

WEEK-5

AIM:A) Learn about stateful and stateless widgets.

DESCRIPTION: In Flutter, a Stateless Widget is immutable and displays fixed content that doesn't change during its lifetime, making it ideal for static UI. A Stateful Widget can change dynamically based on user interaction or data updates, using a State class and setState() to rebuild the UI when needed.

PROGRAM:

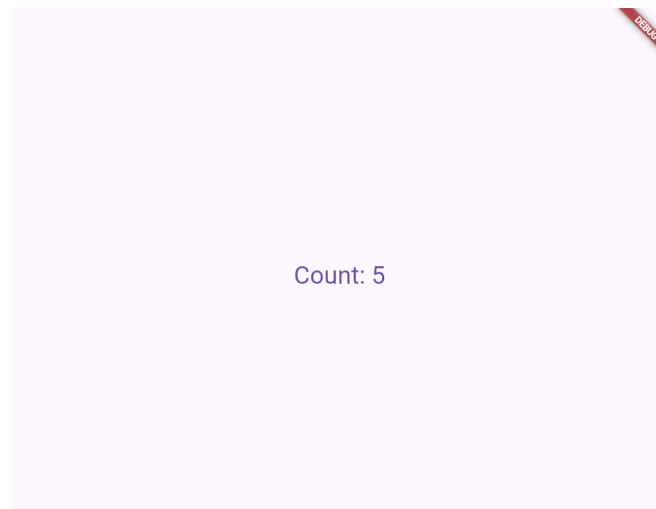
```
import 'package:flutter/material.dart';

void main() => runApp(MaterialApp(home: Counter()));

class Counter extends StatefulWidget {
  @override
  _CounterState createState() => _CounterState();
}

class _CounterState extends State<Counter> {
  int count = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: TextButton(
          onPressed: () => setState(() => count++),
          child: Text('Count: $count', style: TextStyle(fontSize: 28)),
        ),
      ),
    );
  }
}
```

OUTPUT:

AIM:B) Implement state management using set State and Provider.

DESCRIPTION: In Flutter, state management controls how data changes and updates the UI. Using `setState()`, state changes are handled within a single widget by calling `setState()` to rebuild the UI with updated values. The Provider package offers a more scalable approach, allowing state to be stored and shared across multiple widgets in the widget tree, enabling cleaner code and easier data management for larger applications.

PROGRAM:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: SetStateCounter(),
    );
  }
}

class SetStateCounter extends StatefulWidget {
  @override
  _SetStateCounterState createState() => _SetStateCounterState();
}

class _SetStateCounterState extends State<SetStateCounter> {
  int _count = 0;

  void _increment() {
    setState(() {
      _count++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('State Management: SetState')),
      body: Center(
        child: Text('Count: $_count', style: TextStyle(fontSize: 32)),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _increment,
        child: Icon(Icons.add),
      ),
    );
  }
}
```

OUTPUT: