**Experiment No:02**

**Aim:** To design Flutter UI by including common widgets.

**Theory:**

**Common Widgets in Flutter**

Flutter offers a diverse array of widgets that empower developers to craft dynamic and engaging user interfaces. These widgets fulfill a multitude of functions, from presenting textual content and images to managing user interactions and organizing layouts. Below, we delve into some of the frequently utilized widgets in Flutter:

1. **Text Widgets:**

* **Text:** Renders a text string with customizable attributes like font size, color, and alignment.
* **RichText:** Enables intricate text formatting by supporting inline styles and multiple text spans.

1. **Input Widgets:**

* **TextField:** Facilitates user text input with options for customization and validation.
* **Form:** Provides a way to group and validate multiple form fields together.

1. **Button Widgets:**

* **ElevatedButton:** Represents a raised button, typically used for primary actions.
* **TextButton:** Displays a text-based button, suitable for secondary actions.
* **IconButton:** Exhibits a button adorned with an icon, often used for navigation or actions.
* **FloatingActionButton:** Showcases a circular button, commonly employed for prominent actions.

1. **Selection Widgets:**

* **Checkbox:** Allows users to toggle a binary state, such as on/off or selected/unselected.
* **Radio:** Enables users to select a single option from a set of mutually exclusive choices.
* **Switch:** Offers a toggle switch for binary options.
* **Slider:** Allows users to choose a value from a range by sliding a thumb along a track.

1. **Layout Widgets:**

* **Row:** Aligns child widgets horizontally in a row.
* **Column:** Aligns child widgets vertically in a column.
* **Stack:** Stacks widgets on top of each other, enabling complex UI compositions.
* **Container:** Provides a versatile layout widget for customizing child position, size, and appearance.

1. **Scrolling Widgets:**

* **ListView:** Displays a scrollable list of widgets, either vertically or horizontally.
* **GridView:** Presents a grid of widgets in rows and columns, with support for scrolling and item customization.
* **SingleChildScrollView:** Enables scrolling of a single child widget in one direction.

1. **Material Design Widgets:**

* **AppBar:** Represents the top app bar for navigation and branding purposes.
* **Scaffold:** Implements the basic material design layout structure, including app bars, drawers, and bottom navigation.
* **Card:** Displays a material design card, often used to showcase related content or information.

1. **Interaction Widgets:**

* **GestureDetector:** Detects various gestures like taps, swipes, and drags on its child widget.
* **InkWell:** Provides a material design inkwell effect that responds to touches with a splash effect, offering tactile feedback.

**Code:**

import "dart:io";

import "package:flutter/material.dart";

import "package:flutter/widgets.dart";

import "package:image\_picker/image\_picker.dart";

class AddCourse extends StatefulWidget {

const AddCourse({super.key});

@override

State<AddCourse> createState() => \_AddCourseState();

}

class \_AddCourseState extends State<AddCourse> {

final ImagePicker \_picker = ImagePicker();

File? selectedImage;

Future getImage()async{

var image = await \_picker.pickImage(source: ImageSource.gallery);

selectedImage = File(image!.path);

setState(() {

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(

"Add Courses",

style: TextStyle(

color: Colors.black,

fontSize: 30.0,

fontWeight: FontWeight.bold,

fontFamily: 'Poppins'),

)),

body: Container(

margin: EdgeInsets.only(top: 30.0, left: 20.0, right: 20.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

"Upload the Course banner Picture",

style: TextStyle(

color: Colors.black,

fontSize: 18.0,

fontFamily: 'Poppins',

fontWeight: FontWeight.bold),

),

SizedBox(

height: 20.0,

),

selectedImage==null?GestureDetector(

onTap: (){

getImage();

},

child: Center(

child: Material(

elevation: 4.0,

borderRadius: BorderRadius.circular(20),

child: Container(

width: 150,

height: 150,

decoration: BoxDecoration(

border: Border.all(color: Colors.black, width: 1.5),

borderRadius: BorderRadius.circular(20)),

child: Icon(Icons.camera\_alt\_outlined,color: Colors.black),

)),

),

):Center(

child: Material(

elevation: 4.0,

borderRadius: BorderRadius.circular(20),

child: Container(

width: 150,

height: 150,

decoration: BoxDecoration(

border: Border.all(color: Colors.black, width: 1.5),

borderRadius: BorderRadius.circular(20)),

child: Image.file(

selectedImage!,

fit: BoxFit.cover,

),

)),

),

SizedBox(

height: 20.0,

),

Text(

"Course Name",

style: TextStyle(

color: Colors.black,

fontSize: 18.0,

fontFamily: 'Poppins',

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 20.0,),

Container(

padding: EdgeInsets.symmetric(horizontal: 20.0),

width: MediaQuery.of(context).size.width,

decoration: BoxDecoration(

color: Color(0xffececf8),

borderRadius: BorderRadius.circular(10)),

child: TextField(

decoration: InputDecoration(

border: InputBorder.none,

hintText: "Course Name"

),

),

),

SizedBox(height: 30.0,),

Container(

width:MediaQuery.of(context).size.width,

padding: EdgeInsets.symmetric(vertical:5.0 ) ,

decoration: BoxDecoration(

color: Color(0xFF0F4E58),borderRadius: BorderRadius.circular(30)

),

child: Center(

child: Text("Add",style: TextStyle(

color: Colors.white,

fontSize: 20.0,

fontFamily: 'Poppins',

fontWeight: FontWeight.bold

),),

),

)

],

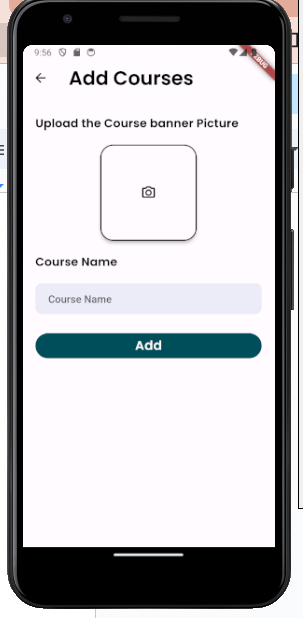
),

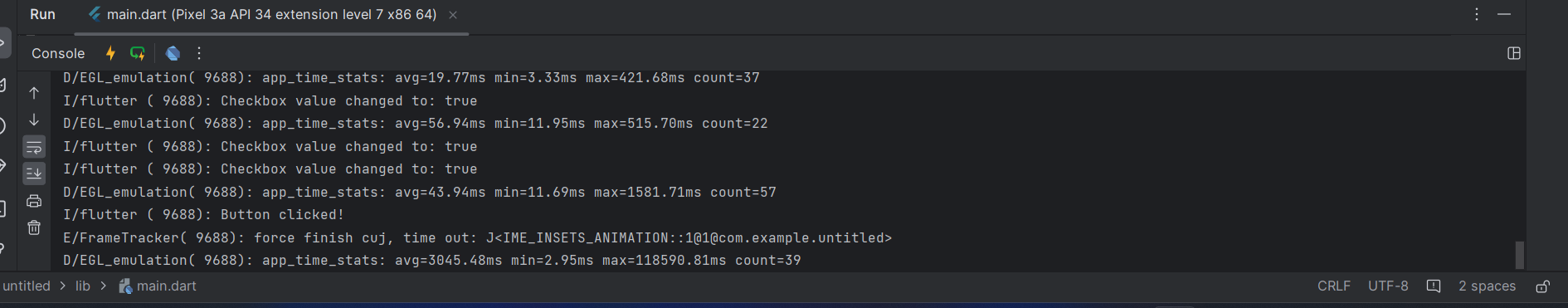
),

);

}

}





**Conclusion:** I have successfully studied and used different Common Widgets used in Flutter UI such as Column Widget, Scaffold, Text, SizedBox etc.