

Experiment No:03

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

THEORY:

Incorporating icons, images, and custom fonts into a Flutter application is essential for enhancing both its visual appeal and functionality. These assets play a crucial role in creating a rich and engaging user experience. Here's a comprehensive guide on how to integrate these assets into your Flutter project:

Icons:

Icons are an integral part of any user interface, providing visual cues and enhancing usability. Flutter offers built-in support for icons through the `Icons` class, which includes a vast collection of Material Design icons. Here's how to include icons in your Flutter app:

1. Using the Icon Widget: Utilize the `Icon` widget to display icons within your app. Simply specify the desired icon using the `Icons` class, along with properties like size and color.
2. Customization: Customize the icon's appearance by adjusting properties such as size, color, and alignment to suit your app's design language and theme.

Images:

Images are essential for conveying information and adding visual interest to your app's user interface. Flutter provides various options for including images in your project. Here's how to integrate images into your Flutter app:

1. Asset Management: Place image files in the ``assets`` directory of your Flutter project. This directory serves as a centralized location for managing your app's image assets.

2. Image Widget: Use the ``Image.asset()`` widget to load images from the asset bundle. Specify the image asset path as a parameter to the ``Image.asset()`` constructor.

3. Customization: Customize image display properties such as width, height, alignment, and scaling to ensure optimal presentation within your app.

Fonts:

Custom fonts allow you to add a unique typographic style to your app's text elements, enhancing readability and visual appeal. Flutter supports the integration of custom fonts, enabling developers to create distinct typographic identities for their apps. Here's how to incorporate custom fonts into your Flutter project:

1. Font Integration: Place custom font files (e.g., ``.ttf`` or ``.otf`` files) in the ``fonts`` directory of your Flutter project.

2. Declaration in `pubspec.yaml`: Declare custom fonts in the ``pubspec.yaml`` file under the ``flutter`` section using the ``fonts`` property. This informs Flutter about the existence and location of custom font files in your project.

3. Application to Text: Apply custom fonts to text elements in your app by specifying the desired font family using the ``fontFamily`` property in the ``TextStyle`` widget.

By following these steps, you can seamlessly integrate icons, images, and custom fonts into your Flutter app, thereby enhancing its visual appeal and user experience. These assets play a vital role in creating engaging and immersive mobile applications that captivate users and leave a lasting impression.

Code:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Icons, Images, and Fonts Example',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyHomePage(),
    );
  }
}
```

```
class MyHomePage extends StatelessWidget {  
  
  @override  
  
  Widget build(BuildContext context) {  
  
    return Scaffold(  
  
      appBar: AppBar(  
  
        title: Text('Expt-3'),  
  
      ),  
  
      body: Center(  
  
        child: Column(  
  
          mainAxisAlignment: MainAxisAlignment.center,  
  
          children: <Widget>[  
  
            Icon(  
  
              Icons.person, // Added Icon widget  
  
              size: 50,  
  
              color: Colors.blue,  
  
            ),  
  
            Text(  
  
              'Payoshni Khekale 29',  
  
              style: TextStyle(  
  
                fontSize: 60,  
  
                color: Colors.black,  
  
              ),  
  
            ),  
  
          ],  
  
        ),  
  
      ),  
  
    ),  
  
  ),  
  
}
```

```
      SizedBox(height: 20),  
      Image.asset(  
        'lib/assets/image.png',  
        width: 100,  
        height: 100,  
      ),  
      SizedBox(height: 20),  
      Text(  
        'Hey, have a nice day',  
        style: TextStyle(  
          fontFamily: 'Roboto',  
          fontSize: 36,  
          fontWeight: FontWeight.bold,  
        ),  
      ),  
    ],  
  ),  
),  
);  
}
```

Pubsec.yaml:

assets:

- lib/assets/image.png

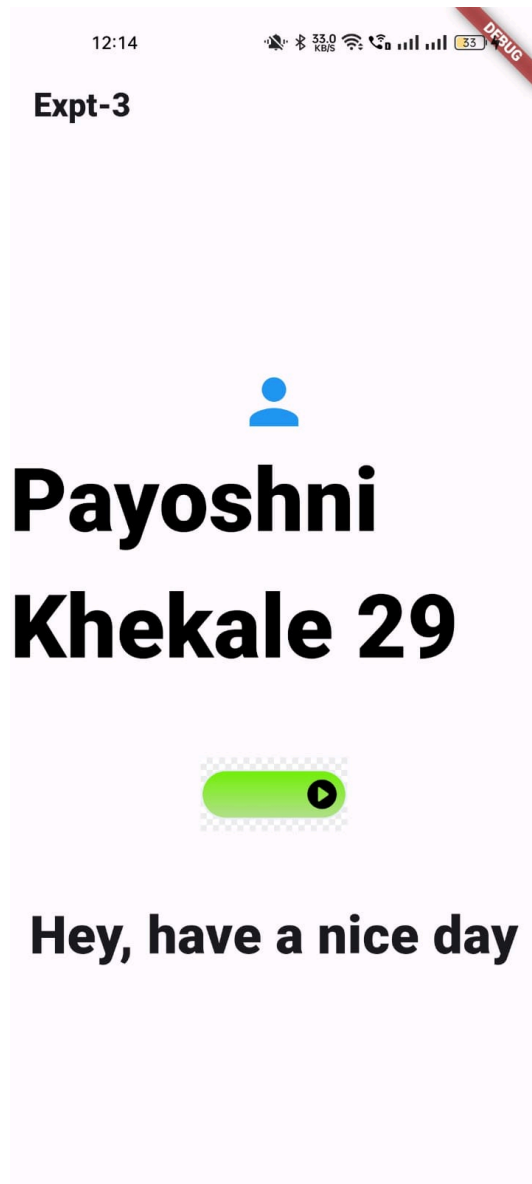
fonts:

- family: Roboto

fonts:

- asset: lib/fonts/Roboto-Black.ttf

Output:



Conclusion:

I have successfully understood and implemented the images , fonts and Icons in a Flutter Application.

