

Name: Abhishek Fatate

roll no.: 15

MAD PWA EXP 2

Aim: To design Flutter UI by including common widgets.

Theory:

We can split the Flutter widget into two categories:

1. Visible (Output and Input)
2. Invisible (Layout and Control)

1. Visible widget

The visible widgets are related to the user input and output data. Some of the important types of

this widget are:

1. Text

A Text widget holds some text to display on the screen. We can align the text widget by using

textAlign property, and style property allow the customization of Text that includes font, font

weight, font style, letter spacing, color, and many more.

2. Button

This widget allows you to perform some action on click. Flutter does not allow you to use the

Button widget directly; instead, it uses a type of buttons like a FlatButton and a RaisedButton.

3. Image

This widget holds the image which can fetch it from multiple sources like from the asset folder or directly from the URL. It provides many constructors for loading image, which

are given below:

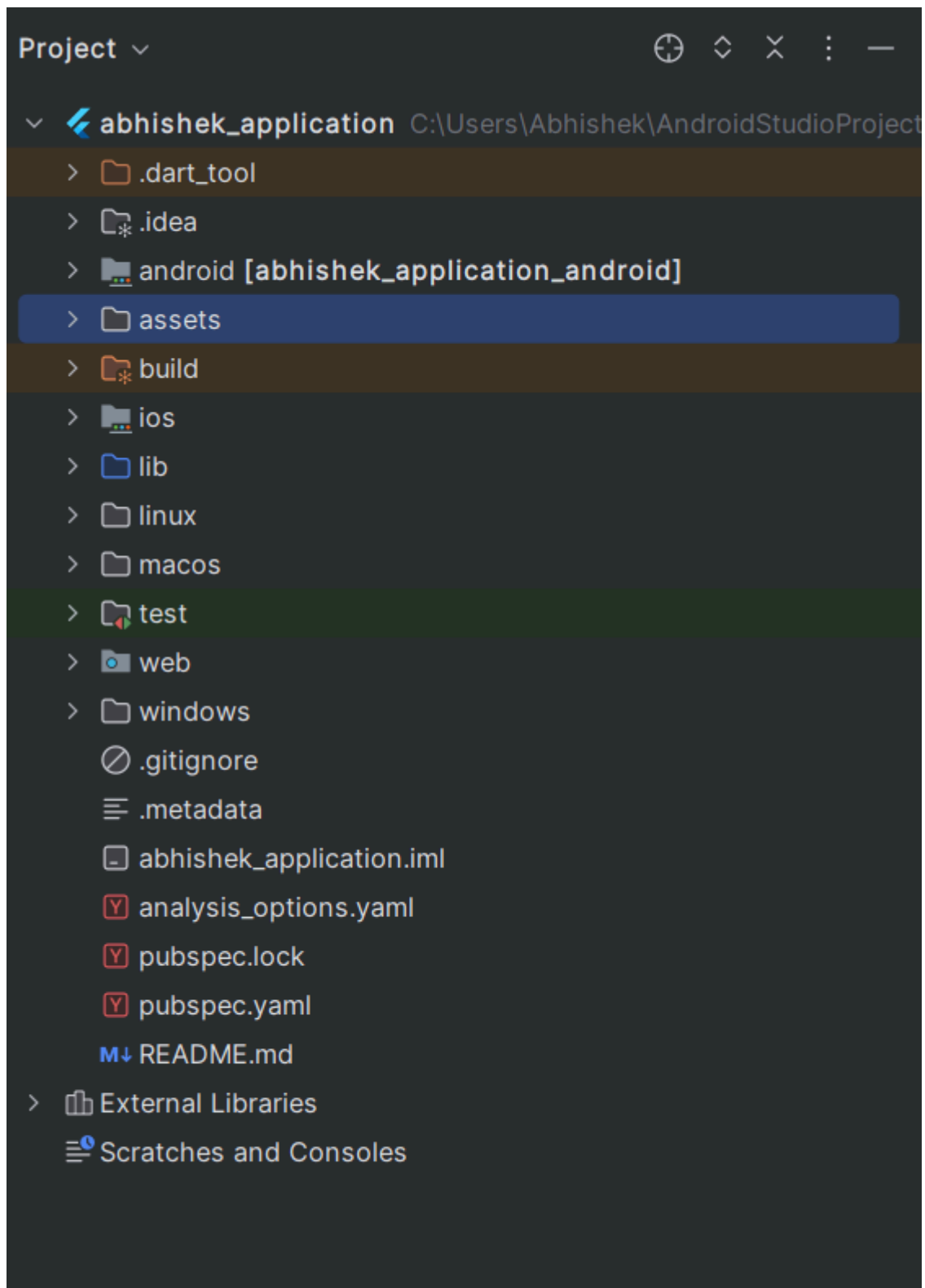
- o Image: It is a generic image loader, which is used by ImageProvider.
- o asset: It load image from your project asset folder.
- o file: It loads images from the system folder.
- o memory: It load image from memory.
- o network: It loads images from the network.

To add an image in the project, you need first to create an assets folder where you keep your images and then add the below line in pubspec.yaml file.

assets:

- assets/comp.jpg

File structure:



Code:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ), // AppBar
        body: Center(
          child: Image.asset('assets/0yo_logo.jpeg'),
        ), // Center
      ), // Scaffold
    ); // MaterialApp
  }
}
```

Output:



Conclusion:

Flutter's widget architecture offers great flexibility for building complex UIs. Understanding key widgets and concepts is essential for effective Flutter development.