Riya Varyani D15A 61

Experiment No. 2

AIM : To design Flutter UI by including common widgets

Theory:

In Flutter, designing UIs involves combining various widgets to build interactive and visually appealing applications. Here's a more detailed overview of key concepts:

1. **Widgets in Flutter**: Everything in Flutter is a widget. Widgets are the building blocks of the UI. There are two main types:
    - **Stateless Widgets**: These are immutable and don't change over time. They are responsible for displaying UI based on fixed data or input.
    - **Stateful Widgets**: These can change their state over time. They are dynamic and are used when the UI needs to update in response to user interaction or other factors.

2. **Layout Widgets**: The layout of your UI is primarily constructed using widgets like:
    - **Container**: A versatile widget used to hold other widgets and apply styling such as padding, margin, colors, and shapes.
    - **Column**: A widget that arranges its children vertically. It's useful for stacking widgets in a vertical list.
    - **Row**: A widget that arranges its children horizontally. It's useful for placing widgets side by side.
    - **Expanded**: A widget that can be used inside a Column, Row, or Flex to make child widgets flexible and fill available space.

3. **Text and Icons**:
    - **Text**: The Text widget is used to display static or dynamic text. It can be styled with custom fonts, sizes, colors, and more.
    - **Icon**: Flutter provides a large set of material design icons, and the Icon widget lets you display them in various sizes and colors.

4. **Buttons and User Interactions**:

    - Flutter provides multiple button widgets like **ElevatedButton**, **TextButton**, and **IconButton** to handle user interaction. These widgets can trigger actions when tapped.

- **TextField**: Used for user input. You can configure it to accept different types of text, such as email or password.
- **Checkbox**, **Radio**, and **Switch**: Used for boolean selections, allowing users to choose options in forms or settings.

5. **Navigation**:
   - Flutter's **Navigator** widget is responsible for managing routes or screens. You use Navigator.push to navigate to a new screen, and Navigator.pop to return to the previous one.
   - **Routes** define the pages of an app, and you can pass data between them using arguments.

6. **Displaying Lists and Grids**:
   - **ListView**: The ListView widget is used to display a list of items that can scroll. It's perfect for long lists that need to be dynamically generated.
   - **GridView**: This widget allows you to display items in a grid format, with configurable row and column layouts.

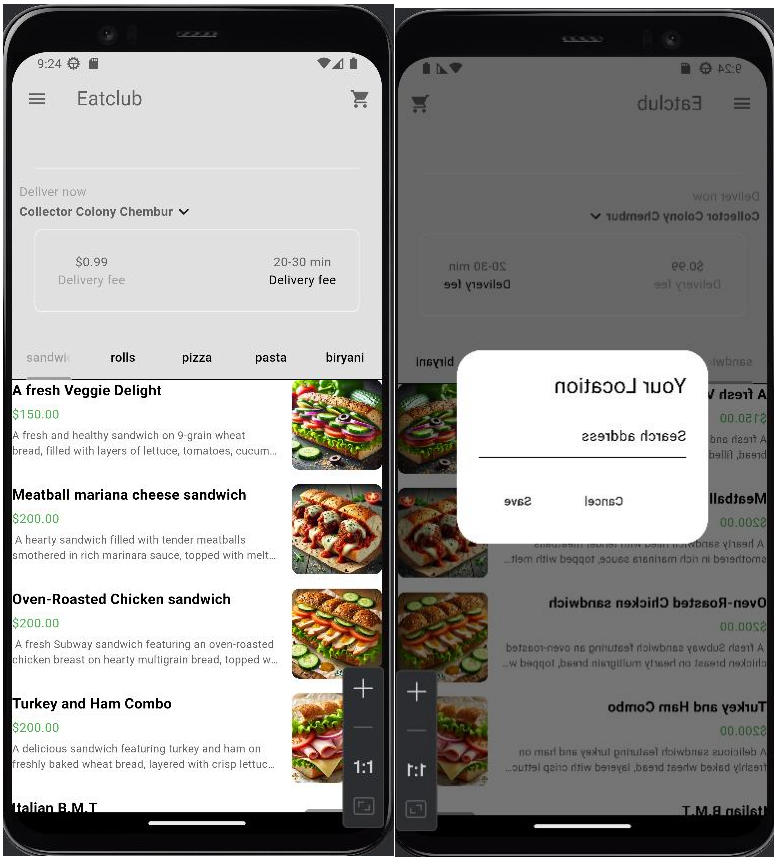
7. **State Management**:
   - Flutter provides a variety of ways to manage state. The simplest approach is using setState() to update the UI. For more complex apps, you can use state management solutions like **Provider**, **Riverpod**, or **Bloc** to separate business logic from UI code.
   - Proper state management ensures your UI stays in sync with the underlying data, especially in interactive or dynamic applications.

8. **Theming and Styling**:
   - Flutter allows you to define a global **Theme** for your app using ThemeData, which ensures consistent styling across the entire app. You can customize colors, typography, and button styles.

9. **Animations and Transitions**:
   - Flutter provides powerful animation support to create smooth and visually appealing transitions between UI states.
   - **AnimatedContainer**: A widget that animates changes in properties like width, height, or color over a given duration.

- You can also create custom animations using **AnimationController** and **Tween**.

Screenshots:



Code Snippets:

Scaffold & Column Widget

```
import 'package:flutter/material.dart';
import 'package:flutter_eats/components/my_current_location.dart';
import 'package:flutter_eats/components/my_description_box.dart';
import 'package:flutter_eats/components/my_food_tile.dart';
import 'package:flutter_eats/components/my_sliver_app_bar.dart';
import 'package:flutter_eats/models/food.dart';
import 'package:flutter_eats/models/restaurant.dart';
import 'package:flutter_eats/pages/food_page.dart';
import 'package:provider/provider.dart';
import '../components/my_drawer.dart';
```

```
import '../components/my_tab_bar.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> with SingleTickerProviderStateMixin {
  // Tab Controller
  late TabController _tabController;

  @override
  void initState() {
    super.initState();
    _tabController = TabController(length: FoodCategory.values.length, vsync: this);
  }

  @override
  void dispose() {
    _tabController.dispose();
    super.dispose();
  }

  // Sort out and return a list of food items that belong to a specific category
  List<Widget> getFoodInThisCategory(List<Food> fullMenu) {
    return FoodCategory.values.map((category) {
      List<Food> categoryMenu = fullMenu.where((food) => food.category == category).toList();

      return ListView.builder(
        itemCount: categoryMenu.length,  // Fixed syntax error (was itemBuilder:
categoryMenu.lenght)
        physics: const NeverScrollableScrollPhysics(),
        padding: EdgeInsets.zero,
        itemBuilder: (context, index) {
          //get individual food
          final food = categoryMenu[index];
          return FoodTile(
            food: food,
            onTap: () => Navigator.push(
              context,
              MaterialPageRoute(
```

```
              builder: (context) => FoodPage(food: food),
            ),
          ),  // Fixed incorrect indexing (was categoryMenu[food])
        );
      },
    );
  }).toList();
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
    drawer: const MyDrawer(),
    body: NestedScrollView(
      headerSliverBuilder: (context, innerBoxIsScrolled) => [
        MySliverAppBar(
          title: MyTabBar(tabController: _tabController),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.end,
            children: [
              Divider(
                indent: 25,
                endIndent: 25,
                color: Theme.of(context).colorScheme.secondary,
              ),
              // My current location
              const MyCurrentLocation(),

              // Description box
              const MyDescriptionBox(),
            ],
          ),
        ),
      ],
      body: Consumer<Restaurant>(
         builder: (context, restaurant, child) => TabBarView(
            controller: _tabController,
            children: getFoodInThisCategory(restaurant.menu),)
      ),
    ),
  );
 }
}
```

Riya Varyani D15A 61