

**Name:** Vidhi Gulhane  
**Roll No:** 15  
**Class:** D15B

## Experiment No: 03

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

### Theory:

Flutter allows developers to enhance their app's UI by incorporating icons, images, and custom fonts. These elements help in improving the app's aesthetics and user experience.

### 1. Including Icons:

Flutter provides built-in Material Icons and supports custom icon sets.

Using Material Icons:

```
Icon(Icons.home, size: 30, color: Colors.blue);
```

Using Custom Icons (from packages like FontAwesome):

```
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
```

```
Icon(FontAwesomeIcons.heart, size: 30, color: Colors.red);
```

### 2. Adding Images:

Images can be added from local assets or loaded from a network.

Adding Local Images:

Place images in the assets folder.

Define them in pubspec.yaml:

yaml

flutter:

assets:

- assets/images/logo.png

Use Image.asset in the app:

```
Image.asset('assets/images/logo.png', width: 100, height: 100);
```

Using Network Images:

```
Image.network('https://example.com/image.png', width: 100, height: 100);
```

### 3. Adding Custom Fonts:

Flutter allows integrating custom fonts for a unique design.

Steps to add a custom font:

Place font files in the assets/fonts folder.

Define them in pubspec.yaml:

yaml

flutter:

  fonts:

    - family: CustomFont

      fonts:

        - asset: assets/fonts/CustomFont-Regular.ttf

Use the custom font in your app:

Text(

  "Hello Flutter",

  style: TextStyle(fontFamily: "CustomFont", fontSize: 20),

);

**Code:**

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

```
import 'package:google_fonts/google_fonts.dart';
```

```
import 'recipe_screen.dart';
```

```
class HomeScreen extends StatelessWidget {
```

```
  final List<Map<String, dynamic>> recipes = [
```

```
  {
```

```
    "name": "Pasta Carbonara",
```

```
    "details": "A delicious Italian pasta with creamy sauce.",
```

```
    "image": "lib/assets/recipe1.jpg",
```

```
    "procedure":
```

```
    "1. Boil pasta.\n2. Cook bacon.\n3. Mix eggs and cheese.\n4. Combine everything and serve.",
```

```
  },
```

```
  {
```

```
    "name": "Spicy Tacos",
```

```
    "details": "Mexican-style tacos filled with spice and flavor.",
```

```
    "image": "lib/assets/recipe2.jpg",
```

```

"procedure":
"1. Cook meat with spices.\n2. Prepare toppings.\n3. Assemble tacos.\n4. Serve and enjoy.",
},
{
"name": "Chocolate Cake",
"details": "Rich and moist chocolate cake for dessert lovers.",
"image": "lib/assets/recipe3.jpg",
"procedure":
"1. Mix dry ingredients.\n2. Add wet ingredients.\n3. Bake at 350°F for 30 minutes.\n4. Let cool and
serve.",
},
];
HomeScreen({super.key});

@override
Widget build(BuildContext context) {
return Scaffold(
  appBar: AppBar(
    title: Text("Recipe App", style: GoogleFonts.pacifico(fontSize: 24)),
    backgroundColor: Colors.orangeAccent,
  ),
  body: Stack(
    children: [
      Container(
        decoration: BoxDecoration(
          image: DecorationImage(
            image: AssetImage('lib/assets/background1.jpg'),
            fit: BoxFit.cover,
          ),
        ),
      ),
    ],
  ),
);

```



```

child: Image.asset(
  recipe["image"],
  width: double.infinity,
  height: 200,
  fit: BoxFit.cover,
),
),
Padding(
  padding: const EdgeInsets.all(10.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        recipe["name"],
        style: GoogleFonts.roboto(
          fontSize: 18,
          fontWeight: FontWeight.bold,
        ),
      ),
      SizedBox(height: 5),
      Text(
        recipe["details"],
        style: GoogleFonts.roboto(
          fontSize: 14,
          color: Colors.black54,
        ),
      ),
      SizedBox(height: 10),
      Align(

```

```

alignment: Alignment.centerRight,

child: TextButton(

  onPressed: () {

    Navigator.push(

      context,

      MaterialPageRoute(

        builder:

          (context) => RecipeScreen(

            recipeName: recipe["name"],

            recipeDetails: recipe["details"],

            procedure: recipe["procedure"],

            nutritionInfo: {},

            processSteps: ),),

        child: Text(

          "View Recipe",

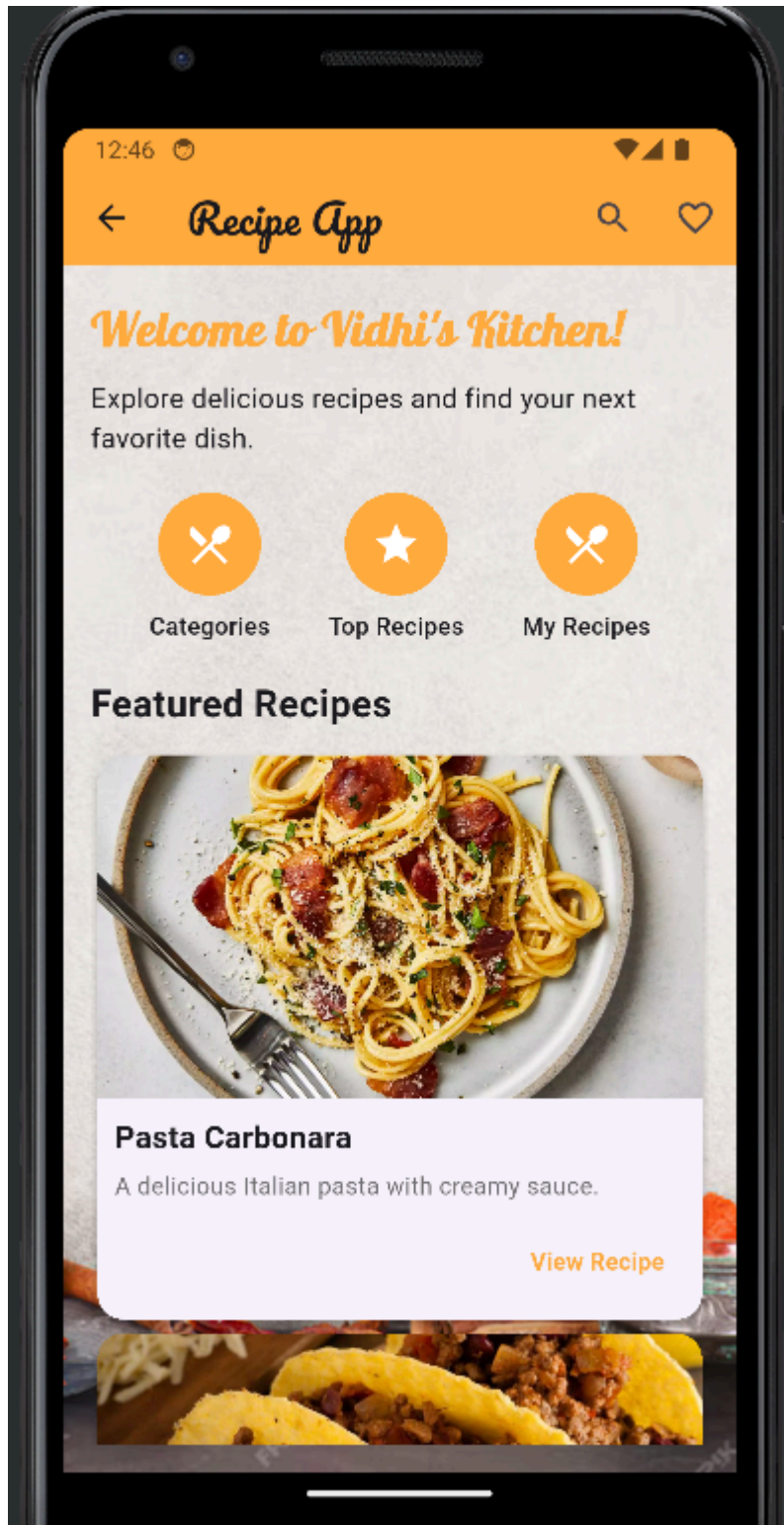
          style: TextStyle(

            color: Colors.orangeAccent,

            fontWeight: FontWeight.bold,)),),],

```

**Output:**



## Conclusion:

By including icons, images, and fonts, we can enhance the visual appeal and usability of a Flutter app. Using built-in Material icons, external icon libraries, asset images, and custom fonts ensures a polished and professional UI. This experiment highlights the flexibility of Flutter in supporting various multimedia elements, making it a powerful tool for mobile app development.

