IS487
Emerging Topics in IS

Dr. Abbas Malik
amaalik@psu.edu.sa

Dr Abbas Malik

- Email: amaalik@psu.edu.sa
- Mobile: 053 186 4813 (**join Whatsapp Group**)
- Office: Building 101, Floor 2, Room E364
- PhD in **Computer Science** (2010), **France**
- Languages: **English**, **French**, **Punjabi**, **Urdu**, **Hindi** and also a bit of **Arabic**

Machine Learning

NLP/MT/ASP
Data Mining

Data Analytics

Course Info

- **Classes**: Sunday to Wednesday at 3:00 – 3:50 PM
- **Room**: Lab *GA13*
- **Marks distribution**

Assessment	Weight %age	Schedule
Certificate (Coursera)	5	Week 12
Assignment	5%	Week 10
Project	15%	Throughout semester
Mid-Term (Practical)	20%	Week 9
Graded Labs (4 labs)	10%	Thought semester
Attendance	5%	
Final Exam	40%	


TIME TO START

What we will learn

1. Foundation & Flutter Mindset
2. Crafting Beautiful & Responsive UI
3. Making Apps Interactive & Dynamic
4. Build Multi-Screen Experience
5. Working with Real Data
6. Tapping into Native Device Features
7. Polishing and Publishing Apps

What is Mobile App Development?

- Process of **creating applications** for **portable devices**

Portable Devices: Compact lightweight computing devices that can be easily carried around

  - Smartphones, tablets, smartwatches, e-readers, gaming consoles, …
- Entertainment and work on the move

Mobile Devices History

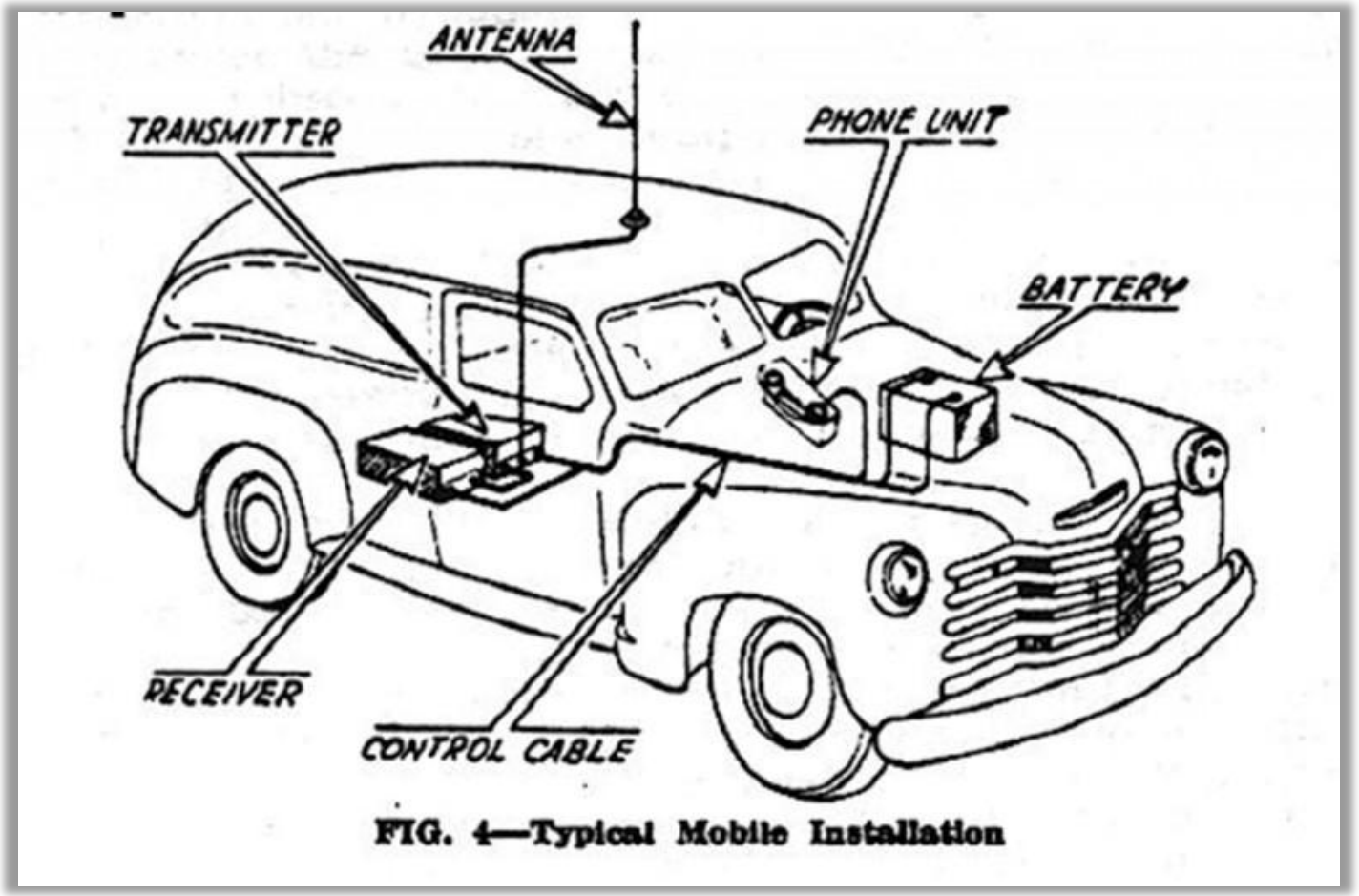
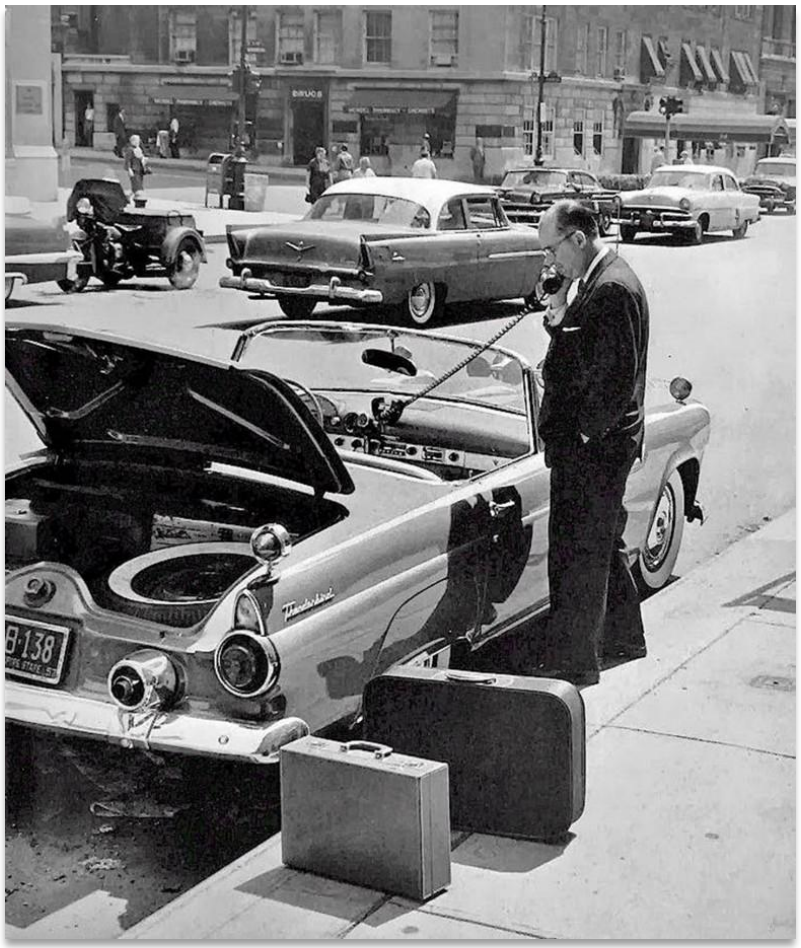


FIG. 4—Typical Mobile Installation



Portable Device Evolution

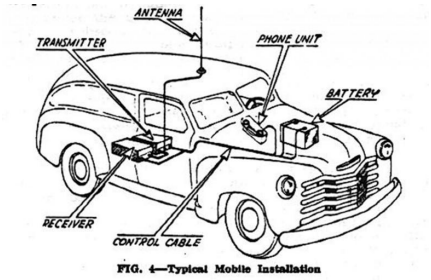


FIG. 4—Typical Mobile Installation

2007

Portable Device Evolution

- **Early Days**
  - PDA & mobile offering basic functionalities: contacts, Voice calls
- **Modern Era**
  - Modern devices: high resolution screens, extensive connectivity
- **Future Trends**
  - High speed connectivity, AR, VR, AI, DA, …

Operating Systems

- Android
- iOS
- Harmony
- Windows
- BlackBerry
- Palm OS
- Symbian
- …

Impact & Key Features

- Mobility
- Connectivity
- Versatility
- Digital Social life
- Video Calls and Meetings
- Health management
- Productivity
- Work on the Go
- Games, movies, …
- Education and Learning

Hardware

- Processors
  - ARM Processors: high performance & low power
- RAM
  - 2 to 12 GB
- Storage
  - Flash memory, SD cards
- Display: LCD, OLED
- Touchscreen: capacitive, resistive

Platforms and Languages

Flutter

Xamarin

React Native

iOS

Kotlin

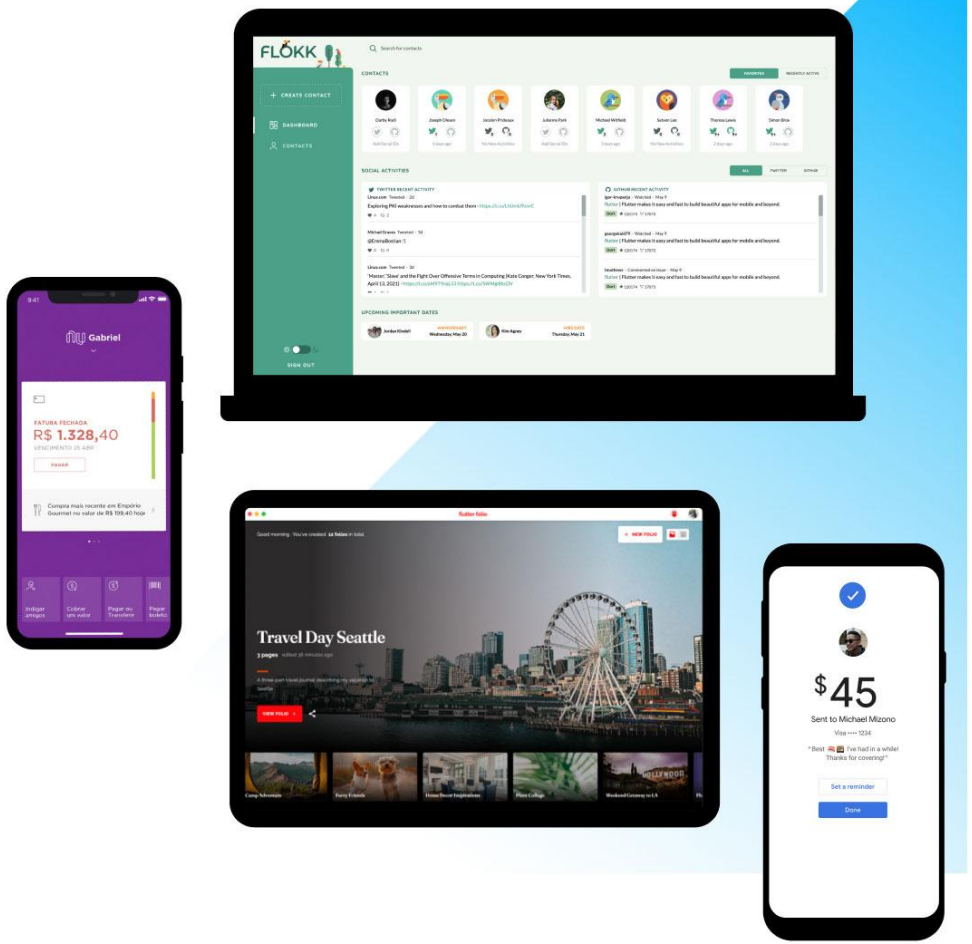
Dart

Flet

Flutter × python

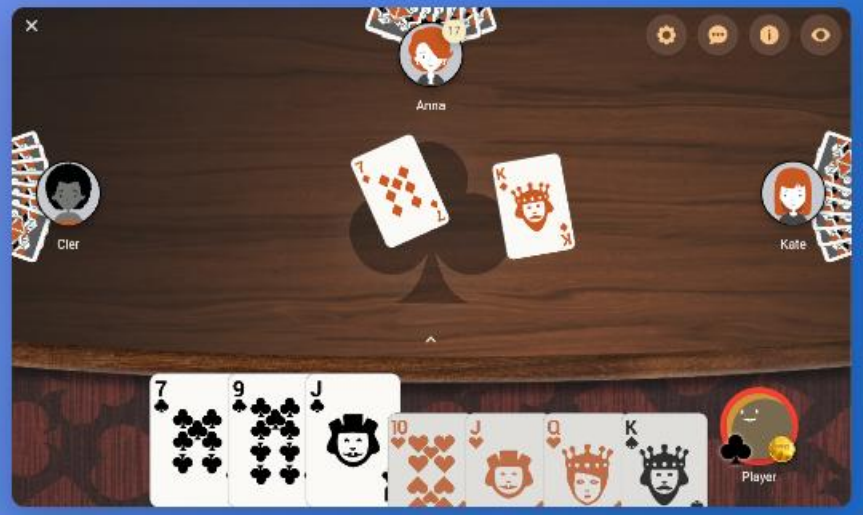
Flutter since 2017

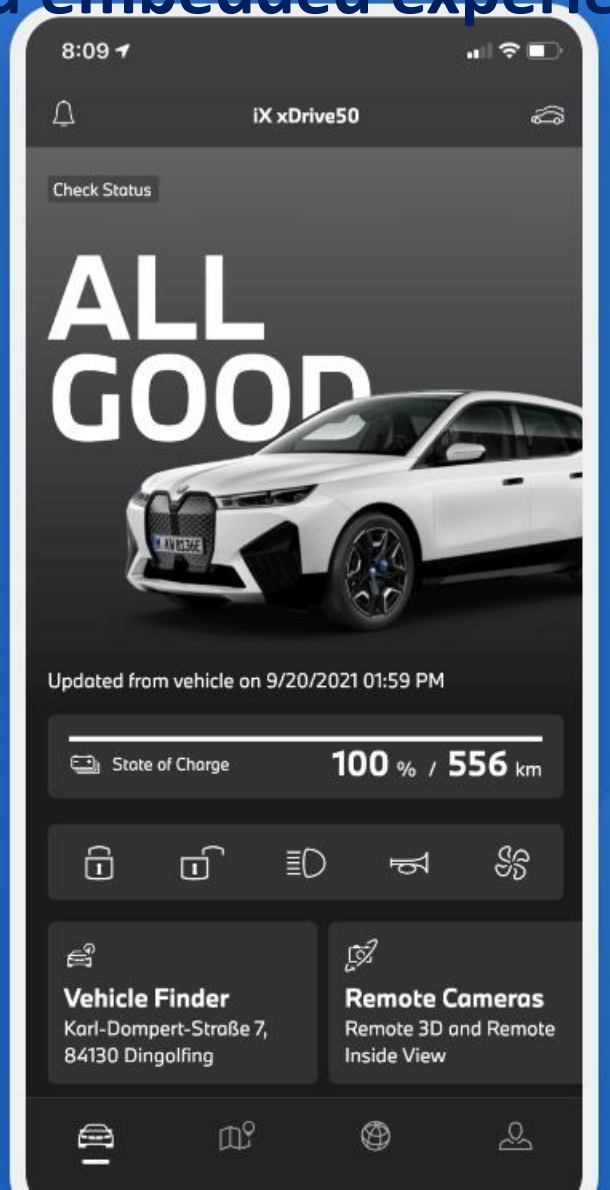
- Cross platform UI toolkit for mobile, web & desktop
- Single Codebase - Dart
- High performance
- Beautiful Uis
- Fast development

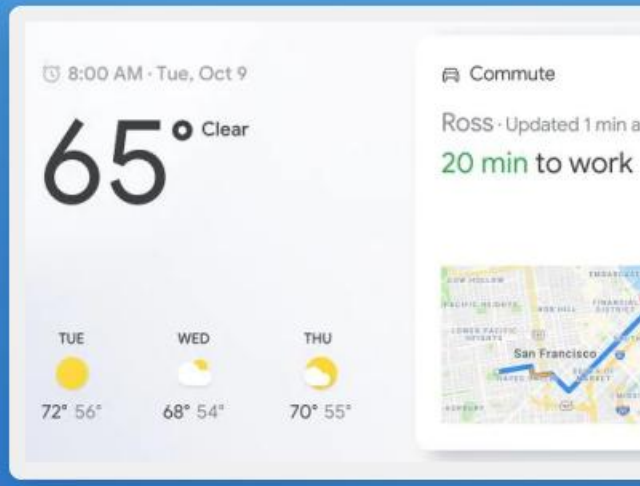


Build for any screen

Flutter transforms the development process. Build, test, and deploy beautiful mobile, web, desktop, and embedded experiences from a single codebase.







Fast

Flutter code compiles to ARM or Intel machine code as well as JavaScript, for fast performance on any device.

Try it in DartPad

Productive

Build and iterate quickly with Hot Reload. Update code and see changes almost instantly, without losing state.

Try it in DartPad

Flexible

Control every pixel to create customized, adaptive designs that look and feel great on any screen.

Try it in DartPad

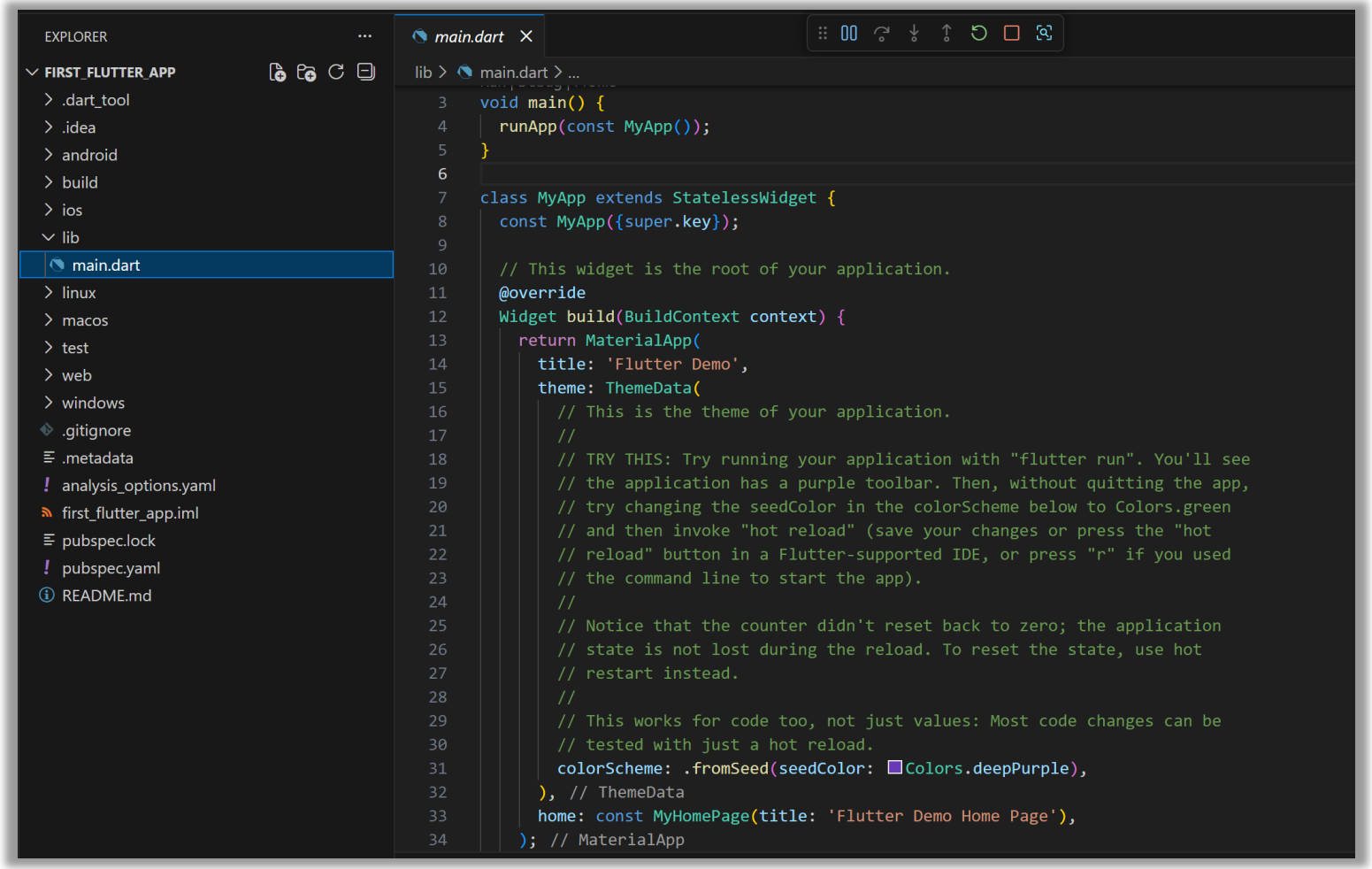
Setup Development Environment

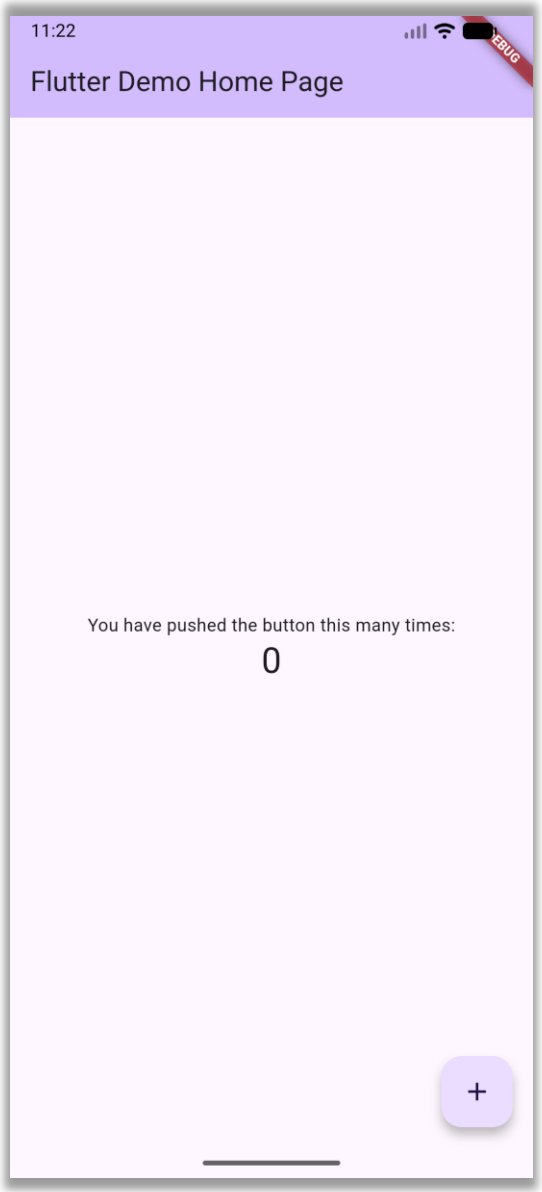
1. Download and install GIT - Download
2. Download and install VS Code - download
3. Download and install Android Studio - download
4. Download Flutter - Download
    1. Unzip the files to %USERPROFILE%\development
    2. Add %USERPROFILE%\development\flutter\bin to the Path (Environment Variable)

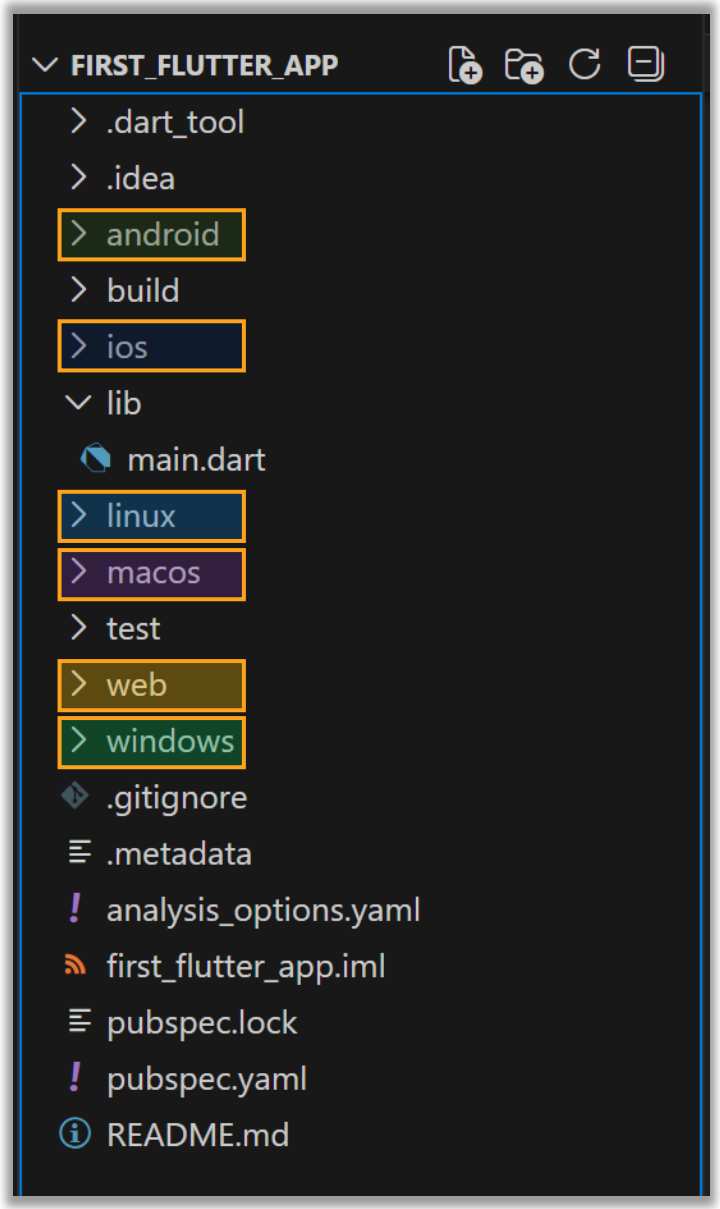
Setup and Test Flutter

- Open Command Prompt and run following commands
  - $ flutter --version
  - $ dart --version
- Check Flutter development environment
  - $ flutter doctor

First App


EXPLORER
FIRST_FLUTTER_APP
.dart_tool
.idea
android
build
ios
lib
main.dart
linux
macos
test
web
windows
.gitignore
.metadata
analysis_options.yaml
first_flutter_app.iml
pubspec.lock
pubspec.yaml
README.md
main.dart
lib > main.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: .fromSeed(seedColor: Colors.deepPurple),
), // ThemeData
home: const MyHomePage(title: 'Flutter Demo Home Page'),
); // MaterialApp


11:22
DEBUG
Flutter Demo Home Page
You have pushed the button this many times:
0
+



Project Folders Explained

android – Folder for **Android** App's specific files.

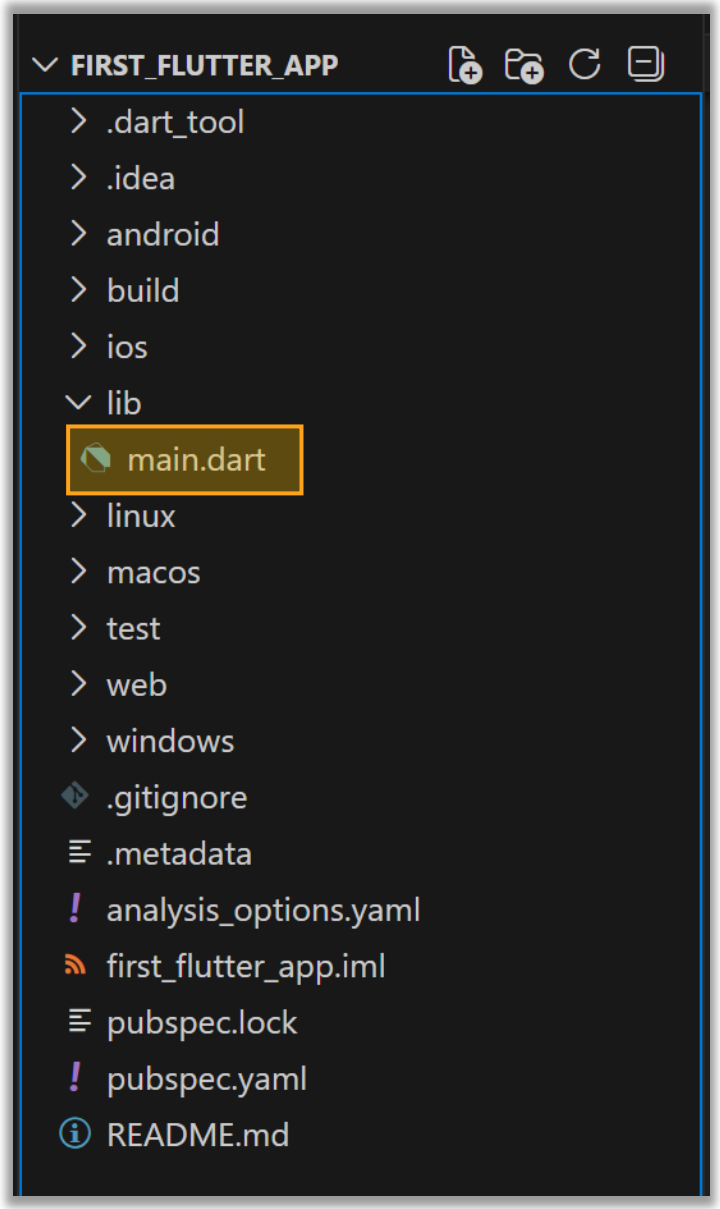
iso – Folder for **iOS** App's specific files.

linux – Folder for **Linux** App's specific files.

macos – Folder for **MACOS** App's specific files.

web – Folder for **Web** App's specific files.

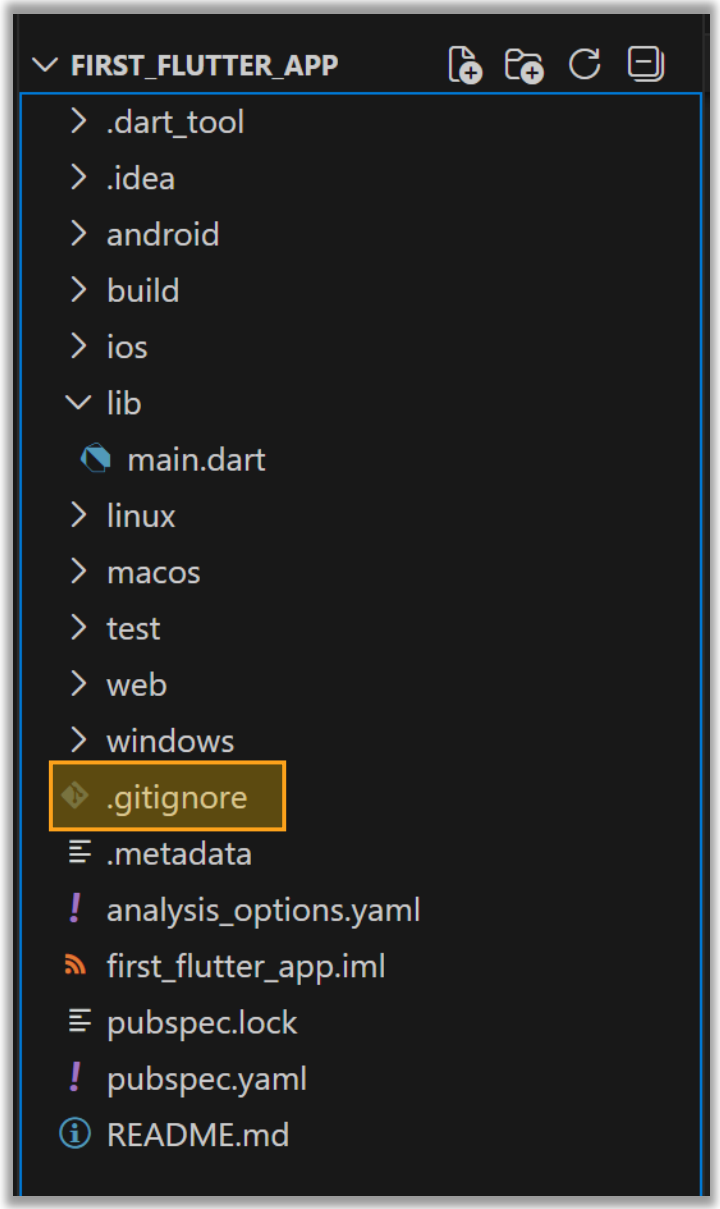
windows – Folder for **Windows** App's specific files.



Files Explained

main.dart

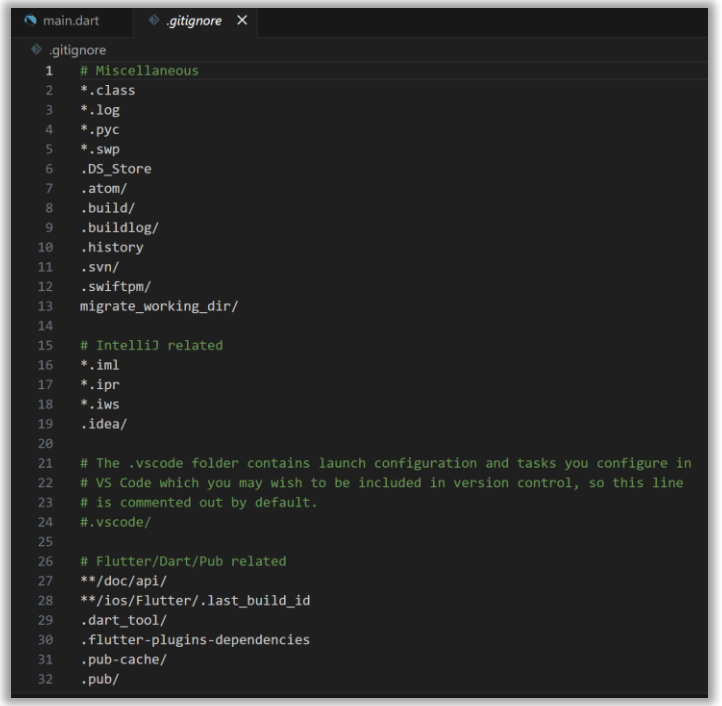
- Main file containing **main()** method
- **Entry point** of the App
- Crucial for initialization of Flutter environment and launching UI



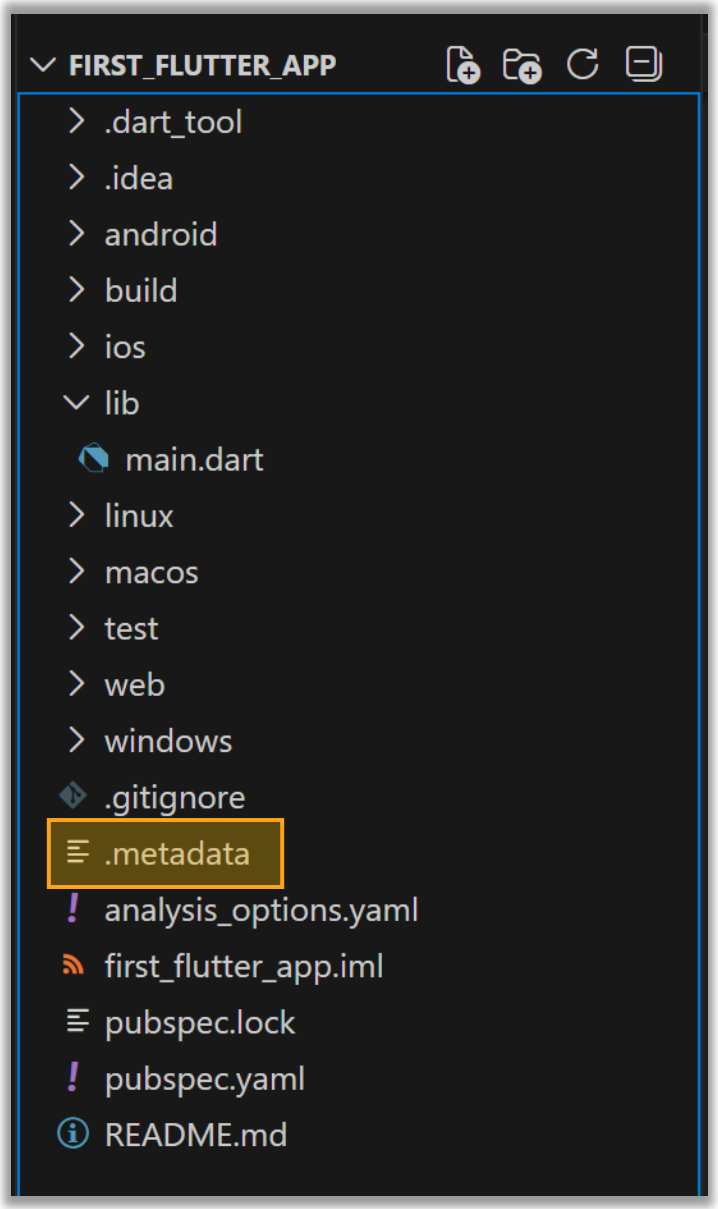
Files Explained

.gitignore

- Used to tell GIT to ignore the files

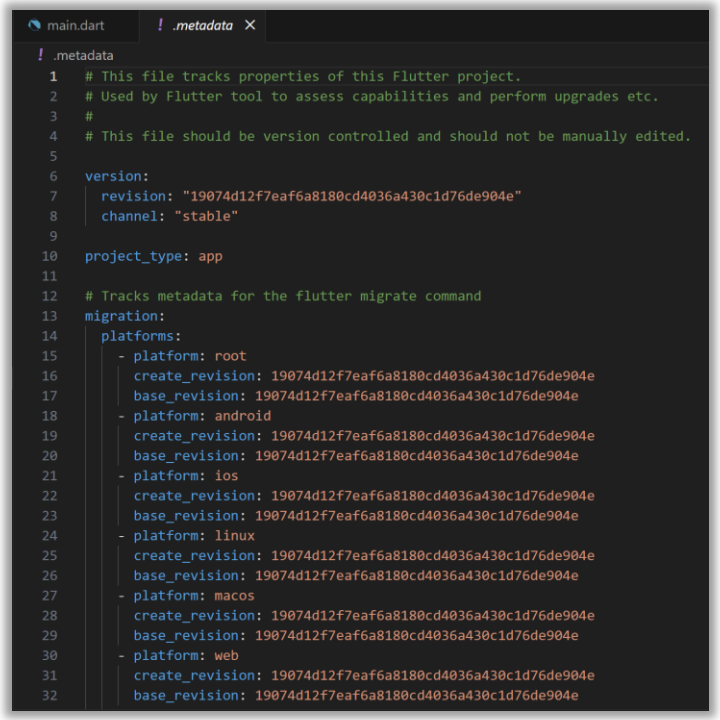


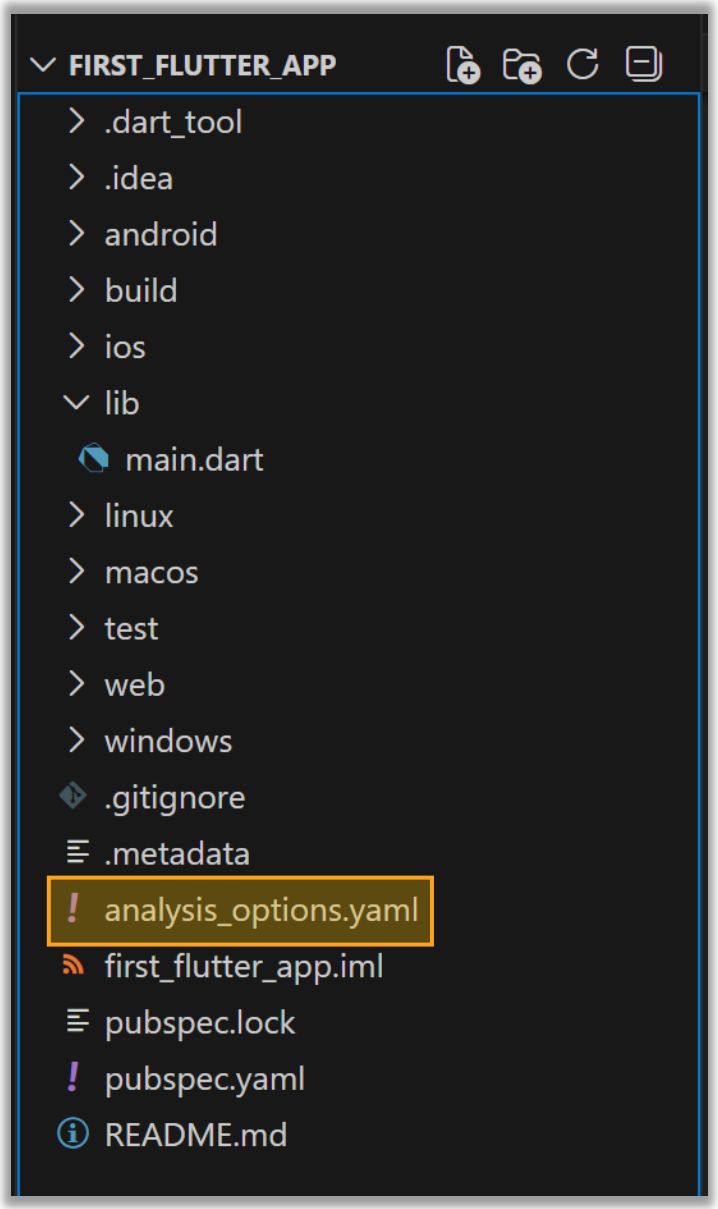
Files Explained



.metadata

- Properties of project

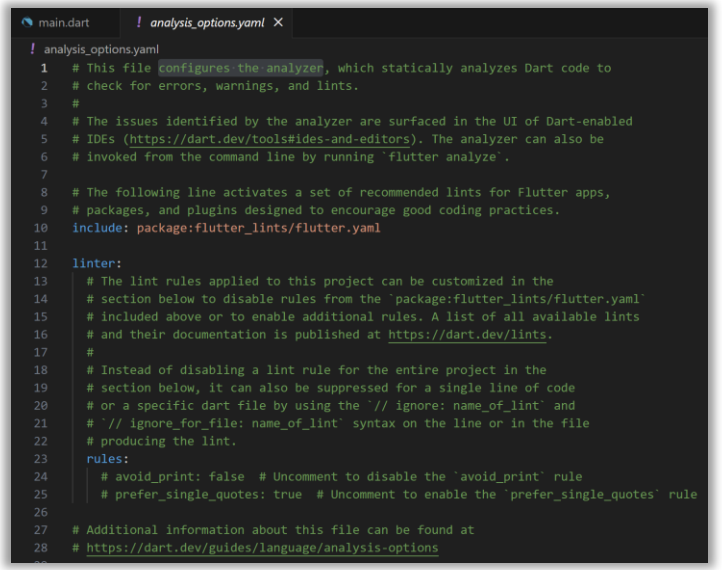


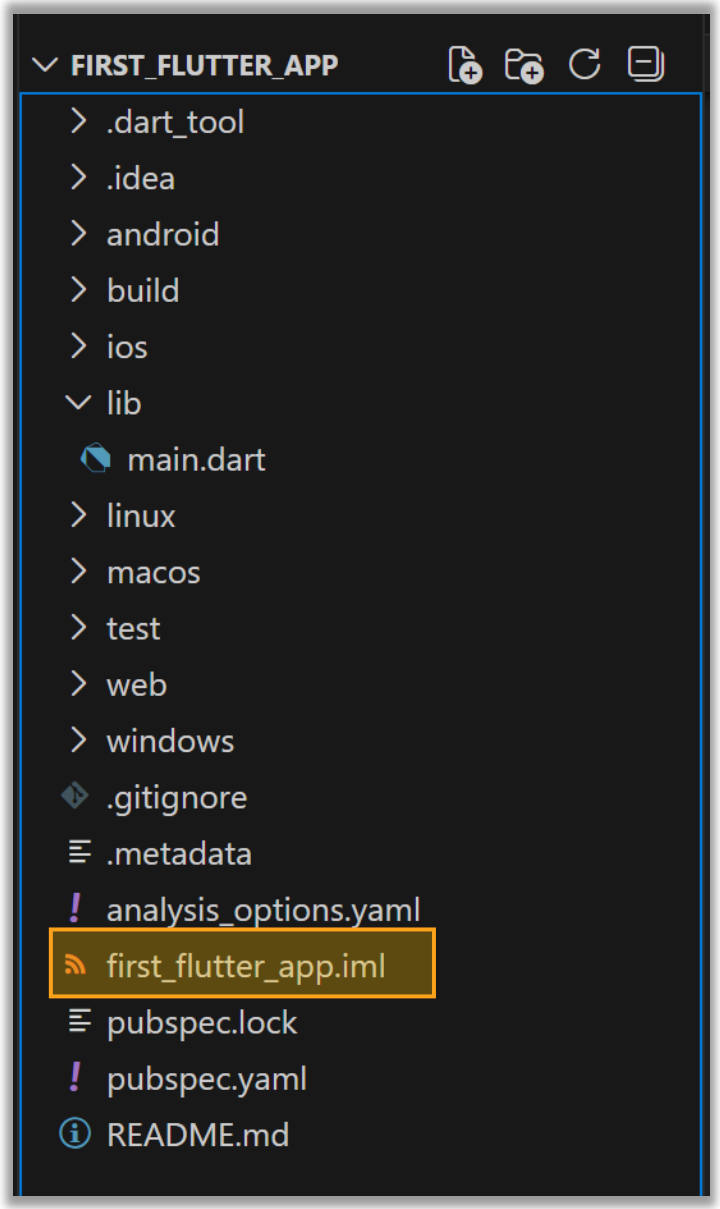


Files Explained

Analysis_options.yaml

- configures the code analyzer

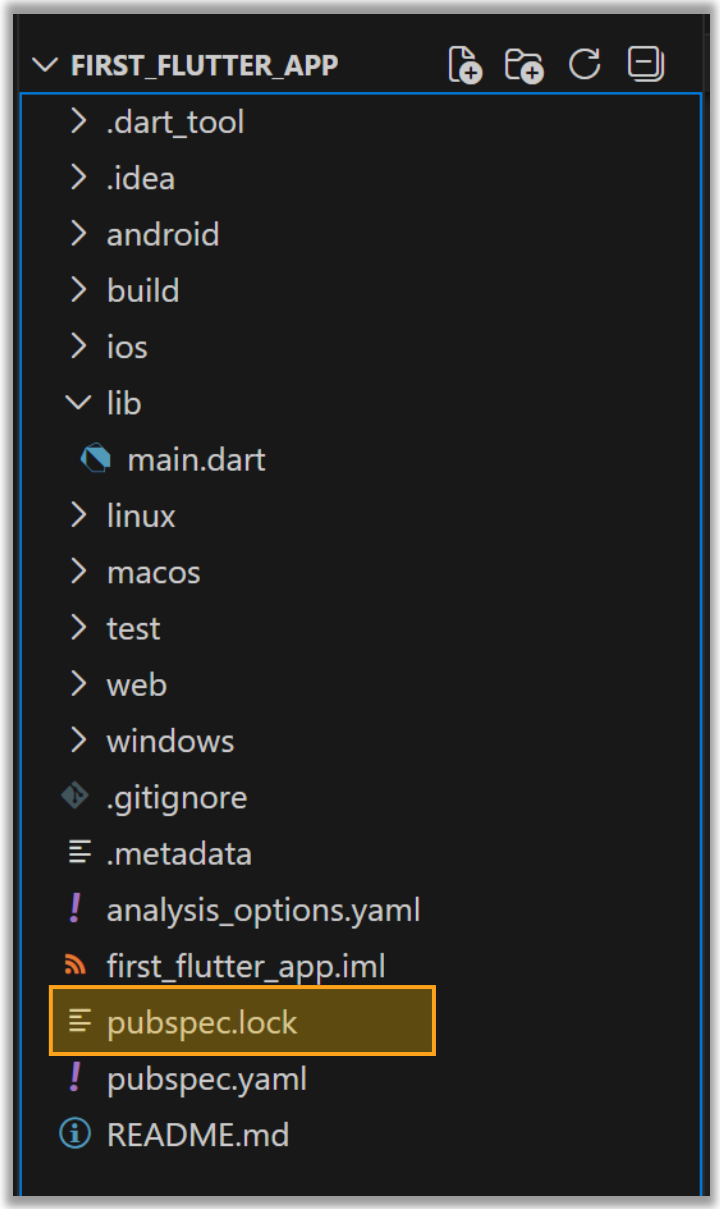




Files Explained

First_flutter_app.imp

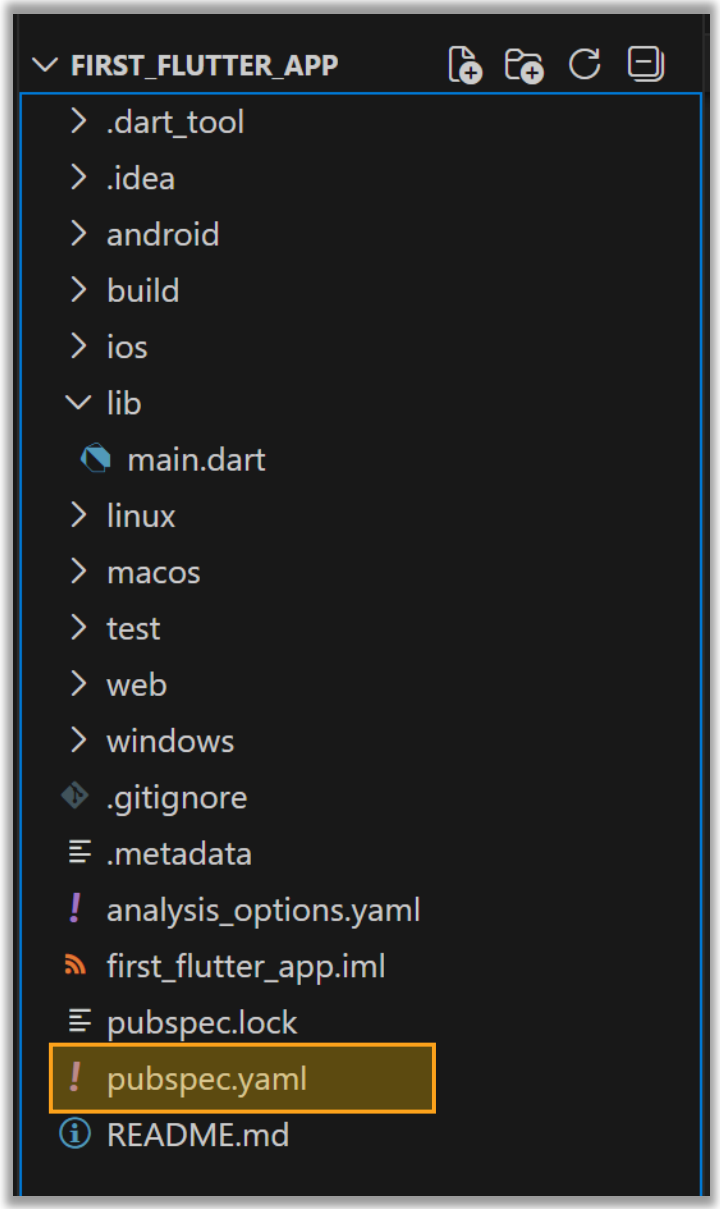
- Stores information about the project's structure, modules, paths, and dependencies for the IDE



Files Explained

pubspec.lock

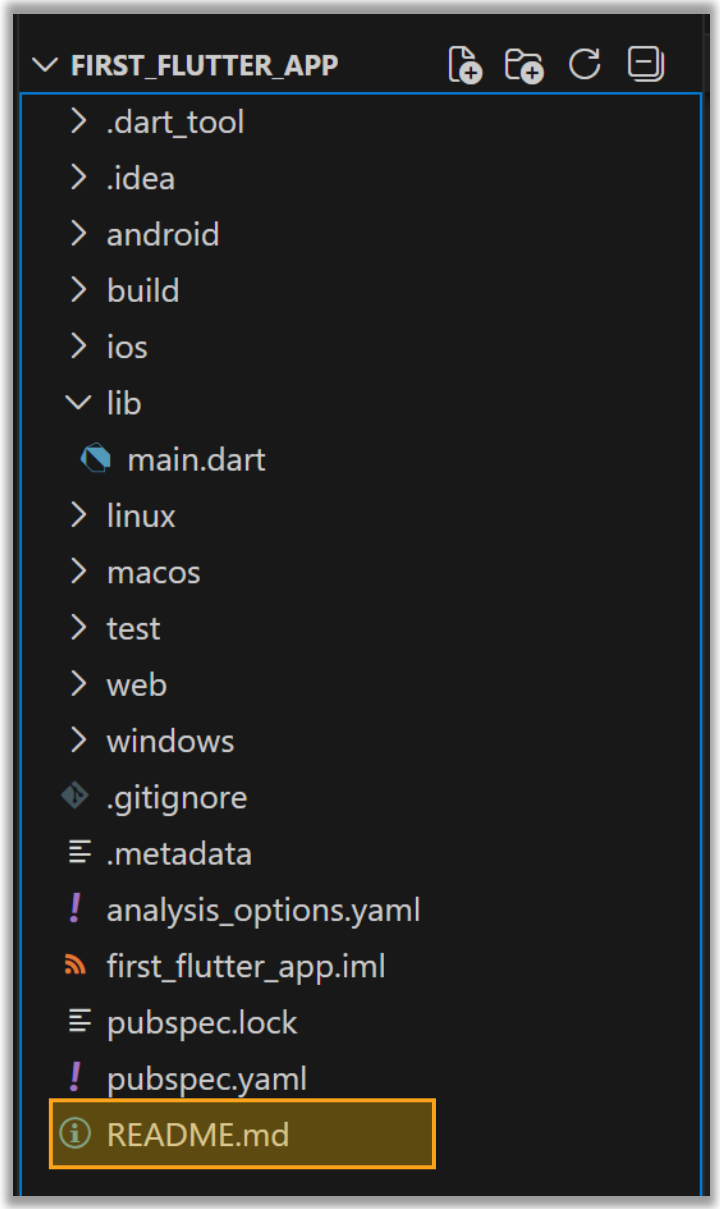
- Stores project dependencies with specific version used in App
- Auto generated by **pubsepc.yaml**



Files Explained

pubspec.yaml

- Manifest of the project
- Essential Metadata
  - Name, description
  - Dependencies and versions
  - Project Settings
  - Resources declaration



Files Explained

README.md

- Read me file that explain the project

main.dart

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

```dart
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: .fromSeed(seedColor: Colors.deepPurple),
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}
```

```
home: const MyHomePage(title: 'Flutter Demo Home Page')
```

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

  final String title;

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

```dart
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;
  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }
```

```dart
@override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: .center,
          children: [
            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),
      ),
    );
  }
}
```

References

- https://flutter.dev/