Name : Akanksha Shinde   Class: D15C   Roll No: 53

Experiment No. 3

AIM: To include icons, images, fonts in Flutter app

THEORY:

Including icons, images, and custom fonts in a Flutter app is a common requirement to enhance the visual appeal and functionality of the application. Let's discuss each of these elements:

1. Icons:

Icons in Flutter are typically represented by the Icon widget. Flutter provides a set of built-in icons through the Icons class, but you can also use custom icons or those from external icon packs.

How to Include Icons:

Built-in Icons:

```
Icon(Icons.star);
```

Custom Icons:

Flutter allows you to use custom icons in various formats, such as SVG or PNG. You can use the Image.asset or Image.network widget to display custom icons.

2. Images:

Displaying images is a crucial part of app development. Flutter supports various image formats, including PNG, JPEG, GIF, and WebP.

How to Include Images:

Asset Images:

- Place your images in the assets folder of your project, and then use the Image.asset widget to display them.

```
Image.asset('assets/images/my_image.png');
```

3. Fonts:

Custom fonts allow you to create a unique typographic style for your app. Flutter supports TrueType (TTF) and OpenType (OTF) fonts.

How to Include Fonts:

Adding Fonts to pubspec.yaml:

Add your font files to the fonts section of the pubspec.yaml file.

```
flutter:

  fonts:
    - family: MyCustomFont
      fonts:
        - asset: assets/fonts/my_custom_font.tt
```

Using Custom Fonts:

Apply custom fonts using the Text widget.

```
Text(

  'Hello, World!',
  style: TextStyle(
    fontFamily: 'MyCustomFont',
    fontSize: 20,
  ),
);
```

CODE:

```
// pages/cleaning.dart
import 'package:animate_do/animate_do.dart';
import 'package:day35/pages/date_time.dart';
import 'package:flutter/material.dart';

class CleaningPage extends StatefulWidget {
  const CleaningPage({ Key? key }) : super(key: key);

  @override
  _CleaningPageState createState() => _CleaningPageState();
}

class _CleaningPageState extends State<CleaningPage> {
  // Rooms to clean
  List<dynamic> _rooms =[
    ['Living Room', 'https://img.icons8.com/officel/2x/living-room.png', Colors.red, 0],
    ['Bedroom', 'https://img.icons8.com/fluency/2x/bedroom.png', Colors.orange, 1],
    ['Bathroom', 'https://img.icons8.com/color/2x/bath.png', Colors.blue, 1],
    ['Kitchen', 'https://img.icons8.com/dusk/2x/kitchen.png', Colors.purple, 0],
    ['Office', 'https://img.icons8.com/color/2x/office.png', Colors.green, 0]
  ];

  List<int> _selectedRooms = [];
```

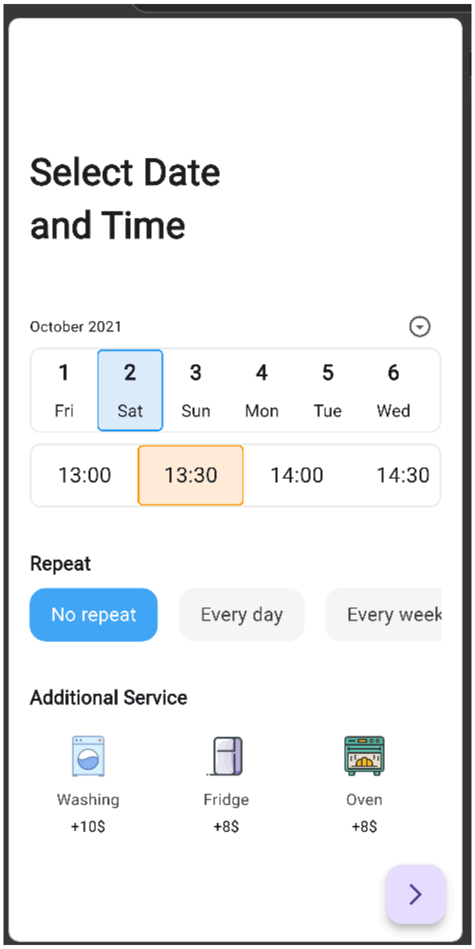
```
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   backgroundColor: Colors.white,
   floatingActionButton: _selectedRooms.length > 0 ? FloatingActionButton(
    onPressed: () {
     Navigator.push(
      context,
      MaterialPageRoute(
       builder: (context) => DateAndTime()
      ),
     );
    },
    child: Row(
     mainAxisAlignment: MainAxisAlignment.center,
     children: [
      Text('${_selectedRooms.length}', style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
      SizedBox(width: 2),
      Icon(Icons.arrow_forward_ios, size: 18,),
     ],
    ),
    backgroundColor: Colors.blue,
   ) : null,
   body: NestedScrollView(
    headerSliverBuilder: (BuildContext context, bool innerBoxIsScrolled) {
     return <Widget>[
      SliverToBoxAdapter(
       child: FadeInUp(child: Padding(
        padding: EdgeInsets.only(top: 120.0, right: 20.0, left: 20.0),
        child: Text(
         'Where do you want \ncleaned?',
         style: TextStyle(
          fontSize: 35,
          color: Colors.grey.shade900,
          fontWeight: FontWeight.bold,
         ),
        ),
       ),
      ))
     ];
    },
    body: Padding(
     padding: EdgeInsets.all(20.0),
     child: ListView.builder(
      physics: NeverScrollableScrollPhysics(),
      itemCount: _rooms.length,
      itemBuilder: (BuildContext context, int index) {
       return FadeInUp(
        delay: Duration(milliseconds: 500 * index),
        duration: Duration(milliseconds: 500),
       child: room(_rooms[index], index));
      }
     ),
    ),
   )
  );
 }

 room(List room, int index) {
  return GestureDetector(
   onTap: () {
    setState(() {
     if (_selectedRooms.contains(index))
      _selectedRooms.remove(index);
     else
      _selectedRooms.add(index);
    });
   },
   child: Container(
    padding: EdgeInsets.symmetric(horizontal: 20.0, vertical: 10),
    margin: EdgeInsets.only(bottom: 20.0),
    decoration: BoxDecoration(
     borderRadius: BorderRadius.circular(10.0),
     color: _selectedRooms.contains(index) ? room[2].shade50.withOpacity(0.5) :
Colors.grey.shade100,
    ),
    child: Column(
     mainAxisAlignment: MainAxisAlignment.start,
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
      Row(
       children: [
        Row(
         children: [
          Image.network(room[1], width: 35, height: 35,),
          SizedBox(width: 10.0,),
          Text(room[0], style: TextStyle(fontSize: 18, fontWeight: FontWeight.w600),),
         ],
        ),
        Spacer(),
        _selectedRooms.contains(index) ?
```

```
                  Container(
                    padding: EdgeInsets.all(5.0),
                    decoration: BoxDecoration(
                      color: Colors.greenAccent.shade100.withOpacity(0.3),
                      borderRadius: BorderRadius.circular(10.0),
                    ),
                    child: Icon(Icons.check, color: Colors.green, size: 20,)
                  ) :
                  SizedBox()
              ],
            ),
            (_selectedRooms.contains(index) && room[3] >= 1) ?
            Container(
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  SizedBox(height: 20.0,),
                  Text("How many ${room[0]}s?", style: TextStyle(fontSize: 15),),
                  SizedBox(height: 10.0,),
                  Container(
                    height: 45,
                    child: ListView.builder(
                      scrollDirection: Axis.horizontal,
                      itemCount: 4,
                      itemBuilder: (BuildContext context, int index) {
                        return GestureDetector(
                          onTap: () {
                            setState(() {
                              room[3] = index + 1;
                            });
                          },
                          child: Container(
                            margin: EdgeInsets.only(right: 10.0),
                            padding: EdgeInsets.all(10.0),
                            width: 50,
                            decoration: BoxDecoration(
                              borderRadius: BorderRadius.circular(10.0),
                              color: room[3] == index + 1 ? room[2].withOpacity(0.5) :
room[2].shade200.withOpacity(0.5),
                            ),
                            child: Center(child: Text((index + 1).toString() , style: TextStyle(fontSize: 22,
color: Colors.white),)),
                          ),
                        );
                      }
                    ),
                  )
                ],
              ),
            ) : SizedBox()
              )
         ),  );
      }
}
```

//pubspec.yaml

```
-         name: day35
-         description: A new Flutter project.
-         publish_to: 'none'
-         version: 1.0.0+1
-
-         environment:
-           sdk: ">=3.5.1"
-
-         dependencies:
-           flutter:
-             sdk: flutter
-           cupertino_icons: ^1.0.8
-           animate_do: ^3.3.4
-           scrollable_positioned_list: ^0.3.8
-
-         dev_dependencies:
-           flutter_test:
-             sdk: flutter
-
-         flutter:
-           uses-material-design: true
```

OUTPUT



CONCLUSION:

In conclusion, enhancing the visual appeal and functionality of a Flutter app involves incorporating icons, images, and custom fonts.

Icons: Utilize the Icon widget for built-in icons and explore custom icons using various formats.
Images: Display images with the Image.asset widget for local assets or Image.network for network images.

Fonts: Include custom fonts in the pubspec.yaml file and apply them using the TextStyle property in the Text widget.