

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

Theory:

In this experiment, we focused on enhancing the Flutter app's user interface by integrating icons, images, and fonts. These components help to create a more engaging, intuitive, and visually appealing app.

1. **Icons:** Icons serve as visual representations of actions, features, or items. In Flutter, we can use pre-defined icons from the Icons class or custom icons to represent various functionalities. Icons help improve navigation and user interaction.
2. **Images:** Images make an app more dynamic and provide visual context. Flutter allows us to include images either from the internet or local assets. Images can be used to represent items, categories, or other visual elements in the app.
3. **Fonts:** Fonts are crucial for displaying text in an aesthetically pleasing way. Flutter supports both default and custom fonts. By setting fonts in the app's theme, we can maintain consistency and style throughout the interface.

Implementation in the Code

In our code, we applied these concepts as follows:

- **Icons:** We used built-in Icons (e.g., Icons.home, Icons.add) for the app bar, navigation bar, and buttons. Custom icons were also used for representing items in the resource list, such as Icons.construction, Icons.grass, and Icons.tv.
- **Images:** We included images for each resource (e.g., Drill Machine, Lawn Mower, Projector) stored locally in the assets/images directory. These images are displayed alongside their respective details (name, category) in the app.
- **Fonts:** We used the default Roboto font throughout the app to keep the text readable and consistent.

Code:

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

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

class LocalResourceApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Local Resource Sharing',
      theme: ThemeData(
```

```
    primaryColor: Colors.teal,  
    scaffoldBackgroundColor: Colors.grey[200],  
    textTheme: TextTheme(  
      bodyLarge: TextStyle(fontFamily: 'Roboto', color: Colors.black),  
      bodyMedium: TextStyle(fontFamily: 'Roboto', color: Colors.black87),  
    ),  
  ),  
  home: HomeScreen(),  
);  
}  
}
```

```
class HomeScreen extends StatefulWidget {  
  @override  
  _HomeScreenState createState() => _HomeScreenState();  
}
```

```
class _HomeScreenState extends State<HomeScreen> {  
  int _selectedIndex = 0;  
  final List<Widget> _pages = [ResourceList(), AddItemScreen()];
```

```
  void _onItemTapped(int index) {  
    setState(() {  
      _selectedIndex = index;  
    });  
  }
```

```
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Row(  
          children: [  
            Icon(Icons.share, color: Colors.white),  
            SizedBox(width: 10),  
            Text('Local Resource Sharing', style: TextStyle(color: Colors.white)),  
          ],  
        ),  
        backgroundColor: Colors.teal,
```

```
    ),  
    body: _pages[_selectedIndex],  
    bottomNavigationBar: BottomNavigationBar(  
      items: [  
        BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),  
        BottomNavigationBarItem(icon: Icon(Icons.add), label: 'Add Item'),  
      ],  
      currentIndex: _selectedIndex,  
      selectedItemColor: Colors.amber,  
      unselectedItemColor: Colors.grey,  
      onTap: _onItemTapped,  
    ),  
  );  
}  
}
```

```
class ResourceList extends StatelessWidget {  
  final List<Map<String, dynamic>> items = [  
    {'name': 'Drill Machine', 'category': 'Home Appliances', 'image': 'assets/images/drill_machine.png',  
     'icon': Icons.construction},  
    {'name': 'Lawn Mower', 'category': 'Gardening', 'image': 'assets/images/lawn_mower.png', 'icon':  
Icons.grass},  
    {'name': 'Projector', 'category': 'Electronics', 'image': 'assets/images/projector.png', 'icon': Icons.tv},  
    {'name': 'Electric Kettle', 'category': 'Kitchen Appliances', 'image': 'assets/images/default.png', 'icon':  
null},  
  ];  
}
```

@override

```
Widget build(BuildContext context) {  
  return ListView.builder(  
    itemCount: items.length,  
    itemBuilder: (context, index) {  
      return Card(  
        color: Colors.white,  
        elevation: 4,  
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)),  
        margin: EdgeInsets.symmetric(horizontal: 10, vertical: 5),  
        child: ListTile(  
          leading: CircleAvatar(  

```

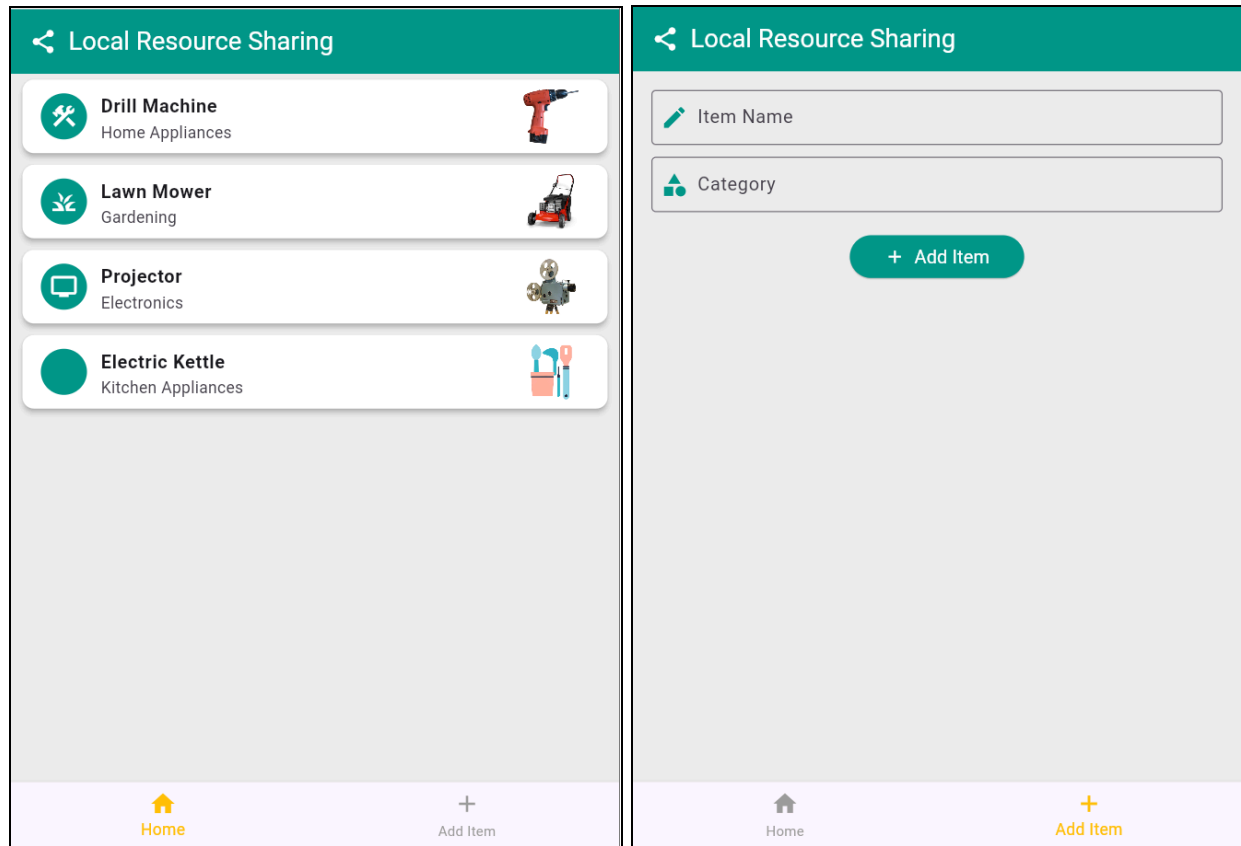
```
        backgroundColor: Colors.teal,
        child: Icon(items[index]['icon'], color: Colors.white),
      ),
      title: Text(items[index]['name'], style: TextStyle(fontWeight: FontWeight.bold)),
      subtitle: Text(items[index]['category']),
      trailing: Image.asset(items[index]['image'], width: 50, height: 50),
      onTap: () {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('${items[index]['name']} selected')),
        );
      },
    ),
  );
},
);
}
```

```
class AddItemScreen extends StatelessWidget {
  final TextEditingController _nameController = TextEditingController();
  final TextEditingController _categoryController = TextEditingController();
```

```
@override
```

```
Widget build(BuildContext context) {
  return Padding(
    padding: EdgeInsets.all(16.0),
    child: Column(
      children: [
        TextField(
          controller: _nameController,
          decoration: InputDecoration(
            labelText: 'Item Name',
            border: OutlineInputBorder(),
            prefixIcon: Icon(Icons.edit, color: Colors.teal),
          ),
        ),
        SizedBox(height: 10),
        TextField(
          controller: _categoryController,
```

```
        decoration: InputDecoration(
          labelText: 'Category',
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.category, color: Colors.teal),
        ),
      ),
    SizedBox(height: 20),
    ElevatedButton.icon(
      onPressed: () {
        if (_nameController.text.isNotEmpty && _categoryController.text.isNotEmpty) {
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text('Item Added: ${_nameController.text}')),
          );
          _nameController.clear();
          _categoryController.clear();
        } else {
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text('Please enter all details')),
          );
        }
      },
      icon: Icon(Icons.add, color: Colors.white),
      label: Text('Add Item', style: TextStyle(color: Colors.white, fontSize: 16)),
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.teal,
        padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),
      ),
    ),
  ],
),
);
}
```

**Conclusion:**

In this experiment, we successfully implemented icons, images, and fonts in the Flutter app, enhancing its user interface. We faced challenges with correctly linking asset images and ensuring proper font application, but resolved them by verifying asset paths and setting up the font in the pubspec.yaml file.