EXPERIMENT NO. 3

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

Theory:

In the context of Flutter app development, incorporating icons, images, and custom fonts involves utilizing various widgets and resources provided by the Flutter framework.

1. Icons:

Icons serve as graphical representations of actions, objects, or concepts within an application's user interface. Flutter supports two main types of icons: Material Design icons and Cupertino icons. Material Design icons adhere to Google's design guidelines, while Cupertino icons follow Apple's iOS design principles.

To include icons in a Flutter app, developers typically use the `Icon` widget provided by the Flutter framework. Material Design icons are accessible through the `Icons` class, while Cupertino icons are available via the `CupertinoIcons` class. Additionally, developers can import and use custom icons as needed.

Example code:

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

class IconExample extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Icon Example'),
      ),
      body: Center(
        child: Icon(
          Icons.favorite, // Using a Material Design icon called favorite
          size: 50,
          color: Colors.red,
        ),
      ),
    );
  }
}
```

2. Images:

Images are essential for displaying graphics, photos, and other visual content within a Flutter application. Flutter supports various image formats, including JPEG, PNG, GIF, WebP, Animated GIFs, and Animated WebP.

To include images in a Flutter app, developers commonly use the `Image` widget. Images can be loaded from different sources such as local assets, network URLs, or memory. The `Image.asset()` constructor is often used to load images from the app's asset bundle.

First we have to update the path of the image in the pubspec.yaml file as follows:

```
# pubspec.yaml
flutter:
  assets:
    - assets/images/my_image.png
```

Then we can use it in our code.

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

class ImageExample extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Image Example'),
      ),
      body: Center(
        child: Image.asset(
          'assets/images/my_image.png', // Loading image from assets
          width: 200,
          height: 200,
          fit: BoxFit.contain,
        ),
      ),
    );
  }
}
```

3. Fonts:

Fonts play a crucial role in defining the visual appearance of text within a Flutter application. Custom fonts allow developers to establish a unique typography style that aligns with the app's branding or design requirements.

Integrating custom fonts in a Flutter app involves several steps. Firstly, developers need to add the font files (typically in TrueType Font or OpenType Font format) to the project's `fonts` directory. Then, they specify the fonts in the `pubspec.yaml` file using the `flutter` section. Finally, developers utilize the `TextStyle` widget to apply custom fonts to text widgets by specifying the `fontFamily` property.

By understanding the theoretical aspects and mechanisms behind incorporating icons, images, and custom fonts into a Flutter application, developers can effectively utilize these visual elements to enhance the user experience and achieve desired design outcomes.

First we have to add the font in the pubspec.yaml file

```
# pubspec.yaml
flutter:
  fonts:
    - family: MyCustomFont
      fonts:
        - asset: fonts/my_custom_font.ttf
```

And then use it in our code

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

class FontExample extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Font Example'),
      ),
      body: Center(
        child: Text(
          'Custom Font Example',
          style: TextStyle(
            fontFamily: 'MyCustomFont', // Applying custom font
            fontSize: 24,
          ),
        ),
      ),
    ); }}
```

Code:

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

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

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter UI Example',
      theme: ThemeData(
        primarySwatch: Colors.purple,
      ),
      home: Scaffold(
        body: Container(
          decoration: const BoxDecoration(
            image: DecorationImage(
              image: AssetImage('assets/bgimg.png'), // Replace with
your image path
              fit: BoxFit.cover,
            ),
          ),
          child: const MyHomePage(),
        ),
      ),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key}) : super(key: key);

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  List<String> tasks = [];
  late String newTask; // Variable to hold the value entered in the
TextFormField

  @override
  void initState() {
    super.initState();
    newTask = ''; // Initialize newTask to empty string
  }

  @override
  Widget build(BuildContext context) {
    return SingleChildScrollView(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: <Widget>[
          const Text(
            'Today\'s Tasks',
            style: TextStyle(
              fontSize: 24,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 20),
          for (var task in tasks) TaskTile(title: task),
          const SizedBox(height: 40),
          const Text(
            'Add New Task',
            style: TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 20),
          TextFormField(
            decoration: const InputDecoration(
              filled: true,
              fillColor: Colors.white,
              labelText: 'Enter task name',
              labelStyle: TextStyle(fontSize: 20),
              border: OutlineInputBorder(),
            ),
            onChanged: (value) {
              setState(() {
                newTask = value;
              });
            },
            onFieldSubmitted: (value) {
              setState(() {
                tasks.add(value);
              });
              // Clear the TextFormField after adding task
              newTask = '';
            },
          ),
          const SizedBox(height: 20),
          ElevatedButton(
            onPressed: () {
              setState(() {
                if (newTask.isNotEmpty) {
                  tasks.add(newTask);
                  // Clear the TextFormField after adding task
                  newTask = '';
                }
              });
            },
            child: const Text('Add Task'),
          ),
          const SizedBox(height: 40),
          const Text(
            'Settings',
            style: TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 20),
          ListTile(
            leading: const Icon(Icons.notifications),
            title: const Text('Notifications'),
            trailing: Switch(
              value: true,
              onChanged: (bool value) {
                print('Notifications value: $value');
              },
            ),
          ),
          ListTile(
            leading: const Icon(Icons.color_lens),
            title: const Text('Theme'),
            trailing: IconButton(
              icon: const Icon(Icons.arrow_forward),
              onPressed: () {
                showDialog(
                  context: context,
                  builder: (_) => AlertDialog(
                    title: const Text('Theme Settings'),
                    content: const Text('Navigate to theme settings
page.'),

                    actions: <Widget>[
                      TextButton(
                        onPressed: () {
                          Navigator.pop(context);
                        },
                        child: const Text('Close'),
                      ),
                    ],
                  ),
                );
              },
            ),
          ),
          ListTile(
            leading: const Icon(Icons.info),
            title: const Text('About'),
            trailing: IconButton(
              icon: const Icon(Icons.arrow_forward),
              onPressed: () {
                ScaffoldMessenger.of(context).showSnackBar(
                  const SnackBar(
                    content: Text('Navigate to about page.'),
                  ),
                );
              },
            ),
          ),
        ],
      ),
    );
```

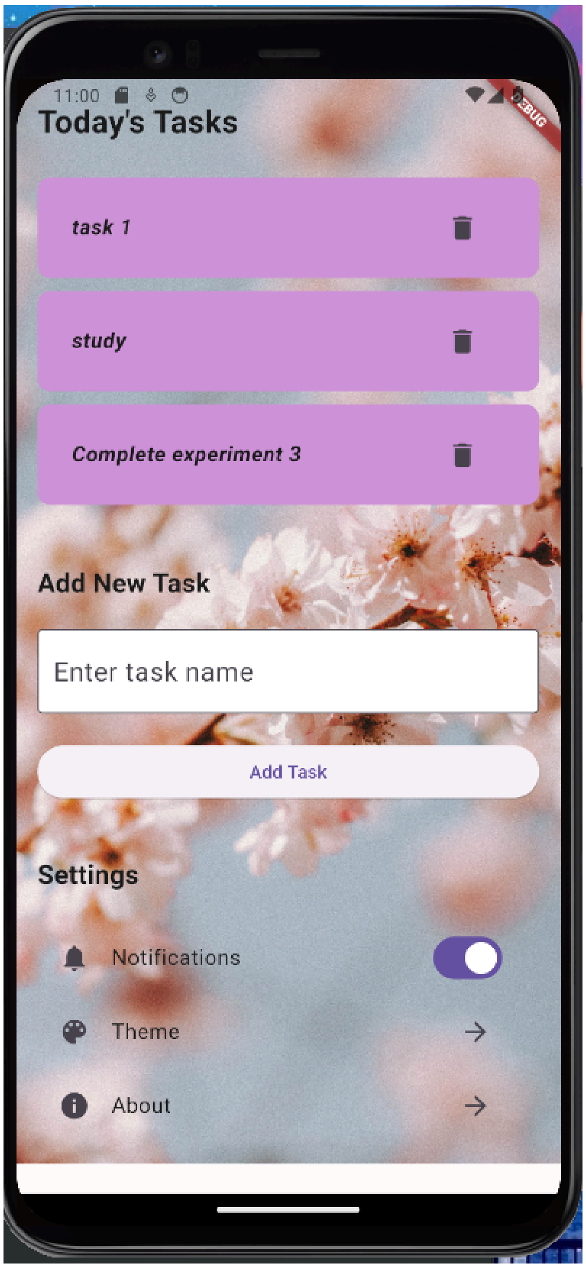
```
  }
}

class TaskTile extends StatelessWidget {
  final String title;

  const TaskTile({Key? key, required this.title}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      margin: const EdgeInsets.symmetric(vertical: 5.0),
      padding: const EdgeInsets.all(10.0),
      decoration: BoxDecoration(
        color: Colors.purple.shade200, // Example color for the
container
        borderRadius: BorderRadius.circular(10.0),
      ),
      child: ListTile(
        title: Text(
          title,
          style: const TextStyle(
            fontFamily: 'Montserrat',
            fontSize: 16,
            fontWeight: FontWeight.bold,
            fontStyle: FontStyle.italic,
          ),
        ),
        trailing: IconButton(
          icon: const Icon(Icons.delete),
          onPressed: () {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text('Task "$title" deleted.'),
                action: SnackBarAction(
                  label: 'Undo',
                  onPressed: () {
                    ScaffoldMessenger.of(context).showSnackBar(
                      const SnackBar(
                        content: Text('Task deletion undone.'),
                      ),
                    );
                  },
                    ),
                  ),
                );
              },
            ),
          ),
        );
      }
    }
```

Updated App Interface With Image Background, various Icons and Fonts.


11:00

Today's Tasks

task 1

study

Complete experiment 3

Add New Task

Enter task name

Add Task

Settings

Notifications

Theme

About