

EXPERIMENT NO :- 02

AIM: - To design Flutter UI by including common widgets.

Theory: -

Each element on the screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps. And the structure of the code of apps is a tree of widgets.

When you made any alteration in the code, the widget rebuilds its description by calculating the difference of previous and current widget to determine the minimal changes for rendering in UI of the app. Widgets are nested with each other to build the app. It means the root of your app is itself a widget, and all the way down is a widget also. For example, a widget can display something, can define design, can handle interaction, etc.

The single child layout widget is a type of widget, which can have only one child widget inside the parent layout widget. These widgets can also contain special layout functionality. Flutter provides us with many single child widgets to make the app UI attractive. If we use these widgets appropriately, it can save us time and make the app code more readable.

The multiple child widgets are a type of widget, which contains more than one child widget, and the layout of these widgets are unique. For example, Row widget laying out of its child widget in a horizontal direction, and Column widget laying out of its child widget in a vertical direction. If we combine the Row and Column widget, then it can build any level of the complex widget.

Type of Widgets

- StatefulWidget
 - A StatefulWidget has state information. It contains mainly two classes: the state object and the widget. It is dynamic because it can change the inner data during the widget lifetime. This widget does not have a build() method. It has a createState() method, which returns a class that extends the Flutter's State Class. The examples of the StatefulWidget are Checkbox, Radio, Slider, InkWell, Form, and TextField.
- StatelessWidget
 - The StatelessWidget does not have any state information. It remains static throughout its lifecycle. The examples of the StatelessWidget are Text, Row, Column, Container, etc.

Some of the commonly used widgets

Container – A box widget used for styling with padding, margins, colors, borders, and constraints. It helps in layout structuring and positioning.

Row & Column – Used to arrange widgets in horizontal (Row) or vertical (Column) orientation. They manage spacing, alignment, and distribution of child widgets.

Stack – Overlaps widgets on top of each other, useful for creating layered UIs like banners, tooltips, or floating elements.

Text – Displays text on the screen with customizable font size, color, alignment, and styling options

Image – Loads and displays images from assets, network, or memory with scaling, fit, properties.

Scaffold – Provides a basic layout structure with an app bar, body, floating action button, and bottom navigation.

ListView – A scrollable list widget that efficiently renders large amounts of dynamic content. Supports both vertical and horizontal scrolling.

GridView – Displays widgets in a grid format, useful for galleries, product listings, or dashboards. It supports dynamic column adjustments.

SizedBox – Used to create space between widgets or define fixed width and height for layout adjustments.

ElevatedButton – A button with elevation that provides a raised effect, customizable with color, shape, and click actions.

TextField – A user input field that supports text entry, keyboard configurations, validation.

AppBar – A top navigation bar that includes a title, actions, and menu icons, commonly used in Scaffold.

BottomNavigationBar – A bar at the bottom of the screen used for navigation between different app sections with icons and labels.

Drawer – A side navigation panel that slides out from the left, typically used for app menus and quick navigation.

Card – A material design component that displays content inside a box with elevation.

Code:

home_screen.dart

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

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

  @override
  State<HomeScreen> createState() =>
    _HomeScreenState();
}

class _HomeScreenState extends
  State<HomeScreen> {
  int _currentCarouselIndex = 0;
  final PageController _pageController =
    PageController();

  final List<Map<String, dynamic>>
    carouselData = [
    {
      'icon': Icons.fitness_center,
      'title': 'Featured Workout',
      'description': 'Try our new HIIT routine for
maximum burn!',
      'color': Colors.blueAccent,
    },
    {
      'icon': Icons.local_dining,
      'title': 'Nutrition Tip',
      'description': 'Include more protein in your
meals for muscle gain.',
      'color': Colors.green,
    },
    {
      'icon': Icons.self_improvement,
      'title': 'Mindfulness',
      'description': 'Take a moment to meditate
and destress.',
      'color': Colors.purple,
    },
  ];

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Home"),
        backgroundColor: Colors.black,
        centerTitle: true,
      ),
      backgroundColor: Colors.black,
      body: SingleChildScrollView(
```

```

child: Column(
  children: [
    // Carousel Section with Custom
    Widgets
    SizedBox(
      height: 200,
      child: PageView.builder(
        controller: _pageController,
        itemCount: carouselData.length,
        onPageChanged: (index) {
          setState(() {
            _currentCarouselIndex = index;
          });
        },
        itemBuilder: (context, index) {
          final item = carouselData[index];
          return Container(
            margin: const
EdgeInsets.symmetric(horizontal: 10, vertical:
10),
            decoration: BoxDecoration(
              borderRadius:
BorderRadius.circular(12),
              color:
item['color'].withOpacity(0.2),
            ),
            padding: const
EdgeInsets.all(16),
            child: Row(
              children: [
                Icon(
                  item['icon'],
                  size: 48,
                  color: item['color'],
                ),
                const SizedBox(width: 16),
                Expanded(
                  child: Column(
                    mainAxisAlignment:
MainAxisAlignment.center,
                    crossAxisAlignment:
CrossAxisAlignment.start,
                    children: [
                      Text(
                        item['title'],
                        style: const TextStyle(
                          color: Colors.white,
                          fontSize: 20,
                          fontWeight:
FontWeight.bold,

```

```

      ),
    ),
    const SizedBox(height: 8),
    Text(
      item['description'],
      style: const TextStyle(
        color: Colors.white70,
        fontSize: 16,
      ),
    ),
  ],
),
],
);
},
),
),
// Carousel Indicator Dots
Row(
  mainAxisAlignment:
MainAxisAlignment.center,
  children:
List.generate(carouselData.length, (index) {
  return AnimatedContainer(
    duration: const
Duration(milliseconds: 300),
    margin: const
EdgeInsets.symmetric(horizontal: 4),
    height: 8,
    width: _currentCarouselIndex ==
index ? 20 : 8,
    decoration: BoxDecoration(
      color: _currentCarouselIndex ==
index ? Colors.blue : Colors.white54,
      borderRadius:
BorderRadius.circular(4),
    ),
  );
}),
),
const SizedBox(height: 20),

// Trackers Section
Padding(
  padding: const
EdgeInsets.symmetric(horizontal: 16),
  child: Row(
    children: const [

```

```

Text(
  "Trackers",
  style: TextStyle(
    color: Colors.white,
    fontSize: 20,
    fontWeight: FontWeight.bold,
  ),
),
],
),
const SizedBox(height: 10),
SizedBox(
  height: 150,
  child: ListView(
    scrollDirection: Axis.horizontal,
    padding: const
EdgeInsets.symmetric(horizontal: 16),
    children: [
      _buildTrackerCard(
        icon: Icons.favorite,
        title: "Heart Rate",
        value: "72 bpm",
      ),
      const SizedBox(width: 10),
      _buildTrackerCard(
        icon: Icons.directions_run,
        title: "Steps",
        value: "5,432",
      ),
      const SizedBox(width: 10),
      _buildTrackerCard(
        icon: Icons.local_fire_department,
        title: "Calories",
        value: "1,234 cal",
      ),
    ],
  ),
),
const SizedBox(height: 20),

// Goals Section
Padding(
  padding: const
EdgeInsets.symmetric(horizontal: 16),
  child: Row(
    children: const [
      Text(
        "Your Goals",
        style: TextStyle(

```

```

    color: Colors.white,
    fontSize: 20,
    fontWeight: FontWeight.bold,
  ),
),
const SizedBox(height: 10),
Padding(
  padding: const
EdgeInsets.symmetric(horizontal: 16),
  child: Row(
    children: [
      Expanded(
        child: _buildGoalCard(
          title: "Remaining Calories",
          value: "850 cal",
          icon:
Icons.local_fire_department,
          color: Colors.orange,
        ),
      ),
      const SizedBox(width: 10),
      Expanded(
        child: _buildGoalCard(
          title: "Water Intake",
          value: "5/8 glasses",
          icon: Icons.water,
          color: Colors.lightBlue,
        ),
      ),
    ],
  ),
const SizedBox(height: 20),

// Graph Section
Padding(
  padding: const
EdgeInsets.symmetric(horizontal: 16),
  child: Row(
    children: const [
      Text(
        "Progress Graph",
        style: TextStyle(
          color: Colors.white,
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),

```

```

    ),
    ],
    ),
    ),
    const SizedBox(height: 10),
    _buildGraphWidget(),
    const SizedBox(height: 30),
    ],
    ),
    ),
    );
}

// Tracker Card
Widget _buildTrackerCard({
  required IconData icon,
  required String title,
  required String value,
}) {
  return Container(
    width: 140,
    padding: const EdgeInsets.all(16),
    decoration: BoxDecoration(
      color: Colors.grey[900],
      borderRadius: BorderRadius.circular(12),
    ),
    child: Column(
      mainAxisAlignment:
MainAxisAlignment.center,
      children: [
        Icon(icon, size: 40, color: Colors.blue),
        const SizedBox(height: 10),
        Text(
          title,
          style: const TextStyle(
            color: Colors.white70,
            fontSize: 16,
          ),
        ),
        ),
        const SizedBox(height: 5),
        Text(
          value,
          style: const TextStyle(
            color: Colors.white,
            fontSize: 18,
            fontWeight: FontWeight.bold,
          ),
        )
      ],
    ),
  );
}

```

```

    );
  }

// Goal Card
Widget _buildGoalCard({
  required String title,
  required String value,
  required IconData icon,
  required Color color,
}) {
  return Container(
    padding: const EdgeInsets.all(16),
    decoration: BoxDecoration(
      color: Colors.grey[900],
      borderRadius: BorderRadius.circular(12),
    ),
    child: Column(
      children: [
        Icon(icon, size: 40, color: color),
        const SizedBox(height: 10),
        Text(
          title,
          textAlign: TextAlign.center,
          style: const TextStyle(
            color: Colors.white70,
            fontSize: 14,
          ),
        ),
        ),
        const SizedBox(height: 5),
        Text(
          value,
          style: const TextStyle(
            color: Colors.white,
            fontSize: 16,
            fontWeight: FontWeight.bold,
          ),
        )
      ],
    ),
    );
}

// Placeholder for Graph Widget
Widget _buildGraphWidget() {
  return Container(
    margin: const
EdgeInsets.symmetric(horizontal: 16),
    height: 200,
    decoration: BoxDecoration(
      color: Colors.grey[900],

```

```

borderRadius: BorderRadius.circular(12),
),
child: const Center(
  child: Text(
    "Graph \n(Progress Over Time)",
    textAlign: TextAlign.center,
    style: TextStyle(

```

```

color: Colors.white70,
fontSize: 16,
),
),
);
}
}

```

main_screen.dart

```

import 'package:flutter/material.dart';
import 'home_screen.dart';
import 'log_food_screen.dart';
import 'settings_screen.dart';

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

  @override
  State<MainScreen> createState() =>
    _MainScreenState();
}

class _MainScreenState extends
  State<MainScreen> {
  int _currentIndex = 0;

  // List of pages corresponding to each
  bottom nav item.
  final List<Widget> _pages = const [
    HomeScreen(),
    LogFoodScreen(),
    SettingsScreen(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black, // Ensures
      a black background for the screen
      body: _pages[_currentIndex], // Display
      selected page

```

```

bottomNavigationBar:
  BottomNavigationBar(
    backgroundColor: Colors.grey[850], //
    Ensure a distinct background color
    currentIndex: _currentIndex,
    unselectedItemColor: Colors.white70,
    selectedItemColor: Colors.blue,
    showUnselectedLabels: true,
    items: const [
      BottomNavigationBarItem(
        icon: Icon(Icons.home),
        label: "Home",
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.fastfood),
        label: "Log Food",
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.settings),
        label: "Settings",
      ),
    ],
    onTap: (index) {
      setState(() {
        _currentIndex = index; // Update the
        current selected index
      });
    },
  );
}
}

```

Screenshots:

