

EXPERIMENT NO:3

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

Theory:

Including icons, images, and custom fonts in a Flutter app allows developers to enhance the visual appeal and functionality of their applications. Here's a brief overview of how to include these assets:

1. Icons:

- a. Flutter provides built-in support for icons through the Icons class, which includes a wide range of Material Design icons.
- b. You can use the Icon widget to display icons in your app. Simply specify the desired icon using the Icons class, along with properties like size and color.

2. Images:

- a. To include images in a Flutter app, you can add image files to the assets directory within your project.
- b. Use the Image widget to display images. Specify the image asset path using the Image.asset() constructor.

3. Fonts:

- a. Custom fonts can be added to a Flutter app by including font files (e.g., .ttf or .otf) in the project's fonts directory.
- b. Declare the custom fonts in the pubspec.yaml file under the flutter section using the fonts property.
- c. Once declared, you can apply the custom font to text in your app using the fontFamily property in the TextStyle widget.

Here's a summarized step-by-step guide:

1. Add Icons:

- a. Use the Icon widget with the desired icon from the Icons class.
- b. Customize the icon size and color as needed.

2. Add Images:

- a. Place image files in the assets directory of your Flutter project.
- b. Use the Image.asset() widget to load images from the asset bundle.
- c. Specify the image asset path as a parameter to the Image.asset() constructor.

3. Add Fonts:

- a. Place custom font files in the fonts directory of your Flutter project.

- b. Declare the custom fonts in the pubspec.yaml file under the flutter section using the fonts property.
- c. Apply the custom font to text using the fontFamily property in the TextStyle widget

CODE:

```
import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'ToDo List',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: ToDoListPage(),
    );
  }
}

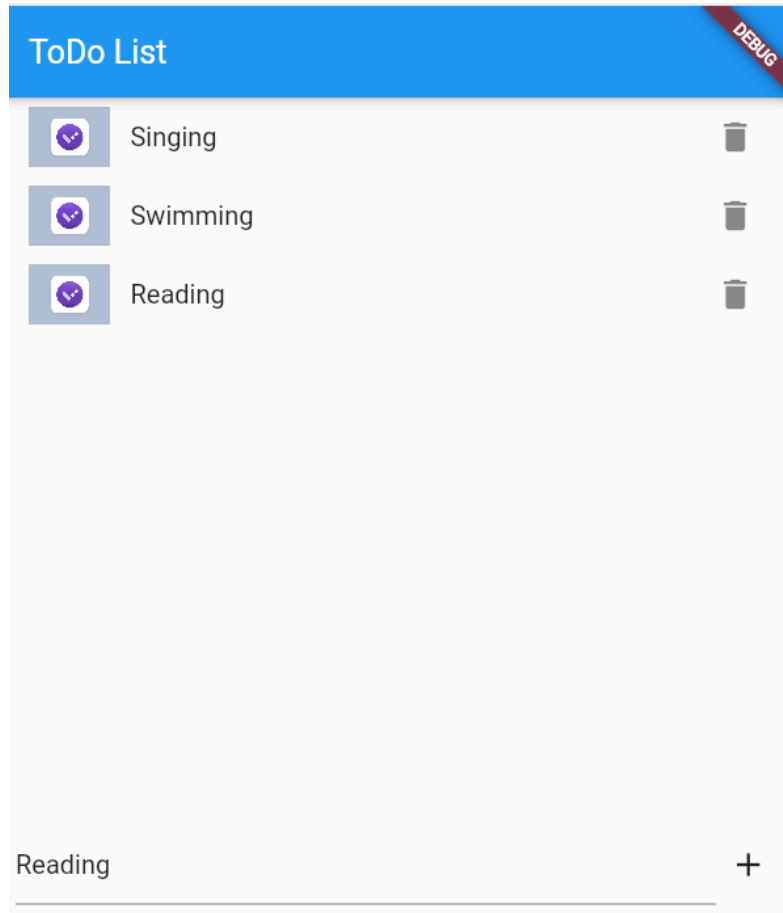
class ToDoListPage extends StatefulWidget {
  @override
  _ToDoListPageState createState() => _ToDoListPageState();
}

class _ToDoListPageState extends State<ToDoListPage> {
  List<String> tasks = [];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('ToDo List'),
      ),
      body: Column(
        children: <Widget>[
          Expanded(
            child: ListView.builder(
              itemCount: tasks.length,
```

```
itemBuilder: (context, index) {  
    return ListTile(  
        title: Text(tasks[index]),  
        leading: tasks.isNotEmpty && tasks.length > index  
            ? Image.asset(  
                'assets/ToDoList.jpg', // Adjust the path to your image  
                width: 50,  
                height: 50,  
            )  
            : Container(), // Placeholder for when the image is not available  
        trailing: IconButton(  
            icon: Icon(Icons.delete),  
            onPressed: () {  
                setState(() {  
                    tasks.removeAt(index);  
                });  
            },  
        ),  
    );  
},  
),  
Padding(  
    padding: const EdgeInsets.all(8.0),  
    child: Row(  
        children: <Widget>[  
            Expanded(  
                child: TextField(  
                    decoration: InputDecoration(  
                        hintText: 'Enter a task',  
                    ),  
                onPressed: (value) {  
                    setState(() {  
                        tasks.add(value);  
                    });  
                },  
            ),  
        ],  
    ),  
    IconButton(  
        icon: Icon(Icons.add),  
        onPressed: () {
```

```
),  
  ],  
),  
),  
  ],  
),  
);
```



CONCLUSION: In this lab we have implemented included icons, images, fonts in Flutter app.