Name: Mitesh A. Dalvi Rol no.: 11
Class: D15B

EXPERIMENT 2

Aim : To design Flutter UI by including common widgets.

Theory :

Flutter Widgets :
Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that you build your UI out of widgets. Widgets describe what their view should look like given their current configuration and state.

Widgets in Flutter are categorized into two main types:

StatelessWidget: Widgets that are immutable, meaning their properties (such as color, size, text) cannot change once they are built. Examples include Text, Icon, Container, and Image.

StatefulWidget: Widgets that maintain state, meaning their properties can change over time, causing the UI to update accordingly. Examples include Checkbox, Slider, TextField, and ListView.

Flutter provides a rich library of pre-built widgets that developers can use to construct their app's UI. Additionally, developers can create their own custom widgets by composing existing widgets or extending StatelessWidget or StatefulWidget classes.

Widgets in Flutter follow a declarative programming paradigm, where you describe how the UI should look based on the current state of the app. Flutter's framework then takes care of efficiently updating the UI to reflect any changes in state. This approach allows for fast, expressive, and flexible UI development.

Types of Widgets :

Flutter offers a wide range of widgets for building user interfaces. Here's a list of some common widgets along with brief explanations and examples:

1. Text: Used for displaying text.
   Text('Hello, Flutter!')

2. Image: Displays an image.

```
  Image.network('https://example.com/image.jpg')
```

3. Container: A container that can have a background color, padding, and more.

```
Container(
  color: Colors.blue,
  padding: EdgeInsets.all(16.0),
  child: Text('Container Example'),
)
```

4. Row and Column: Used for arranging widgets horizontally and vertically, respectively.

```
Row(
  children: <Widget>[
    Text('Item 1'),
    Text('Item 2'),
  ],
)

Column(
  children: <Widget>[
    Text('Item 1'),
    Text('Item 2'),
  ],
)
```

5. ListView: A scrollable list of widgets.

```
ListView(
  children: <Widget>[
    ListTile(title: Text('Item 1')),
    ListTile(title: Text('Item 2')),
  ],
)
```

6. Card: A material design card.

```
Card(
  child: ListTile(
    title: Text('Card Example'),
  ),
)
```

7. TextField: Allows users to input text.

```
TextField(
  decoration: InputDecoration(labelText: 'Enter your name'),
)
```

8. Button: There are various types of buttons like RaisedButton, FlatButton, and IconButton.

```
RaisedButton(
  onPressed: () {
    // Handle button press
  },
  child: Text('Press Me'),
)
```

9. Checkbox and Radio: For checkboxes and radio buttons.

```
Checkbox(
  value: true,
  onChanged: (bool newValue) {
    // Handle checkbox change
  },
)

Radio(
  value: 1,
  groupValue: selectedValue,
  onChanged: (int? newValue) {
    // Handle radio button change
  },
)
```

10. Switch: A material design switch.

```
Switch(
  value: switchValue,
  onChanged: (bool newValue) {
    // Handle switch change
  },
)
```

11. DropdownButton: A dropdown button for selecting items.

```
DropdownButton<String>(
  value: selectedValue,
  items: <String>['Option 1', 'Option 2', 'Option 3']
      .map<DropdownMenuItem<String>>((String value) {
```

```
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value),
    );
  }).toList(),
  onChanged: (String? newValue) {
    // Handle dropdown selection
  },
)
```

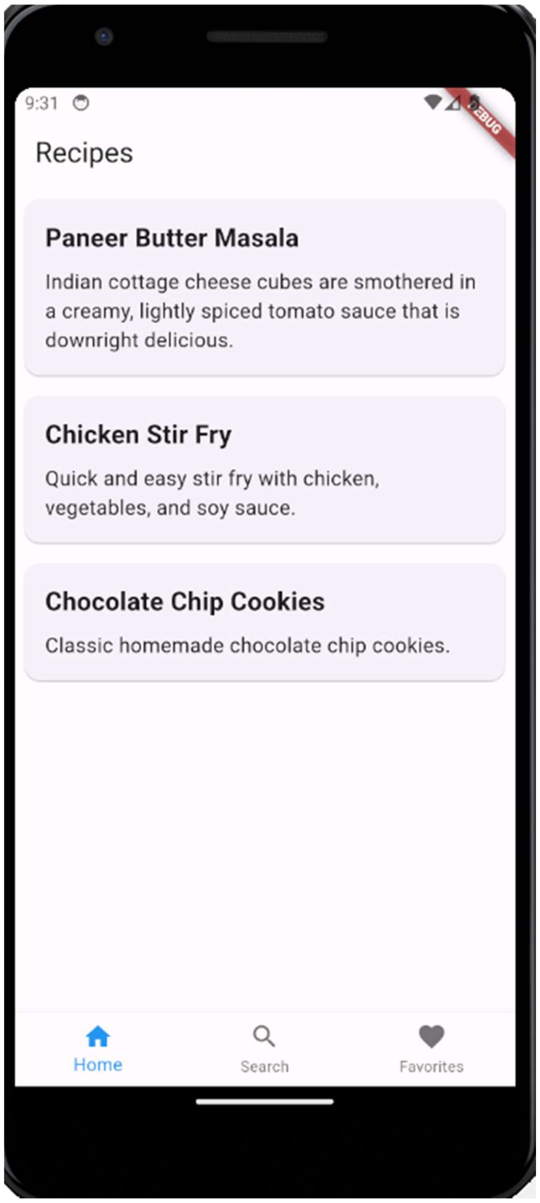
12. SnackBar: A temporary message that appears at the bottom of the screen.

```
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(
    content: Text('This is a SnackBar'),
  ),
)
```

Program :

```
import 'package:flutter/material.dart';

void main() {
  runApp(RecipeApp());
}

class RecipeApp extends StatefulWidget {
  @override
  _RecipeAppState createState() => _RecipeAppState();
}

class _RecipeAppState extends State<RecipeApp> {
  int _selectedIndex = 0;

  static List<Widget> _widgetOptions = <Widget>[
    HomePage(),
    // Add more screens here as needed
    Placeholder(), // Example additional screen
    Placeholder(), // Example additional screen
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Recipe App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(
          title: Text('Recipes'),
        ),
        body: Center(
          child: _widgetOptions.elementAt(_selectedIndex),
        ),
        bottomNavigationBar: BottomNavigationBar(
          items: const <BottomNavigationBarItem>[
            BottomNavigationBarItem(
              icon: Icon(Icons.home),
              label: 'Home',
            ),
            // Add more bottom navigation bar items as needed
            BottomNavigationBarItem(
              icon: Icon(Icons.search),
              label: 'Search',
            ),
            BottomNavigationBarItem(
              icon: Icon(Icons.favorite),
              label: 'Favorites',
            ),
          ],
          currentIndex: _selectedIndex,
          selectedItemColor: Colors.blue,
          onTap: _onItemTapped,
        ),
      ),
    );
  }
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ListView(
      children: <Widget>[
        RecipeCard(
          title: 'Paneer Butter Masala',
          description:
              'Indian cottage cheese cubes are smothered in a creamy, lightly spiced
tomato sauce that is downright delicious.',
        ),
        RecipeCard(
          title: 'Chicken Stir Fry',
          description:
              'Quick and easy stir fry with chicken, vegetables, and soy sauce.',
        ),
        RecipeCard(
          title: 'Chocolate Chip Cookies',
          description: 'Classic homemade chocolate chip cookies.',
        ),
      ],
    );
  }
}

class RecipeCard extends StatelessWidget {
  final String title;
  final String description;

  const RecipeCard({
    Key? key,
    required this.title,
    required this.description,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: EdgeInsets.all(8),
      child: Padding(
        padding: EdgeInsets.all(16),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Text(
              title,
              style: TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
              ),
            ),
            SizedBox(height: 8),
            Text(
              description,
              style: TextStyle(fontSize: 16),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output:



Paneer Butter Masala

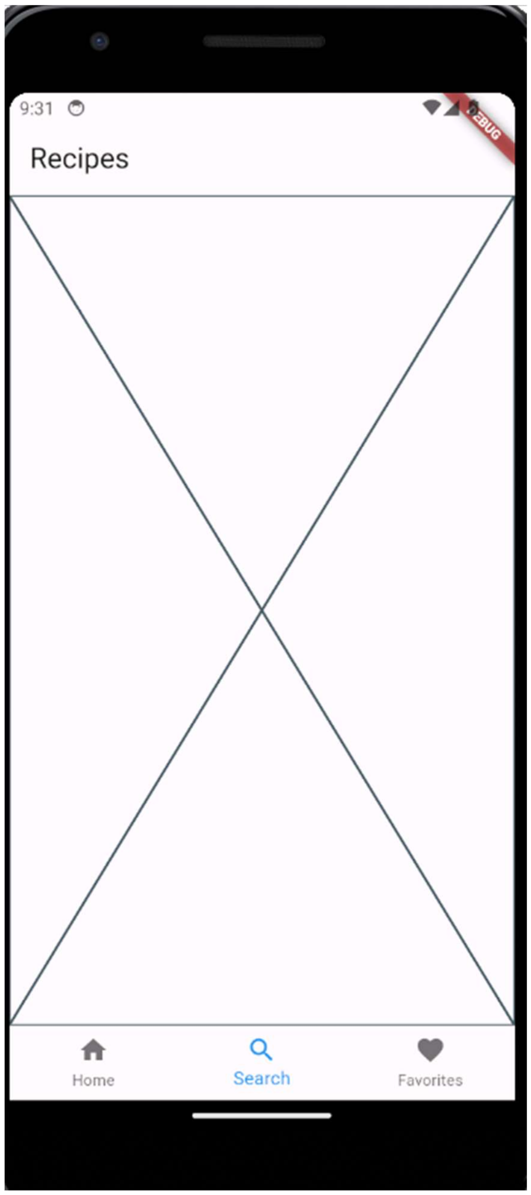
Indian cottage cheese cubes are smothered in a creamy, lightly spiced tomato sauce that is downright delicious.

Chicken Stir Fry

Quick and easy stir fry with chicken, vegetables, and soy sauce.

Chocolate Chip Cookies

Classic homemade chocolate chip cookies.



Conclusion : From this experiment, we have understood what are widgets and how to use it to create a basic UI. Also, how to orient the widgets to make a UI interactive and creative to the user. We have used various types of widgets like row, column to align and text as well.