**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 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Exp-2',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: MyHomePage(),

);

}

}

class MyHomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Flutter Expt-2'),

),

body: Center(child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Welcome to Flutter Exp-2',

style: TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: () {

print('Button clicked!');

},

child: Text('Click Me'),

),

SizedBox(height: 20),

TextField(

decoration: InputDecoration(

hintText: 'Enter your name',

labelText: 'Name',

border: OutlineInputBorder(),

),

),

SizedBox(height: 20),

CheckboxListTile(

title: Text('I agree to the terms and conditions'),

value: false,

onChanged: (value) {

print('Checkbox value changed to: $value');

},

),

SizedBox(height: 20),

DropdownButton<String>(

value: 'Option 1',

onChanged: (value) {

print('Dropdown value changed to: $value');

},

items: <String>['Option 1', 'Option 2', 'Option 3']

.map<DropdownMenuItem<String>>((String value) {

return DropdownMenuItem<String>(

value: value,

child: Text(value),);

}).toList(),

),

],

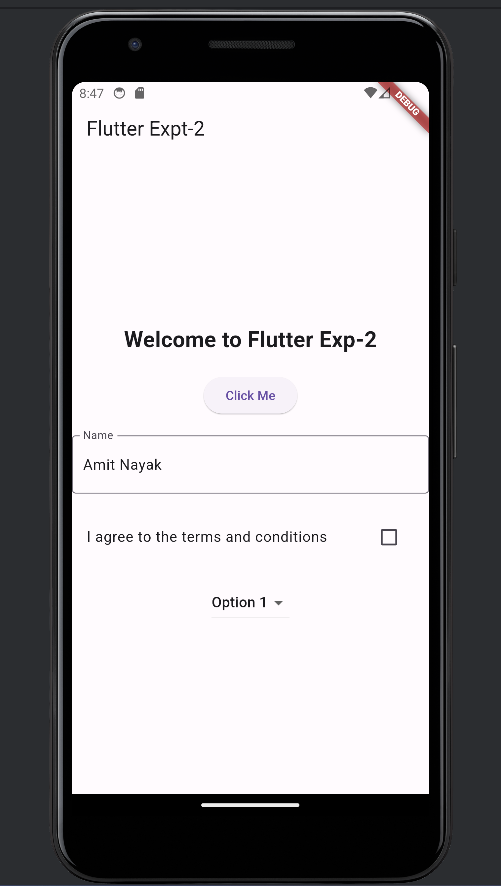
),

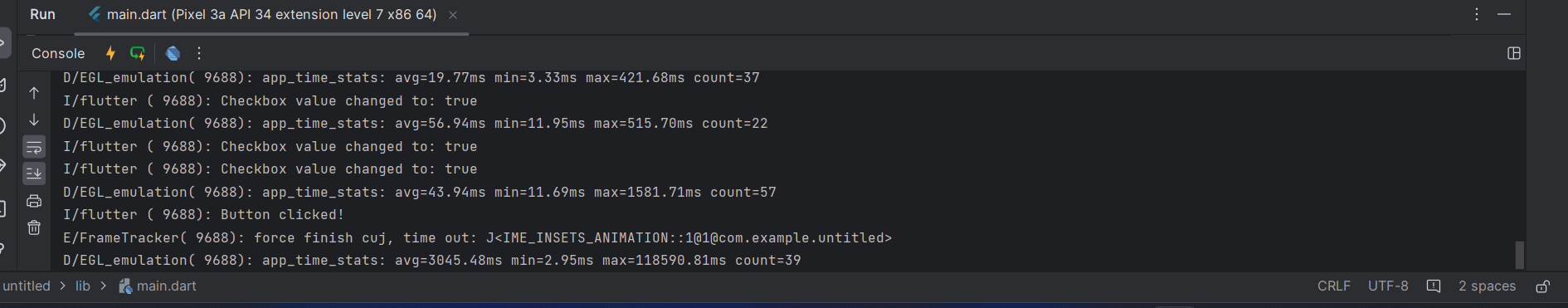
),

);

}

}





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