Experiment No: 3

Name:- Vishal Gori **Batch**: A
Roll No.: 18 **Division**:- D15B

AIM:- To add advanced Flutter UI by including widgets like Image, Fonts, Icons.

THEORY:-

Flutter provides robust mechanisms for working with images, fonts, and icons in your app's user interface. Here's a summary of their functionalities and considerations:

Images:

- **Loading and Displaying:** Use the Image widget to load and display images from various sources like assets, network URLs, or files. Adjust properties like fit, alignment, and opacity for customization.
- **Asset Management:** Store images within your app's assets directory (usually under assets/images/). Flutter automatically handles different screen resolutions and densities.
- **Network Images:** Use the Image.network constructor to directly load images from URLs. Ensure proper internet connectivity and consider caching mechanisms for efficiency.
- **Caching and Performance:** Flutter automatically caches downloaded images. For complex scenarios, explore advanced caching libraries like cached_network_image.

Fonts:

- **Using System Fonts:** Access system fonts available on the device using the Text widget's fontFamily property.
- **Custom Fonts:** Include custom fonts in your app's pubspec.yaml file and integrate them using the GoogleFonts package or by loading font files manually.
- **Font Styling:** Control font properties like size, weight, color, and more using the TextStyle class within the Text widget.

- **Text Layouts and Effects:** Flutter offers rich text editing and layout features. Explore properties like textAlign, overflow, and textSpan for advanced text formatting and effects.

Icons:

- **Material Icons:** Flutter provides built-in access to a vast collection of Material Design icons through the Icons class. Use them with the Icon widget for simple icon display.
- **Custom Icons:** You can create custom vector icons or use icon fonts. Popular packages like flutter_icons and font_awesome_flutter provide diverse icon sets.
- **Icon Styling:** Modify icons' colors, sizes, and other properties directly through the Icon widget's parameters.
- **Animations and Interactions:** Integrate icon animations and interactions using gestures, animations, and state management techniques.

CODE:-

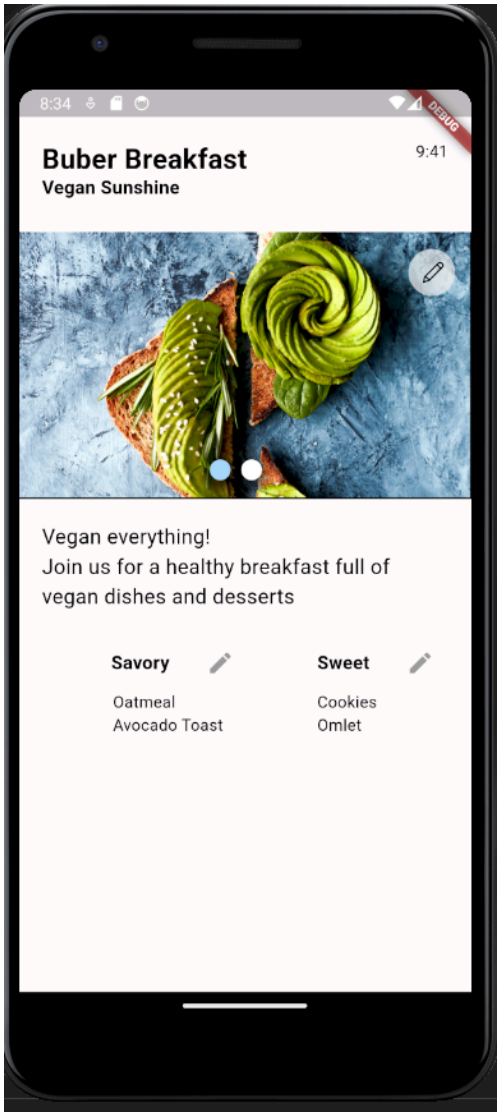
main.dart

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      themeMode: ThemeMode.dark,
      home: Scaffold(
        body: SafeArea(
          child: Column(
            children: [
              Stack(
                children: [
                  Padding(
                    padding: const EdgeInsets.fromLTRB(0, 100, 0, 0),
```

```
          child: Image.asset('assets/images/avocado_toast.jpg'),
        ),
        const Positioned(
          top: 20.0,
          left: 20.0,
          child: Text(
            'Buber Breakfast',
            style: TextStyle(
              color: Colors.black,
              fontSize: 24.0,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
        const Positioned(
          top: 50.0,
          left: 20.0,
          child: Text(
            'Vegan Sunshine',
            style: TextStyle(
              fontWeight: FontWeight.bold,
              color: Colors.black,
              fontSize: 16.0,
            ),
          ),
        ),
        const Positioned(
          top: 20.0,
          right: 20.0,
          child: Text(
            '9:41',
            style: TextStyle(color: Colors.black),
          ),
        ),
      ],
    ),
    const Padding(
      padding: EdgeInsets.all(20.0),
      child: Text(
```

```
                  'Vegan everything!\nJoin us for a healthy breakfast full
of vegan dishes and desserts',
                  style: TextStyle(
                    color: Colors.black,
                    fontSize: 18.0,
                  ),
                ),
              ),
              const Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                  Padding(
                    padding: EdgeInsets.fromLTRB(60, 0, 0, 0),
                    child: Text(
                      'Savory',
                      style: TextStyle(
                        fontFamily: 'Roboto',
                        color: Colors.black,
                        fontSize: 16.0,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                  IconButton(
                      icon: Icon(
                        Icons.edit,
                        color: Colors.grey, // Adjust color as needed
                        // size: 24.0,
                      ),
                      onPressed: null),
                  Padding(
                    padding: EdgeInsets.fromLTRB(40, 0, 0, 0),
                    child: Text(
                      'Sweet',
                      style: TextStyle(
                        fontFamily: 'Roboto',
                        color: Colors.black,
                        fontSize: 16.0,
                        fontWeight: FontWeight.bold,
                      ),
```

```
                    ),
                ),
                IconButton(
                    icon: Icon(
                      Icons.edit,
                      color: Colors.grey, // Adjust color as needed
                      // size: 24.0,
                    ),
                    onPressed: null),
              ],
            ),
            const Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: [
                Text(
                  'Oatmeal\nAvocado Toast',
                  style: TextStyle(color: Colors.black),
                ),
                Text(
                  'Cookies\nOmlet',
                  style: TextStyle(color: Colors.black),
                ),
              ],
            ),
          ],
        ),
      ),
    ),
  );
  }
}
```



CONCLUSION:- In this experiment, we explored the diverse tools Flutter offers for crafting engaging visuals and user experiences.

- **Images**: We learned how to seamlessly integrate images from various sources, ensuring optimal performance and accessibility.
- **Fonts**: We discovered the power of using system and custom fonts, allowing for expressive and personalized text displays.
- **Icons**: We delved into the world of Material Icons and custom icon options, adding interactivity and visual clarity to our interfaces