WEEK-2

a) Explore various Flutter widgets (Text, Image, Container, etc.).

Aim: To explore and understand the use of basic Flutter widgets such as Text, Image, and Container to build a simple user interface.

Description: This activity focuses on learning how to use core Flutter widgets to create a basic UI layout. The Text widget is used to display text on the screen, the Image widget is used to load and show images from the network or local assets, and the Container widget is used for styling and structuring elements with properties like padding, margin, color, and size. Understanding these foundational widgets is essential for building user-friendly and visually appealing Flutter applications**.**

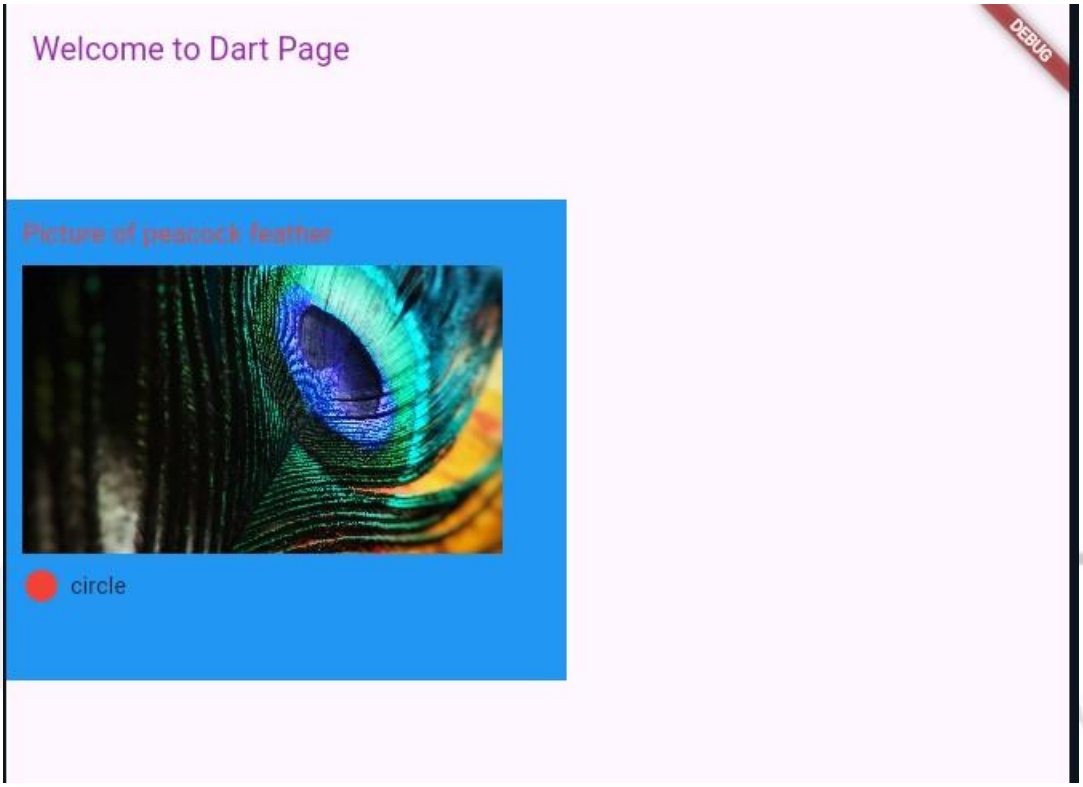
Source Code:

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

void main() => runApp(
  MaterialApp(
    home: Scaffold(
      appBar: AppBar(
       title: Text(
         'Welcome to Dart Page',
         style: TextStyle(
           color: Colors.purple,
           fontSize: 20,
         ),
       ),
      ),
      body: Center(
        child: Column(
          children: [
            Text(
              'Picture of peacock feather',
              style: TextStyle(
                color: Colors.red,
                fontSize: 20,
              ),
            ),
            Image.network(
              'https://images.pexels.com/photos/674010/pexels-photo-674010.jpeg',
              width: 400,
              height: 300,
            ),
            Icon(Icons.circle, color: Colors.red),
            Text('circle'),
          ],
```

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

Output:


Welcome to Dart Page
DEBUG
Picture of peacock feather
circle

b)Implement different layout structures using Row, Column, and Stack widgets.

Aim: To implement and understand different layout structures in Flutter using Row, Column, and Stack widgets for arranging UI elements horizontally, vertically, and in overlapping layers.

Description: This experiment demonstrates how to design and structure user interfaces in Flutter using fundamental layout widgets such as Row, Column, and Stack. The Row widget arranges its children horizontally, the Column widget arranges them vertically, and the Stack widget allows widgets to be placed on top of each other, enabling layered UI designs. By implementing these layout structures, developers can create responsive and organized interfaces, manage widget alignment and positioning effectively, and build complex UIs with ease. This exercise helps in understanding how different layout widgets work together to form visually structured applications in Flutter.

Source Code:

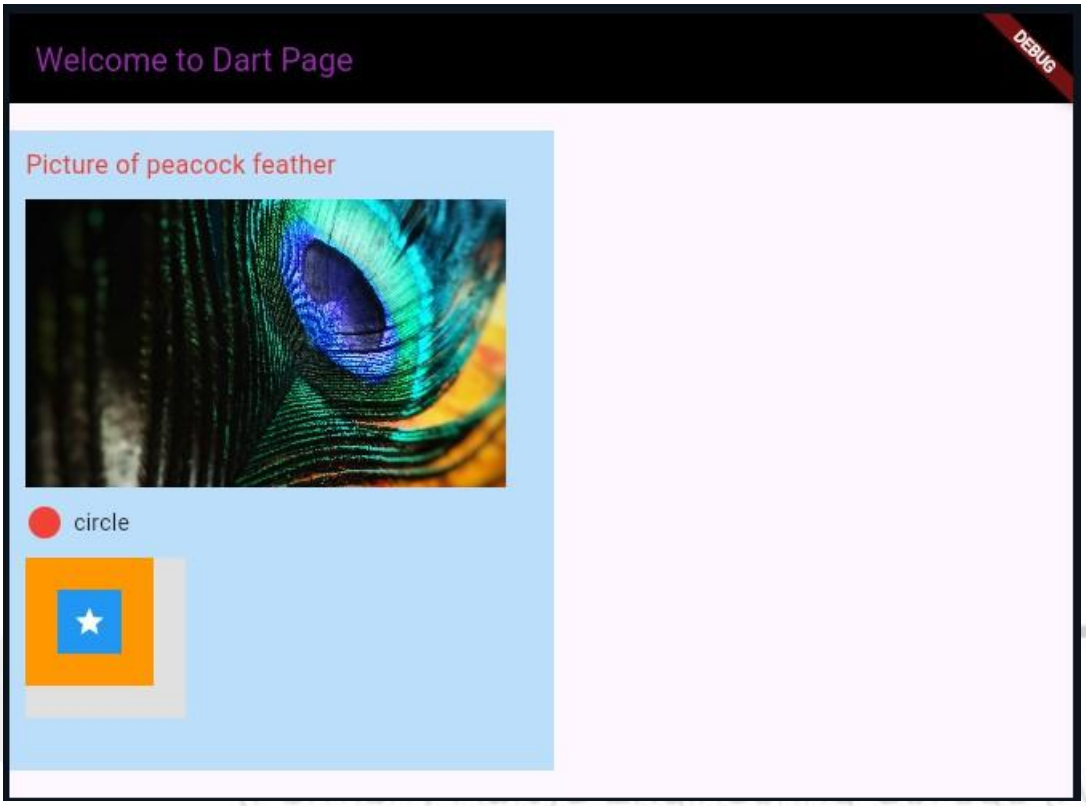
```
import 'package:flutter/material.dart';

void main() => runApp(
  MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text(
          'Welcome to Dart Page',
          style: TextStyle(
            color: Colors.purple,
            fontSize: 20,
          ),
        ),
        backgroundColor: Colors.black,
      ),
      body: Align(
        alignment: Alignment.centerLeft,
        child: Container(
          width: 340,
          height: 400,
          color: Colors.blue[100],
          padding: EdgeInsets.all(10),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Picture of peacock feather',
                style: TextStyle(
                  color: Colors.red,
                  fontSize: 16,
                ),
              ),
```

```
          SizedBox(height: 10),
          Image.network(
            'https://images.pexels.com/photos/674010/pexels-photo-674010.jpeg',
            width: 300,
            height: 180,
            fit: BoxFit.cover,
          ),
          SizedBox(height: 10),
          Row(
            children: [
              Icon(Icons.circle, color: Colors.red, size: 24),
              SizedBox(width: 6),
              Text('circle', style: TextStyle(fontSize: 14)),
            ],
          ),
          SizedBox(height: 10),
          Container(
            width: 100,
            height: 100,
            color: Colors.grey[300],
            child: Stack(
              children: [
                Container(
                  width: 80,
                  height: 80,
                  color: Colors.orange,
                ),
                Positioned(
                  left: 20,
                  top: 20,
                  child: Container(
                    width: 40,
                    height: 40,
                    color: Colors.blue,
                  ),
                ),
                Positioned(
                  left: 30,
                  top: 30,
                  child: Icon(Icons.star, color: Colors.white, size: 20),
                ),
              ],
            ),
          ),
        ],
      ),
    ),
  ),
),
```

);

Output:


Welcome to Dart Page
DEBUG
Picture of peacock feather
circle