🧪 Flutter Lab 2 — Section 2: Widget State, Layout, and Hierarchy

---

🎯 Objectives

By the end of this lab, students will be able to:

1. Explain the difference between **Stateless** and **Stateful** widgets.
2. Use layout widgets (`Row`, `Column`, `Stack`, etc.) to arrange elements precisely.
3. Understand parent-child relationships and widget hierarchy.
4. Combine layout widgets to build multi-element UIs.

🧩 1. Stateless vs Stateful Widgets

🧠 Concept

In Flutter, every UI component is a **widget**, but not all widgets behave the same way over time.

Type	Description	Example Use Case
StatelessWidget	Has no internal state. Renders once and doesn't change unless rebuilt from outside.	Displaying static text, icons, or images.
StatefulWidget	Can change over time — reacts to user input, data, or animations.	Buttons, counters, forms, or animations.

🪶 Example 1 — StatelessWidget

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

class GreetingText extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text(
      'Welcome to Flutter!',
      style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
    );
  }
}
```

✅ The text will never change during runtime — it's "stateless."

⚙️ Example 2 — StatefulWidget

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

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

class CounterApp extends StatefulWidget {
  @override
  _CounterAppState createState() => _CounterAppState();
}

class _CounterAppState extends State<CounterApp> {
  int counter = 0;

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('Counter Example')),
        body: Center(
          child: Text(
            'Count: $counter',
            style: TextStyle(fontSize: 28),
          ),
        ),
        floatingActionButton: FloatingActionButton(
          onPressed: () {
            setState(() {
              counter++;
            });
          },
          child: Icon(Icons.add),
        ),
      ),
    );
  }
}
```

✅ When you press the button, `setState()` triggers a rebuild — the UI updates instantly.

🧱 2. Layout Basics — The Widget Tree

Every UI you build is a **tree** of widgets:
parents contain children → children can contain more children → forming the **Widget Tree**.

Example Visualization:

```
Scaffold
 ├─ AppBar
 └─ Body (Column)
      ├─ Text
      ├─ Row
      │   ├─ Icon
      │   └─ Text
      └─ ElevatedButton
```

🧩 3. Row and Column

These are the fundamental layout widgets in Flutter.

➡️ Row — Horizontal Layout

```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceAround,
  children: [
    Icon(Icons.home, color: Colors.blue),
    Icon(Icons.star, color: Colors.orange),
    Icon(Icons.person, color: Colors.green),
  ],
);
```

🧠 Concept:

- `mainAxisAlignment` controls horizontal distribution.
- `crossAxisAlignment` controls vertical alignment.

⬇️ Column — Vertical Layout

```
Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text('Name: Flutter User'),
    Text('Email: flutter@dev.com'),
    ElevatedButton(onPressed: () {}, child: Text('Edit Profile')),
  ],
);
```

🧠 Tip:
Use `Column` for stacked elements like forms, lists, and info cards.

🪆 4. Nesting Layouts

You can combine Rows and Columns to form complex hierarchies.

```
Column(
  children: [
    Text('Account'),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Text('Username'),
        Text('FlutterDev'),
      ],
    ),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Text('Email'),
        Text('dev@flutter.com'),
      ],
    ),
  ],
);
```

🧩 5. Stack and Positioned

A **Stack** allows widgets to overlap — useful for badges, banners, or layers.

```
Stack(
  alignment: Alignment.center,
  children: [
    Container(width: 200, height: 200, color: Colors.blue),
    Container(width: 100, height: 100, color: Colors.orange),
    Text('Top Layer', style: TextStyle(color: Colors.white)),
  ],
);
```

With `Positioned`, you can place elements precisely:

```
Stack(
  children: [
    Image.network('https://picsum.photos/300'),
    Positioned(
      bottom: 20,
      right: 20,
      child: Icon(Icons.favorite, color: Colors.red, size: 32),
    ),
  ],
);
```

⚖️ 6. Expanded and Flexible

These control how widgets **share available space**.

```
Row(
  children: [
    Expanded(
      child: Container(color: Colors.blue, height: 100),
    ),
    Expanded(
      child: Container(color: Colors.green, height: 100),
    ),
  ],
);
```

🧠 **Expanded:** fills remaining space.
🧠 **Flexible:** lets child decide size but respects constraints.

🧩 7. Alignment and Spacing

Flutter provides precise control over how widgets are aligned.

```
Container(
  color: Colors.amber,
  height: 200,
  child: Align(
    alignment: Alignment.bottomRight,
    child: Text('Bottom Right'),
  ),
);
```

or use `Center`, `Padding`, and `Spacer` for simplicity.

🧠 8. Key Takeaways

- **StatelessWidget** = static appearance.
- **StatefulWidget** = dynamic, can rebuild via `setState()`.
- **Row** and **Column** are your main layout tools.
- **Stack** overlays elements.
- **Expanded/Flexible** balance available space.
- Everything is a **child of something** — the *Widget Tree* defines structure.

🧪 9. Exercises

🧩 Exercise 1 — "Counter with Layout"

Modify the counter app:

- Add a title above the count using a `Column`.
- Add a `Row` of two FABs: one to increment, one to decrement.

🧩 Exercise 2 — "Profile Layout"

Build this layout:

```
+-------------------------------------+
| [Avatar]                            |
| Name: Flutter User                  |
| Bio: Loves widgets & hot reload.    |
| [Edit Profile]                      |
+-------------------------------------+
```

Hints:

- Use a `Column`.
- Use `CircleAvatar`, `Text`, `ElevatedButton`.
- Add `Padding` and `Center` for spacing.

🧩 Exercise 3 — "Stack Badge"

Place an image and overlay a red notification badge at the top-right.

Hint: `Stack` + `Positioned`.

🏆 Challenge Task

Recreate this layout:

```
+-----------------------------------------------+
| AppBar: "Dashboard"                           |
|-----------------------------------------------|
| [Profile Image]     Flutter Dev               |
|                   flutter@devmail.com         |
|-----------------------------------------------|
| [Row of 3 buttons: Home | Stats | Settings]   |
|-----------------------------------------------|
| [Large Card with text centered inside]        |
+-----------------------------------------------+
```

Bonus:
Make one of the buttons toggle the text inside the card using a **StatefulWidget**.