|  |  |
| --- | --- |
| **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** | Aditya Reddy |
| **UCID** | 2024510049 |
| **Class** | FY MCA |
| **Batch** | C |
| **Date of Submission** | 09-04-2025 |

|  |  |
| --- | --- |
| **Technology used** | **Flutter, Dart** |
| **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** | SplashScreen  import 'package:flutter/material.dart'; import 'dart:async'; import 'package:lottie/lottie.dart'; import 'home\_screen.dart';  class SplashScreen extends StatefulWidget {  const SplashScreen({super.key});   @override  \_SplashScreenState createState() => \_SplashScreenState(); }  class \_SplashScreenState extends State<SplashScreen> {  @override  void initState() {  super.initState();  Timer(const Duration(seconds: 3), () {  Navigator.*pushReplacement*(  context,  MaterialPageRoute(builder: (context) => HomeScreen()),  );  });  }   @override  Widget build(BuildContext context) {  return Scaffold(  backgroundColor: Colors.*indigoAccent*.shade200,  body: Center(  child: SingleChildScrollView(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  Lottie.*asset*(  'assets/learningAnimation.json',  width: 180,  height: 180,  fit: BoxFit.contain,  ),  const SizedBox(height: 25),  const Text(  "Launching Your Learning Journey!",  style: TextStyle(  fontSize: 22,  fontWeight: FontWeight.*w600*,  color: Colors.*white*,  letterSpacing: 1.1,  ),  textAlign: TextAlign.center,  ),  const SizedBox(height: 15),  const CircularProgressIndicator(  color: Colors.*white*,  strokeWidth: 3,  ),  ],  ),  ),  ),  );  } }  home.dart  import 'package:flutter/material.dart'; import 'practice\_screen.dart'; import 'quiz\_screen.dart'; import 'leaderboard\_screen.dart'; import 'profile\_screen.dart';  class HomeScreen extends StatelessWidget {  HomeScreen({super.key});   @override  Widget build(BuildContext context) {  return DefaultTabController(  length: 4,  child: Scaffold(  appBar: AppBar(  title: const Text(  "LearnPlay Hub",  style: TextStyle(fontWeight: FontWeight.*bold*),  ),  centerTitle: true,  elevation: 4,  bottom: const TabBar(  indicatorColor: Colors.*blueGrey*,  labelColor: Colors.*white*,  unselectedLabelColor: Colors.*white70*,  labelStyle: TextStyle(fontWeight: FontWeight.*bold*),  tabs: [  Tab(text: "Practice", icon: Icon(Icons.*book*)),  Tab(text: "Quiz", icon: Icon(Icons.*quiz*)),  Tab(text: "Leaderboard", icon: Icon(Icons.*leaderboard*)),  Tab(text: "Profile", icon: Icon(Icons.*person*)),  ],  ),  ),  body: TabBarView(  // Not const to allow non-const constructors inside  children: [  PracticeScreen(),  QuizScreen(),  LeaderboardScreen(),  ProfileScreen(),  ],  ),  ),  );  } }  number\_screen.dart  import 'package:flutter/material.dart';  class NumbersScreen extends StatelessWidget {  final List<Map<String, String>> numbers = List.generate(  20,  (index) => {  'number': (index + 1).toString(),  'text': *\_numberToWord*(index + 1),  },  );   NumbersScreen({super.key});   static String *\_numberToWord*(int number) {  const List<String> words = [  'One', 'Two', 'Three', 'Four', 'Five',  'Six', 'Seven', 'Eight', 'Nine', 'Ten',  'Eleven', 'Twelve', 'Thirteen', 'Fourteen', 'Fifteen',  'Sixteen', 'Seventeen', 'Eighteen', 'Nineteen', 'Twenty',  ];  return words[number - 1];  }   final List<List<Color>> gradientColors = [  [Color(0xFFFF9A9E), Color(0xFFFAD0C4)],  [Color(0xFFB5FFFC), Color(0xFF91EAE4)],  [Color(0xFFB3FFAB), Color(0xFF12FFF7)],  [Color(0xFFF6D365), Color(0xFFFDA085)],  [Color(0xFF84FAB0), Color(0xFF8FD3F4)],  ];   @override  Widget build(BuildContext context) {  return Scaffold(  backgroundColor: const Color(0xFFF8F9FB),  body: GridView.builder(  padding: const EdgeInsets.all(12),  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  crossAxisCount: 3,  crossAxisSpacing: 12,  mainAxisSpacing: 12,  childAspectRatio: 1,  ),  itemCount: numbers.length,  itemBuilder: (context, index) {  final gradient = gradientColors[index % gradientColors.length];   return GestureDetector(  onTap: () {  ScaffoldMessenger.*of*(context).showSnackBar(  SnackBar(  content: Text(  "You tapped: ${numbers[index]['text']}",  style: const TextStyle(fontSize: 16),  ),  duration: const Duration(milliseconds: 800),  behavior: SnackBarBehavior.floating,  ),  );  },  child: TweenAnimationBuilder(  tween: Tween<double>(begin: 0.95, end: 1.0),  duration: const Duration(milliseconds: 300),  curve: Curves.*easeOutBack*,  builder: (context, scale, child) {  return Transform.scale(  scale: scale,  child: Container(  decoration: BoxDecoration(  gradient: LinearGradient(  colors: gradient,  begin: Alignment.*topLeft*,  end: Alignment.*bottomRight*,  ),  borderRadius: BorderRadius.circular(18),  boxShadow: const [  BoxShadow(  color: Colors.*black12*,  blurRadius: 5,  offset: Offset(3, 3),  ),  ],  ),  child: Center(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  Text(  numbers[index]['number']!,  style: const TextStyle(  fontSize: 34,  fontWeight: FontWeight.*bold*,  color: Colors.*white*,  ),  ),  const SizedBox(height: 6),  Text(  numbers[index]['text']!,  style: const TextStyle(  fontSize: 18,  fontWeight: FontWeight.*w600*,  color: Colors.*white70*,  ),  ),  ],  ),  ),  ),  );  },  ),  );  },  ),  );  } }  alphabet\_screen.dart  import 'package:flutter/material.dart'; import 'dart:math';  class AlphabetsScreen extends StatelessWidget {  final List<Map<String, String>> alphabets = List.generate(  26,  (index) => {  'upper': String.fromCharCode(65 + index),  'lower': String.fromCharCode(97 + index),  },  );   AlphabetsScreen({super.key});   final List<List<Color>> gradientColors = [  [Color(0xFF89F7FE), Color(0xFF66A6FF)],  [Color(0xFFFFD3A5), Color(0xFFFFA6B7)],  [Color(0xFFA1FFCE), Color(0xFFFAFFD1)],  [Color(0xFF84FAB0), Color(0xFF8FD3F4)],  [Color(0xFFF6D365), Color(0xFFFDA085)],  ];   @override  Widget build(BuildContext context) {  return Scaffold(  backgroundColor: const Color(0xFFFDF6F0),  body: GridView.builder(  padding: const EdgeInsets.all(16),  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  crossAxisCount: 3,  crossAxisSpacing: 14,  mainAxisSpacing: 14,  childAspectRatio: 1,  ),  itemCount: alphabets.length,  itemBuilder: (context, index) {  final gradient = gradientColors[index % gradientColors.length];   return GestureDetector(  onTap: () {  ScaffoldMessenger.*of*(context).showSnackBar(  SnackBar(  content: Text(  "You tapped: ${alphabets[index]['upper']}",  style: const TextStyle(fontSize: 16),  ),  duration: const Duration(milliseconds: 800),  behavior: SnackBarBehavior.floating,  ),  );  },  child: TweenAnimationBuilder(  tween: Tween<double>(begin: 0.95, end: 1.0),  duration: const Duration(milliseconds: 400),  curve: Curves.*easeInOut*,  builder: (context, scale, child) {  return Transform.scale(  scale: scale,  child: Container(  decoration: BoxDecoration(  gradient: LinearGradient(  colors: gradient,  begin: Alignment.*topLeft*,  end: Alignment.*bottomRight*,  ),  borderRadius: BorderRadius.circular(18),  boxShadow: const [  BoxShadow(  color: Colors.*black12*,  blurRadius: 6,  offset: Offset(3, 3),  ),  ],  ),  child: Center(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  Text(  alphabets[index]['upper']!,  style: const TextStyle(  fontSize: 38,  fontWeight: FontWeight.w800,  color: Colors.white,  ),  ),  const SizedBox(height: 5),  Text(  alphabets[index]['lower']!,  style: const TextStyle(  fontSize: 24,  fontWeight: FontWeight.w500,  color: Colors.white70,  ),  ),  ],  ),  ),  ),  );  },  ),  );  },  ),  );  } }  quiz.dart  import 'package:flutter/material.dart'; import 'dart:math'; import 'package:shared\_preferences/shared\_preferences.dart';  class QuizScreen extends StatefulWidget {  const QuizScreen({super.key});   @override  \_QuizScreenState createState() => \_QuizScreenState(); }  class \_QuizScreenState extends State<QuizScreen> {  late List<Map<String, dynamic>> quizData;  int currentQuestionIndex = 0;  int score = 0;   @override  void initState() {  super.initState();  quizData = \_generateQuizData();  \_shuffleQuizData();  }   static String *\_numberToWord*(int number) {  const List<String> words = [  'One', 'Two', 'Three', 'Four', 'Five',  'Six', 'Seven', 'Eight', 'Nine', 'Ten',  'Eleven', 'Twelve', 'Thirteen', 'Fourteen', 'Fifteen',  'Sixteen', 'Seventeen', 'Eighteen', 'Nineteen', 'Twenty'  ];  return words[number - 1];  }   List<Map<String, dynamic>> \_generateQuizData() {  return [  {  'type': 'alphabet',  'question': 'Which letter comes after A?',  'display': 'A',  'options': ['B', 'C', 'D'],  'answer': 'B',  },  {  'type': 'alphabet',  'question': 'Which one is a vowel?',  'display': 'Letters: A, B, C',  'options': ['A', 'B', 'C'],  'answer': 'A',  },  {  'type': 'number',  'question': 'What number is this?',  'display': '3',  'options': ['3', 'A', '!'],  'answer': '3',  },  {  'type': 'number',  'question': 'Which number comes after 7?',  'display': '7',  'options': ['6', '8', '10'],  'answer': '8',  },  {  'type': 'math',  'question': 'What is 2 + 2?',  'display': '?',  'options': ['4', '3', '5'],  'answer': '4',  },  {  'type': 'math',  'question': 'How many sides does a triangle have?',  'display': '▲',  'options': ['3', '4', '5'],  'answer': '3',  },  {  'type': 'logic',  'question': 'What color is the sky on a clear day?',  'display': '🌤️',  'options': ['Blue', 'Green', 'Yellow'],  'answer': 'Blue',  },  {  'type': 'logic',  'question': 'Which one can fly?',  'display': '🐦 🐶 🐘',  'options': ['Bird', 'Dog', 'Elephant'],  'answer': 'Bird',  },  {  'type': 'alphabet',  'question': 'What is the first letter of “Apple”?',  'display': '🍎 Apple',  'options': ['A', 'P', 'L'],  'answer': 'A',  },  {  'type': 'number',  'question': 'How many fingers do you have on one hand?',  'display': '✋',  'options': ['4', '5', '6'],  'answer': '5',  },  ];  }    void \_shuffleQuizData() {  quizData.shuffle(Random());  for (var question in quizData) {  question['options'].shuffle(Random());  }  }   void checkAnswer(String selectedOption) async {  bool isCorrect = selectedOption == quizData[currentQuestionIndex]['answer'];  if (isCorrect) score++;   ScaffoldMessenger.*of*(context).showSnackBar(  SnackBar(  content: Text(isCorrect ? "Correct!" : "Wrong!"),  backgroundColor: isCorrect ? Colors.*green* : Colors.*red*,  duration: const Duration(milliseconds: 300),  ),  );   await Future.delayed(const Duration(milliseconds: 300));  setState(() {  if (currentQuestionIndex < 9) {  currentQuestionIndex++;  } else {  \_saveScore();  \_showResultDialog();  }  });  }   void \_saveScore() async {  final prefs = await SharedPreferences.*getInstance*();  String name = prefs.getString('userName') ?? 'Saumya';  await prefs.setInt('latestScore\_$name', score);  }   void \_showResultDialog() {  showDialog(  context: context,  builder: (context) => AlertDialog(  title: const Text("🎉 Quiz Completed!"),  content: Text("Your score: $score/10"),  actions: [  TextButton(  onPressed: () {  Navigator.*pop*(context);  setState(() {  currentQuestionIndex = 0;  score = 0;  \_shuffleQuizData();  });  },  child: const Text("Restart"),  ),  ],  ),  );  }   @override  Widget build(BuildContext context) {  final currentQuestion = quizData[currentQuestionIndex];   return Padding(  padding: const EdgeInsets.all(16.0),  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  LinearProgressIndicator(  value: (currentQuestionIndex + 1) / 10,  color: Colors.*lightBlueAccent*,  backgroundColor: Colors.*grey*[300],  minHeight: 8,  ),  const SizedBox(height: 20),  Text(  "Question ${currentQuestionIndex + 1}/10",  style: const TextStyle(  fontSize: 20,  fontWeight: FontWeight.*w600*,  color: Colors.*blueGrey*,  ),  ),  const SizedBox(height: 20),  Text(  currentQuestion['question'],  style: const TextStyle(  fontSize: 64,  fontWeight: FontWeight.*bold*,  color: Colors.*black87*,  ),  ),  const SizedBox(height: 20),  const Text(  "What is this?",  style: TextStyle(  fontSize: 24,  fontWeight: FontWeight.*bold*,  color: Colors.*black54*,  ),  ),  const SizedBox(height: 20),  Column(  children: currentQuestion['options'].map<Widget>((option) {  return Padding(  padding: const EdgeInsets.symmetric(vertical: 6.0),  child: ElevatedButton(  onPressed: () => checkAnswer(option),  style: ElevatedButton.*styleFrom*(  backgroundColor: Colors.*lightBlueAccent*,  foregroundColor: Colors.*white*,  minimumSize: const Size.fromHeight(50),  shape: RoundedRectangleBorder(  borderRadius: BorderRadius.circular(12),  ),  ),  child: Text(  option,  style: const TextStyle(fontSize: 18),  ),  ),  );  }).toList(),  ),  ],  ),  );  } }  profile.dart  import 'package:flutter/material.dart'; import 'package:shared\_preferences/shared\_preferences.dart';  class ProfileScreen extends StatefulWidget {  const ProfileScreen({super.key});   @override  State<ProfileScreen> createState() => \_ProfileScreenState(); }  class \_ProfileScreenState extends State<ProfileScreen> {  String userName = 'Kshitij';  int stars = 0;  final TextEditingController \_nameController = TextEditingController();   @override  void initState() {  super.initState();  \_loadProfile();  }   Future<void> \_loadProfile() async {  final prefs = await SharedPreferences.*getInstance*();  setState(() {  userName = prefs.getString('userName') ?? 'Kshitij';  stars = (prefs.getInt('latestScore\_$userName') ?? 0) ~/ 2;  \_nameController.text = userName;  });  }   Future<void> \_saveName(String newName) async {  final prefs = await SharedPreferences.*getInstance*();  int oldScore = prefs.getInt('latestScore\_$userName') ?? 0;  await prefs.setString('userName', newName);  await prefs.setInt('latestScore\_$newName', oldScore);  setState(() {  userName = newName;  stars = oldScore ~/ 2;  });  Navigator.*pop*(context);  }   void \_showEditNameDialog() {  showDialog(  context: context,  builder: (context) => AlertDialog(  title: const Text("Edit Name"),  content: TextField(  controller: \_nameController,  decoration: const InputDecoration(  hintText: "Enter your name",  ),  ),  actions: [  TextButton(  onPressed: () => Navigator.*pop*(context),  child: const Text("Cancel", style: TextStyle(color: Colors.*black54*)),  ),  TextButton(  onPressed: () => \_saveName(\_nameController.text.trim()),  child: const Text("Save", style: TextStyle(color: Colors.*blue*)),  ),  ],  ),  );  }   @override  Widget build(BuildContext context) {  return Center(  child: Padding(  padding: const EdgeInsets.all(20.0),  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  const CircleAvatar(  radius: 60,  backgroundColor: Colors.*lightBlueAccent*,  child: Icon(Icons.*person*, size: 80, color: Colors.*white*),  ),  const SizedBox(height: 20),  Text(  userName,  style: const TextStyle(  fontSize: 28,  fontWeight: FontWeight.*bold*,  color: Colors.*black87*,  ),  ),   const SizedBox(height: 15),  ElevatedButton(  onPressed: \_showEditNameDialog,  style: ElevatedButton.*styleFrom*(  backgroundColor: Colors.*black12*,  foregroundColor: Colors.*white*,  padding: const EdgeInsets.symmetric(horizontal: 32, vertical: 14),  ),  child: const Text("Edit Name", style: TextStyle(fontSize: 18)),  ),  ],  ),  ),  );  } } |
| **Screenshots** |  |
| **Question and Answers** | Answer the following Questions:  1. How to create Upper Tabs in Flutter?  To create upper tabs in Flutter, you typically use TabBar and TabBarView within a DefaultTabController. Here’s a basic structure:  DefaultTabController(  length: 3,  child: Scaffold(  appBar: AppBar(  title: Text('Upper Tabs Example'),  bottom: TabBar(  tabs: [  Tab(text: 'Tab 1'),  Tab(text: 'Tab 2'),  Tab(text: 'Tab 3'),  ],  ),  ),  body: TabBarView(  children: [  Center(child: Text('Content 1')),  Center(child: Text('Content 2')),  Center(child: Text('Content 3')),  ],  ),  ),  )  2. How did you use 60-30-10 rule in your application?  The 60-30-10 rule is a design principle used to create visually balanced UI:   * 60% is the primary color (e.g., background or large areas), * 30% is the secondary color (e.g., cards, containers), * 10% is the accent color (e.g., buttons, highlights).   In Flutter, I implemented this using the ThemeData and styled widgets:  ThemeData(  primaryColor: Colors.white,  secondaryHeaderColor: Colors.blueGrey,  accentColor: Colors.orangeAccent,  );  This rule ensures the UI stays clean and balanced, enhancing the user experience.  3. Which new elements did you use for creating UI components?  Some modern and useful Flutter elements I’ve used include:   * ListView.builder – for dynamic lists * Stack & Positioned – for layered UI * ClipRRect & Container with BoxDecoration – for rounded corners and gradients * CustomPaint – for drawing custom UI * AnimatedContainer – for smooth transitions * Flutter Hooks / Riverpod – for reactive UI with better state management   4. In pubspec.yaml file, what dependencies need to be there?  Common dependencies in pubspec.yaml depend on your app’s requirements, but a basic example:  dependencies:  flutter:  sdk: flutter  cupertino\_icons: ^1.0.2  provider: ^6.1.1  http: ^0.14.0  flutter\_svg: ^2.0.7  shared\_preferences: ^2.2.2  You should only include packages that your app actually uses to keep the build clean.  5. What is the use of Splash Screen?  A Splash Screen is the first screen users see when launching the app. It serves multiple purposes:   * Gives the app time to load resources * Enhances branding with logo or animation * Improves user experience by avoiding a blank screen   In Flutter, you can implement it using a native splash (via [flutter\_native\_splash](https://pub.dev/packages/flutter_native_splash)) or with a simple Flutter widget that navigates after a delay. |
| **Conclusion** | In building this Flutter application, I aimed to create a fun and educational experience for children to learn and recognize numbers and alphabets interactively. The app starts with an engaging Splash Screen, showcasing the branding and setting a cheerful tone.  We implemented a tab-based navigation using four upper tabs:   1. Practice/Lessons – Offers interactive lessons with visual aids for learning alphabets and numbers. 2. Quiz – Reinforces learning by turning it into a fun quiz format, helping children recognize letters and digits through repetition and engagement. 3. Leader Board – Encourages healthy competition by showcasing top scores and motivating kids to improve. 4. Profile – Allows users (or parents) to track progress, manage settings, and view achievements.   The use of a colorful, child-friendly UI, animations, and sound effects made the app more lively and interactive. With clear design principles like the 60-30-10 color rule, modern Flutter components, and thoughtful navigation, this app delivers a joyful and educational digital experience for young learners. |