🧪 Flutter Lab 2 — Section 1: Widget Catalog & Composition Basics

---

🎯 Objectives

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

1. Recognize and use the most common Flutter widgets.
2. Understand the idea of a **widget tree** and **composition**.
3. Combine widgets to create simple, visually pleasing screens.
4. Experiment with **visual and interactive widgets** (text, images, buttons, inputs).

🧩 1. Introduction: Everything is a Widget

In Flutter, *everything* you see on the screen is a widget:

- Texts
- Images
- Buttons
- Layout containers
- Even the whole app itself

Widgets form a **tree** — each widget can contain others inside it.

```
void main() {
  runApp(
    MaterialApp(
      home: Center(
        child: Text('Hello Flutter!'),
      ),
    ),
  );
}
```

🪴 Think of a Flutter UI like a **plant**:

- The root (`runApp`) gives life to the tree.
- The branches (`MaterialApp`, `Scaffold`, `Container`) organize it.
- The leaves (`Text`, `Icon`, `Image`) make it beautiful.

🧱 2. Structural Widgets

These widgets define the *structure* of your screen.

`Scaffold`

Provides a high-level layout structure for Material apps.

```
Scaffold(
  appBar: AppBar(title: Text('My First App')),
  body: Center(
    child: Text('Hello world!'),
  ),
);
```

🧠 **Concept:**
Use `Scaffold` as your main screen container.
It automatically provides areas for the AppBar, Drawer, FAB, etc.

`Container`

A versatile box model widget for styling and layout control.

```
Container(
  color: Colors.blueAccent,
  padding: EdgeInsets.all(16),
  margin: EdgeInsets.all(8),
  child: Text('I live inside a box!'),
);
```

🧠 **Concept:**
`Container` = background + padding + size + alignment in one.

`Center`, `Padding`, `Align`

Used to control *where* your widget sits on the screen.

```
Center(child: Text('Centered!'));

Padding(
  padding: EdgeInsets.symmetric(horizontal: 20),
  child: Text('With some padding'),
);
```

🧩 3. Content Widgets

`Text`

Displays text with styling.

```
Text(
  'Flutter UI Design',
  style: TextStyle(
    fontSize: 24,
    fontWeight: FontWeight.bold,
    color: Colors.deepPurple,
  ),
);
```

`Image`

Displays images from assets, network, or memory.

```
Image.network(
  'https://flutter.dev/images/flutter-logo-sharing.png',
  width: 120,
);
```

`Icon`

Displays material icons.

```
Icon(
  Icons.favorite,
  color: Colors.pink,
  size: 32,
);
```

🧩 4. Interactive Widgets

`ElevatedButton`

```
ElevatedButton(
  onPressed: () {
    print('Button pressed!');
  },
  child: Text('Press Me'),
);
```

`TextField`

```
TextField(
  decoration: InputDecoration(
    labelText: 'Enter your name',
    border: OutlineInputBorder(),
  ),
);
```

🧩 5. Composition Example

Let’s combine what we learned into a small “Profile Card” UI.

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

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

class ProfileApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.grey[200],
        appBar: AppBar(title: Text('Profile Card')),
        body: Center(
          child: Container(
            padding: EdgeInsets.all(20),
            margin: EdgeInsets.all(16),
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(20),
              boxShadow: [
                BoxShadow(
                  color: Colors.black26,
                  blurRadius: 10,
                  offset: Offset(0, 4),
                ),
              ],
            ),
            child: Column(
              mainAxisSize: MainAxisSize.min,
              children: [
                CircleAvatar(
                  radius: 40,
                  backgroundImage: NetworkImage(
                    'https://i.pravatar.cc/150?img=3',
                  ),
                ),
                SizedBox(height: 12),
                Text('Anas Shaikhany',
                    style: TextStyle(
                        fontSize: 20, fontWeight: FontWeight.bold)),
                Text('Flutter Developer',
                    style: TextStyle(color: Colors.grey[600])),
                SizedBox(height: 10),
                ElevatedButton(
                  onPressed: () {},
                  child: Text('Follow'),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

🧠 6. Key Takeaways

- Every element on the screen is a **widget**.
- Widgets can contain other widgets (composition).
- `Scaffold` defines structure; `Container` defines style.
- Layout and style are achieved by combining small building blocks.

🧩 7. Exercises

🧪 Exercise 1 — “Settings Card”

Create a card with:

- A title “Settings”
- Two `Row`s:
  - Row 1: Icon (volume) + “Sound”
  - Row 2: Icon (wifi) + “Wi-Fi”

🧪 Exercise 2 — “Login Form”

Design a simple form using:

- Two `TextField`s (Email, Password)
- One `ElevatedButton` (Login)
- Hint: Wrap with `Padding` and `Column`

🧪 Exercise 3 — “Info Panel”

Display a horizontal layout:

- Left: an `Icon`
- Center: `Text` with a title and subtitle
- Right: a small `ElevatedButton`

🏆 Challenge Task

Recreate the following layout (describe or show in class):

```
+--------------------------------------+
| AppBar: "My Card"                    |
|--------------------------------------|
|   [Avatar]  Name: Flutter User       |
|             Email: user@mail.com     |
|                                      |
|          [Contact Me Button]         |
+--------------------------------------+
```

🧩 *Tip:* Use `Row`, `Column`, `CircleAvatar`, and `Padding` creatively.

🧰 Bonus Tools

- Run your app with **Hot Reload** to see changes instantly.
- Try `Flutter Inspector` to visualize the widget tree.
- Enable **debugPaintSizeEnabled** to see widget boundaries:

  ```
  debugPaintSizeEnabled = true;
  ```