Sairam Konar D15C 27 2024-2025

Aim: To include icons, fonts in Flutter app

Theory:

Flutter is an open-source UI toolkit developed by Google for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. In this experiment, we explore three key UI elements:

- **Icons** – used to represent actions or content visually.
- **Images** – used to enhance visual appeal and deliver information graphically.
- **Fonts** – used to style the text to match branding and improve readability.

Each of these components plays a vital role in enhancing user interface and user experience.

Icons in Flutter

Icons are visual representations of commands, files, directories, devices, or common actions. They help users quickly understand functionality without relying on text labels.

Flutter provides a rich set of **Material Design Icons** out of the box through the Icons class. These are scalable vector icons which can be resized, colored, or styled as needed.

Types of Icons:

- **Material Icons**: Predefined and bundled with Flutter.
- **Custom Icon Fonts**: Imported manually or using external packages like font_awesome_flutter.

Basic Syntax for Using Icons:

```
Icon(Icons.icon_name)
```

Example:

```
Icon(Icons.home)
```

With styling:

```
Icon(Icons.person, color: Colors.teal, size: 30)
```

Fonts in Flutter

Fonts determine the style and appearance of textual content in the app. Using custom fonts in your Flutter app gives you more control over branding and presentation.

Default Font:

- Flutter uses **Roboto** as its default font.

Custom Fonts:

You can import and use any .ttf or .otf font in your Flutter project by following these steps:

Step 1: Add font files Place your font files (e.g., Poppins-Regular.ttf) in the directory: assets/fonts/

Step 2: Declare in pubspec.yaml

```
fonts:
  - family: Poppins
    fonts:
      - asset: assets/fonts/Poppins-Regular.ttf
```

Step 3: Use the font in your Flutter app Apply the font globally in ThemeData:

```
theme: ThemeData(
  fontFamily: 'Poppins',
)
```

Or apply it locally in a Text widget:

```
Text("Hello", style: TextStyle(fontFamily: 'Poppins'))
```

Images in Flutter

Images are used to visually convey information, decorate the app, or present media. Flutter allows you to use images from assets, files, or the network.

Types of Images:

- **Asset Images**: Stored in the app's asset directory.
- **Network Images**: Loaded from a web URL.
- **File Images**: Loaded from the device storage.

How to Use Asset Images:

Step 1: Place the image in your assets folder
Example path: assets/images/sample.jpg

Step 2: Declare the image in pubspec.yaml

```
assets:
  - assets/images/sample.jpg
```

Step 3: Use it in your code

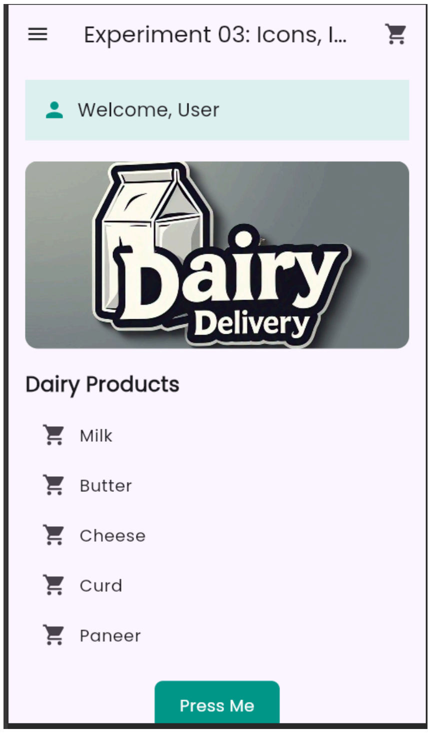
```
Image.asset("assets/images/sample.jpg")
```

For styled use with decorations:

```
Container(
  height: 180,
  width: double.infinity,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(12),
    image: DecorationImage(
      image: AssetImage("assets/images/sample.jpg"),
      fit: BoxFit.cover,
    ),
  ),
)
```

Code (Github): https://github.com/Sairam-Vk-sudo/mplExp27/tree/main/exp%203

Output:



Icons used (Cart, User) and Image

Conclusion: In conclusion, this experiment provided a comprehensive understanding of how to integrate icons, images, and custom fonts in a Flutter application. Icons enhance user interaction by visually representing actions and navigation elements, making the interface more intuitive. Custom fonts allow developers to personalize the appearance of text, aligning the design with branding requirements and improving readability. Images, on the other hand, add visual appeal and context, contributing to a richer user experience. By combining these elements effectively, developers can create visually engaging and user-friendly mobile applications. This foundational knowledge is essential for building polished, professional Flutter apps.