Department of Information Technology

Academic Year: 2022-23
Semester: VI
Class / Branch / Div: TE- IT A/B
Subject: MAD & PWA Lab
Name of Instructor:

Name of Student:
Student ID:
Roll No.
Date of Submission:

Experiment No.:2

Aim: To design Flutter UI by including common widgets.

Theory:

Everything in Flutter is a widget and their different types available in the Flutter framework

Whenever you are going to code for building anything in Flutter, it will be inside a widget. The central purpose is to build the app out of widgets. It describes how your app view should look like with their current configuration and state.

Widget Build Visualization

In Flutter, widgets can be grouped into multiple categories based on their features, as listed below −

1. Platform specific widgets
2. Layout widgets
3. State maintenance widgets
4. Platform independent / basic widgets

A. Platform specific widgets

Flutter has widgets specific to a particular platform - Android or iOS.

Android specific widgets are designed in accordance with Material design guideline by Android OS. Android specific widgets are called as Material widgets.

Some of the most used material widgets are as follows −

Scaffold, AppBar, BottomNavigationBar, TabBar, TabBarView, ListTile, RaisedButton, FloatingActionButton, FlatButton, IconButton, DropdownButton, PopupMenuButton ButtonBar, TextField, Checkbox, Radio, Switch, Slider, Date & Time Pickers, SimpleDialog, AlertDialog

.

A. Text

Text widget is used to display a piece of string. The style of the string can be set by using style property and TextStyle class. The sample code for this purpose is as follows −

The most important properties of the Text widget are as follows −

maxLines, int − Maximum number of lines to show

overflow, TextOverFlow − Specify how visual overflow is handled using TextOverFlow class

style, TextStyle − Specify the style of the string using TextStyle class

textAlign, TextAlign − Alignment of the text like right, left, justify, etc., using TextAlign class

textDirection, TextDirection − Direction of text to flow, either left-to-right or right-to-left

B. Image

Image widget is used to display an image in the application. Image widget provides different constructors to load images from multiple sources and they are as follows −

Image − Generic image loader using ImageProvider

Image.asset − Load image from flutter project's assets

Image.file − Load image from system folder

Image.memory − Load image from memory

Image.Network − Load image from network

The easiest option to load and display an image in Flutter is by including the image as assets of the application and load it into the widget on demand.

Create a folder, assets in the project folder and place the necessary images.

Specify the assets in the pubspec.yaml as shown below −

```
flutter:

  assets:

    - assets/smiley.png
```

Now, load and display the image in the application.

Image.asset('assets/smiley.png')

The most important properties of the Image widget are as follows −

image, ImageProvider − Actual image to load

width, double − Width of the image

height, double − Height of the image

alignment, AlignmentGeometry − How to align the image within its bounds

C. Icon

Icon widget is used to display a glyph from a font described in IconData class. The code to load a simple email icon is as follows −

D. 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. We can use it as like below code snippets.

Conclusion: In this experiment we have created user interface by using flutter common widgets.