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	Abhijeet S Jadhav
UCID	2024510021
Class	MCA
Batch	A
Date of Submission	12-04-2025

Technology used	Framework- Flutter, Tool used - Android Studio Programming- 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) Leaderboard 4) Profile
Code with proper label	home_tabs.dart :

```dart
import 'package:flutter/material.dart';
import 'practice_page.dart';
import 'quiz_page.dart';
import 'leaderboard_page.dart';
import 'profile_page.dart';
class HomeTabs extends StatefulWidget {
  const HomeTabs({super.key});
  @override
  State<HomeTabs> createState() => _HomeTabsState();
}
class _HomeTabsState extends State<HomeTabs> with
SingleTickerProviderStateMixin {
  late AnimationController _controller;
  @override
  void initState() {
    super.initState();
    _controller = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
```

```
    ).repeat(reverse: true); // Looping left and right
  }
  @override
  void dispose() {
    _controller.dispose();
    super.dispose();
  }
  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 4,
      child: Scaffold(
        backgroundColor: const Color(0xFFC8E6C9),
        appBar: AppBar(
          backgroundColor: const Color(0xFFFF6F61),
          title: AnimatedBuilder(
            animation: _controller,
            builder: (context, child) {
              double dx = 10 * (1 - (_controller.value *
2)).abs(); // Oscillates from -10 to +10
              return Transform.translate(
                offset: Offset(dx, 0),
                child: child,
              );
            },
            child: const Text(
              "Kids Learning",
              style: TextStyle(
                fontWeight: FontWeight.bold,
                fontSize: 26,
              color: Color.fromARGB(255, 241, 232, 230),
                shadows: [
                  Shadow(
                    blurRadius: 12.0,
                    color: Colors.yellowAccent,
                    offset: Offset(0, 0),
                  ),
                ],
              ),
            ),
          ),
          bottom: const TabBar(
            // isScrollable: true,
```

```
            labelColor: Color.fromARGB(255, 236, 247,
79),
            unselectedLabelColor: Colors.white,
            labelStyle: TextStyle(fontWeight:
FontWeight.bold, fontSize: 16),
            indicatorColor: Color(0xFFFFB74D),
            tabs: [
              Tab(text: 'Practice'),
              Tab(text: 'Quiz'),
              Tab(text: 'Leaderboard'),
              Tab(text: 'Profile'),
            ],
          ),
        ),
        body: TabBarView(
          children: [
            PracticePage(),
            const QuizPage(),
            const LeaderboardPage(),
            const ProfilePage(),
          ],
        ),
      ),
    );
  }
}
```

main.dart :

```
import 'package:flutter/material.dart';
import '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(
        scaffoldBackgroundColor: const Color(0xFFC8E6C9),
// Mint Green background
        primaryColor: const Color(0xFFFF6F61), // Coral
Pink
        textTheme: const TextTheme(
          headlineSmall: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: Color.fromARGB(255, 219, 247, 79), //
Bright Sky Blue
          ),
          bodyMedium: TextStyle(
            fontSize: 18,
            color: Color.fromARGB(255, 236, 247, 79), //
Bright Sky Blue
          ),
        ),
        appBarTheme: const AppBarTheme(
          backgroundColor: Color(0xFFFF6F61), // Coral
Pink
          foregroundColor: Colors.white, // White text
          centerTitle: true,
          elevation: 4,
        ),
        tabBarTheme: const TabBarTheme(
          labelColor: Color(0xFF4FC3F7), // Bright Sky
Blue
          unselectedLabelColor: Colors.white, // Soft
white
          indicator: UnderlineTabIndicator(
            borderSide: BorderSide(color:
Color(0xFFFFB74D), width: 3), // Orange Accent
          ),
        ),
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            backgroundColor: Color(0xFFFF6F61), // Coral
Pink
            foregroundColor: Colors.white, // White text
            textStyle: const TextStyle(fontSize: 16,
```

```
fontWeight: FontWeight.bold),
            padding: const EdgeInsets.symmetric(vertical:
12, horizontal: 24),
            shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
          ),
        ),
      ),
      home: const SplashScreen(),
    );
  }
}
```

leaderboard_page.dart :

```
import 'package:flutter/material.dart';
import
'package:shared_preferences/shared_preferences.dart';

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

  @override
  State<LeaderboardPage> createState() =>
_LeaderboardPageState();
}

class _LeaderboardPageState extends
State<LeaderboardPage> {
  List<Map<String, dynamic>> leaders = [];

  @override
  void initState() {
    super.initState();
    loadLeaderboard();
  }

  Future<void> loadLeaderboard() async {
    final prefs = await SharedPreferences.getInstance();
    final leaderboard =
prefs.getStringList('leaderboard') ?? [];
```

```
    leaders = leaderboard.map((entry) {
      final parts = entry.split(':');
      return {
        'name': parts[0],
        'score': int.tryParse(parts[1]) ?? 0,
      };
    }).toList();

    leaders.sort((a, b) =>
b['score'].compareTo(a['score']));
    setState(() {});
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFFFFEBEE),
      body: Padding(
        padding: const EdgeInsets.symmetric(horizontal:
16, vertical: 30),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Center(
              child: Container(
                padding: const
EdgeInsets.symmetric(horizontal: 24, vertical: 12),
                decoration: BoxDecoration(
                  color: const Color.fromARGB(255, 243,
242, 244),
                  borderRadius:
BorderRadius.circular(12),
                  border: Border.all(
                    color: const Color.fromARGB(255, 96,
10, 109),
                    width: 3,
                  ),
                  boxShadow: [
                    BoxShadow(
                      color: const Color.fromARGB(255,
92, 13, 106).withOpacity(0.3),
                      blurRadius: 6,
                      offset: const Offset(0, 4),
```

```
                    ),
                  ],
                ),
                child: const Text(
                  "Top Learners",
                  style: TextStyle(
                    color: Color.fromARGB(255, 140, 19,
145),
                    fontWeight: FontWeight.bold,
                    fontSize: 22,
                    letterSpacing: 1.2,
                  ),
                  textAlign: TextAlign.center,
                ),
              ),
            ),
            const SizedBox(height: 20),
            Expanded(
              child: leaders.isEmpty
                  ? const Center(
                      child: Text(
                        "No scores yet!",
                        style: TextStyle(color:
Colors.white, fontSize: 18),
                      ),
                    )
                  : ListView.builder(
                      itemCount: leaders.length,
                      itemBuilder: (context, index) {
                        final leader = leaders[index];
                        return Card(
                          color: const Color(0xFFFFD700),
                          shape: RoundedRectangleBorder(
                            borderRadius:
BorderRadius.circular(16),
                          ),
                          child: ListTile(
                            leading: CircleAvatar(
                              backgroundColor: const
Color(0xFF4B0082),
                              child: Text(
                                "#${index + 1}",
                                style: const TextStyle(
```

```
                                    color:
Color(0xFFFFD700),
                                    fontWeight:
FontWeight.bold,
                                  ),
                                ),
                              ),
                              title: Text(
                                leader["name"],
                                style: const TextStyle(
                                  color:
Color.fromARGB(255, 93, 15, 122),
                                  fontWeight:
FontWeight.bold,
                                  fontSize: 18,
                                ),
                              ),
                              trailing: Text(
                                "${leader["score"]} pts",
                                style: const TextStyle(
                                  color:
Color.fromARGB(255, 55, 10, 99),
                                  fontWeight:
FontWeight.w600,
                                  fontSize: 16,
                                ),
                              ),
                            ),
                          );
                        },
                      ),
                  ),
                ],
              ),
            ),
          );
        }
}
```

practice_page.dart :

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

```
import 'package:animate_do/animate_do.dart';
import 'dart:math';

class PracticePage extends StatelessWidget {
  PracticePage({super.key});

  final FlutterTts tts = FlutterTts();
  final Random random = Random();

  final List<String> alphabets =
      List.generate(26, (index) => String.fromCharCode(65
+ index));
  final List<String> numbers =
      List.generate(10, (index) => (index +
1).toString());

  final List<Color> cardColors = [
    Color.fromARGB(255, 245, 242, 62), // Light Aqua
  ];

  Future<void> speak(String value) async {
    await tts.setLanguage("en-US");
    await tts.setPitch(1);
    await tts.setSpeechRate(0.4);
    await tts.speak(value);
  }

  Widget buildSection({
    required String title,
    required List<String> items,
    required String type,
    required Color backgroundColor,
  }) {
    return Container(
      color: backgroundColor,
      padding: const EdgeInsets.only(bottom: 24),
      child: ExpansionTile(
        initiallyExpanded: true,
        tilePadding: const
EdgeInsets.symmetric(horizontal: 16),
        childrenPadding: const EdgeInsets.only(top: 8),
        title: Container(
          padding: const EdgeInsets.symmetric(vertical:
```

```
12, horizontal: 16),
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(16),
            border: Border.all(color: Colors.deepPurple,
width: 2),
            boxShadow: [
              BoxShadow(
                color:
Colors.deepPurple.withOpacity(0.15),
                blurRadius: 8,
                offset: const Offset(0, 4),
              ),
            ],
          ),
          child: Row(
            children: [
              const Icon(Icons.menu_book_outlined, color:
Colors.deepPurple),
              const SizedBox(width: 8),
              Text(
                title,
                style: const TextStyle(
                  fontSize: 22,
                  fontWeight: FontWeight.bold,
                  color: Colors.deepPurple,
                ),
              ),
              const Spacer(),
              const Icon(Icons.expand_more, color:
Colors.deepPurple),
            ],
          ),
        ),
        children: [
          GridView.count(
            crossAxisCount: 3,
            shrinkWrap: true,
            physics: const
NeverScrollableScrollPhysics(),
            padding: const EdgeInsets.all(12),
            mainAxisSpacing: 16,
            crossAxisSpacing: 16,
```

```
            childAspectRatio: 0.9,
            children: items.map((item) {
              final imagePath = 'assets/$type/$item.png';
              final cardColor =
cardColors[random.nextInt(cardColors.length)];

              return BounceInDown(
                child: GestureDetector(
                  onTap: () => speak(item),
                  child: Card(
                    color: cardColor,
                    shape: RoundedRectangleBorder(
                      borderRadius:
BorderRadius.circular(16),
                    ),
                    elevation: 4,
                    shadowColor: Colors.grey.shade300,
                    child: Padding(
                      padding: const
EdgeInsets.all(10.0),
                      child: Container(
                        decoration: type == 'numbers'
                            ? BoxDecoration(
                                color:
Colors.orangeAccent, // Orange background for numbers
                                borderRadius:
BorderRadius.circular(12),
                              )
                            : null,
                        padding: const EdgeInsets.all(8),
                        child: Image.asset(
                          imagePath,
                          fit: BoxFit.contain,
                        ),
                      ),
                    ),
                  ),
                ),
              );
            }).toList(),
          ),
        ],
      ),
```

```
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SingleChildScrollView(
        child: Column(
          children: [
            buildSection(
              title: "Alphabets",
              items: alphabets,
              type: "alphabets",
              backgroundColor: const Color(0xFFFFF3E0),
// Soft Pink
            ),
            buildSection(
              title: "Numbers",
              items: numbers,
              type: "numbers",
              backgroundColor: const Color(0xFFFFEBEE),
// Light Orange
            ),
          ],
        ),
      ),
    );
  }
}
```

profile_page.dart :

```
import 'dart:math';
import 'package:flutter/material.dart';
import
'package:shared_preferences/shared_preferences.dart';

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

  @override
  State<ProfilePage> createState() =>
```

```
_ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> with
TickerProviderStateMixin {
  String userName = '';
  String age = '';
  String favoriteSubject = '';
  int totalPoints = 0;

  bool isEditingName = false;
  bool isEditingAge = false;
  bool isEditingSubject = false;

  TextEditingController nameController =
TextEditingController();
  TextEditingController ageController =
TextEditingController();
  TextEditingController subjectController =
TextEditingController();

  final List<String> randomAvatars = [
    '🦁', '🐵', '🦊', '🐰', '🐱', '🐼', '🐸', '🐯', '🐶'
  ];

  late AnimationController _avatarController;
  late Animation<double> _bounceAnimation;

  @override
  void initState() {
    super.initState();
    loadProfile();

    _avatarController = AnimationController(
      vsync: this,
      duration: const Duration(milliseconds: 800),
    )..repeat(reverse: true);

    _bounceAnimation = Tween<double>(begin: 0, end:
10).animate(
      CurvedAnimation(parent: _avatarController, curve:
Curves.easeInOut),
    );
```

```
  }

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

  Future<void> loadProfile() async {
    final prefs = await SharedPreferences.getInstance();
    setState(() {
      userName = prefs.getString('username') ?? '';
      age = prefs.getString('age') ?? '';
      favoriteSubject = prefs.getString('subject') ?? '';
      totalPoints = prefs.getInt('points') ?? 0;
    });
  }

  Future<void> saveField(String key, String value) async
{
    final prefs = await SharedPreferences.getInstance();
    await prefs.setString(key, value);
    setState(() {
      if (key == 'username') {
        userName = value;
        isEditingName = false;
      } else if (key == 'age') {
        age = value;
        isEditingAge = false;
      } else if (key == 'subject') {
        favoriteSubject = value;
        isEditingSubject = false;
      }
    });
  }

  String getAvatarEmoji() {
    if (userName.isNotEmpty) {
      return userName[0].toUpperCase();
    } else {
      return
randomAvatars[Random().nextInt(randomAvatars.length)];
    }
```

```
  }

  Widget buildProfileCard({
    required String title,
    required String value,
    required IconData icon,
    VoidCallback? onEdit,
  }) {
    final bool isValueSet = value.isNotEmpty;

    return AnimatedContainer(
      duration: const Duration(milliseconds: 400),
      curve: Curves.easeOutBack,
      margin: const EdgeInsets.symmetric(vertical: 10),
      decoration: BoxDecoration(
        color: const Color(0xFFFFD700),
        borderRadius: BorderRadius.circular(20),
        boxShadow: [
          BoxShadow(
            color: Colors.white.withOpacity(0.1),
            blurRadius: 8,
            offset: const Offset(0, 4),
          )
        ],
      ),
      child: ListTile(
        leading: Icon(icon, color: const
Color(0xFF4B0082)),
        title: Text(
          title,
          style: const TextStyle(
            fontWeight: FontWeight.bold,
            color: Color(0xFF4B0082),
          ),
        ),
        trailing: Row(
          mainAxisSize: MainAxisSize.min,
          children: [
            Text(
              isValueSet ? value : "Not set",
              style: const TextStyle(
                fontWeight: FontWeight.w600,
                fontSize: 16,
```

```
                color: Color(0xFF4B0082),
              ),
            ),
            if (onEdit != null) ...[
              const SizedBox(width: 8),
              IconButton(
                icon: const Icon(Icons.edit, size: 20,
color: Color(0xFF4B0082)),
                onPressed: onEdit,
              ),
            ]
          ],
        ),
      ),
    );
  }

  Widget buildEditField({
    required String label,
    required TextEditingController controller,
    required VoidCallback onSave,
  }) {
    return Column(
      children: [
        TextField(
          controller: controller,
          decoration: InputDecoration(
            hintText: "Enter $label",
            filled: true,
            fillColor: Colors.white,
            border: OutlineInputBorder(borderRadius:
BorderRadius.circular(16)),
          ),
        ),
        const SizedBox(height: 10),
        ElevatedButton.icon(
          style: ElevatedButton.styleFrom(
            backgroundColor: const Color(0xFFFF6F61),
          ),
          onPressed: onSave,
          icon: const Icon(Icons.check, color:
Color(0xFF4B0082)),
          label: Text(
```

```
            "Save $label",
            style: const TextStyle(color:
Color(0xFF4B0082)),
          ),
        ),
      ],
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFFFFF3E0),
      body: SafeArea(
        child: SingleChildScrollView(
          padding: const EdgeInsets.all(20),
          child: Column(
            children: [
              AnimatedBuilder(
                animation: _bounceAnimation,
                builder: (_, child) {
                  return Transform.translate(
                    offset: Offset(0,
-_bounceAnimation.value),
                    child: child,
                  );
                },
                child: CircleAvatar(
                  radius: 60,
                  backgroundColor: const
Color(0xFFFF6F61),
                  child: Text(
                    getAvatarEmoji(),
                    style: const TextStyle(fontSize: 48,
color: Color(0xFF4B0082)),
                  ),
                ),
              ),
              const SizedBox(height: 20),
              Text(
                userName.isEmpty ? "Hello, Kiddo!" :
"Hello, $userName!",
                style: const TextStyle(
```

```
                    fontSize: 26,
                    fontWeight: FontWeight.bold,
                    color: Color(0xFFFF6F61),
                  ),
                ),
                const SizedBox(height: 20),

                if (isEditingName)
                  buildEditField(
                    label: "Name",
                    controller: nameController,
                    onSave: () {
                      if
(nameController.text.trim().isNotEmpty) {
                        saveField('username',
nameController.text.trim());
                      }
                    },
                  ),

                if (isEditingAge)
                  buildEditField(
                    label: "Age",
                    controller: ageController,
                    onSave: () {
                      if
(ageController.text.trim().isNotEmpty) {
                        saveField('age',
ageController.text.trim());
                      }
                    },
                  ),

                if (isEditingSubject)
                  buildEditField(
                    label: "Favorite Subject",
                    controller: subjectController,
                    onSave: () {
                      if
(subjectController.text.trim().isNotEmpty) {
                        saveField('subject',
subjectController.text.trim());
                      }
```

```
                    },
                  ),

                const SizedBox(height: 20),

                buildProfileCard(
                  title: "Name",
                  value: userName,
                  icon: Icons.person,
                  onEdit: () {
                    setState(() {
                      isEditingName = true;
                      nameController.text = userName;
                    });
                  },
                ),
                buildProfileCard(
                  title: "Age",
                  value: age,
                  icon: Icons.cake,
                  onEdit: () {
                    setState(() {
                      isEditingAge = true;
                      ageController.text = age;
                    });
                  },
                ),
                buildProfileCard(
                  title: "Favorite Subject",
                  value: favoriteSubject,
                  icon: Icons.school,
                  onEdit: () {
                    setState(() {
                      isEditingSubject = true;
                      subjectController.text =
favoriteSubject;
                    });
                  },
                ),
                buildProfileCard(
                  title: "Total Points",
                  value: "$totalPoints pts",
                  icon: Icons.star,
```

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

quiz_page.dart :

```
import 'package:flutter/material.dart';
import
'package:shared_preferences/shared_preferences.dart';
import 'dart:math';
import 'leaderboard_page.dart';

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

  @override
  State<QuizPage> createState() => _QuizPageState();
}

class _QuizPageState extends State<QuizPage> {
  final List<String> allChars = [
    ...'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.split(''),
    ...List.generate(10, (index) => (index +
1).toString())
  ];

  late String correctAnswer;
  late List<String> options;
  int score = 0;
  int questionCount = 0;

  @override
  void initState() {
    super.initState();
    generateQuestion();
  }

  void generateQuestion() {
```

```
    final rand = Random();
    correctAnswer =
allChars[rand.nextInt(allChars.length)];
    options = [correctAnswer];

    while (options.length < 4) {
      String newOption =
allChars[rand.nextInt(allChars.length)];
      if (!options.contains(newOption)) {
        options.add(newOption);
      }
    }

    options.shuffle();
  }

  Future<void> checkAnswer(String selectedOption) async {
    final isCorrect = selectedOption == correctAnswer;
    questionCount++;

    if (isCorrect) score++;

    showDialog(
      context: context,
      builder: (_) => AlertDialog(
        backgroundColor: const Color(0xFF4B0082),
        title: Text(
          isCorrect ? "🎉 Correct!" : "❌ Oops!",
          style: TextStyle(
            color: isCorrect ? Colors.greenAccent :
Colors.redAccent,
            fontWeight: FontWeight.bold,
          ),
        ),
        content: Text(
          "Answer: $correctAnswer",
          style: const TextStyle(color:
Color(0xFFADD8E6)),
        ),
        actions: [
          TextButton(
            child: const Text(
              "Next",
```

```
                style: TextStyle(color: Color(0xFFFFD700)),
              ),
              onPressed: () {
                Navigator.of(context).pop();
                if (questionCount < 10) {
                  setState(() => generateQuestion());
                } else {
                  showFinalScore();
                }
              },
            ),
          ],
        ),
      );
    }

    Future<void> showFinalScore() async {
      final prefs = await SharedPreferences.getInstance();
      String name = prefs.getString('username') ??
'Anonymous';

      List<String> leaderboard =
prefs.getStringList('leaderboard') ?? [];
      leaderboard.add('$name:$score');
      await prefs.setStringList('leaderboard',
leaderboard);

      int currentPoints = prefs.getInt('points') ?? 0;
      await prefs.setInt('points', currentPoints + score);

      if (!mounted) return;
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (_) => const
LeaderboardPage()),
      );
    }

    String getImagePath(String char) {
      if (int.tryParse(char) != null) {
        return 'assets/numbers/$char.png';
      } else {
        return 'assets/alphabets/$char.png';
```

```
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFFFFF3E0), // Soft
Pink
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.all(20),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                const SizedBox(height: 10),
                Center(
                  child: Container(
                    padding: const EdgeInsets.symmetric(
                        vertical: 10, horizontal: 20),
                    decoration: BoxDecoration(
                      color: Colors.white,
                      borderRadius:
BorderRadius.circular(12),
                      border: Border.all(
                        color: Color(0xFF6A1B9A), // Deep
Purple
                        width: 3,
                      ),
                      boxShadow: [
                        BoxShadow(
                          color:
Colors.purple.withOpacity(0.2),
                          blurRadius: 8,
                          offset: const Offset(0, 4),
                        ),
                      ],
                    ),
                    child: Text(
                      "Question ${questionCount + 1} of
10",
                      style: const TextStyle(
```

```
                        fontSize: 20,
                        color: Color.fromARGB(255, 83,
22, 157),
                        fontWeight: FontWeight.bold,
                        letterSpacing: 1.2,
                      ),
                    ),
                  ),
                ),
                const SizedBox(height: 20),
                const Text(
                  "Which character is shown?",
                  style: TextStyle(
                    fontSize: 26,
                    color: Color(0xFFFF6F61),
                    fontWeight: FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 30),
                Center(
                  child: Container(
                    padding: const EdgeInsets.all(20),
                    decoration: BoxDecoration(
                      color: const Color.fromARGB(255,
235, 246, 43),
                      borderRadius:
BorderRadius.circular(20),
                      boxShadow: [
                        BoxShadow(
                          color: const
Color.fromARGB(255, 227, 8, 247),
                          blurRadius: 10,
                          spreadRadius: 2,
                          offset: const Offset(0, 4),
                        ),
                      ],
                    ),
                    child: Image.asset(
                      getImagePath(correctAnswer),
                      height: 160,
                      width: 160,
                      fit: BoxFit.contain,
                    ),
```

```
                    ),
                  ),
                  const SizedBox(height: 40),
                  Column(
                    children: options.map((option) {
                      return Padding(
                        padding: const
EdgeInsets.symmetric(vertical: 8),
                        child: ElevatedButton(
                          style: ElevatedButton.styleFrom(
                            backgroundColor: const
Color(0xFFFFD700),
                            foregroundColor: const
Color(0xFF4B0082),
                            minimumSize: const
Size(double.infinity, 50),
                            shape: RoundedRectangleBorder(
                              borderRadius:
BorderRadius.circular(12),
                              side: const BorderSide(
                                color: Color(0xFF6A1B9A),
                                width: 2,
                              ),
                            ),
                            elevation: 5,
                          ),
                          onPressed: () =>
checkAnswer(option),
                          child: Text(
                            option,
                            style: const TextStyle(
                              fontSize: 22,
                              fontWeight: FontWeight.w600,
                            ),
                          ),
                        ),
                      );
                    }).toList(),
                  ),
                  const SizedBox(height: 20),
                ],
              ),
            ),
```

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

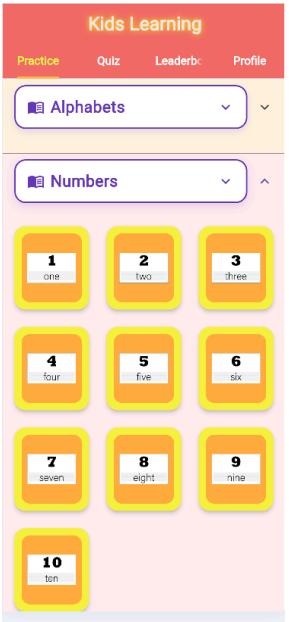
splash_screen.dart :

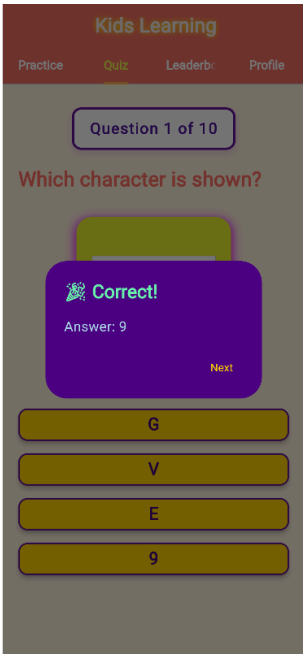
```
import 'dart:async';
import 'package:flutter/material.dart';
import 'home_tabs.dart';

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

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

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

  @override
  void initState() {
    super.initState();

    // Animation controller for scaling effect
    _controller = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 4),
    );

    // Zoom-out to zoom-in effect
    _animation = Tween<double>(begin: 1.0, end:
4.0).animate(
      CurvedAnimation(
        parent: _controller,
        curve: Curves.easeInOut,
      ),
```

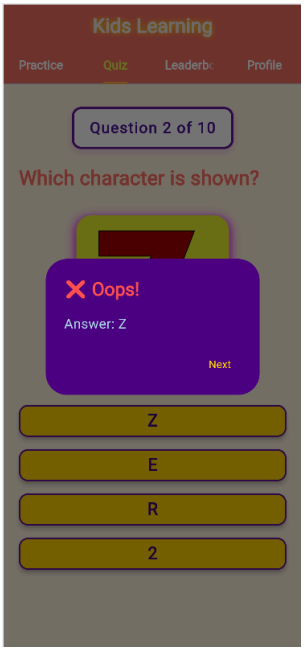
```
    );

    // Start animation
    _controller.forward();

    // Navigate to HomeTabs after 4 seconds
    Timer(const Duration(seconds: 4), () {
      Navigator.of(context).pushReplacement(
        MaterialPageRoute(builder: (context) => const
HomeTabs()),
      );
    });
  }

  @override
  void dispose() {
    _controller.dispose(); // Dispose animation
controller
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white, // New background
color
      body: Center(
        child: AnimatedBuilder(
          animation: _animation,
          builder: (context, child) {
            return Transform.scale(
              scale: _animation.value,
              child: Image.asset(
                'assets/numbers/kids.png', // New image
path
                width: 150,
                height: 150,
              ),
            );
          },
        ),
      ),
    );
```

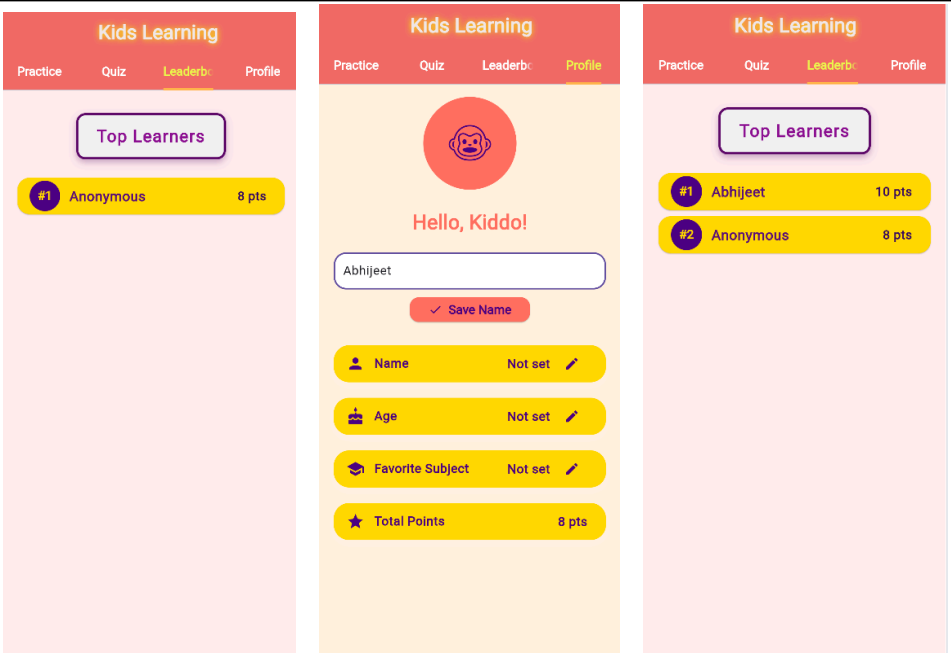
```
    }
}
```

Screenshots









Question and Answers

Answer the following Questions:

1. How to create Upper Tabs in Flutter?

Ans : DefaultTabController(
length: 3,
child: Scaffold(
appBar: AppBar(
title: Text('Tabs Example'),
bottom: TabBar(
tabs: [
Tab(text: 'Tab 1'),
Tab(text: 'Tab 2'),
Tab(text: 'Tab 3'),
],
),
),
body: TabBarView(
children: [
Center(child: Text('Tab 1 Content')),
Center(child: Text('Tab 2 Content')),
Center(child: Text('Tab 3 Content')),
],
),
),
)

2. How did you use 60-30-10 rule in your application?

Ans : The 60-30-10 rule is a classic design principle used to maintain color harmony:

60% Primary Color – Used for the main background or base (like the app's scaffold background or main UI area).

30% Secondary Color – Used for cards, containers, buttons, etc., to contrast and support the primary color.

	10% Accent Color – Used for highlights, icons, CTA buttons, or alerts to attract attention. In our Flutter app: We used 60% light neutral tone (like white or light grey) for the base UI, 30% secondary color (like deep blue or teal) for cards and section headers, 10% accent color (like amber or red) for icons and important action buttons. 3. Which new elements did you use for creating UI components? **Ans :** Card, ListTile, ElevatedButton, Container, Stack, GridView, ClipRRect, CustomPaint, ExpansionTile, and SliverAppBar for dynamic UI 4. In pubspec.yaml file, what dependencies need to be there? **Ans :** dependencies: flutter: sdk: flutter cupertino_icons: ^1.0.2 provider: ^6.1.0 http: ^0.14.0 flutter_svg: ^2.0.7 google_fonts: ^6.0.0 5. What is the use of Splash Screen? **Ans :** A Splash Screen is the initial screen that shows when the app is launching. Purpose: Displays your brand logo or app name. Gives users a smooth transition while the app initializes. Helps preload essential data before navigating to the main UI In Flutter, you can implement it using: flutter_native_splash package (recommended for native splash). A custom SplashScreen widget with a timer or logic before redirecting to the main screen.
Conclusion	Through this Flutter application, we effectively implemented key UI and design principles to enhance user experience. We utilized upper tabs using TabBar and TabBarView for intuitive navigation, applied the 60-30-10 color rule to maintain visual harmony, and integrated modern UI components like Card, ListTile, and SliverAppBar for a dynamic layout. Essential dependencies such as provider, http, and flutter_svg were included to support state management, API integration, and SVG rendering. Overall, the app delivers a clean, responsive, and engaging interface.