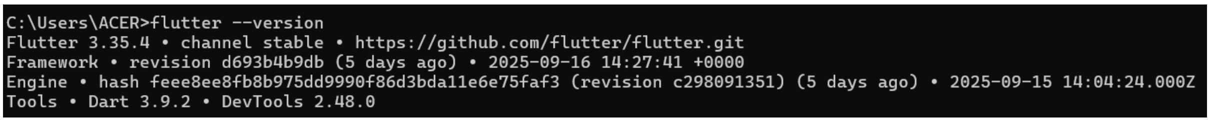
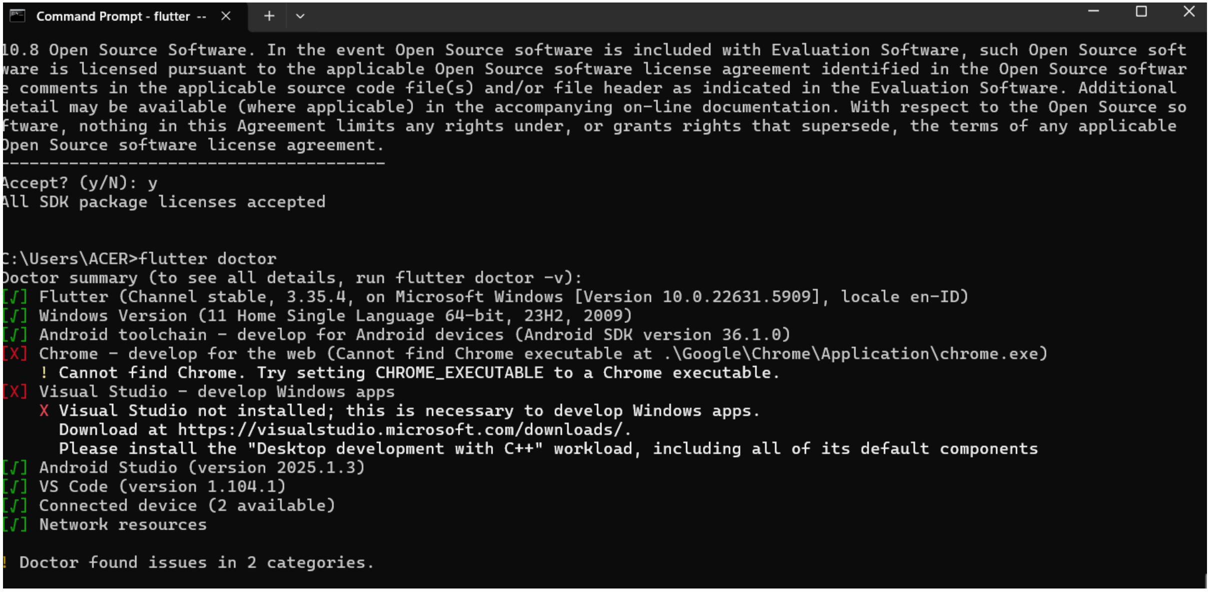
1. Instal Flutter

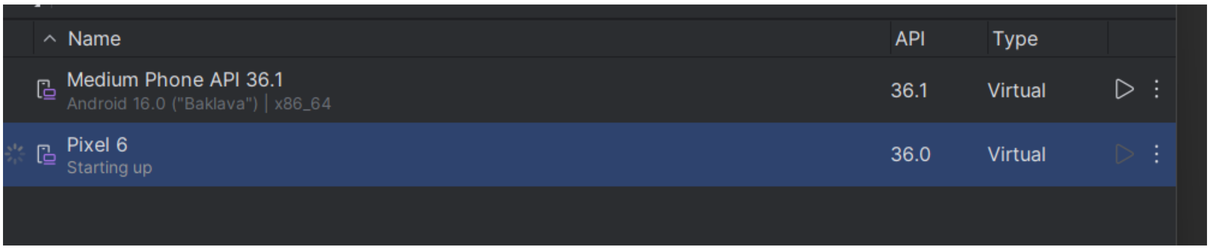


```
C:\Users\ACER>flutter --version
Flutter 3.35.4 • channel stable • https://github.com/flutter/flutter.git
Framework • revision d693b4b9db (5 days ago) • 2025-09-16 14:27:41 +0000
Engine • hash feee8ee8fb8b975dd9990f86d3bda11e6e75faf3 (revision c298091351) (5 days ago) • 2025-09-15 14:04:24.000Z
Tools • Dart 3.9.2 • DevTools 2.48.0
```



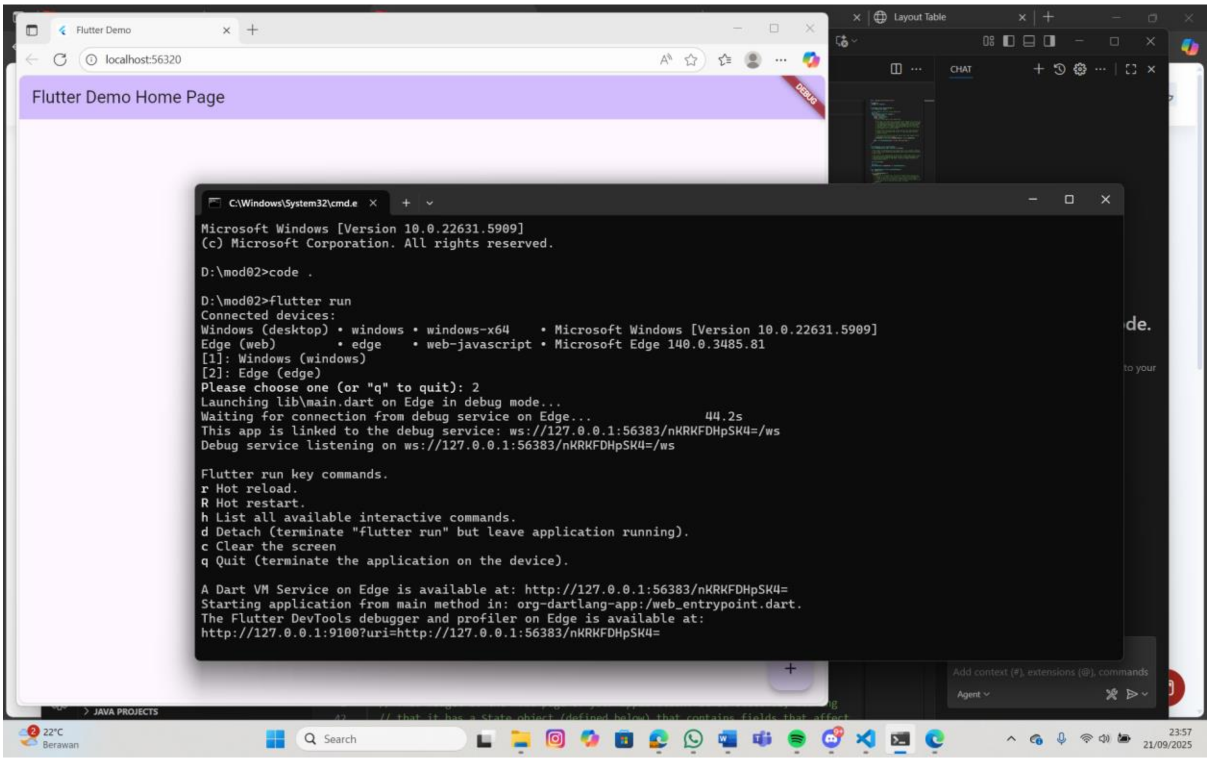
```
10.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source soft
ware is licensed pursuant to the applicable Open Source software license agreement identified in the Open Source softwar
e comments in the applicable source code file(s) and/or file header as indicated in the Evaluation Software. Additional
detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Open Source so
ftware, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable
Open Source software license agreement.
---------------------------------------
Accept? (y/N): y
All SDK package licenses accepted


C:\Users\ACER>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.35.4, on Microsoft Windows [Version 10.0.22631.5909], locale en-ID)
[√] Windows Version (11 Home Single Language 64-bit, 23H2, 2009)
[√] Android toolchain - develop for Android devices (Android SDK version 36.1.0)
[X] Chrome - develop for the web (Cannot find Chrome executable at .\Google\Chrome\Application\chrome.exe)
    ! Cannot find Chrome. Try setting CHROME_EXECUTABLE to a Chrome executable.
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[√] Android Studio (version 2025.1.3)
[√] VS Code (version 1.104.1)
[√] Connected device (2 available)
[√] Network resources

! Doctor found issues in 2 categories.
```

2. Virtual Device

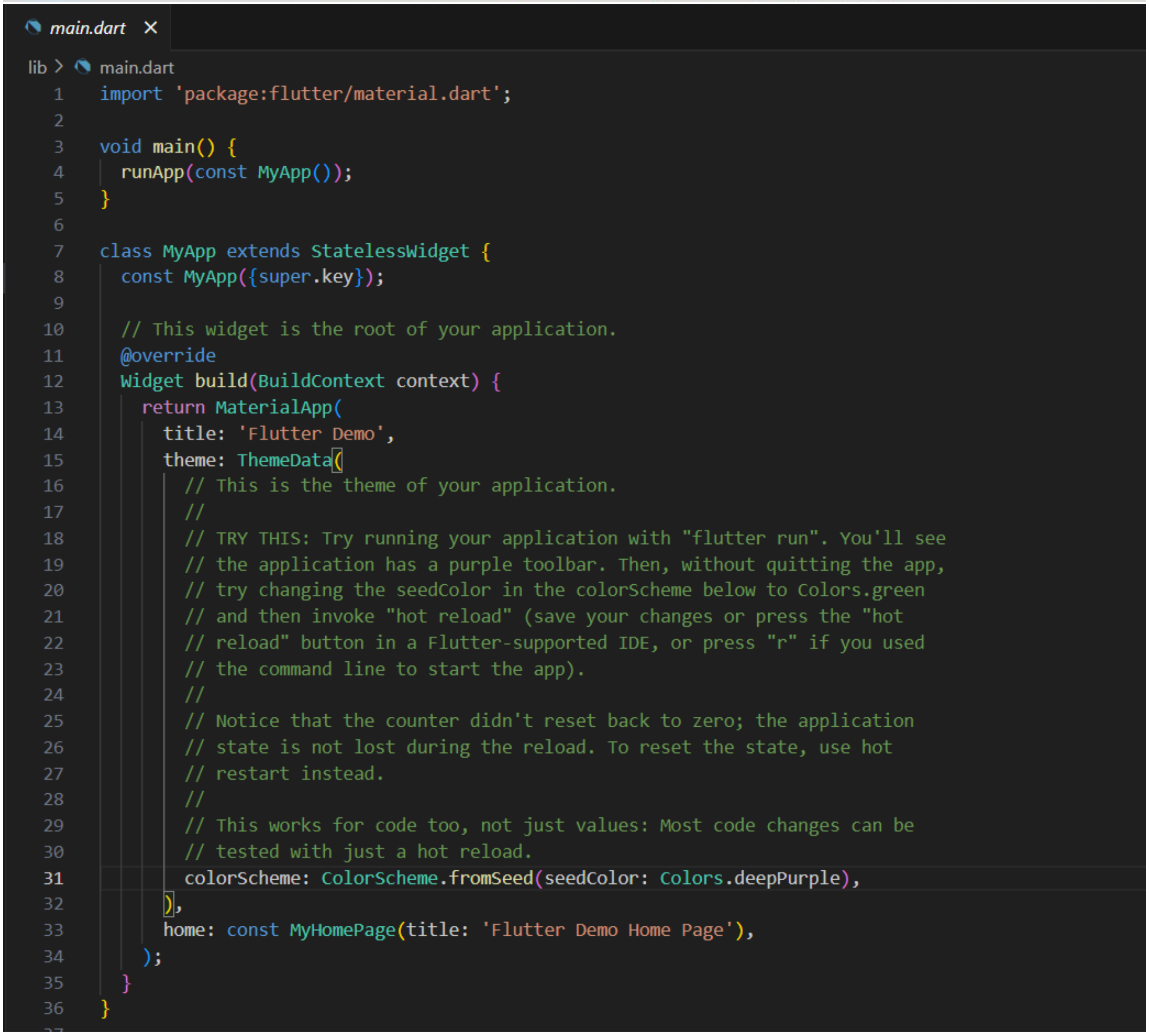


Name	API	Type
Medium Phone API 36.1 Android 16.0 ("Baklava") | x86_64	36.1	Virtual
Pixel 6 Starting up	36.0	Virtual

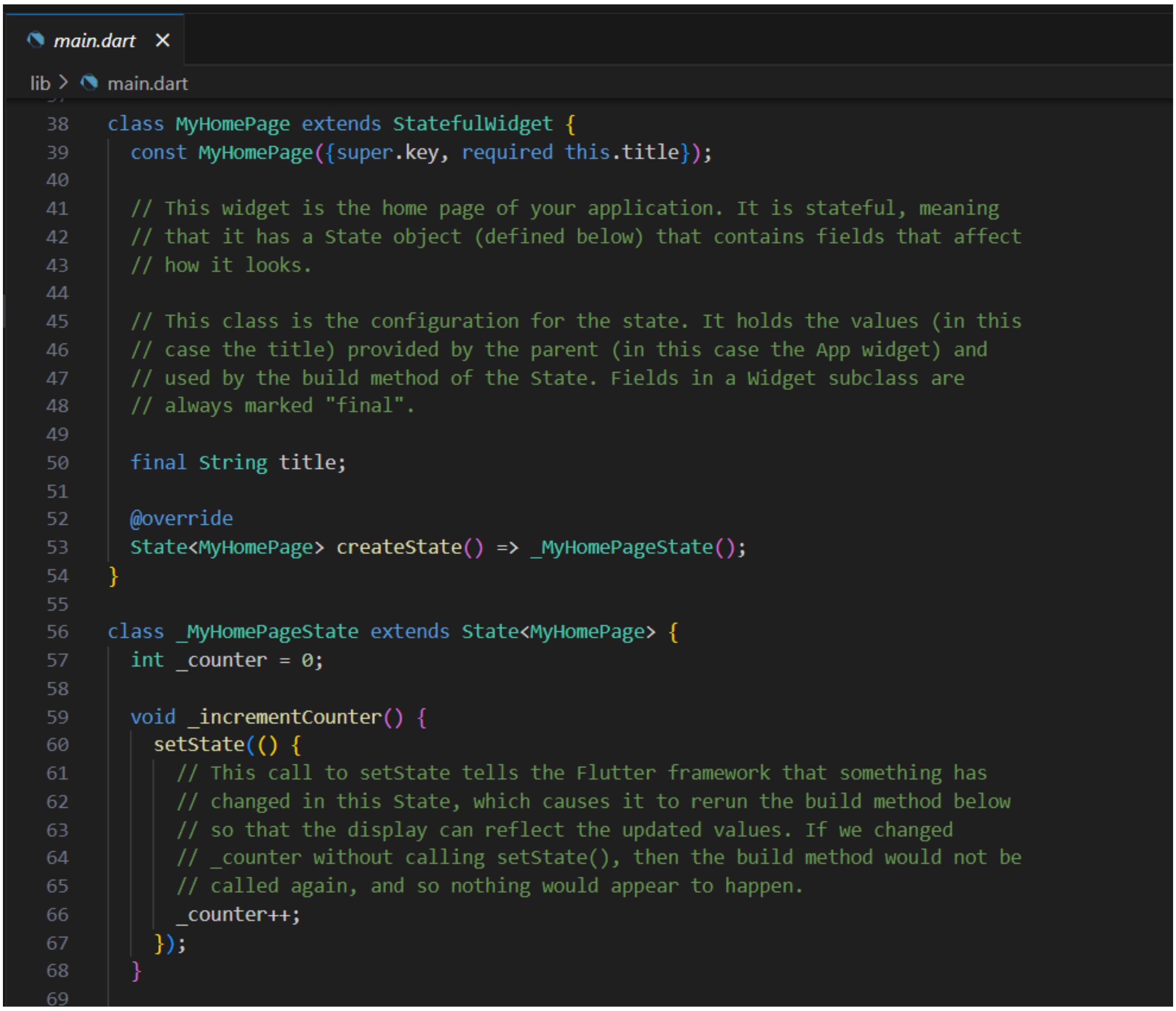
3. Project Flutter



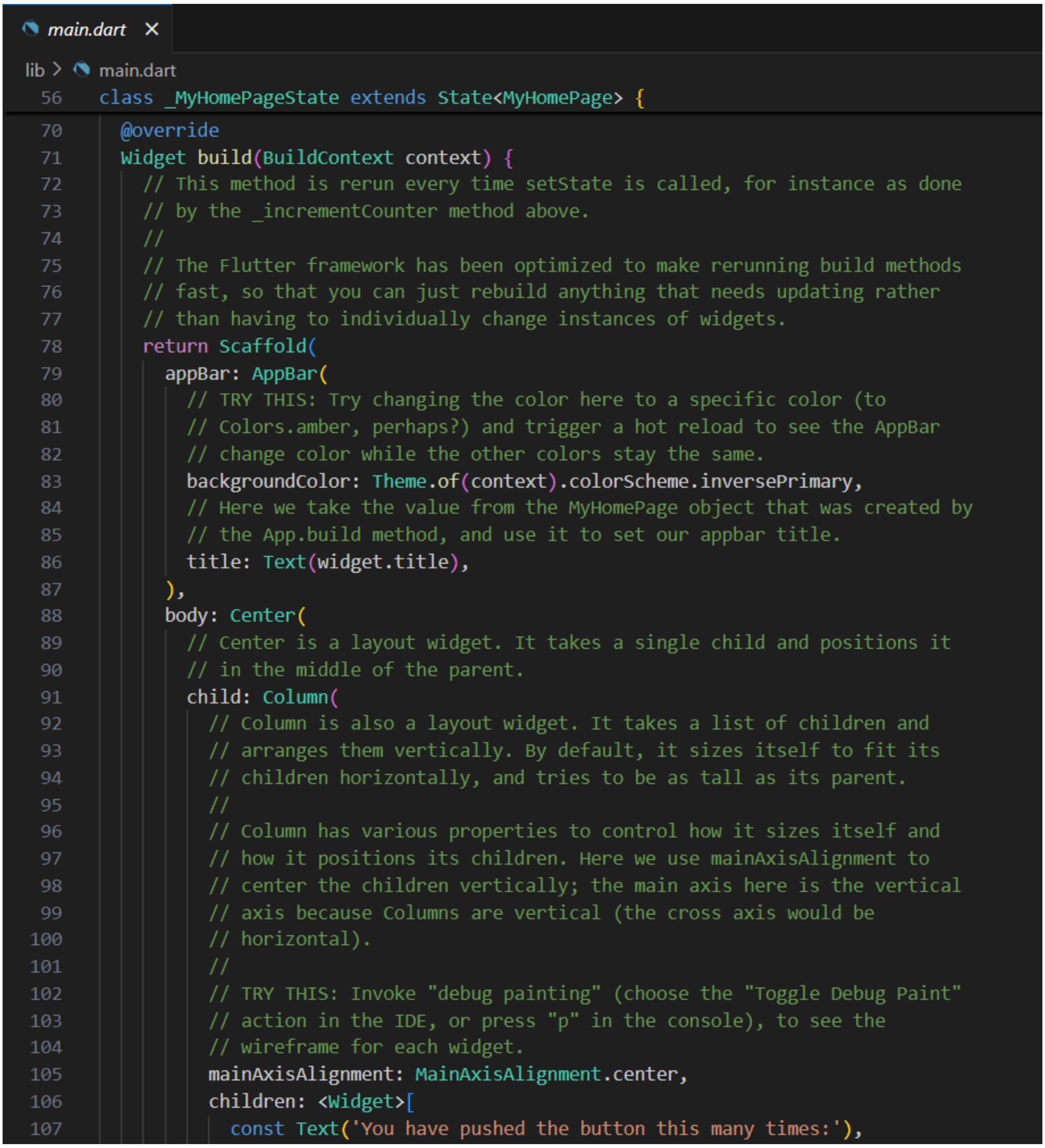
```
Microsoft Windows [Version 10.0.22631.5909]
(c) Microsoft Corporation. All rights reserved.

D:\mod02>code .

D:\mod02>flutter run
Connected devices:
Windows (desktop) • windows • windows-x64    • Microsoft Windows [Version 10.0.22631.5909]
Edge (web)        • edge    • web-javascript • Microsoft Edge 140.0.3485.81
[1]: Windows (windows)
[2]: Edge (edge)
Please choose one (or "q" to quit): 2
Launching lib\main.dart on Edge in debug mode...
Waiting for connection from debug service on Edge...               44.2s
This app is linked to the debug service: ws://127.0.0.1:56383/nKRKFDHpSK4=/ws
Debug service listening on ws://127.0.0.1:56383/nKRKFDHpSK4=/ws

Flutter run key commands.
r Hot reload.
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

A Dart VM Service on Edge is available at: http://127.0.0.1:56383/nKRKFDHpSK4=
Starting application from main method in: org-dartlang-app:/web_entrypoint.dart.
The Flutter DevTools debugger and profiler on Edge is available at:
http://127.0.0.1:9100?uri=http://127.0.0.1:56383/nKRKFDHpSK4=
```



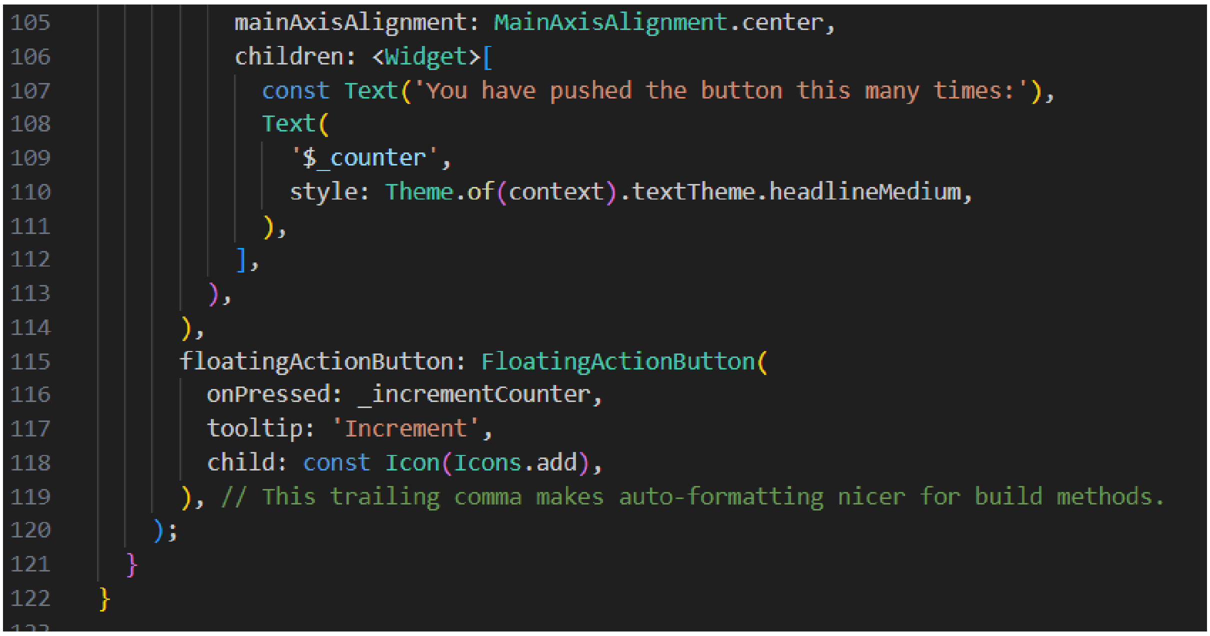
```dart
import 'package:flutter/material.dart';

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(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}
```



```dart
class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
      // so that the display can reflect the updated values. If we changed
      // _counter without calling setState(), then the build method would not be
      // called again, and so nothing would appear to happen.
      _counter++;
    });
  }

```



```dart
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    // This method is rerun every time setState is called, for instance as done
    // by the _incrementCounter method above.
    //
    // The Flutter framework has been optimized to make rerunning build methods
    // fast, so that you can just rebuild anything that needs updating rather
    // than having to individually change instances of widgets.
    return Scaffold(
      appBar: AppBar(
        // TRY THIS: Try changing the color here to a specific color (to
        // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
        // change color while the other colors stay the same.
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        // Here we take the value from the MyHomePage object that was created by
        // the App.build method, and use it to set our appbar title.
        title: Text(widget.title),
      ),
      body: Center(
        // Center is a layout widget. It takes a single child and positions it
        // in the middle of the parent.
        child: Column(
          // Column is also a layout widget. It takes a list of children and
          // arranges them vertically. By default, it sizes itself to fit its
          // children horizontally, and tries to be as tall as its parent.
          //
          // Column has various properties to control how it sizes itself and
          // how it positions its children. Here we use mainAxisAlignment to
          // center the children vertically; the main axis here is the vertical
          // axis because Columns are vertical (the cross axis would be
          // horizontal).
          //
          // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
          // action in the IDE, or press "p" in the console), to see the
          // wireframe for each widget.
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text('You have pushed the button this many times:'),
```



```
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text('You have pushed the button this many times:'),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headlineMedium,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ), // This trailing comma makes auto-formatting nicer for build methods.
    );
  }
}
```