PRACTICAL 3

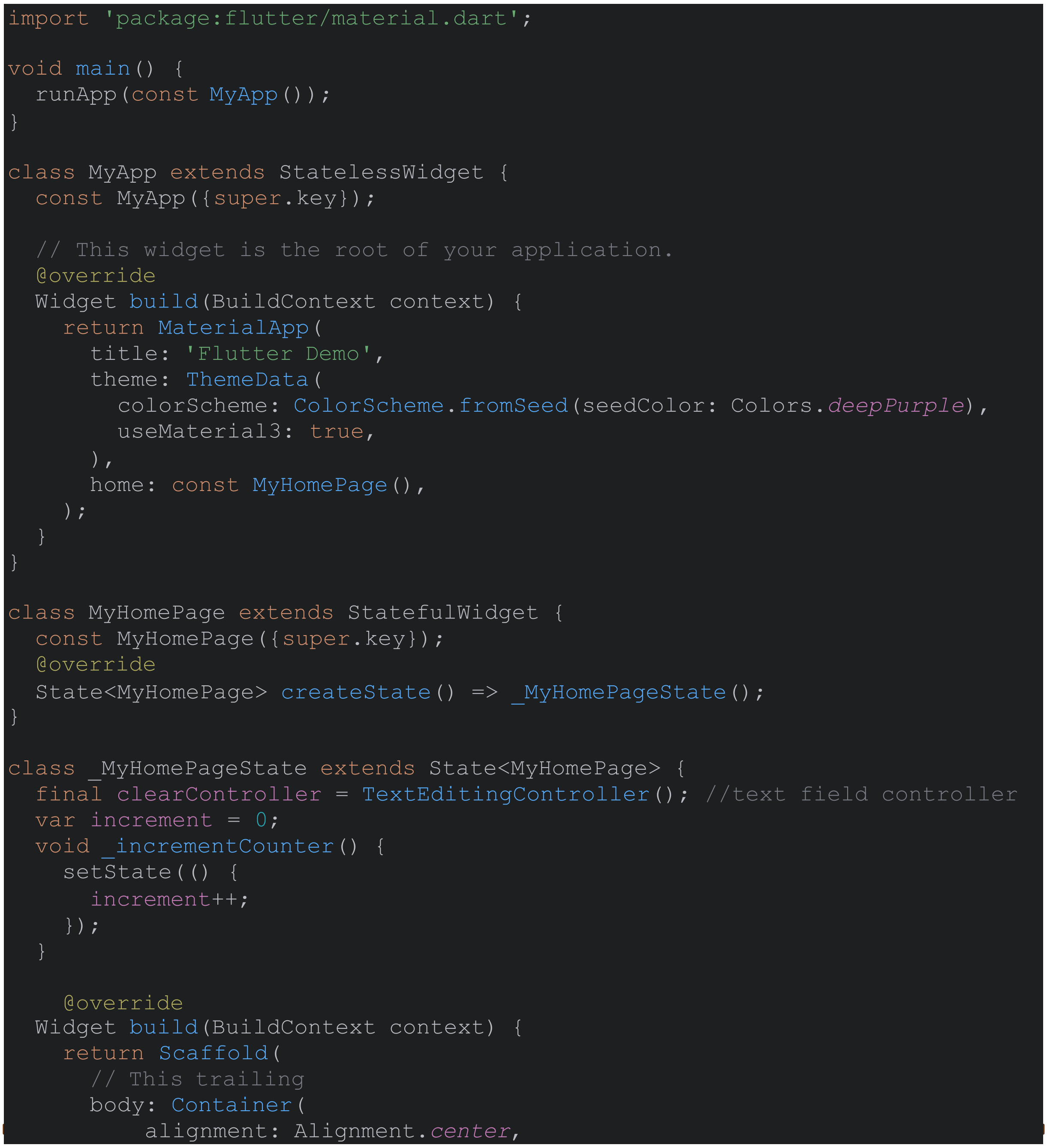
AIM:

Create an application using user interfaces with Flutter. (Basic Widgets, Material Components, Handling Gestures)

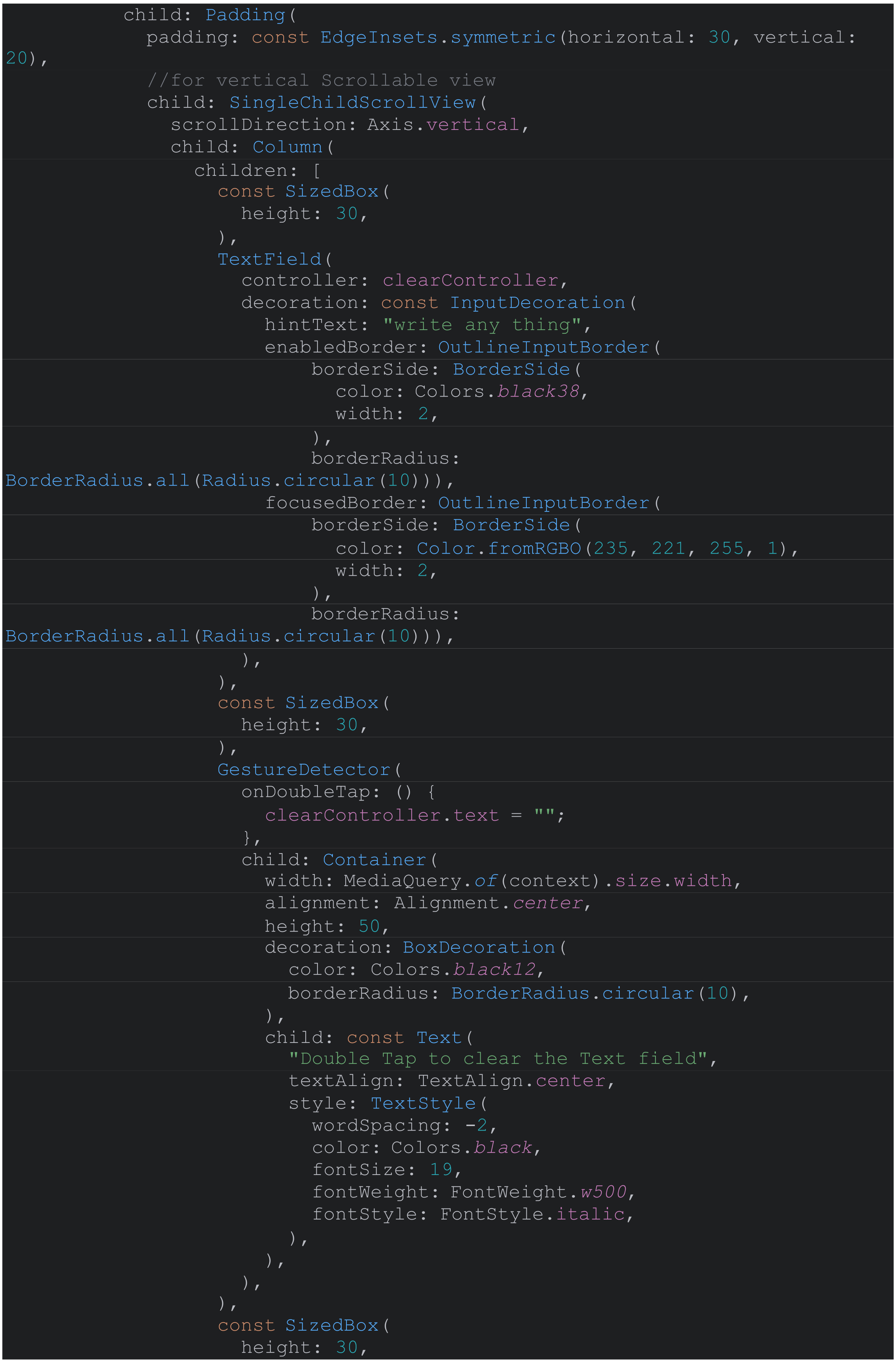
THEORY:

- **Widget:** Flutter apps are built with a composition of widgets, each representing a UI element or layout.
- **Material Design:** The app uses Material Components, a set of pre-designed widgets
- **Gesture Detection:** Flutter allows handling various gestures (taps, double taps, swipes, etc.) using **GestureDetector** widgets.

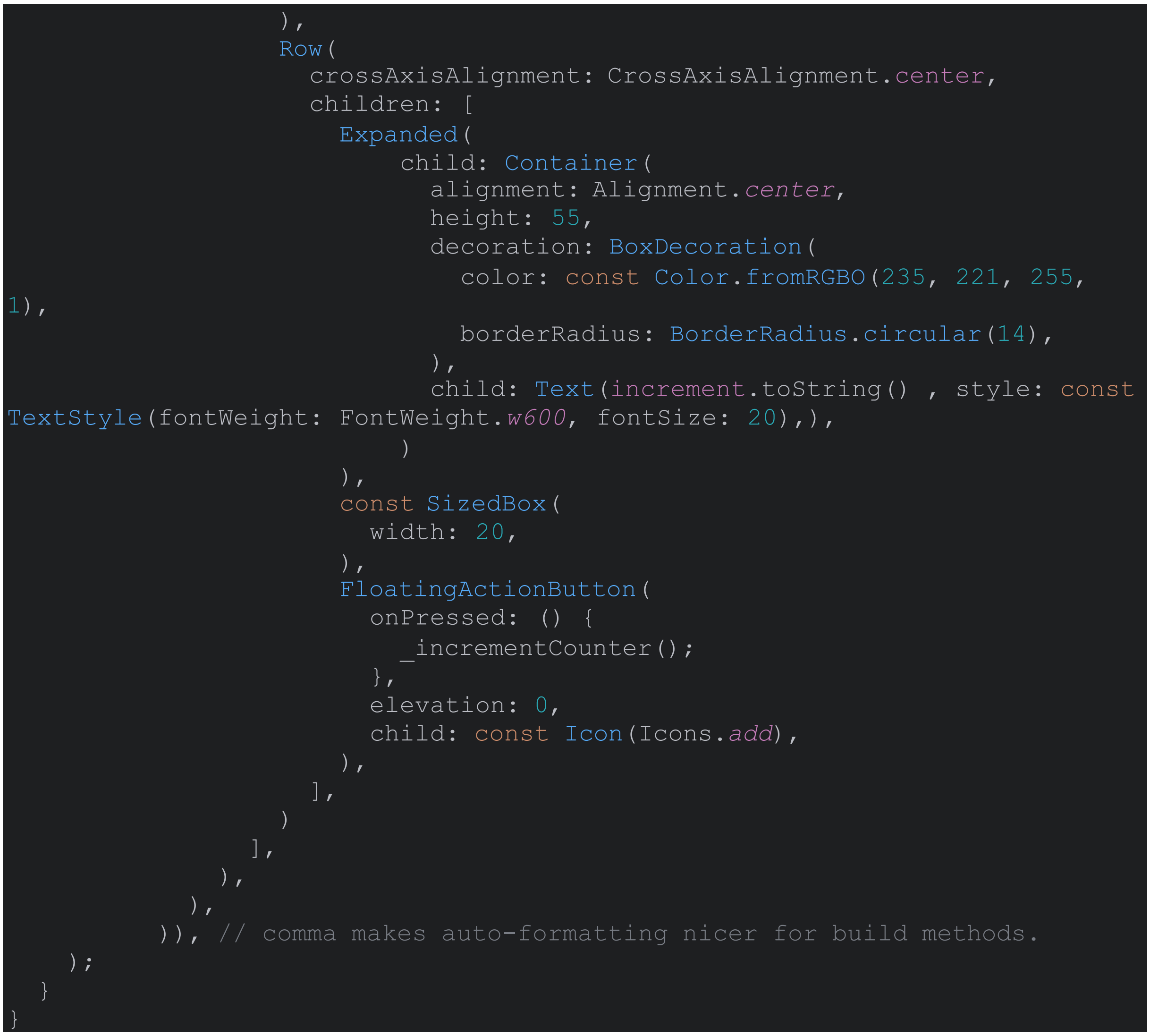
CODE:



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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key});
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  final clearController = TextEditingController(); //text field controller
  var increment = 0;
  void _incrementCounter() {
    setState(() {
      increment++;
    });
  }

    @override
  Widget build(BuildContext context) {
    return Scaffold(
      // This trailing
      body: Container(
          alignment: Alignment.center,
```



```
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 30, vertical:
20),
            //for vertical Scrollable view
            child: SingleChildScrollView(
              scrollDirection: Axis.vertical,
              child: Column(
                children: [
                  const SizedBox(
                    height: 30,
                  ),
                  TextField(
                    controller: clearController,
                    decoration: const InputDecoration(
                      hintText: "write any thing",
                      enabledBorder: OutlineInputBorder(
                          borderSide: BorderSide(
                            color: Colors.black38,
                            width: 2,
                          ),
                          borderRadius:
BorderRadius.all(Radius.circular(10))),
                      focusedBorder: OutlineInputBorder(
                          borderSide: BorderSide(
                            color: Color.fromRGBO(235, 221, 255, 1),
                            width: 2,
                          ),
                          borderRadius:
BorderRadius.all(Radius.circular(10))),
                    ),
                  ),
                  const SizedBox(
                    height: 30,
                  ),
                  GestureDetector(
                    onDoubleTap: () {
                      clearController.text = "";
                    },
                    child: Container(
                      width: MediaQuery.of(context).size.width,
                      alignment: Alignment.center,
                      height: 50,
                      decoration: BoxDecoration(
                        color: Colors.black12,
                        borderRadius: BorderRadius.circular(10),
                      ),
                      child: const Text(
                        "Double Tap to clear the Text field",
                        textAlign: TextAlign.center,
                        style: TextStyle(
                          wordSpacing: -2,
                          color: Colors.black,
                          fontSize: 19,
                          fontWeight: FontWeight.w500,
                          fontStyle: FontStyle.italic,
                        ),
                      ),
                    ),
                  ),
                  const SizedBox(
                    height: 30,
```



```
                  ),
                  Row(
                    crossAxisAlignment: CrossAxisAlignment.center,
                    children: [
                      Expanded(
                          child: Container(
                            alignment: Alignment.center,
                            height: 55,
                            decoration: BoxDecoration(
                              color: const Color.fromRGBO(235, 221, 255,
1),

                              borderRadius: BorderRadius.circular(14),
                            ),
                            child: Text(increment.toString() , style: const
TextStyle(fontWeight: FontWeight.w600, fontSize: 20),),
                          )
                      ),
                      const SizedBox(
                        width: 20,
                      ),
                      FloatingActionButton(
                        onPressed: () {
                          _incrementCounter();
                        },
                        elevation: 0,
                        child: const Icon(Icons.add),
                      ),
                    ],
                  )
                ],
              ),
            ),
          )), // comma makes auto-formatting nicer for build methods.
    );
  }
}
```

OUTPUT:

Here, we enter the values in text field. This text field has the **TextEditingController** named clearController.

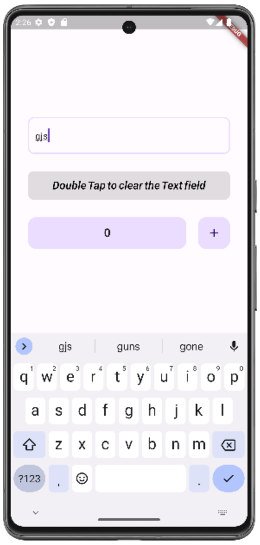


Figure 1 text field with the TextEditingController

Now, on double tapping the container the text field will be empty again, because of the **gestureDetector** widget wrapped around the container, and is tasked to erase the text field to empty again, with the help of the 'clearController'.

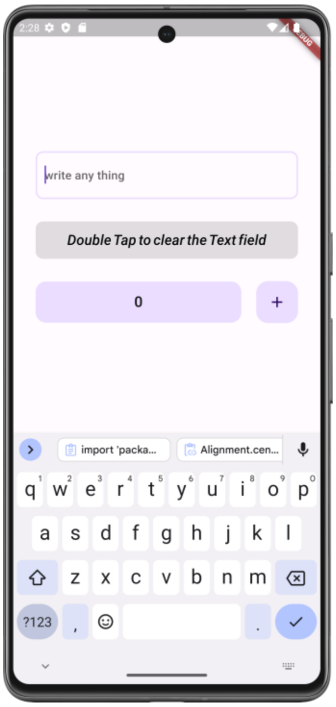


Figure 2 gesture detector used for clearing text field

Then, the **FloatingActionButton** the '+' sign increments the counter and display at the container to it's right.

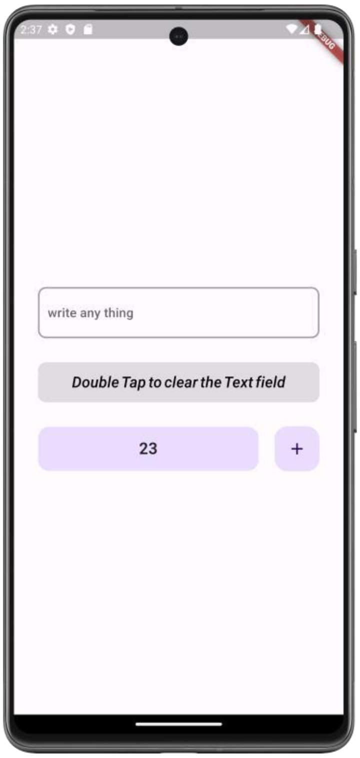


Figure 3 increment counter

Latest Applications:

For providing more simpler and understandable UI to an application and assigning actions to the gestures perform by the user.

Learning Outcome:

- Flutter's declarative UI approach makes building interfaces efficient and readable.
- Widgets are composable, allowing you to create complex UI from simpler building blocks
- Material Design provides easy visual and interaction language for apps.
- Gesture detectors enable intuitive user interactions beyond simple button presses.