAnswer = selectedButton.getText().toString();  
 String correctAnswer = questions[currentQuestionIndex].correctAnswer;  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 Toast.*makeText*(this, "Correct!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(this, "Incorrect! La bonne réponse est : " + correctAnswer, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 answered = true;  
 nextButton.setEnabled(true);  
 submitButton.setEnabled(false);  
 } else {  
 Toast.*makeText*(this, "Sélectionnez une réponse!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void loadNextQuestion() {  
 if (answered) {  
 currentQuestionIndex++;  
 loadQuestion();  
 } else {  
 Toast.*makeText*(this, "Soumettez d'abord votre réponse!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private static class Question {  
 int imageResId;  
 String correctAnswer;  
 String[] options;  
  
 Question(int imageResId, String correctAnswer, String[] options) {  
 this.imageResId = imageResId;  
 this.correctAnswer = correctAnswer;  
 this.options = options;  
 }  
 }  
}

FrenchQuizActivity.java

package com.example.myapplicationapp;  
  
import android.content.res.AssetFileDescriptor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import org.tensorflow.lite.Interpreter;  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.nio.MappedByteBuffer;  
import java.nio.channels.FileChannel;  
import java.util.Arrays;  
import java.util.List;  
  
public class FrenchQuizActivity extends AppCompatActivity {  
 private Interpreter tflite;  
 private TextView questionTextView;  
 private RadioGroup optionsGroup;  
 private Button submitButton;  
 private String[] questions = {  
 "Quelle est la capitale de la France ?",  
 "Comment dit-on 'apple' en français ?"  
 };  
 private String[][] options = {  
 {"Londres", "Paris", "Madrid", "Rome"},  
 {"Pomme", "Orange", "Banane", "Raisin"}  
 };  
 private int questionIndex = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_french\_quiz*);  
  
 questionTextView = findViewById(R.id.*questionTextView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 submitButton = findViewById(R.id.*submitButton*);  
  
 try {  
 tflite = new Interpreter(loadModelFile());  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 }  
  
 private MappedByteBuffer loadModelFile() throws IOException {  
 AssetFileDescriptor fileDescriptor = getAssets().openFd("model\_french.tflite");  
 FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());  
 FileChannel fileChannel = inputStream.getChannel();  
 return fileChannel.map(FileChannel.MapMode.*READ\_ONLY*, fileDescriptor.getStartOffset(), fileDescriptor.getDeclaredLength());  
 }  
  
 private void loadQuestion() {  
 if (questionIndex < questions.length) {  
 questionTextView.setText(questions[questionIndex]);  
 optionsGroup.removeAllViews();  
 for (String option : options[questionIndex]) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 optionsGroup.addView(radioButton);  
 }  
 } else {  
 questionTextView.setText("Quiz terminé !");  
 submitButton.setEnabled(false);  
 }  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId == -1) return;  
  
 int predictedIndex = predictAnswer(questions[questionIndex]);  
 int correctIndex = Arrays.*asList*(options[questionIndex]).indexOf(options[questionIndex][predictedIndex]);  
  
 RadioButton selectedRadioButton = findViewById(selectedId);  
 String selectedAnswer = selectedRadioButton.getText().toString();  
 String correctAnswer = options[questionIndex][correctIndex];  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 questionTextView.setText("Bonne réponse !");  
 } else {  
 questionTextView.setText("Mauvaise réponse. Réessayez.");  
 }  
  
 questionIndex++;  
 loadQuestion();  
 }  
  
 private int predictAnswer(String question) {  
 float[][] input = new float[1][20]; // Assume padded sequence size  
 float[][] output = new float[1][4]; // 4 answer choices  
  
 tflite.run(input, output);  
 return getMaxIndex(output[0]);  
 }  
  
 private int getMaxIndex(float[] arr) {  
 int maxIndex = 0;  
 for (int i = 1; i < arr.length; i++) {  
 if (arr[i] > arr[maxIndex]) maxIndex = i;  
 }  
 return maxIndex;  
 }  
}

FrenchSpeechRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.Manifest;  
import android.content.pm.PackageManager;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.os.CountDownTimer;  
import android.speech.tts.TextToSpeech;  
import android.speech.RecognizerIntent;  
import android.speech.SpeechRecognizer;  
import android.speech.RecognitionListener;  
import android.content.Intent;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
import java.util.ArrayList;  
import java.util.Locale;  
import java.util.Random;  
  
public class FrenchSpeechRecognitionActivity extends AppCompatActivity {  
 private TextToSpeech textToSpeech;  
 private SpeechRecognizer speechRecognizer;  
 private TextView wordToSpell, recognitionResult, scoreView, wordHistory, hintView, timerView, highScoreView;  
 private Button btnListen, btnSpeak, btnNext, btnRetry, btnHint, btnChangeLevel;  
 private CountDownTimer timer;  
  
 private String[][] phrases = {  
 {"bonjour", "merci", "au revoir", "s'il vous plaît", "oui", "non"}, // Beginner  
 {"je suis content", "quelle heure est-il", "pouvez-vous m'aider", "il fait beau aujourd'hui"}, // Intermediate  
 {"j'aimerais réserver une table", "je voudrais une chambre avec vue", "le train part à quelle heure"} // Advanced  
 };  
  
 private String correctWord = "";  
 private int score = 0, attempts = 0, highScore = 0, difficulty = 0;  
 private StringBuilder history = new StringBuilder();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_french\_speech\_recognition*);  
  
 wordToSpell = findViewById(R.id.*word\_to\_spell*);  
 recognitionResult = findViewById(R.id.*recognition\_result*);  
 scoreView = findViewById(R.id.*score\_view*);  
 wordHistory = findViewById(R.id.*word\_history*);  
 hintView = findViewById(R.id.*hint\_view*);  
 timerView = findViewById(R.id.*timer\_view*);  
 highScoreView = findViewById(R.id.*high\_score\_view*);  
 btnListen = findViewById(R.id.*btn\_listen*);  
 btnSpeak = findViewById(R.id.*btn\_speak*);  
 btnNext = findViewById(R.id.*btn\_next*);  
 btnRetry = findViewById(R.id.*btn\_retry*);  
 btnHint = findViewById(R.id.*btn\_hint*);  
 btnChangeLevel = findViewById(R.id.*btn\_change\_level*);  
  
 requestAudioPermission();  
 setNewWord();  
  
 textToSpeech = new TextToSpeech(this, status -> {  
 if (status == TextToSpeech.*SUCCESS*) {  
 int result = textToSpeech.setLanguage(Locale.*FRENCH*);  
 if (result == TextToSpeech.*LANG\_MISSING\_DATA* || result == TextToSpeech.*LANG\_NOT\_SUPPORTED*) {  
 Toast.*makeText*(getApplicationContext(), "French TTS not supported!", Toast.*LENGTH\_SHORT*).show();  
 }  
 } else {  
 Toast.*makeText*(getApplicationContext(), "TTS Initialization failed!", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 speechRecognizer = SpeechRecognizer.*createSpeechRecognizer*(this);  
 speechRecognizer.setRecognitionListener(new RecognitionListener() {  
 @Override  
 public void onResults(Bundle results) {  
 ArrayList<String> matches = results.getStringArrayList(SpeechRecognizer.*RESULTS\_RECOGNITION*);  
 if (matches != null && !matches.isEmpty()) {  
 String recognizedText = matches.get(0);  
 recognitionResult.setText("You said: " + recognizedText);  
  
 if (recognizedText.equalsIgnoreCase(correctWord)) {  
 recognitionResult.setText("✅ Correct!");  
 recognitionResult.setTextColor(Color.*GREEN*);  
 score++;  
 updateHistory(correctWord + " ✅");  
 disableRetry();  
 updateHighScore();  
 } else {  
 recognitionResult.setText("❌ Try again!");  
 recognitionResult.setTextColor(Color.*RED*);  
 updateHistory(correctWord + " ❌");  
 enableRetry();  
 }  
 attempts++;  
 updateScore();  
 }  
 }  
  
 @Override public void onReadyForSpeech(Bundle params) {}  
 @Override public void onBeginningOfSpeech() {}  
 @Override public void onRmsChanged(float rmsdB) {}  
 @Override public void onBufferReceived(byte[] buffer) {}  
 @Override public void onEndOfSpeech() {}  
 @Override public void onError(int error) {  
 Toast.*makeText*(getApplicationContext(), "Speech recognition error!", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override public void onPartialResults(Bundle partialResults) {}  
 @Override public void onEvent(int eventType, Bundle params) {}  
 });  
  
 btnListen.setOnClickListener(v -> speakWord());  
 btnSpeak.setOnClickListener(v -> startSpeechRecognition());  
 btnNext.setOnClickListener(v -> setNewWord());  
 btnRetry.setOnClickListener(v -> startSpeechRecognition());  
 btnHint.setOnClickListener(v -> showHint());  
 btnChangeLevel.setOnClickListener(v -> changeDifficulty());  
  
 disableRetry();  
 }  
  
 private void requestAudioPermission() {  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*RECORD\_AUDIO*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*RECORD\_AUDIO*}, 1);  
 }  
 }  
  
 private void setNewWord() {  
 Random random = new Random();  
 correctWord = phrases[difficulty][random.nextInt(phrases[difficulty].length)];  
 wordToSpell.setText(correctWord);  
 recognitionResult.setText("");  
 hintView.setText("");  
 startTimer();  
 disableRetry();  
 }  
  
 private void speakWord() {  
 if (textToSpeech != null) {  
 textToSpeech.speak(correctWord, TextToSpeech.*QUEUE\_FLUSH*, null, null);  
 }  
 }  
  
 private void startSpeechRecognition() {  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, Locale.*FRENCH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_PROMPT*, "Pronounce the word...");  
 try {  
 speechRecognizer.startListening(intent);  
 } catch (Exception e) {  
 Toast.*makeText*(getApplicationContext(), "Speech recognition not supported!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void updateScore() {  
 scoreView.setText("Score: " + score + " / " + attempts);  
 }  
  
 private void updateHighScore() {  
 if (score > highScore) {  
 highScore = score;  
 highScoreView.setText("High Score: " + highScore);  
 }  
 }  
  
 private void updateHistory(String wordResult) {  
 history.append(wordResult).append("\n");  
 wordHistory.setText(history.toString());  
 }  
  
 private void enableRetry() {  
 btnRetry.setEnabled(true);  
 }  
  
 private void disableRetry() {  
 btnRetry.setEnabled(false);  
 }  
  
 private void showHint() {  
 hintView.setText("Hint: " + correctWord.substring(0, correctWord.length() / 2) + "...");  
 }  
  
 private void startTimer() {  
 if (timer != null) {  
 timer.cancel();  
 }  
 timer = new CountDownTimer(10000, 1000) {  
 @Override  
 public void onTick(long millisUntilFinished) {  
 timerView.setText("Time left: " + millisUntilFinished / 1000 + "s");  
 }  
  
 @Override  
 public void onFinish() {  
 timerView.setText("⏳ Time up! Try again.");  
 enableRetry();  
 }  
 }.start();  
 }  
  
 private void changeDifficulty() {  
 difficulty = (difficulty + 1) % 3;  
 setNewWord();  
 Toast.*makeText*(this, "Difficulty changed!", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 protected void onDestroy() {  
 if (textToSpeech != null) {  
 textToSpeech.stop();  
 textToSpeech.shutdown();  
 }  
 if (speechRecognizer != null) {  
 speechRecognizer.destroy();  
 }  
 super.onDestroy();  
 }  
}

GermanActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class  
  
  
  
GermanActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_german*);  
  
 Button btnStartLearning = findViewById(R.id.*btn\_start\_learning*);  
  
 // Navigate to German Quiz Activity  
 btnStartLearning.setOnClickListener(v -> {  
 Intent intent = new Intent(GermanActivity.this, LearningOptionsActivity\_German.class);  
 startActivity(intent);  
 });  
 }  
}

GermanCrosswordActivity.java

package com.example.myapplicationapp;  
  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class GermanCrosswordActivity extends AppCompatActivity {  
  
 private GermanCrosswordGridAdapter adapter;  
 private SharedPreferences prefs;  
 private static final String *PREFS\_NAME* = "GermanCrosswordPrefs";  
  
 private String[][] crosswordGrid = {  
 {"H", "", "S", ""},  
 {"A", "U", ""},  
 {"S", "", "N"},  
 {"H", "", "D", "", "E"},  
 {"K", "", "Z", ""},  
 {"F", "", "S"},  
 };  
  
 private String[] words = {"HAUS", "SEE", "SONNE", "HUND", "KATZE", "FLUSS"};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*german\_crossword*);  
  
 GridView gridView = findViewById(R.id.*crosswordGrid*);  
 Button submitButton = findViewById(R.id.*btn\_submit*);  
 Button saveButton = findViewById(R.id.*btn\_save*);  
  
 prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 adapter = new GermanCrosswordGridAdapter(this, crosswordGrid, prefs);  
 gridView.setAdapter(adapter);  
  
 submitButton.setOnClickListener(view -> checkAnswers());  
 saveButton.setOnClickListener(view -> saveProgress());  
 }  
  
 private void checkAnswers() {  
 if (adapter.isCrosswordCorrect(words)) {  
 Toast.*makeText*(this, "Gut gemacht! Du hast das Kreuzworträtsel gelöst!", Toast.*LENGTH\_LONG*).show();  
 } else {  
 Toast.*makeText*(this, "Versuche es noch einmal. Einige Wörter sind falsch.", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 private void saveProgress() {  
 adapter.saveUserProgress();  
 Toast.*makeText*(this, "Fortschritt gespeichert!", Toast.*LENGTH\_SHORT*).show();  
 }  
}

GermanCrosswordGridAdapter.java

package com.example.myapplicationapp;  
  
import android.content.Context;  
import android.content.SharedPreferences;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.EditText;  
import java.util.HashSet;  
  
public class GermanCrosswordGridAdapter extends BaseAdapter {  
 private Context context;  
 private String[][] grid;  
 private EditText[][] inputCells;  
 private SharedPreferences prefs;  
  
 public GermanCrosswordGridAdapter(Context context, String[][] grid, SharedPreferences prefs) {  
 this.context = context;  
 this.grid = grid;  
 this.prefs = prefs;  
 this.inputCells = new EditText[grid.length][grid[0].length];  
 }  
  
 @Override  
 public int getCount() {  
 return grid.length \* grid[0].length;  
 }  
  
 @Override  
 public Object getItem(int position) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
 return grid[row][col];  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
  
 // Prevent index out of bounds error  
 if (row >= grid.length || col >= grid[row].length) {  
 return new View(context);  
 }  
  
 EditText cell;  
 if (convertView == null) {  
 cell = new EditText(context);  
 } else {  
 cell = (EditText) convertView;  
 }  
  
 if (!grid[row][col].equals("")) {  
 cell.setText(grid[row][col]);  
 cell.setEnabled(false);  
 } else {  
 String savedValue = prefs.getString(row + "\_" + col, "");  
 cell.setText(savedValue);  
 cell.setHint("\_");  
 }  
  
 return cell;  
 }  
  
 public boolean isCrosswordCorrect(String[] correctWords) {  
 HashSet<String> enteredWords = new HashSet<>();  
 StringBuilder wordBuilder = new StringBuilder();  
  
 for (int i = 0; i < grid.length; i++) {  
 wordBuilder.setLength(0);  
 for (int j = 0; j < grid[i].length; j++) {  
 String letter = inputCells[i][j].getText().toString().toUpperCase();  
 if (!letter.isEmpty()) {  
 wordBuilder.append(letter);  
 }  
 }  
 if (wordBuilder.length() > 1) {  
 enteredWords.add(wordBuilder.toString());  
 }  
 }  
  
 for (String word : correctWords) {  
 if (!enteredWords.contains(word)) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
 public void saveUserProgress() {  
 SharedPreferences.Editor editor = prefs.edit();  
 for (int i = 0; i < grid.length; i++) {  
 for (int j = 0; j < grid[i].length; j++) {  
 editor.putString(i + "\_" + j, inputCells[i][j].getText().toString());  
 }  
 }  
 editor.apply();  
 }  
}

GermanPictureRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Collections;  
import java.util.List;  
  
public class GermanPictureRecognitionActivity extends AppCompatActivity {  
  
 private ImageView imageView;  
 private RadioGroup optionsGroup;  
 private Button nextButton, submitButton;  
  
 private final Question[] questions = {  
 new Question(R.drawable.*brezel*, "Welche dieser Speisen gehört zur deutschen Küche?",  
 new String[]{"Brezel","Beethoven","Berlin","Brandenburger Tor"}),  
  
 new Question(R.drawable.*beethoven*, "Welcher berühmte deutsche Komponist schrieb die 9. Sinfonie?",  
 new String[]{"Beethoven","Brezel","Berlin","Brandenburger Tor"}),  
  
 new Question(R.drawable.*berlin*, "Welche Stadt ist die Hauptstadt Deutschlands?",  
 new String[]{"Berlin","Beethoven","Brandenburger Tor","Brezel"}),  
  
 new Question(R.drawable.*brandenburger\_tor*, "Welches Wahrzeichen steht in Berlin?",  
 new String[]{"Brandenburger Tor","Berlin","Beethoven","Brezel"}),  
  
 new Question(R.drawable.*berliner\_mauer*, "Wann fiel die Berliner Mauer?",  
 new String[]{"1989", "1991", "1993","1990"})  
 };  
  
  
 private int currentQuestionIndex = 0;  
 private boolean answered = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_german\_picture\_recognition*);  
  
 imageView = findViewById(R.id.*imageView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 nextButton = findViewById(R.id.*btnNext*);  
 submitButton = findViewById(R.id.*btnSubmit*);  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 nextButton.setOnClickListener(v -> loadNextQuestion());  
 }  
  
 private void loadQuestion() {  
 if (currentQuestionIndex >= questions.length) {  
 Toast.*makeText*(this, "Quiz beendet!", Toast.*LENGTH\_LONG*).show();  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(false);  
 return;  
 }  
  
 Question question = questions[currentQuestionIndex];  
  
 imageView.setImageResource(question.imageResId);  
  
 List<String> shuffledOptions = new ArrayList<>(Arrays.*asList*(question.options));  
 Collections.*shuffle*(shuffledOptions);  
  
 optionsGroup.removeAllViews();  
  
 for (String option : shuffledOptions) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 radioButton.setTextSize(18);  
 radioButton.setPadding(10, 10, 10, 10);  
 optionsGroup.addView(radioButton);  
 }  
  
 optionsGroup.clearCheck();  
 answered = false;  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(true);  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId != -1) {  
 RadioButton selectedButton = findViewById(selectedId);  
 String selectedAnswer = selectedButton.getText().toString();  
 String correctAnswer = questions[currentQuestionIndex].correctAnswer;  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 Toast.*makeText*(this, "Richtig!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(this, "Falsch! Die richtige Antwort ist: " + correctAnswer, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 answered = true;  
 nextButton.setEnabled(true);  
 submitButton.setEnabled(false);  
 } else {  
 Toast.*makeText*(this, "Bitte eine Antwort auswählen!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void loadNextQuestion() {  
 if (answered) {  
 currentQuestionIndex++;  
 loadQuestion();  
 } else {  
 Toast.*makeText*(this, "Bitte zuerst die Antwort einreichen!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private static class Question {  
 int imageResId;  
 String correctAnswer;  
 String[] options;  
  
 Question(int imageResId, String correctAnswer, String[] options) {  
 this.imageResId = imageResId;  
 this.correctAnswer = correctAnswer;  
 this.options = options;  
 }  
 }  
}

GermanQuizActivity.java

package com.example.myapplicationapp;  
  
import android.content.res.AssetFileDescriptor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import org.tensorflow.lite.Interpreter;  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.nio.MappedByteBuffer;  
import java.nio.channels.FileChannel;  
import java.util.Arrays;  
import java.util.List;  
  
public class GermanQuizActivity extends AppCompatActivity {  
 private Interpreter tflite;  
 private TextView questionTextView;  
 private RadioGroup optionsGroup;  
 private Button submitButton;  
 private String[] questions = {  
 "What is the capital of Germany?",  
 "How do you say 'apple' in German?"  
 };  
 private String[][] options = {  
 {"London", "Berlin", "Madrid", "Rom"},  
 {"Apfel", "Orange", "Banane", "Traube"}  
 };  
 private int questionIndex = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_german\_quiz*);  
  
 questionTextView = findViewById(R.id.*questionTextView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 submitButton = findViewById(R.id.*submitButton*);  
  
 try {  
 tflite = new Interpreter(loadModelFile());  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 }  
  
 private MappedByteBuffer loadModelFile() throws IOException {  
 AssetFileDescriptor fileDescriptor = getAssets().openFd("model\_german.tflite");  
 FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());  
 FileChannel fileChannel = inputStream.getChannel();  
 return fileChannel.map(FileChannel.MapMode.*READ\_ONLY*, fileDescriptor.getStartOffset(), fileDescriptor.getDeclaredLength());  
 }  
  
 private void loadQuestion() {  
 if (questionIndex < questions.length) {  
 questionTextView.setText(questions[questionIndex]);  
 optionsGroup.removeAllViews();  
 for (String option : options[questionIndex]) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 optionsGroup.addView(radioButton);  
 }  
 } else {  
 questionTextView.setText("Quiz finished!");  
 submitButton.setEnabled(false);  
 }  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId == -1) return;  
  
 int predictedIndex = predictAnswer(questions[questionIndex]);  
 int correctIndex = Arrays.*asList*(options[questionIndex]).indexOf(options[questionIndex][predictedIndex]);  
  
 RadioButton selectedRadioButton = findViewById(selectedId);  
 String selectedAnswer = selectedRadioButton.getText().toString();  
 String correctAnswer = options[questionIndex][correctIndex];  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 questionTextView.setText("Correct!");  
 } else {  
 questionTextView.setText("Incorrect. Try again.");  
 }  
  
 questionIndex++;  
 loadQuestion();  
 }  
  
 private int predictAnswer(String question) {  
 float[][] input = new float[1][20]; // Assume padded sequence size  
 float[][] output = new float[1][4]; // 4 answer choices  
  
 tflite.run(input, output);  
 return getMaxIndex(output[0]);  
 }  
  
 private int getMaxIndex(float[] arr) {  
 int maxIndex = 0;  
 for (int i = 1; i < arr.length; i++) {  
 if (arr[i] > arr[maxIndex]) maxIndex = i;  
 }  
 return maxIndex;  
 }  
}

GermanSpeechRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.os.Bundle;  
import android.speech.tts.TextToSpeech;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.Locale;  
import java.util.Random;  
  
public class GermanSpeechRecognitionActivity extends AppCompatActivity {  
  
 private TextToSpeech textToSpeech;  
 private TextView txtWord, txtFeedback, txtInstructions, txtScore, txtDifficulty, txtAttempts;  
 private Button btnSpeak, btnNewPhrase, btnNext, btnRepeat, btnIncreaseDifficulty, btnDecreaseDifficulty, btnReset;  
 private String[] germanPhrases = {  
 "Hallo", "Guten Morgen", "Wie geht es Ihnen?", "Ich liebe Deutsch", "Sprechen Sie Deutsch?",  
 "Wo ist der Bahnhof?", "Könnten Sie das bitte wiederholen?", "Ich verstehe nicht", "Ich möchte etwas essen", "Danke schön"  
 };  
 private String currentPhrase;  
 private int score = 0;  
 private int difficultyLevel = 1;  
 private int attempts = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_german\_speech\_recognition*);  
  
 txtWord = findViewById(R.id.*txtWord*);  
 txtFeedback = findViewById(R.id.*txtFeedback*);  
 txtInstructions = findViewById(R.id.*txtInstructions*);  
 txtScore = findViewById(R.id.*txtScore*);  
 txtDifficulty = findViewById(R.id.*txtDifficulty*);  
 txtAttempts = findViewById(R.id.*txtAttempts*);  
 btnSpeak = findViewById(R.id.*btnSpeak*);  
 btnNewPhrase = findViewById(R.id.*btnNewPhrase*);  
 btnNext = findViewById(R.id.*btnNext*);  
 btnRepeat = findViewById(R.id.*btnRepeat*);  
 btnIncreaseDifficulty = findViewById(R.id.*btnIncreaseDifficulty*);  
 btnDecreaseDifficulty = findViewById(R.id.*btnDecreaseDifficulty*);  
 btnReset = findViewById(R.id.*btnReset*);  
  
 txtInstructions.setText("Tap 'Listen to Pronunciation' to hear the phrase and try to repeat it!");  
 updateUI();  
 generateNewPhrase();  
  
 textToSpeech = new TextToSpeech(this, status -> {  
 if (status == TextToSpeech.*SUCCESS*) {  
 textToSpeech.setLanguage(Locale.*GERMAN*);  
 textToSpeech.setSpeechRate(0.9f);  
 textToSpeech.setPitch(1.0f);  
 }  
 });  
  
 btnSpeak.setOnClickListener(v -> speakPhrase());  
 btnNewPhrase.setOnClickListener(v -> generateNewPhrase());  
 btnNext.setOnClickListener(v -> generateNewPhrase());  
 btnRepeat.setOnClickListener(v -> repeatPhrase());  
 btnIncreaseDifficulty.setOnClickListener(v -> changeDifficulty(1));  
 btnDecreaseDifficulty.setOnClickListener(v -> changeDifficulty(-1));  
 btnReset.setOnClickListener(v -> resetGame());  
 }  
  
 private void generateNewPhrase() {  
 Random random = new Random();  
 currentPhrase = germanPhrases[random.nextInt(germanPhrases.length)];  
 txtWord.setText(currentPhrase);  
 txtFeedback.setText("");  
 attempts = 0;  
 updateUI();  
 }  
  
 private void speakPhrase() {  
 if (textToSpeech != null) {  
 textToSpeech.speak(currentPhrase, TextToSpeech.*QUEUE\_FLUSH*, null, null);  
 }  
 }  
  
 private void repeatPhrase() {  
 attempts++;  
 txtFeedback.setText("Try pronouncing: " + currentPhrase);  
 updateUI();  
 }  
  
 private void updateUI() {  
 txtScore.setText("Score: " + score);  
 txtDifficulty.setText("Difficulty: " + difficultyLevel);  
 txtAttempts.setText("Attempts: " + attempts);  
 }  
  
 private void changeDifficulty(int change) {  
 difficultyLevel = Math.*max*(1, Math.*min*(5, difficultyLevel + change));  
 updateUI();  
 Toast.*makeText*(this, "Difficulty set to " + difficultyLevel, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 private void resetGame() {  
 score = 0;  
 attempts = 0;  
 difficultyLevel = 1;  
 updateUI();  
 generateNewPhrase();  
 Toast.*makeText*(this, "Game Reset!", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 private void checkPronunciation(String userSpeech) {  
 if (userSpeech.equalsIgnoreCase(currentPhrase)) {  
 score++;  
 txtFeedback.setText("✅ Correct! You said: " + userSpeech);  
 txtFeedback.setTextColor(getResources().getColor(android.R.color.*holo\_green\_dark*));  
 } else {  
 txtFeedback.setText("❌ Incorrect. You said: " + userSpeech + "\nCorrect phrase: " + currentPhrase);  
 txtFeedback.setTextColor(getResources().getColor(android.R.color.*holo\_red\_dark*));  
 }  
 updateUI();  
 }  
  
 @Override  
 protected void onDestroy() {  
 if (textToSpeech != null) {  
 textToSpeech.stop();  
 textToSpeech.shutdown();  
 }  
 super.onDestroy();  
 }  
}

HelpActivity.java

package com.example.myapplicationapp;  
  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class HelpActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_help*); // Ensure you create activity\_help.xml  
 }  
}

HomeActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.LinearLayout;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class HomeActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_home*);  
  
 LinearLayout cardFrench = findViewById(R.id.*cardFrench*);  
 LinearLayout cardSpanish = findViewById(R.id.*cardSpanish*);  
 LinearLayout cardJapanese = findViewById(R.id.*cardJapanese*);  
 LinearLayout cardGerman = findViewById(R.id.*cardGerman*);  
  
 Button btnProfile = findViewById(R.id.*btnProfile*);  
 Button btnHelp = findViewById(R.id.*btnHelp*);  
 Button btnMore = findViewById(R.id.*btnMore*);  
  
 cardFrench.setOnClickListener(view -> navigateToActivity(FrenchActivity.class, "French selected"));  
 cardSpanish.setOnClickListener(view -> navigateToActivity(SpanishActivity.class, "Spanish selected"));  
 cardJapanese.setOnClickListener(view -> navigateToActivity(JapaneseActivity.class, "Japanese selected"));  
 cardGerman.setOnClickListener(view -> navigateToActivity(GermanActivity.class, "German selected"));  
  
 btnProfile.setOnClickListener(view -> navigateToActivity(ProfileActivity.class, "Navigating to Profile"));  
 btnHelp.setOnClickListener(view -> navigateToActivity(HelpActivity.class, "Navigating to Help"));  
 btnMore.setOnClickListener(view -> navigateToActivity(MoreActivity.class, "Navigating to More"));  
 }  
  
  
  
 private void navigateToActivity(Class<?> targetActivity, String message) {  
 Toast.*makeText*(this, message, Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(this, targetActivity));  
 }  
}

Interpreter.java

package com.example.myapplicationapp;  
  
public class Interpreter {  
 public void run(float[][] input, float[][] output) {  
 }  
  
 public void close() {  
  
 }  
}

JapaneseActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class JapaneseActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_japanese*);  
  
 Button btnStartLearning = findViewById(R.id.*btn\_start\_learning*);  
  
 // Navigate to French Quiz Activity  
 btnStartLearning.setOnClickListener(v -> {  
 Intent intent = new Intent(JapaneseActivity.this, LearningOptionsActivity\_Japanese.class);  
 startActivity(intent);  
 });  
 }  
}

JapaneseCrosswordActivity.java

package com.example.myapplicationapp;  
  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class JapaneseCrosswordActivity extends AppCompatActivity {  
  
 private JapaneseCrosswordGridAdapter adapter;  
 private SharedPreferences prefs;  
 private static final String *PREFS\_NAME* = "JapaneseCrosswordPrefs";  
  
 // Updated crossword grid with more five-letter Japanese words  
 private String[][] crosswordGrid = {  
 {"さ", "く", "", "ん", ""},  
 {"", "ま", "", "り",},  
 {"た", "", "の", "",},  
 {"み", "", "ん", "", "い"},  
 {"す", "", "か", "", "ろ"}  
 };  
  
 private String[] words = {"さくらんぼ", "ひまわり", "たけのこ", "みかんせい", "すいかめろ"};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*japanese\_crossword*);  
  
 GridView gridView = findViewById(R.id.*crosswordGrid*);  
 Button submitButton = findViewById(R.id.*btn\_submit*);  
 Button saveButton = findViewById(R.id.*btn\_save*);  
  
 prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 adapter = new JapaneseCrosswordGridAdapter(this, crosswordGrid, prefs);  
 gridView.setAdapter(adapter);  
  
 submitButton.setOnClickListener(view -> checkAnswers());  
 saveButton.setOnClickListener(view -> saveProgress());  
 }  
  
 private void checkAnswers() {  
 if (adapter.isCrosswordCorrect(words)) {  
 Toast.*makeText*(this, "素晴らしい！クロスワードを完成しました！", Toast.*LENGTH\_LONG*).show();  
 } else {  
 Toast.*makeText*(this, "もう一度試してください。間違った単語があります。", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 private void saveProgress() {  
 adapter.saveUserProgress();  
 Toast.*makeText*(this, "進捗が保存されました!", Toast.*LENGTH\_SHORT*).show();  
 }  
}

JapaneseCrosswordGridAdapter.java

package com.example.myapplicationapp;  
  
import android.content.Context;  
import android.content.SharedPreferences;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.EditText;  
import java.util.HashSet;  
  
public class JapaneseCrosswordGridAdapter extends BaseAdapter {  
 private Context context;  
 private String[][] grid;  
 private EditText[][] inputCells;  
 private SharedPreferences prefs;  
  
 public JapaneseCrosswordGridAdapter(Context context, String[][] grid, SharedPreferences prefs) {  
 this.context = context;  
 this.grid = grid;  
 this.prefs = prefs;  
 this.inputCells = new EditText[grid.length][grid[0].length];  
 }  
  
 public JapaneseCrosswordGridAdapter() {  
  
 }  
  
 @Override  
 public int getCount() {  
 return grid.length \* grid[0].length;  
 }  
  
 @Override  
 public Object getItem(int position) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
 return grid[row][col];  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
  
 // Prevent index out of bounds error  
 if (row >= grid.length || col >= grid[row].length) {  
 return new View(context);  
 }  
  
 EditText cell;  
 if (convertView == null) {  
 cell = new EditText(context);  
 } else {  
 cell = (EditText) convertView;  
 }  
  
 if (!grid[row][col].equals("")) {  
 cell.setText(grid[row][col]);  
 cell.setEnabled(false);  
 } else {  
 String savedValue = prefs.getString(row + "\_" + col, "");  
 cell.setText(savedValue);  
 cell.setHint("\_");  
 }  
  
 return cell;  
 }  
  
 public boolean isCrosswordCorrect(String[] correctWords) {  
 HashSet<String> enteredWords = new HashSet<>();  
 StringBuilder wordBuilder = new StringBuilder();  
  
 for (int i = 0; i < grid.length; i++) {  
 wordBuilder.setLength(0);  
 for (int j = 0; j < grid[i].length; j++) {  
 String letter = inputCells[i][j].getText().toString().toUpperCase();  
 if (!letter.isEmpty()) {  
 wordBuilder.append(letter);  
 }  
 }  
 if (wordBuilder.length() > 1) {  
 enteredWords.add(wordBuilder.toString());  
 }  
 }  
  
 for (String word : correctWords) {  
 if (!enteredWords.contains(word)) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
 public void saveUserProgress() {  
 SharedPreferences.Editor editor = prefs.edit();  
 for (int i = 0; i < grid.length; i++) {  
 for (int j = 0; j < grid[i].length; j++) {  
 editor.putString(i + "\_" + j, inputCells[i][j].getText().toString());  
 }  
 }  
 editor.apply();  
 }  
}

JapanesePictureRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Collections;  
import java.util.List;  
  
public class JapanesePictureRecognitionActivity extends AppCompatActivity {  
  
 private ImageView imageView;  
 private RadioGroup optionsGroup;  
 private Button nextButton, submitButton;  
  
 private final Question[] questions = {  
 new Question(R.drawable.*sushi*, "寿司", new String[]{"寿司", "ラーメン", "天ぷら", "うどん", "焼肉"}), // Cuisine  
 new Question(R.drawable.*kabuki*, "歌舞伎", new String[]{"歌舞伎", "能", "狂言", "茶道", "華道"}), // Traditional Japanese Performing Art  
 new Question(R.drawable.*kyoto*, "京都", new String[]{"京都", "大阪", "東京", "奈良", "福岡"}), // City  
 new Question(R.drawable.*samurai*, "侍", new String[]{"侍", "忍者", "武士", "僧侶", "剣士"}), // History  
 new Question(R.drawable.*jupiter*, "木星", new String[]{"木星", "地球", "火星", "金星", "土星"}) // Science  
 };  
  
  
 private int currentQuestionIndex = 0;  
 private boolean answered = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_japanese\_picture\_recognition*);  
  
 imageView = findViewById(R.id.*imageView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 nextButton = findViewById(R.id.*btnNext*);  
 submitButton = findViewById(R.id.*btnSubmit*);  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 nextButton.setOnClickListener(v -> loadNextQuestion());  
 }  
  
 private void loadQuestion() {  
 if (currentQuestionIndex >= questions.length) {  
 Toast.*makeText*(this, "クイズ終了!", Toast.*LENGTH\_LONG*).show();  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(false);  
 return;  
 }  
  
 Question question = questions[currentQuestionIndex];  
  
 imageView.setImageResource(question.imageResId);  
  
 List<String> shuffledOptions = new ArrayList<>(Arrays.*asList*(question.options));  
 Collections.*shuffle*(shuffledOptions);  
  
 optionsGroup.removeAllViews();  
  
 for (String option : shuffledOptions) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 radioButton.setTextSize(18);  
 radioButton.setPadding(10, 10, 10, 10);  
 optionsGroup.addView(radioButton);  
 }  
  
 optionsGroup.clearCheck();  
 answered = false;  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(true);  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId != -1) {  
 RadioButton selectedButton = findViewById(selectedId);  
 String selectedAnswer = selectedButton.getText().toString();  
 String correctAnswer = questions[currentQuestionIndex].correctAnswer;  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 Toast.*makeText*(this, "正解!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(this, "不正解! 正しい答えは: " + correctAnswer, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 answered = true;  
 nextButton.setEnabled(true);  
 submitButton.setEnabled(false);  
 } else {  
 Toast.*makeText*(this, "回答を選択してください!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void loadNextQuestion() {  
 if (answered) {  
 currentQuestionIndex++;  
 loadQuestion();  
 } else {  
 Toast.*makeText*(this, "まず回答を提出してください!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private static class Question {  
 int imageResId;  
 String correctAnswer;  
 String[] options;  
  
 Question(int imageResId, String correctAnswer, String[] options) {  
 this.imageResId = imageResId;  
 this.correctAnswer = correctAnswer;  
 this.options = options;  
 }  
 }  
}

JapaneseQuizActivity.java

package com.example.myapplicationapp;  
  
import android.content.res.AssetFileDescriptor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import org.tensorflow.lite.Interpreter;  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.nio.MappedByteBuffer;  
import java.nio.channels.FileChannel;  
import java.util.Arrays;  
import java.util.List;  
  
public class JapaneseQuizActivity extends AppCompatActivity {  
 private Interpreter tflite;  
 private TextView questionTextView;  
 private RadioGroup optionsGroup;  
 private Button submitButton;  
 private String[] questions = {  
 "What is the capital of Japan?",  
 "How do you say 'apple' in Japanese?"  
 };  
 private String[][] options = {  
 {"ロンドン", "東京", "マドリード", "ローマ"},  
 {"リンゴ", "オレンジ", "バナナ", "ブドウ"}  
 };  
 private int questionIndex = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_french\_quiz*);  
  
 questionTextView = findViewById(R.id.*questionTextView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 submitButton = findViewById(R.id.*submitButton*);  
  
 try {  
 tflite = new Interpreter(loadModelFile());  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 }  
  
 private MappedByteBuffer loadModelFile() throws IOException {  
 AssetFileDescriptor fileDescriptor = getAssets().openFd("model\_Japanese.tflite");  
 FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());  
 FileChannel fileChannel = inputStream.getChannel();  
 return fileChannel.map(FileChannel.MapMode.*READ\_ONLY*, fileDescriptor.getStartOffset(), fileDescriptor.getDeclaredLength());  
 }  
  
 private void loadQuestion() {  
 if (questionIndex < questions.length) {  
 questionTextView.setText(questions[questionIndex]);  
 optionsGroup.removeAllViews();  
 for (String option : options[questionIndex]) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 optionsGroup.addView(radioButton);  
 }  
 } else {  
 questionTextView.setText("Quiz finished!");  
 submitButton.setEnabled(false);  
 }  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId == -1) return;  
  
 int predictedIndex = predictAnswer(questions[questionIndex]);  
 int correctIndex = Arrays.*asList*(options[questionIndex]).indexOf(options[questionIndex][predictedIndex]);  
  
 RadioButton selectedRadioButton = findViewById(selectedId);  
 String selectedAnswer = selectedRadioButton.getText().toString();  
 String correctAnswer = options[questionIndex][correctIndex];  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 questionTextView.setText("Correct!");  
 } else {  
 questionTextView.setText("Incorrect. Try again.");  
 }  
  
 questionIndex++;  
 loadQuestion();  
 }  
  
 private int predictAnswer(String question) {  
 float[][] input = new float[1][20]; // Assume padded sequence size  
 float[][] output = new float[1][4]; // 4 answer choices  
  
 tflite.run(input, output);  
 return getMaxIndex(output[0]);  
 }  
  
 private int getMaxIndex(float[] arr) {  
 int maxIndex = 0;  
 for (int i = 1; i < arr.length; i++) {  
 if (arr[i] > arr[maxIndex]) maxIndex = i;  
 }  
 return maxIndex;  
 }  
}

JapaneseSpeechRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.tts.TextToSpeech;  
import android.speech.RecognizerIntent;  
import android.speech.SpeechRecognizer;  
import android.speech.RecognitionListener;  
import android.media.MediaPlayer;  
import android.os.Vibrator;  
import android.content.Context;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import java.util.Locale;  
import java.util.Random;  
  
public class JapaneseSpeechRecognitionActivity extends AppCompatActivity {  
 private TextToSpeech textToSpeech;  
 private SpeechRecognizer speechRecognizer;  
 private TextView textViewWord, textViewResult, textViewFeedback, textViewScore;  
 private Button buttonStart, buttonListen, buttonNewWord, buttonAddCustom;  
 private EditText editTextCustomWord;  
 private MediaPlayer successSound;  
 private Vibrator vibrator;  
 private int score = 0;  
  
 // Japanese words & sentences  
 private String[] japaneseWords = {  
 "こんにちは", "ありがとう", "さようなら", "おはようございます", "おやすみなさい", "すみません"  
 };  
  
 private String[] japaneseSentences = {  
 "私は日本語を勉強しています", "これはペンです", "あなたの名前は何ですか", "日本の文化が好きです", "明日は晴れますか？"  
 };  
  
 private String currentWord;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_japanese\_speech\_recognition*);  
  
 textViewWord = findViewById(R.id.*textViewWord*);  
 textViewResult = findViewById(R.id.*textViewResult*);  
 textViewFeedback = findViewById(R.id.*textViewFeedback*);  
 textViewScore = findViewById(R.id.*textViewScore*);  
 buttonStart = findViewById(R.id.*buttonStart*);  
 buttonListen = findViewById(R.id.*buttonListen*);  
 buttonNewWord = findViewById(R.id.*buttonNewWord*);  
 buttonAddCustom = findViewById(R.id.*buttonAddCustom*);  
 editTextCustomWord = findViewById(R.id.*editTextCustomWord*);  
  
 // Initialize TTS  
 textToSpeech = new TextToSpeech(this, status -> {  
 if (status == TextToSpeech.*SUCCESS*) {  
 textToSpeech.setLanguage(Locale.*JAPANESE*);  
 }  
 });  
  
 // Initialize Sound & Vibration  
 vibrator = (Vibrator) getSystemService(Context.*VIBRATOR\_SERVICE*);  
  
 try {  
 successSound = MediaPlayer.*create*(this, R.raw.*success\_sound*);  
 } catch (Exception e) {  
 successSound = null; // If the sound file is missing, prevent crash  
 }  
  
 // Initialize Speech Recognizer  
 speechRecognizer = SpeechRecognizer.*createSpeechRecognizer*(this);  
 speechRecognizer.setRecognitionListener(new RecognitionListener() {  
 @Override  
 public void onResults(Bundle results) {  
 ArrayList<String> matches = results.getStringArrayList(SpeechRecognizer.*RESULTS\_RECOGNITION*);  
 if (matches != null && !matches.isEmpty()) {  
 String recognizedText = matches.get(0);  
 textViewResult.setText("You said: " + recognizedText);  
 checkPronunciation(recognizedText, currentWord);  
 }  
 }  
  
 @Override public void onError(int error) {  
 textViewFeedback.setText("❌ Recognition error. Try again!");  
 }  
 @Override public void onReadyForSpeech(Bundle params) {}  
 @Override public void onBeginningOfSpeech() {}  
 @Override public void onRmsChanged(float rmsdB) {}  
 @Override public void onBufferReceived(byte[] buffer) {}  
 @Override public void onEndOfSpeech() {}  
 @Override public void onPartialResults(Bundle partialResults) {}  
 @Override public void onEvent(int eventType, Bundle params) {}  
 });  
  
 // Set a random word initially  
 setRandomWord();  
  
 // Listen to pronunciation  
 buttonListen.setOnClickListener(v -> textToSpeech.speak(currentWord, TextToSpeech.*QUEUE\_FLUSH*, null, null));  
  
 // Start Speech Recognition  
 buttonStart.setOnClickListener(v -> {  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, "ja-JP");  
 intent.putExtra(RecognizerIntent.*EXTRA\_PROMPT*, "Say: " + currentWord);  
 speechRecognizer.startListening(intent);  
 });  
  
 // Get new word  
 buttonNewWord.setOnClickListener(v -> setRandomWord());  
  
 // Add custom word  
 buttonAddCustom.setOnClickListener(v -> {  
 String customWord = editTextCustomWord.getText().toString().trim();  
 if (!customWord.isEmpty()) {  
 currentWord = customWord;  
 textViewWord.setText(currentWord);  
 textViewResult.setText("Your Pronunciation:");  
 textViewFeedback.setText("Try pronouncing the word!");  
 editTextCustomWord.setText("");  
 }  
 });  
 }  
  
 // Selects a random word or sentence  
 private void setRandomWord() {  
 Random random = new Random();  
 if (random.nextBoolean()) {  
 currentWord = japaneseWords[random.nextInt(japaneseWords.length)];  
 } else {  
 currentWord = japaneseSentences[random.nextInt(japaneseSentences.length)];  
 }  
 textViewWord.setText(currentWord);  
 textViewResult.setText("Your Pronunciation:");  
 textViewFeedback.setText("Try pronouncing the word!");  
 }  
  
 // Compare pronunciation accuracy  
 private void checkPronunciation(String recognized, String correct) {  
 int distance = levenshteinDistance(recognized, correct);  
 if (distance == 0) {  
 textViewFeedback.setText("✅ Perfect pronunciation! 🎉");  
 score++;  
 if (successSound != null) successSound.start();  
 } else if (distance <= 2) {  
 textViewFeedback.setText("🟡 Almost correct! Keep practicing. 😊");  
 } else {  
 textViewFeedback.setText("❌ Try again! 😅");  
 if (vibrator != null) vibrator.vibrate(300); // Vibrate on incorrect answer  
 }  
 textViewScore.setText("Score: " + score);  
 }  
  
 // Levenshtein Distance Algorithm  
 private int levenshteinDistance(String s1, String s2) {  
 int[][] dp = new int[s1.length() + 1][s2.length() + 1];  
 for (int i = 0; i <= s1.length(); i++) {  
 for (int j = 0; j <= s2.length(); j++) {  
 if (i == 0) dp[i][j] = j;  
 else if (j == 0) dp[i][j] = i;  
 else {  
 int cost = (s1.charAt(i - 1) == s2.charAt(j - 1)) ? 0 : 1;  
 dp[i][j] = Math.*min*(Math.*min*(dp[i - 1][j] + 1, dp[i][j - 1] + 1), dp[i - 1][j - 1] + cost);  
 }  
 }  
 }  
 return dp[s1.length()][s2.length()];  
 }  
  
 @Override  
 protected void onDestroy() {  
 if (textToSpeech != null) {  
 textToSpeech.stop();  
 textToSpeech.shutdown();  
 }  
 if (speechRecognizer != null) {  
 speechRecognizer.destroy();  
 }  
 if (successSound != null) {  
 successSound.release();  
 }  
 super.onDestroy();  
 }  
}

LearningOptionsActivity\_French.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class LearningOptionsActivity\_French extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_learning\_options\_french*);  
  
 Button btnQuiz = findViewById(R.id.*btn\_quiz*);  
 Button btnCrossword = findViewById(R.id.*btn\_crossword*);  
 Button btnPictureRecognition = findViewById(R.id.*btn\_picture\_recognition*);  
 Button btnSpeechRecognition = findViewById(R.id.*btn\_speech\_recognition*);  
  
 btnQuiz.setOnClickListener(v -> startActivity(new Intent(this, FrenchQuizActivity.class)));  
 btnCrossword.setOnClickListener(v -> startActivity(new Intent(this, FrenchCrosswordActivity.class)));  
 btnPictureRecognition.setOnClickListener(v -> startActivity(new Intent(this, FrenchPictureRecognitionActivity.class)));  
 btnSpeechRecognition.setOnClickListener(v -> startActivity(new Intent(this, FrenchSpeechRecognitionActivity.class)));  
 }  
}

LearningOptionsActivity\_German.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class LearningOptionsActivity\_German extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_learning\_options\_german*);  
  
 Button btnQuiz = findViewById(R.id.*btn\_quiz*);  
 Button btnCrossword = findViewById(R.id.*btn\_crossword*);  
 Button btnPictureRecognition = findViewById(R.id.*btn\_picture\_recognition*);  
 Button btnSpeechRecognition = findViewById(R.id.*btn\_speech\_recognition*);  
  
 btnQuiz.setOnClickListener(v -> startActivity(new Intent(this, GermanQuizActivity.class)));  
 btnCrossword.setOnClickListener(v -> startActivity(new Intent(this, GermanCrosswordActivity.class)));  
 btnPictureRecognition.setOnClickListener(v -> startActivity(new Intent(this, GermanPictureRecognitionActivity.class)));  
 btnSpeechRecognition.setOnClickListener(v -> startActivity(new Intent(this, GermanSpeechRecognitionActivity.class)));  
 }  
}

LearningOptionsActivity\_Japanese.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class LearningOptionsActivity\_Japanese extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_learning\_options\_japanese*);  
  
 Button btnQuiz = findViewById(R.id.*btn\_quiz*);  
 Button btnCrossword = findViewById(R.id.*btn\_crossword*);  
 Button btnPictureRecognition = findViewById(R.id.*btn\_picture\_recognition*);  
 Button btnSpeechRecognition = findViewById(R.id.*btn\_speech\_recognition*);  
  
 btnQuiz.setOnClickListener(v -> startActivity(new Intent(this, JapaneseQuizActivity.class)));  
 btnCrossword.setOnClickListener(v -> startActivity(new Intent(this, JapaneseCrosswordActivity.class)));  
 btnPictureRecognition.setOnClickListener(v -> startActivity(new Intent(this, JapanesePictureRecognitionActivity.class)));  
 btnSpeechRecognition.setOnClickListener(v -> startActivity(new Intent(this, JapaneseSpeechRecognitionActivity.class)));  
 }  
}

LearningOptionsActivity\_Spanish.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class LearningOptionsActivity\_Spanish extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_learning\_options\_spanish*);  
  
 Button btnQuiz = findViewById(R.id.*btn\_quiz*);  
 Button btnCrossword = findViewById(R.id.*btn\_crossword*);  
 Button btnPictureRecognition = findViewById(R.id.*btn\_picture\_recognition*);  
 Button btnSpeechRecognition = findViewById(R.id.*btn\_speech\_recognition*);  
  
 btnQuiz.setOnClickListener(v -> startActivity(new Intent(this, SpanishQuizActivity.class)));  
 btnCrossword.setOnClickListener(v -> startActivity(new Intent(this, SpanishCrosswordActivity.class)));  
 btnPictureRecognition.setOnClickListener(v -> startActivity(new Intent(this, SpanishPictureRecognitionActivity.class)));  
 btnSpeechRecognition.setOnClickListener(v -> startActivity(new Intent(this, SpanishSpeechRecognitionActivity.class)));  
 }  
}

LoginActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class LoginActivity extends AppCompatActivity {  
 private EditText emailLogin, passwordLogin;  
 private Button loginButton;  
 private TextView signupRedirect;  
 private FirebaseAuth auth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 emailLogin = findViewById(R.id.*emailLogin*);  
 passwordLogin = findViewById(R.id.*passwordLogin*);  
 loginButton = findViewById(R.id.*loginButton*);  
 signupRedirect = findViewById(R.id.*signupRedirect*);  
 auth = FirebaseAuth.*getInstance*();  
  
 loginButton.setOnClickListener(v -> {  
 String email = emailLogin.getText().toString().trim();  
 String password = passwordLogin.getText().toString().trim();  
  
 if (!email.isEmpty() && !password.isEmpty()) {  
 auth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(task -> {  
 if (task.isSuccessful()) {  
 Toast.*makeText*(LoginActivity.this, "Login Successful!", Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(LoginActivity.this, HomeActivity.class));  
 } else {  
 Toast.*makeText*(LoginActivity.this, "Login Failed: " + task.getException().getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 } else {  
 Toast.*makeText*(LoginActivity.this, "All fields are required", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 signupRedirect.setOnClickListener(v -> {  
 startActivity(new Intent(LoginActivity.this, SignupActivity.class));  
 finish();  
 });  
 }  
}

MainActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Find buttons by their IDs  
 Button loginButton = findViewById(R.id.*loginActivityButton*);  
 Button signupButton = findViewById(R.id.*signupActivityButton*);  
  
 // Set click listeners  
 loginButton.setOnClickListener(v -> {  
 // Navigate to LoginActivity  
 Intent intent = new Intent(MainActivity.this, LoginActivity.class);  
 startActivity(intent);  
 });  
  
 signupButton.setOnClickListener(v -> {  
 // Navigate to SignupActivity  
 Intent intent = new Intent(MainActivity.this, SignupActivity.class);  
 startActivity(intent);  
 });  
 }  
}

MoreActivity.java

package com.example.myapplicationapp;  
  
import android.os.Bundle;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MoreActivity extends AppCompatActivity {  
  
 TextView infoText;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_more*);  
  
 // Initialize TextViews  
 TextView frenchText = findViewById(R.id.*frenchText*);  
 TextView germanText = findViewById(R.id.*germanText*);  
 TextView spanishText = findViewById(R.id.*spanishText*);  
 TextView japaneseText = findViewById(R.id.*japaneseText*);  
 infoText = findViewById(R.id.*infoText*);  
  
 // Set Click Listeners  
 frenchText.setOnClickListener(view -> showLanguageInfo("French"));  
 germanText.setOnClickListener(view -> showLanguageInfo("German"));  
 spanishText.setOnClickListener(view -> showLanguageInfo("Spanish"));  
 japaneseText.setOnClickListener(view -> showLanguageInfo("Japanese"));  
 }  
  
 private void showLanguageInfo(String language) {  
 String info;  
 switch (language) {  
 case "French":  
 info = "• Spoken by over 300 million people worldwide, French is an official language in 29 countries.\n" +  
 "• Originates from Latin and has influences from Germanic and English languages.\n" +  
 "• Used in diplomacy, the UN, EU, and the Olympic Games.\n" +  
 "• Uses Latin script with accents such as é, è, à, ô, and ç.";  
 break;  
 case "German":  
 info = "• Spoken by over 130 million people, German is the most widely spoken native language in Europe.\n" +  
 "• Official language of Germany, Austria, Switzerland, and parts of Italy and Belgium.\n" +  
 "• Known for long compound words like 'Donaudampfschifffahrtsgesellschaft'.\n" +  
 "• Uses Latin script with umlauts (ä, ö, ü) and Eszett (ß).";  
 break;  
 case "Spanish":  
 info = "• Spanish is spoken by over 500 million people and is the official language in 21 countries.\n" +  
 "• Evolved from Latin with strong Arabic influences during the Moorish rule in Spain.\n" +  
 "• Known for phonetic spelling and simple pronunciation.\n" +  
 "• Uses Latin script with accents such as á, é, í, ó, ú, and ñ.";  
 break;  
 case "Japanese":  
 info = "• Spoken by over 125 million people, primarily in Japan.\n" +  
 "• Uses three writing systems: Hiragana, Katakana, and Kanji.\n" +  
 "• Has a complex honorific system reflecting social hierarchy.\n" +  
 "• Borrowed many words from Chinese and English.";  
 break;  
 default:  
 info = "Language information not available.";  
 }  
 infoText.setText(info);  
 }  
}

ProfileActivity.java

package com.example.myapplicationapp;  
  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
import android.widget.Button;

simport android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private ImageView userAvatar;  
 private EditText editUserName, editUserEmail;  
 private ProgressBar progressGerman, progressFrench, progressSpanish, progressJapanese;  
 private Button btnEditProfile, btnChooseAvatar;  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 // Initialize Views  
 userAvatar = findViewById(R.id.*user\_avatar*);  
 editUserName = findViewById(R.id.*edit\_user\_name*);  
 editUserEmail = findViewById(R.id.*edit\_user\_email*);  
 progressGerman = findViewById(R.id.*progress\_german*);  
 progressFrench = findViewById(R.id.*progress\_french*);  
 progressSpanish = findViewById(R.id.*progress\_spanish*);  
 progressJapanese = findViewById(R.id.*progress\_japanese*);  
 btnEditProfile = findViewById(R.id.*btn\_edit\_profile*);  
 btnChooseAvatar = findViewById(R.id.*btn\_choose\_avatar*);  
  
 // Set Initial User Data (Example: Replace with dynamic data fetched from a database or API)  
 editUserName.setText("John Doe");  
 editUserEmail.setText("johndoe@example.com");  
 progressGerman.setProgress(60);  
 progressFrench.setProgress(40);  
 progressSpanish.setProgress(80);  
 progressJapanese.setProgress(20);  
  
 // Button to Choose Avatar  
 btnChooseAvatar.setOnClickListener(v -> {  
 // Logic to open an avatar picker or gallery  
 Toast.*makeText*(ProfileActivity.this, "Choose Avatar Clicked", Toast.*LENGTH\_SHORT*).show();  
 });  
  
 // Save Profile Changes  
 btnEditProfile.setOnClickListener(v -> {  
 String userName = editUserName.getText().toString();  
 String userEmail = editUserEmail.getText().toString();  
  
 if (userName.isEmpty() || userEmail.isEmpty()) {  
 Toast.*makeText*(ProfileActivity.this, "Please fill in all fields", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 // Save the data (to local storage, database, or an API)  
 Toast.*makeText*(ProfileActivity.this, "Profile Updated Successfully", Toast.*LENGTH\_SHORT*).show();  
  
 // Update UI with the new data (if necessary)  
 // Example: Display updated name and email  
 editUserName.setText(userName);  
 editUserEmail.setText(userEmail);  
 }  
 });  
 }  
}

QuizQuestion.java

package com.example.myapplicationapp;  
  
public class QuizQuestion {  
 public QuizQuestion(String question, String option, String option1, String option2, String option3, String correctAnswer) {  
 }  
  
 public int getOption1() {  
 return 0;  
 }  
  
 public int getOption2() {  
 return 0;  
 }  
  
 public int getQuestion() {  
 return 0;  
 }  
  
 public int getOption3() {  
 return 0;  
 }  
  
 public int getOption4() {  
 return 0;  
 }  
}

SignupActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class SignupActivity extends AppCompatActivity {  
 private EditText emailSignup, passwordSignup;  
 private Button signupButton;  
 private TextView loginRedirect;  
 private FirebaseAuth auth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_signup*);  
  
 emailSignup = findViewById(R.id.*emailSignup*);  
 passwordSignup = findViewById(R.id.*passwordSignup*);  
 signupButton = findViewById(R.id.*signupButton*);  
 loginRedirect = findViewById(R.id.*loginRedirect*);  
 auth = FirebaseAuth.*getInstance*();  
  
 signupButton.setOnClickListener(v -> {  
 String email = emailSignup.getText().toString().trim();  
 String password = passwordSignup.getText().toString().trim();  
  
 if (!email.isEmpty() && !password.isEmpty()) {  
 auth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(task -> {  
 if (task.isSuccessful()) {  
 Toast.*makeText*(SignupActivity.this, "Signup Successful!", Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(SignupActivity.this, LoginActivity.class));  
 } else {  
 Toast.*makeText*(SignupActivity.this, "Signup Failed: " + task.getException().getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 } else {  
 Toast.*makeText*(SignupActivity.this, "All fields are required", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 loginRedirect.setOnClickListener(v -> {  
 startActivity(new Intent(SignupActivity.this, LoginActivity.class));  
 finish();  
 });  
 }  
}

SpanishActivity.java

package com.example.myapplicationapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class SpanishActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_german*);  
  
 Button btnStartLearning = findViewById(R.id.*btn\_start\_learning*);  
  
 // Navigate to German Quiz Activity  
 btnStartLearning.setOnClickListener(v -> {  
 Intent intent = new Intent(SpanishActivity.this, LearningOptionsActivity\_Spanish.class);  
 startActivity(intent);  
 });  
 }  
}

SpanishCrosswordActivity.java

package com.example.myapplicationapp;  
  
import android.annotation.SuppressLint;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class SpanishCrosswordActivity extends AppCompatActivity {  
  
 private SpanishCrosswordGridAdapter adapter;  
 private SharedPreferences prefs;  
 private static final String *PREFS\_NAME* = "SpanishCrosswordPrefs";  
  
 private String[][] crosswordGrid = {  
 {"C", "", "S", ""},  
 {"M", "A", "", ""},  
 {"S", "", "L", ""},  
 {"P", "", "R", "", "O"},  
 {"G", "", "T", ""},  
 {"R", "", "O", ""}  
 };  
  
 private String[] words = {"CASA", "MAR", "SOL", "PERRO", "GATO", "RÍO"};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*spanish\_crossword*);  
  
 GridView gridView = findViewById(R.id.*crosswordGrid*);  
 Button submitButton = findViewById(R.id.*btn\_submit*);  
 Button saveButton = findViewById(R.id.*btn\_save*);  
  
 prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 adapter = new SpanishCrosswordGridAdapter(this, crosswordGrid, prefs);  
 gridView.setAdapter(adapter);  
  
 submitButton.setOnClickListener(view -> checkAnswers());  
 saveButton.setOnClickListener(view -> saveProgress());  
 }  
  
 private void checkAnswers() {  
 if (adapter.isCrosswordCorrect(words)) {  
 Toast.*makeText*(this, "¡Bravo! Has completado el crucigrama!", Toast.*LENGTH\_LONG*).show();  
 } else {  
 Toast.*makeText*(this, "Inténtalo de nuevo. Algunas palabras son incorrectas.", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 private void saveProgress() {  
 adapter.saveUserProgress();  
 Toast.*makeText*(this, "Progreso guardado!", Toast.*LENGTH\_SHORT*).show();  
 }  
}

SpanishCrosswordGridAdapter.java

package com.example.myapplicationapp;  
  
import android.content.Context;  
import android.content.SharedPreferences;  
import android.text.InputFilter;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.EditText;  
import java.util.HashSet;  
  
public class SpanishCrosswordGridAdapter extends BaseAdapter {  
 private Context context;  
 private String[][] grid;  
 private EditText[][] inputCells;  
 private SharedPreferences prefs;  
  
 public SpanishCrosswordGridAdapter(Context context, String[][] grid, SharedPreferences prefs) {  
 this.context = context;  
 this.grid = grid;  
 this.prefs = prefs;  
 this.inputCells = new EditText[grid.length][grid[0].length];  
 }  
  
 @Override  
 public int getCount() {  
 return grid.length \* grid[0].length;  
 }  
  
 @Override  
 public Object getItem(int position) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
 return grid[row][col];  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 int row = position / grid[0].length;  
 int col = position % grid[0].length;  
  
 EditText cell = new EditText(context);  
 cell.setFilters(new InputFilter[]{new InputFilter.LengthFilter(1)}); // Restrict to 1 letter  
  
 if (!grid[row][col].equals("")) {  
 cell.setText(grid[row][col]);  
 cell.setEnabled(false);  
 } else {  
 String savedValue = prefs.getString(row + "\_" + col, "");  
 cell.setText(savedValue);  
 cell.setHint("\_");  
 }  
  
 inputCells[row][col] = cell;  
 return cell;  
 }  
  
 public boolean isCrosswordCorrect(String[] correctWords) {  
 HashSet<String> enteredWords = new HashSet<>();  
  
 // Check words in rows  
 for (int i = 0; i < grid.length; i++) {  
 StringBuilder wordBuilder = new StringBuilder();  
 for (int j = 0; j < grid[i].length; j++) {  
 if (inputCells[i][j] != null) {  
 String letter = inputCells[i][j].getText().toString().toUpperCase();  
 if (!letter.isEmpty()) {  
 wordBuilder.append(letter);  
 }  
 }  
 }  
 if (wordBuilder.length() > 1) {  
 enteredWords.add(wordBuilder.toString());  
 }  
 }  
  
 // Check words in columns  
 for (int j = 0; j < grid[0].length; j++) {  
 StringBuilder wordBuilder = new StringBuilder();  
 for (int i = 0; i < grid.length; i++) {  
 if (inputCells[i][j] != null) {  
 String letter = inputCells[i][j].getText().toString().toUpperCase();  
 if (!letter.isEmpty()) {  
 wordBuilder.append(letter);  
 }  
 }  
 }  
 if (wordBuilder.length() > 1) {  
 enteredWords.add(wordBuilder.toString());  
 }  
 }  
  
 for (String word : correctWords) {  
 if (!enteredWords.contains(word)) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
 public void saveUserProgress() {  
 SharedPreferences.Editor editor = prefs.edit();  
 for (int i = 0; i < grid.length; i++) {  
 for (int j = 0; j < grid[i].length; j++) {  
 if (inputCells[i][j] != null) {  
 editor.putString(i + "\_" + j, inputCells[i][j].getText().toString());  
 }  
 }  
 }  
 editor.apply();  
 }  
}

SpanishPictureRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Collections;  
import java.util.List;  
  
public class SpanishPictureRecognitionActivity extends AppCompatActivity {  
  
 private ImageView imageView;  
 private RadioGroup optionsGroup;  
 private Button nextButton, submitButton;  
  
 private final Question[] questions = {  
 new Question(R.drawable.*paella*, "Paella", new String[]{"Paella", "Tacos", "Sushi", "Pizza", "Arepa"}), // Cuisine  
 new Question(R.drawable.*guernica*, "Guernica", new String[]{"Las Meninas", "El Grito", "Guernica", "La Gioconda", "El Jardín de las Delicias"}), // Art  
 new Question(R.drawable.*madrid*, "Madrid", new String[]{"Barcelona", "Madrid", "Sevilla", "Valencia", "Bilbao"}), // Cities  
 new Question(R.drawable.*independencia*, "1810", new String[]{"1492", "1810", "1914", "1789", "1605"}), // History (Spanish Independence)  
 new Question(R.drawable.*alhambra*, "Alhambra", new String[]{"Sagrada Familia", "Alhambra", "Mezquita de Córdoba", "Prado", "Acueducto de Segovia"}) // Landmarks  
 };  
  
  
 private int currentQuestionIndex = 0;  
 private boolean answered = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_spanish\_picture\_recognition*);  
  
 imageView = findViewById(R.id.*imageView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 nextButton = findViewById(R.id.*btnNext*);  
 submitButton = findViewById(R.id.*btnSubmit*);  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 nextButton.setOnClickListener(v -> loadNextQuestion());  
 }  
  
 private void loadQuestion() {  
 if (currentQuestionIndex >= questions.length) {  
 Toast.*makeText*(this, "¡Quiz terminado!", Toast.*LENGTH\_LONG*).show();  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(false);  
 return;  
 }  
  
 Question question = questions[currentQuestionIndex];  
  
 imageView.setImageResource(question.imageResId);  
  
 List<String> shuffledOptions = new ArrayList<>(Arrays.*asList*(question.options));  
 Collections.*shuffle*(shuffledOptions);  
  
 optionsGroup.removeAllViews();  
  
 for (String option : shuffledOptions) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 radioButton.setTextSize(18);  
 radioButton.setPadding(10, 10, 10, 10);  
 optionsGroup.addView(radioButton);  
 }  
  
 optionsGroup.clearCheck();  
 answered = false;  
 nextButton.setEnabled(false);  
 submitButton.setEnabled(true);  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId != -1) {  
 RadioButton selectedButton = findViewById(selectedId);  
 String selectedAnswer = selectedButton.getText().toString();  
 String correctAnswer = questions[currentQuestionIndex].correctAnswer;  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 Toast.*makeText*(this, "¡Correcto!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(this, "¡Incorrecto! La respuesta correcta es: " + correctAnswer, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 answered = true;  
 nextButton.setEnabled(true);  
 submitButton.setEnabled(false);  
 } else {  
 Toast.*makeText*(this, "¡Seleccione una respuesta!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void loadNextQuestion() {  
 if (answered) {  
 currentQuestionIndex++;  
 loadQuestion();  
 } else {  
 Toast.*makeText*(this, "¡Envíe su respuesta primero!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private static class Question {  
 int imageResId;  
 String correctAnswer;  
 String[] options;  
  
 Question(int imageResId, String correctAnswer, String[] options) {  
 this.imageResId = imageResId;  
 this.correctAnswer = correctAnswer;  
 this.options = options;  
 }  
 }  
}

SpanishQuizActivity.java

package com.example.myapplicationapp;  
  
import android.content.res.AssetFileDescriptor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import org.tensorflow.lite.Interpreter;  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.nio.MappedByteBuffer;  
import java.nio.channels.FileChannel;  
import java.util.Arrays;  
import java.util.List;  
  
public class SpanishQuizActivity extends AppCompatActivity {  
 private Interpreter tflite;  
 private TextView questionTextView;  
 private RadioGroup optionsGroup;  
 private Button submitButton;  
 private String[] questions = {  
 "What is the capital of Spain?",  
 "How do you say 'apple' in Spanish?"  
 };  
 private String[][] options = {  
 {"Londres", "Madrid", "París", "Roma"},  
 {"Manzana", "Naranja", "Plátano", "Uva"}  
 };  
 private int questionIndex = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_spanish\_quiz*);  
  
 questionTextView = findViewById(R.id.*questionTextView*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
 submitButton = findViewById(R.id.*submitButton*);  
  
 try {  
 tflite = new Interpreter(loadModelFile());  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 loadQuestion();  
  
 submitButton.setOnClickListener(v -> checkAnswer());  
 }  
  
 private MappedByteBuffer loadModelFile() throws IOException {  
 AssetFileDescriptor fileDescriptor = getAssets().openFd("model\_spanish.tflite");  
 FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());  
 FileChannel fileChannel = inputStream.getChannel();  
 return fileChannel.map(FileChannel.MapMode.*READ\_ONLY*, fileDescriptor.getStartOffset(), fileDescriptor.getDeclaredLength());  
 }  
  
 private void loadQuestion() {  
 if (questionIndex < questions.length) {  
 questionTextView.setText(questions[questionIndex]);  
 optionsGroup.removeAllViews();  
 for (String option : options[questionIndex]) {  
 RadioButton radioButton = new RadioButton(this);  
 radioButton.setText(option);  
 optionsGroup.addView(radioButton);  
 }  
 } else {  
 questionTextView.setText("Quiz finished!");  
 submitButton.setEnabled(false);  
 }  
 }  
  
 private void checkAnswer() {  
 int selectedId = optionsGroup.getCheckedRadioButtonId();  
 if (selectedId == -1) return;  
  
 int predictedIndex = predictAnswer(questions[questionIndex]);  
 int correctIndex = Arrays.*asList*(options[questionIndex]).indexOf(options[questionIndex][predictedIndex]);  
  
 RadioButton selectedRadioButton = findViewById(selectedId);  
 String selectedAnswer = selectedRadioButton.getText().toString();  
 String correctAnswer = options[questionIndex][correctIndex];  
  
 if (selectedAnswer.equals(correctAnswer)) {  
 questionTextView.setText("Correct!");  
 } else {  
 questionTextView.setText("Incorrect. Try again.");  
 }  
  
 questionIndex++;  
 loadQuestion();  
 }  
  
 private int predictAnswer(String question) {  
 float[][] input = new float[1][20]; // Assume padded sequence size  
 float[][] output = new float[1][4]; // 4 answer choices  
  
 tflite.run(input, output);  
 return getMaxIndex(output[0]);  
 }  
  
 private int getMaxIndex(float[] arr) {  
 int maxIndex = 0;  
 for (int i = 1; i < arr.length; i++) {  
 if (arr[i] > arr[maxIndex]) maxIndex = i;  
 }  
 return maxIndex;  
 }  
}

SpanishSpeechRecognitionActivity.java

package com.example.myapplicationapp;  
  
import android.content.ActivityNotFoundException;  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.RecognizerIntent;  
import android.speech.tts.TextToSpeech;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.\*;  
  
public class SpanishSpeechRecognitionActivity extends AppCompatActivity {  
  
 private TextToSpeech tts;  
 private TextView txtWord, txtFeedback, txtScore, txtDifficulty;  
 private Button btnListen, btnSpeak, btnNext, btnChangeDifficulty;  
 private int score = 0;  
 private String targetPhrase;  
 private String difficultyLevel = "Easy"; // Default level  
 private Random random = new Random();  
  
 private Map<String, List<String>> difficultyWords = new HashMap<String, List<String>>() {{  
 put("Easy", Arrays.*asList*("Hola", "Adiós", "Gracias", "Perro", "Gato"));  
 put("Medium", Arrays.*asList*("¿Cómo estás?", "Me gusta el español", "Voy a la tienda"));  
 put("Hard", Arrays.*asList*("El clima es muy bueno hoy", "Necesito ayuda con mi tarea", "Mañana voy a viajar a España"));  
 }};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_spanish\_speech\_recognition*);  
  
 txtWord = findViewById(R.id.*txt\_word*);  
 txtFeedback = findViewById(R.id.*txt\_feedback*);  
 txtScore = findViewById(R.id.*txt\_score*);  
 txtDifficulty = findViewById(R.id.*txt\_difficulty*);  
 btnListen = findViewById(R.id.*btn\_listen*);  
 btnSpeak = findViewById(R.id.*btn\_speak*);  
 btnNext = findViewById(R.id.*btn\_next*);  
 btnChangeDifficulty = findViewById(R.id.*btn\_change\_difficulty*);  
  
 // Initialize TTS  
 tts = new TextToSpeech(this, status -> {  
 if (status == TextToSpeech.*SUCCESS*) {  
 tts.setLanguage(new Locale("es", "ES")); // Spanish  
 }  
 });  
  
 btnListen.setOnClickListener(v -> speakWord());  
 btnSpeak.setOnClickListener(v -> startSpeechRecognition());  
 btnNext.setOnClickListener(v -> selectNewPhrase());  
 btnChangeDifficulty.setOnClickListener(v -> changeDifficulty());  
  
 selectNewPhrase();  
 }  
  
 private void selectNewPhrase() {  
 List<String> words = difficultyWords.get(difficultyLevel);  
 targetPhrase = words.get(random.nextInt(words.size()));  
 txtWord.setText(targetPhrase);  
 txtFeedback.setText("Try pronouncing the phrase.");  
 }  
  
 private void speakWord() {  
 tts.speak(targetPhrase, TextToSpeech.*QUEUE\_FLUSH*, null, null);  
 }  
  
 private void startSpeechRecognition() {  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, "es-ES");  
 intent.putExtra(RecognizerIntent.*EXTRA\_PROMPT*, "Pronounce: " + targetPhrase);  
  
 try {  
 startActivityForResult(intent, 100);  
 } catch (ActivityNotFoundException e) {  
 Toast.*makeText*(this, "Speech recognition not supported", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == 100 && resultCode == *RESULT\_OK*) {  
 ArrayList<String> results = data.getStringArrayListExtra(RecognizerIntent.*EXTRA\_RESULTS*);  
 if (results != null && !results.isEmpty()) {  
 String userSpeech = results.get(0).toLowerCase();  
 int similarity = calculateSimilarity(targetPhrase.toLowerCase(), userSpeech);  
 if (similarity >= 80) { // Threshold for correct pronunciation  
 txtFeedback.setText("✅ Perfect pronunciation! Similarity: " + similarity + "%");  
 score++;  
 } else {  
 txtFeedback.setText("❌ Try again! You said: " + userSpeech + " (Similarity: " + similarity + "%)");  
 }  
 txtScore.setText("Score: " + score);  
 }  
 }  
 }  
  
 private int calculateSimilarity(String original, String spoken) {  
 int maxLength = Math.*max*(original.length(), spoken.length());  
 int minLength = Math.*min*(original.length(), spoken.length());  
 int matches = 0;  
  
 for (int i = 0; i < minLength; i++) {  
 if (original.charAt(i) == spoken.charAt(i)) {  
 matches++;  
 }  
 }  
  
 return (int) (((double) matches / maxLength) \* 100);  
 }  
  
 private void changeDifficulty() {  
 if (difficultyLevel.equals("Easy")) {  
 difficultyLevel = "Medium";  
 } else if (difficultyLevel.equals("Medium")) {  
 difficultyLevel = "Hard";  
 } else {  
 difficultyLevel = "Easy";  
 }  
 txtDifficulty.setText("Difficulty: " + difficultyLevel);  
 selectNewPhrase();  
 }  
  
 @Override  
 protected void onDestroy() {  
 if (tts != null) {  
 tts.stop();  
 tts.shutdown();  
 }  
 super.onDestroy();  
 }  
}

SpeechServiceListener.java

package com.example.myapplicationapp;  
  
import org.vosk.android.RecognitionListener;  
  
public class SpeechServiceListener implements RecognitionListener {  
 @Override  
 public void onPartialResult(String hypothesis) {  
  
 }  
  
 @Override  
 public void onResult(String hypothesis) {  
  
 }  
  
 @Override  
 public void onFinalResult(String hypothesis) {  
  
 }  
  
 @Override  
 public void onError(Exception exception) {  
  
 }  
  
 @Override  
 public void onTimeout() {  
  
 }  
}

activity\_french.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <Button  
 android:id="@+id/btn\_start\_learning"  
 android:layout\_width="328dp"  
 android:layout\_height="62dp"  
 android:text="Start Learning"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_french\_picture\_recongnition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:scaleType="fitCenter"  
 android:contentDescription="Image to recognize" />  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnSubmit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Soumettre"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnNext"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Suivant"  
 android:layout\_marginTop="8dp"  
 android:enabled="false" />  
  
</LinearLayout>

activity\_french\_quiz.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/questionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Question"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp"/>  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:id="@+id/submitButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Soumettre"  
 android:layout\_marginTop="16dp"  
 android:layout\_gravity="center\_horizontal"/>  
</LinearLayout>

activity\_french\_speech\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="20dp"  
 android:gravity="center"  
 android:background="@color/white">  
  
 <!-- Timer Display -->  
 <TextView  
 android:id="@+id/timer\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="⏳ Time left: 10s"  
 android:textSize="18sp"  
 android:textColor="@color/black"  
 android:gravity="center"  
 android:padding="5dp"  
 android:textStyle="bold" />  
  
 <!-- Word to Spell -->  
 <TextView  
 android:id="@+id/word\_to\_spell"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Bonjour"  
 android:textSize="28sp"  
 android:textColor="@color/black"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:padding="10dp" />  
  
 <!-- Hint Display -->  
 <TextView  
 android:id="@+id/hint\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hint: Bo..."  
 android:textSize="16sp"  
 android:textColor="@color/gray"  
 android:gravity="center"  
 android:padding="5dp"  
 android:visibility="gone"/>  
  
 <!-- Listen & Speak Buttons -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:padding="10dp">  
  
 <Button  
 android:id="@+id/btn\_listen"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="🔊 Listen"  
 android:backgroundTint="@color/blue"  
 android:textColor="@color/white"  
 android:padding="10dp"/>  
  
 <Button  
 android:id="@+id/btn\_speak"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="🎤 Speak"  
 android:backgroundTint="@color/green"  
 android:textColor="@color/white"  
 android:padding="10dp"  
 android:layout\_marginStart="10dp"/>  
 </LinearLayout>  
  
 <!-- Recognition Result -->  
 <TextView  
 android:id="@+id/recognition\_result"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="You said: ..."  
 android:textSize="20sp"  
 android:textColor="@color/black"  
 android:gravity="center"  
 android:padding="10dp"/>  
  
 <!-- Score & High Score -->  
 <TextView  
 android:id="@+id/score\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Score: 0 / 0"  
 android:textSize="18sp"  
 android:textColor="@color/black"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:padding="5dp"/>  
  
 <TextView  
 android:id="@+id/high\_score\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="High Score: 0"  
 android:textSize="18sp"  
 android:textColor="@color/red"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:padding="5dp"/>  
  
 <!-- Word History -->  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="150dp"  
 android:padding="5dp">  
  
 <TextView  
 android:id="@+id/word\_history"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="History: \n"  
 android:textSize="16sp"  
 android:textColor="@color/black"/>  
 </ScrollView>  
  
 <!-- Control Buttons -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:padding="10dp">  
  
 <Button  
 android:id="@+id/btn\_hint"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="💡 Hint"  
 android:backgroundTint="@color/yellow"  
 android:textColor="@color/black"  
 android:padding="10dp"/>  
  
 <Button  
 android:id="@+id/btn\_retry"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="🔄 Retry"  
 android:backgroundTint="@color/orange"  
 android:textColor="@color/white"  
 android:padding="10dp"  
 android:layout\_marginStart="10dp"  
 android:enabled="false"/>  
  
 <Button  
 android:id="@+id/btn\_next"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="➡ Next"  
 android:backgroundTint="@color/purple"  
 android:textColor="@color/white"  
 android:padding="10dp"  
 android:layout\_marginStart="10dp"/>  
 </LinearLayout>  
  
 <!-- Change Difficulty -->  
 <Button  
 android:id="@+id/btn\_change\_level"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="🎚 Change Difficulty"  
 android:backgroundTint="@color/gray"  
 android:textColor="@color/white"  
 android:padding="10dp"  
 android:layout\_marginTop="10dp"/>  
</LinearLayout>

activity\_german.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <Button  
 android:id="@+id/btn\_start\_learning"  
 android:layout\_width="328dp"  
 android:layout\_height="62dp"  
 android:text="Start Learning"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_german\_picture\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:scaleType="fitCenter"  
 android:contentDescription="Bild zum Erkennen" />  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnSubmit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Einreichen"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnNext"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Weiter"  
 android:layout\_marginTop="8dp"  
 android:enabled="false" />  
  
</LinearLayout>

activity\_german\_quiz.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/questionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Question"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp"/>  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:id="@+id/submitButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Soumettre"  
 android:layout\_marginTop="16dp"  
 android:layout\_gravity="center\_horizontal"/>  
</LinearLayout>

activity\_german\_speech\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center\_horizontal">  
  
 <TextView  
 android:id="@+id/txtInstructions"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Tap 'Listen to Pronunciation' to hear the phrase and try to repeat it!"  
 android:textSize="18sp"  
 android:textStyle="italic"  
 android:paddingBottom="16dp"/>  
  
 <TextView  
 android:id="@+id/txtWord"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="German Phrase"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="16dp"/>  
  
 <Button  
 android:id="@+id/btnSpeak"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Listen to Pronunciation"  
 android:layout\_marginTop="8dp"/>  
  
 <Button  
 android:id="@+id/btnNewPhrase"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="New Phrase"  
 android:layout\_marginTop="8dp"/>  
  
 <Button  
 android:id="@+id/btnNext"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Next"  
 android:layout\_marginTop="8dp"/>  
  
 <Button  
 android:id="@+id/btnRepeat"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Repeat"  
 android:layout\_marginTop="8dp"/>  
  
 <Button  
 android:id="@+id/btnIncreaseDifficulty"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Increase Difficulty"  
 android:layout\_marginTop="8dp"/>  
  
 <Button  
 android:id="@+id/btnDecreaseDifficulty"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Decrease Difficulty"  
 android:layout\_marginTop="8dp"/>  
  
 <Button  
 android:id="@+id/btnReset"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Reset"  
 android:layout\_marginTop="8dp"/>  
  
 <TextView  
 android:id="@+id/txtScore"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Score: 0"  
 android:textSize="18sp"  
 android:layout\_marginTop="16dp"  
 android:textStyle="bold"/>  
  
 <TextView  
 android:id="@+id/txtDifficulty"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Difficulty: 1"  
 android:textSize="18sp"  
 android:layout\_marginTop="8dp"  
 android:textStyle="bold"/>  
  
 <TextView  
 android:id="@+id/txtAttempts"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Attempts: 0"  
 android:textSize="18sp"  
 android:layout\_marginTop="8dp"  
 android:textStyle="bold"/>  
  
 <TextView  
 android:id="@+id/txtFeedback"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Feedback"  
 android:textSize="18sp"  
 android:textColor="@android:color/holo\_red\_dark"  
 android:layout\_marginTop="8dp"/>  
  
</LinearLayout>

activity\_help.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_home.xml

<ScrollView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#F5F5F5"  
 tools:context=".HomeActivity">  
  
 <!-- Parent Layout -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <!-- Main Content -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <!-- Header Section -->  
 <TextView  
 android:id="@+id/tvHeader"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Language Selection"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 android:textColor="#000000"  
 android:layout\_gravity="center"  
 android:layout\_marginBottom="16dp" />  
  
 <!-- French Card -->  
 <LinearLayout  
 android:id="@+id/cardFrench"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/card\_gradient\_blue"  
 android:padding="16dp"  
 android:layout\_marginBottom="16dp"  
 android:elevation="4dp"  
 android:gravity="center\_horizontal"  
 android:clickable="true"  
 android:focusable="true"  
 android:onClick="onFrenchCardClick"  
 tools:ignore="OnClick">  
  
 <ImageView  
 android:id="@+id/French"  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:src="@drawable/france"  
 android:contentDescription="French Icon" />  
  
 <TextView  
 android:id="@+id/tvFrench"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="French"  
 android:textColor="#000000"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:layout\_marginTop="8dp" />  
 </LinearLayout>  
  
 <!-- Spanish Card -->  
 <LinearLayout  
 android:id="@+id/cardSpanish"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/card\_gradient\_orange"  
 android:padding="16dp"  
 android:layout\_marginBottom="16dp"  
 android:elevation="4dp"  
 android:gravity="center\_horizontal"  
 android:clickable="true"  
 android:focusable="true"  
 android:onClick="onSpanishCardClick"  
 tools:ignore="OnClick">  
  
 <ImageView  
 android:id="@+id/spain"  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:src="@drawable/spain"  
 android:contentDescription="Spanish Icon" />  
  
 <TextView  
 android:id="@+id/tvSpanish"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Spanish"  
 android:textColor="#000000"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:layout\_marginTop="8dp" />  
 </LinearLayout>  
  
 <!-- Japanese Card -->  
 <LinearLayout  
 android:id="@+id/cardJapanese"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/card\_gradient\_purple"  
 android:padding="16dp"  
 android:layout\_marginBottom="16dp"  
 android:elevation="4dp"  
 android:gravity="center\_horizontal"  
 android:clickable="true"  
 android:focusable="true"  
 android:onClick="onJapaneseCardClick"  
 tools:ignore="OnClick">  
  
 <ImageView  
 android:id="@+id/japan"  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:src="@drawable/japan"  
 android:contentDescription="Japanese Icon" />  
  
 <TextView  
 android:id="@+id/tvJapanese"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Japanese"  
 android:textColor="#000000"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:layout\_marginTop="8dp" />  
 </LinearLayout>  
  
 <!-- German Card -->  
 <LinearLayout  
 android:id="@+id/cardGerman"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/card\_gradient\_red"  
 android:padding="16dp"  
 android:layout\_marginBottom="16dp"  
 android:elevation="4dp"  
 android:gravity="center\_horizontal"  
 android:clickable="true"  
 android:focusable="true"  
 android:onClick="onGermanCardClick"  
 tools:ignore="OnClick">  
  
 <ImageView  
 android:id="@+id/germany"  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:src="@drawable/germany"  
 android:contentDescription="German Icon" />  
  
 <TextView  
 android:id="@+id/tvGerman"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="German"  
 android:textColor="#000000"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:layout\_marginTop="8dp" />  
 </LinearLayout>  
 </LinearLayout>  
  
 <!-- Footer Section -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:padding="16dp"  
 android:background="#E0E0E0"  
 android:layout\_gravity="bottom">  
  
 <Button  
 android:id="@+id/btnProfile"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Profile"  
 android:backgroundTint="#6200EE"  
 android:textColor="#FFFFFF"  
 android:layout\_marginEnd="8dp" />  
  
 <Button  
 android:id="@+id/btnHelp"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Help"  
 android:backgroundTint="#03DAC5"  
 android:textColor="#FFFFFF"  
 android:layout\_marginEnd="8dp" />  
  
 <Button  
 android:id="@+id/btnMore"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="More"  
 android:backgroundTint="#FF5722"  
 android:textColor="#FFFFFF" />  
 </LinearLayout>  
 </LinearLayout>  
</ScrollView>

activity\_japanese.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <Button  
 android:id="@+id/btn\_start\_learning"  
 android:layout\_width="328dp"  
 android:layout\_height="62dp"  
 android:text="Start Learning"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_japanese\_picture\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:scaleType="fitCenter"  
 android:contentDescription="認識する画像" />  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnSubmit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="提出"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnNext"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="次へ"  
 android:layout\_marginTop="8dp"  
 android:enabled="false" />  
  
</LinearLayout>

activity\_japanese\_quiz.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/questionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="質問"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp"/>  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:id="@+id/submitButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="送信"  
 android:layout\_marginTop="16dp"  
 android:layout\_gravity="center\_horizontal"/>  
</LinearLayout>

activity\_japanese\_speech\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center"  
 android:background="#F5F5F5">  
  
 <!-- Word/Sentence Display -->  
 <TextView  
 android:id="@+id/textViewWord"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="26sp"  
 android:text="Word to Pronounce"  
 android:textStyle="bold"  
 android:textColor="#000000"  
 android:padding="10dp"/>  
  
 <!-- Listen Button (TTS) -->  
 <Button  
 android:id="@+id/buttonListen"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="🔊 Listen to Pronunciation"  
 android:backgroundTint="#FF9800"  
 android:textColor="#FFFFFF"  
 android:layout\_marginTop="10dp"/>  
  
 <!-- Start Speech Recognition -->  
 <Button  
 android:id="@+id/buttonStart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="🎙️ Start Speech Recognition"  
 android:backgroundTint="#4CAF50"  
 android:textColor="#FFFFFF"  
 android:layout\_marginTop="10dp"/>  
  
 <!-- Get New Random Word -->  
 <Button  
 android:id="@+id/buttonNewWord"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="🔄 Get New Word/Sentence"  
 android:backgroundTint="#2196F3"  
 android:textColor="#FFFFFF"  
 android:layout\_marginTop="10dp"/>  
  
 <!-- Display the Recognized Word -->  
 <TextView  
 android:id="@+id/textViewResult"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="22sp"  
 android:text="Your Pronunciation: "  
 android:textColor="#333333"  
 android:padding="10dp"  
 android:textStyle="bold"/>  
  
 <!-- Feedback on Pronunciation -->  
 <TextView  
 android:id="@+id/textViewFeedback"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="22sp"  
 android:text="Feedback"  
 android:textColor="#FF5722"  
 android:padding="10dp"  
 android:textStyle="bold"/>  
  
 <!-- Score Tracking -->  
 <TextView  
 android:id="@+id/textViewScore"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="22sp"  
 android:text="Score: 0"  
 android:textColor="#673AB7"  
 android:padding="10dp"  
 android:textStyle="bold"/>  
  
 <!-- Custom Word Input -->  
 <EditText  
 android:id="@+id/editTextCustomWord"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter your own Japanese word"  
 android:textSize="18sp"  
 android:padding="8dp"  
 android:background="#FFFFFF"  
 android:elevation="2dp"  
 android:layout\_marginTop="10dp"/>  
  
 <!-- Add Custom Word -->  
 <Button  
 android:id="@+id/buttonAddCustom"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="➕ Add Custom Word"  
 android:backgroundTint="#9C27B0"  
 android:textColor="#FFFFFF"  
 android:layout\_marginTop="10dp"/>  
  
</LinearLayout>

activity\_learning\_options\_french.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent">  
  
 <Button  
 android:id="@+id/btn\_quiz"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Quiz" />  
  
 <Button  
 android:id="@+id/btn\_crossword"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Crossword"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_picture\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Picture Recognition"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_speech\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Speech Recognition"  
 android:layout\_marginTop="16dp" />  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_learning\_options\_german.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent">  
  
 <Button  
 android:id="@+id/btn\_quiz"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Quiz" />  
  
 <Button  
 android:id="@+id/btn\_crossword"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Crossword"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_picture\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Picture Recognition"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_speech\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Speech Recognition"  
 android:layout\_marginTop="16dp" />  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_learning\_options\_japanese.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent">  
  
 <Button  
 android:id="@+id/btn\_quiz"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Quiz" />  
  
 <Button  
 android:id="@+id/btn\_crossword"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Crossword"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_picture\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Picture Recognition"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_speech\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Speech Recognition"  
 android:layout\_marginTop="16dp" />  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_learning\_options\_spanish.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent">  
  
 <Button  
 android:id="@+id/btn\_quiz"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Quiz" />  
  
 <Button  
 android:id="@+id/btn\_crossword"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Crossword"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_picture\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Picture Recognition"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btn\_speech\_recognition"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:text="Speech Recognition"  
 android:layout\_marginTop="16dp" />  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_login.xml

<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center">  
  
 <!-- Logo -->  
 <ImageView  
 android:id="@+id/logo"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:src="@drawable/logo"  
 android:contentDescription="App Logo"  
 android:layout\_marginBottom="16dp" />  
  
 <!-- Welcome Text -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Welcome to My App!"  
 android:textSize="24sp"  
 android:layout\_gravity="center"  
 android:paddingBottom="32dp"  
 android:textColor="@android:color/black" />  
  
 <EditText  
 android:id="@+id/emailLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Email"  
 android:inputType="textEmailAddress" />  
  
 <EditText  
 android:id="@+id/passwordLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Password"  
 android:inputType="textPassword" />  
  
 <Button  
 android:id="@+id/loginButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Login" />  
  
 <TextView  
 android:id="@+id/signupRedirect"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Don't have an account? Sign Up"  
 android:gravity="center"  
 android:paddingTop="16dp"  
 android:textColor="@android:color/holo\_blue\_dark" />  
</LinearLayout>

activity\_main.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center"  
 android:background="@drawable/gradient\_background"  
 tools:ignore="ExtraText">  
  
 <!-- Logo -->  
 <ImageView  
 android:id="@+id/logo"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:src="@drawable/logo"  
 android:contentDescription="App Logo"  
 android:layout\_marginBottom="24dp"  
 android:elevation="4dp" />  
  
 <!-- Welcome Text -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Welcome to My App"  
 android:textSize="26sp"  
 android:textStyle="bold"  
 android:textColor="#000000"  
 android:layout\_gravity="center"  
 android:paddingBottom="32dp" />  
  
 <!-- Login Button -->  
 <Button  
 android:id="@+id/loginActivityButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:textAllCaps="false"  
 android:background="@drawable/button\_background\_primary"  
 android:textColor="#FFFFFF"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginBottom="8dp"  
 android:padding="12dp"  
 android:layout\_marginHorizontal="16dp"  
 android:elevation="6dp" />  
  
 <!-- Sign Up Button -->  
 <Button  
 android:id="@+id/signupActivityButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Sign Up"  
 android:textAllCaps="false"  
 android:background="@drawable/button\_background\_secondary"  
 android:textColor="#000000"  
 android:layout\_marginTop="16dp"  
 android:padding="12dp"  
 android:layout\_marginHorizontal="16dp"  
 android:elevation="6dp" />  
</LinearLayout>

activity\_more.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp">  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <!-- Language Selection Buttons -->  
 <TextView  
 android:id="@+id/frenchText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="French Language (Français)"  
 android:textStyle="bold"  
 android:textSize="20sp"  
 android:padding="8dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:background="?attr/selectableItemBackground" />  
  
 <TextView  
 android:id="@+id/germanText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="German Language (Deutsch)"  
 android:textStyle="bold"  
 android:textSize="20sp"  
 android:padding="8dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:background="?attr/selectableItemBackground" />  
  
 <TextView  
 android:id="@+id/spanishText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Spanish Language (Español)"  
 android:textStyle="bold"  
 android:textSize="20sp"  
 android:padding="8dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:background="?attr/selectableItemBackground" />  
  
 <TextView  
 android:id="@+id/japaneseText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Japanese Language (日本語)"  
 android:textStyle="bold"  
 android:textSize="20sp"  
 android:padding="8dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:background="?attr/selectableItemBackground" />  
  
 <!-- Dynamic TextView for Information -->  
 <TextView  
 android:id="@+id/infoText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Click on a language to see details."  
 android:textSize="16sp"  
 android:padding="16dp" />  
  
 </LinearLayout>  
 </ScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_profile.xml

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#F9F9F9"  
 android:padding="16dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <!-- User Avatar -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:layout\_marginBottom="16dp">  
  
 <ImageView  
 android:id="@+id/user\_avatar"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:src="@drawable/ic\_user\_avatar"  
 android:contentDescription="@string/avatar\_desc"  
 android:background="@drawable/circle\_background"  
 android:layout\_marginBottom="8dp" />  
  
 <Button  
 android:id="@+id/btn\_choose\_avatar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Choose Avatar"  
 android:layout\_marginTop="8dp" />  
 </LinearLayout>  
  
 <!-- User Name Input -->  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Name"  
 android:textSize="16sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="4dp" />  
  
 <EditText  
 android:id="@+id/edit\_user\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter your name"  
 android:inputType="textPersonName"  
 android:layout\_marginBottom="16dp" />  
  
 <!-- User Email Input -->  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Email"  
 android:textSize="16sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="4dp" />  
  
 <EditText  
 android:id="@+id/edit\_user\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter your email"  
 android:inputType="textEmailAddress"  
 android:layout\_marginBottom="16dp" />  
  
 <!-- Language Progress Section -->  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Language Progress"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="8dp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <!-- German Progress -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:paddingVertical="8dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="German"  
 android:textSize="16sp" />  
  
 <ProgressBar  
 android:id="@+id/progress\_german"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:max="100" />  
 </LinearLayout>  
  
 <!-- French Progress -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:paddingVertical="8dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="French"  
 android:textSize="16sp" />  
  
 <ProgressBar  
 android:id="@+id/progress\_french"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:max="100" />  
 </LinearLayout>  
  
 <!-- Spanish Progress -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:paddingVertical="8dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Spanish"  
 android:textSize="16sp" />  
  
 <ProgressBar  
 android:id="@+id/progress\_spanish"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:max="100" />  
 </LinearLayout>  
  
 <!-- Japanese Progress -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:paddingVertical="8dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Japanese"  
 android:textSize="16sp" />  
  
 <ProgressBar  
 android:id="@+id/progress\_japanese"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:max="100" />  
 </LinearLayout>  
 </LinearLayout>  
  
 <!-- Edit Profile Button -->  
 <Button  
 android:id="@+id/btn\_edit\_profile"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Save Changes"  
 android:layout\_marginTop="16dp" />  
 </LinearLayout>  
</ScrollView>

activity\_signup.xml

<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center">  
  
  
 <!-- Logo -->  
 <ImageView  
 android:id="@+id/logo"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:src="@drawable/logo"  
 android:contentDescription="App Logo"  
 android:layout\_marginBottom="16dp" />  
  
 <!-- Welcome Text -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Welcome to My App!"  
 android:textSize="24sp"  
 android:layout\_gravity="center"  
 android:paddingBottom="32dp" />  
  
 <EditText  
 android:id="@+id/emailSignup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Email"  
 android:inputType="textEmailAddress" />  
  
 <EditText  
 android:id="@+id/passwordSignup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Password"  
 android:inputType="textPassword" />  
  
 <Button  
 android:id="@+id/signupButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Sign Up" />  
  
 <TextView  
 android:id="@+id/loginRedirect"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Already have an account? Log In"  
 android:gravity="center"  
 android:paddingTop="16dp"  
 android:textColor="@android:color/holo\_blue\_dark" />  
</LinearLayout>

activity\_spanish.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <Button  
 android:id="@+id/btn\_start\_learning"  
 android:layout\_width="328dp"  
 android:layout\_height="62dp"  
 android:text="Start Learning"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_spanish\_picture\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:scaleType="fitCenter"  
 android:contentDescription="Imagen para reconocer" />  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnSubmit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Enviar"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnNext"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Siguiente"  
 android:layout\_marginTop="8dp"  
 android:enabled="false" />  
  
</LinearLayout>

activity\_spanish\_quiz.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/questionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Pregunta"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp"/>  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:id="@+id/submitButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Enviar"  
 android:layout\_marginTop="16dp"  
 android:layout\_gravity="center\_horizontal"/>  
</LinearLayout>

activity\_spanish\_speech\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="20dp"  
 android:gravity="center">  
  
 <TextView  
 android:id="@+id/txt\_score"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Score: 0"  
 android:textSize="20sp"  
 android:textStyle="bold"/>  
  
 <TextView  
 android:id="@+id/txt\_difficulty"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Difficulty: Easy"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:paddingBottom="10dp"/>  
  
 <Button  
 android:id="@+id/btn\_change\_difficulty"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="🔄 Change Difficulty"/>  
  
 <TextView  
 android:id="@+id/txt\_word"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Word"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 android:paddingBottom="20dp"/>  
  
 <Button  
 android:id="@+id/btn\_listen"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="🔊 Listen"/>  
  
 <Button  
 android:id="@+id/btn\_speak"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="🎙 Speak"/>  
  
 <Button  
 android:id="@+id/btn\_next"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="➡ Next Word/Sentence"/>  
  
 <TextView  
 android:id="@+id/txt\_feedback"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Your feedback will appear here"  
 android:textSize="18sp"  
 android:paddingTop="20dp"/>  
</LinearLayout>

french\_activity\_speech\_recognition.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/questionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Pregunta"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp"/>  
  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:id="@+id/submitButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Enviar"  
 android:layout\_marginTop="16dp"  
 android:layout\_gravity="center\_horizontal"/>  
</LinearLayout>

french\_crossword.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:background="@android:color/white"  
 android:padding="20dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="French Crossword"  
 android:textSize="22sp"  
 android:textStyle="bold"  
 android:paddingBottom="16dp"  
 android:textColor="@android:color/black"/>  
  
 <GridView  
 android:id="@+id/crosswordGrid"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:numColumns="10"  
 android:verticalSpacing="6dp"  
 android:horizontalSpacing="6dp"  
 android:background="@android:color/darker\_gray"/>  
  
 <Button  
 android:id="@+id/btn\_submit"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:text="Submit"  
 android:layout\_marginTop="20dp"/>  
  
 <Button  
 android:id="@+id/btn\_save"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:text="Save Progress"  
 android:layout\_marginTop="10dp"/>  
</LinearLayout>

german\_crossword.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/white"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:padding="20dp"  
 tools:ignore="ExtraText">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingBottom="16dp"  
 android:text="Japanese Crossword"  
 android:textColor="@android:color/black"  
 android:textSize="22sp"  
 android:textStyle="bold" />  
  
 <GridView  
 android:id="@+id/crosswordGrid"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/darker\_gray"  
 android:horizontalSpacing="6dp"  
 android:numColumns="4"  
 android:verticalSpacing="6dp" />  
  
 <Button  
 android:id="@+id/btn\_submit"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="Confirm"  
 tools:ignore="HardcodedText" /> <!-- "Confirm" in Japanese -->  
  
 <Button  
 android:id="@+id/btn\_save"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:text="save" /> <!-- "Save" in Japanese -->  
  
</LinearLayout>

japanese\_crossword.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:background="@android:color/white"  
 android:padding="20dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Crucigrama en Español"  
 android:textSize="22sp"  
 android:textStyle="bold"  
 android:paddingBottom="16dp"  
 android:textColor="@android:color/black"/>  
  
 <GridView  
 android:id="@+id/crosswordGrid"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:numColumns="4"  
 android:verticalSpacing="6dp"  
 android:horizontalSpacing="6dp"  
 android:background="@android:color/darker\_gray"/>  
  
 <Button  
 android:id="@+id/btn\_submit"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:text="Verificar"  
 android:layout\_marginTop="20dp"/>  
  
 <Button  
 android:id="@+id/btn\_save"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:text="Guardar"  
 android:layout\_marginTop="10dp"/>  
</LinearLayout>

spanish\_crossword.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:background="@android:color/white"  
 android:padding="20dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Crucigrama en Español"  
 android:textSize="22sp"  
 android:textStyle="bold"  
 android:paddingBottom="16dp"  
 android:textColor="@android:color/black"/>  
  
 <GridView  
 android:id="@+id/crosswordGrid"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:numColumns="4"  
 android:verticalSpacing="6dp"  
 android:horizontalSpacing="6dp"  
 android:background="@android:color/darker\_gray"/>  
  
 <Button  
 android:id="@+id/btn\_submit"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:text="Verificar"  
 android:layout\_marginTop="20dp"/>  
  
 <Button  
 android:id="@+id/btn\_save"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:text="Guardar"  
 android:layout\_marginTop="10dp"/>  
</LinearLayout>

**4.2: Data Dictionary**

**Sub-topics:**

1. **Database Structure**
   * Tables for user profiles, language content, progress tracking, and AI model data.
   * Relationships between tables (e.g., User → Progress Tracking → Lessons).
2. **Fields and Attributes**
   * Fields for each table (e.g., User\_ID, Name, Preferred\_Language, Progress\_Level, etc.).
   * Data types and constraints for each field (e.g., User\_ID as Primary Key, Language\_Code as VARCHAR).
3. **Keys and Relationships**
   * Primary and foreign keys for tables.
   * Referential integrity constraints.
4. **Sample Data**
   * Example entries for key tables to showcase how data is stored.
5. **Indexing and Optimization**
   * Indexed columns for faster search and retrieval.

Database normalization level

**4.3: Program Description**

**Sub-topics:**

1. **Overall Functionality**
   * Overview of how the app operates, including AI-based recommendations and language lessons.
2. **Modules and Components**
   * Explanation of each module:
     + **User Module**: Account creation, login, and preferences.
     + **Lesson Module**: Language content delivery.
     + **AI Recommendation Module**: Personalized content suggestions.
     + **Progress Tracker**: Tracks user learning progress.
3. **Algorithms and Logic**
   * AI-based logic for language learning suggestions.
   * Algorithms for evaluating quizzes and pronunciation assessments.
4. **Technology Stack**
   * Frontend: Android Studio.
   * Backend: Firebase/SQL Server.
   * AI: TensorFlow/Keras for machine learning.
5. **Integration and API Usage**
   * APIs used (e.g., Google Translate API, Speech-to-Text API).

Communication

**4.4: Naming Conventions**

**Sub-topics:**

1. **General Naming Standards**
   * Consistent use of camelCase for variables and methods.
   * PascalCase for classes.
2. **Database Naming**
   * Table names: Singular (e.g., User, Lesson, Progress).
   * Column names: Descriptive (e.g., Lesson\_Title, User\_Email).
3. **Code Naming**
   * Functions: Verb-noun structure (e.g., getUserProgress, loadLessonContent).
   * Constants: Uppercase with underscores (e.g., MAX\_ATTEMPTS, API\_KEY).
4. **UI Elements**
   * Activity and fragment IDs: Prefix activity\_ or fragment\_.
   * Widget IDs: Prefix by type (e.g., btn\_submit, txt\_username).
5. **File Naming**
   * Activity files: ActivityNameActivity.java.
   * XML layouts: activity\_name.xml.

**4.5: Validations**

**Sub-topics:**

1. **User Input Validations**
   * Registration fields (e.g., email format, strong password criteria).
   * Name and username validation (e.g., alphanumeric characters only).
2. **Data Integrity Validations**
   * Prevent duplicate user accounts using unique constraints.
   * Ensure proper language codes for lessons.
3. **AI and Learning Content Validations**
   * Validation of AI responses against predefined thresholds.
   * Content integrity checks during updates.
4. **Error Handling and Feedback**
   * Descriptive error messages for invalid inputs.
   * Retry mechanisms for failed API calls.
5. **Security Validations**
   * SQL injection prevention.
   * Secure storage of sensitive data (e.g., encrypted passwords).

**Program Listing - Sub Topics**

**1. User Interface (UI) & Navigation**

• Main Menu (Home, Courses, Progress, Settings)

• Language Selection

• Course Modules UI (Lessons, Quizzes, Crosswords)

• Interactive Elements (Drag & Drop, Speech Input, Picture Identification)

**2. AI-Powered Language Learning Features**

• Offline AI Chatbot for Language Practice

• AI-based Adaptive Quiz System

• Speech Recognition & Pronunciation Correction

• Levenshtein Distance Algorithm for Spelling and Sentence Correction

**3. Content Management System (CMS)**

• Lesson Creation & Management

• Dynamic Quiz & Crossword Generation using AI

• Flashcards & Vocabulary Builder

**4. Machine Learning Integration (TensorFlow Lite - TFLite)**

• Model Training for Language Processing

• AI-Based Quiz Difficulty Adaptation

• AI-Powered Word Suggestions & Sentence Formation

**5. Database & Storage**

• Local Database (Room/SQLite) for Offline Data

• Cloud Storage for User Progress Sync (Optional)

• JSON Data Handling for Lesson Content

**6. Gamification & User Engagement**

• XP & Level System

• Badges & Achievements

• Daily Streak Tracking

**7. Security & User Authentication**

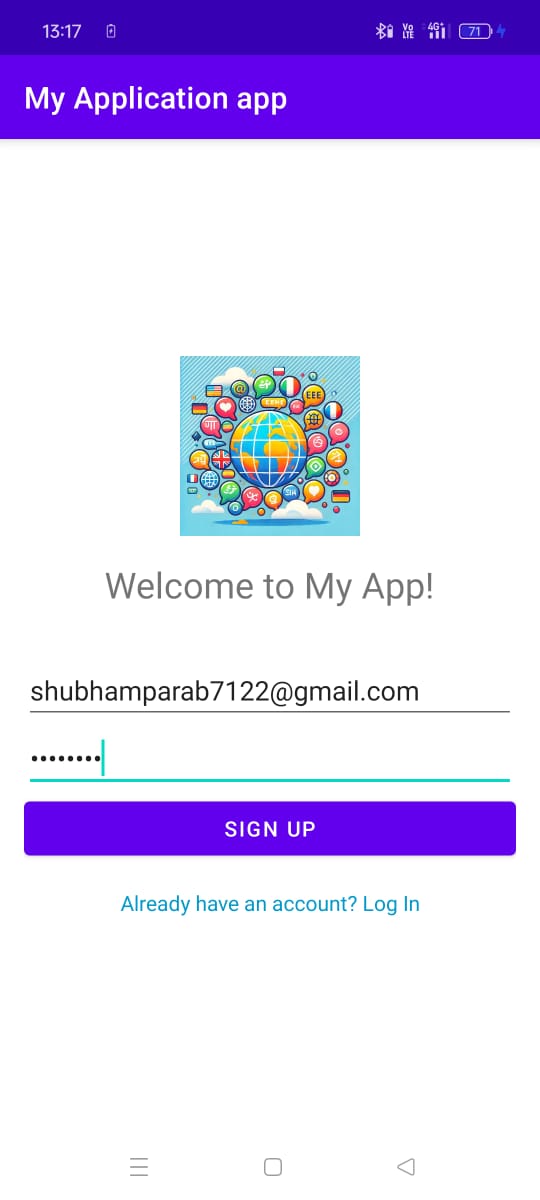
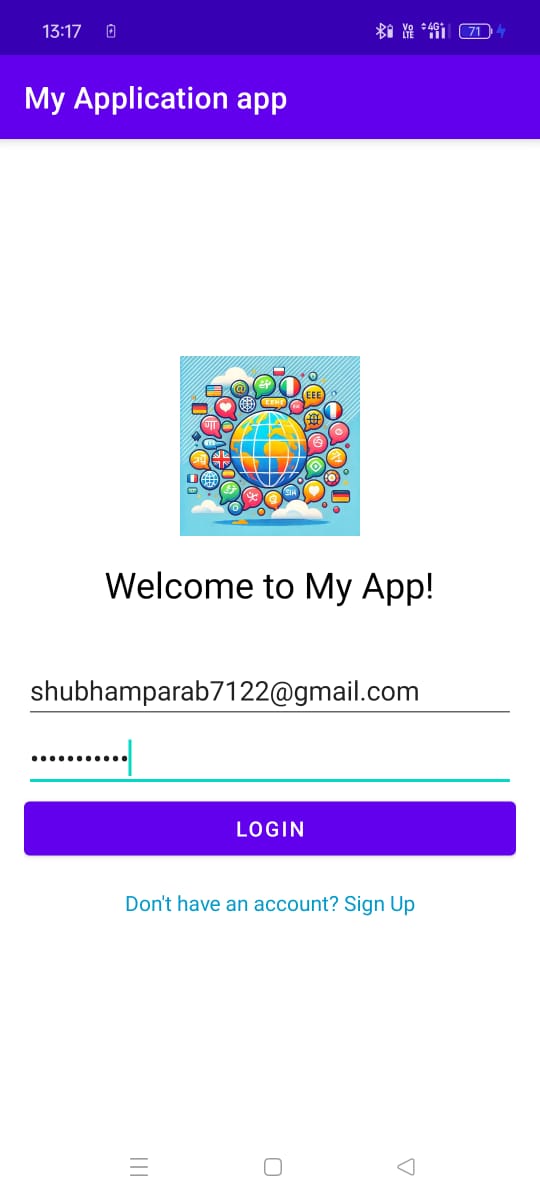
• Login & Registration (Google Sign-In, Email Authentication)

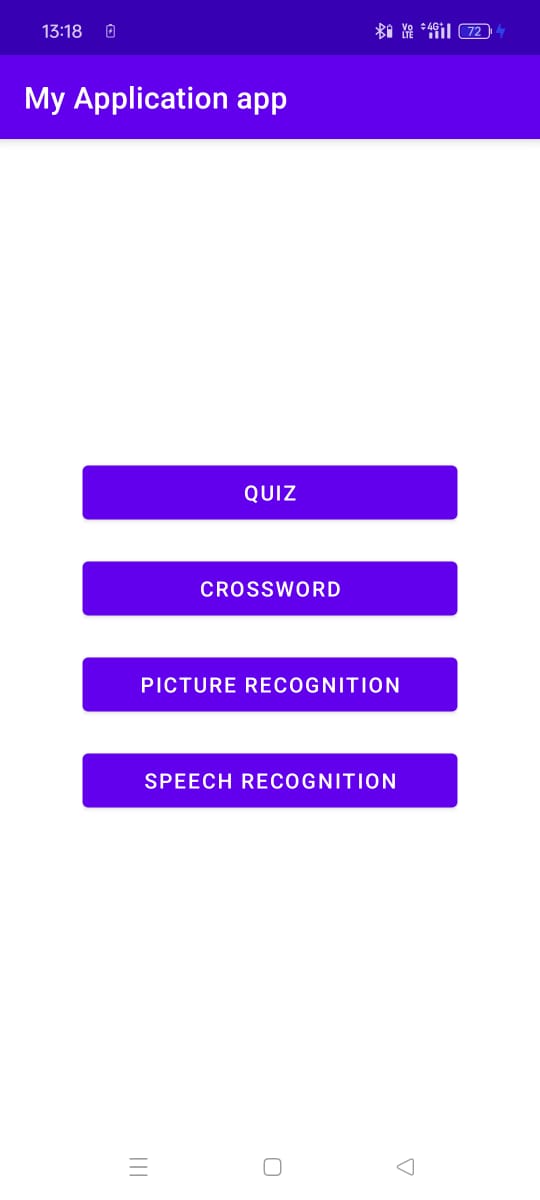
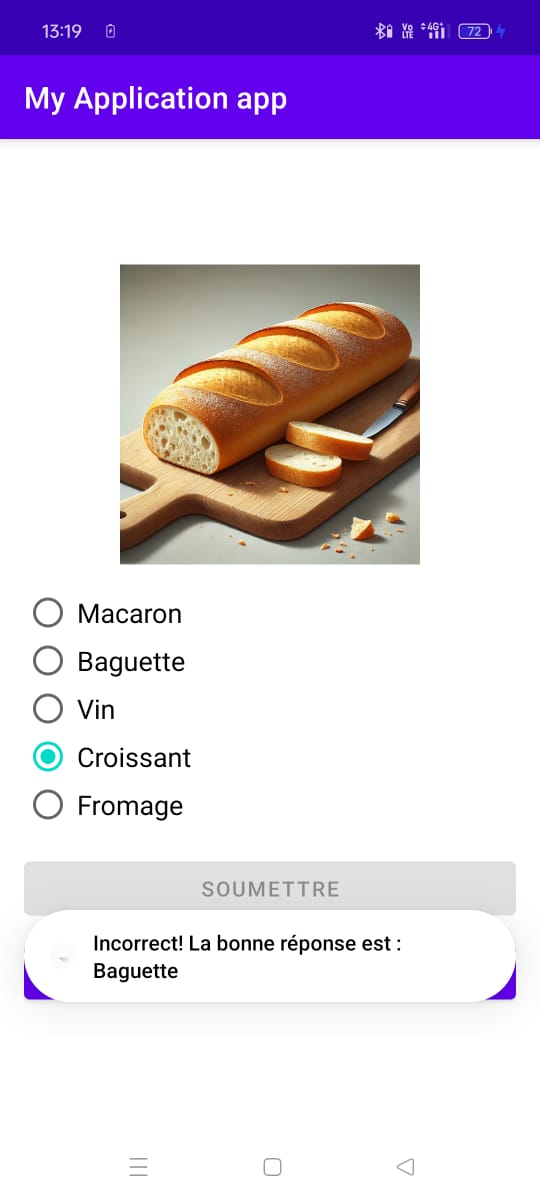
• Data Encryption for User Progress & Preferences

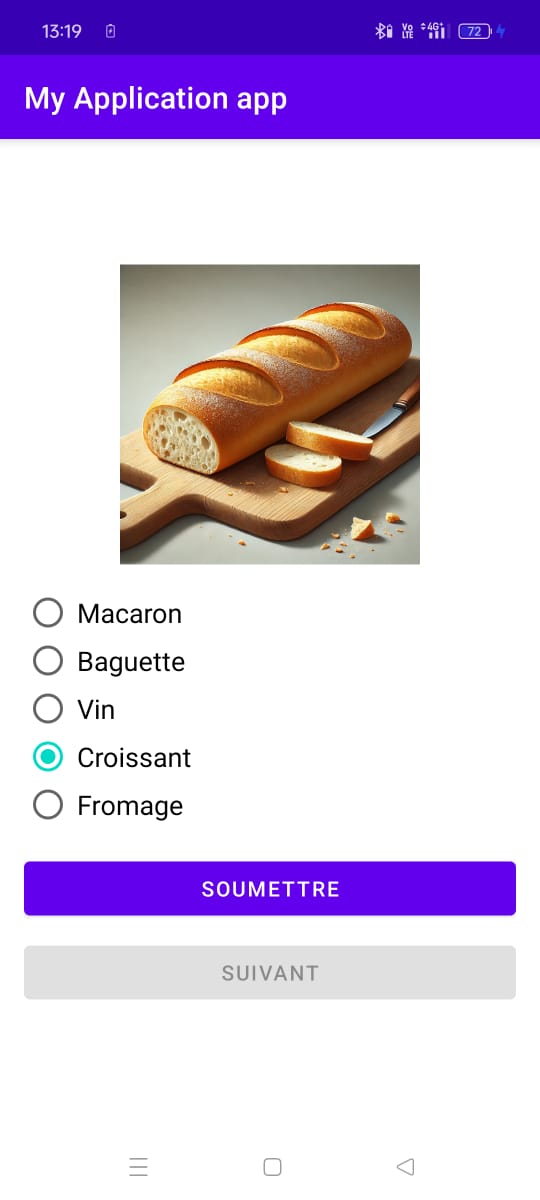
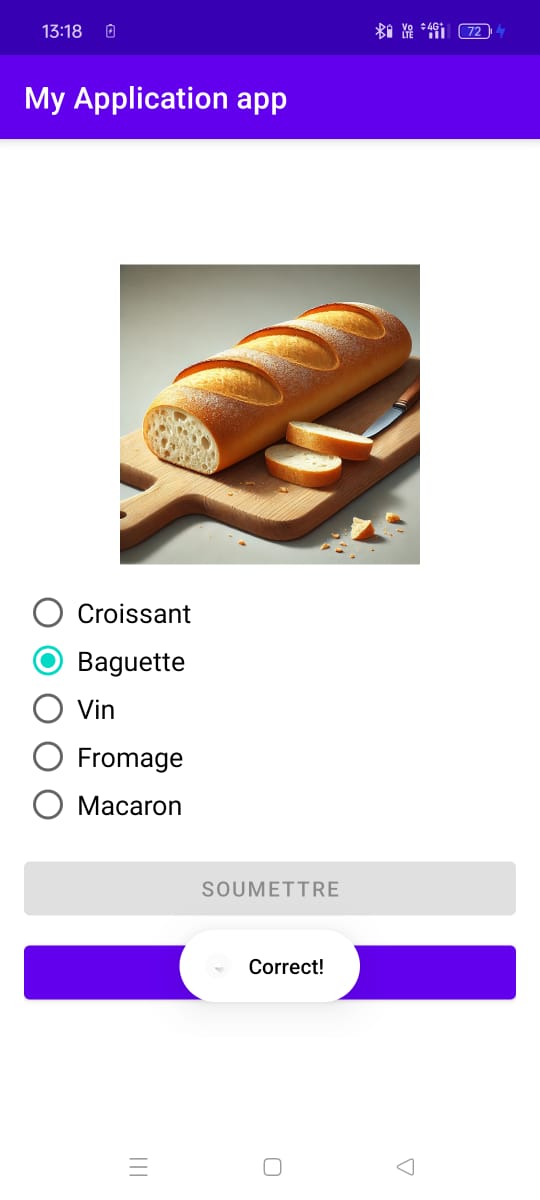
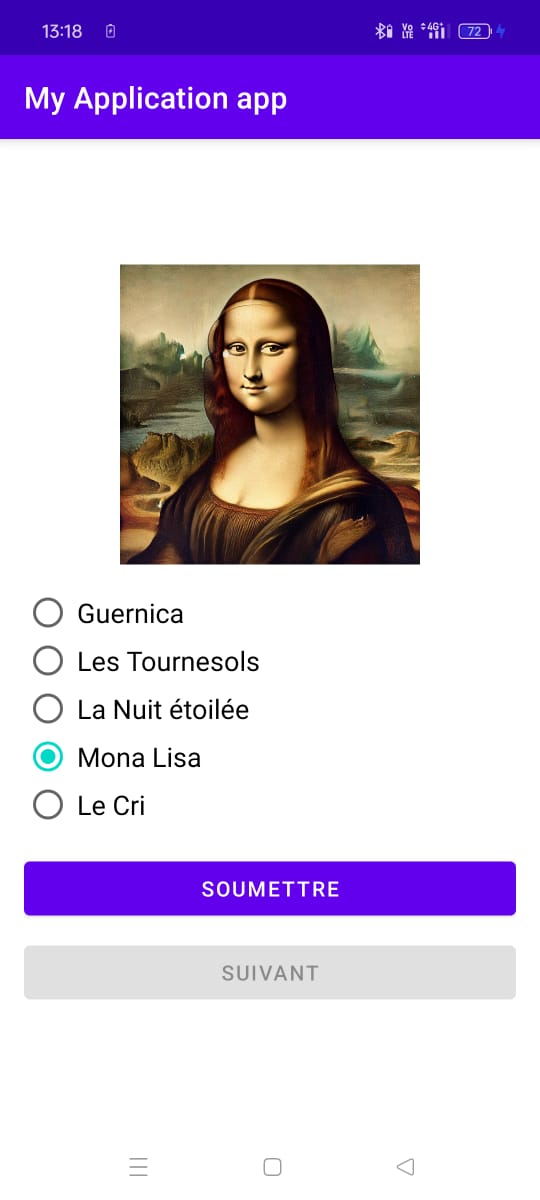
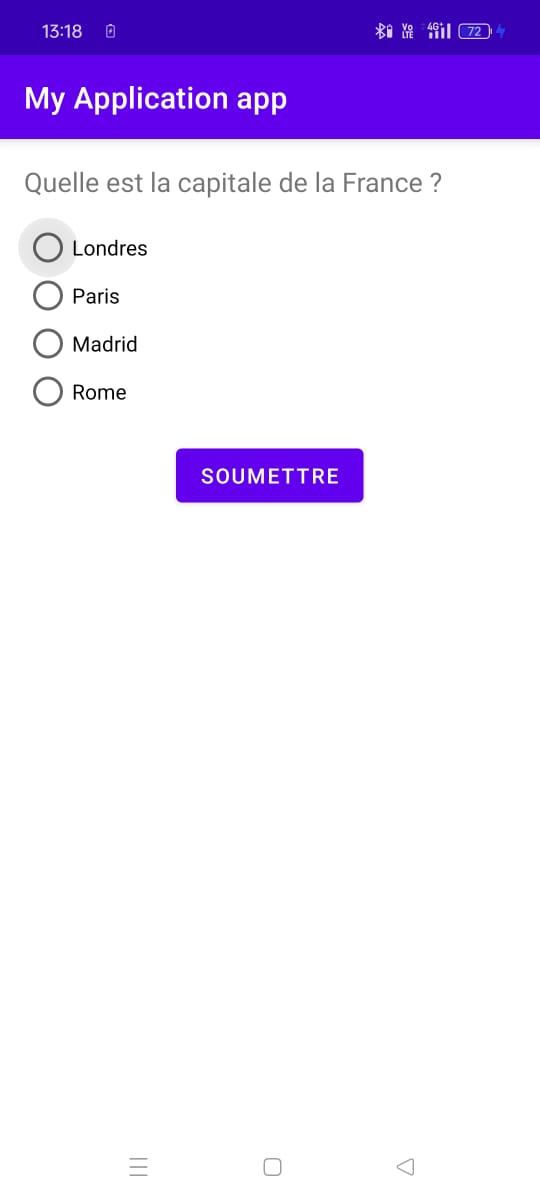
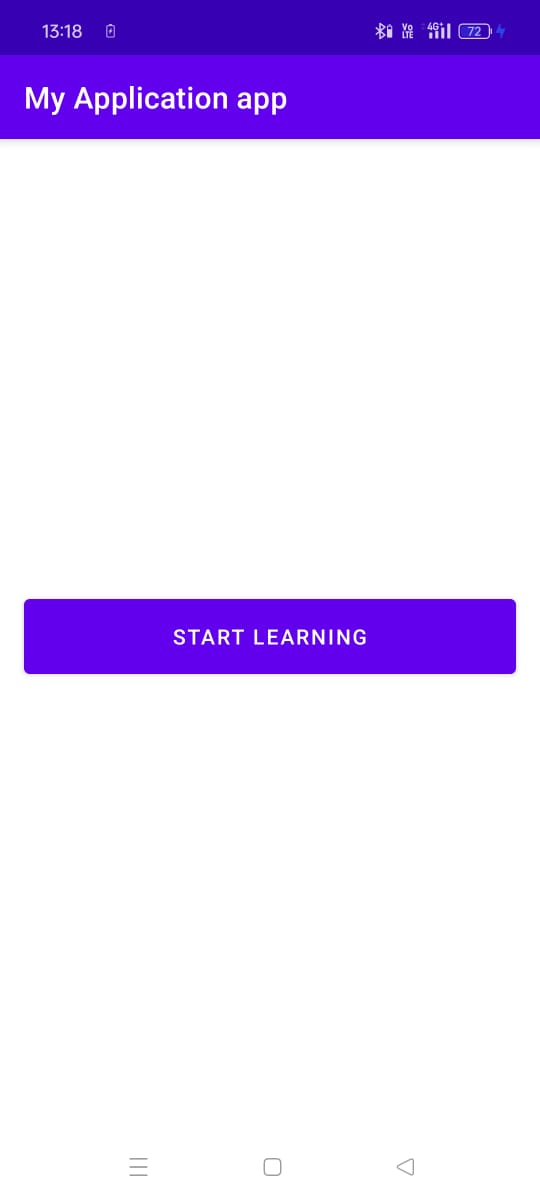
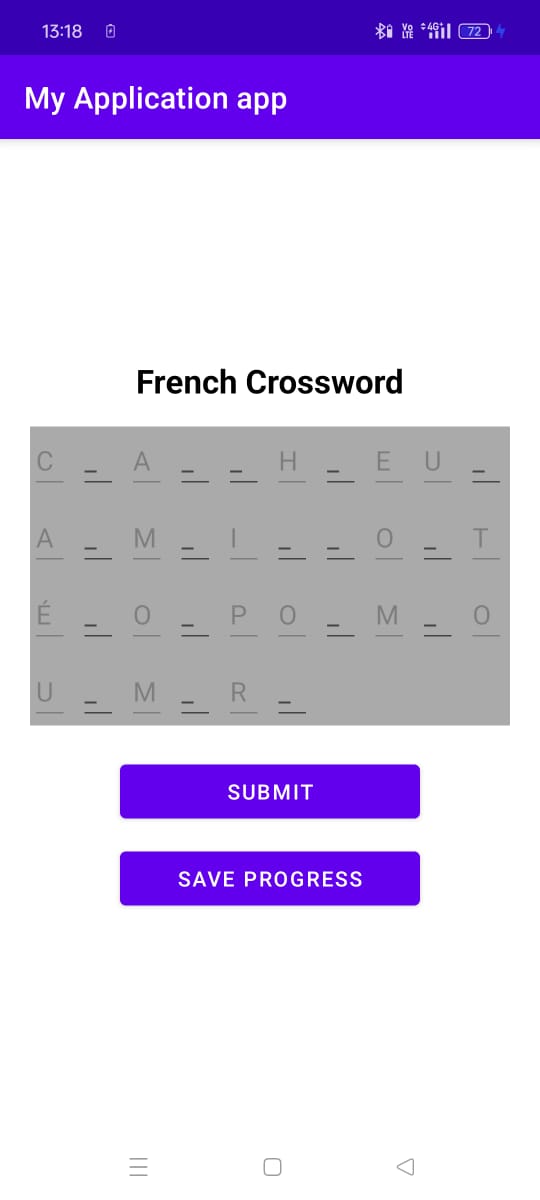
**5.1 Cost Estimation**

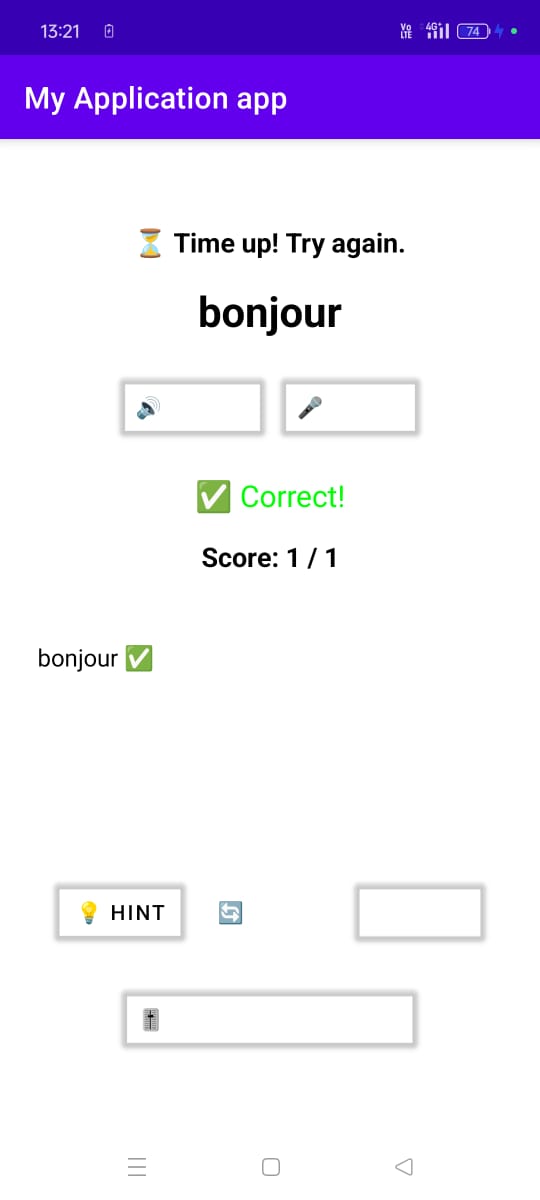
|  |  |
| --- | --- |
| Category | Estimated Cost (₹) |
| Development Tools & Software |  |
| Android Studio (Free) | 0 |
| TensorFlow Lite (Open Source) | 0 |
| Firebase (Optional – Free Tier) | 0 |
| GitHub/GitLab (Version Control - Free) | 0 |
| Development & AI Model Training |  |
| AI Model Training (Cloud GPU Usage) | 5,000 |
| Data Collection & Preprocessing | 3,000 |
| Design & UI/UX |  |
| UI/UX Design (Freelancer or Figma Pro) | 5,000 |
| Icons & Graphics | 2,000 |
| App Development |  |
| Android Developer (Freelancer – Partial Assistance) | 15,000 |
| Backend Setup & Integration | 5,000 |
| Testing & Debugging |  |
| Device Testing (Purchase/Rent Extra Devices) | 5,000 |
| Beta Testing & User Feedback | 2,000 |
| Marketing & Deployment |  |
| Google Play Store Listing Fee (One-time) | 2,500 |
| App Promotion & Initial Marketing | 5,000 |
| Total Estimated Cost | ₹49,500 |

**App Screenshots**









**5.2 Test Cases for LinguaEase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Steps** | **Expected Outcome** | **Status** |
| **TC01** | **Test app launch** | **Open the LinguaEase app on an Android device** | **The app launches successfully without crashes** | **Passed** |
| **TC02** | **Test language selection** | **Select a language (French, German, Spanish, or Japanese) from the options** | **The selected language is set, and UI updates accordingly** | **Passed** |
| **TC03** | **Test text-based quiz** | **Attempt a text-based quiz by selecting or typing answers** | **The app evaluates responses, provides feedback, and updates the score** | **Passed** |
| **TC04** | **Test speech recognition** | **Speak a word/phrase into the microphone for pronunciation practice** | **The app processes the input, compares it with the correct pronunciation, and provides feedback** | **Passed** |
| **TC05** | **Test picture identification** | **Select the correct image for a given word/phrase in the chosen language** | **The app verifies the selection and provides appropriate feedback** | **Passed** |
| **TC06** | **Test adaptive AI quiz** | **Complete multiple quizzes to test AI adaptation** | **The AI adjusts difficulty based on user performance** | **Passed** |
| **TC07** | **Test offline mode** | **Disconnect from the internet and access learning modules** | **The app functions smoothly with offline AI-driven quizzes** | **Passed** |
|  |  |  |  |  |
| **TC08** | **Test app performance** | **Use the app on different Android devices with varying hardware capabilities** | **The app maintains smooth performance and responsiveness** | **Passed** |
| **TC09** | **Test user progress tracking** | **Complete multiple exercises and revisit progress tracking** | **User progress is stored and displayed correctly** | **Passed** |

1. **Conclusion**

**The LinguaEase app is designed to provide an immersive and effective language-learning experience by leveraging AI-powered features**

**By incorporating intelligent learning mechanisms and an intuitive user interface, LinguaEase aims to make language acquisition more accessible and engaging for users of all proficiency levels.**

**Furthermore, by integrating user feedback, the app will evolve to meet learners' changing needs, ensuring a refined experience that stays relevant in the competitive landscape of language-learning applications.**

**7. Limitations of the System**

1. **Hardware Limitations**
   * **Older devices may experience lag due to AI computations.**
   * **Speech recognition accuracy may vary based on microphone quality.**
2. **Offline Functionality**
   * **Some advanced AI features might require periodic internet access for updates.**
3. **AI Adaptation Challenges**
   * **The adaptive difficulty system may need further refinement to match user skill levels accurately.**
4. **Speech Recognition Accuracy**
   * **Variability in pronunciation and accents may affect recognition quality.**
5. **Limited Gamification Features**
   * **Additional interactive elements and rewards could enhance engagement.**
6. **User Interface (UI) Challenges**
   * **Complex UI elements might be difficult to navigate for first-time users.**
   * **Some UI elements may not scale properly on different screen sizes.**
7. **Limited Personalization**
   * **While AI adjusts quiz difficulty, personalized lesson plans and study paths are limited.**
8. **Data Privacy Concerns**
   * **If cloud-based updates are implemented, ensuring data security and privacy compliance could be challenging.**
9. **Limited Multimodal Learning**
   * **The app focuses primarily on text, speech, and image-based learning but lacks interactive storytelling or immersive AR/VR experiences.**
10. **Cross-Platform Compatibility**

* **The app is currently designed for Android, limiting its accessibility to iOS and web users.**

1. **Limited Cultural Context Integration**

* **Language learning might lack cultural nuances, affecting real-world applicability.**

1. **Dependence on AI Training Data**

* **The effectiveness of AI-driven quizzes and speech recognition is reliant on the quality and diversity of training data.**

**8. Future Enhancements**

1. **Expanded Language Support – Add more languages such as Mandarin, Italian, and Korean.**
2. **Multiplayer Mode – Enable real-time quiz challenges with other users.**
3. **Enhanced Gamification – Introduce streaks, badges, and leaderboards.**
4. **More AI Personalization – Improve adaptive learning models for a tailored experience.**
5. **Speech Synthesis Improvements – Enhance pronunciation feedback with AI-generated speech.**

**9. Bibliography**

1. **Android Developer Guide: https://developer.android.com/**
2. **TensorFlow Lite Documentation: https://www.tensorflow.org/lite**
3. **Google Speech Recognition API: https://cloud.google.com/speech-to-text**
4. **Duolingo Research Papers: https://research.duolingo.com/**
5. **AI in Language Learning: https://www.nature.com/articles/s41599-020-0468-5**