Bharatiya Vidya Bhavan's
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri (W), Mumbai – 400 058.
Department of Computer Science and Engineering

Experiment	6
Aim	Create an app for children where children can learn numbers and alphabets
Objective	• To Create App of children • To use Upper Tabs in navigation bar • To implement Splash Screen in Flutter
Name	**Prabhat Anand Tiwari**
UCID	**2024510066**
Class	**FYMCA**
Batch	**C**
Date of Submission	**18/03/2025**

Technology used	**Flutter**
Task	Children should be able to even recognize the numbers and alphabets by quiz form. Create 4 upper Tabs with Splash Screen at the start: 1) Practice/Lessons 2) Quiz 3) Leader Board 4) Profile
Code with proper label	**Folder structure:** lib models screens tabs main.dart **main.dart**

```dart
import 'package:flutter/material.dart';
import 'screens/splash_screen.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Kids Learning App',
```

```
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.blue,
        fontFamily: 'Comic Sans MS',
        colorScheme: ColorScheme.fromSeed(
          seedColor: Colors.blue,
          primary: Colors.blue,
          secondary: Colors.orange,
          tertiary: Colors.green,
        ),
        textTheme: const TextTheme(
          displayLarge: TextStyle(
              fontSize: 32, fontWeight:
FontWeight.bold, color: Colors.blue),
          displayMedium: TextStyle(
              fontSize: 24, fontWeight:
FontWeight.bold, color: Colors.blue),
          bodyLarge: TextStyle(fontSize: 18, color:
Colors.black87),
        ),
        useMaterial3: true,
      ),
      home: const SplashScreen(),
    );
  }
}
```

splash_screen.dart

```
import 'package:flutter/material.dart';
import 'dart:async';
import 'home_screen.dart';

class SplashScreen extends StatefulWidget {
  const SplashScreen({super.key});

  @override
  State<SplashScreen> createState() =>
_SplashScreenState();
```

```
}

class _SplashScreenState extends
State<SplashScreen>
    with SingleTickerProviderStateMixin {
  late AnimationController
_animationController;
  late Animation<double> _animation;

  @override
  void initState() {
    super.initState();
    _animationController =
AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animation = CurvedAnimation(
      parent: _animationController,
      curve: Curves.bounceOut,
    );
    _animationController.forward();

    Timer(const Duration(seconds: 3), () {
      if (mounted) {

Navigator.of(context).pushReplacement(
          MaterialPageRoute(builder:
(context) => const HomeScreen()),
        );
      }
    });
  }
```

```
  @override
  void dispose() {
    _animationController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Center(
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            ScaleTransition(
              scale: _animation,
              child: Image.asset(
                'assets/splash.png',
                height: 200,
                width: 200,
              ),
            ),
            const SizedBox(height: 30),
            const Text(
              'Kids Learning App',
              style: TextStyle(
                fontSize: 28,
                fontWeight: FontWeight.bold,
                color: Colors.blue,
              ),
            ),
```

```
            const SizedBox(height: 20),
            const Text(
              'Learn Numbers & Alphabets',
              style: TextStyle(
                fontSize: 18,
                color: Colors.orange,
              ),
            ),
            const SizedBox(height: 30),
            const CircularProgressIndicator(
              valueColor:
AlwaysStoppedAnimation<Color>(Colors.blue),
            ),
          ],
        ),
      ),
    );
  }
}
```

quiz_detail_screen.dart

```
import 'package:flutter/material.dart';
import '../models/quiz.dart';
import '../models/quiz_question.dart';
import 'dart:async';

class QuizDetailScreen extends
StatefulWidget {
  final Quiz quiz;

  const QuizDetailScreen({super.key,
required this.quiz});
```

```
  @override
  State<QuizDetailScreen> createState() =>
_QuizDetailScreenState();
}

class _QuizDetailScreenState extends
State<QuizDetailScreen> {
  int _currentQuestionIndex = 0;
  int _score = 0;
  bool _quizCompleted = false;
  bool _answerSelected = false;
  String? _selectedAnswer;
  late List<QuizQuestion> _questions;
  late Timer _timer;
  int _timeRemaining = 0;

  @override
  void initState() {
    super.initState();
    _questions = _getQuizQuestions();
    _timeRemaining = widget.quiz.timeLimit *
60; // Convert minutes to seconds
    _startTimer();
  }

  void _startTimer() {
    _timer = Timer.periodic(const
Duration(seconds: 1), (timer) {
      if (_timeRemaining > 0) {
        setState(() {
          _timeRemaining--;
        });
```

```
      } else {
        _timer.cancel();
        _endQuiz();
      }
    });
  }

  @override
  void dispose() {
    _timer.cancel();
    super.dispose();
  }

  List<QuizQuestion> _getQuizQuestions() {
    if
(widget.quiz.title.contains('Alphabet')) {
      return [
        QuizQuestion(
          id: '1',
          question: 'Which letter comes
after A?',
          options: ['B', 'C', 'D', 'E'],
          correctAnswer: 'B',
          imageUrl: 'assets/ques_1.png',
        ),
        QuizQuestion(
          id: '2',
          question: 'Which letter makes the
"mmm" sound?',
          options: ['N', 'M', 'W', 'P'],
          correctAnswer: 'M',
          imageUrl: 'assets/ques_2.png',
        ),
```

```
        QuizQuestion(
          id: '3',
          question: 'What letter does
"Apple" start with?',
          options: ['B', 'A', 'P', 'O'],
          correctAnswer: 'A',
          imageUrl: 'assets/ques_3.png',
        ),
        QuizQuestion(
          id: '4',
          question: 'Which letter is a
vowel?',
          options: ['T', 'D', 'E', 'R'],
          correctAnswer: 'E',
          imageUrl: 'assets/ques_4.png',
        ),
        QuizQuestion(
          id: '5',
          question: 'What letter does "Zoo"
start with?',
          options: ['X', 'Y', 'Z', 'W'],
          correctAnswer: 'Z',
          imageUrl: 'assets/ques_5.png',
        ),
      ];
    } else if
(widget.quiz.title.contains('Number')) {
      return [
        QuizQuestion(
          id: '1',
          question: 'What number comes after
3?',
          options: ['2', '4', '5', '6'],
```

```
          correctAnswer: '4',
          imageUrl: 'assets/ques_1.png',
        ),
        QuizQuestion(
          id: '2',
          question: 'How many fingers are on
one hand?',
          options: ['4', '5', '6', '10'],
          correctAnswer: '5',
          imageUrl: 'assets/ques_2.png',
        ),
        QuizQuestion(
          id: '3',
          question: 'What is 2 + 2?',
          options: ['3', '4', '5', '6'],
          correctAnswer: '4',
          imageUrl: 'assets/ques_3.png',
        ),
        QuizQuestion(
          id: '4',
          question: 'Which number is
greater: 7 or 9?',
          options: ['7', '9', 'They are
equal', 'None'],
          correctAnswer: '9',
          imageUrl: 'assets/ques_4.png',
        ),
        QuizQuestion(
          id: '5',
          question: 'What is 5 - 2?',
          options: ['2', '3', '4', '7'],
          correctAnswer: '3',
          imageUrl: 'assets/ques_5.png',
```

```
        ),
      ];
    } else {
      // Default questions
      return [
        QuizQuestion(
          id: '1',
          question: 'Sample Question 1',
          options: ['Option A', 'Option B',
'Option C', 'Option D'],
          correctAnswer: 'Option A',
          imageUrl: 'assets/ques_1.png',
        ),
        QuizQuestion(
          id: '2',
          question: 'Sample Question 2',
          options: ['Option A', 'Option B',
'Option C', 'Option D'],
          correctAnswer: 'Option B',
          imageUrl: 'assets/ques_2.png',
        ),
        QuizQuestion(
          id: '3',
          question: 'Sample Question 3',
          options: ['Option A', 'Option B',
'Option C', 'Option D'],
          correctAnswer: 'Option C',
          imageUrl: 'assets/ques_3.png',
        ),
        QuizQuestion(
          id: '4',
          question: 'Sample Question 4',
          options: ['Option A', 'Option B',
```

```
'Option C', 'Option D'],
          correctAnswer: 'Option D',
          imageUrl: 'assets/ques_4.png',
        ),
        QuizQuestion(
          id: '5',
          question: 'Sample Question 5',
          options: ['Option A', 'Option B',
'Option C', 'Option D'],
          correctAnswer: 'Option A',
          imageUrl: 'assets/ques_5.png',
        ),
      ];
    }
  }

  void _checkAnswer(String answer) {
    if (_answerSelected) return;

    setState(() {
      _answerSelected = true;
      _selectedAnswer = answer;
    });

    final isCorrect = answer ==
_questions[_currentQuestionIndex].correctAns
wer;

    if (isCorrect) {
      setState(() {
        _score++;
      });
    }
```

```
    // Wait before moving to next question
    Future.delayed(const Duration(seconds:
1), () {
      if (_currentQuestionIndex <
_questions.length - 1) {
        setState(() {
          _currentQuestionIndex++;
          _answerSelected = false;
          _selectedAnswer = null;
        });
      } else {
        _endQuiz();
      }
    });
  }

  void _endQuiz() {
    _timer.cancel();
    setState(() {
      _quizCompleted = true;
    });
  }

  String _formatTime(int seconds) {
    final minutes = seconds ~/ 60;
    final remainingSeconds = seconds % 60;
    return
'$minutes:${remainingSeconds.toString().padL
eft(2, '0')}';
  }

  @override
```

```
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.quiz.title),
        backgroundColor: Colors.orange,
        foregroundColor: Colors.white,
      ),
      body: _quizCompleted ?
_buildResultScreen() : _buildQuizScreen(),
    );
  }

  Widget _buildQuizScreen() {
    final currentQuestion =
_questions[_currentQuestionIndex];

    return Column(
      children: [
        // Timer and progress bar
        Container(
          padding: const EdgeInsets.all(16),
          color: Colors.orange.shade50,
          child: Row(
            mainAxisAlignment:
MainAxisAlignment.spaceBetween,
            children: [
              Text(
                'Question
${_currentQuestionIndex +
1}/${_questions.length}',
                style: const TextStyle(
                  fontSize: 16,
                  fontWeight:
```

```
FontWeight.bold,
                ),
              ),
              Row(
                children: [
                  const Icon(Icons.timer,
color: Colors.orange),
                  const SizedBox(width: 4),
                  Text(

_formatTime(_timeRemaining),
                    style: const TextStyle(
                      fontSize: 16,
                      fontWeight:
FontWeight.bold,
                    ),
                  ),
                ],
              ),
            ],
          ),
        ),
        LinearProgressIndicator(
          value: (_currentQuestionIndex + 1)
/ _questions.length,
          backgroundColor:
Colors.grey.shade200,
          valueColor: const
AlwaysStoppedAnimation<Color>(Colors.orange)
,
          minHeight: 10,
        ),
```

```
        // Question content
        Expanded(
          child: SingleChildScrollView(
            padding: const
EdgeInsets.all(20),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.center,
              children: [
                const SizedBox(height: 20),
                if (currentQuestion.imageUrl
!= null)
                  Image.asset(

currentQuestion.imageUrl!,
                    height: 150,
                    width: 150,
                  ),
                const SizedBox(height: 30),
                Text(
                  currentQuestion.question,
                  textAlign:
TextAlign.center,
                  style: const TextStyle(
                    fontSize: 24,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 40),

...currentQuestion.options.map((option) {
                  final isSelected =
```

```
_selectedAnswer == option;
                  final isCorrect = option
== currentQuestion.correctAnswer;

                  Color backgroundColor =
Colors.white;
                  Color borderColor =
Colors.grey.shade300;

                  if (_answerSelected) {
                    if (isSelected &&
isCorrect) {
                      backgroundColor =
Colors.green.shade100;
                      borderColor =
Colors.green;
                    } else if (isSelected &&
!isCorrect) {
                      backgroundColor =
Colors.red.shade100;
                      borderColor =
Colors.red;
                    } else if (isCorrect) {
                      backgroundColor =
Colors.green.shade100;
                      borderColor =
Colors.green;
                    }
                  } else if (isSelected) {
                    backgroundColor =
Colors.orange.shade100;
                    borderColor =
Colors.orange;
```

```
                    }

                    return GestureDetector(
                      onTap: () =>
_answerSelected ? null :
_checkAnswer(option),
                      child: Container(
                        width:
double.infinity,
                        margin: const
EdgeInsets.only(bottom: 16),
                        padding: const
EdgeInsets.all(16),
                        decoration:
BoxDecoration(
                          color:
backgroundColor,
                          border:
Border.all(color: borderColor, width: 2),
                          borderRadius:
BorderRadius.circular(12),
                        ),
                        child: Text(
                          option,
                          textAlign:
TextAlign.center,
                          style: TextStyle(
                            fontSize: 18,
                            fontWeight:
                                isSelected ?
FontWeight.bold : FontWeight.normal,
                          ),
                        ),
```

```
                    ),
                  );
                }),
              ],
            ),
          ),
        ),
      ],
    );
  }

  Widget _buildResultScreen() {
    final percentage = (_score /
_questions.length) * 100;
    final isPassed = percentage >= 70;

    return Center(
      child: Column(
        mainAxisAlignment:
MainAxisAlignment.center,
        children: [
          const SizedBox(height: 40),
          Icon(
            isPassed ? Icons.emoji_events :
Icons.sentiment_satisfied_alt,
            size: 100,
            color: isPassed ? Colors.amber :
Colors.orange,
          ),
          const SizedBox(height: 20),
          Text(
            isPassed ? 'Great Job!' : 'Good
Try!',
```

```
            style: TextStyle(
              fontSize: 28,
              fontWeight: FontWeight.bold,
              color: isPassed ? Colors.green
: Colors.orange,
            ),
          ),
          const SizedBox(height: 20),
          Text(
            'Your Score',
            style: TextStyle(
              fontSize: 20,
              color: Colors.grey.shade700,
            ),
          ),
          const SizedBox(height: 10),
          Text(
            '$_score/${_questions.length}',
            style: const TextStyle(
              fontSize: 48,
              fontWeight: FontWeight.bold,
              color: Colors.orange,
            ),
          ),
          Text(

'${percentage.toStringAsFixed(0)}%',
            style: TextStyle(
              fontSize: 24,
              fontWeight: FontWeight.bold,
              color: Colors.grey.shade700,
            ),
          ),
```

```
            const SizedBox(height: 40),
            ElevatedButton.icon(
              onPressed: () {
                Navigator.pop(context);
              },
              icon: const Icon(Icons.home),
              label: const Text('Back to
Quizzes'),
              style: ElevatedButton.styleFrom(
                padding: const
EdgeInsets.symmetric(horizontal: 30,
vertical: 15),
              ),
            ),
            const SizedBox(height: 20),
            TextButton.icon(
              onPressed: () {
                setState(() {
                  _currentQuestionIndex = 0;
                  _score = 0;
                  _quizCompleted = false;
                  _answerSelected = false;
                  _selectedAnswer = null;
                  _timeRemaining =
widget.quiz.timeLimit * 60;
                });
                _startTimer();
              },
              icon: const Icon(Icons.refresh),
              label: const Text('Try Again'),
            ),
          ],
        ),
```

```
    );
  }
}
```

lesson_dart_Screen.dart

```
import 'package:flutter/material.dart';
import '../models/lesson.dart';
import
'package:flutter_tts/flutter_tts.dart';

class LessonDetailScreen extends
StatefulWidget {
  final Lesson lesson;

  const LessonDetailScreen({super.key,
required this.lesson});

  @override
  State<LessonDetailScreen> createState() =>
_LessonDetailScreenState();
}

class _LessonDetailScreenState extends
State<LessonDetailScreen> {
  int _currentPage = 0;
  final PageController _pageController =
PageController();
  final FlutterTts flutterTts =
FlutterTts();

  Future<void> _speak(String letter) async {
    await flutterTts.speak(letter);
  }
```

```
  // Sample lesson content based on the
lesson type
  List<Map<String, dynamic>>
_getLessonContent() {
    if (widget.lesson.title.contains('A-F'))
{
      return [
        {
          'letter': 'A',
          'image': 'assets/apple.png',
          'word': 'Apple',
          'description': 'A is for Apple',
        },
        {
          'letter': 'B',
          'image': 'assets/ball.png',
          'word': 'Ball',
          'description': 'B is for Ball',
        },
        {
          'letter': 'C',
          'image': 'assets/cat.png',
          'word': 'Cat',
          'description': 'C is for Cat',
        },
        {
          'letter': 'D',
          'image': 'assets/dog.png',
          'word': 'Dog',
          'description': 'D is for Dog',
        },
        {
```

```
          'letter': 'E',
          'image': 'assets/elephant.png',
          'word': 'Elephant',
          'description': 'E is for
Elephant',
        },
        {
          'letter': 'F',
          'image': 'assets/frog.png',
          'word': 'Frog',
          'description': 'F is for Frog',
        },
      ];
    } else if
(widget.lesson.title.contains('Numbers
1-5')) {
      return [
        {
          'number': '1',
          'image': 'assets/apple.png',
          'word': 'One',
          'description': 'This is the number
1',
        },
        {
          'number': '2',
          'image': 'assets/apple.png',
          'word': 'Two',
          'description': 'This is the number
2',
        },
        {
          'number': '3',
```

```
          'image': 'assets/apple.png',
          'word': 'Three',
          'description': 'This is the number
3',
        },
        {
          'number': '4',
          'image': 'assets/apple.png',
          'word': 'Four',
          'description': 'This is the number
4',
        },
        {
          'number': '5',
          'image': 'assets/apple.png',
          'word': 'Five',
          'description': 'This is the number
5',
        },
      ];
    } else {
      // Default content for other lessons
      return [
        {
          'title': 'Introduction',
          'image': 'assets/apple.png',
          'description': 'Welcome to
${widget.lesson.title}',
        },
        {
          'title': 'Lesson Content',
          'image': 'assets/apple.png',
          'description': 'This is the main
```

```
content of the lesson',
        },
        {
          'title': 'Practice',
          'image': 'assets/apple.png',
          'description': 'Let\'s practice
what we learned',
        },
        {
          'title': 'Summary',
          'image': 'assets/apple.png',
          'description': 'Great job
completing this lesson!',
        },
      ];
    }
  }

  @override
  void dispose() {
    _pageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    final lessonContent =
_getLessonContent();

    return Scaffold(
      appBar: AppBar(
        title: Text(widget.lesson.title),
        backgroundColor: Colors.blue,
```

```
        foregroundColor: Colors.white,
      ),
      body: Column(
        children: [
          LinearProgressIndicator(
            value: _currentPage /
(lessonContent.length - 1),
            backgroundColor:
Colors.grey.shade200,
            valueColor: const
AlwaysStoppedAnimation<Color>(Colors.blue),
            minHeight: 10,
          ),
          Expanded(
            child: PageView.builder(
              controller: _pageController,
              itemCount:
lessonContent.length,
              onPageChanged: (index) {
                setState(() {
                  _currentPage = index;
                });
              },
              itemBuilder: (context, index)
{
                final content =
lessonContent[index];
                return
_buildLessonPage(content);
              },
            ),
          ),
          Padding(
```

```
            padding: const
EdgeInsets.all(16.0),
            child: Row(
              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
              children: [
                if (_currentPage > 0)
                  ElevatedButton.icon(
                    onPressed: () {

_pageController.previousPage(
                        duration: const
Duration(milliseconds: 300),
                        curve:
Curves.easeInOut,
                      );
                    },
                    icon: const
Icon(Icons.arrow_back),
                    label: const
Text('Previous'),
                    style:
ElevatedButton.styleFrom(
                      backgroundColor:
Colors.grey.shade300,
                      foregroundColor:
Colors.black,
                    ),
                  )
                else
                  const SizedBox(width:
100),
                Text(
```

```
                    '${_currentPage +
1}/${lessonContent.length}',
                    style: const TextStyle(
                      fontSize: 16,
                      fontWeight:
FontWeight.bold,
                    ),
                  ),
                  if (_currentPage <
lessonContent.length - 1)
                    ElevatedButton.icon(
                      onPressed: () {

_pageController.nextPage(
                          duration: const
Duration(milliseconds: 300),
                          curve:
Curves.easeInOut,
                        );
                      },
                      icon: const
Icon(Icons.arrow_forward),
                      label: const
Text('Next'),
                    )
                  else
                    ElevatedButton.icon(
                      onPressed: () {

Navigator.pop(context);

ScaffoldMessenger.of(context).showSnackBar(
                          const SnackBar(
```

```
                            content:
Text('Lesson completed! Great job!'),
                            backgroundColor:
Colors.green,
                          ),
                        );
                      },
                      icon: const
Icon(Icons.check),
                      label: const
Text('Complete'),
                      style:
ElevatedButton.styleFrom(
                        backgroundColor:
Colors.green,
                      ),
                    ),
                ],
              ),
            ),
          ],
        ),
      );
    }

  Widget _buildLessonPage(Map<String,
dynamic> content) {
    if (content.containsKey('letter')) {
      // Alphabet lesson
      return Center(
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
```

```
          children: [
            Text(
              content['letter'],
              style: const TextStyle(
                fontSize: 120,
                fontWeight: FontWeight.bold,
                color: Colors.blue,
              ),
            ),
            const SizedBox(height: 20),
            Image.asset(
              content['image'],
              height: 150,
              width: 150,
            ),
            const SizedBox(height: 20),
            Text(
              content['word'],
              style: const TextStyle(
                fontSize: 32,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 10),
            Text(
              content['description'],
              style: const TextStyle(
                fontSize: 20,
              ),
            ),
            const SizedBox(height: 30),
            ElevatedButton.icon(
              onPressed: () {
```

```
                _speak(content['word']);
              },
              icon: const
Icon(Icons.volume_up),
              label: const Text('Listen'),
              style:
ElevatedButton.styleFrom(
                padding:
                    const
EdgeInsets.symmetric(horizontal: 30,
vertical: 15),
              ),
            ),
          ],
        ),
      );
    } else if
(content.containsKey('number')) {
      // Number lesson
      return Center(
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            Text(
              content['number'],
              style: const TextStyle(
                fontSize: 120,
                fontWeight: FontWeight.bold,
                color: Colors.orange,
              ),
            ),
            const SizedBox(height: 20),
```

```
            Image.asset(
              content['image'],
              height: 150,
              width: 150,
            ),
            const SizedBox(height: 20),
            Text(
              content['word'],
              style: const TextStyle(
                fontSize: 32,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 10),
            Text(
              content['description'],
              style: const TextStyle(
                fontSize: 20,
              ),
            ),
            const SizedBox(height: 30),
            Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                ElevatedButton.icon(
                  onPressed: () {

_speak(content['number']);
                  },
                  icon: const
Icon(Icons.volume_up),
                  label: const
```

```
Text('Listen'),
                  style:
ElevatedButton.styleFrom(
                    padding: const
EdgeInsets.symmetric(
                        horizontal: 20,
vertical: 15),
                  ),
                ),
              ],
            ),
          ],
        ),
      );
    } else {
      // Generic lesson
      return Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            Text(
              content['title'],
              style: const TextStyle(
                fontSize: 28,
                fontWeight: FontWeight.bold,
                color: Colors.blue,
              ),
            ),
            const SizedBox(height: 30),
            Image.asset(
              content['image'],
```

```
              height: 200,
              width: 200,
            ),
            const SizedBox(height: 30),
            Text(
              content['description'],
              textAlign: TextAlign.center,
              style: const TextStyle(
                fontSize: 20,
              ),
            ),
          ],
        ),
      );
    }
  }
}
```

home_screen.dart

```
import 'package:flutter/material.dart';
import 'tabs/lessons_tab.dart';
import 'tabs/quiz_tab.dart';
import 'tabs/leaderboard_tab.dart';
import 'tabs/profile_tab.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({super.key});

  @override
  State<HomeScreen> createState() =>
_HomeScreenState();
}
```

```
class _HomeScreenState extends
State<HomeScreen> {
  int _currentIndex = 0;
  final List<Widget> _tabs = [
    const LessonsTab(),
    const QuizTab(),
    const LeaderboardTab(),
    const ProfileTab(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: _tabs[_currentIndex],
      bottomNavigationBar:
BottomNavigationBar(
        currentIndex: _currentIndex,
        onTap: (index) {
          setState(() {
            _currentIndex = index;
          });
        },
        type: BottomNavigationBarType.fixed,
        backgroundColor: Colors.white,
        selectedItemColor: Colors.blue,
        unselectedItemColor: Colors.grey,
        selectedLabelStyle: const
TextStyle(fontWeight: FontWeight.bold),
        items: const [
          BottomNavigationBarItem(
            icon: Icon(Icons.book),
            label: 'Lessons',
          ),
```

```
          BottomNavigationBarItem(
            icon: Icon(Icons.quiz),
            label: 'Quiz',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.leaderboard),
            label: 'Leaders',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.person),
            label: 'Profile',
          ),
        ],
      ),
    );
  }
}
```

leadboard_tab.dart

```
import 'package:flutter/material.dart';
import
'../../models/leaderboard_entry.dart';

class LeaderboardTab extends StatelessWidget
{
  const LeaderboardTab({super.key});

  @override
  Widget build(BuildContext context) {
    // Sample leaderboard data
    final List<LeaderboardEntry> entries = [
      LeaderboardEntry(
```

```
        id: '1',
        name: 'Prabhat',
        score: 950,
        quizzesTaken: 12,
        avatarUrl: 'prabhat.jpeg',
      ),
      LeaderboardEntry(
        id: '2',
        name: 'Abhishek',
        score: 920,
        quizzesTaken: 10,
        avatarUrl: 'abhishek.jpeg',
      ),
      LeaderboardEntry(
        id: '3',
        name: 'Deepak',
        score: 880,
        quizzesTaken: 11,
        avatarUrl: 'deepak.jpeg',
      ),
      LeaderboardEntry(
        id: '4',
        name: 'Aditi',
        score: 850,
        quizzesTaken: 9,
        avatarUrl: 'aditi.jpeg',
      ),
      LeaderboardEntry(
        id: '5',
        name: 'Aditya',
        score: 820,
        quizzesTaken: 10,
        avatarUrl: 'aditya.jpeg',
```

```
      ),
      LeaderboardEntry(
        id: '6',
        name: 'Aryan',
        score: 790,
        quizzesTaken: 8,
        avatarUrl: 'prabhat.jpeg',
      ),
      LeaderboardEntry(
        id: '7',
        name: 'Kshitija',
        score: 760,
        quizzesTaken: 9,
        avatarUrl: 'prabhat.jpeg',
      ),
      LeaderboardEntry(
        id: '8',
        name: 'Tejas',
        score: 730,
        quizzesTaken: 7,
        avatarUrl: 'prabhat.jpeg',
      ),
      LeaderboardEntry(
        id: '9',
        name: 'Riddhi',
        score: 700,
        quizzesTaken: 8,
        avatarUrl: 'prabhat.jpeg',
      ),
      LeaderboardEntry(
        id: '10',
        name: 'Anjali',
        score: 670,
```

```
        quizzesTaken: 7,
        avatarUrl: 'prabhat.jpeg',
      ),
    ];

    return Scaffold(
      appBar: AppBar(
        title: const Text('Leaderboard'),
        backgroundColor: Colors.green,
        foregroundColor: Colors.white,
      ),
      body: Column(
        children: [

_buildTopThree(entries.take(3).toList()),
          const SizedBox(height: 20),
          const Padding(
            padding:
EdgeInsets.symmetric(horizontal: 16.0),
            child: Text(
              'Top Players',
              style: TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
                color: Colors.green,
              ),
            ),
          ),
          const SizedBox(height: 10),
          Expanded(
            child: ListView.builder(
              padding: const
EdgeInsets.symmetric(horizontal: 16.0),
```

```
              itemCount: entries.length,
              itemBuilder: (context, index)
{
                return
_buildLeaderboardItem(entries[index], index
+ 1);
              },
            ),
          ),
        ],
      ),
    );
  }

  Widget
_buildTopThree(List<LeaderboardEntry>
topThree) {
    return Container(
      padding: const
EdgeInsets.symmetric(vertical: 20),
      decoration: BoxDecoration(
        color: Colors.green.shade50,
        borderRadius: const
BorderRadius.only(
          bottomLeft: Radius.circular(30),
          bottomRight: Radius.circular(30),
        ),
      ),
      child: Row(
        mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
        crossAxisAlignment:
CrossAxisAlignment.end,
```

```
        children: [
          if (topThree.length > 1)
            _buildTopPlayer(topThree[1], 2,
Colors.grey.shade400, 80),
          if (topThree.isNotEmpty)
            _buildTopPlayer(topThree[0], 1,
Colors.amber, 100),
          if (topThree.length > 2)
            _buildTopPlayer(topThree[2], 3,
Colors.brown.shade300, 60),
        ],
      ),
    );
  }

  Widget _buildTopPlayer(
      LeaderboardEntry entry, int position,
Color color, double height) {
    return Column(
      children: [
        Container(
          padding: const EdgeInsets.all(8),
          decoration: BoxDecoration(
            color: color,
            shape: BoxShape.circle,
          ),
          child: Text(
            '$position',
            style: const TextStyle(
              color: Colors.white,
              fontWeight: FontWeight.bold,
            ),
          ),
```

```
        ),
        const SizedBox(height: 8),
        CircleAvatar(
          radius: position == 1 ? 35 : 25,
          backgroundImage:
AssetImage(entry.avatarUrl),
        ),
        const SizedBox(height: 8),
        Text(
          entry.name,
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: position == 1 ? 18 :
16,
          ),
        ),
        const SizedBox(height: 4),
        Container(
          padding: const
EdgeInsets.symmetric(horizontal: 12,
vertical: 4),
          decoration: BoxDecoration(
            color: Colors.green,
            borderRadius:
BorderRadius.circular(20),
          ),
          child: Text(
            '${entry.score} pts',
            style: const TextStyle(
              color: Colors.white,
              fontWeight: FontWeight.bold,
            ),
          ),
```

```
        ),
      ],
    );
  }

  Widget
_buildLeaderboardItem(LeaderboardEntry
entry, int position) {
    return Card(
      margin: const EdgeInsets.only(bottom:
10),
      elevation: 2,
      shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
      child: Padding(
        padding: const EdgeInsets.all(12.0),
        child: Row(
          children: [
            Container(
              width: 30,
              height: 30,
              alignment: Alignment.center,
              decoration: BoxDecoration(
                color:
_getPositionColor(position),
                shape: BoxShape.circle,
              ),
              child: Text(
                '$position',
                style: const TextStyle(
                  color: Colors.white,
                  fontWeight:
```

```
FontWeight.bold,
                  ),
                ),
              ),
              const SizedBox(width: 12),
              CircleAvatar(
                backgroundImage:
AssetImage(entry.avatarUrl),
                radius: 20,
              ),
              const SizedBox(width: 12),
              Expanded(
                child: Column(
                  crossAxisAlignment:
CrossAxisAlignment.start,
                  children: [
                    Text(
                      entry.name,
                      style: const TextStyle(
                        fontWeight:
FontWeight.bold,
                        fontSize: 16,
                      ),
                    ),
                    Text(
                      '${entry.quizzesTaken}
quizzes completed',
                      style: TextStyle(
                        fontSize: 12,
                        color:
Colors.grey.shade600,
                      ),
                    ),
```

```
                ],
              ),
            ),
            Container(
              padding: const
EdgeInsets.symmetric(horizontal: 12,
vertical: 6),
              decoration: BoxDecoration(
                color:
Colors.green.shade100,
                borderRadius:
BorderRadius.circular(20),
              ),
              child: Text(
                '${entry.score} pts',
                style: TextStyle(
                  color:
Colors.green.shade800,
                  fontWeight:
FontWeight.bold,
                ),
              ),
            ),
          ],
        ),
      ),
    );
  }

  Color _getPositionColor(int position) {
    switch (position) {
      case 1:
        return Colors.amber;
```

```
      case 2:
        return Colors.grey.shade400;
      case 3:
        return Colors.brown.shade300;
      default:
        return Colors.green;
    }
  }
}
```

lessons_tab.dart

```
import 'package:flutter/material.dart';
import '../../models/lesson.dart';
import '../lesson_detail_screen.dart';

class LessonsTab extends StatelessWidget {
  const LessonsTab({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Learn &
Practice'),
        backgroundColor: Colors.blue,
        foregroundColor: Colors.white,
      ),
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
```

```
            const Text(
              'Choose a Category',
              style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold,
                color: Colors.blue,
              ),
            ),
            const SizedBox(height: 20),
            _buildCategoryCard(
              context,
              'Alphabets',
              'Learn all 26 letters',
              Colors.purple.shade100,
              Icons.text_fields,
              [
                Lesson(id: '1', title: 'A-F
Letters', description: 'Learn the first 6
letters', icon: Icons.abc),
                Lesson(id: '2', title: 'G-L
Letters', description: 'Learn the next 6
letters', icon: Icons.abc),
                Lesson(id: '3', title: 'M-R
Letters', description: 'Learn the middle 6
letters', icon: Icons.abc),
                Lesson(id: '4', title: 'S-Z
Letters', description: 'Learn the last 8
letters', icon: Icons.abc),
              ],
            ),
            const SizedBox(height: 16),
            _buildCategoryCard(
              context,
```

```
              'Numbers',
              'Learn numbers 1-20',
              Colors.blue.shade100,
              Icons.numbers,
              [
                Lesson(id: '5', title:
'Numbers 1-5', description: 'Learn to count
from 1 to 5', icon: Icons.filter_1),
                Lesson(id: '6', title:
'Numbers 6-10', description: 'Learn to count
from 6 to 10', icon: Icons.filter_5),
                Lesson(id: '7', title:
'Numbers 11-15', description: 'Learn to
count from 11 to 15', icon: Icons.filter_9),
                Lesson(id: '8', title:
'Numbers 16-20', description: 'Learn to
count from 16 to 20', icon:
Icons.filter_9_plus),
              ],
            ),
            const SizedBox(height: 16),
            _buildCategoryCard(
              context,
              'Shapes',
              'Learn basic shapes',
              Colors.green.shade100,
              Icons.category,
              [
                Lesson(id: '9', title:
'Basic Shapes', description: 'Circle,
Square, Triangle', icon: Icons.category),
                Lesson(id: '10', title:
'More Shapes', description: 'Rectangle,
```

```
Oval, Diamond', icon: Icons.category),
                ],
              ),
              const SizedBox(height: 16),
              _buildCategoryCard(
                context,
                'Colors',
                'Learn primary colors',
                Colors.orange.shade100,
                Icons.color_lens,
                [
                  Lesson(id: '11', title:
'Primary Colors', description: 'Red, Blue,
Yellow', icon: Icons.color_lens),
                  Lesson(id: '12', title:
'Secondary Colors', description: 'Green,
Orange, Purple', icon: Icons.color_lens),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }

  Widget _buildCategoryCard(
    BuildContext context,
    String title,
    String subtitle,
    Color color,
    IconData icon,
    List<Lesson> lessons,
  ) {
```

```
    return GestureDetector(
      onTap: () {
        _showLessonsBottomSheet(context,
title, lessons);
      },
      child: Card(
        elevation: 4,
        shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(16)),
        child: Container(
          decoration: BoxDecoration(
            borderRadius:
BorderRadius.circular(16),
            color: color,
          ),
          padding: const EdgeInsets.all(16),
          child: Row(
            children: [
              Icon(icon, size: 50, color:
Colors.blue.shade800),
              const SizedBox(width: 16),
              Expanded(
                child: Column(
                  crossAxisAlignment:
CrossAxisAlignment.start,
                  children: [
                    Text(
                      title,
                      style: const
TextStyle(
                        fontSize: 22,
                        fontWeight:
```

```
FontWeight.bold,
                      ),
                    ),
                    Text(
                      subtitle,
                      style: TextStyle(
                        fontSize: 16,
                        color:
Colors.grey.shade700,
                      ),
                    ),
                  ],
                ),
              ),
              const
Icon(Icons.arrow_forward_ios, color:
Colors.blue),
            ],
          ),
        ),
      ),
    );
  }

  void _showLessonsBottomSheet(BuildContext
context, String category, List<Lesson>
lessons) {
    showModalBottomSheet(
      context: context,
      isScrollControlled: true,
      shape: const RoundedRectangleBorder(
        borderRadius:
BorderRadius.vertical(top:
```

```
Radius.circular(20)),
      ),
      builder: (context) {
        return Container(
          padding: const EdgeInsets.all(20),
          height:
MediaQuery.of(context).size.height * 0.7,
          child: Column(
            crossAxisAlignment:
CrossAxisAlignment.start,
            children: [
              Row(
                mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                children: [
                  Text(
                    '$category Lessons',
                    style: const TextStyle(
                      fontSize: 24,
                      fontWeight:
FontWeight.bold,
                      color: Colors.blue,
                    ),
                  ),
                  IconButton(
                    icon: const
Icon(Icons.close),
                    onPressed: () =>
Navigator.pop(context),
                  ),
                ],
              ),
              const SizedBox(height: 20),
```

```
              Expanded(
                child: ListView.builder(
                  itemCount: lessons.length,
                  itemBuilder: (context,
index) {
                    return Card(
                      margin: const
EdgeInsets.only(bottom: 16),
                      elevation: 2,
                      shape:
RoundedRectangleBorder(
                        borderRadius:
BorderRadius.circular(12),
                      ),
                      child: ListTile(
                        contentPadding:
const EdgeInsets.symmetric(
                          horizontal: 16,
                          vertical: 8,
                        ),
                        leading:
CircleAvatar(
                          backgroundColor:
Colors.blue.shade100,
                          child:
Icon(lessons[index].icon, color:
Colors.blue),
                        ),
                        title: Text(

lessons[index].title,
                          style: const
TextStyle(
```

```
                                fontWeight:
FontWeight.bold,
                                fontSize: 18,
                              ),
                            ),
                            subtitle:
Text(lessons[index].description),
                            trailing: const
Icon(Icons.arrow_forward_ios, size: 16),
                            onTap: () {

Navigator.pop(context);
                              Navigator.push(
                                context,

MaterialPageRoute(
                                  builder:
(context) => LessonDetailScreen(lesson:
lessons[index]),
                                ),
                              );
                            },
                          ),
                        );
                      },
                    ),
                  ),
                ],
              ),
            );
          },
        );
      }
```

```
}
```

profile.dart

```
import 'package:flutter/material.dart';
import '../../models/achievement.dart';

class ProfileTab extends StatefulWidget {
  const ProfileTab({super.key});

  @override
  State<ProfileTab> createState() =>
_ProfileTabState();
}

class _ProfileTabState extends
State<ProfileTab>
    with SingleTickerProviderStateMixin {
  late TabController _tabController;
  final TextEditingController
_nameController =
      TextEditingController(text:
'Prabhat');
  final TextEditingController _ageController
= TextEditingController(text: '7');
  bool _isEditing = false;

  @override
  void initState() {
    super.initState();
    _tabController = TabController(length:
3, vsync: this);
  }
```

```
  @override
  void dispose() {
    _tabController.dispose();
    _nameController.dispose();
    _ageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('My Profile'),
        backgroundColor: Colors.purple,
        foregroundColor: Colors.white,
        actions: [
          IconButton(
            icon: Icon(_isEditing ?
Icons.save : Icons.edit),
            onPressed: () {
              setState(() {
                _isEditing = !_isEditing;
              });
              if (!_isEditing) {

ScaffoldMessenger.of(context).showSnackBar(
                  const SnackBar(
                    content: Text('Profile
updated successfully!'),
                    backgroundColor:
Colors.green,
                  ),
```

```
                );
              }
            },
          ),
        ],
        bottom: TabBar(
          controller: _tabController,
          indicatorColor: Colors.white,
          tabs: const [
            Tab(text: 'Profile'),
            Tab(text: 'Progress'),
            Tab(text: 'Achievements'),
          ],
        ),
      ),
      body: TabBarView(
        controller: _tabController,
        children: [
          _buildProfileTab(),
          _buildProgressTab(),
          _buildAchievementsTab(),
        ],
      ),
    );
  }

  Widget _buildProfileTab() {
    return SingleChildScrollView(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.center,
        children: [
```

```
          const SizedBox(height: 20),
          Stack(
            children: [
              CircleAvatar(
                radius: 60,
                backgroundImage:
AssetImage('prabhat.jpeg'),
              ),
              if (_isEditing)
                Positioned(
                  bottom: 0,
                  right: 0,
                  child: Container(
                    padding: const
EdgeInsets.all(4),
                    decoration: const
BoxDecoration(
                      color: Colors.purple,
                      shape:
BoxShape.circle,
                    ),
                    child: const Icon(
                      Icons.camera_alt,
                      color: Colors.white,
                      size: 20,
                    ),
                  ),
                ),
            ],
          ),
          const SizedBox(height: 20),
          _isEditing
              ? TextField(
```

```
                    controller:
_nameController,
                    decoration: const
InputDecoration(
                      labelText: 'Name',
                      border:
OutlineInputBorder(),
                    ),
                    textAlign:
TextAlign.center,
                    style: const TextStyle(
                      fontSize: 24,
                      fontWeight:
FontWeight.bold,
                    ),
                  )
                : Text(
                    _nameController.text,
                    style: const TextStyle(
                      fontSize: 24,
                      fontWeight:
FontWeight.bold,
                    ),
                  ),
            const SizedBox(height: 10),
            _isEditing
                ? TextField(
                    controller:
_ageController,
                    decoration: const
InputDecoration(
                      labelText: 'Age',
                      border:
```

```
OutlineInputBorder(),
                  ),
                  textAlign:
TextAlign.center,
                  keyboardType:
TextInputType.number,
                )
              : Text(
                  '${_ageController.text}
years old',
                  style: TextStyle(
                    fontSize: 16,
                    color:
Colors.grey.shade600,
                  ),
                ),
          const SizedBox(height: 30),
          const Divider(),
          const SizedBox(height: 20),
          _buildInfoCard('Total Points',
'750', Icons.stars, Colors.amber),
          const SizedBox(height: 16),
          _buildInfoCard('Quizzes
Completed', '15', Icons.quiz,
Colors.orange),
          const SizedBox(height: 16),
          _buildInfoCard('Lessons
Completed', '22', Icons.book, Colors.blue),
          const SizedBox(height: 16),
          _buildInfoCard(
              'Achievements', '8/20',
Icons.emoji_events, Colors.purple),
        ],
```

```
      ),
    );
  }

  Widget _buildProgressTab() {
    return SingleChildScrollView(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          const Text(
            'Learning Progress',
            style: TextStyle(
              fontSize: 22,
              fontWeight: FontWeight.bold,
              color: Colors.purple,
            ),
          ),
          const SizedBox(height: 20),
          _buildProgressCard('Alphabets',
0.85, '85%', Colors.blue),
          const SizedBox(height: 16),
          _buildProgressCard('Numbers', 0.7,
'70%', Colors.orange),
          const SizedBox(height: 16),
          _buildProgressCard('Shapes', 0.6,
'60%', Colors.green),
          const SizedBox(height: 16),
          _buildProgressCard('Colors', 0.9,
'90%', Colors.red),
          const SizedBox(height: 30),
          const Text(
```

```
            'Recent Activities',
            style: TextStyle(
              fontSize: 22,
              fontWeight: FontWeight.bold,
              color: Colors.purple,
            ),
          ),
          const SizedBox(height: 20),
          _buildActivityItem(
            'Completed "Numbers 1-5"
lesson',
            '2 hours ago',
            Icons.book,
            Colors.blue,
          ),
          _buildActivityItem(
            'Scored 90% in Alphabet Quiz',
            'Yesterday',
            Icons.quiz,
            Colors.orange,
          ),
          _buildActivityItem(
            'Earned "Number Master" badge',
            '2 days ago',
            Icons.emoji_events,
            Colors.purple,
          ),
          _buildActivityItem(
            'Completed "Colors" lesson',
            '3 days ago',
            Icons.book,
            Colors.red,
          ),
```

```
        ],
      ),
    );
  }

  Widget _buildAchievementsTab() {
    final List<Achievement> achievements = [
      Achievement(
        id: '1',
        title: 'First Steps',
        description: 'Complete your first
lesson',
        icon: Icons.emoji_events,
        color: Colors.blue,
        isUnlocked: true,
      ),
      Achievement(
        id: '2',
        title: 'Quiz Whiz',
        description: 'Score 100% on any
quiz',
        icon: Icons.emoji_events,
        color: Colors.orange,
        isUnlocked: true,
      ),
      Achievement(
        id: '3',
        title: 'Alphabet Master',
        description: 'Complete all alphabet
lessons',
        icon: Icons.emoji_events,
        color: Colors.purple,
        isUnlocked: true,
```

```
      ),
      Achievement(
        id: '4',
        title: 'Number Genius',
        description: 'Complete all number
lessons',
        icon: Icons.emoji_events,
        color: Colors.green,
        isUnlocked: false,
      ),
      Achievement(
        id: '5',
        title: 'Perfect Streak',
        description: 'Complete 5 quizzes
with 100% score',
        icon: Icons.emoji_events,
        color: Colors.red,
        isUnlocked: false,
      ),
      Achievement(
        id: '6',
        title: 'Learning Explorer',
        description: 'Complete at least one
lesson in each category',
        icon: Icons.emoji_events,
        color: Colors.teal,
        isUnlocked: true,
      ),
      Achievement(
        id: '7',
        title: 'Quiz Champion',
        description: 'Score in the top 3 of
the leaderboard',
```

```
        icon: Icons.emoji_events,
        color: Colors.amber,
        isUnlocked: true,
      ),
      Achievement(
        id: '8',
        title: 'Dedicated Learner',
        description: 'Complete 20 lessons',
        icon: Icons.emoji_events,
        color: Colors.indigo,
        isUnlocked: false,
      ),
    ];

    return SingleChildScrollView(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          const Text(
            'My Achievements',
            style: TextStyle(
              fontSize: 22,
              fontWeight: FontWeight.bold,
              color: Colors.purple,
            ),
          ),
          const SizedBox(height: 8),
          Text(
            'You have unlocked
${achievements.where((a) =>
a.isUnlocked).length}/${achievements.length}
```

```
achievements',
              style: TextStyle(
                fontSize: 16,
                color: Colors.grey.shade600,
              ),
            ),
            const SizedBox(height: 20),
            GridView.builder(
              shrinkWrap: true,
              physics: const
NeverScrollableScrollPhysics(),
              gridDelegate: const
SliverGridDelegateWithFixedCrossAxisCount(
                crossAxisCount: 2,
                childAspectRatio: 0.85,
                crossAxisSpacing: 8,
                mainAxisSpacing: 8,
              ),
              itemCount: achievements.length,
              itemBuilder: (context, index) {
                return
_buildAchievementCard(achievements[index]);
              },
            ),
          ],
        ),
      );
  }

  Widget _buildInfoCard(
      String title, String value, IconData
icon, Color color) {
    return Card(
```

```
      elevation: 3,
      shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Row(
          children: [
            CircleAvatar(
              backgroundColor:
color.withOpacity(0.2),
              radius: 25,
              child: Icon(icon, color:
color),
            ),
            const SizedBox(width: 16),
            Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Text(
                  title,
                  style: TextStyle(
                    fontSize: 16,
                    color:
Colors.grey.shade600,
                  ),
                ),
                const SizedBox(height: 4),
                Text(
                  value,
                  style: const TextStyle(
                    fontSize: 20,
```

```
                    fontWeight:
FontWeight.bold,
                  ),
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }

  Widget _buildProgressCard(
      String title, double progress, String
percentage, Color color) {
    return Card(
      elevation: 3,
      shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
            Row(
              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
              children: [
                Text(
                  title,
                  style: const TextStyle(
```

```
                    fontSize: 18,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
                Text(
                  percentage,
                  style: TextStyle(
                    fontSize: 18,
                    fontWeight:
FontWeight.bold,
                    color: color,
                  ),
                ),
              ],
            ),
            const SizedBox(height: 12),
            LinearProgressIndicator(
              value: progress,
              backgroundColor:
Colors.grey.shade200,
              valueColor:
AlwaysStoppedAnimation<Color>(color),
              minHeight: 10,
              borderRadius:
BorderRadius.circular(5),
            ),
          ],
        ),
      ),
    );
  }
```

```
  Widget _buildActivityItem(
      String title, String time, IconData
icon, Color color) {
    return Padding(
      padding: const EdgeInsets.only(bottom:
16.0),
      child: Row(
        children: [
          CircleAvatar(
            backgroundColor:
color.withOpacity(0.2),
            radius: 20,
            child: Icon(icon, color: color,
size: 20),
          ),
          const SizedBox(width: 16),
          Expanded(
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Text(
                  title,
                  style: const TextStyle(
                    fontSize: 16,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
                Text(
                  time,
                  style: TextStyle(
                    fontSize: 14,
```

```
                    color:
Colors.grey.shade600,
                  ),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }

  Widget _buildAchievementCard(Achievement
achievement) {
    return Card(
      elevation: 3,
      shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
      child: Container(
        decoration: BoxDecoration(
          borderRadius:
BorderRadius.circular(12),
          color: achievement.isUnlocked ?
Colors.white : Colors.grey.shade200,
        ),
        padding: const EdgeInsets.all(8),
        child: LayoutBuilder(builder:
(context, constraints) {
          return Column(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
```

```
              CircleAvatar(
                backgroundColor:
achievement.isUnlocked
                    ?
achievement.color.withOpacity(0.2)
                    : Colors.grey.shade300,
                radius: 22,
                child: Icon(
                  achievement.icon,
                  color:
                      achievement.isUnlocked
? achievement.color : Colors.grey,
                  size: 22,
                ),
              ),
              const SizedBox(height: 6),
              SizedBox(
                width: constraints.maxWidth,
                child: Text(
                  achievement.title,
                  textAlign:
TextAlign.center,
                  maxLines: 1,
                  overflow:
TextOverflow.ellipsis,
                  style: TextStyle(
                    fontSize: 13,
                    fontWeight:
FontWeight.bold,
                    color:
achievement.isUnlocked ? Colors.black :
Colors.grey,
                  ),
```

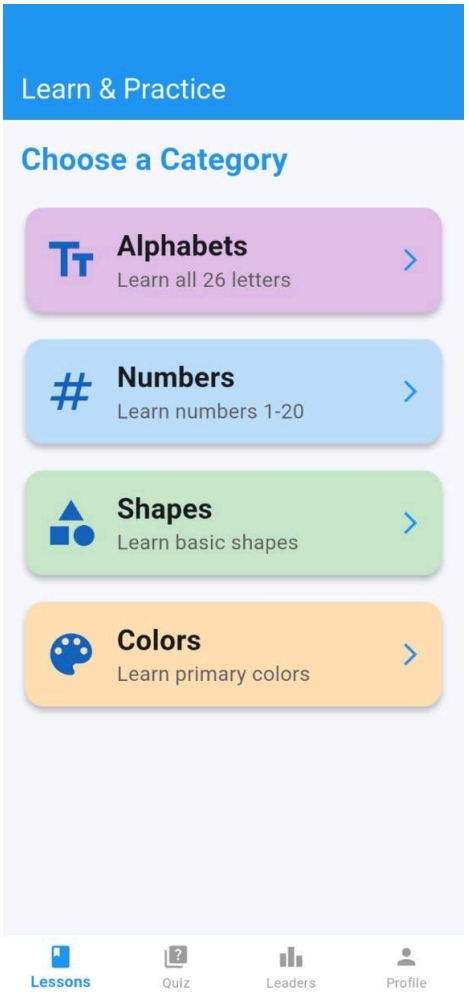
```
                  ),
                ),
                const SizedBox(height: 4),
                SizedBox(
                  width: constraints.maxWidth,
                  height: 28,
                  child: Text(
                    achievement.description,
                    textAlign:
TextAlign.center,
                    maxLines: 2,
                    overflow:
TextOverflow.ellipsis,
                    style: TextStyle(
                      fontSize: 10,
                      height: 1.2,
                      color:
Colors.grey.shade600,
                    ),
                  ),
                ),
                const SizedBox(height: 4),
                Container(
                  padding: const
EdgeInsets.symmetric(horizontal: 6,
vertical: 2),
                  decoration: BoxDecoration(
                    color:
achievement.isUnlocked
                        ?
achievement.color.withOpacity(0.2)
                        :
Colors.grey.shade300,
```

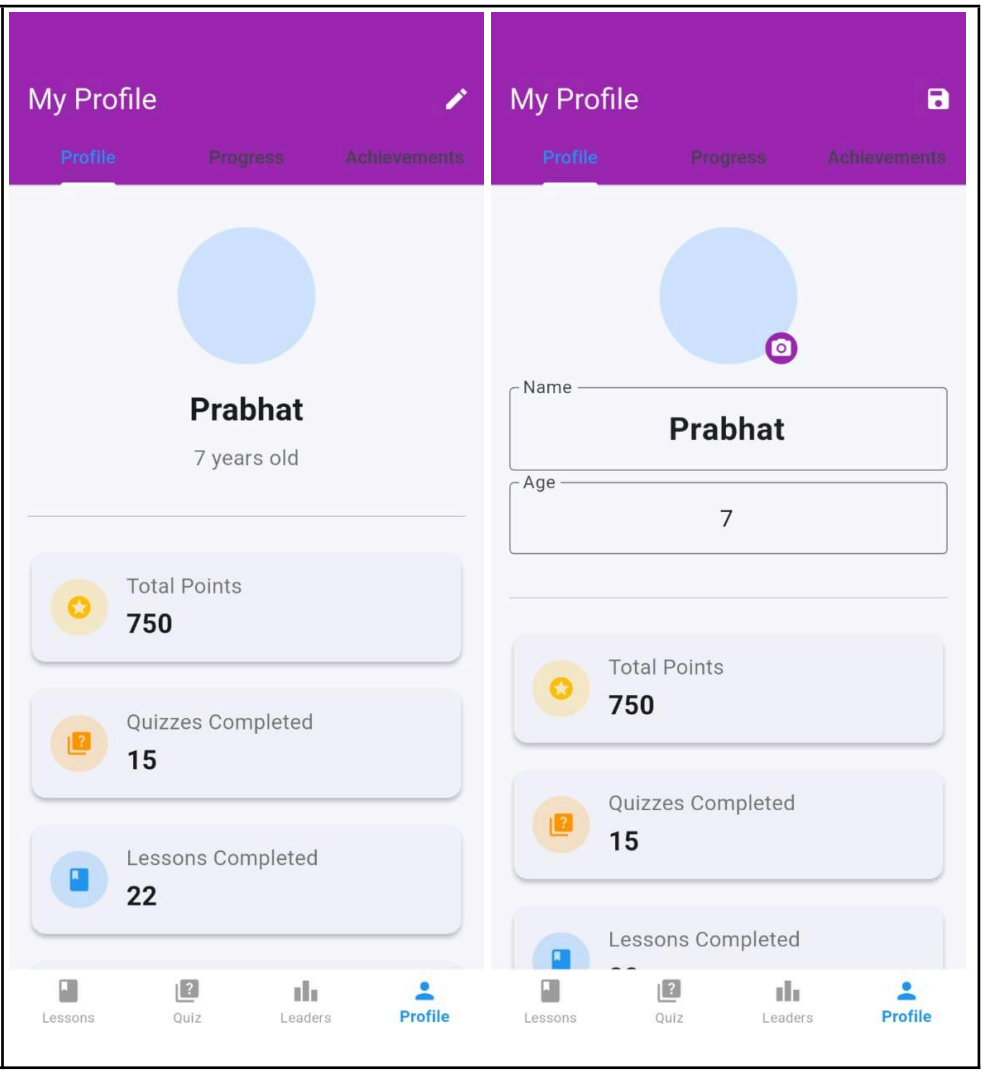
```
                    borderRadius:
BorderRadius.circular(10),
                  ),
                  child: Text(
                    achievement.isUnlocked ?
'Unlocked' : 'Locked',
                    style: TextStyle(
                      fontSize: 9,
                      fontWeight:
FontWeight.bold,
                      color:
achievement.isUnlocked
                          ? achievement.color
                          : Colors.grey,
                    ),
                  ),
                ),
              ],
            );
          }),
      ),
    );
  }
}
```

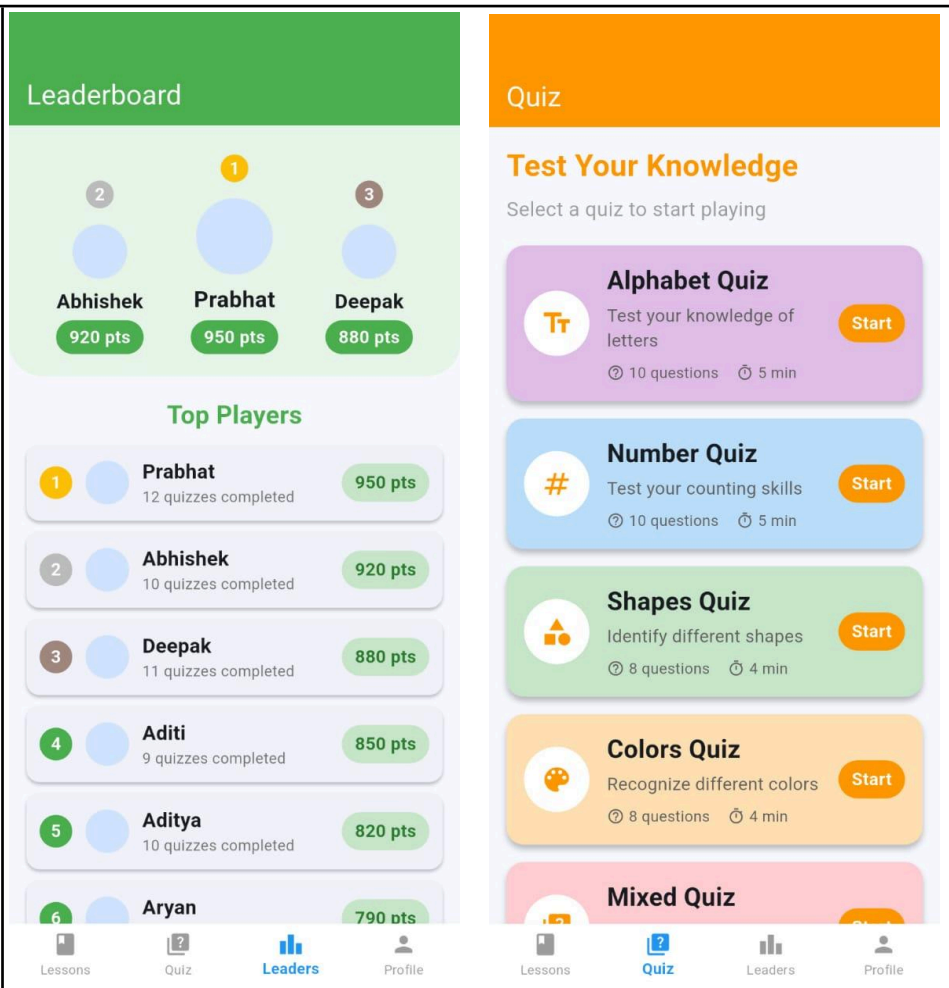
Screenshots

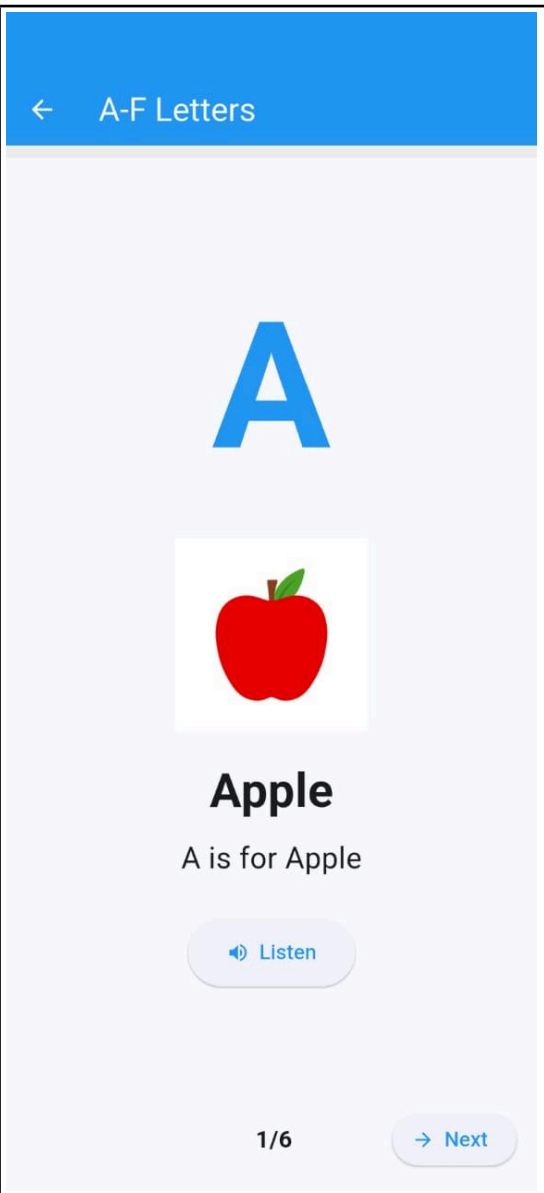
Kids Learning App

Learn Numbers & Alphabets




My Profile
Profile
Progress
Achievements
Prabhat
7 years old
Total Points
750
Quizzes Completed
15
Lessons Completed
22
Lessons
Quiz
Leaders
Profile
My Profile
Profile
Progress
Achievements
Name
Prabhat
Age
7
Total Points
750
Quizzes Completed
15
Lessons Completed
Lessons
Quiz
Leaders
Profile


Leaderboard
1
2
3
Abhishek
920 pts
Prabhat
950 pts
Deepak
880 pts
Top Players
1 Prabhat
12 quizzes completed
950 pts
2 Abhishek
10 quizzes completed
920 pts
3 Deepak
11 quizzes completed
880 pts
4 Aditi
9 quizzes completed
850 pts
5 Aditya
10 quizzes completed
820 pts
6 Aryan
790 pts
Lessons
Quiz
Leaders
Profile
Quiz
Test Your Knowledge
Select a quiz to start playing
Alphabet Quiz
Test your knowledge of letters
10 questions
5 min
Start
Number Quiz
Test your counting skills
10 questions
5 min
Start
Shapes Quiz
Identify different shapes
8 questions
4 min
Start
Colors Quiz
Recognize different colors
8 questions
4 min
Start
Mixed Quiz
Lessons
Quiz
Leaders
Profile



Question and Answers

Answer the following Questions:

1. How to create Upper Tabs in Flutter?
   To create upper tabs in Flutter**, we use the DefaultTabController, TabBar, and TabBarView widgets together

2. How did you use 60-30-10 rule in your application?
   In the E-Learning app, the 60-30-10 color rule was applied as follows:
   60%: Used for the app background and headers, giving a sense of trust and focus.
   30%: Applied to cards and content areas to differentiate sections.
   10% : Used for action buttons, progress indicators, and icons to draw attention.
   This color balance maintains a visually appealing and focused learning environment.

3. Which new elements did you use for creating UI components?
   To enhance the app's UI/UX, the following Flutter elements were used:

	TabBar & TabBarView: For top tab navigation. ExpansionTile: To display course modules and expandable lessons. LinearProgressIndicator: To show course completion progress. ListView.builder: For dynamically rendering course lists. Custom Widgets: Created reusable components like CourseCard and LessonTile. 4. In pubspec.yaml file, what dependencies need to be there? flutter_tts: ^4.2.2 5. What is the use of Splash Screen? The Splash Screen serves multiple purposes in the E-Learning app: Brand Introduction: It showcases the app's logo and tagline, reinforcing the platform's identity. Initial Load Buffer: It allows the app to initialize resources like user sessions or fetch data from local storage or the internet. Improved UX: Provides a smooth transition into the main content, preventing abrupt loading experiences.
Conclusion	Through this project, I have learned how to effectively implement both SideDrawer and Tabs Navigation to create a seamless user experience. I gained the 60-30-10 color rule's importance in maintaining a visually appealing UI. Additionally, I explored new UI components like ExpansionTile, WebView, Drawer, and TabBar, enhancing my ability to structure and organize content dynamically. Working with pubspec.yaml dependencies helped me integrate external functionalities such as file handling, web embedding, and URL launching. Finally, implementing expandable views and WebView strengthened my skills in creating interactive and informative sections within the app.