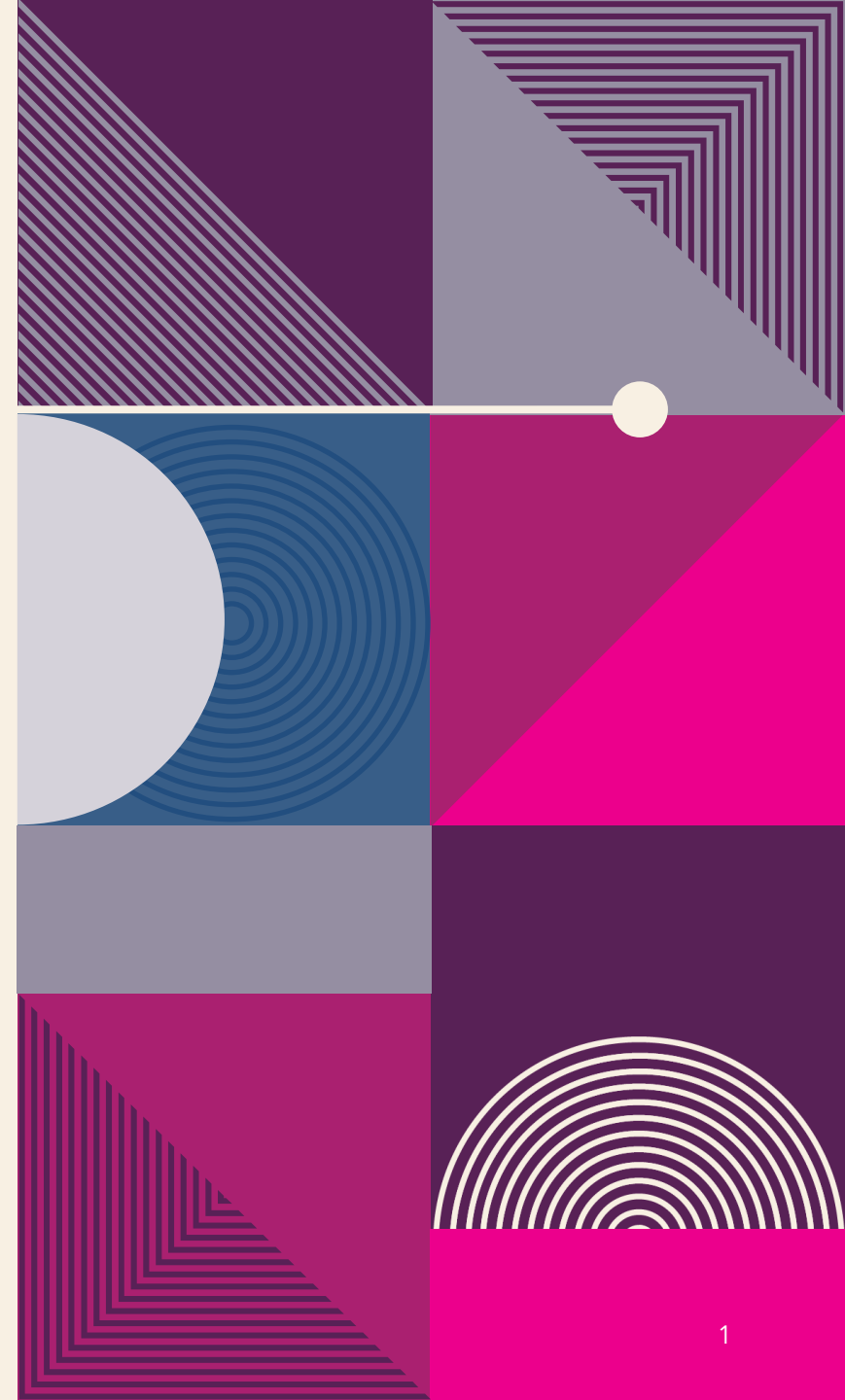


Using Common Widgets

Flutter Buttons



Flutter Buttons

Buttons are the graphical control element that **provides a user to trigger an event** such as taking actions, making choices, searching things, and many more. They can be placed anywhere in our UI like dialogs, forms, cards, toolbars, etc.

Buttons are the Flutter widgets, which is a part of the material design library. Flutter provides several types of buttons that have different shapes, styles, and features.

Features of Buttons

The standard features of a button in Flutter are given below:

1. We can easily apply themes on buttons, shapes, color, animation, and behavior.
2. We can also theme icons and text inside the button.
3. Buttons can be composed of different child widgets for different characteristics.

Types of Flutter Buttons

1. Text Button

It is a **text label button** that does not have much decoration and displayed **without any elevation**. The flat button has two required properties that are: **child and onPressed()**. It is mostly used in toolbars, dialogs, or inline with other content. By default, the flat button has no color, and its text is black. But, we can use color to the button and text using **color and textColor** attributes, respectively.

2. ElevatedButton

It is a button, which is based on the material widget and has a **rectangular body**. It is similar to a text button, but it **has an elevation** that will increase when the button is pressed. It adds dimension to the UI along Z-axis. It has several properties like text color, shape, padding, button color, the color of a button when disabled, animation time, elevation, etc.

This button has **two callback functions**.

onPressed(): It is triggered when the button is pressed.

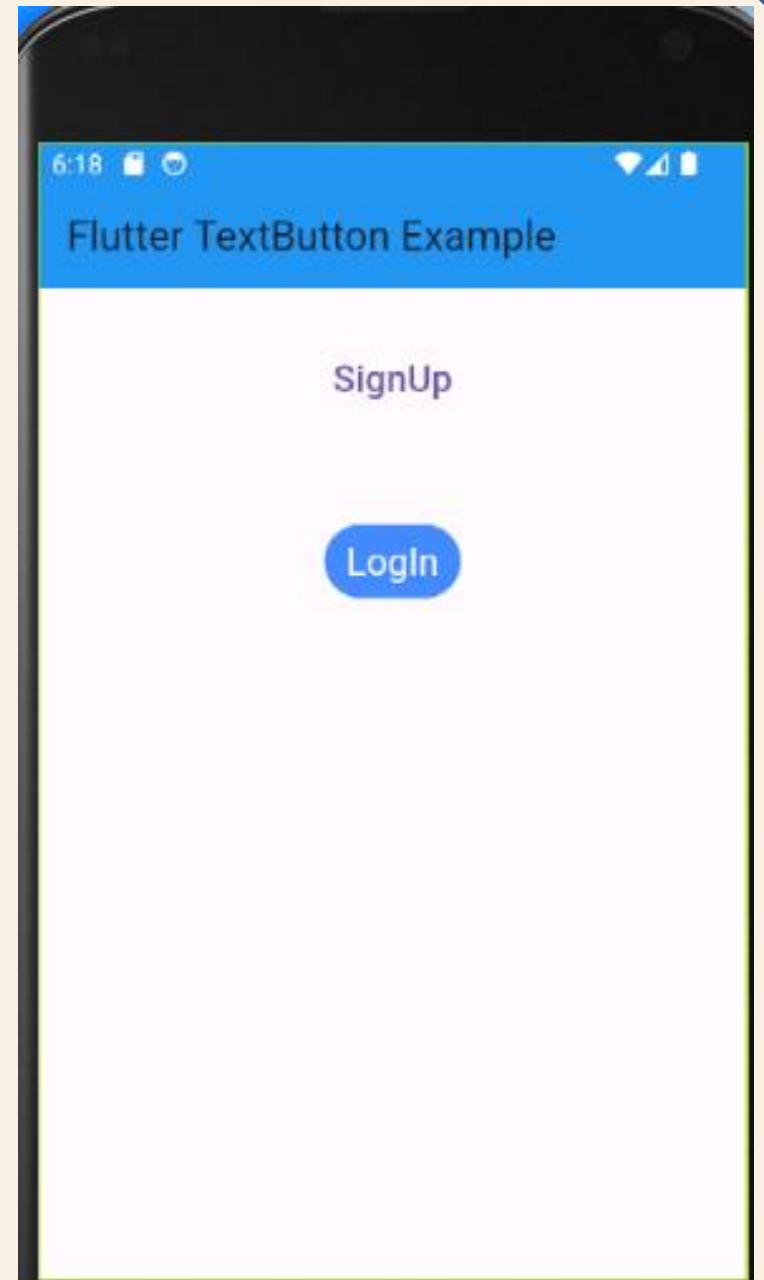
onLongPress(): It is triggered when the button is long pressed.

It is to note that this button is in a **disabled state** if `onPressed()` and `onLongPressed()` callbacks are not specified.

```

body: Center(
  child: Column(children: <Widget>[
    Container(
      margin: EdgeInsets.all(25),
      child: TextButton(
        child: Text( 'SignUp',
          style: TextStyle(fontSize: 20.0), ),
        onPressed: () {}, ),
      ),
    Container(
      margin: EdgeInsets.all(25),
      child: TextButton(
        style: TextButton.styleFrom(
          backgroundColor: Colors.blueAccent,
          foregroundColor: Colors.white,
          textStyle: TextStyle(fontSize: 20)),
        child: Text(
          'LogIn',
          style: TextStyle(fontSize: 20.0),
        ),
        onPressed: () {}, ),
      ),
    ),
  ),

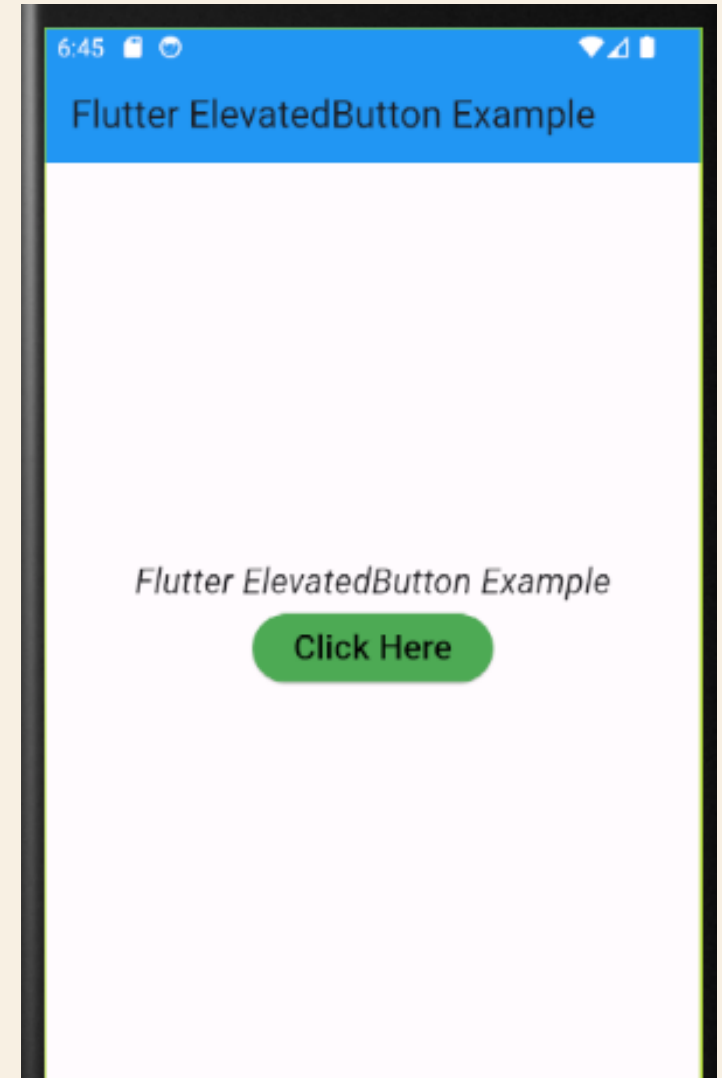
```



```

body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text(
        msg,
        style: TextStyle(fontSize: 20, fontStyle: FontStyle.italic),),
      ElevatedButton(
        onPressed: _changeText,
        style: ButtonStyle(
          foregroundColor: MaterialStateProperty.all(Colors.black),
          backgroundColor:
            MaterialStateProperty.all(Colors.green), ),
        child: Text(
          "Click Here",
          style: TextStyle(fontSize: 20), ), ) ], ),), ), ); }
_changeText() {
  setState(() {
    if (msg.startsWith('F')) {
      msg = 'We have learned Flutter Elevated button example.';
    } else {
      msg = 'Flutter ElevatedButton Example';
    } }); }}

```



3. Floating Action Button

A FAB button is a **circular icon button** that triggers the primary action in our application. It is the most used button in today's applications. We can use this button for adding, refreshing, or sharing the content. Flutter suggests using at most one FAB button per screen. There are two types of Floating Action Button:

FloatingActionButton: It creates a simple circular floating button with a child widget inside it. It must have a child parameter to display a widget.

FloatingActionButton.extended: It creates a wide floating button along with an icon and a label inside it. Instead of a child, it uses labels and icon parameters.

4. DropDown Button

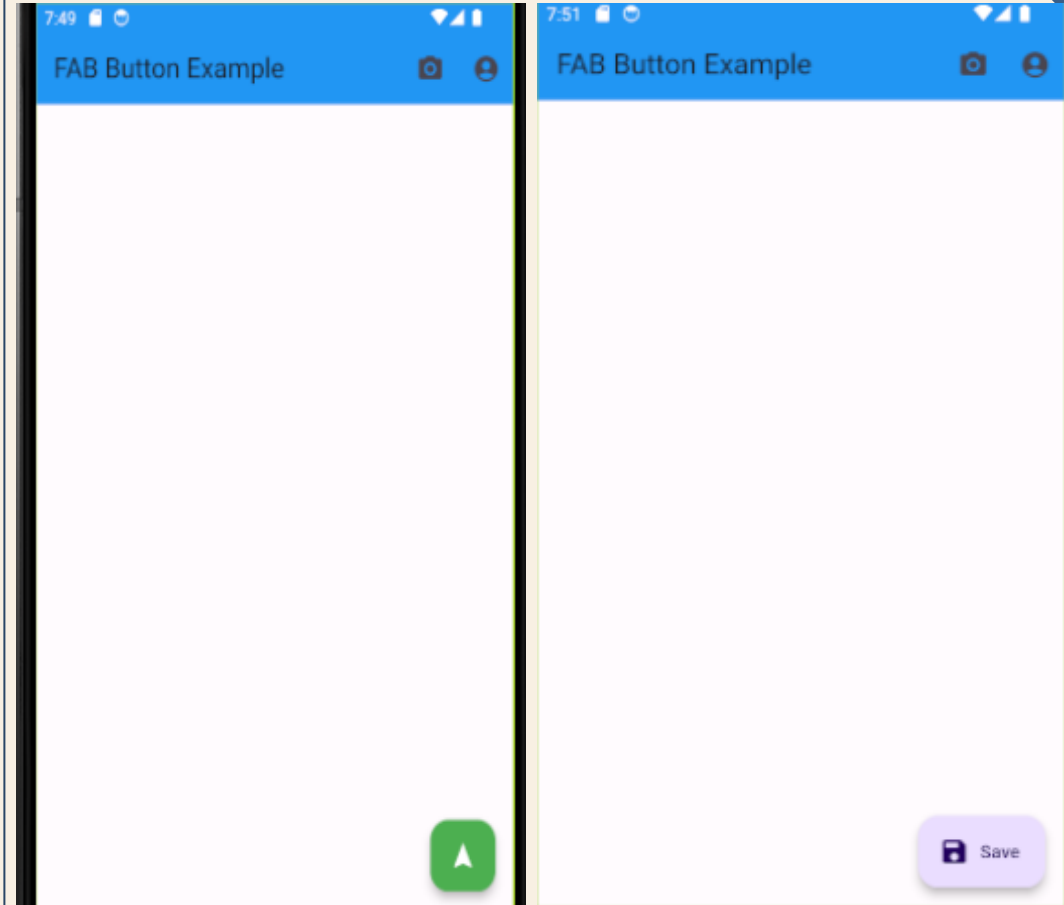
A drop-down button is used to create a nice overlay on the screen that allows the user to select any item from multiple options. Flutter allows a simple way to implement a drop-down box or drop-down button. This button shows the currently selected item and an arrow that opens a menu to select an item from multiple options.

Flutter provides a **DropDownButton widget** to implement a drop-down list. We can place it anywhere in our app.

```

actions: <Widget>[
    IconButton(icon: Icon(Icons.camera_alt),
onPressed: () => {}),
    IconButton(icon: Icon(Icons.account_circle),
onPressed: () => {}
    ],
),
floatingActionButton: FloatingActionButton(
    backgroundColor: Colors.green,
    foregroundColor: Colors.white,
    onPressed: () => {},
    child: Icon(Icons.navigation),
),
/*floatingActionButton:FloatingActionButton.extended(
    onPressed: () {},
    icon: Icon(Icons.save),
    label: Text("Save"),
), */
);
}

```



5. Icon Button

An IconButton is a **picture printed** on the Material widget. It is a useful widget that gives the Flutter UI a material design feel. We can also customize the look and feel of this button. In simple terms, it is an icon that reacts when the user will touch it.

6. Inkwell Button

InkWell button is a material design concept, which is used for **touch response**. This widget comes under the Material widget where the ink reactions are actually painted. It creates the app UI interactive by adding gesture feedback. It is mainly used for adding **splash ripple effect**.

7. PopupMenu Button

It is a button that **displays the menu** when it is pressed and then calls the **onSelected** method the menu is dismissed. It is because the item from the multiple options is selected. This button contains a text and an image. It will mainly use with **Settings** menu to list all options. It helps in making a great user experience.

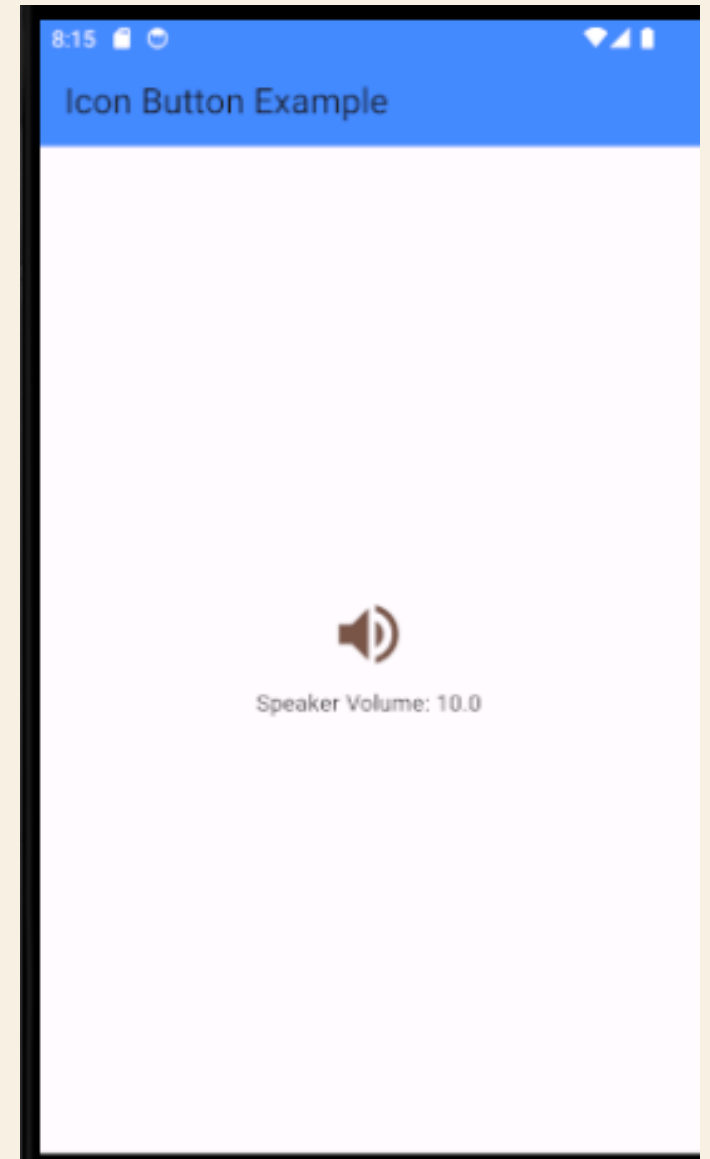
8. Outline Button

It is similar to the flat button, but it contains a thin grey rounded rectangle border. Its outline border is defined by the shape attribute.


```

body: Center(
  child: MyStatefulWidget(), ),),,);}}
double _speakervolume = 0.0;
class MyStatefulWidget extends StatefulWidget {
  MyStatefulWidget({Key? key}) : super(key: key);
  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: <Widget>[
        IconButton(
          icon: const Icon(Icons.volume_up),
          iconSize: 50,
          color: Colors.brown,
          tooltip: 'Increase volume by 5',
          onPressed: () {
            setState(() {
              _speakervolume += 5; }); }, ),
        Text('Speaker Volume: $_speakervolume') ], );
  }
}

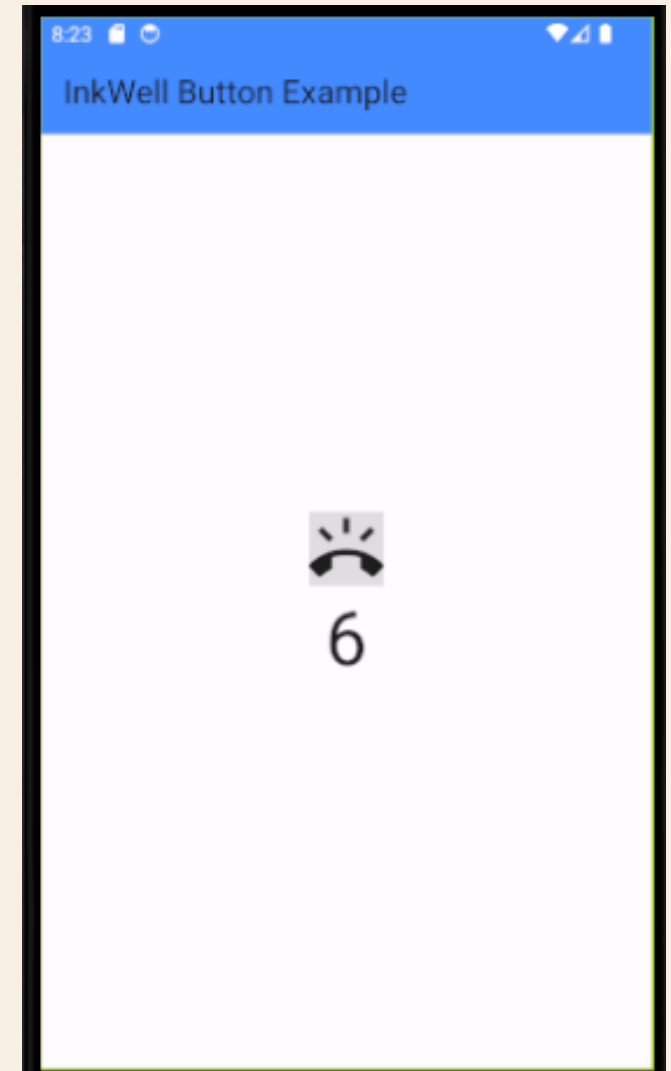
```



```

body: Center(
  child: Column(
    mainAxisAlignment:
MainAxisAlignment.center,
    children: <Widget>[
      InkWell(
        splashColor: Colors.green,
        highlightColor: Colors.blue,
        child: Icon(Icons.ring_volume, size: 50),
        onTap: () {
          setState(() {
            _volume += 2; }); },),
      Text(_volume.toString(), style: TextStyle(fontSize:
50)), ],
    ),
  ),
); }

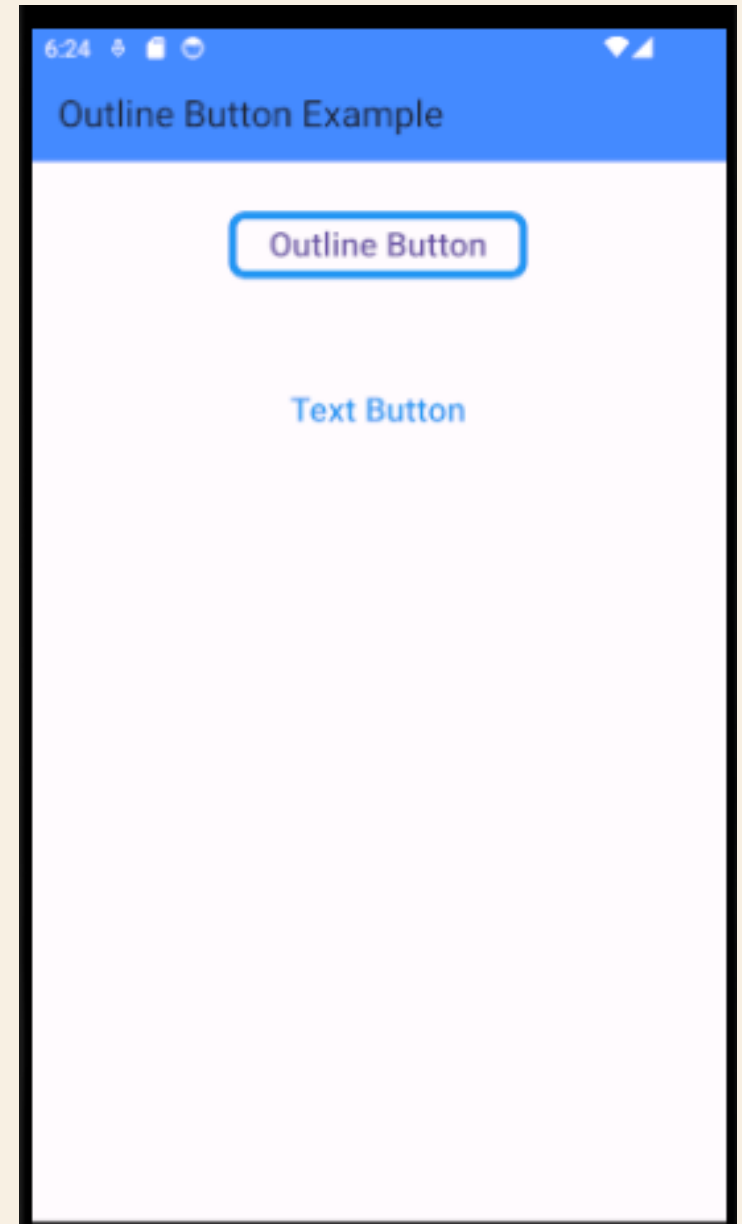
```



```

body: Center(
  child: Column(
    children: <Widget>[
      Container(
        margin: EdgeInsets.all(25),
        child: OutlinedButton(
          style: OutlinedButton.styleFrom(
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(10.0), ),
            side: const BorderSide(
              width: 5.0,
              color: Colors.blue, ), ),
          onPressed: () {},
          child: Text(
            "Outline Button",
            style: TextStyle(fontSize: 20.0), ), ), ),
      Container(
        margin: EdgeInsets.all(25),
        child: TextButton(
          child: Text( 'Text Button',
            style: TextStyle(fontSize: 20.0), ),
          style: TextButton.styleFrom(
            foregroundColor: Colors.blue, ),
          onPressed: () {}, ), ), ], ), ),

```



Button Bar

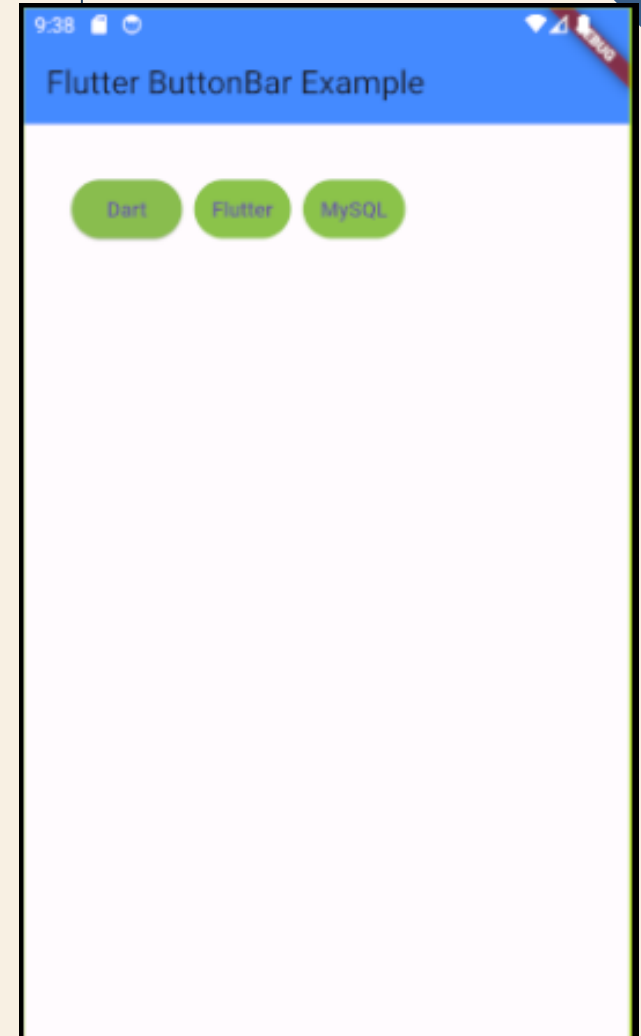
Flutter provides the flexibility to **arrange the buttons in a bar or a row**. ButtonBar widget contains three properties: **alignment, children, and mainAxisAlignment**.

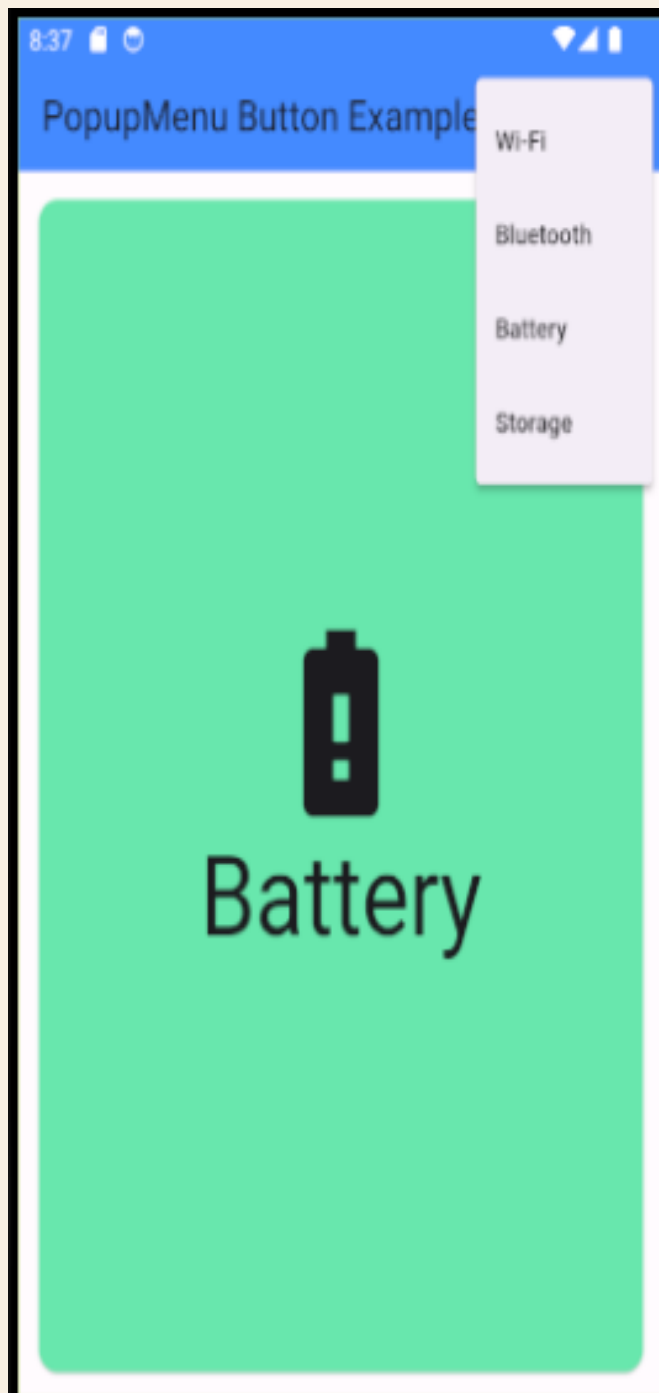
- Alignment is used to present the aligning option to the entire button bar widget.
- Children attribute is used to take the number of buttons in a bar.
- mainAxisSize attribute is used to provide the horizontal space for the button bar.

```

body: Padding(
  padding: EdgeInsets.all(10),
  child: Column(
    children: <Widget>[
      Padding(
        padding: EdgeInsets.all(15),
        child: ButtonBar(
          mainAxisAlignment: MainAxisAlignment.min,
          children: <Widget>[
            ElevatedButton(
              child: new Text('Dart'),
              style: ButtonStyle(
                backgroundColor: MaterialStatePropertyAll( Colors.lightGreen,), ),
              onPressed: () {/** */}, ),
            TextButton(
              child: Text('Flutter'),
              style: ButtonStyle(
                backgroundColor: MaterialStatePropertyAll( Colors.lightGreen, ), ),
              onPressed: () {/** */}, ),
            TextButton( child: Text('MySQL'),
              style: ButtonStyle( backgroundColor: MaterialStatePropertyAll( Colors.lightGreen ),
                ),
              onPressed: () {/** */}, ), ], ), ],
    )); }

```





```

return MaterialApp(
  debugShowCheckedModeBanner: false,
  home: Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.blueAccent,
      title: const Text('PopupMenu Button Example'),
      actions: <Widget>[
        PopupMenuButton<Choice>(
          onSelected: _select,
          itemBuilder: (BuildContext context) {
            return choices.skip(0).map((Choice choice) {
              return PopupMenuItem<Choice>(
                value: choice,
                child: Text(choice.name), );
            }).toList();}, ), ],
      ),
    body: Padding(
      padding: const EdgeInsets.all(10.0),
      child: ChoiceCard(choice: _selectedOption),
    ),
  ),
);
}

```

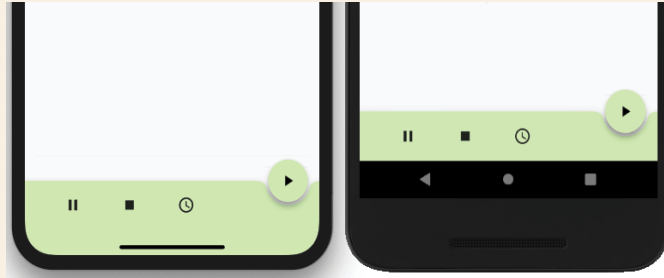
```

class Choice {
  const Choice({required this.name, required this.icon});
  final String name;
  final IconData icon;}
const List<Choice> choices = const <Choice>[
  const Choice(name: 'Wi-Fi', icon: Icons.wifi),
  const Choice(name: 'Bluetooth', icon: Icons.bluetooth),
  const Choice(name: 'Battery', icon: Icons.battery_alert),
  const Choice(name: 'Storage', icon: Icons.storage),];
class ChoiceCard extends StatelessWidget {
  const ChoiceCard({Key? key, required this.choice}) : super(key: key);
  final Choice choice;
  @override
  Widget build(BuildContext context) {
    final TextStyle? textStyle = Theme.of(context).textTheme.displayLarge;
    return Card(
      color: Colors.greenAccent,
      child: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: <Widget>[
            Icon(choice.icon, size: 115.0, color: textStyle?.color),
            Text(choice.name, style: textStyle), ], ),
    );
  }
}

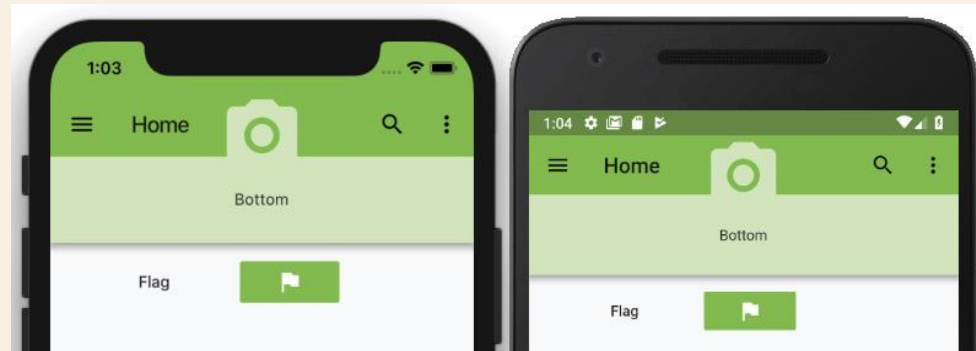
```


Buttons

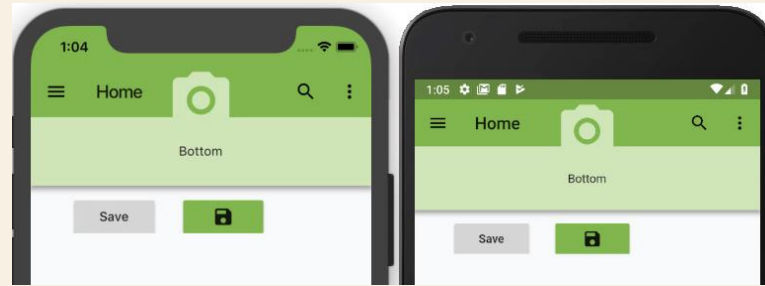
1. **FloatingActionButton** : The **FloatingActionButton** widget is usually placed on the bottom **right** or **center** of the **main screen** in the **Scaffold floatingActionButton** property.



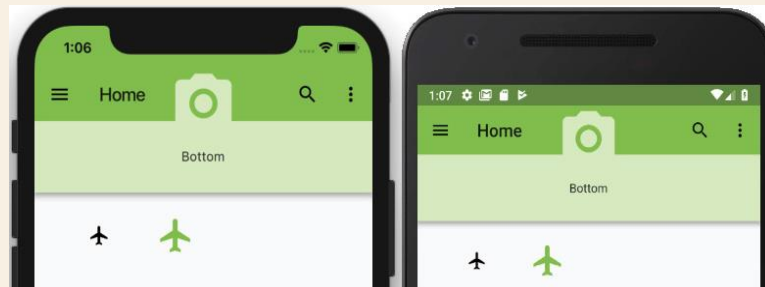
2. **TextButton**: A text button is a **label** child displayed on a (zero elevation) Material widget. The label's **Text** and **Icon** widgets are displayed in the style's **ButtonStyle.foregroundColor**.



3. ElevatedButton : An elevated button is a label [child](#) displayed on a [Material](#) widget whose [Material.elevation](#) increases when the button is pressed. The label's [Text](#) and [Icon](#) widgets are displayed in [style](#)'s [ButtonStyle.foregroundColor](#) and the button's filled background is the [ButtonStyle.backgroundColor](#).



4. IconButton : The [IconButton](#) widget uses an [Icon](#) widget on a [Material Component](#) widget that reacts to touches by filling with color (ink).



5. PopupMenuButton : The [PopupMenuButton](#) widget displays a list of menu items. When a menu item is pressed, the value passes to the [onSelected](#) property.

Creating the **PopupMenuButton** and the **Items' Class** and **List**

Before you add the **PopupMenuButton** widgets, let's create the **Class** and **List** necessary to **build the items** to be displayed.

• **Create** a **TodoMenuItem** class. **make sure it's not inside another class**. Create the **class** and **list** at the end of the file after the last closing curly bracket, `}`. The **TodoMenuItem** class contains a title and an icon.

```
class TodoMenuItem {
    final String title;
    final Icon icon;

    TodoMenuItem({this.title, this.icon});
}
```

1. **Create** a **List** of **TodoMenuItem**. This **List<TodoMenuItem>** will be called **foodMenuList** and will contain a List (array) of **TodoMenuItems**.

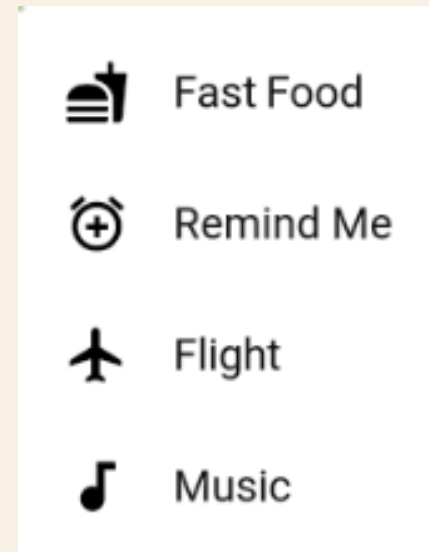
```
List<TodoMenuItem> foodMenuList = [
    TodoMenuItem(title: 'Fast Food', icon: Icon(Icons.fastfood)),
    TodoMenuItem(title: 'Remind Me', icon: Icon(Icons.add_alarm)),
    TodoMenuItem(title: 'Flight', icon: Icon(Icons.flight)),
    TodoMenuItem(title: 'Music', icon: Icon(Icons.audiotrack)),
];
```

- 3. Create** a **PopupMenuButton**. You will use an **itemBuilder** to build the **List** of **TodoMenuItems**.
- If you do not set an **icon** for the **PopupMenuButton**, a default menu icon is used by default.
 - The **onSelected** will retrieve the item selected on the list.
 - Use the **itemBuilder** to build a list of **foodMenuList** and **map** to **TodoMenuItem**.
 - A **PopupMenuItem** is returned for each item in the **foodMenuList**. For the **PopupMenuItem** child, you use a **Row** widget to show the **Icon** and **Text** widgets together.

```

child: PopupMenuButton<TodoMenuItem>(
  icon: Icon(Icons.view_list),
  onSelect: ((valueSelected) {
    print('valueSelected: ${valueSelected.title}');
  }),
  itemBuilder: (BuildContext context) {
    return foodMenuList.map((TodoMenuItem todoMenuItem) {
      return PopupMenuItem<TodoMenuItem>(
        value: todoMenuItem,
        child: Row(
          children: <Widget>[
            Icon(todoMenuItem.icon.icon),
            Padding(
              padding: EdgeInsets.all(8.0),
            ),
            Text(todoMenuItem.title),
          ],
        ),
      );
    }).toList();
  },
),

```



4. **Modify** the **AppBar** bottom property by adding the **widget class** name: **PopupMenuButtonWidget()**.

```
bottom: PreferredSize(  
  child: Container(  
    color: Colors.lightGreen.shade100,  
    height: 75.0,  
    width: double.infinity,  
    child: Center(  
      child: const PopupMenuButtonWidget(),  
    ),  
  ),  
  preferredSize: Size.fromHeight(75.0),  
),
```

5. Create the **PopupMenuButtonWidget()** widget **class** after the **ColumnAndRowNestingWidget()** widget **class**.

- Since the bottom property is expecting a **PreferredSizeWidget**, you use the keyword **implements** **PreferredSizeWidget** in the class declaration. The class **extends** the **StatelessWidget** and **implements** the **PreferredSizeWidget**.

```
@override  
// implement preferredSize  
Size get preferredSize => Size.fromHeight(75.0);
```

The following is the **full PopupMenuButtonWidget** widget **class**.

```
class PopupMenuButtonWidget extends StatelessWidget implements PreferredSizeWidget {
  const PopupMenuButtonWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.lightGreen.shade100,
      height: preferredSize.height,
      width: double.infinity,
      child: Center(
        //child: Text('Bottom'),
        child: PopupMenuButton<TodoMenuItem>(
          icon: Icon(Icons.view_list),
          onPressed: ((valueSelected) {
            print('valueSelected: ${valueSelected.title}');
          }),
        ),
      ),
    );
  }
}
```

```

itemBuilder: (BuildContext context) {
  return foodMenuList.map((TodoMenuItem todoMenuItem) {
    return PopupMenuItem<TodoMenuItem>(
      value: todoMenuItem,
      child: Row(
        children: <Widget>[
          Icon(todoMenuItem.icon.icon),
          Padding(
            padding: EdgeInsets.all(8.0),
          ),
          Text(todoMenuItem.title),
        ],
      ),
    );
  }).toList();
},
),
);
}

@override
// implement preferredSize
Size get preferredSize => Size.fromHeight(75.0);
}

```


Adding **Buttons** as Widget Classes

1. **Add** the widget **class** names **ButtonsWidget()** and **ButtonBarWidget()** to the **Column** children widget list. The **Column** is located in the **body property**. Add a **Divider()** widget between each widget class name. Make sure each widget class uses the **const** keyword.

```
body: Padding(  
  padding: EdgeInsets.all(16.0),  
  child: SafeArea(  
    child: SingleChildScrollView(  
      child: Column(  
        children: <Widget>[  
          //ContainerWithBoxDecorationWidget  
          const ContainerWithBoxDecorationWidget(),  
          Divider(),  
          //ColumnWidget,  
          const ColumnWidget(),  
          Divider(),  
          //RowWidget,  
          const RowWidget(),  
          Divider(),  
          //ColumnAndRowNestingWidget,  
          const ColumnAndRowNestingWidget(),  
          Divider(),  
          //ButtonsWidget,  
          const ButtonsWidget(),  
          Divider(),  
          //ButtonBarWidget,  
          const ButtonBarWidget(),  
        ],  
      ),  
    ),  
  ),  
)
```

2. Create the **ButtonsWidget()** widget class after the **ColumnAndRowNestingWidget()** widget class. The class returns a **Column** with three **Row** widgets for the children list of Widget. Each Row children list of Widget contains different buttons such as the **TextButton**, **ElevatedButton**, and **IconButton** buttons.

```
class ButtonsWidget extends StatelessWidget {
  const ButtonsWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Row(
          children: <Widget>[
            Padding(padding: EdgeInsets.all(16.0)),
            TextButton(
              onPressed: () {},
              child: Text('Flag'),
              style: TextButton.styleFrom(primary: Colors.black),
            ),
            Padding(padding: EdgeInsets.all(16.0)),
            TextButton(
              onPressed: () {},
              child: Icon(Icons.flag),
              style: TextButton.styleFrom(primary: Colors.lightGreen),
              //colors: Colors.lightGreen,
              //textColor: Colors.white,
            ),
          ],
        ),
      ],
    ),
  ),
}
```

```

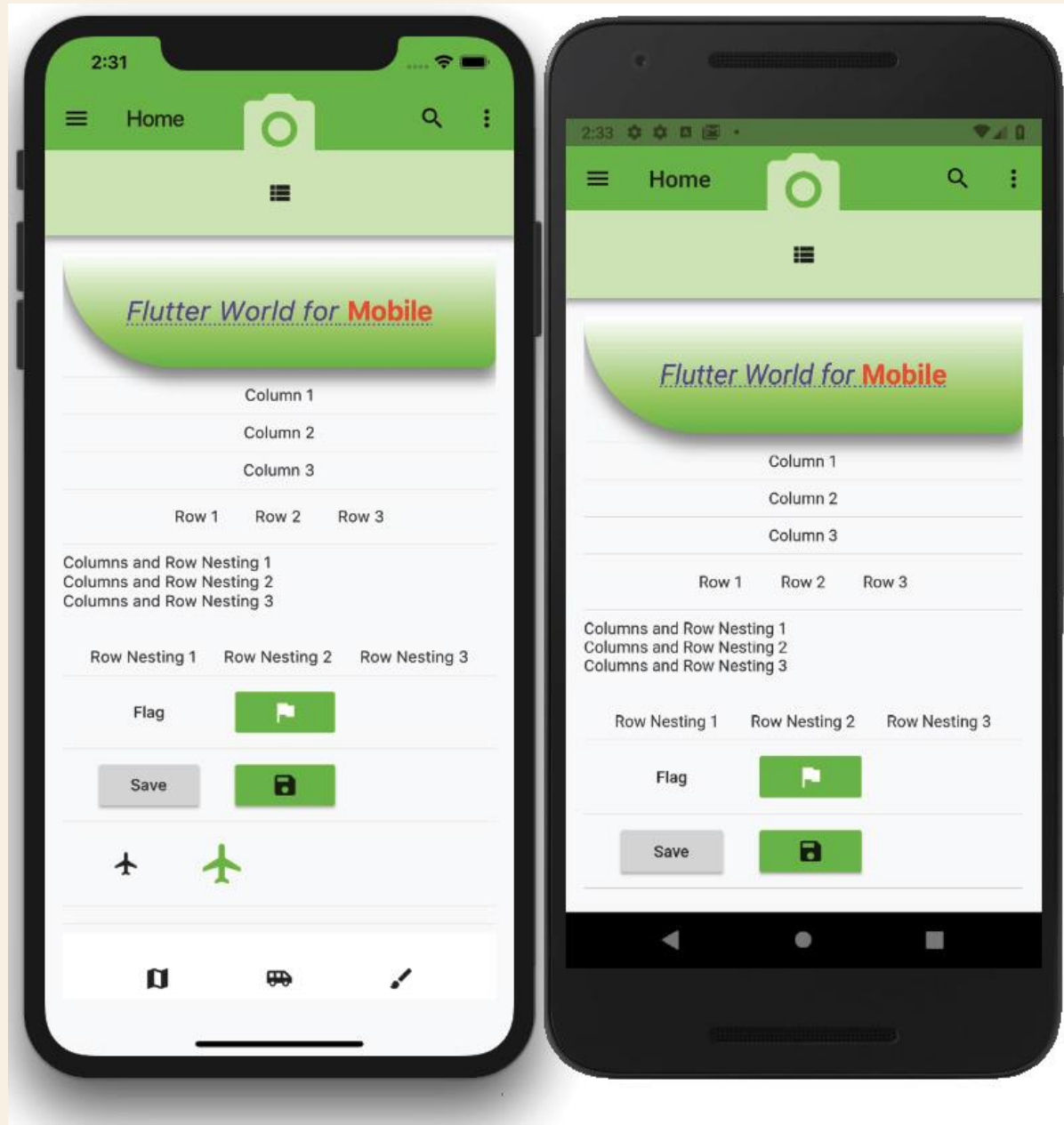
Divider(),
  Row(
    children: <Widget>[
      Padding(padding: EdgeInsets.all(16.0)),
      ElevatedButton(
        onPressed: () {},
        child: Text('Save'),
        style: ElevatedButton.styleFrom(primary: Colors.transparent),
      ),
      Padding(padding: EdgeInsets.all(16.0)),
      ElevatedButton(
        onPressed: () {},
        child: Icon(Icons.save),
        style: ElevatedButton.styleFrom(primary: Colors.lightGreen),
      ),
    ],
  ),
Divider(),
Row(
  children: <Widget>[
    Padding(padding: EdgeInsets.all(16.0)),
    IconButton(
      icon: Icon(Icons.flight),
      onPressed: () {},
    ),
    Padding(padding: EdgeInsets.all(16.0)),
    IconButton(
      icon: Icon(Icons.flight),
      iconSize: 42.0,
      color: Colors.lightGreen,
      tooltip: 'Flight',
      onPressed: () {},
    ),
  ],
),
Divider(),
],
);
}

```

3. Create the `ButtonBarWidget()` widget class after the `ButtonsWidget()` widget class. The class returns a `Container` with a `ButtonBar` as a child.

```
class ButtonBarWidget extends StatelessWidget {
  const ButtonBarWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.white70,
      child: ButtonBar(
        alignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          IconButton(
            icon: Icon(Icons.map),
            onPressed: () {},
          ),
          IconButton(
            icon: Icon(Icons.airport_shuttle),
            onPressed: () {},
          ),
          IconButton(
            icon: Icon(Icons.brush),
            highlightColor: Colors.purple,
            onPressed: () {},
          ),
        ],
      ),
    );
  }
}
```



```

return MaterialApp(
  debugShowCheckedModeBanner: false,
  home: Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.blue,
      title: Text('Flutter TextButton Example'),    ),
    body: Center(
      child: Column(children: <Widget>[
        Container(
          margin: EdgeInsets.all(25),
          child: TextButton(
            child: Text( 'SignUp',
              style: TextStyle(fontSize: 20.0),    ),
            onPressed: () {},    ),
        Container(
          margin: EdgeInsets.all(25),
          child: TextButton(
            style: TextButton.styleFrom(
              backgroundColor: Colors.blueAccent,
              foregroundColor: Colors.white,
              textStyle: const TextStyle(fontSize: 20)),
            child: Text(
              'LogIn',
              style: TextStyle(fontSize: 20.0),
            ),
            onPressed: () {},    ),
      ])),    );  }}

```

```

return MaterialApp(
  debugShowCheckedModeBanner: false,
  home: Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.blue,
      title: Text('Flutter ElevatedButton Example'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            msg,
            style: TextStyle(fontSize: 20, fontStyle: FontStyle.italic),),
          ElevatedButton(
            onPressed: _changeText,
            style: ButtonStyle(
              foregroundColor: MaterialStateProperty.all(Colors.black),
              backgroundColor: MaterialStateProperty.all(Colors.green),
            ),
            child: Text(
              "Click Here",
              style: TextStyle(fontSize: 20), ), ) ], ),), ), ); }

_changeText() {
  setState(() {
    if (msg.startsWith('F')) {
      msg = 'We have learned Flutter Elevated button example.';
    } else {
      msg = 'Flutter ElevatedButton Example';
    }
  });
}

```

```

return MaterialApp(
  debugShowCheckedModeBanner: false,
  home: Scaffold(
    appBar: AppBar(
      title: Text("FAB Button Example"),
      backgroundColor: Colors.blue,
      actions: <Widget>[
        IconButton(icon: Icon(Icons.camera_alt), onPressed: () => {}),
        IconButton(icon: Icon(Icons.account_circle), onPressed: () => {})
      ],
    ),
    floatingActionButton: FloatingActionButton(
      backgroundColor: Colors.green,
      foregroundColor: Colors.white,
      onPressed: () => {},
      child: Icon(Icons.navigation),
    ),
    /*floatingActionButton:FloatingActionButton.extended(
      onPressed: () {},
      icon: Icon(Icons.save),
      label: Text("Save"),
    ), */
  );
}

```



```

return MaterialApp(
  home: Scaffold(
    appBar: AppBar(
      title: const Text("Icon Button Example"), ),
    body: Center(
      child: MyStatefulWidget(), ),),);}}
double _speakervolume = 0.0;
class MyStatefulWidget extends StatefulWidget {
  MyStatefulWidget({Key? key}) : super(key: key);
  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: <Widget>[
        IconButton(
          icon: const Icon(Icons.volume_up),
          iconSize: 50,
          color: Colors.brown,
          tooltip: 'Increase volume by 5',
          onPressed: () {
            setState(() {
              _speakervolume += 5; }); }, ),
        Text('Speaker Volume: $_speakervolume') ], ); }}

```

```

int _volume = 0;
@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    home: Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blueAccent,
        title: Text('InkWell Button Example'), ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            InkWell(
              splashColor: Colors.green,
              highlightColor: Colors.blue,
              child: Icon(Icons.ring_volume, size: 50),
              onTap: () {
                setState(() {
                  _volume += 2; }); },),
            Text(_volume.toString(), style: TextStyle(fontSize: 50)), ],
          ),
        ),
      ),
    );
}
}

```

```

import 'package:flutter/material.dart';
void main() { runApp(MyApp());}
class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();}
class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          backgroundColor: Colors.blueAccent,
          title: Text('Outline Button Example'), ),
        body: Center( child: Column( children: <Widget>[
          Container(
            margin: EdgeInsets.all(25),
            child: OutlinedButton(
              style: OutlinedButton.styleFrom(
                shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(10.0), ),
                side: const BorderSide(
                  width: 5.0,
                  color: Colors.blue, ), ),
              onPressed: () {},
              child: Text( "Outline Button",
style: TextStyle(fontSize: 20.0), ),),),

```

```

Container(
  margin: EdgeInsets.all(25),
  child: TextButton(
    child: Text('Text Button',
style: TextStyle(fontSize: 20.0),),
    style: TextButton.styleFrom(
      foregroundColor: Colors.blue,
    ),
    onPressed: () {},
  ),
),
],
),
),
),
);
}
}

```

```
class MyApp extends StatefulWidget {
  @override
  _State createState() => _State();}
class _State extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blueAccent,
        title: Text('Flutter ButtonBar Example'), ),
      body: Padding(
        padding: EdgeInsets.all(10),
        child: Column(
          children: <Widget>[
            Padding(
              padding: EdgeInsets.all(15),
              child: ButtonBar(
                mainAxisSize: MainAxisSize.min,
                children: <Widget>[
                  ElevatedButton(
                    child: new Text('Dart'),
                    style: ButtonStyle(
                      backgroundColor: MaterialStatePropertyAll( Colors.lightGreen, ), ),
                    onPressed: () {/** */}, ),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```
TextButton(  
  child: Text('Flutter'),  
  style: ButtonStyle(  
    backgroundColor: MaterialStatePropertyAll( Colors.lightGreen,  
      ),  
    ),  
    onPressed: () {/** */},  
  ),  
  TextButton(  
    child: Text('MySQL'),  
    style: ButtonStyle(  
      backgroundColor: MaterialStatePropertyAll( Colors.lightGreen,  
        ),  
      ),  
      onPressed: () {/** */},  
    ),  
  ],  
),  
),  
],  
)))  
}}
```