

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

The screenshot shows the DartPad interface. On the left, the code editor displays the following Dart code:

```
1 import 'package:flutter/material';
2
3 void main() => runApp(
4   MaterialApp(
5     home: Scaffold(
6       body: Center(
7         child: Text(
8           "Hello Flutter!",
9           style: TextStyle(
10             fontSize: 30,
11             color: Colors.blue, // ⚡ Text color
12           ),
13         ),
14       ),
15     ),
16   ),
17 );
18
```

The right side of the interface shows the resulting Flutter application running in a simulator. The screen displays the text "Hello Flutter!" in a large, blue, sans-serif font, centered on the screen. A red banner at the top right of the simulator window says "DEBUG".

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

DartPad

Create Create with Gemini Samples Open in ☰ : DEBUG

```
1 import 'package:flutter/material.dart';
2 void main() => runApp(
3     MaterialApp(
4         home: Scaffold(
5             body: Column(
6                 mainAxisAlignment: MainAxisAlignment.center,
7                 children: [
8                     Row(
9                         mainAxisAlignment: MainAxisAlignment.center,
10                        children: [
11                            Text("A",
12                                style: TextStyle(color: Colors.red,
13                                    fontSize: 24)),
14                            Text("B",
15                                style: TextStyle(color:
16                                    Colors.blue, fontSize: 24)),
17                        ],
18                    ),
19                ],
20            ),
21        ),
22    ),
23    ),
24    ),
25    ),
26    ),
27    ),
28    ),
29    ),
30    ),
31 );
```

A B

Below Row

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(
                        mainAxisSize: MainAxisSize.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
```



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;
}
```

```
↳ Opt
```

The screenshot shows a Flutter development interface. On the left, there is a code editor with the following Dart code:

```
1 import 'package:flutter/material';
2 void main() => runApp(
3   MaterialApp(
4     home: Scaffold(
5       body: CustomPaint(painter: MyDraw(), child:
6           Container()),
7     ),
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 }
```

At the top of the code editor, there are several icons: a question mark, a circular icon with a dot, a circular icon with a minus sign, a circular icon with a plus sign, a circular icon with a checkmark, a circular icon with an exclamation mark, a reload button, and a run button.

To the right of the code editor is a preview window. The preview shows a white screen with a blue horizontal bar at the top. In the center of the screen is a red circle with a blue outline. The word "DEBUG" is visible in the top right corner of the preview area.

5.Notification Manager

Aim:

To develop an application using notification Manager.

Program:

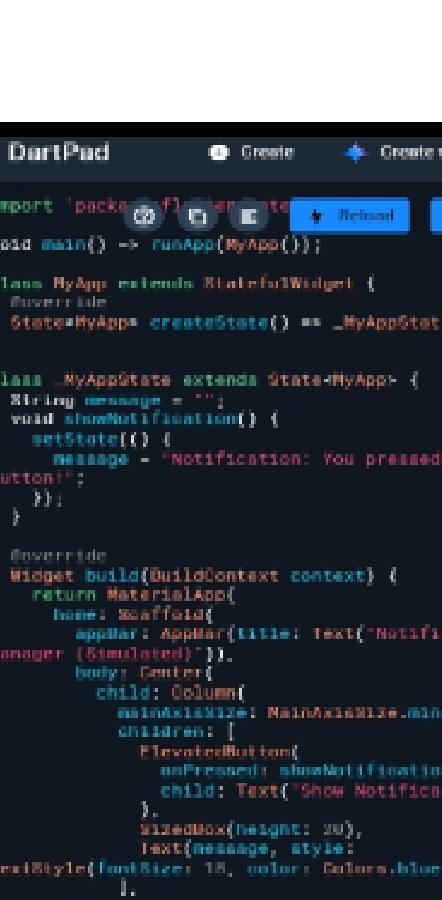
```
import 'package:flutter/material.dart';

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!";
    });
  }

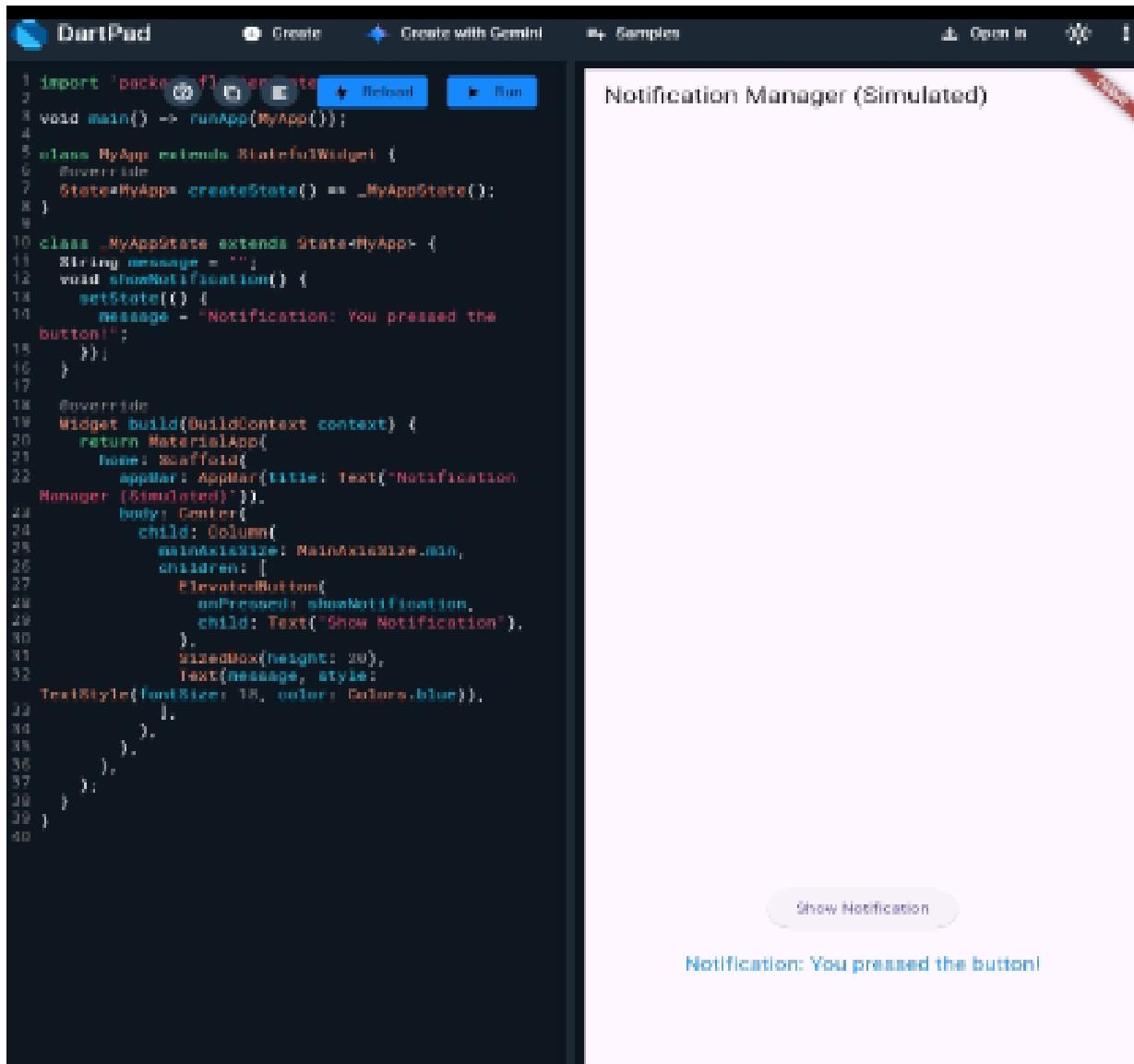
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text("Notification Manager (Simulated)")),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
       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



DartPad

Create Create with Code

```
1 import 'package:flutter/material.dart';
2 void main() => runApp(MyApp());
3
4 class MyApp extends StatelessWidget {
5   @override
6   State<MyApp> createState() => _MyAppState();
7 }
8
9 class _MyAppState extends State<MyApp> {
10   String message = "";
11   void showNotification() {
12     setState(() {
13       message = "Notification: You pressed the button!";
14     });
15   }
16
17   @override
18   Widget build(BuildContext context) {
19     return MaterialApp(
20       title: "Notification Manager (Simulated)",
21       home: Scaffold(
22         appBar: AppBar(title: Text("Notification Manager (Simulated)")),
23         body: Center(
24           child: Column(
25             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
26             children: [
27               ElevatedButton(
28                 onPressed: showNotification,
29                 child: Text("Show Notification"),
30               ),
31               SizedBox(height: 20),
32               Text(message, style: TextStyle(fontSize: 18, color: Colors.blue)),
33             ],
34           ),
35         ),
36       ),
37     );
38   }
39 }
```



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

```

The screenshot shows the DartPad interface with the code for the Animation Demo. The code defines a main application class and an animation demo stateful widget. The animation demo uses a GestureDetector to toggle a boolean value 'big' and an AnimatedContainer to change the width and height of a square based on this value. The preview window on the right shows a solid blue square.

```

1 import 'package:flutter/material.dart';
2 void main() {
3   runApp(const MyApp());
4 }
5
6 class MyApp extends StatelessWidget {
7   const MyApp({super.key});
8
9   @override
10  Widget build(BuildContext context) {
11    return const MaterialApp(
12      home: AnimationDemo(),
13    );
14  }
15 }
16
17 class AnimationDemo extends StatefulWidget {
18   const AnimationDemo({super.key});
19
20   @override
21   State<AnimationDemo> createState() =>
22     _AnimationDemoState();
23 }
24
25 class _AnimationDemoState extends
26 State<AnimationDemo> {
27   bool big = false;
28
29   @override
30   Widget build(BuildContext context) {
31     return Scaffold(
32       appBar: AppBar(title: const Text("Animation
33       App")),
34       body: Center(
35         child: GestureDetector(
36           onTap: () {
37             setState(() {
38               big = !big;
39             });
40             child: AnimatedContainer(
41               duration: const Duration(seconds: 1),
42               width: big ? 200 : 100,
43               height: big ? 200 : 100,
44               color: big ? Colors.blue : Colors.red,
45             ),
46           ),
47         );
48       );
49     }
50   }

```

DartPad

Create Create with Gemini Samples Open in ☀️ ⚙️

```
1 import 'package:flutter/material.dart';
2 void main() {
3   runApp(const MyApp());
4 }
5
6 class MyApp extends StatelessWidget {
7   const MyApp({super.key});
8
9   @override
10  Widget build(BuildContext context) {
11    return const MaterialApp(
12      home: AnimationDemo(),
13    );
14  }
15}
16
17 class AnimationDemo extends StatefulWidget {
18   const AnimationDemo({super.key});
19
20   @override
21   State<AnimationDemo> createState() =>
22     _AnimationDemoState();
23 }
24
25 class _AnimationDemoState extends
26   State<AnimationDemo> {
27   bool big = false;
28
29   @override
30   Widget build(BuildContext context) {
31     return Scaffold(
32       appBar: AppBar(title: const Text("Animation
33       App")),
34       body: Center(
35         child: GestureDetector(
36           onTap: () {
37             setState(() {
38               big = !big;
39             });
40             child: AnimatedContainer(
41               duration: const Duration(seconds: 1),
42               width: big ? 200 : 100,
43               height: big ? 200 : 100,
44               color: big ? Colors.blue : Colors.red,
45             ),
46           ),
47         );
48       );
49     }
50 }
```

Animation App

DEBUG

7. Application using multi-threading

Aim:

To develop an application using multi-threading

Program:

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

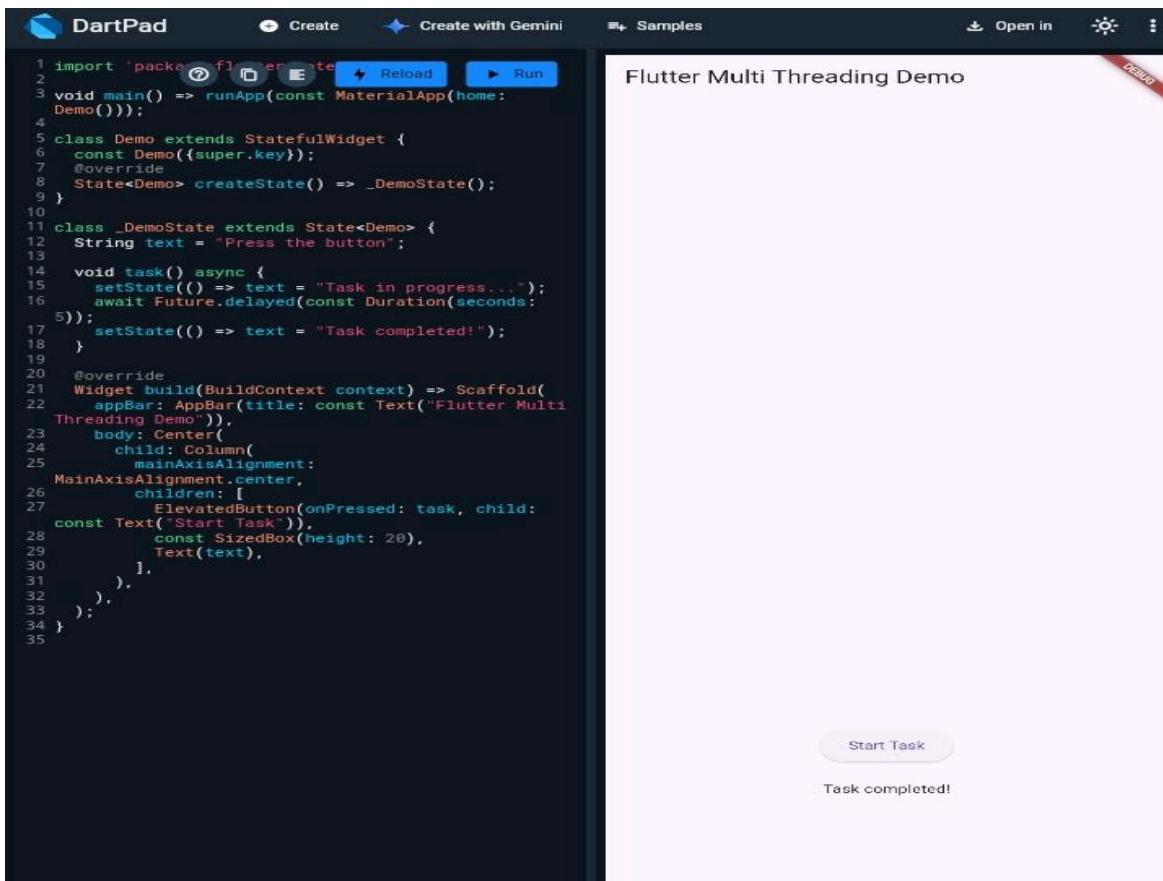
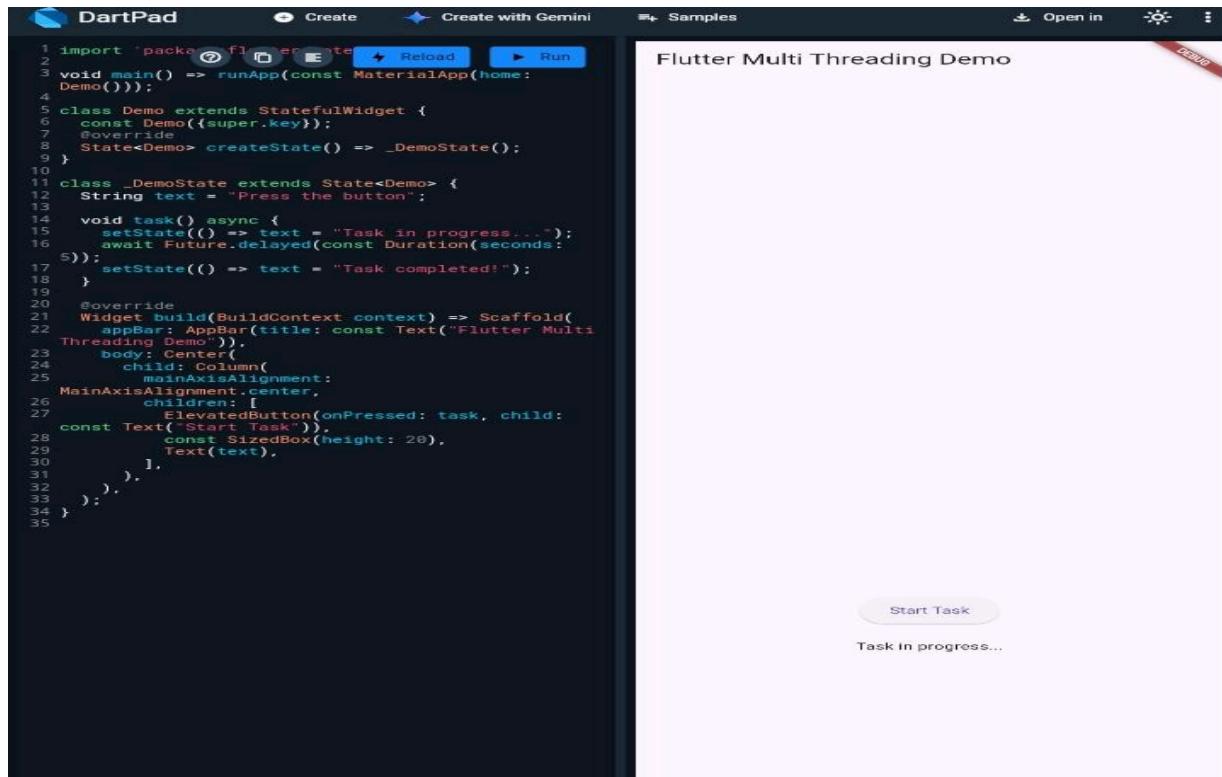
void main() => runApp(const MaterialApp(home: Demo()));

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

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

The screenshot shows the DartPad interface with the code for an alert dialog demo. The code defines an `App` widget that runs a `Demo` screen. The `Demo` screen contains a button labeled "Show Alert" that triggers a `showDialog` call. This dialog has a title "Message", content "New message received", and an "OK" action. The DartPad interface includes tabs for Create, Create with Gemini, Samples, and Open in, along with a Run button and a status bar indicating "DEBUT".

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

DartPad

Create Create with Gemini Samples Open in : Dart Demo

```
1 import 'package:flutter/material.dart';
2 void main() => runApp(const App());
3
4 class App extends StatelessWidget {
5   const App({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     return const MaterialApp(home: Demo());
10   }
11 }
12
13 class Demo extends StatelessWidget {
14   const Demo({super.key});
15
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       appBar: AppBar(title: const Text("Alert Demo")),
20       body: Center(
21         child: ElevatedButton(
22           child: const Text("Show Alert"),
23           onPressed: () {
24             showDialog(
25               context: context,
26               builder: (_) => AlertDialog(
27                 title: const Text("Message"),
28                 content: const Text("New message
29 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 }
```

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"),
      ],
    );
}
```

DartPad

Create Create with Gemini Samples Open in DEBUG

```
1 import 'dart:ui';
2 import 'package:flutter/material';
3
4 void main() => runApp(const MaterialApp(home: G()));
5
6 class G extends StatefulWidget {
7   const G({super.key});
8   @override
9   State<G> createState() => _G();
10 }
11
12 class _G extends State<G> {
13   int s = 0;
14   double x = 50, y = 50;
15   final r = Random();
16   @override
17   Widget build(c) => Stack(
18     children: [
19       Positioned(
20         left: x,
21         top: y,
22         child: GestureDetector(
23           onTap: () => setState(() {
24             s++;
25             x = r.nextDouble() * 200;
26             y = r.nextDouble() * 400;
27           }),
28           child: Container(width: 30, height: 30,
29                             color: Colors.blue),
30         ),
31         Text("Score:$s"),
32       ],
33     );
34   );
35 }
```

Score:0

DartPad

Create Create with Gemini Samples Open in DEBUG

```
1 import 'dart:ui';
2 import 'package:flutter/material';
3
4 void main() => runApp(const MaterialApp(home: G()));
5
6 class G extends StatefulWidget {
7   const G({super.key});
8   @override
9   State<G> createState() => _G();
10 }
11
12 class _G extends State<G> {
13   int s = 0;
14   double x = 50, y = 50;
15   final r = Random();
16   @override
17   Widget build(c) => Stack(
18     children: [
19       Positioned(
20         left: x,
21         top: y,
22         child: GestureDetector(
23           onTap: () => setState(() {
24             s++;
25             x = r.nextDouble() * 200;
26             y = r.nextDouble() * 400;
27           }),
28           child: Container(width: 30, height: 30,
29                             color: Colors.blue),
30         ),
31         Text("Score:$s"),
32       ],
33     );
34   );
35 }
```

Score:2

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

Simple Calculator

Enter first number
5

Enter second number
2

+ - × ÷

Result: 7