

CCIS

كلية علوم الحاسوب والمعلومات
COLLEGE OF COMPUTER &
INFORMATION SCIENCES



جامعة الأمير سلطان
PRINCE SULTAN
UNIVERSITY

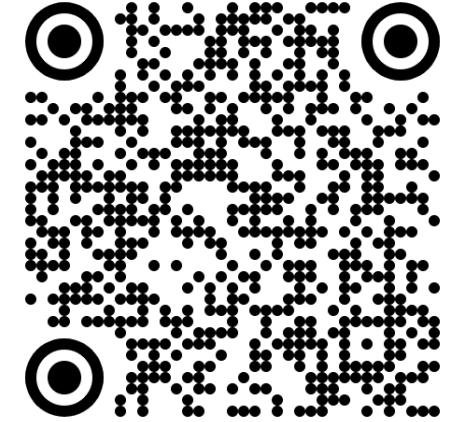


IS487 Emerging Topics in IS

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**



Data Analytics

Machine Learning



NLP/MT/ASP
Data Mining



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

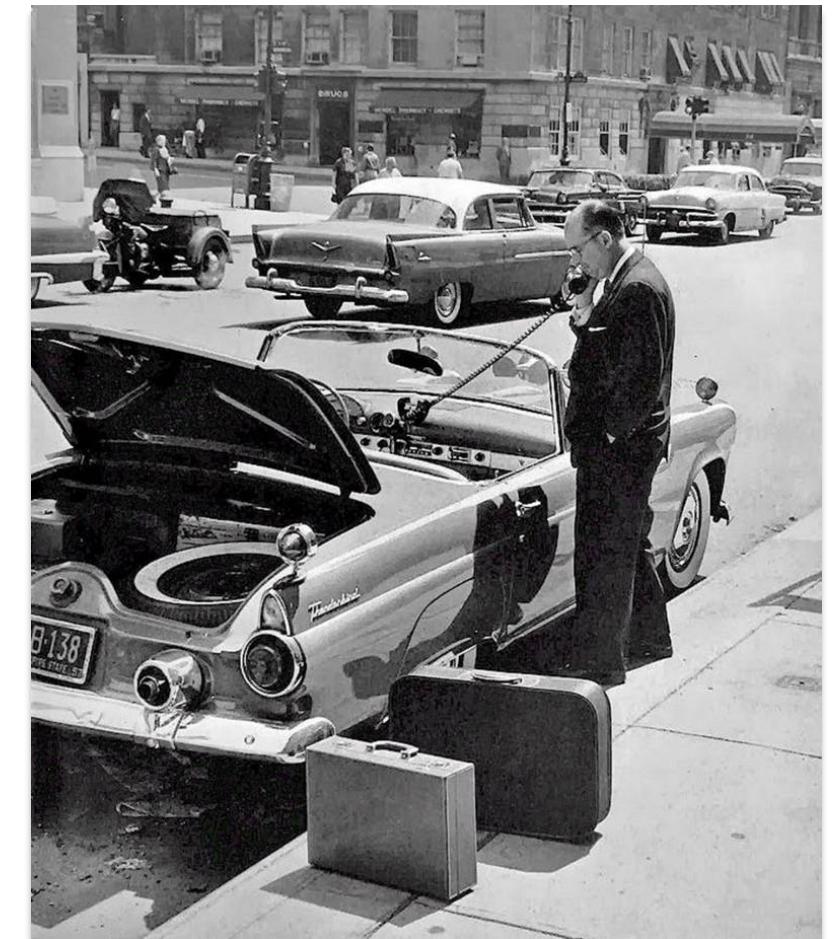
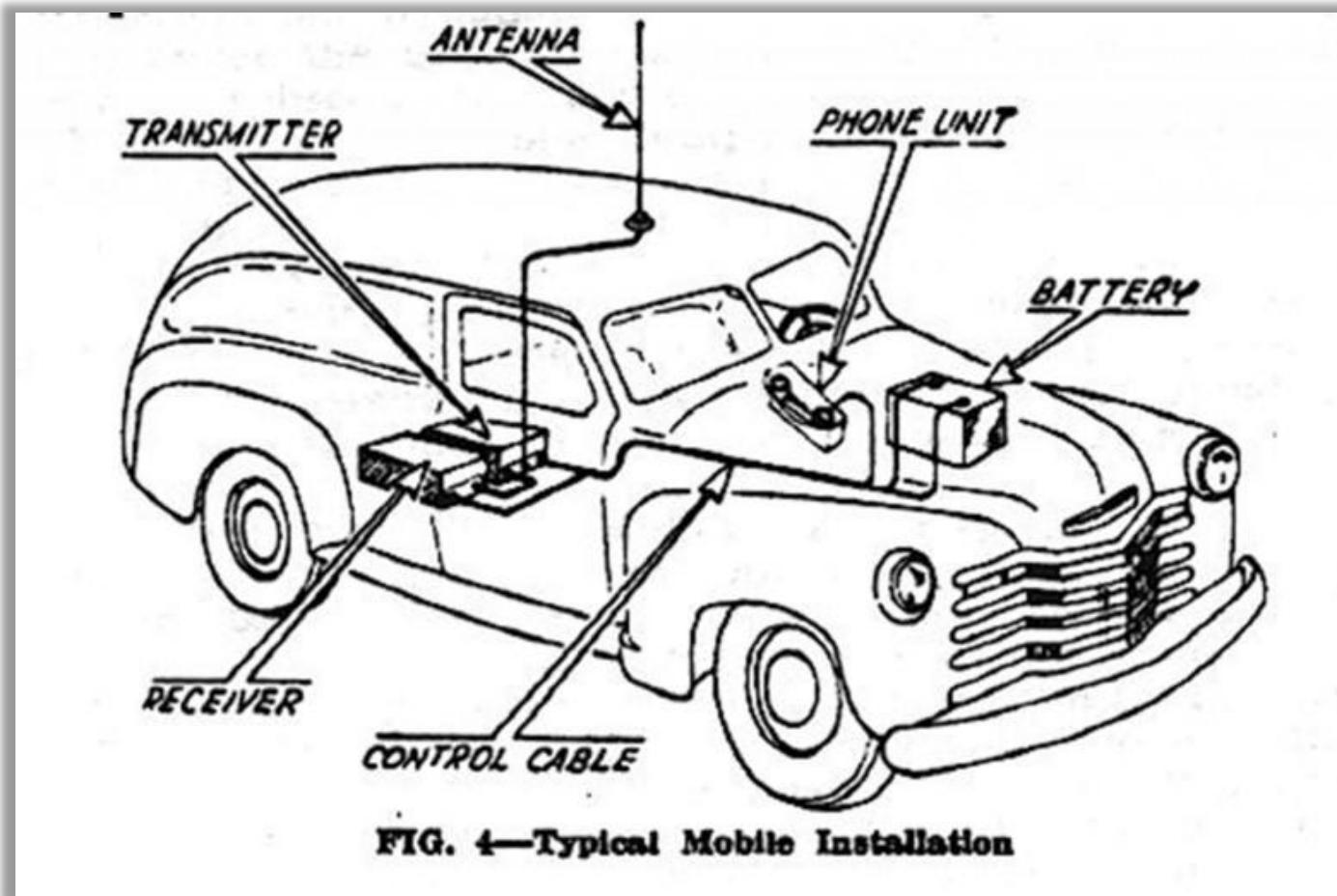
What is Mobile App Development?

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

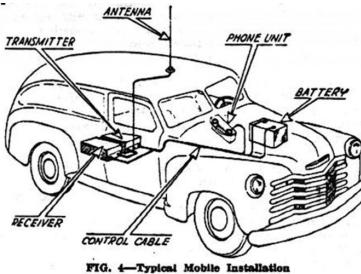
Portable Devices: Compact lightweight computing devices that can be easily carried around

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

Mobile Devices History



Portable Device Evolution



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

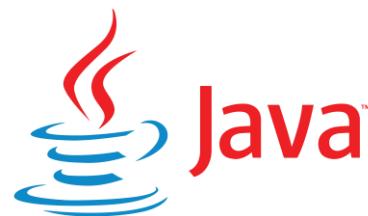
Platforms and Languages



Flutter

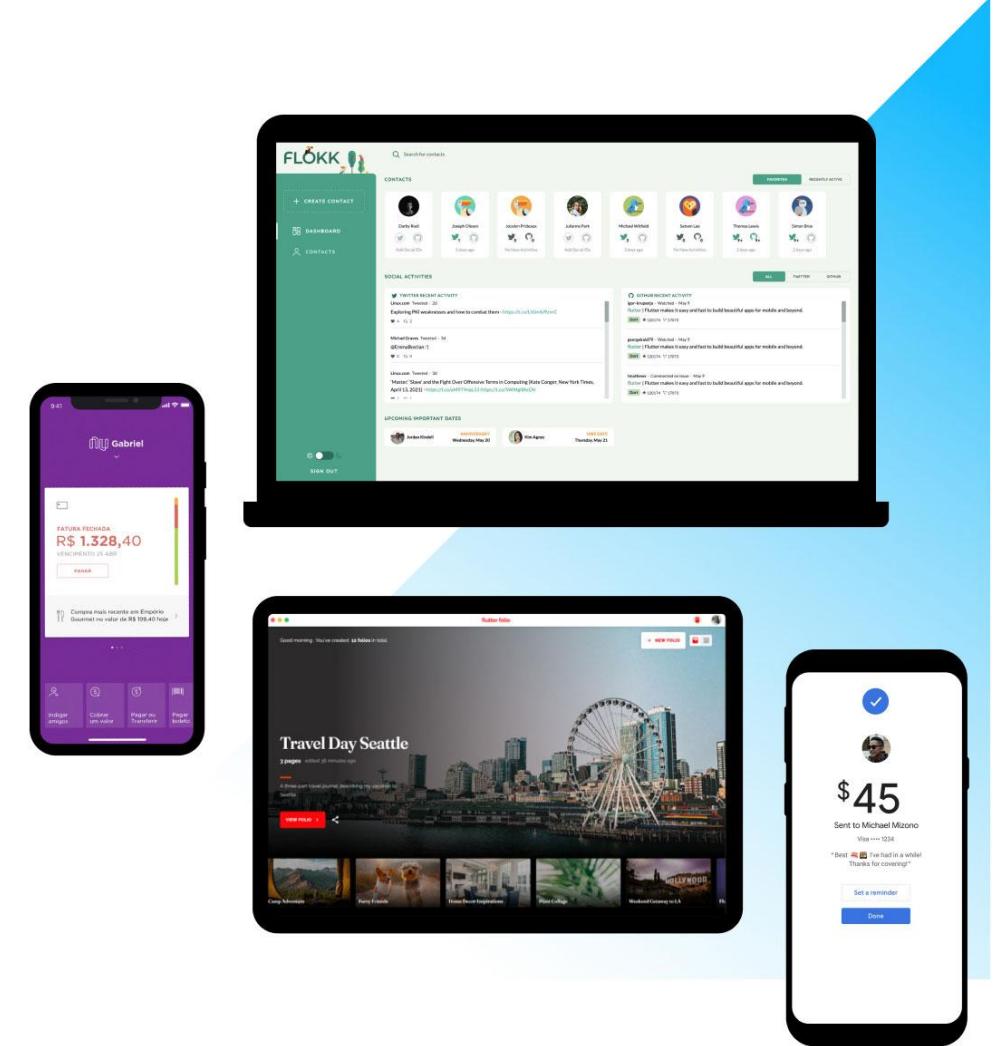


Dart



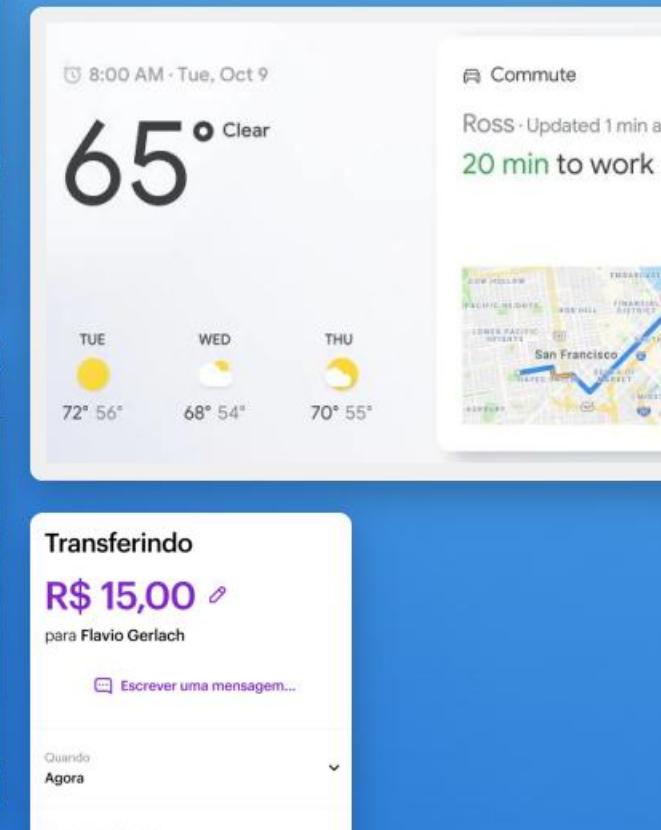
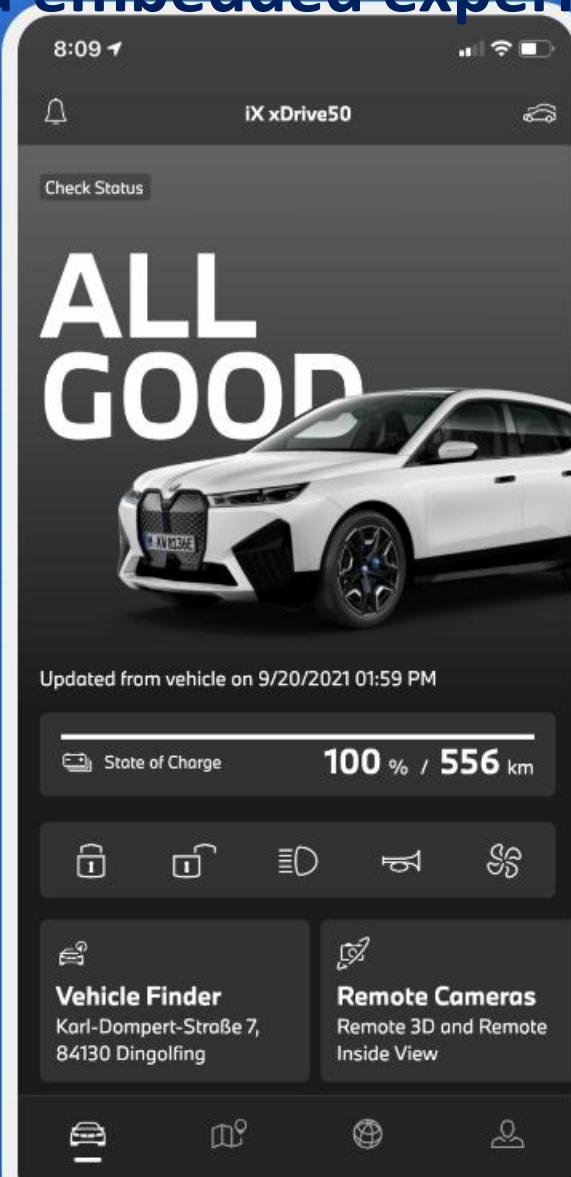
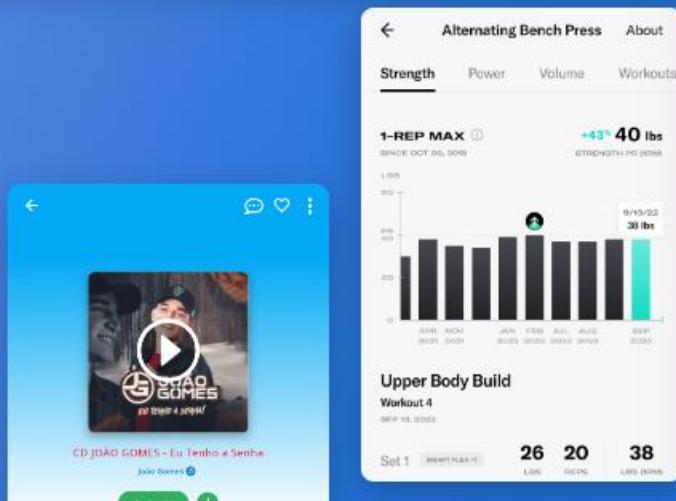
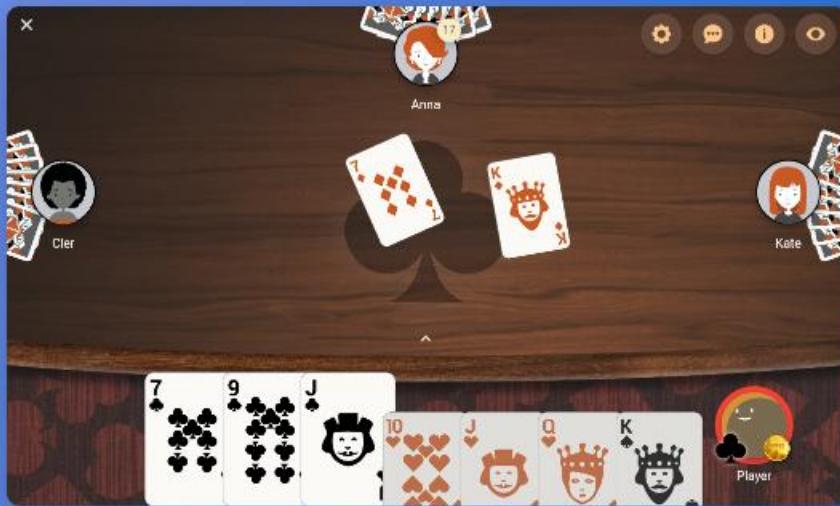
Flutter since 2017

- 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.



```
Widget build(BuildContext  
context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: const  
Text('Composition FTW!'),  
    ),  
    body: ...  
  );  
}
```

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.

```
class MyApp extends StatelessWidget
{
  const MyApp({Key? key}) :
super(key: key);
  @override
  Widget build(BuildContext
context) {
  return MaterialApp(
    theme: ThemeData.light(),
    home: const CounterPage(),
  );
}
```

[Try it in DartPad](#)



```
Scaffold(  
  appBar: AppBar(  
    title: const Text('Automatic adaptivity!'),  
  ),  
  body: ...  
)
```

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)

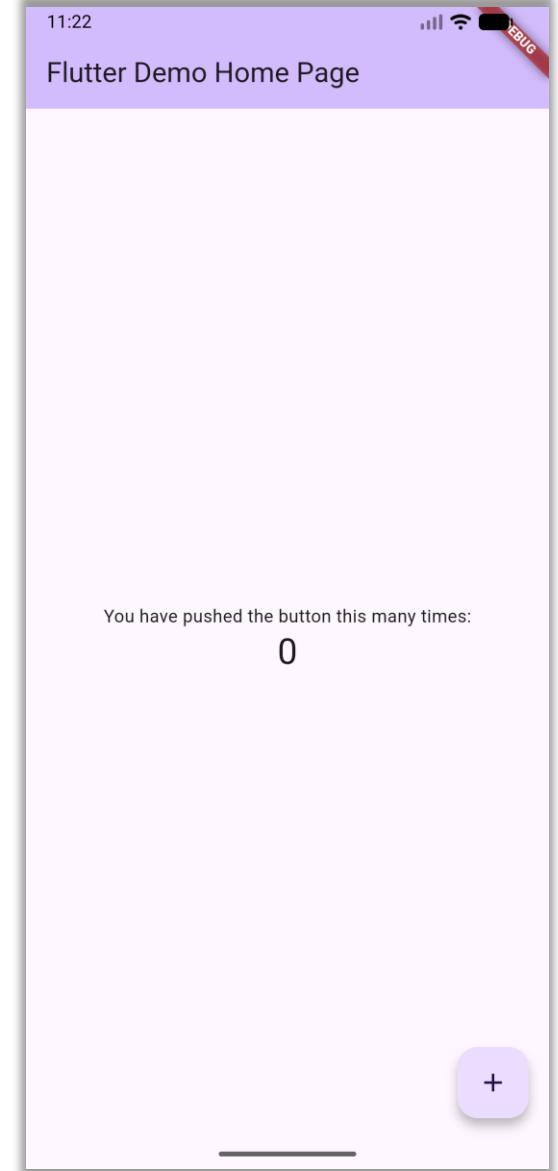
Setup and Test Flutter

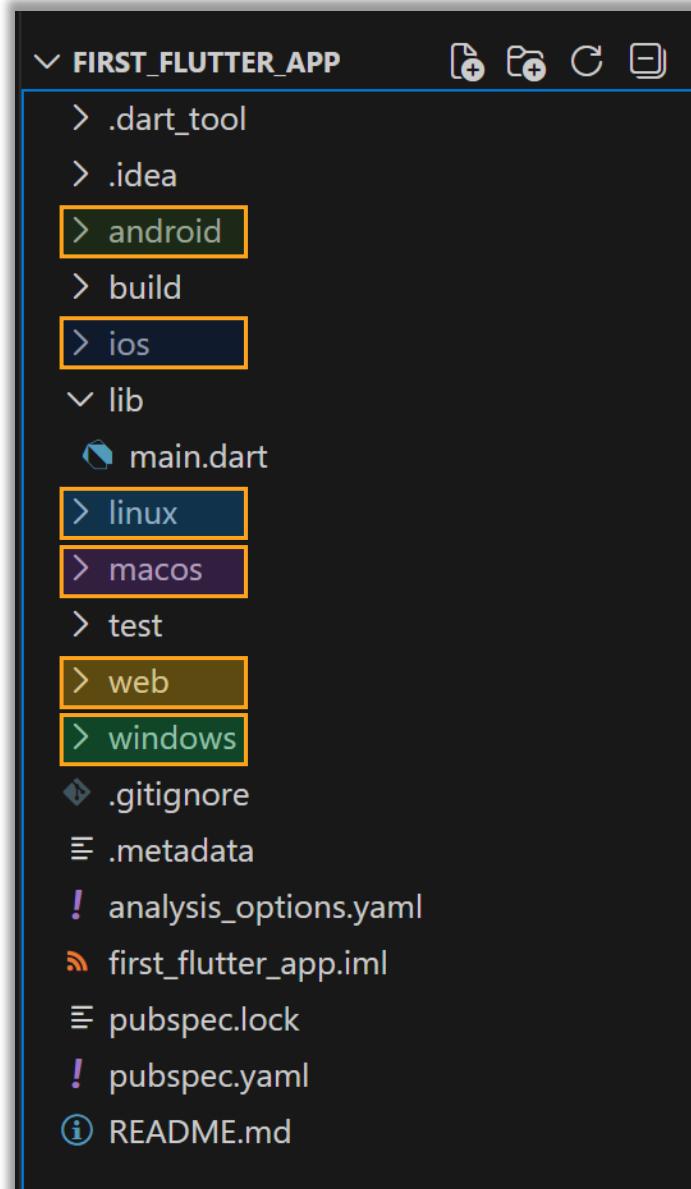
- Open Command Prompt and run following commands
 - \$ flutter --version
 - \$ dart --version
- Check Flutter development environment
 - \$ flutter doctor

First App

The screenshot shows a code editor with the main.dart file open. The code defines a simple Flutter application with a purple theme and a counter.

```
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
```





Project Folders Explained

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

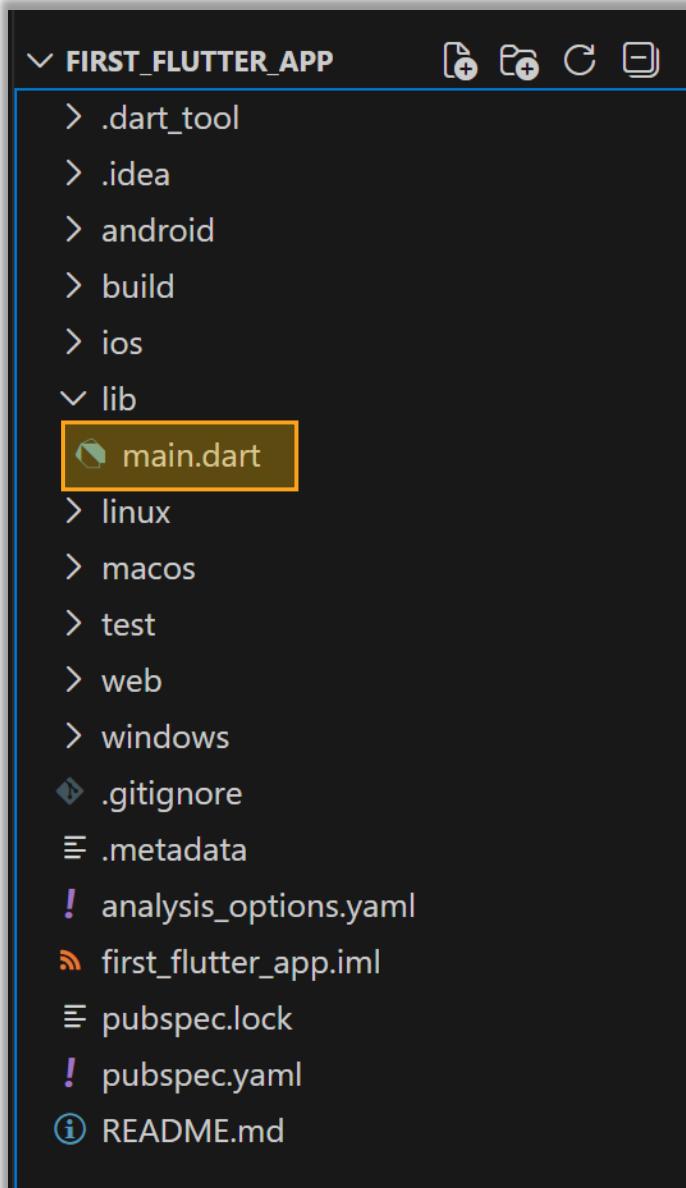
iso – Folder for **iOS** App's specific files.

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

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

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

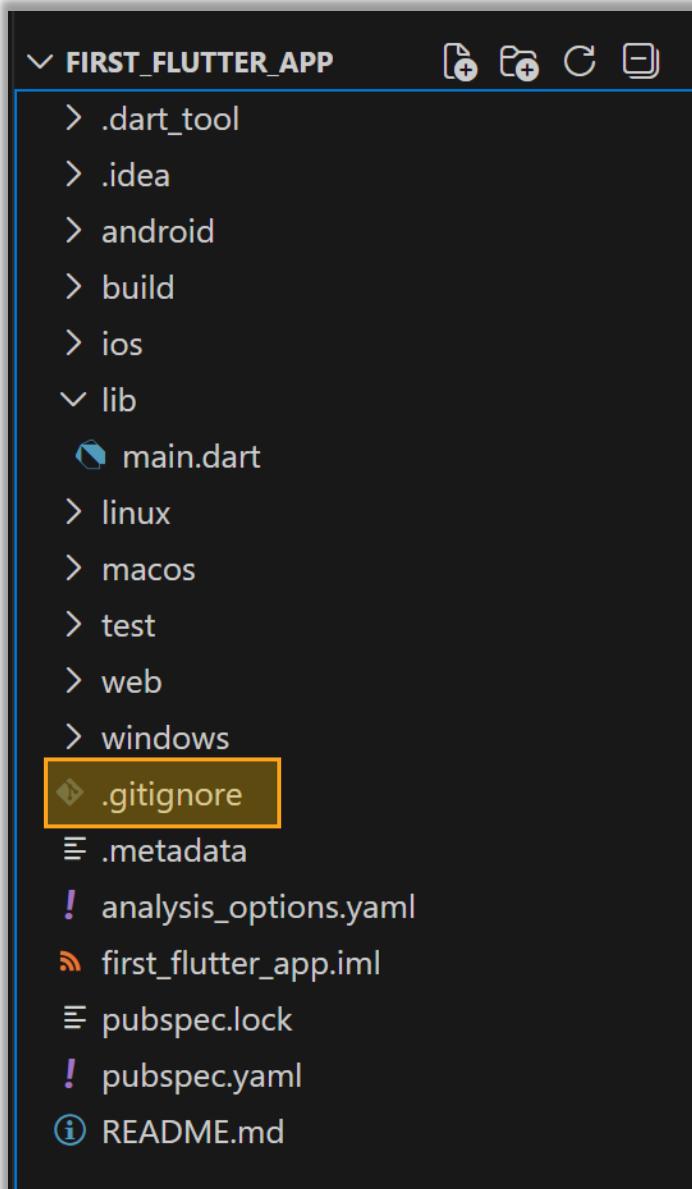
windows – Folder for **Windows** App's specific files.



Files Explained

main.dart

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

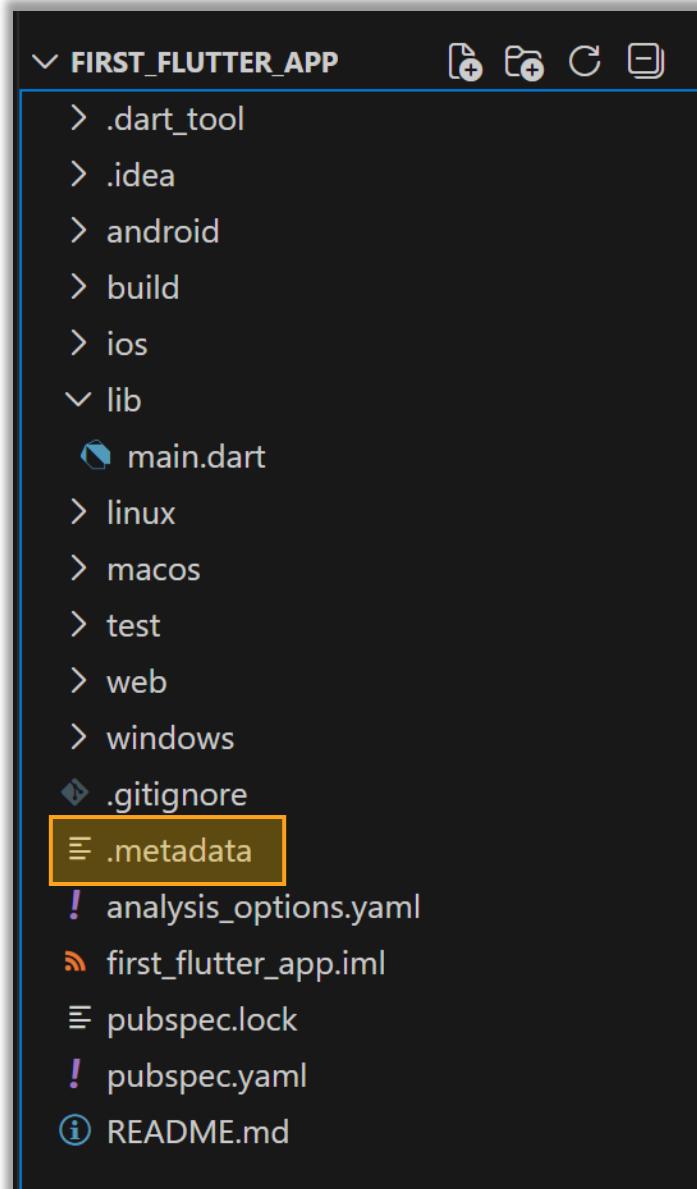


Files Explained

.gitignore

- Used to tell GIT to ignore the files

```
# Miscellaneous
*.class
*.log
*.pyc
*.swp
.DS_Store
.atom/
.build/
.buildlog/
.history
.svn/
.swiftpm/
migrate_working_dir/
# IntelliJ related
*.iml
*.ipr
*.iws
.idea/
# The .vscode folder contains launch configuration and tasks you configure in
# VS Code which you may wish to be included in version control, so this line
# is commented out by default.
#.vscode/
# Flutter/Dart/Pub related
**/doc/api/
**/ios/Flutter/.last_build_id
.dart_tool/
.flutter-plugins-dependencies
.pub-cache/
.pub/
```



Files Explained

.metadata

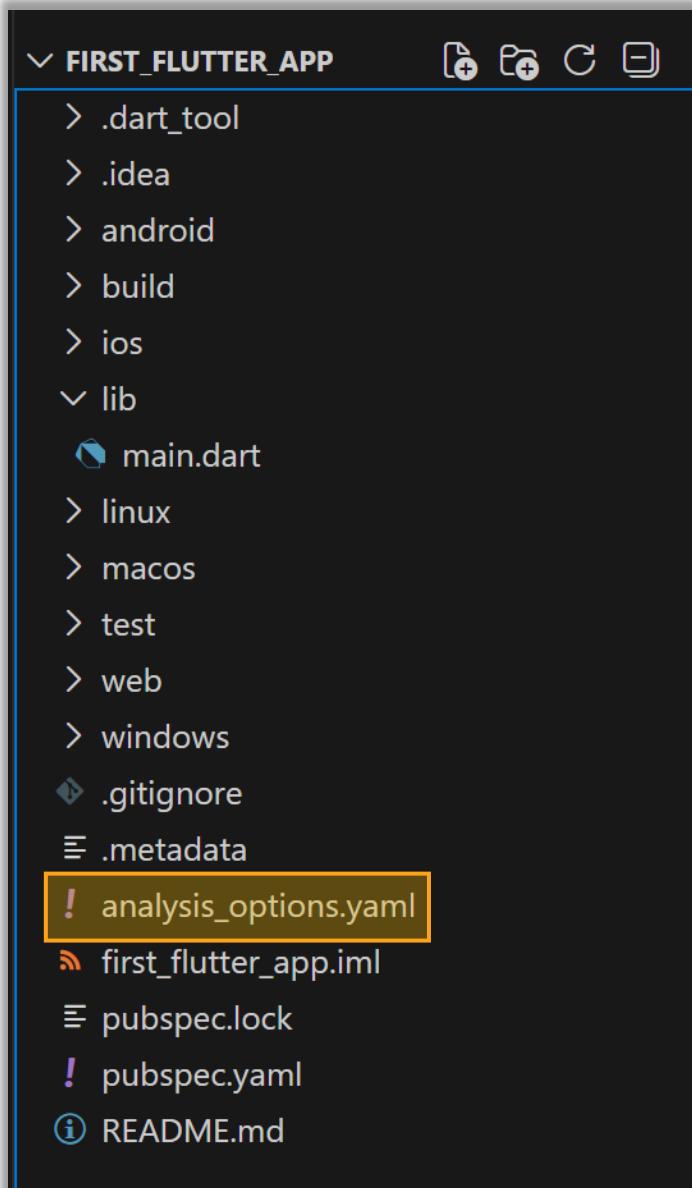
- Properties of project

```
# This file tracks properties of this Flutter project.
# Used by Flutter tool to assess capabilities and perform upgrades etc.
#
# This file should be version controlled and should not be manually edited.

version:
  revision: "19074d12f7eaf6a8180cd4036a430c1d76de904e"
  channel: "stable"

project_type: app

# Tracks metadata for the flutter migrate command
migration:
  platforms:
    - platform: root
      create_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
      base_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
    - platform: android
      create_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
      base_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
    - platform: ios
      create_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
      base_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
    - platform: linux
      create_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
      base_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
    - platform: macos
      create_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
      base_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
    - platform: web
      create_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
      base_revision: 19074d12f7eaf6a8180cd4036a430c1d76de904e
```

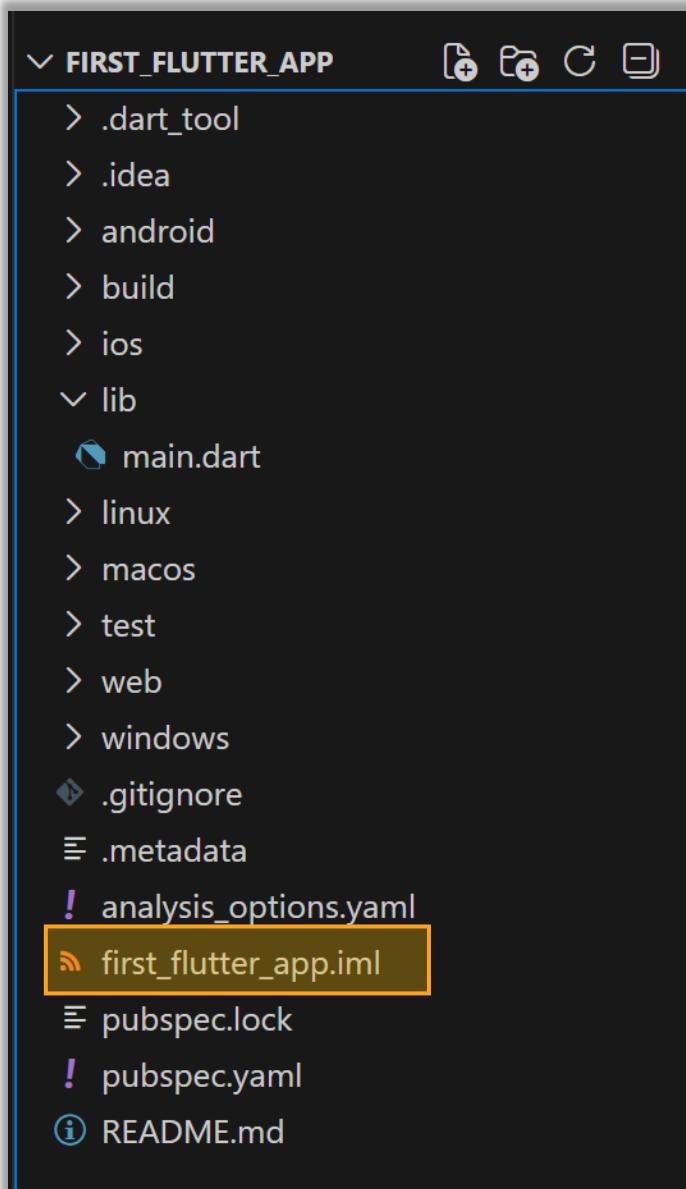


Files Explained

Analysis_options.yaml

- configures the code analyzer

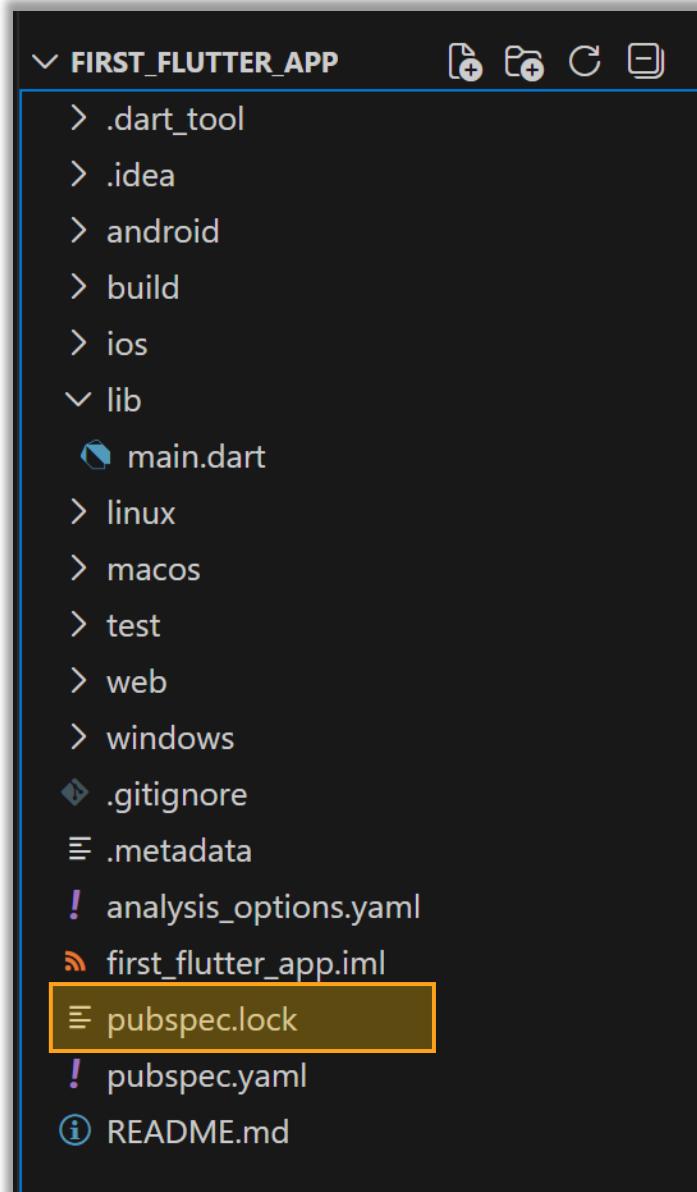
```
! analysis_options.yaml
1 # This file configures the analyzer, which statically analyzes Dart code to
2 # check for errors, warnings, and lints.
3 #
4 # The issues identified by the analyzer are surfaced in the UI of Dart-enabled
5 # IDEs (https://dart.dev/tools#ides-and-editors). The analyzer can also be
6 # invoked from the command line by running 'flutter analyze'.
7 #
8 # The following line activates a set of recommended lints for Flutter apps,
9 # packages, and plugins designed to encourage good coding practices.
10 include: package:flutter_lints/flutter.yaml
11
12 linter:
13   # The lint rules applied to this project can be customized in the
14   # section below to disable rules from the package:flutter_lints/flutter.yaml
15   # included above or to enable additional rules. A list of all available lints
16   # and their documentation is published at https://dart.dev/lints.
17   #
18   # Instead of disabling a lint rule for the entire project in the
19   # section below, it can also be suppressed for a single line of code
20   # or a specific dart file by using the '// ignore: name_of_lint' and
21   # '// ignore_for_file: name_of_lint' syntax on the line or in the file
22   # producing the lint.
23   rules:
24     # avoid_print: false # Uncomment to disable the 'avoid_print' rule
25     # prefer_single_quotes: true # Uncomment to enable the 'prefer_single_quotes' rule
26
27   # Additional information about this file can be found at
28   # https://dart.dev/guides/language/analysis-options
```



Files Explained

First_flutter_app.iml

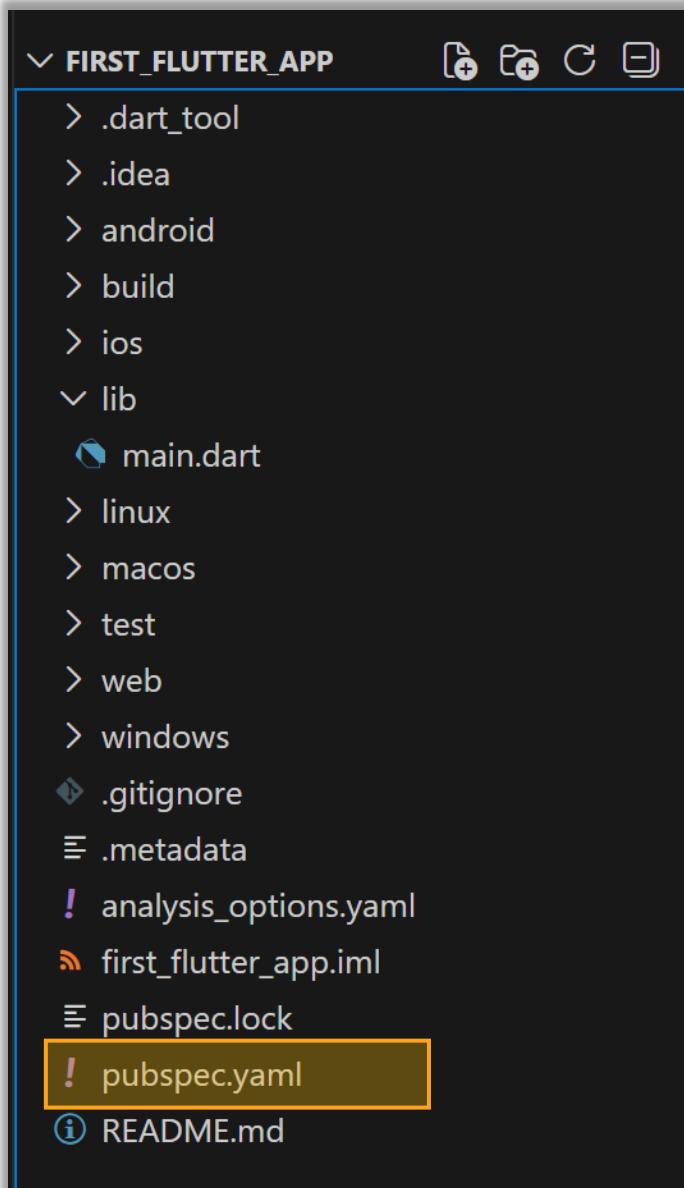
- Stores information about the project's structure, modules, paths, and dependencies for the IDE



Files Explained

pubspec.lock

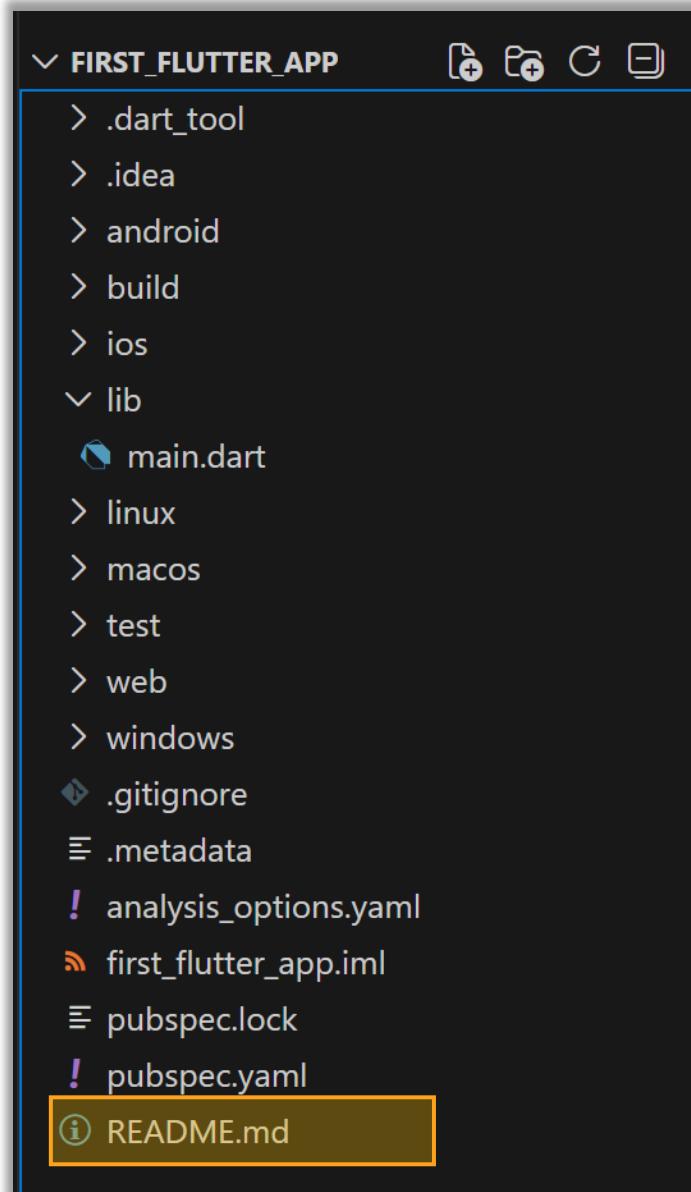
- Stores project dependencies with specific version used in App
- Auto generated by **pubspec.yaml**



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

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

main.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();  
}
```

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

@override

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      backgroundColor: Theme.of(context).colorScheme.inversePrimary,  
      title: Text(widget.title),  
    ),  
    body: Center(  
      child: Column(  
        mainAxisAlignment: 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/>