

Aim: To design a Flutter UI by including common widgets.

Introduction

Flutter is an open-source UI software development toolkit created by Google. It allows developers to build natively compiled applications for mobile, web, and desktop from a single codebase. One of Flutter's key strengths is its widget-based architecture, where everything in the UI is a widget, enabling a highly customizable and flexible design.

A Flutter UI is structured using **widgets**, which can be broadly classified into:

- **StatelessWidget:** A widget that does not change over time.
- **StatefulWidget:** A widget that can update dynamically based on user interaction.

Flutter applications typically use a **Scaffold widget**, which provides a structure that includes an AppBar, body, floating action buttons, bottom navigation bars, and other UI elements.

Implementation in Our Project

To achieve the aim of designing a Flutter UI with common widgets, we have implemented the following in our project:

1. **Scaffold and AppBar**
 - The **Scaffold** widget provides the basic structure for our app, including an **AppBar** to display the title.
2. **Bottom Navigation Bar**
 - We have integrated a **BottomNavigationBar** to allow users to navigate between different sections of the app smoothly.
3. **ListView and Cards**
 - The **ListView** widget is used to display a scrollable list of shared resources, with each item represented using a **Card** widget for better UI presentation.
4. **TextFields for Input**
 - The **TextField** widget allows users to enter item details such as name and category when adding a resource.
5. **Buttons and Snackbars**
 - We have implemented **ElevatedButton** to submit input, and **SnackBar** provides instant feedback on user actions.
6. **State Management**
 - We use the **StatefulWidget** to manage dynamic UI elements, such as switching between screens and updating lists.

```
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(primarySwatch: Colors.blue),
      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: Text('Local Resource Sharing')),
      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.blueAccent,
        unselectedItemColor: Colors.grey,
      ),
    );
  }
}
```

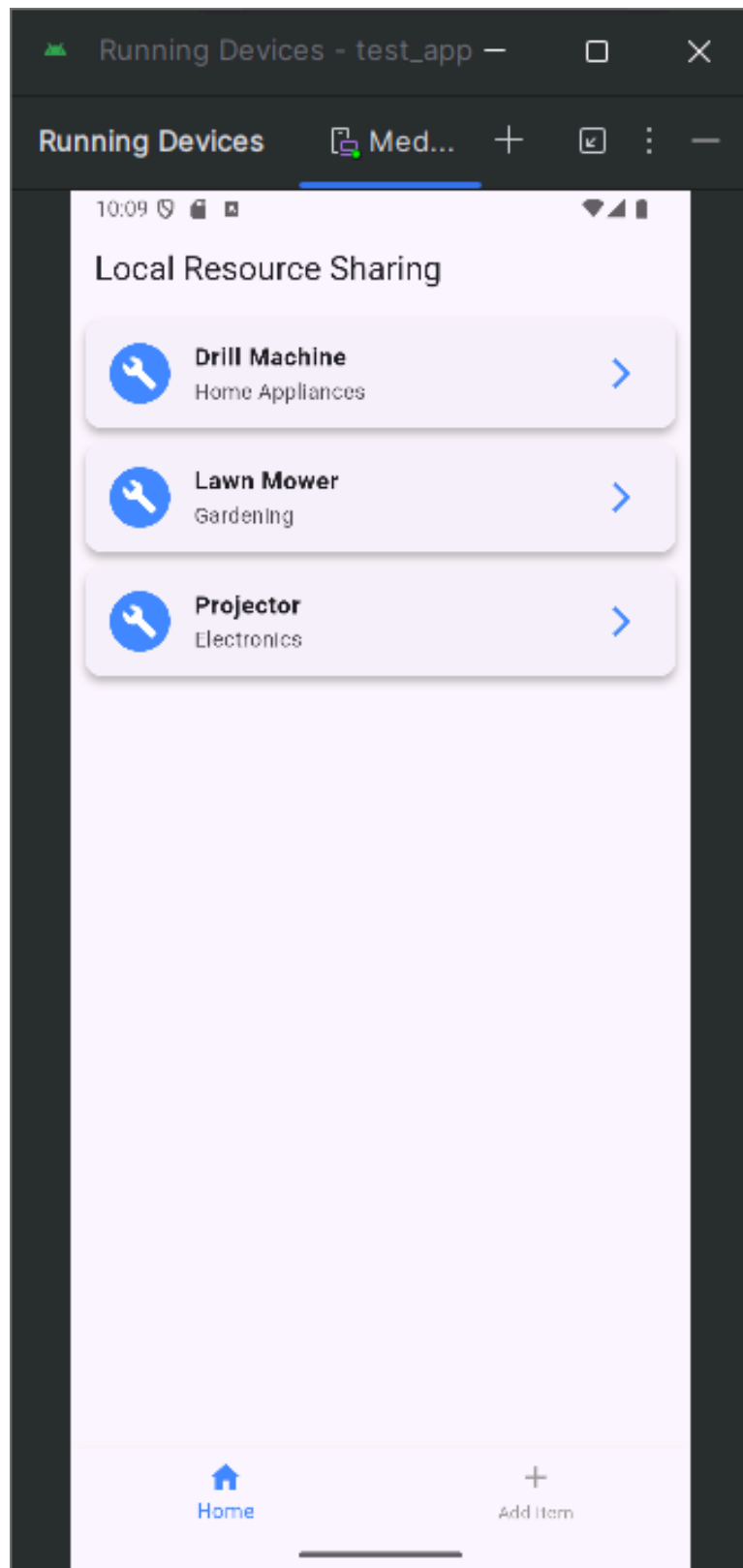
```
        onTap: _onItemTapped,
      ),
    );
  }
}
```

```
class ResourceList extends StatelessWidget {
  final List<Map<String, dynamic>> items = [
    {'name': 'Drill Machine', 'category': 'Home Appliances'},
    {'name': 'Lawn Mower', 'category': 'Gardening'},
    {'name': 'Projector', 'category': 'Electronics'},
  ];

  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      itemCount: items.length,
      itemBuilder: (context, index) {
        return Card(
          elevation: 4,
          shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)),
          margin: EdgeInsets.symmetric(horizontal: 10, vertical: 5),
          child: ListTile(
            leading: CircleAvatar(
              backgroundColor: Colors.blueAccent,
              child: Icon(Icons.build, color: Colors.white),
            ),
            title: Text(items[index]['name'], style: TextStyle(fontWeight: FontWeight.bold)),
            subtitle: Text(items[index]['category']),
            trailing: Icon(Icons.arrow_forward_ios, color: Colors.blueAccent),
            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()),
        ),
        SizedBox(height: 10),
        TextField(
          controller: _categoryController,
          decoration: InputDecoration(labelText: 'Category', border: OutlineInputBorder()),
        ),
        SizedBox(height: 20),
        ElevatedButton(
          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')),
              );
            }
          },
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.blueAccent,
            padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),
          ),
          child: Text('Add Item', style: TextStyle(color: Colors.white, fontSize: 16)),
        )
      ],
    ),
  );
}
```



Conclusion

In this experiment, we implemented a Flutter UI using common widgets like `Scaffold`, `AppBar`, `BottomNavigationBar`, `ListView`, `TextField`, and `ElevatedButton` to create an interactive interface. During development, we faced issues such as improper widget alignment and navigation errors, which we resolved by adjusting layout constraints and properly managing state changes.