

S. NO	Programs
1.	Flutter GUI Components
2.	Flutter Layout Design
3.	Flutter Event Handling
4.	Basic graphical primitives
5.	notification Manager
6.	Application using Animation
7.	Application using multi-threading
8.	Message Alert Application
9.	Simple Game Application using Flutter
10.	Simple Calculator using Flutter

1. Flutter GUI Components

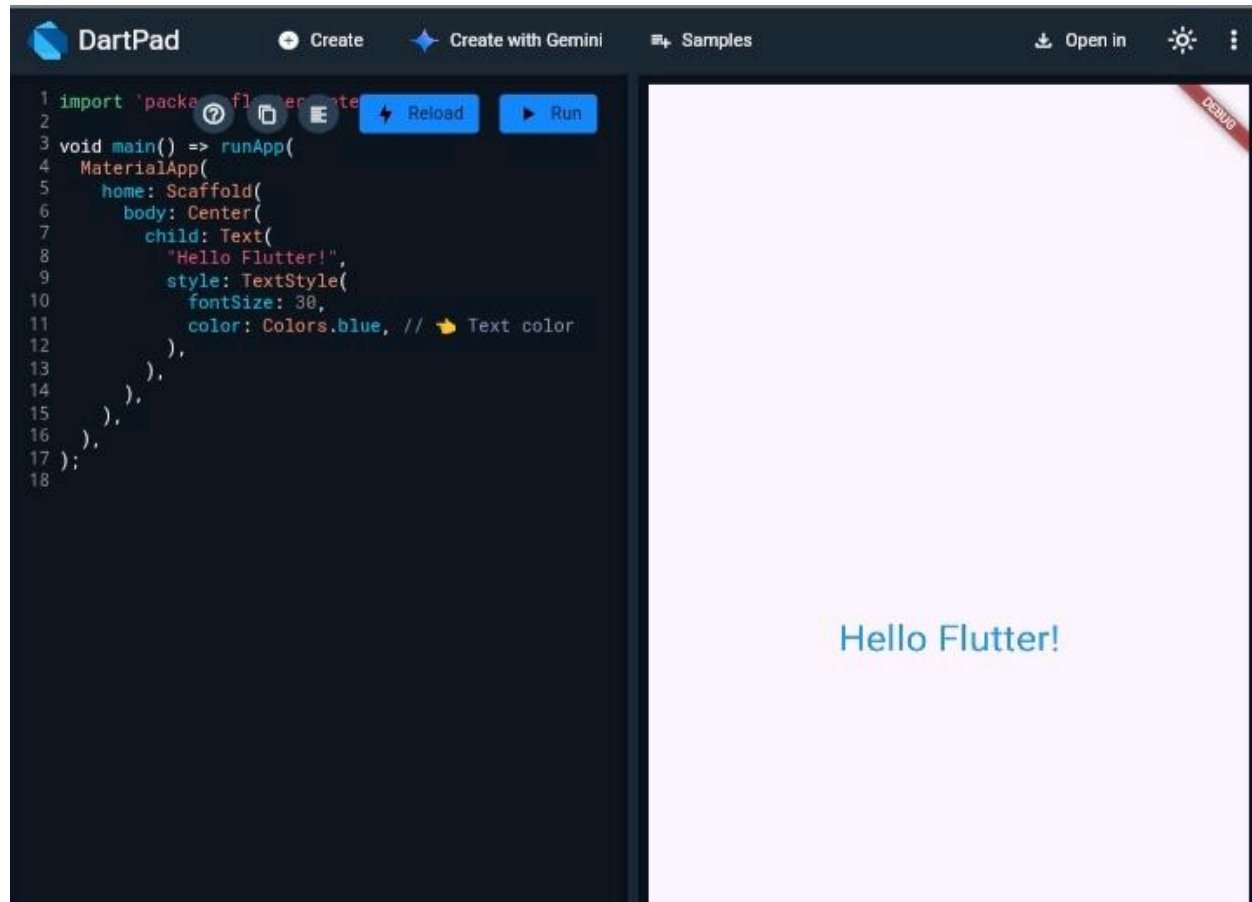
Aim:

To develop a Flutter application using basic GUI components with fonts and colors.

Program:

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

void main() => runApp(
  MaterialApp(
    home: Scaffold(
      body: Center(
        child: Text(
          "Hello Flutter!",
          style: TextStyle(
            fontSize: 30,
            color: Colors.blue, // 👉 Text color
          ),
        ),
      ),
    ),
  ),
);
```



2.Flutter Layout Design

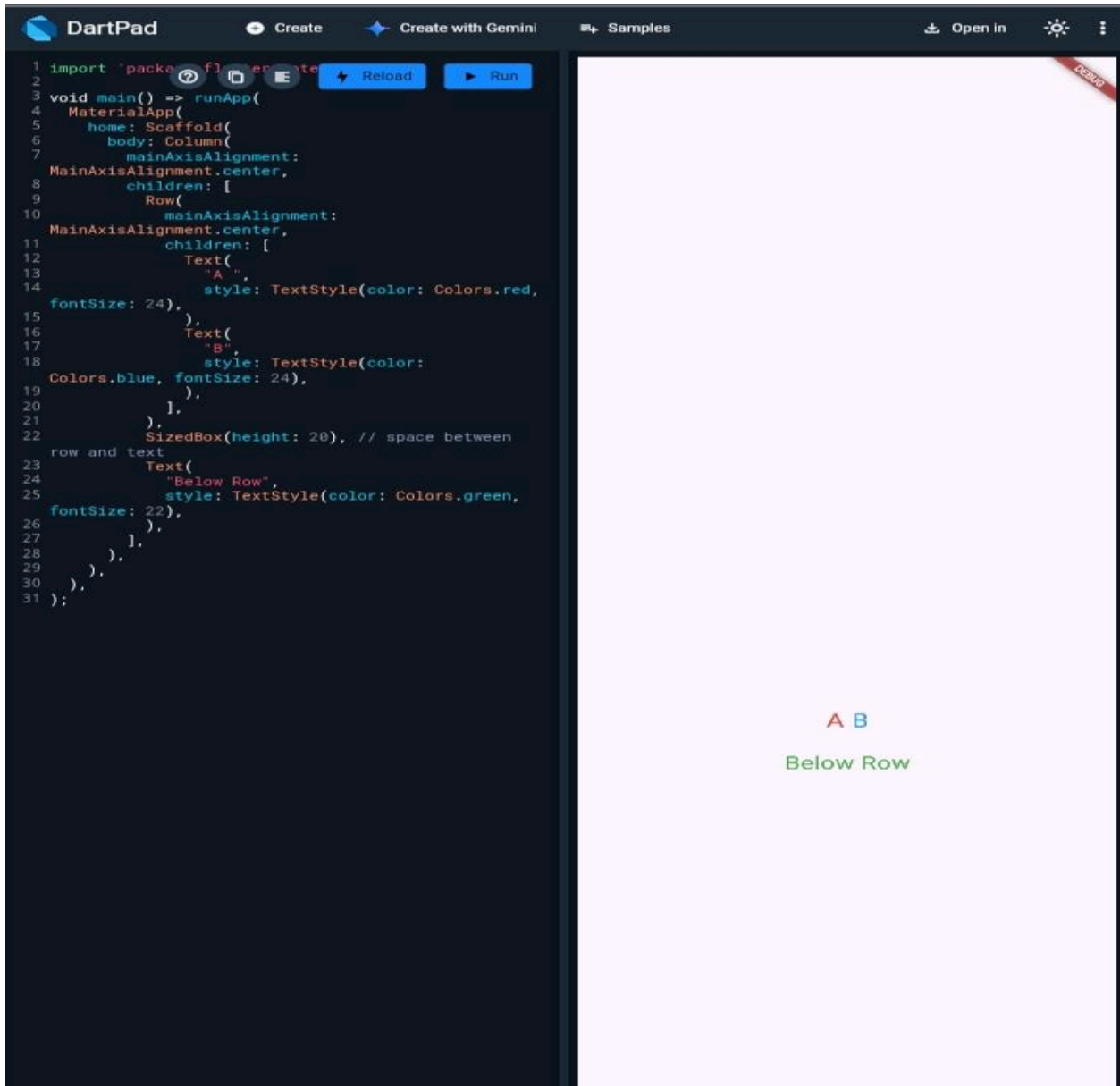
Aim:

To develop an application using layout managers.

Program:

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

void main() => runApp(
  MaterialApp(
    home: Scaffold(
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                "A ",
                style: TextStyle(color: Colors.red, fontSize: 24),
              ),
              Text(
                "B",
                style: TextStyle(color: Colors.blue, fontSize: 24),
              ),
            ],
          ),
          SizedBox(height: 20), // space between row and text
          Text(
            "Below Row",
            style: TextStyle(color: Colors.green, fontSize: 22),
          ),
        ],
      ),
    ),
  );
```



3.Flutter Event Handling

Aim:

To develop a Flutter application using event listeners.

Program:

```
import 'package:flutter/material.dart';
int _i = 0;
void main() => runApp(
  MaterialApp(
    home: Scaffold(
      body: Center(
        child: StatefulBuilder(
          builder: (c, set) => Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              Text("Tap Count: $_i"),
              ElevatedButton(
                onPressed: () => set(() => _i++),
                child: Text("Tap"),
              ),
            ],
          ),
        ),
      ),
    ),
  );
```

```

1 import 'package:flutter/material.dart';
2
3 int i = 0;
4
5 void main() => runApp(MaterialApp(
6   home: Scaffold(
7     body: Center(
8       child: StatefulBuilder(
9         builder: (c, set) => Column(
10          mainAxisAlignment: MainAxisAlignment.min,
11          children: [
12            Text("Tap Count: $i"),
13            ElevatedButton(
14              onPressed: () => set(() => i++),
15              child: Text("Tap"),
16            ),
17          ],
18        ),
19      ),
20    ),
21  ));
22
23

```

Tap Count: 9

Tap

4.Basic graphical primitives

Aim:

To develop an application for drawing basic graphical primitives on the screen.

Program:

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

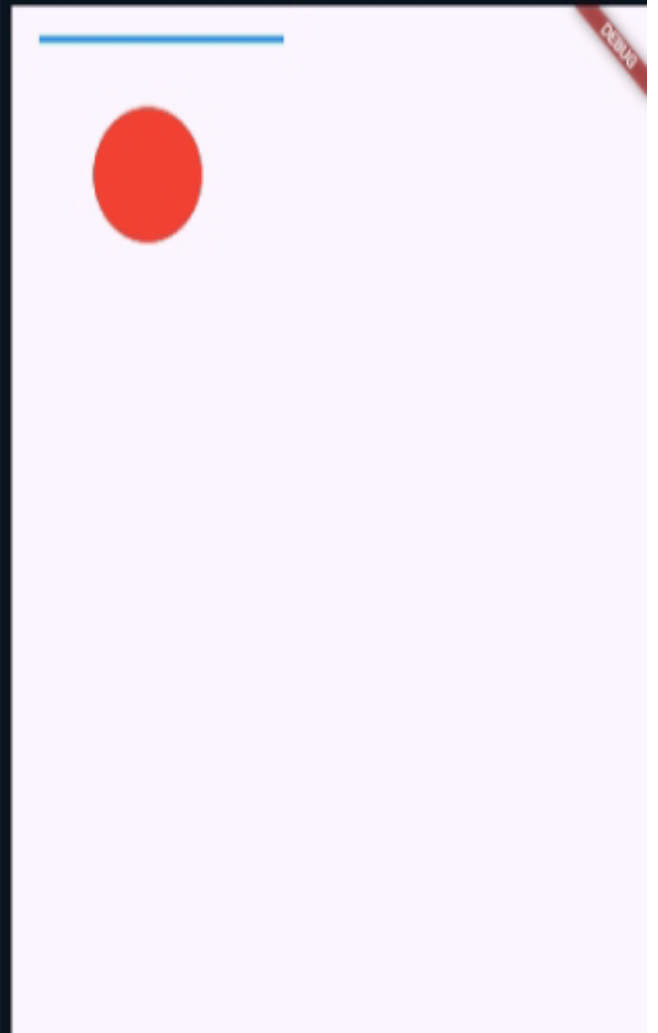
void main() => runApp(
  MaterialApp(
    home: Scaffold(
      body: CustomPaint(painter: MyDraw(), child: Container()),
    ),
  ),
);

class MyDraw extends CustomPainter {
  @override
  void paint(Canvas c, Size s) {
    var p = Paint()
      ..color = Colors.blue
      ..strokeWidth = 4;
    c.drawLine(Offset(20, 20), Offset(200, 20), p);
    c.drawCircle(Offset(100, 100), 40, p..color = Colors.red);
  }

  @override
  bool shouldRepaint(covariant CustomPainter old) => false;
}
```


Qpi

```
1 import 'package:flutter/material'
2
3 void main() => runApp(
4   MaterialApp(
5     home: Scaffold(
6       body: CustomPaint(painter: MyDraw(), child:
7         Container()),
8     ),
9 );
10
11 class MyDraw extends CustomPainter {
12   @override
13   void paint(Canvas c, Size s) {
14     var p = Paint()
15       ..color = Colors.blue
16       ..strokeWidth = 4;
17     c.drawLine(Offset(20, 20), Offset(200, 20), p);
18     c.drawCircle(Offset(100, 100), 40, p..color =
19       Colors.red);
20   }
21   @override
22   bool shouldRepaint(covariant CustomPainter old)
23     => false;
24 }
```



5.Notification Manager

Aim:

To develop an application using notification Manager.

Program:

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


void main() => runApp(MyApp());

class MyApp extends StatefulWidget {
  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  String message = "";
  void showNotification() {
    setState(() {
      message = "Notification: You pressed the button!";
    });
  }
}

@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(title: Text("Notification Manager (Simulated)")),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            ElevatedButton(
              onPressed: showNotification,
              child: Text("Show Notification"),
            ),
            SizedBox(height: 20),
            Text(message, style: TextStyle(fontSize: 18, color: Colors.blue)),
          ],
        ),
      ),
    ),
  ),
)
```

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

 DartPad

Create

Create with Gemini

Samples

Open in

import 'package:flutter/material.dart';
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
 // This widget is the root of your application.
 State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
 String message = '';
 void showNotification() {
 setState(() {
 message = 'Notification: You pressed the button!';
 });
 }

 @override
 Widget build(BuildContext context) {
 return MaterialApp(
 home: Scaffold(
 appBar: AppBar(title: Text('Notification Manager (Simulated)')),
 body: Center(
 child: Column(
 mainAxisAlignment: MainAxisAlignment.min,
 children: [
 ElevatedButton(
 onPressed: showNotification,
 child: Text('Show Notification'),
),
 SizedBox(height: 20),
 Text(message, style:
 TextStyle(fontSize: 18, color: Colors.blue)),
],
),
),
),
);
 }
}
}

Notification Manager (Simulated)

Show Notification

Notification: You pressed the button!

6. Application using Animation

Aim:

To develop an Application using Animation

Program:

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

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: AnimationDemo(),
    );
  }
}

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

  @override
  State<AnimationDemo> createState() => _AnimationDemoState();
}

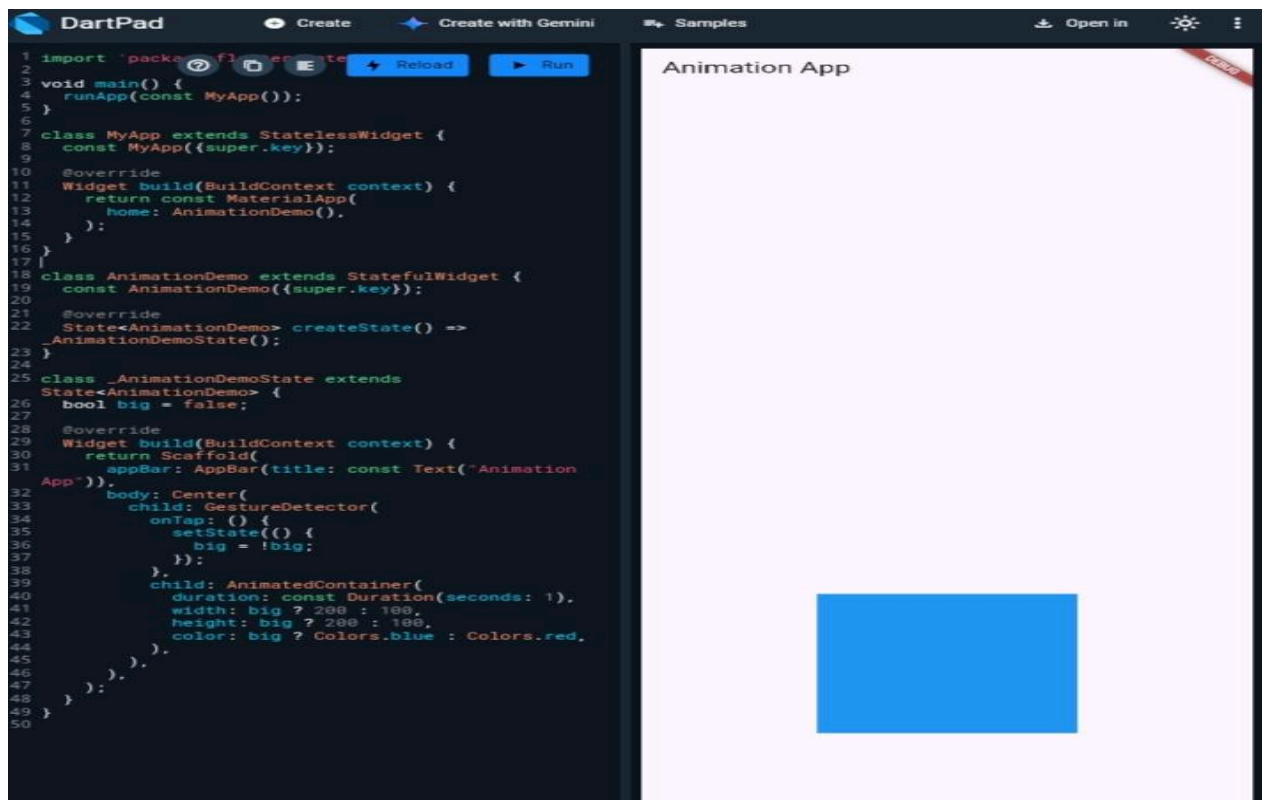
class _AnimationDemoState extends State<AnimationDemo> {
  bool big = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Animation App")),
      body: Center(
        child: GestureDetector(
```

```

onTap: () {
  setState(() {
    big = !big;
  });
},
child: AnimatedContainer(
  duration: const Duration(seconds: 1),
  width: big ? 200 : 100,
  height: big ? 200 : 100,
  color: big ? Colors.blue : Colors.red,
),
),
),
);
}
}

```



7. Application using multi-threading

Aim:

To develop an application using multi-threading

Program:

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

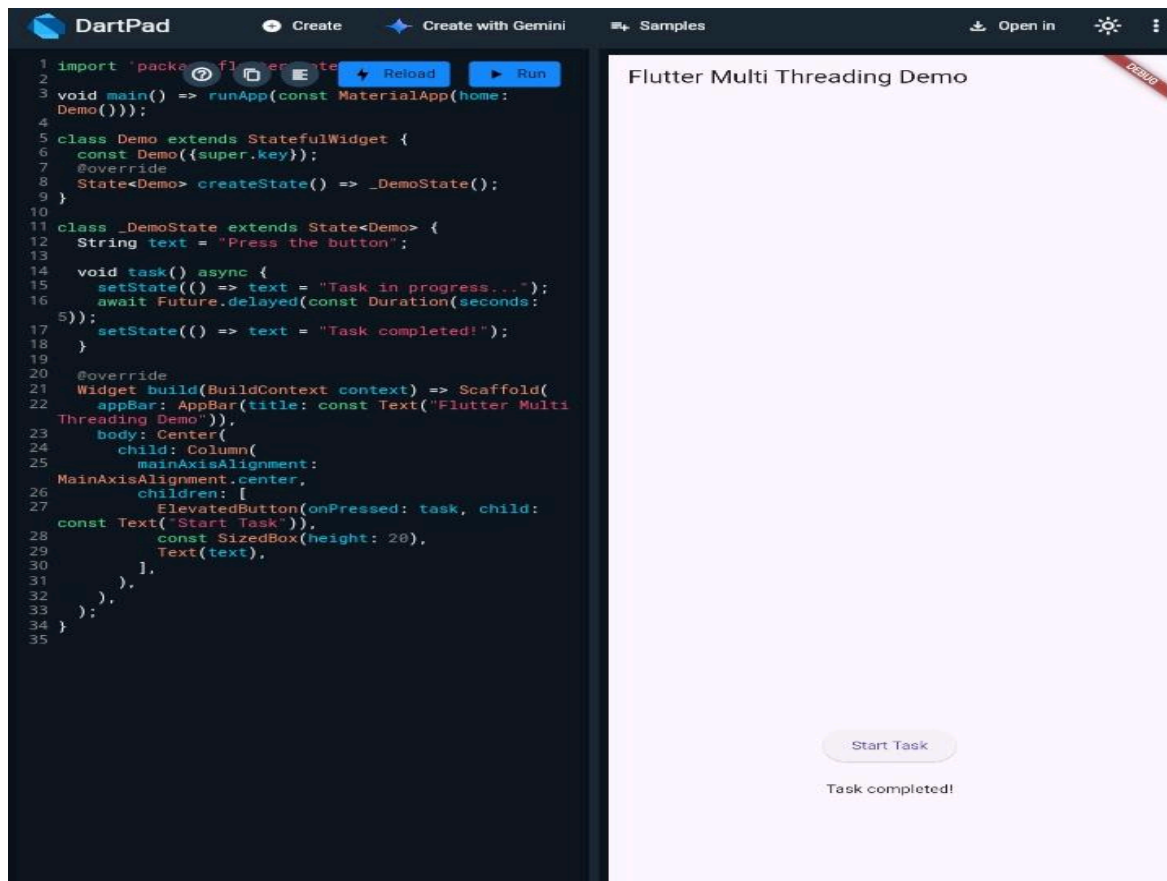
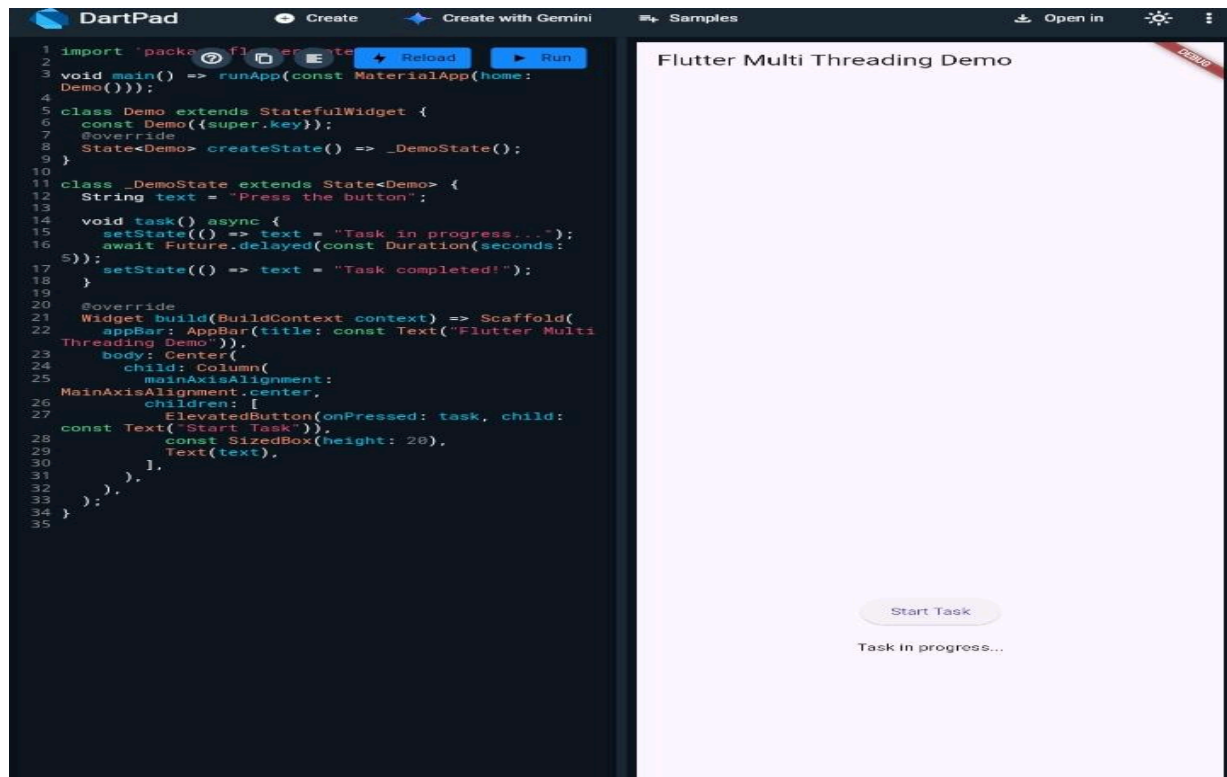
void main() => runApp(const MaterialApp(home: Demo()));

class Demo extends StatefulWidget {
  const Demo({super.key});
  @override
  State<Demo> createState() => _DemoState();
}

class _DemoState extends State<Demo> {
  String text = "Press the button";

  void task() async {
    setState(() => text = "Task in progress...");
    await Future.delayed(const Duration(seconds: 5));
    setState(() => text = "Task completed!");
  }

  @override
  Widget build(BuildContext context) => Scaffold(
    appBar: AppBar(title: const Text("Flutter Multi Threading Demo")),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          ElevatedButton(onPressed: task, child: const Text("Start Task")),
          const SizedBox(height: 20),
          Text(text),
        ],
      ),
    ),
  );
```



8. Message Alert Application

Aim:

To develop an application for creating an alert upon receiving a message.

Program:

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

void main() => runApp(const App());

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(home: Demo());
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Alert Demo")),
      body: Center(
        child: ElevatedButton(
          child: const Text("Show Alert"),
          onPressed: () {
            showDialog(
              context: context,
              builder: (_) => AlertDialog(

title: const Text("Message"),
                content: const Text("New message received"),
                actions: [
                  TextButton(
                    onPressed: () => Navigator.pop(context),
                    child: const Text("OK"),
```

```

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

```

Create
Create with Gemini
Samples
Open in

```

1 import 'package:flutter/material.dart';
2
3 void main() => runApp(const App());
4
5 class App extends StatelessWidget {
6   const App({super.key});
7
8   @override
9   Widget build(BuildContext context) {
10     return const MaterialApp(home: Demo());
11   }
12 }
13
14 class Demo extends StatelessWidget {
15   const Demo({super.key});
16
17   @override
18   Widget build(BuildContext context) {
19     return Scaffold(
20       appBar: AppBar(title: const Text("Alert Demo")),
21       body: Center(
22         child: ElevatedButton(
23           child: const Text("Show Alert"),
24           onPressed: () {
25             showDialog(
26               context: context,
27               builder: (_) => AlertDialog(
28                 title: const Text("Message"),
29                 content: const Text("New message received"),
30                 actions: [
31                   TextButton(
32                     onPressed: () =>
33                       Navigator.pop(context),
34                     child: const Text("OK"),
35                   ),
36                 ],
37               );
38             );
39             );
40             );
41             );
42           );
43         );
44       );
45     );
46   }
47 }

```

Alert Demo

The screenshot shows a mobile application interface with a light purple background. At the bottom center, there is a white button with rounded corners and a thin grey border, containing the text "Show Alert".

DartPad

Create

Create with Gemini

Samples

Open in

1import 'package:flutter/material.dart';

2

3void main() => runApp(const App());

4

5class App extends StatelessWidget {

6 const App({super.key});

7

8 @override

9 Widget build(BuildContext context) {

10 return const MaterialApp(home: Demo());

11 }

12}

13

14class Demo extends StatelessWidget {

15 const Demo({super.key});

16

17 @override

18 Widget build(BuildContext context) {

19 return Scaffold(

20 appBar: AppBar(title: const Text("Alert

21 Demo")),

22 body: Center(

23 child: ElevatedButton(

24 child: const Text("Show Alert"),

25 onPressed: () {

26 showDialog(

27 context: context,

28 builder: (_) => AlertDialog(

29 title: const Text("Message"),

30 content: const Text("New message

31 received"),

32 actions: [

33 TextButton(

34 onPressed: () =>

35 Navigator.pop(context),

36 child: const Text("OK"),

37),

38),

39),

40),

41),

42),

43),

44);

Alert Demo

Message

New message received

OK

9.Simple Game Application using Flutter

Aim:

To develop a simple game application using Flutter.

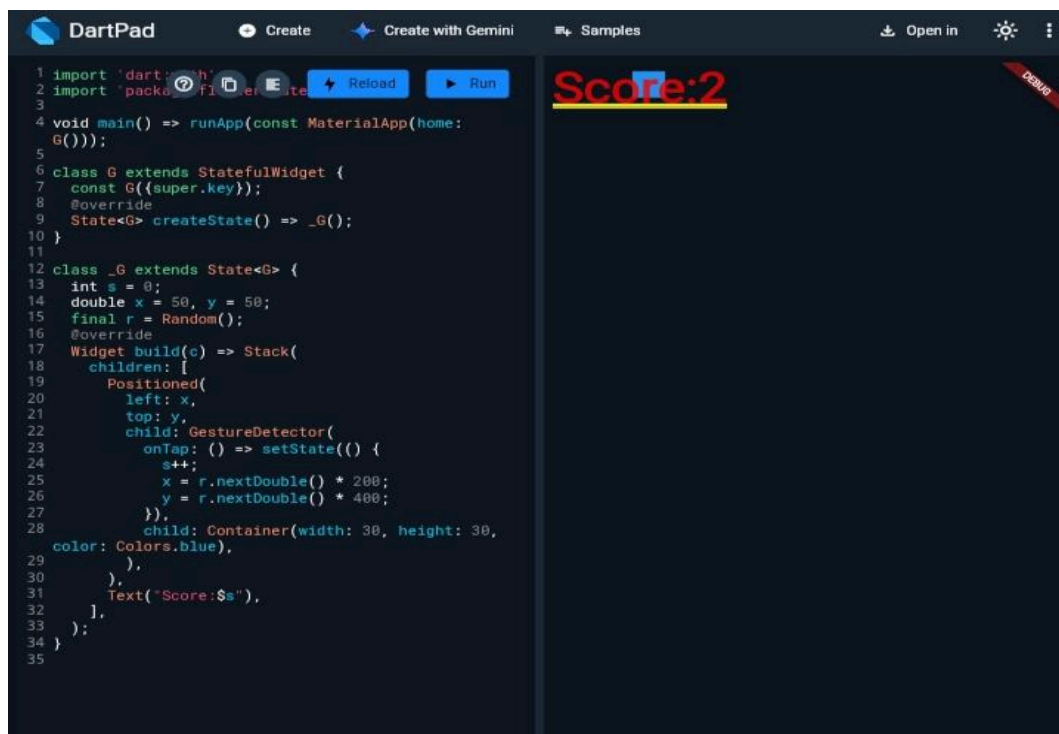
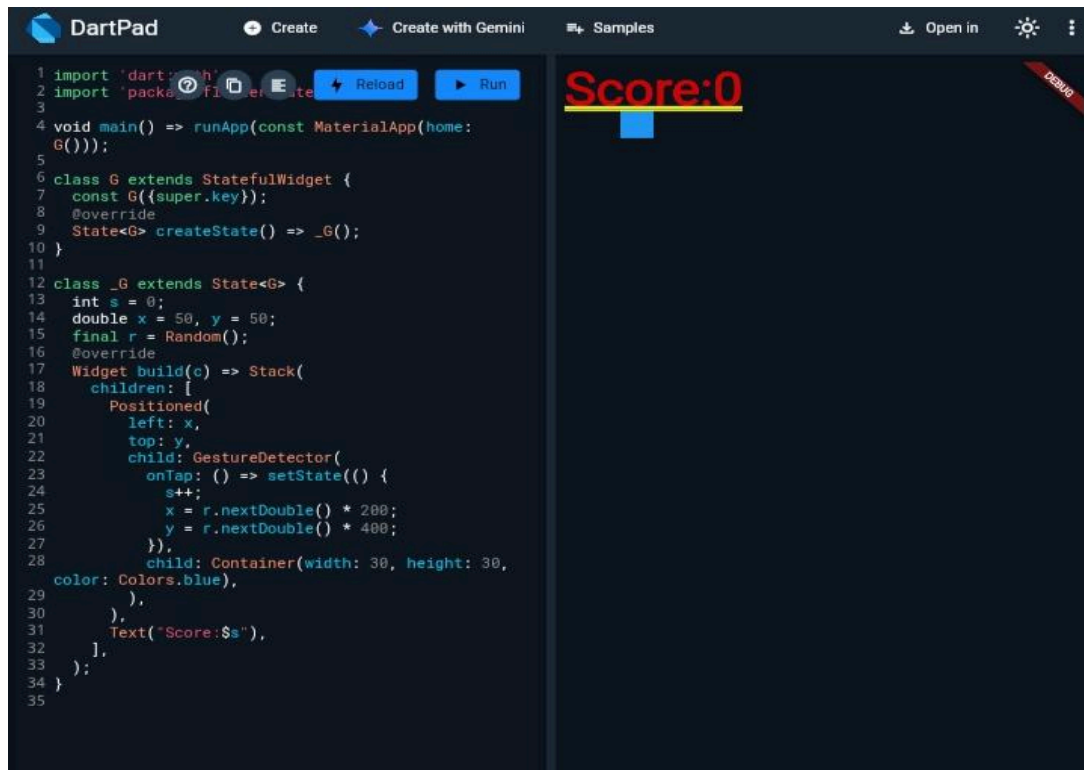
Program:

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

void main() => runApp(const MaterialApp(home: G()));

class G extends StatefulWidget {
  const G({super.key});
  @override
  State<G> createState() => _G();
}

class _G extends State<G> {
  int s = 0;
  double x = 50, y = 50;
  final r = Random();
  @override
  Widget build(c) => Stack(
    children: [
      Positioned(
        left: x,
        top: y,
        child: GestureDetector(
          onTap: () => setState(() {
            s++;
            x = r.nextDouble() * 200;
            y = r.nextDouble() * 400;
          }),
        child: Container(width: 30, height: 30, color: Colors.blue),
      ),
      Text("Score:$s"),
    ],
  );
}
```



10.Simple Calculator using Flutter

Aim:

To develop a simple calculator application using Flutter that performs basic arithmetic operations such as Addition, Subtraction, Multiplication, and Division using user input.

Program:

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

void main() {
  runApp(MaterialApp(debugShowCheckedModeBanner: false, home: Calc()));
}

class Calc extends StatefulWidget {
  @override
  _CalcState createState() => _CalcState();
}

class _CalcState extends State<Calc> {
  TextEditingController t1 = TextEditingController();
  TextEditingController t2 = TextEditingController();
  String result = "";

  void calculate(String op) {
    double a = double.parse(t1.text);
    double b = double.parse(t2.text);

    setState(() {
      if (op == "+") result = (a + b).toString();
      if (op == "-") result = (a - b).toString();
      if (op == "*") result = (a * b).toString();
      if (op == "/") result = (a / b).toString();
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Simple Calculator")),

```

```


body: Padding(
  padding: EdgeInsets.all(15),
  child: Column(
    children: [
      TextField(
        controller: t1,
        keyboardType: TextInputType.number,
        decoration: InputDecoration(labelText: "Enter first number"),
      ),
      TextField(
        controller: t2,
        keyboardType: TextInputType.number,
        decoration: InputDecoration(labelText: "Enter second number"),
      ),
      SizedBox(height: 20),

      Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          ElevatedButton(
            onPressed: () => calculate("+"),
            child: Text("+"),
          ),
          ElevatedButton(
            onPressed: () => calculate("-"),
            child: Text("-"),
          ),
          ElevatedButton(
            onPressed: () => calculate("*"),
            child: Text("*"),
          ),
          ElevatedButton(
            onPressed: () => calculate("/"),
            child: Text("/"),
          ),
        ],
      ),

      SizedBox(height: 20),
      Text("Result: $result", style: TextStyle(fontSize: 22)),
    ],
  ),

```

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


 DartPad


Create

Create with Gemini

Samples

Open in





```
1 import 'package:flutter/material.dart';  
2  
3 void main() {  
4   runApp(MaterialApp(debugShowCheckedModeBanner:  
5     false, home: Calc()));  
6 }  
7  
8 class Calc extends StatefulWidget {  
9   @override  
10  _CalcState createState() => _CalcState();  
11 }  
12  
13 class _CalcState extends State<Calc> {  
14   TextEditingController t1 =  
15     TextEditingController();  
16   TextEditingController t2 =  
17     TextEditingController();  
18   String result = "";  
19  
20   void calculate(String op) {  
21     double a = double.parse(t1.text);  
22     double b = double.parse(t2.text);  
23  
24     setState(() {  
25       if (op == "+") result = (a + b).toString();  
26       if (op == "-") result = (a - b).toString();  
27       if (op == "*") result = (a * b).toString();  
28       if (op == "/") result = (a / b).toString();  
29     });  
30   }  
31  
32   @override  
33   Widget build(BuildContext context) {  
34     return Scaffold(  
35       appBar: AppBar(title: Text("Simple  
36         Calculator")),  
37       body: Padding(  
38         padding: EdgeInsets.all(15),  
39         child: Column(  
40           children: [  
41             TextField(  
42               controller: t1,  
43               keyboardType: TextInputType.number,  
44               decoration:  
45                 InputDecoration(labelText: "Enter first number"),  
46             ),  
47             TextField(  
48               controller: t2,  
49               keyboardType: TextInputType.number,  
50               decoration:  
51                 InputDecoration(labelText: "Enter second number"),  
52             ),  
53             Row(  
54               mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
55               children: [  
56                 IconButton(icon: Icon(Icons.add), onPressed: () => calculate("+")),  
57                 IconButton(icon: Icon(Icons.subtract), onPressed: () => calculate("-")),  
58                 IconButton(icon: Icon(Icons.multiply), onPressed: () => calculate("*")),  
59                 IconButton(icon: Icon(Icons.divide), onPressed: () => calculate("/")),  
60               ],  
61             ),  
62             Text(result),  
63           ],  
64         ),  
65     );  
66   }  
67 }
```

Simple Calculator

Enter first number
5

Enter second number
2

+

-

*

/

Result: 7