**Practical Number: 01**

* 1. **What is Flutter?**

- Flutter is an open-source UI software development toolkit created by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It provides a rich set of pre-designed widgets that enable developers to create visually appealing and high-performance applications with ease. Flutter uses the Dart programming language and offers a reactive framework that allows for fast development and a hot-reload feature, which makes testing and debugging efficient. Its layered architecture enables full customization, making it a powerful tool for creating expressive and flexible designs. With its growing community and robust libraries, Flutter has become a popular choice for cross-platform app development.

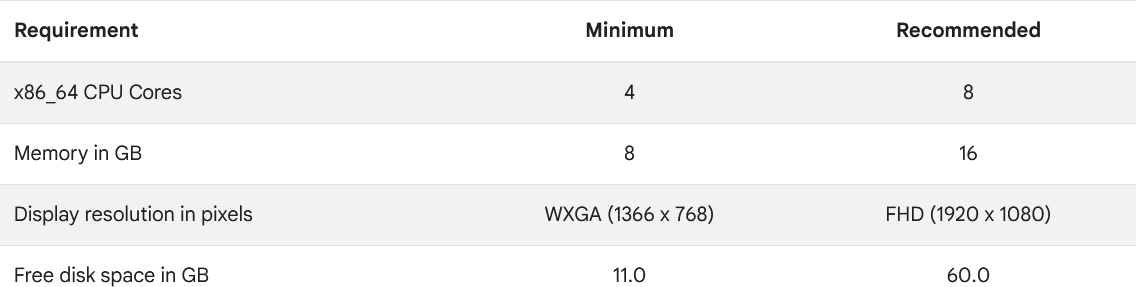
* 1. **Why to use Flutter?**
     + Build applications for Android, iOS, web, and desktop using a single codebase.
     + Provides a rich set of widgets to create attractive and responsive designs.
     + Real-time code updates and testing are possible with the hot reload feature, making development faster and more efficient.
     + Offers near-native performance with smooth animations and fast load times through Dart compilation and a powerful rendering engine.
     + Saves time, effort, and cost by eliminating the need for separate development for different platforms.
     + Comes with extensive libraries, plugins, and tools to enhance functionality and accelerate development.
     + Backed by Google and a growing global community, ensuring continuous improvements and support.
     + Easily integrates with existing systems and supports third-party tools, making it versatile for diverse applications.
     + Regular updates and new features ensure it remains relevant for modern development needs.
     + A cost-efficient solution for startups and businesses aiming for quality applications on multiple platforms.
  2. **What is dart?**

- Dart is an open-source, general-purpose programming language developed by Google, designed for building fast, scalable, and reliable applications. It is the language behind Flutter, enabling developers to create high-performance mobile, web, and desktop applications from a single codebase. Dart is an object-oriented, class-based language with a syntax similar to languages like Java, JavaScript, and C#, making it easy to learn for developers familiar with those languages. It supports both ahead-of-time (AOT) and just-in-time (JIT) compilation, ensuring fast execution and dynamic development capabilities. With features like strong typing, asynchronous programming support, and a rich standard library, Dart is optimized for creating modern, feature-rich applications.

**Steps to install Flutter:**

* + - Verify System Requirements
      * Hardware Requirements

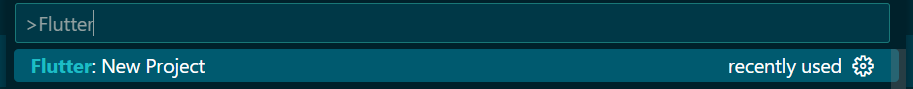
**Practical Number: 01**

****

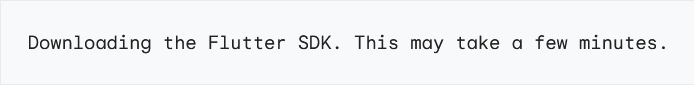
* + - * Software Requirements

Flutter supports 64-bit version of Microsoft Windows 10 or later. These versions of Windows should include the required Windows PowerShell 5 or later.

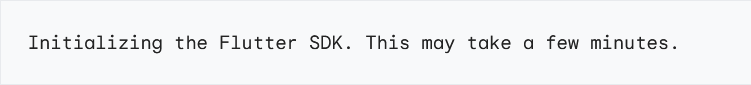
* + - Use a text-editor or IDE like Visual Studio, Android Studio or Visual Studio Code and download Git before proceeding.
    - You can proceed by either downloading the flutter SDK directly from the website [https://docs.flutter.dev/get-](https://docs.flutter.dev/get-started/install/windows/mobile#install-the-flutter-sdk) [started/install/windows/mobile#install-the-flutter-sdk](https://docs.flutter.dev/get-started/install/windows/mobile#install-the-flutter-sdk) or download it using Visual Studio Code.
    - We will proceed using VS code
      * Launch VS code
      * Press CTRL + Shift + P to open the command palette.
      * Type Flutter and select the prompt **Flutter: New Project**

****

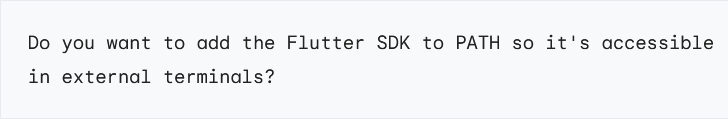
* + - * If you’ve already downloaded the flutter SDK select **Locate SDK** or click **Download SDK**
      * Choose the location you want to install flutter SDK like ‘C:\’
      * Click Clone Flutter. A popup will show up like this



* + - * Once it finishes, VS code will start initializing it directly.

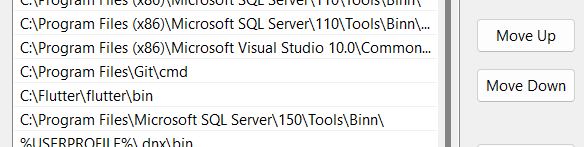


* + - * When the Flutter install succeeds, VS Code displays this pop-up notification; Proceed with ‘Yes’

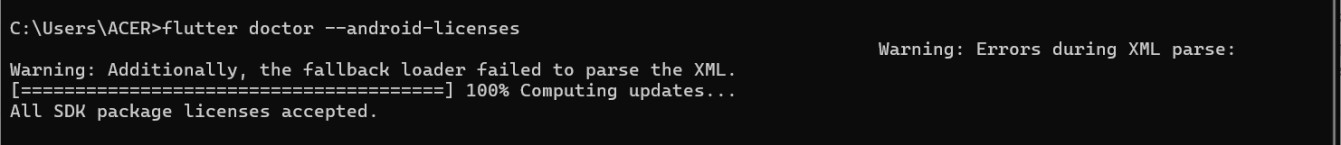


**Practical Number: 01**

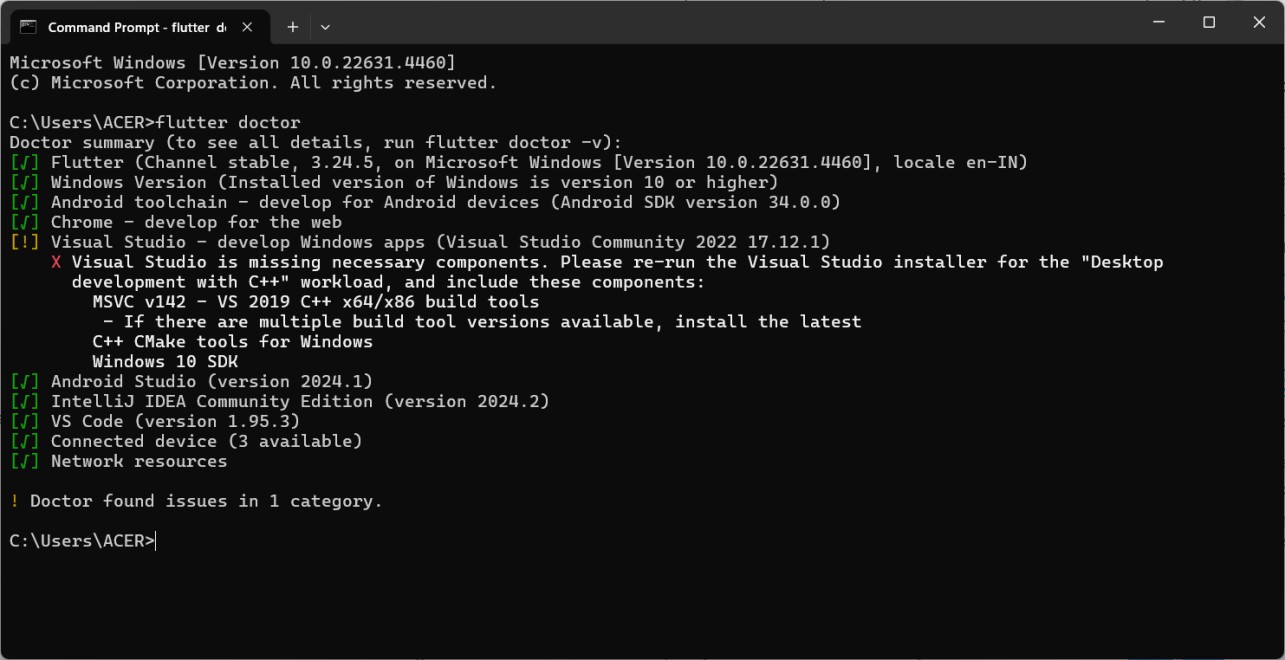
* + - * Flutter SDK’s bin folder will be added to environment variable.



* + - * Run the flutter doctor --android-licenses command to accept the android licenses



* + - * Restart VS code to finish successfully.
      * You can run the command ‘flutter doctor’ to check if the configuration is successful.



* + - * If the flutter doctor command returns an error for any of these components, run it again with the verbose flag.



**Q1) Demonstrate For and For of Loop in Dart.**

**Code:**

void main() {

for (int i = 1; i <= 10; i++) {

if(i%2==0){

print(i);

}

}

print(" ");

List planetList=["Mercury","Venus","Earth","Mars"];

for(String planet in planetList){

print(planet);

}

print(" ");

List fruitList=["Mango","Apple","Banana"];

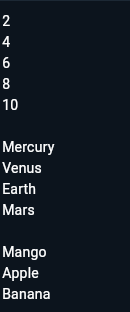
for(String fruit in fruitList){

print(fruit);

}

}

**Output:**

****

**Q2) Write a Program to Print the table using For Loop.**

**Code:**

void main(){

int n=5;

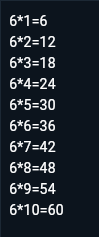
for(int i=1;i<=10;i++){

print("$n\*$i=${n \* i}");

}

}

**Output:**

****

**Q3) Demonstrate the use of Function in Dart.**

**Code:**

void main() {

findPerimeter(4,2);

int rectArea = getArea(10,5);

print("The area is $rectArea");

}

void findPerimeter(int length, int breadth){

int perimeter = 2\*(length+breadth);

print("The perimeter is $perimeter");

}

int getArea(int length, int breadth) {

int area = length \* breadth;

return area;

}

**Ouput:**

****

**Q4) Demonstrate the use of Different Types of Parameter.**

**Required Parameter**

**Code:**

void main(){

  Cities("New york","New delhi","Sydeny","India");

  print(" ");

}

void Cities(String a, String b, String c, String d) {

  print("Name is $a");

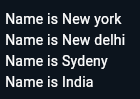
  print("Name is $b");

  print("Name is $c");

print("Name is $d");

}

**Output:**

****

**Optional Positional Parameter**

**Code:**

void main() {

Cities("New york", "New delhi", "Hong Kong");

print(" ");

}

void Cities(String n1, String n2, [String n3 = ""]) {

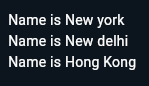
print("Name is $n1");

print("Name is $n2");

print("Name is $n3");

}

**Output:**

****

**Named Optional Parameter**

**Code:**

void main() {

Cities("New york", "New delhi", n3: "Goa");

print(" ");

}

void Cities(String n1, String n2, {String n3 = ""}) {

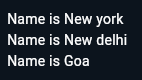
print("Name is $n1");

print("Name is $n2");

print("Name is $n3");

}

**Output:**

****

**Q5) Demonstrate the use of Class & Object in Dart.**

**Code:**

void main() {

var student1 = student();

student1.id = 54;

student1.name = "Shashank Kavlekar";

print("Student Id: ${student1.id} ");

print("Student Name :${student1.name}");

student1.play();

student1.sleep();

}

class student {

int id = -1;

String name = "";

void play() {

print("${this.name} is now Playing");

}

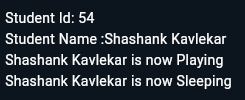
void sleep() {

print("${this.name} is now Sleeping ");

}

}

**Output:**

****

**Q1) Demonstrate the use of Flatter Layout.**

**Code**

import 'package:flutter/material.dart';

void main() {

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Flutter Demo',

      theme: ThemeData(

        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),

        useMaterial3: true,

      ),

      home: MyWidget(),

    );

  }

}

class MyWidget extends StatelessWidget {

  MyWidget({super.key});

  final List<Map<String, dynamic>> layout = [

    {'title': 'Column', 'widget': ColumnLayout(), 'color': Colors.green},

    {'title': 'Row', 'widget': RowLayout(), 'color': Colors.yellow},

    {'title': 'Stack', 'widget': StackLayout(), 'color': Colors.orange},

    {'title': 'Grid', 'widget': GridLayout(), 'color': Colors.blue},

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text('Flutter Widget'),

      ),

      body: ListView.builder(

        itemCount: layout.length,

        itemBuilder: (context, index) {

          return Container(

            margin: EdgeInsets.symmetric(vertical: 5, horizontal: 10),

            decoration: BoxDecoration(

              color: layout[index]['color'], // Set different colors for each item

              borderRadius: BorderRadius.circular(10),

            ),

            child: ListTile(

              title: Text(

                layout[index]['title'],

                style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold),

              ),

              trailing: Icon(Icons.arrow\_forward, color: Colors.white),

              tileColor: layout[index]['color'], // Apply background color

              shape: RoundedRectangleBorder(

                borderRadius: BorderRadius.circular(10),

              ),

              onTap: () {

                Navigator.push(

                  context,

                  MaterialPageRoute(

                    builder: (context) => layout[index]['widget'],

                  ),

                );

              },

            ),

          );

        },

      ),

    );

  }

}

class ColumnLayout extends StatelessWidget {

  const ColumnLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Column Layout')),

      body: Container(

        color: Colors.green[50],

        child: Center(

          child: Container(

            width: 300,

            padding: EdgeInsets.all(16),

            decoration: BoxDecoration(

              color: Colors.greenAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: Column(

              mainAxisAlignment: MainAxisAlignment.center,

              children: [

                Text('Child 1', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 2', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 3', style: TextStyle(fontSize: 20, color: Colors.white)),

              ],

            ),

          ),

        ),

      ),

    );

  }

}

class RowLayout extends StatelessWidget {

  const RowLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Row Layout')),

      body: Container(

        color: Colors.yellow[50],

        child: Center(

          child: Container(

            width: 300,

            padding: EdgeInsets.all(16),

            decoration: BoxDecoration(

              color: Colors.yellowAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: Row(

              mainAxisAlignment: MainAxisAlignment.spaceEvenly,

              children: [

                Text('Child 1', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 2', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 3', style: TextStyle(fontSize: 20, color: Colors.white)),

              ],

            ),

          ),

        ),

      ),

    );

  }

}

class StackLayout extends StatelessWidget {

  const StackLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Stack Layout')),

      body: Container(

        color: Colors.orange[50],

        child: Center(

          child: Container(

            width: 250,

            height: 250,

            decoration: BoxDecoration(

              color: Colors.orangeAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: Stack(

              alignment: Alignment.center,

              children: [

                Container(width: 150, height: 150, color: Colors.blue),

                Container(width: 100, height: 100, color: Colors.red),

                Container(width: 50, height: 50, color: Colors.purple),

              ],

            ),

          ),

        ),

      ),

    );

  }

}

class GridLayout extends StatelessWidget {

  const GridLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Grid Layout')),

      body: Container(

        color: Colors.blue[50],

        child: Center(

          child: Container(

            width: 300,

            height: 300,

            padding: EdgeInsets.all(8),

            decoration: BoxDecoration(

              color: Colors.blueAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: GridView.builder(

              itemCount: 4,

              gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(

                crossAxisCount: 2,

                crossAxisSpacing: 8,

                mainAxisSpacing: 8,

              ),

              itemBuilder: (context, index) {

                return Container(

                  decoration: BoxDecoration(

                    color: Colors.blueGrey,

                    borderRadius: BorderRadius.circular(10),

                  ),

                  child: Center(

                    child: Text(

                      'Item $index',

                      style: TextStyle(fontSize: 20, color: Colors.white),

                    ),

                  ),

                );

              },

            ),

          ),

        ),

      ),

    );

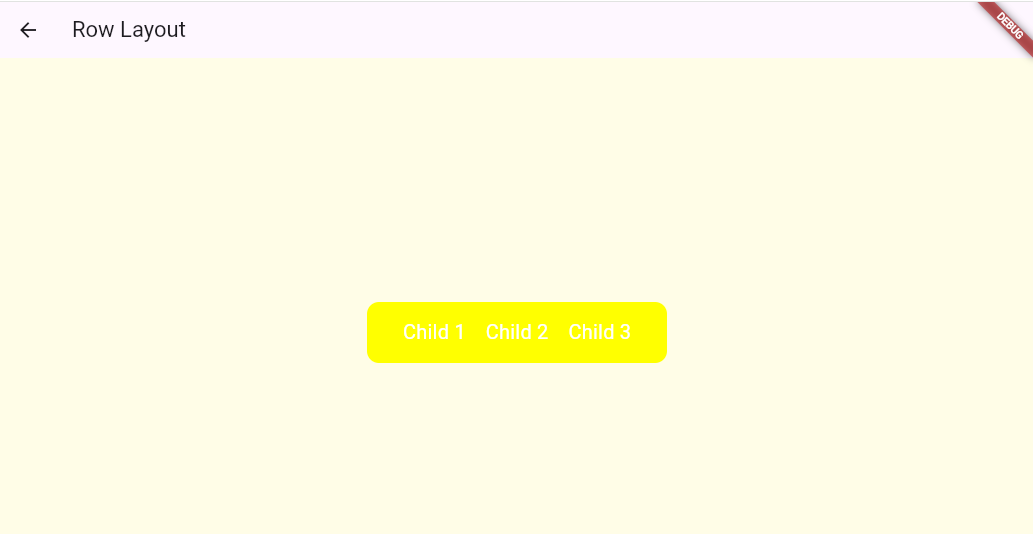
  }

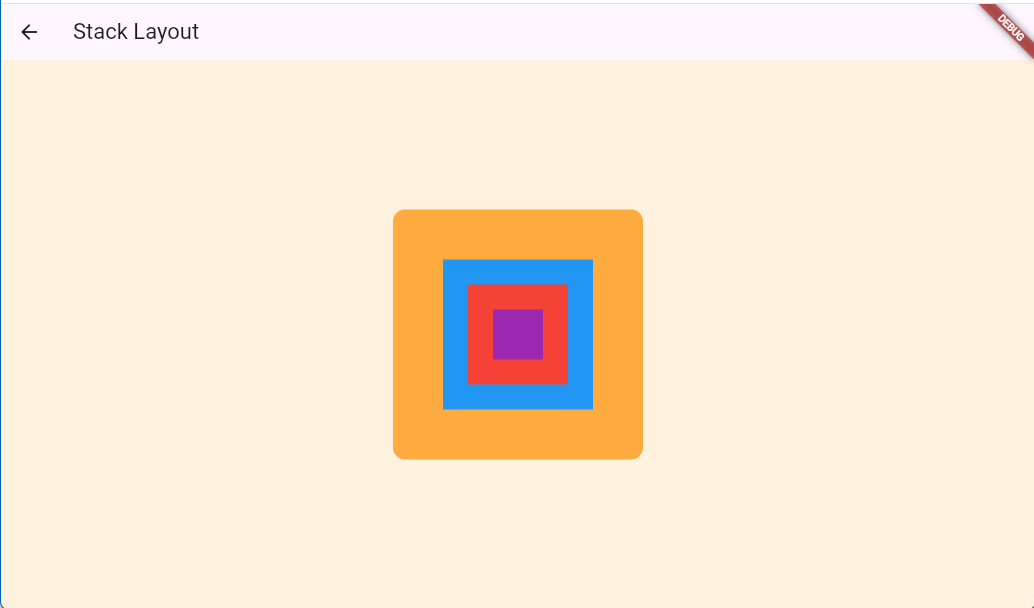
}

**Output:**











**Q1) Demonstrate the Use of Flatter UI Components.**

**Code:**

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'To-do App',

      theme: ThemeData(primaryColor: Colors.blue),

      home:TodoApp(),

    );

  }

}

class TodoApp extends StatefulWidget {

  @override

  \_TodoAppState createState()=>\_TodoAppState();

}

class \_TodoAppState extends State<TodoApp> {

  List<Map<String,dynamic>> \_tasks=[];

  int \_selectedIndex=0;

  final TextEditingController \_taskController=TextEditingController();

  void \_addTask(){

    if(\_taskController.text.isNotEmpty){

      setState(() {

        \_tasks.add({"title": \_taskController.text,"completed":false});

      });

      \_taskController.clear();

      Navigator.pop(context);

    }

  }

  void \_toggleTask(int index){

    setState(() {

      \_tasks[index]["completed"]= !\_tasks[index]["completed"];

    });

  }

  void \_deleteTask(int index){

    setState(() {

      \_tasks.removeAt(index);

    });

  }

  void \_showAddTaskDialog(){

    showDialog(

      context: context,

      builder:(context){

        return AlertDialog(

          title: Text("Add New Task"),

          content: TextField(

            controller: \_taskController,

            decoration: InputDecoration(hintText:"Enter task"),

          ),

          actions:[

            TextButton(

              child: Text("cancel"),

              onPressed:(){

                \_taskController.clear();

                Navigator.pop(context);

              },

            ),

            ElevatedButton(

              child: Text("Add"),

              onPressed: \_addTask,

            ),

          ],

        );

      },

    );

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text ("To-Do App")),

      drawer: Drawer(

        child: ListView(

          padding: EdgeInsets.zero,

          children: [

            DrawerHeader(

              decoration: BoxDecoration(color: Colors.blue),

              child:Text("Menu",style: TextStyle(fontSize: 24,color:Colors.white)),

            ),

            ListTile(

              leading: Icon(Icons.home),

              title:Text("Home"),

              onTap: ()=>Navigator.pop(context),

            ),

            ListTile(

              leading: Icon(Icons.settings),

              title:Text("Settings"),

              onTap: () =>Navigator.pop(context),

            ),

          ],

        ),

      ),

      body:\_selectedIndex ==0

      ? (\_tasks.isEmpty

        ? Center(child: Text("No tasks yet! Add a new task."))

        : ListView.builder(

            itemCount: \_tasks.length,

            itemBuilder: (context, index) {

              return Card(

                child: ListTile(

                  leading: IconButton(

                    icon: Icon(

                      \_tasks[index]["completed"]?Icons.check\_circle:Icons.radio\_button\_unchecked,

                      color:\_tasks[index]["completed"]?Colors.green :Colors.grey,

                      ),

                      onPressed: () =>\_toggleTask(index),

                    ),

                    title:Text(

                      \_tasks[index]["title"],

                      style: TextStyle(

                        decoration: \_tasks[index]["completed"]

                        ? TextDecoration.lineThrough

                        : TextDecoration.none,

                      ),

                    ),

                    trailing: IconButton(

                    icon: Icon(Icons.delete, color: Colors.red),

                    onPressed: () => \_deleteTask(index),

                  ),

                ),

              );

            },

        ))

      : Center(child:Text("Settings page")),

    floatingActionButton: FloatingActionButton(

      onPressed: \_showAddTaskDialog,

      child:Icon(Icons.add),

    ),

    bottomNavigationBar: BottomNavigationBar(

      items: [

        BottomNavigationBarItem(icon: Icon(Icons.list), label: "Tasks"),

        BottomNavigationBarItem(icon: Icon(Icons.settings), label: "Settings"),

      ],

      currentIndex: \_selectedIndex,

      onTap: (index){

        setState(() {

          \_selectedIndex = index;

        });

      },

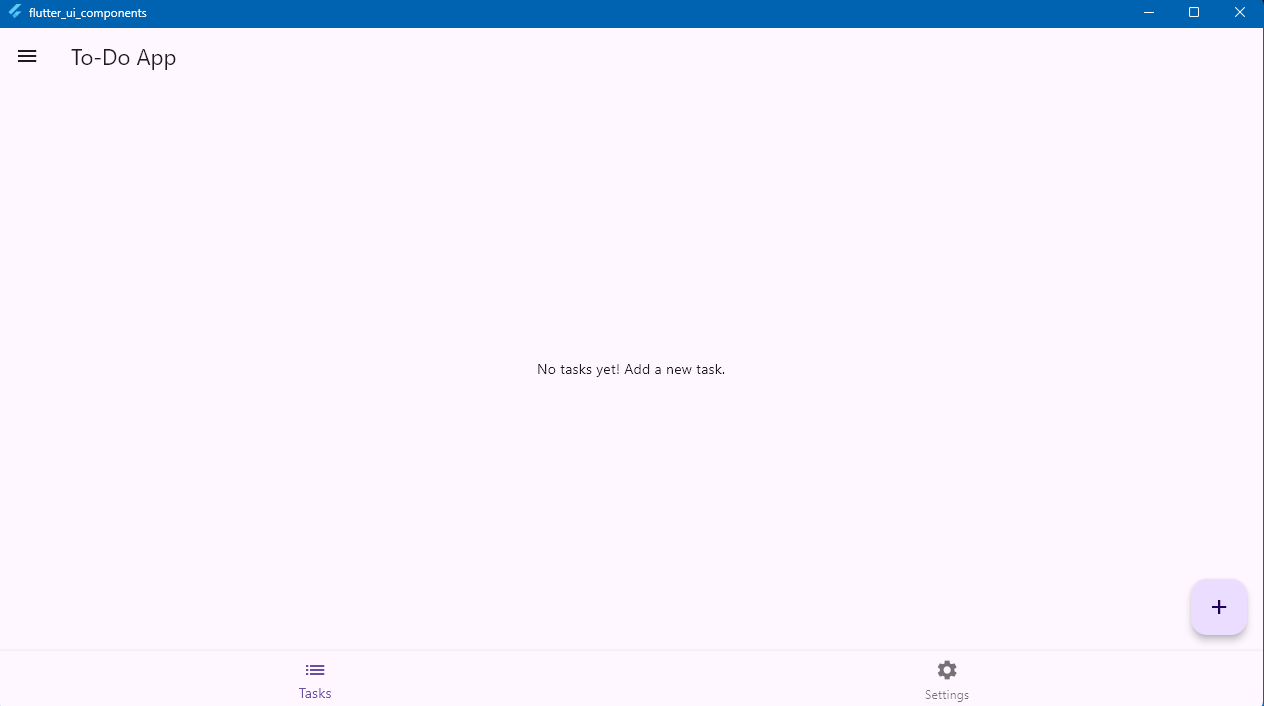
      ),

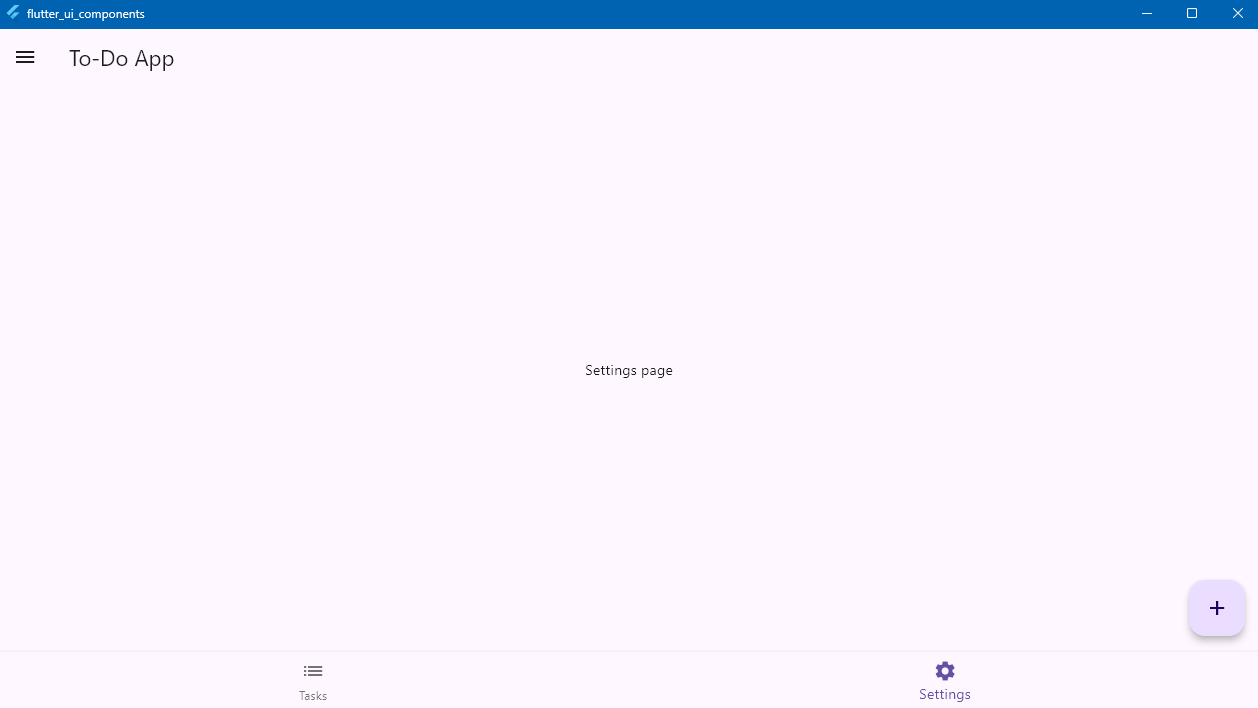
      );

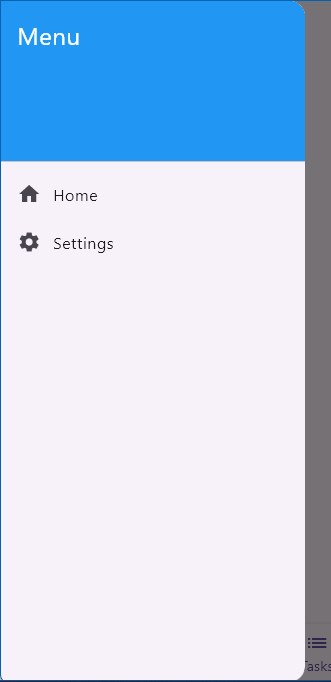
  }

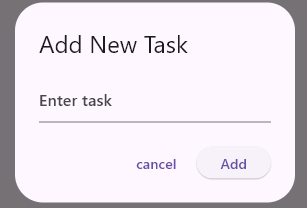
}

**Output:**

****

****

****

****

****

Q)Demonstrate the use of colours and fonts & Images and assets.

Code:

Main.dart

import 'package:flutter/material.dart';

import 'package:google\_fonts/google\_fonts.dart';

import 'package:carousel\_slider/carousel\_slider.dart';

import 'package:flutter\_colorpicker/flutter\_colorpicker.dart';

void main() {

  runApp(BasicDemoApp());

}

class BasicDemoApp extends StatefulWidget {

  @override

  \_BasicDemoAppState createState() => \_BasicDemoAppState();

}

class \_BasicDemoAppState extends State<BasicDemoApp> {

  Color selectedColor = Colors.white;

  void \_pickColor(BuildContext context) {

    showDialog(

      context: context,

      builder: (context) {

        return AlertDialog(

          title: Text('Pick a color'),

          content: SingleChildScrollView(

            child: ColorPicker(

              pickerColor: selectedColor,

              onColorChanged: (color) {

                setState(() {

                  selectedColor = color;

                });

              },

              enableAlpha: false,

              showLabel: true,

            ),

          ),

          actions: <Widget>[

            TextButton(

              child: Text('Select'),

              onPressed: () {

                Navigator.of(context).pop();

              },

            ),

          ],

        );

      },

    );

  }

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Image Carousel App',

      theme: ThemeData(

        primarySwatch: Colors.pink,

        textTheme: GoogleFonts.lobsterTextTheme(),

        scaffoldBackgroundColor: selectedColor,

      ),

      home: BasicDemoScreen(onColorPick: () => \_pickColor(context)),

    );

  }

}

class BasicDemoScreen extends StatelessWidget {

  final VoidCallback onColorPick;

  BasicDemoScreen({required this.onColorPick});

  final List<String> imagePaths = [

    'assets/1.png',

    'assets/2.png',

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text('Assets Usage',

            style: GoogleFonts.lobster(

                fontWeight: FontWeight.bold, color: Colors.white)),

        backgroundColor: Colors.deepOrange,

        elevation: 5,

        shadowColor: Colors.black45,

        actions: [

          IconButton(

            icon: Icon(Icons.color\_lens),

            onPressed: onColorPick,

          )

        ],

      ),

      body: Container(

        padding: EdgeInsets.all(20),

        decoration: BoxDecoration(

          gradient: LinearGradient(

            colors: [Colors.orangeAccent, Colors.yellowAccent],

            begin: Alignment.topLeft,

            end: Alignment.bottomRight,

          ),

          borderRadius: BorderRadius.circular(20),

        ),

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          crossAxisAlignment: CrossAxisAlignment.center,

          children: [

            Text(

              'Add Your Roll No.!',

              style: GoogleFonts.poppins(

                fontSize: 28,

                fontWeight: FontWeight.bold,

                color: Colors.white,

              ),

              textAlign: TextAlign.center,

            ),

            SizedBox(height: 20),

            CarouselSlider(

              options: CarouselOptions(

                height: 200,

                autoPlay: true,

                enlargeCenterPage: true,

                aspectRatio: 16 / 9,

                enableInfiniteScroll: true,

              ),

              items: imagePaths.map((imagePath) {

                return ClipRRect(

                  borderRadius: BorderRadius.circular(15),

                  child: Image.asset(

                    imagePath,

                    width: double.infinity,

                    fit: BoxFit.cover,

                  ),

                );

              }).toList(),

            ),

            SizedBox(height: 20),

            ElevatedButton(

              onPressed: () {},

              child: Text('Explore More',

                  style: GoogleFonts.poppins(

                      fontSize: 18, fontWeight: FontWeight.w600)),

              style: ElevatedButton.styleFrom(

                padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),

                backgroundColor: Colors.deepOrange,

                shape: RoundedRectangleBorder(

                  borderRadius: BorderRadius.circular(30),

                ),

              ),

            ),

          ],

        ),

      ),

    );

  }

}

Pubspec.yml

name: colorsfontsdemo

description: "A new Flutter project."

publish\_to: "none"

version: 1.0.0+1

environment:

  sdk: ^3.6.1

dependencies:

  flutter:

    sdk: flutter

  flutter\_colorpicker: any

  google\_fonts: ^6.1.0

  carousel\_slider: ^5.0.0

  cupertino\_icons: ^1.0.8

dev\_dependencies:

  flutter\_test:

    sdk: flutter

  flutter\_lints: ^5.0.0

flutter:

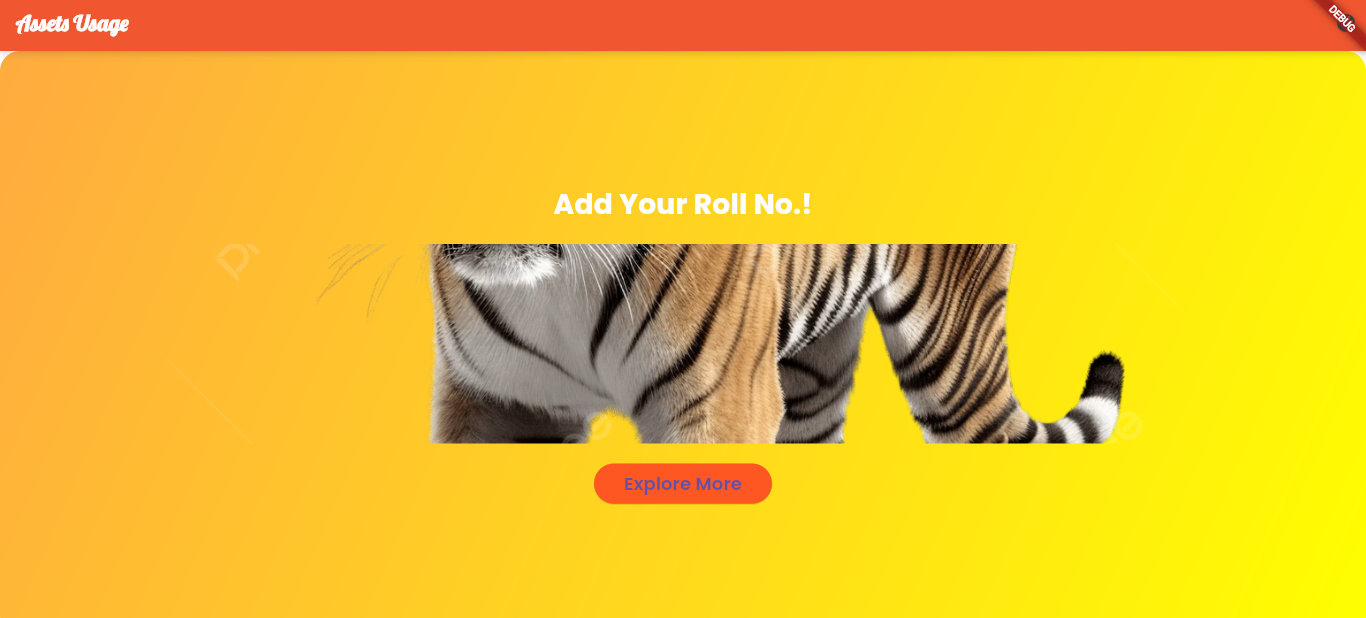
  uses-material-design: true

  assets:

    - assets/1.png

    - assets/2.png

Output:



**Q) Demonstrate a flutter program for state management**

**Code :**

import 'package:flutter/material.dart';

void main() {

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Flutter Demo',

      theme: ThemeData(

        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),

        useMaterial3: true,

      ),

      home: MyWidget(),

    );

  }

}

class MyWidget extends StatelessWidget {

  MyWidget({super.key});

  final List<Map<String, dynamic>> layout = [

    {'title': 'Column', 'widget': ColumnLayout(), 'color': Colors.green},

    {'title': 'Row', 'widget': RowLayout(), 'color': Colors.yellow},

    {'title': 'Stack', 'widget': StackLayout(), 'color': Colors.orange},

    {'title': 'Grid', 'widget': GridLayout(), 'color': Colors.blue},

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text('Flutter Widget'),

      ),

      body: ListView.builder(

        itemCount: layout.length,

        itemBuilder: (context, index) {

          return Container(

            margin: EdgeInsets.symmetric(vertical: 5, horizontal: 10),

            decoration: BoxDecoration(

              color: layout[index]['color'], // Set different colors for each item

              borderRadius: BorderRadius.circular(10),

            ),

            child: ListTile(

              title: Text(

                layout[index]['title'],

                style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold),

              ),

              trailing: Icon(Icons.arrow\_forward, color: Colors.white),

              tileColor: layout[index]['color'], // Apply background color

              shape: RoundedRectangleBorder(

                borderRadius: BorderRadius.circular(10),

              ),

              onTap: () {

                Navigator.push(

                  context,

                  MaterialPageRoute(

                    builder: (context) => layout[index]['widget'],

                  ),

                );

              },

            ),

          );

        },

      ),

    );

  }

}

class ColumnLayout extends StatelessWidget {

  const ColumnLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Column Layout')),

      body: Container(

        color: Colors.green[50],

        child: Center(

          child: Container(

            width: 300,

            padding: EdgeInsets.all(16),

            decoration: BoxDecoration(

              color: Colors.greenAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: Column(

              mainAxisAlignment: MainAxisAlignment.center,

              children: [

                Text('Child 1', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 2', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 3', style: TextStyle(fontSize: 20, color: Colors.white)),

              ],

            ),

          ),

        ),

      ),

    );

  }

}

class RowLayout extends StatelessWidget {

  const RowLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Row Layout')),

      body: Container(

        color: Colors.yellow[50],

        child: Center(

          child: Container(

            width: 300,

            padding: EdgeInsets.all(16),

            decoration: BoxDecoration(

              color: Colors.yellowAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: Row(

              mainAxisAlignment: MainAxisAlignment.spaceEvenly,

              children: [

                Text('Child 1', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 2', style: TextStyle(fontSize: 20, color: Colors.white)),

                Text('Child 3', style: TextStyle(fontSize: 20, color: Colors.white)),

              ],

            ),

          ),

        ),

      ),

    );

  }

}

class StackLayout extends StatelessWidget {

  const StackLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Stack Layout')),

      body: Container(

        color: Colors.orange[50],

        child: Center(

          child: Container(

            width: 250,

            height: 250,

            decoration: BoxDecoration(

              color: Colors.orangeAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: Stack(

              alignment: Alignment.center,

              children: [

                Container(width: 150, height: 150, color: Colors.blue),

                Container(width: 100, height: 100, color: Colors.red),

                Container(width: 50, height: 50, color: Colors.purple),

              ],

            ),

          ),

        ),

      ),

    );

  }

}

class GridLayout extends StatelessWidget {

  const GridLayout({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Grid Layout')),

      body: Container(

        color: Colors.blue[50],

        child: Center(

          child: Container(

            width: 300,

            height: 300,

            padding: EdgeInsets.all(8),

            decoration: BoxDecoration(

              color: Colors.blueAccent,

              borderRadius: BorderRadius.circular(12),

            ),

            child: GridView.builder(

              itemCount: 4,

              gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(

                crossAxisCount: 2,

                crossAxisSpacing: 8,

                mainAxisSpacing: 8,

              ),

              itemBuilder: (context, index) {

                return Container(

                  decoration: BoxDecoration(

                    color: Colors.blueGrey,

                    borderRadius: BorderRadius.circular(10),

                  ),

                  child: Center(

                    child: Text(

                      'Item $index',

                      style: TextStyle(fontSize: 20, color: Colors.white),

                    ),

                  ),

                );

              },

            ),

          ),

        ),

      ),

    );

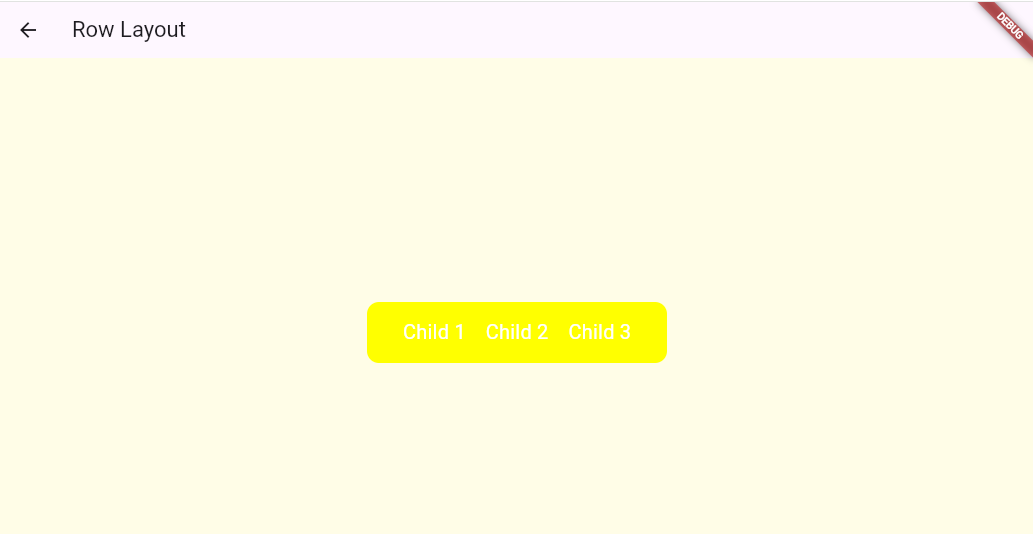
  }

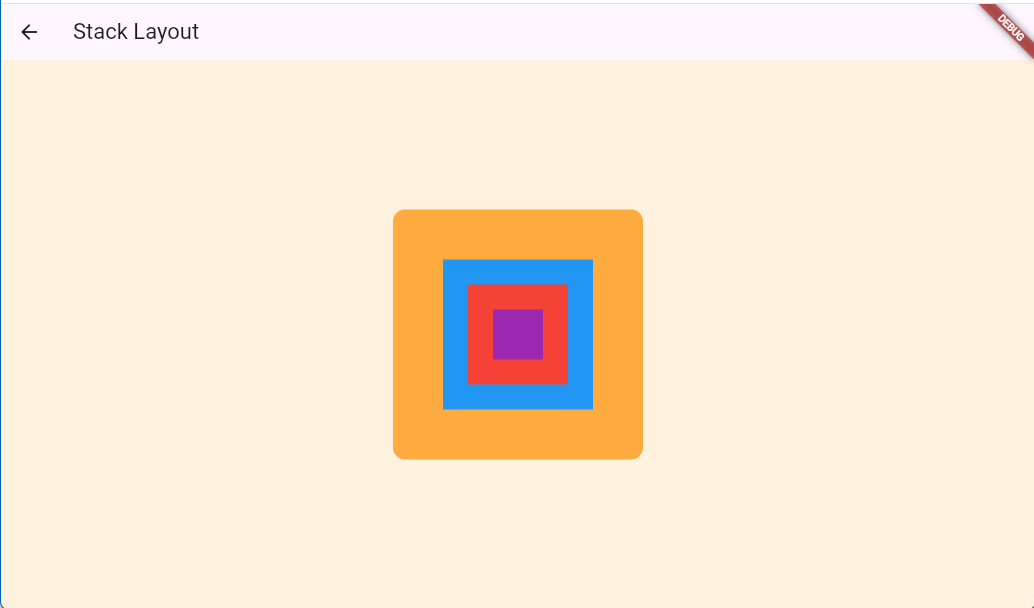
}

**Output:**











Q)Demonstrate a flutter program for implementing various designs and animationss

Code:

Main.dart

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'package:math\_expressions/math\_expressions.dart';

void main() {

  runApp(const CalculatorApp());

}

class CalculatorApp extends StatelessWidget {

  const CalculatorApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      theme: ThemeData.dark().copyWith(

        scaffoldBackgroundColor: Colors.black,

      ),

      home: const CalculatorScreen(),

    );

  }

}

class CalculatorScreen extends StatefulWidget {

  const CalculatorScreen({super.key});

  @override

  State<CalculatorScreen> createState() => \_CalculatorScreenState();

}

class \_CalculatorScreenState extends State<CalculatorScreen>

    with SingleTickerProviderStateMixin {

  String \_expression = "";

  String \_result = "0";

  late AnimationController \_controller;

  @override

  void initState() {

    super.initState();

    \_controller = AnimationController(

      duration: const Duration(milliseconds: 200),

      vsync: this,

      lowerBound: 0.9,

      upperBound: 1.0,

    );

  }

  void \_onButtonPressed(String value) {

    setState(() {

      if (value == "C") {

        \_expression = "";

        \_result = "0";

      } else if (value == "=") {

        try {

          Parser p = Parser();

          Expression exp = p.parse(\_expression);

          ContextModel cm = ContextModel();

          \_result = exp.evaluate(EvaluationType.REAL, cm).toString();

        } catch (e) {

          \_result = "Error";

        }

      } else {

        \_expression += value;

      }

    });

    \_controller.forward().then((\_) => \_controller.reverse());

  }

  Widget \_buildButton(String text, {Color? color}) {

    return ScaleTransition(

      scale: \_controller,

      child: ElevatedButton(

        onPressed: () => \_onButtonPressed(text),

        style: ElevatedButton.styleFrom(

          shape: const CircleBorder(),

          padding: const EdgeInsets.all(20),

          backgroundColor: color ?? Colors.grey[850],

        ),

        child: Text(

          text,

          style: const TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

        ),

      ),

    );

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: const Text("Flutter Calculator")),

      body: Column(

        mainAxisAlignment: MainAxisAlignment.end,

        children: [

          Padding(

            padding: const EdgeInsets.all(16.0),

            child: Align(

              alignment: Alignment.centerRight,

              child: Text(

                \_expression,

                style: const TextStyle(fontSize: 32, color: Colors.white70),

              ),

            ),

          ),

          Padding(

            padding: const EdgeInsets.all(16.0),

            child: Align(

              alignment: Alignment.centerRight,

              child: Text(

                \_result,

                style:

                    const TextStyle(fontSize: 48, fontWeight: FontWeight.bold),

              ),

            ),

          ),

          Expanded(

            child: GridView.count(

              padding: const EdgeInsets.all(16),

              crossAxisCount: 4,

              children: [

                \_buildButton("7"),

                \_buildButton("8"),

                \_buildButton("9"),

                \_buildButton("/", color: Colors.orange),

                \_buildButton("4"),

                \_buildButton("5"),

                \_buildButton("6"),

                \_buildButton("\*", color: Colors.orange),

                \_buildButton("1"),

                \_buildButton("2"),

                \_buildButton("3"),

                \_buildButton("-", color: Colors.orange),

                \_buildButton("0"),

                \_buildButton("C", color: Colors.red),

                \_buildButton("=", color: Colors.green),

                \_buildButton("+", color: Colors.orange),

              ],

            ),

          ),

        ],

      ),

    );

  }

}

Pubspec.yaml

name: calculator

description: "A new Flutter project."

publish\_to: "none"

version: 1.0.0+1

environment:

  sdk: ^3.6.1

dependencies:

  flutter:

    sdk: flutter

  cupertino\_icons: ^1.0.8

  math\_expressions: ^2.6.0

dev\_dependencies:

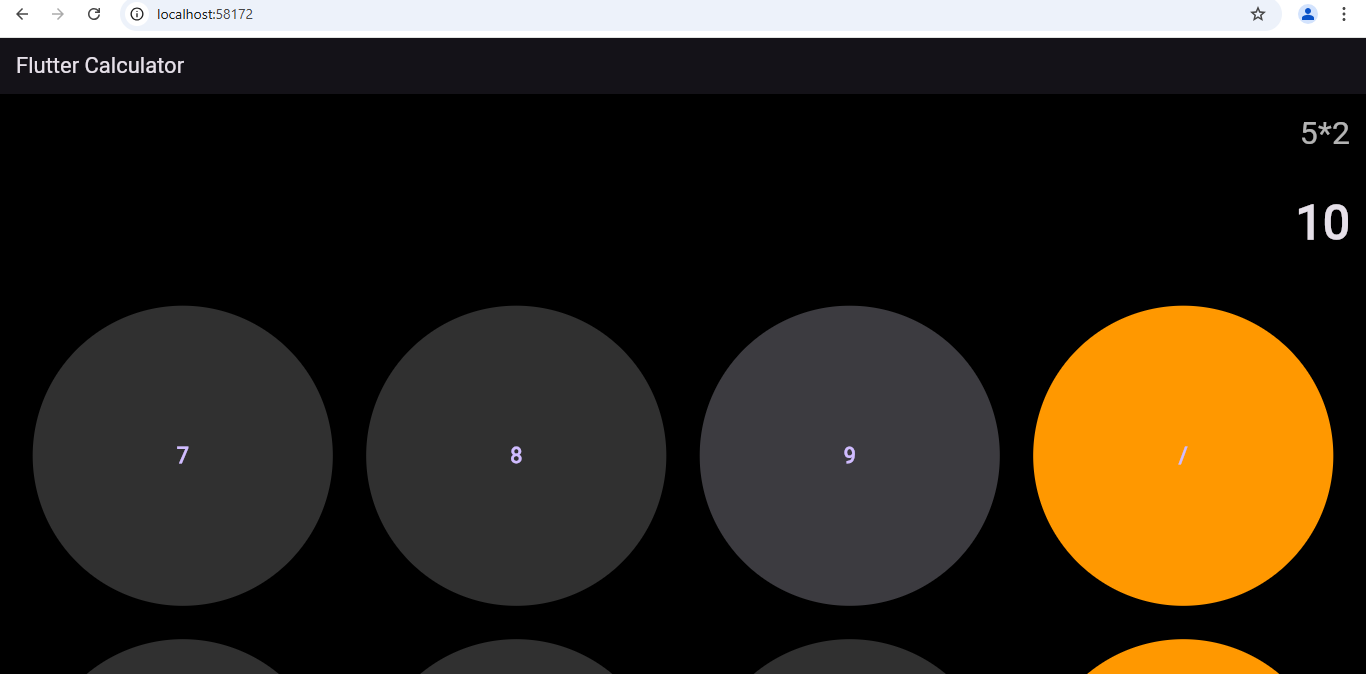
  flutter\_test:

    sdk: flutter

  flutter\_lints: ^5.0.0

flutter:

  uses-material-design: true



Q)Demonstrate a flutter program for implementing form validation.

Code:

import 'package:flutter/material.dart';

void main() {

  runApp(const FormValidationApp());

}

class FormValidationApp extends StatelessWidget {

  const FormValidationApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      theme: ThemeData(primarySwatch: Colors.blue),

      home: const FormScreen(),

    );

  }

}

class FormScreen extends StatefulWidget {

  const FormScreen({super.key});

  @override

  State<FormScreen> createState() => \_FormScreenState();

}

class \_FormScreenState extends State<FormScreen> {

  final \_formKey = GlobalKey<FormState>();

  final TextEditingController \_emailController = TextEditingController();

  final TextEditingController \_passwordController = TextEditingController();

  void \_submitForm() {

    if (\_formKey.currentState!.validate()) {

      ScaffoldMessenger.of(context).showSnackBar(

        const SnackBar(content: Text("Form submitted successfully!")),

      );

    } else {

      \_emailController.clear();

      \_passwordController.clear();

    }

  }

  @override

Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: const Text("Form Validation Example")),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Form(

          key: \_formKey,

          child: Column(

            children: [

              TextFormField(

                controller: \_emailController,

                decoration: const InputDecoration(labelText: "Email"),

                keyboardType: TextInputType.emailAddress,

                validator: (value) {

                  if (value == null || value.isEmpty) {

                    return "Please enter an email";

                  } else if (!RegExp(

                          r'^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$')

                      .hasMatch(value)) {

                    return "Enter a valid email";

                  }

                  return null;

                },

              ),

              const SizedBox(height: 16),

              TextFormField(

                controller: \_passwordController,

                decoration: const InputDecoration(labelText: "Password"),

                obscureText: true,

                validator: (value) {

                  if (value == null || value.isEmpty) {

                    return "Please enter a password";

                  } else if (value.length < 6) {

                    return "Password must be at least 6 characters";

                  }

                  return null;

                },

              ),

              const SizedBox(height: 24),

              ElevatedButton(

                onPressed: \_submitForm,

                child: const Text("Submit"),

              ),

            ],

          ),

        ),

      ),

    );

  }

}

Output:

