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

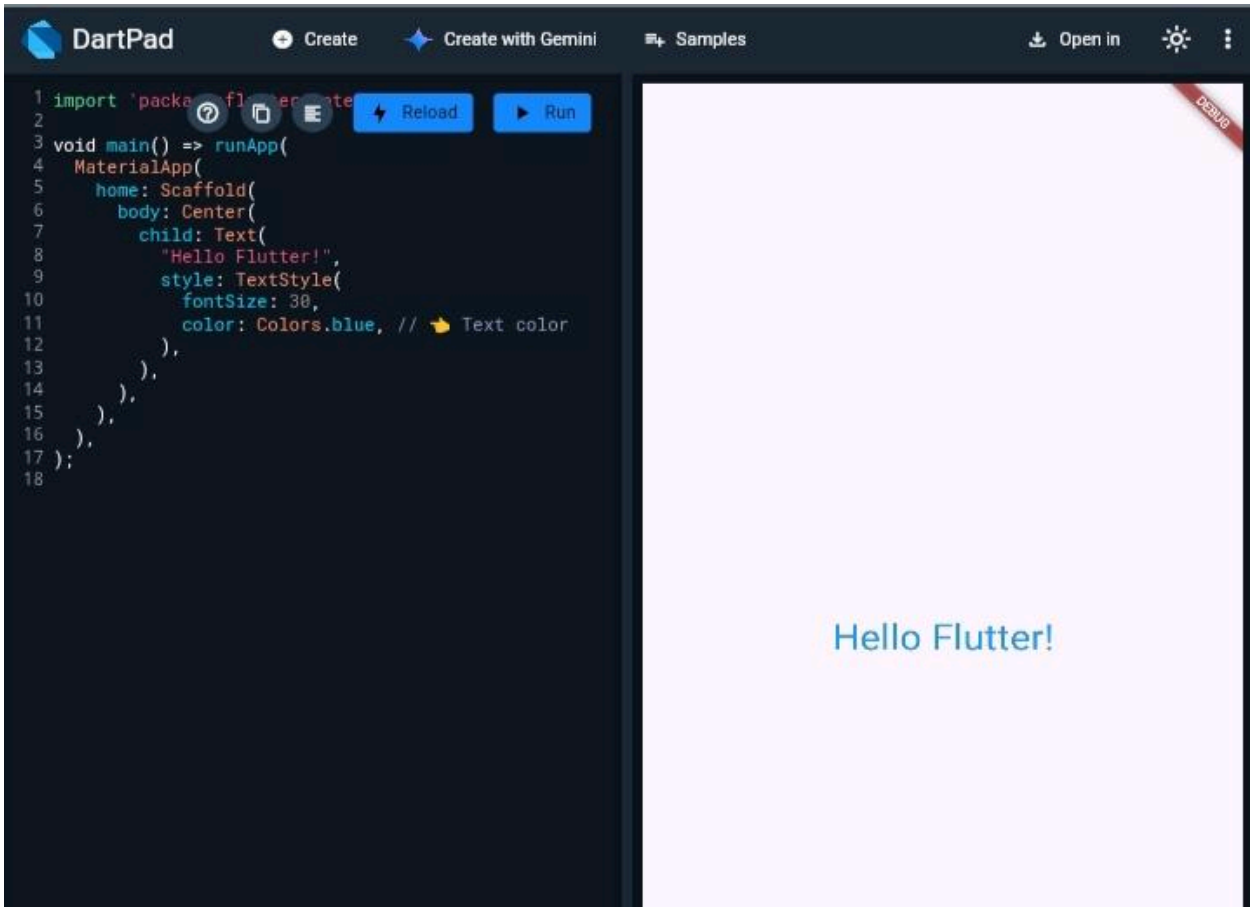
1. Flutter GUI Components

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


DartPad
Create
Create with Gemini
Samples
Open in
Reload
Run
1 import 'packa
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
DEBUG
Hello Flutter!

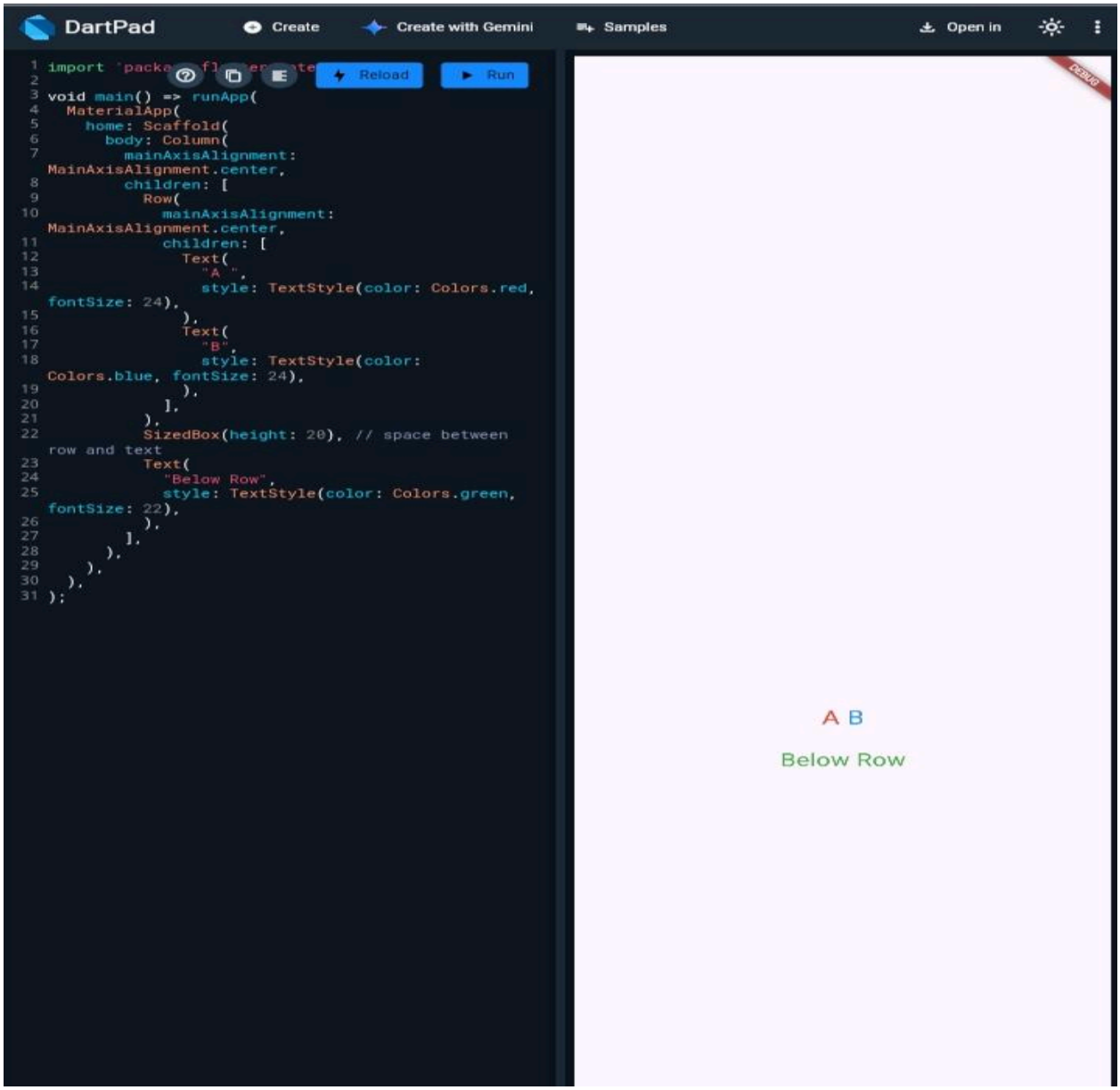
2.Flutter Layout Design

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
Reload
Run
DEBUG
A B
Below Row

3.Flutter Event Handling

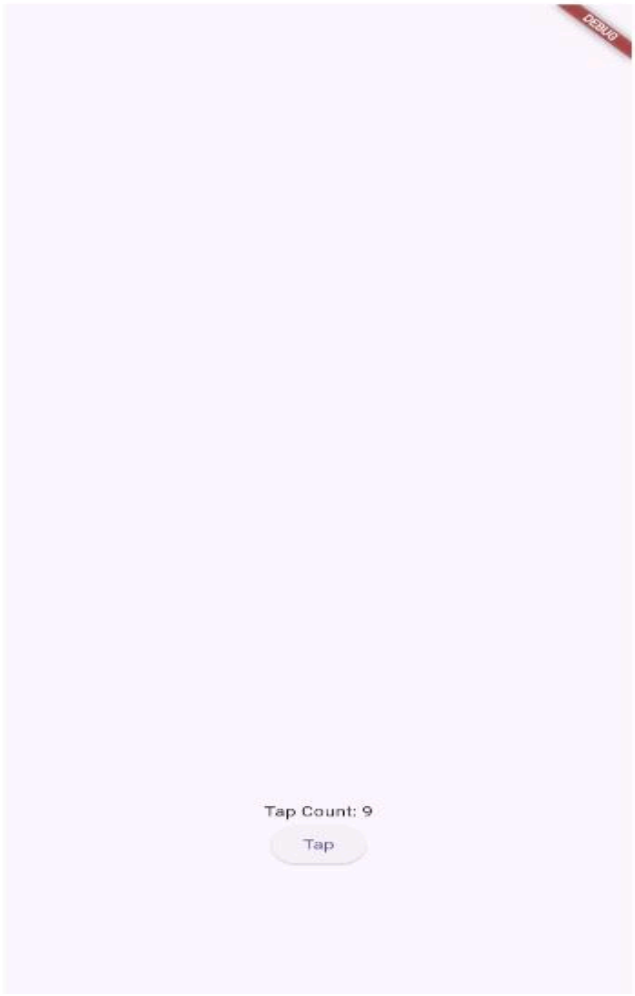
Aim:

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

```dart
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"),
            )
          ],
        ),
      ),
    ),
  ),
));
```



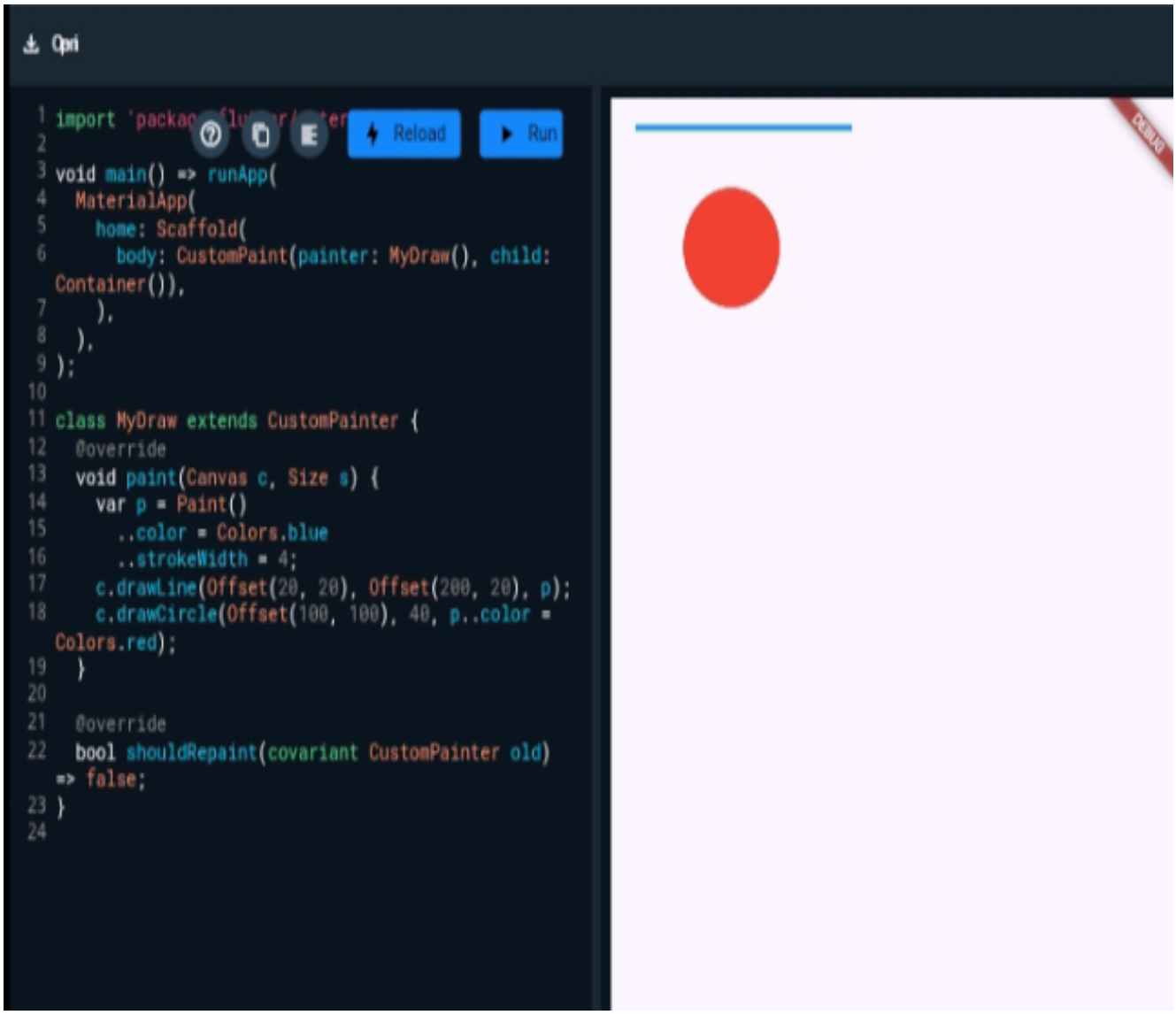
4.Basic graphical primitives

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;
}
```


Dart
Reload
Run
1 import 'packa
2
3 void main() => runApp(
4 MaterialApp(
5 home: Scaffold(
6 body: CustomPaint(painter: MyDraw(), child:
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 =
Colors.red);
19 }
20
21 @override
22 bool shouldRepaint(covariant CustomPainter old)
=> false;
23 }
24
DEBUG

5.Notification Manager

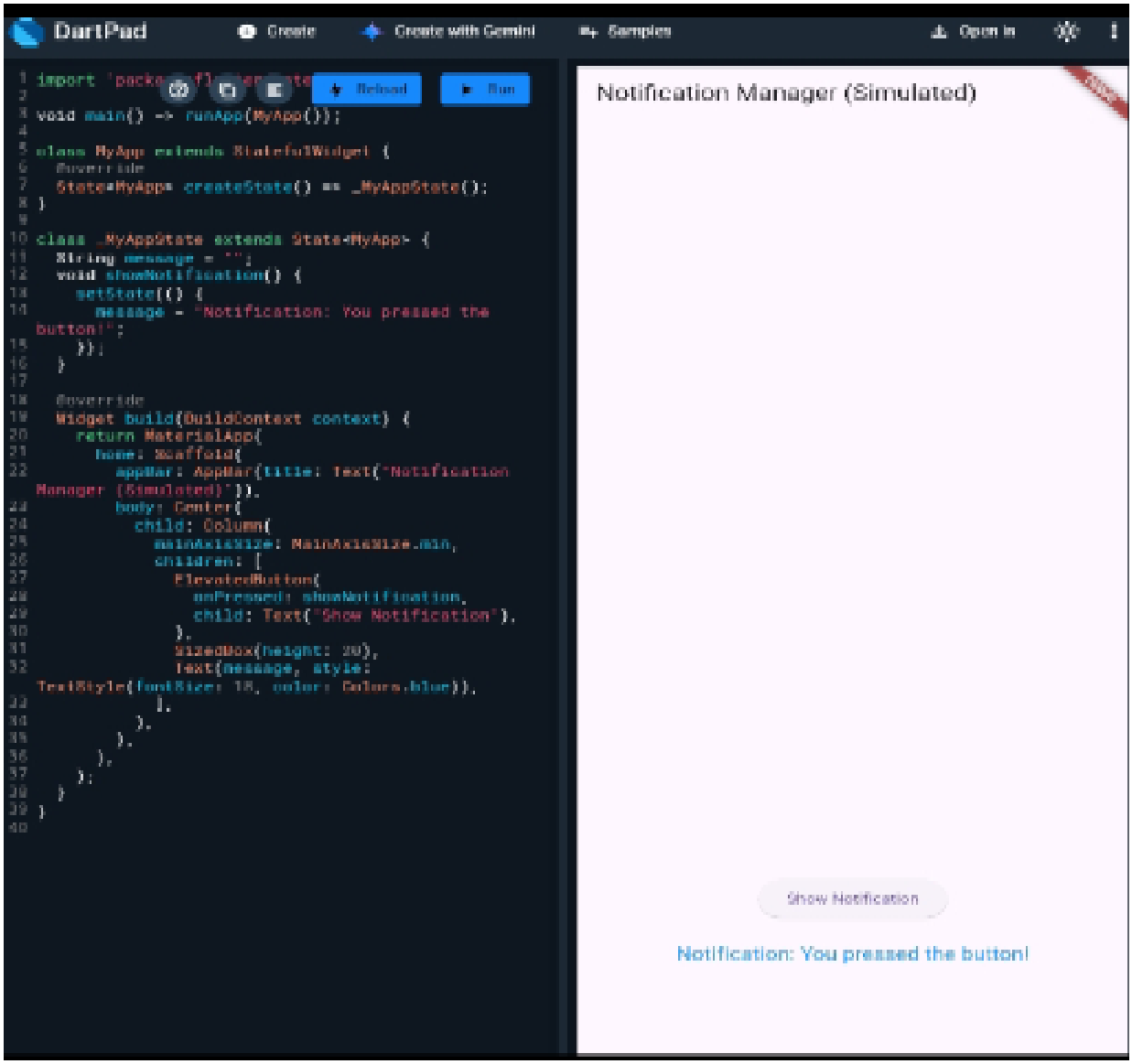
Aim:

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(
            mainAxisSize: MainAxisSize.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
Reload
Run
void main() => runApp(MyApp());
class MyApp extends StatefulWidget {
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(
mainAxisSize: MainAxisSize.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

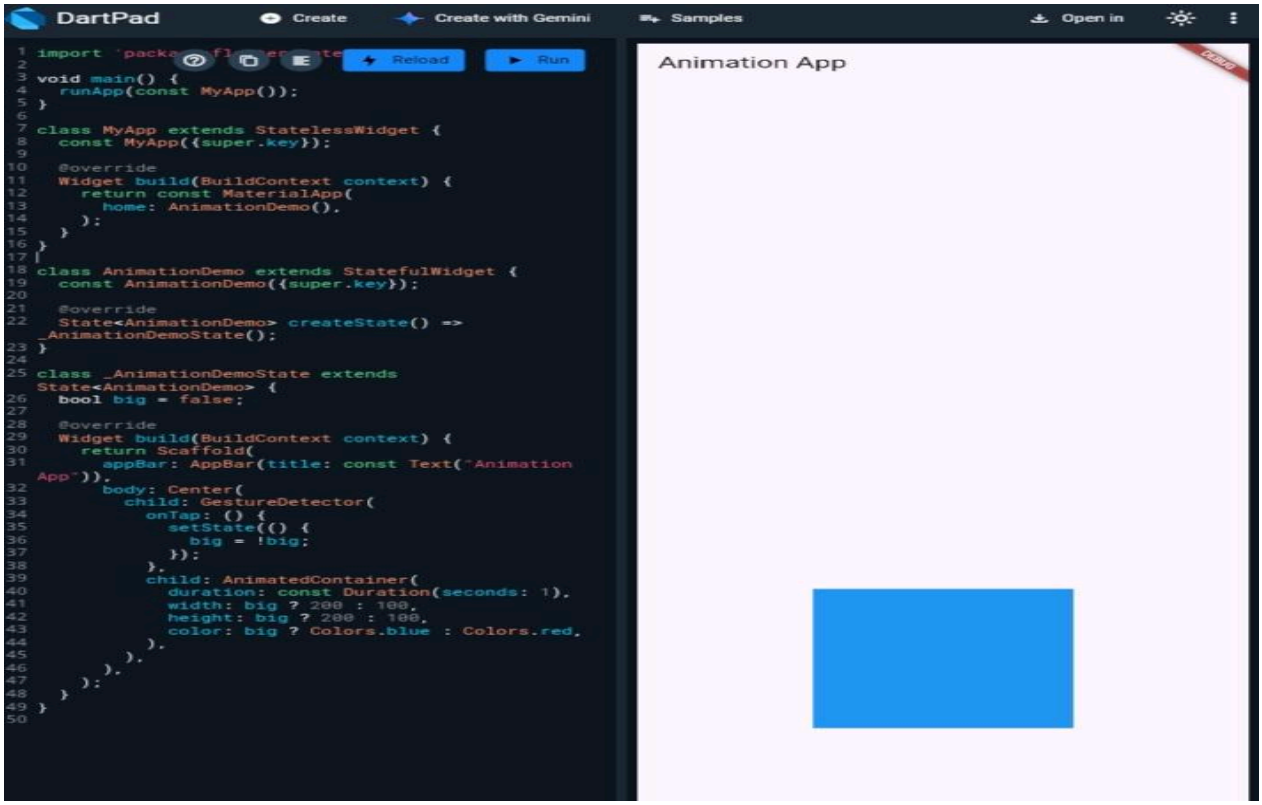
Aim:

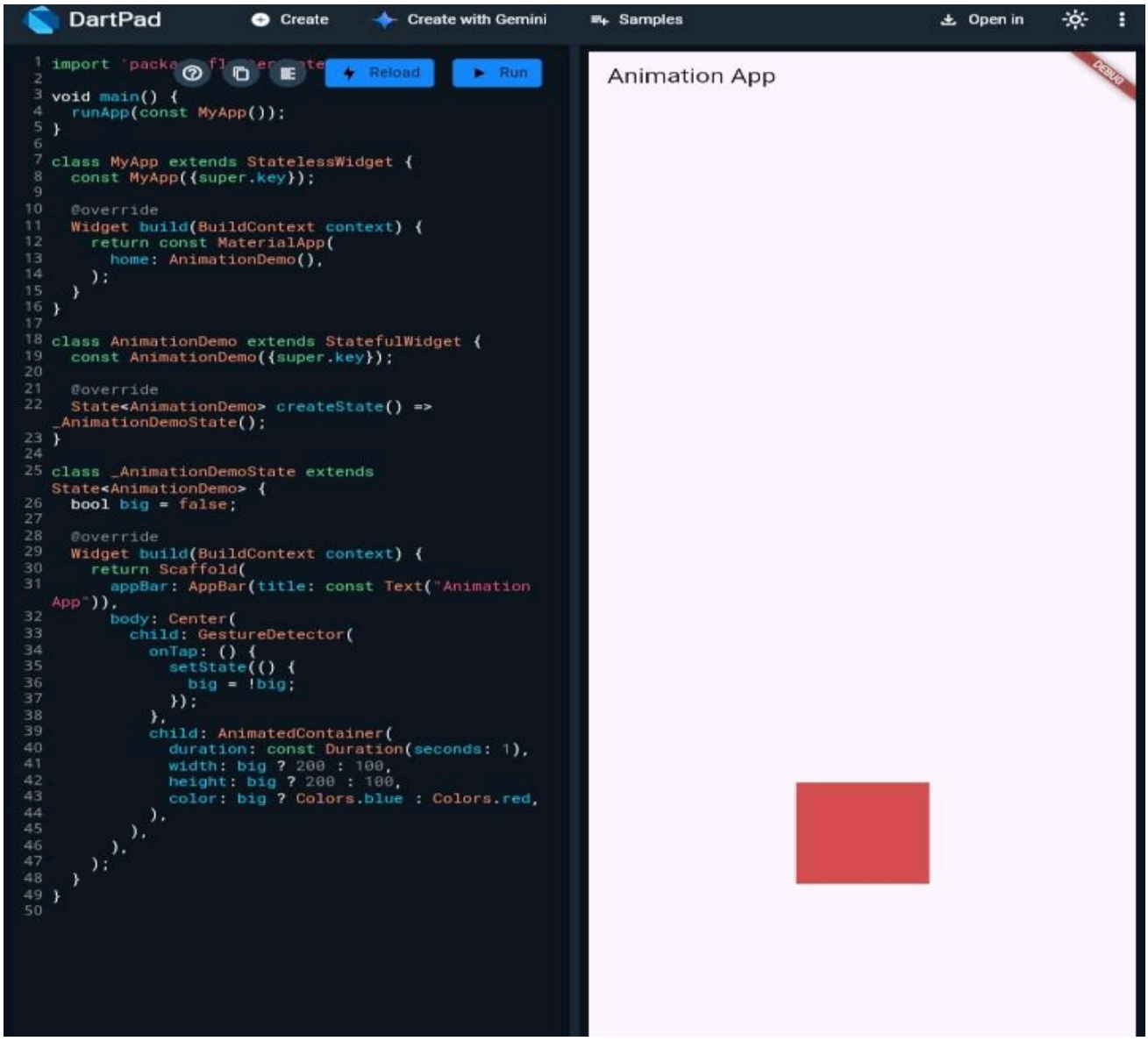
To develop an Application using Animation

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




DartPad
Create
Create with Gemini
Samples
Open in
Reload
Run
Animation App
DEBUG
import 'packa fl er ate
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,
),
),
),
);
}
}

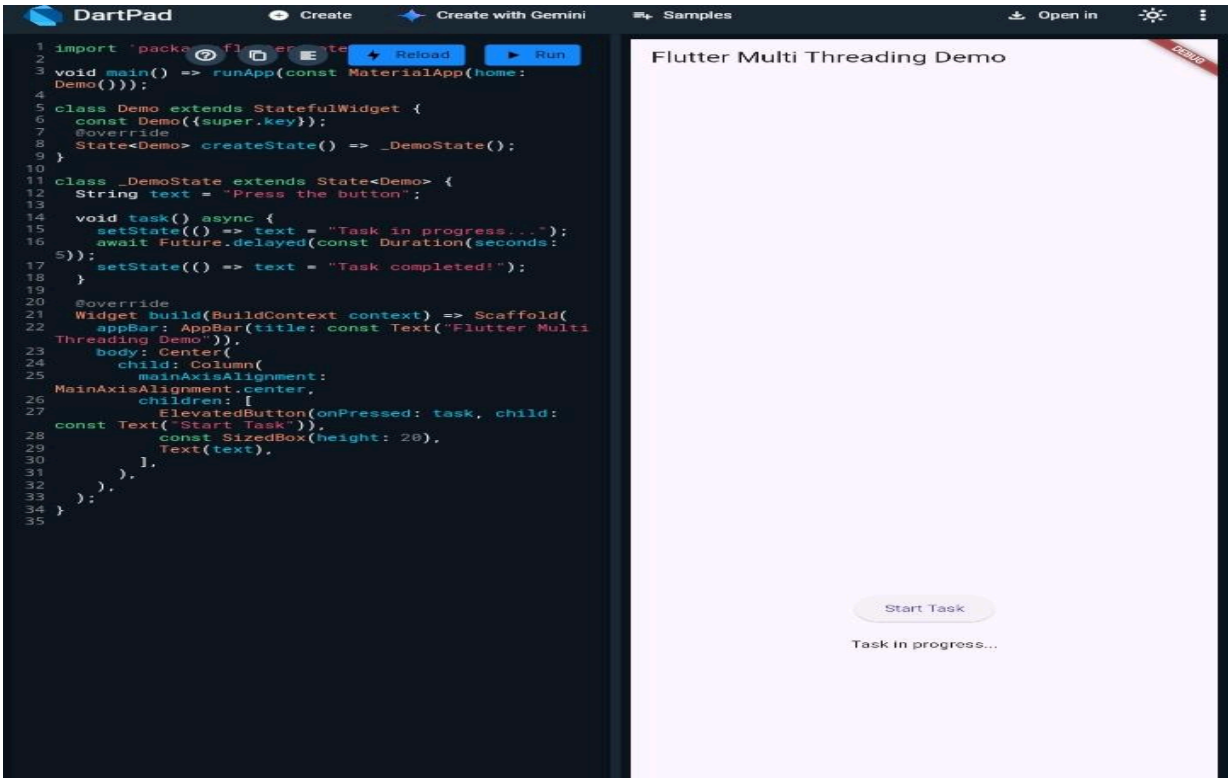
7. Application using multi-threading

Aim:

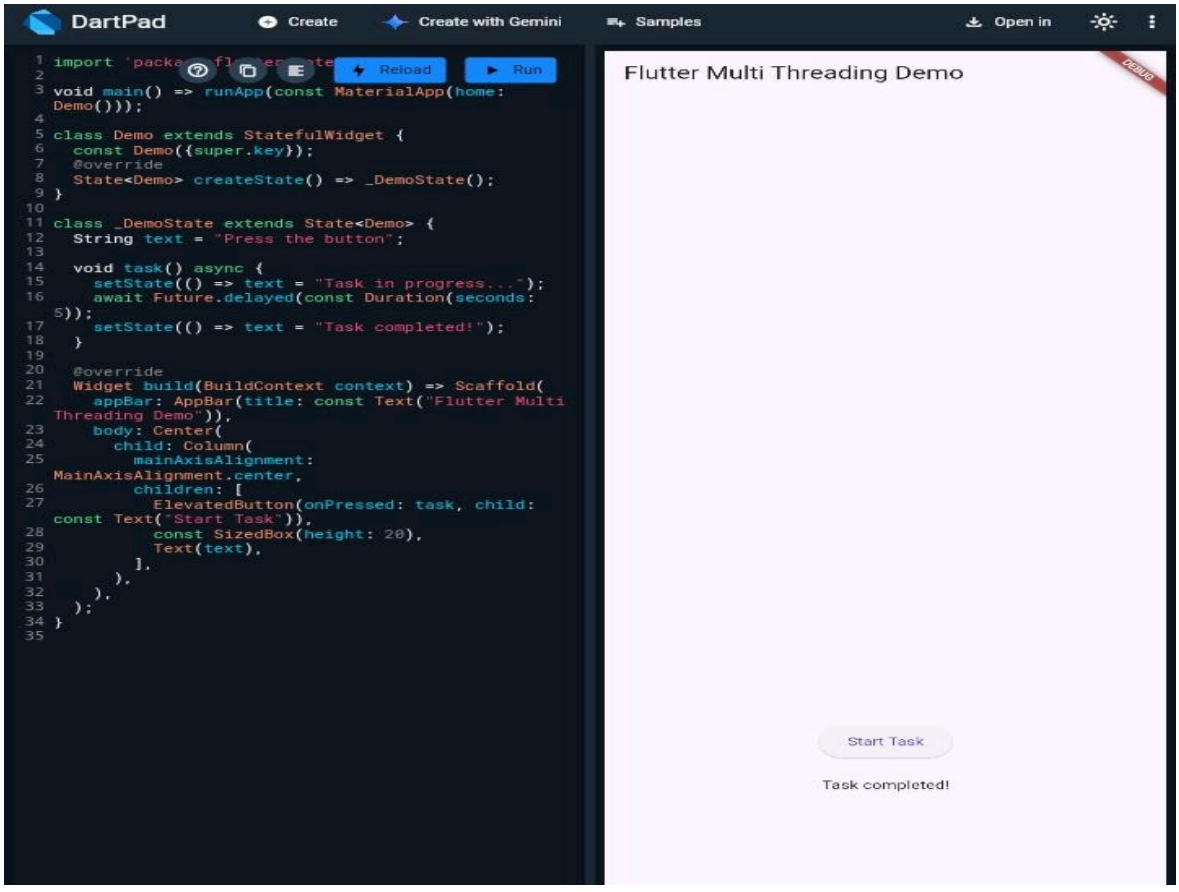
To develop an application using multi-threading

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




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




```dart
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

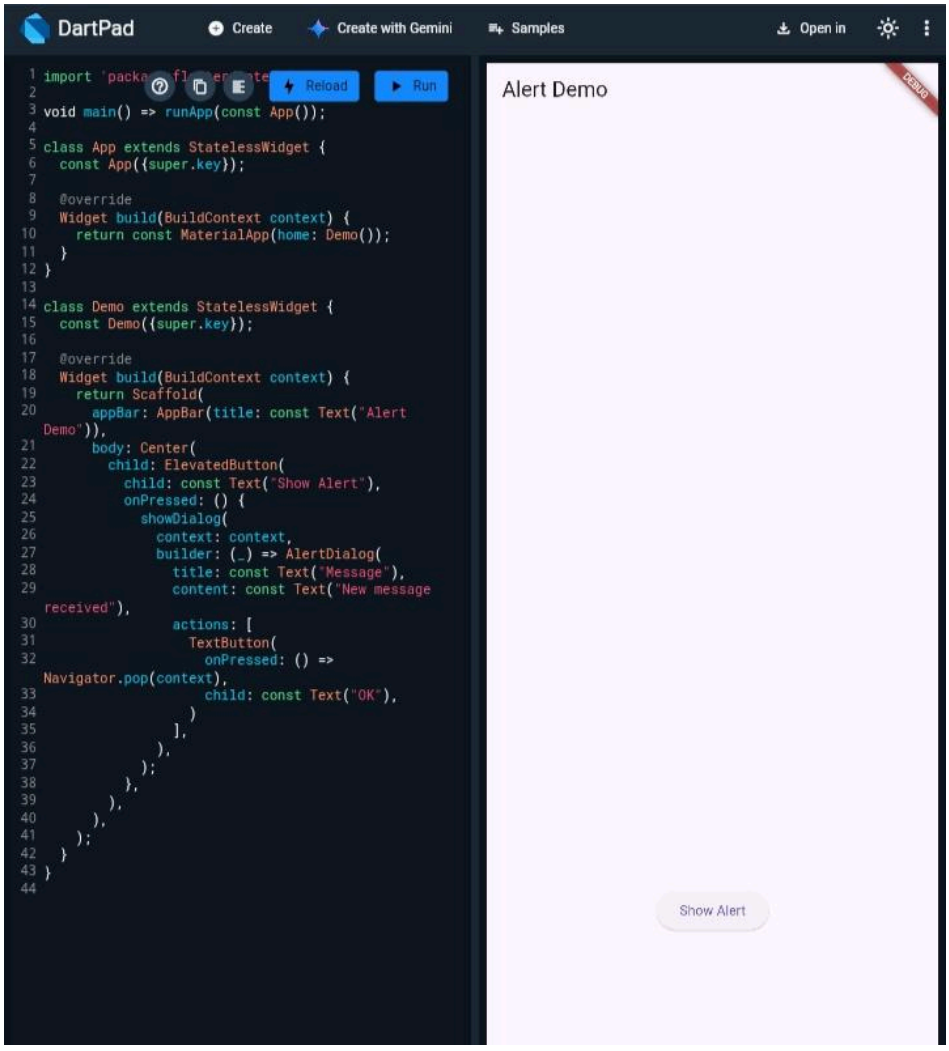
Aim:

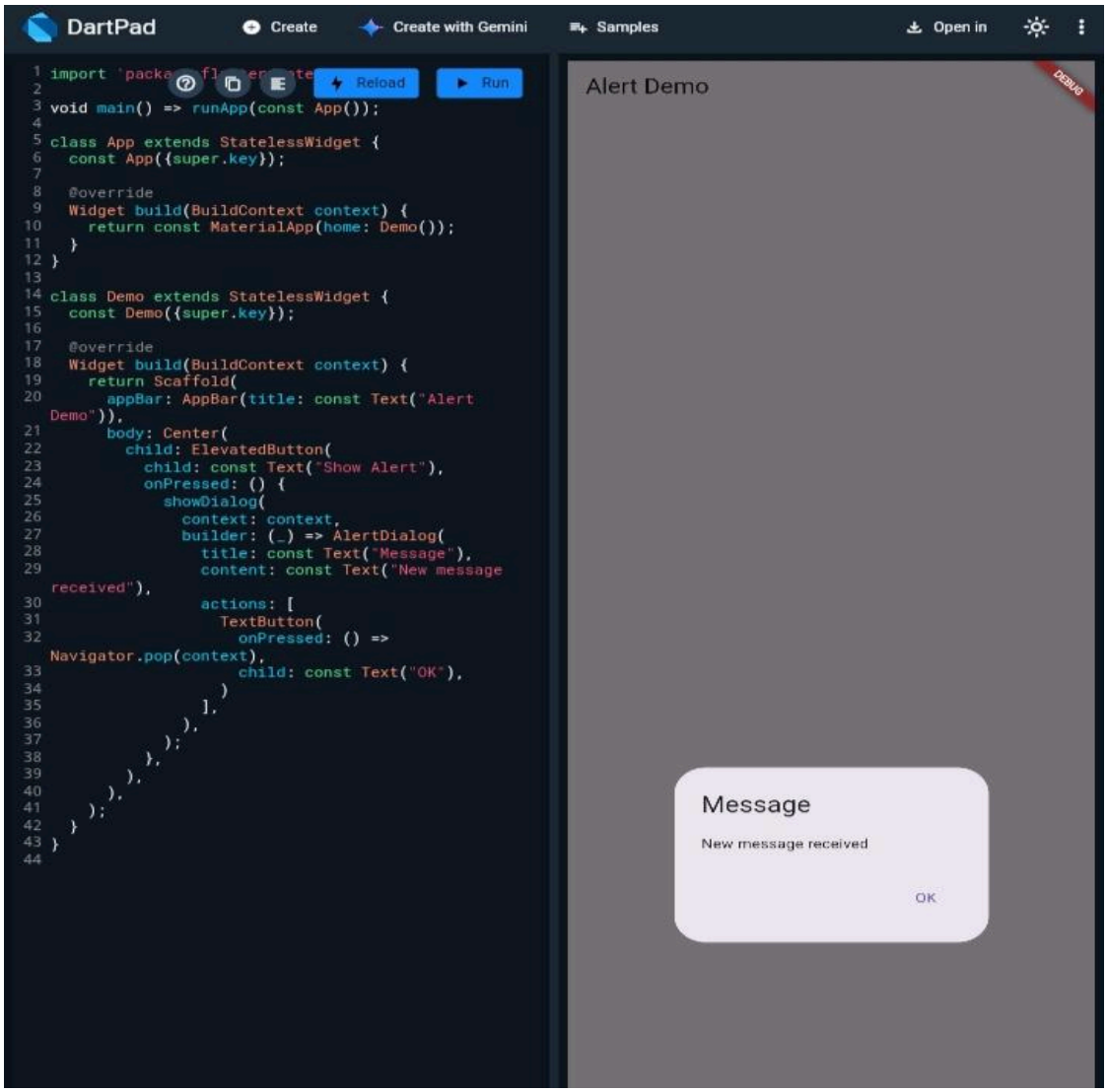
To develop an application for creating an alert upon receiving a message.

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"),
```

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


DartPad
Create
Create with Gemini
Samples
Open in
Reload
Run
Alert Demo
DEBUG
Show Alert


DartPad
Create
Create with Gemini
Samples
Open in
Reload
Run
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"),
)
],
),
);
},
),
),
);
}
}
Alert Demo
DEBUG
Message
New message received
OK

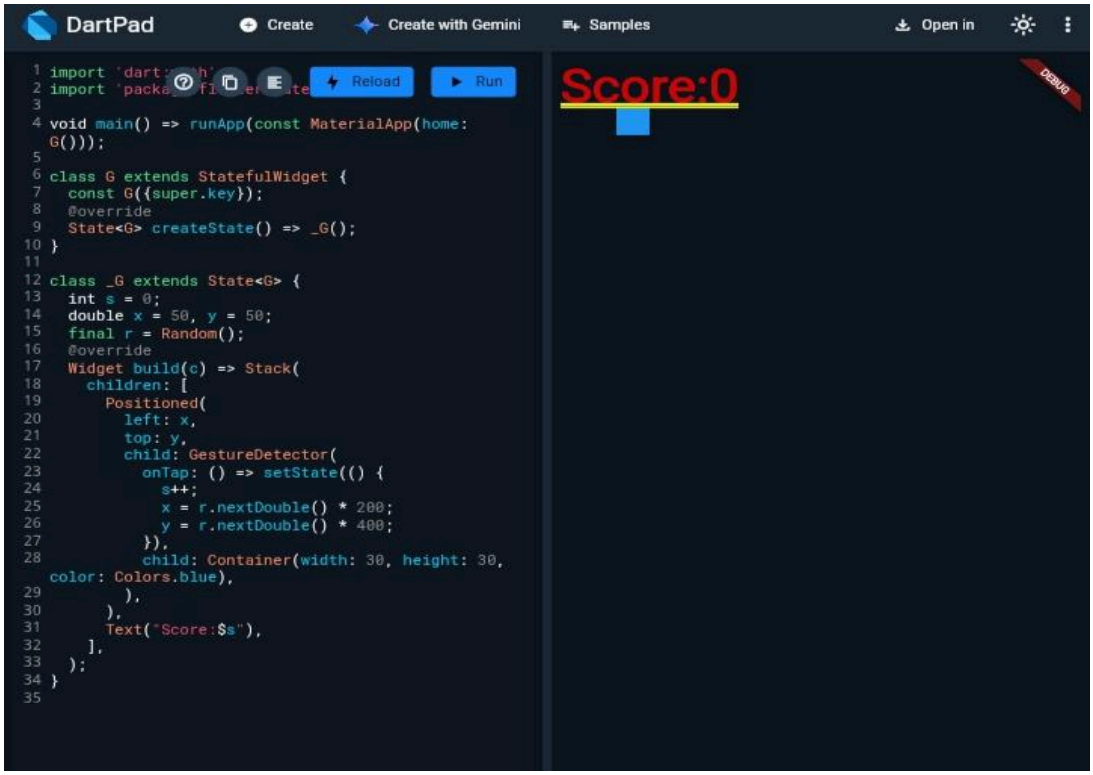
9.Simple Game Application using Flutter

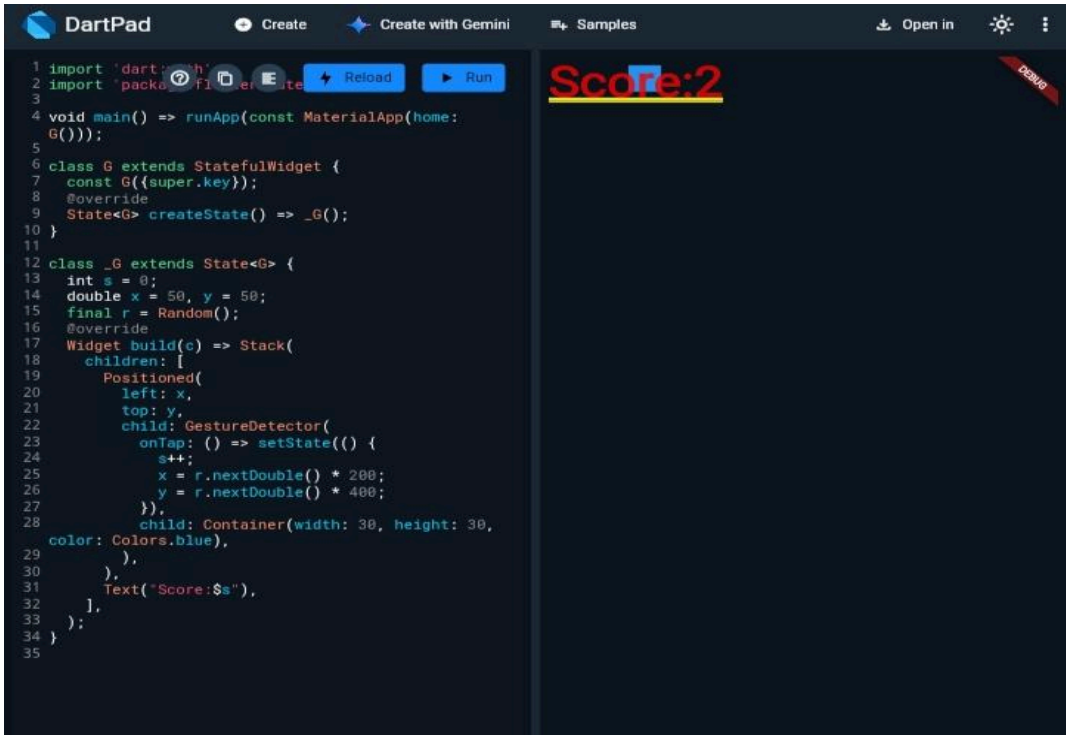
Aim:

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
Reload
Run
Score:0
DEBUG
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
Reload
Run
Score:2
DEBUG
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:

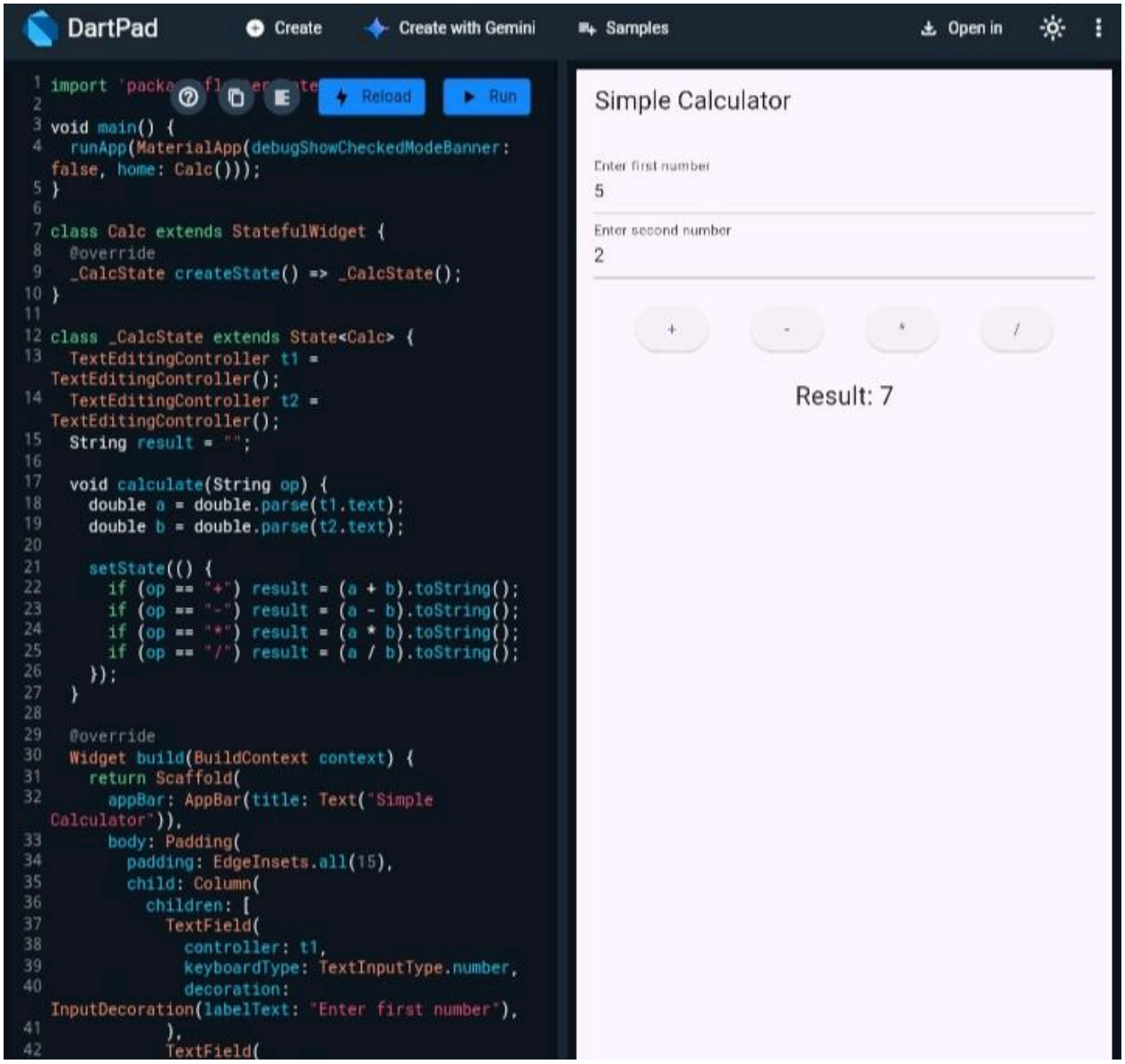
To develop a simple calculator application using Flutter that performs basic arithmetic operations such as Addition, Subtraction, Multiplication, and Division using user input.

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
Reload
Run
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(
Simple Calculator
Enter first number
5
Enter second number
2
+
-
*
/
Result: 7