# Mobile Application Development Lab Report

## Department of Computer Science

**NUML**

**Student Name:** Muhammad Awais

**Roll No:** 2883

**Semester:** Fall 2025

**Instructor:** Sir Shahid Malik

**Date of Submission:** October 8, 2025

# Lab Task 1: Stateless and Statefull

## Objectives

- To understand the difference between **StatelessWidget** and **StatefulWidget** in Flutter.
- To learn how to manage and update the UI dynamically using the `setState()` method.
- To implement basic Flutter widgets such as `Text`, `Scaffold`, `ElevatedButton`, and `FloatingActionButton`.
- To develop a simple counter app demonstrating increment, decrement, and reset functionalities.
- To create a basic Flutter app that displays static text using a Stateless Widget.

## Description

This experiment demonstrates the core concepts of **Stateless** and **Stateful** widgets in Flutter.

**1. Hello World App (Stateless Widget):**

- A simple Flutter application is created using a `StatelessWidget`.
- The app displays a centered `Text` widget with the message “Hello World!”.
- This example helps understand that the UI remains constant, as Stateless widgets do not update dynamically.

**2. Counter App (Stateful Widget):**

- A `StatefulWidget` is used to create a counter application.
- The app uses `setState()` to rebuild the UI whenever the counter value changes.
- Three buttons (`+`, `-`, and `Reset`) allow users to increment, decrement, or reset the counter.
- A `FloatingActionButton` is also added for incrementing the counter quickly.
- The counter value is displayed dynamically using the `Text` widget and updates in real-time when user interacts.

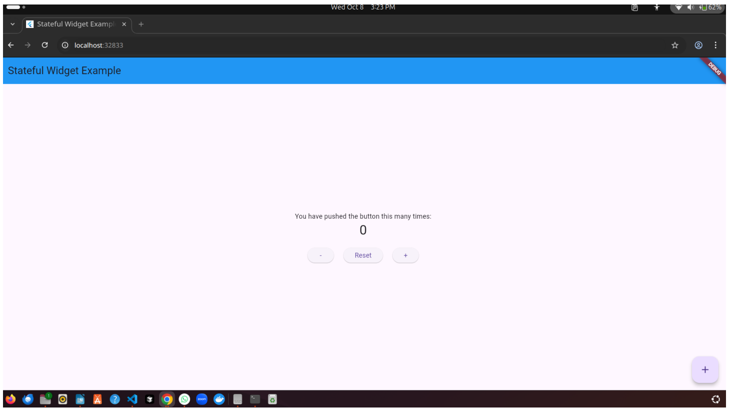
## Code Snippet stateless with statefull

```
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: ’Stateful Widget Example’,
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: const CounterApp(),
);
}
}
class CounterApp extends StatefulWidget {
const CounterApp({super.key});
@override
State<CounterApp> createState() => _CounterAppState();
}
class _CounterAppState extends State<CounterApp> {
int _counter = 0;
void _incrementCounter() {
setState(() {
_counter++;
});

}
void _decrementCounter() {
setState(() {
_counter--;
});
}
void _resetCounter() {
setState(() {
_counter = 0;
});
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: const Text(’Stateful Widget Example’),
backgroundColor: Colors.blue,
),
body: Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
const Text(
’You have pushed the button this many times:’,
),
Text(
’$_counter’,
style: Theme.of(context).textTheme.headlineMedium,
),
const SizedBox(height: 20),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
ElevatedButton(
onPressed: _decrementCounter,

child: const Text('-'),
),
const SizedBox(width: 20),
ElevatedButton(
onPressed: _resetCounter,
child: const Text('Reset'),
),
const SizedBox(width: 20),
ElevatedButton(
onPressed: _incrementCounter,
child: const Text('+'),
),
],
),
],
),
),
floatingActionButton: FloatingActionButton(
onPressed: _incrementCounter,
tooltip: 'Increment',
child: const Icon(Icons.add),
),
);
}
}
```

Listing 1: main.dart

## Output Screenshot

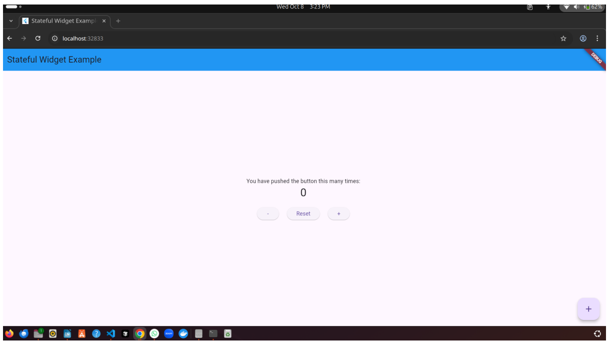


Stateful Widget Example

You have pushed the button this many times:

0

- Reset +



Stateful Widget Example

You have pushed the button this many times:

0

- Reset +

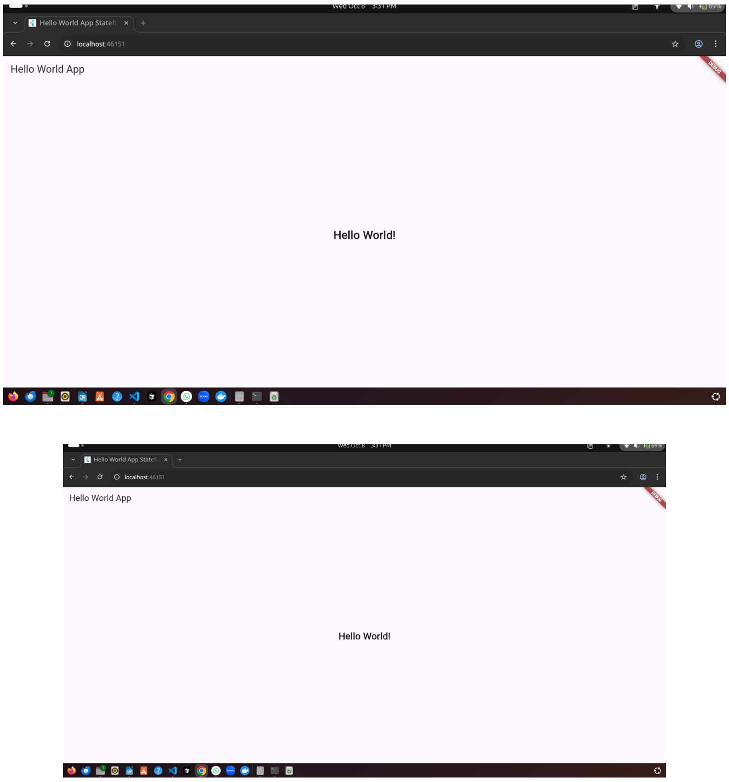
Figure 1: Main APP

## Code Snippet stateless

```
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: 'Hello World App Statefull',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: const HelloWorldWidget(),
);
}
}
class HelloWorldWidget extends StatelessWidget {
const HelloWorldWidget({super.key});
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: const Text('Hello World App'),
),
body: const Center(
child: Text(
'Hello World!',
style: TextStyle(
fontSize: 24,
fontWeight: FontWeight.bold,

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

Listing 2: main.dart

## Output Screenshot



Hello World App

Hello World!

Figure 2: Main APP

# Lab Task 2: Practice on Dart Widgets

## Objective

To understand and demonstrate the use of basic Flutter widgets such as Container, Row, Column, Stack, Padding, Image, and different types of buttons for designing simple and structured user interfaces.

## Description

In this experiment, a Flutter application is created using multiple widgets to explore how Flutter builds UI hierarchically using a **widget tree**.

The app demonstrates the following concepts:

- **Container with Text:**
  - Demonstrates the use of `Container` with margin, padding, color, and text styling.
  - Helps understand layout spacing and decoration.
- **Column with Circular Image and Text:**
  - Displays an image using `Image.asset()` and makes it circular using `ClipOval()`.
  - Shows vertical arrangement of widgets using `Column`.
- **Row with Icons and Text:**
  - Demonstrates horizontal alignment using `Row` and `mainAxisAlignment`.
  - Combines `Icon` and `Text` widgets for a simple feature bar (Call, Message, Share).
- **Stack Widget:**
  - Shows how to overlay widgets on top of each other.
  - Displays text on an image using `Stack` with a semi-transparent background.
- **Buttons (Elevated, Outlined, Text):**
  - Demonstrates interactive widgets that trigger actions via `onPressed()`.
- **Padding with Image and Text:**
  - Combines `Padding`, `Row`, and `Expanded` widgets to place text beside an image.
- **Decorated Container:**
  - Uses `BoxDecoration` with borders, colors, and rounded corners to enhance UI aesthetics.

The entire layout is wrapped inside a `SingleChildScrollView` to make the screen scrollable vertically when the content exceeds the display height.

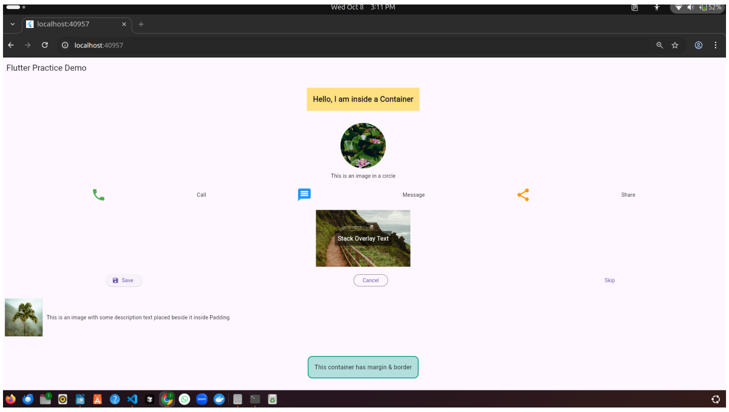
This experiment provides hands-on experience with layout management, widget alignment, and basic interactivity in Flutter's declarative UI framework.

## Code Snippet

```
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(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(title: const Text("Flutter Practice Demo")),
        body: Center(
          child: SingleChildScrollView(
            child: Column(
              children: [
                /// 1. CONTAINER with TEXT inside
                Container(
                  margin: const EdgeInsets.all(12),
                  padding: const EdgeInsets.all(16),
                  color: Colors.amber.shade200,
                  child: const Text(
                    "Hello, I am inside a Container",
                    style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
                  ),
                ),

                const SizedBox(height: 20),

                /// 2. IMAGE in Circle + Text Below (Column)
                Column(
                  children: [
                    ClipOval(
                      child: Image.asset(
                        "assets/images/p2.jpg",
                        width: 120,
                        height: 120,
                        fit: BoxFit.cover,
                      ),
                    ),
                    const SizedBox(height: 10),
                    const Text("This is an image in a circle"),
                  ],
                ),

                const SizedBox(height: 20),

                /// 3. ROW with ICONS and TEXT
                Row(
                  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                  children: const [
                    Icon(Icons.phone, color: Colors.green, size: 40),
                    Text("Call"),
                    Icon(Icons.message, color: Colors.blue, size: 40),
                    Text("Message"),
                    Icon(Icons.share, color: Colors.orange, size: 40),
                    Text("Share"),
                  ],
                ),

                const SizedBox(height: 20),

                /// 4. STACK with Text over Image
                Stack(
                  alignment: Alignment.center,
                  children: [
                    Image.asset(
                      "assets/images/p3.jpg",
                      width: 250,
                      height: 150,
                      fit: BoxFit.cover,
                    ),
                    Container(
                      color: Colors.black54,
                      padding: const EdgeInsets.all(8),
                      child: const Text(
                        "Stack Overlay Text",
                        style: TextStyle(color: Colors.white, fontSize: 16),
                      ),
                    ),
                  ],
                ),

                const SizedBox(height: 20),

                /// 5. BUTTON Row
                Row(
                  mainAxisAlignment: MainAxisAlignment.spaceAround,
                  children: [
                    ElevatedButton.icon(
                      onPressed: () => debugPrint("Saved!"),
                      icon: const Icon(Icons.save),
                      label: const Text("Save"),
                    ),
                    OutlinedButton(
                      onPressed: () => debugPrint("Cancelled!"),
                      child: const Text("Cancel"),
                    ),
                    TextButton(
                      onPressed: () => debugPrint("Skipped!"),
                      child: const Text("Skip"),
                    ),
                  ],
                ),

                const SizedBox(height: 20),

                /// 6. PADDING with IMAGE + TEXT side by side
                Padding(
                  padding: const EdgeInsets.all(12.0),
                  child: Row(
                    children: [
                      Image.asset("assets/images/p1.jpg",
                          width: 100, height: 100, fit: BoxFit.cover),
                      const SizedBox(width: 10),
                      const Expanded(
                        child: Text(
                          "This is an image with some description text placed beside it inside Padding",
                          style: TextStyle(fontSize: 14),
                        ),
                      ),
                    ],
                  ),
                ),

                const SizedBox(height: 20),

                /// 7. CONTAINER with Margin (different style)
                Container(
                  margin: const EdgeInsets.symmetric(vertical: 20, horizontal: 40),
                  padding: const EdgeInsets.all(16),


                  decoration: BoxDecoration(
                    color: Colors.teal.shade100,
                    borderRadius: BorderRadius.circular(12),
                    border: Border.all(color: Colors.teal, width: 2),
                  ),
                  child: const Text(
                    "This container has margin & border",
                    style: TextStyle(fontSize: 16, fontWeight: FontWeight.w500),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

Listing 3: main.dart

# Output Screenshot



Flutter Practice Demo

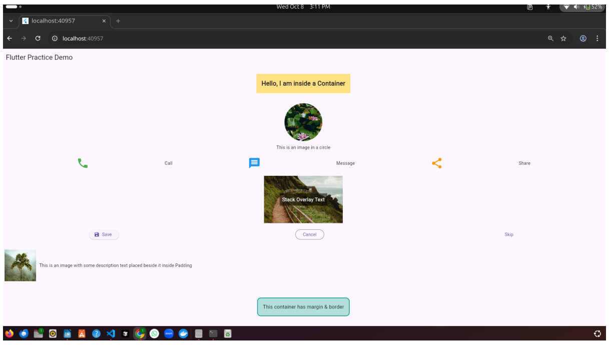
Hello, I am inside a Container

This is an image in a circle

Stack Overlay Text

This is an image with some description text placed beside it inside Padding

This container has margin & border


Flutter Practice Demo

Hello, I am inside a Container

This is an image in a circle

Stack Overlay Text

This is an image with some description text placed beside it inside Padding

This container has margin & border

Figure 3: Main APP