EXPERIMENT NO: - 02

Name:- Swaraj Patil **Class:-** D15A **Roll:No: -** 39 **AIM: -** To design Flutter UI by including

common widgets.

Theory: -

https://github.com/Swap0-4/MPL-LAB/blob/main/MPL%20EXPERIMENT%20NO

%2002.pdf

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 many single child widgets to make the app UI attractive. If we use these widgets appropriately, it can save our time and makes 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 Widgetss

➢ **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 createState() method, which returns a class that extends the Flutters 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: -

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









void main() {



  runApp(MyApp());



}
```

```dart
class MyApp extends StatelessWidget {

  const MyApp({super.key});




  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Telegram',

      theme: ThemeData(

        primarySwatch: Colors.blue,

        appBarTheme: const AppBarTheme(
```

```
      color: Colors.white,

      iconTheme: IconThemeData(color: Colors.blue),

      elevation: 0,

      titleTextStyle: TextStyle(

        color: Colors.black,

        fontSize: 20,

        fontWeight: FontWeight.bold,

      ),

    ),

    scaffoldBackgroundColor: Colors.white,
```

```
      ),

      home: const HomePage(), // Corrected: Now points to HomePage

    );

  }

}

class Chat {

  final String name;

  final String lastMessage;

  final String time;
```

```dart
final int unreadCount;

final bool isOnline;

final bool isTyping;

final bool isMuted;

final bool isPinned;




Chat({

  required this.name,

  required this.lastMessage,

  required this.time,
```

```
    this.unreadCount = 0,

    this.isOnline = false,

    this.isTyping = false,

    this.isMuted = false,

    this.isPinned = false,

  });

}

class HomePage extends StatelessWidget {

  const HomePage({super.key});
```

```dart
@override

Widget build(BuildContext context) {

  final List<Chat> chats = [ // Initialized here!

    Chat(

      name: 'Emma',

      lastMessage: 'Hey, are you coming to the party?',

      time: '10:02',

      unreadCount: 2,

      isOnline: true,
```

```
    ),

    Chat(
      name: 'Noah',
      lastMessage: 'Check out this new app!',
      time: '09:45',
      isTyping: true,
    ),

    Chat(
      name: 'Olivia',
      lastMessage: '📷 Photo',
```

```
    time: '09:30',
    isMuted: true,
  ),
  Chat(
    name: 'Liam',
    lastMessage: 'Thanks for the help!',
    time: '08:15',
    unreadCount: 1,
  ),
  Chat(
```

```
      name: 'Telegram',

      lastMessage: 'Update available',

      time: '07:00',

      isPinned: true,

    ),

  ];

  return Scaffold(

    appBar: AppBar(

      title: const Text('Telegram'),
```

```
      actions: [

        IconButton(

          icon: const Icon(Icons.search, color: Colors.blue),

          onPressed: () {},

        ),

        PopupMenuButton<String>(

          icon: const Icon(Icons.more_vert, color: Colors.blue),

          onSelected: (value) {},

          itemBuilder: (BuildContext context) => [

            const PopupMenuItem(value: 'New Group', child: Text('New
```

```
Group')),


              const PopupMenuItem(value: 'New Channel', child: Text('New
Channel')),


              const PopupMenuItem(value: 'Settings', child:
Text('Settings')),


            ],


          ),


        ],


      ),


      body: ListView.separated(


        itemCount: chats.length,


        separatorBuilder: (context, index) => const Divider(height: 1),
```

```
      itemBuilder: (context, index) {

        final chat = chats[index];

        return ListTile(

          leading: _buildAvatar(chat),

          title: Row(

            children: [

              Text(

                chat.name,

                style: TextStyle(

                  fontWeight: chat.unreadCount > 0 || chat.isPinned
```

```
                    ? FontWeight.bold


                    : FontWeight.normal,


                ),


              ),


              if (chat.isMuted)


                const Icon(Icons.volume_off, size: 16, color:
Colors.grey),


            ],


          ),


          subtitle: chat.isTyping
```

```
              ? const Text('typing...', style: TextStyle(color:
Colors.green))



              : Text(



                  chat.lastMessage,



                  style: TextStyle(



                    color: chat.unreadCount > 0 ? Colors.black :
Colors.grey,



                    fontWeight: chat.unreadCount > 0



                        ? FontWeight.bold



                        : FontWeight.normal,



                  ),
```

```
                maxLines: 1,

                overflow: TextOverflow.ellipsis,

              ),

        trailing: _buildTrailing(chat),

        onTap: () {},

      );

    },

  ),

  floatingActionButton: FloatingActionButton(

    backgroundColor: Colors.blue,
```

```
      onPressed: () {},

      child: const Icon(Icons.edit, color: Colors.white),

    ),

    bottomNavigationBar: BottomNavigationBar(

      currentIndex: 0,

      selectedItemColor: Colors.blue,

      unselectedItemColor: Colors.grey,

      type: BottomNavigationBarType.fixed,

      items: const [ // Added const here

        BottomNavigationBarItem(
```

```
        icon: Icon(Icons.chat_bubble),

        label: 'Chats',

      ),

      BottomNavigationBarItem(

        icon: Icon(Icons.call),

        label: 'Calls',

      ),

      BottomNavigationBarItem(

        icon: Icon(Icons.people),

        label: 'Contacts',
```

```
        ),

        BottomNavigationBarItem(

          icon: Icon(Icons.settings),

          label: 'Settings',

        ),

      ],

    ),

  );

}
```

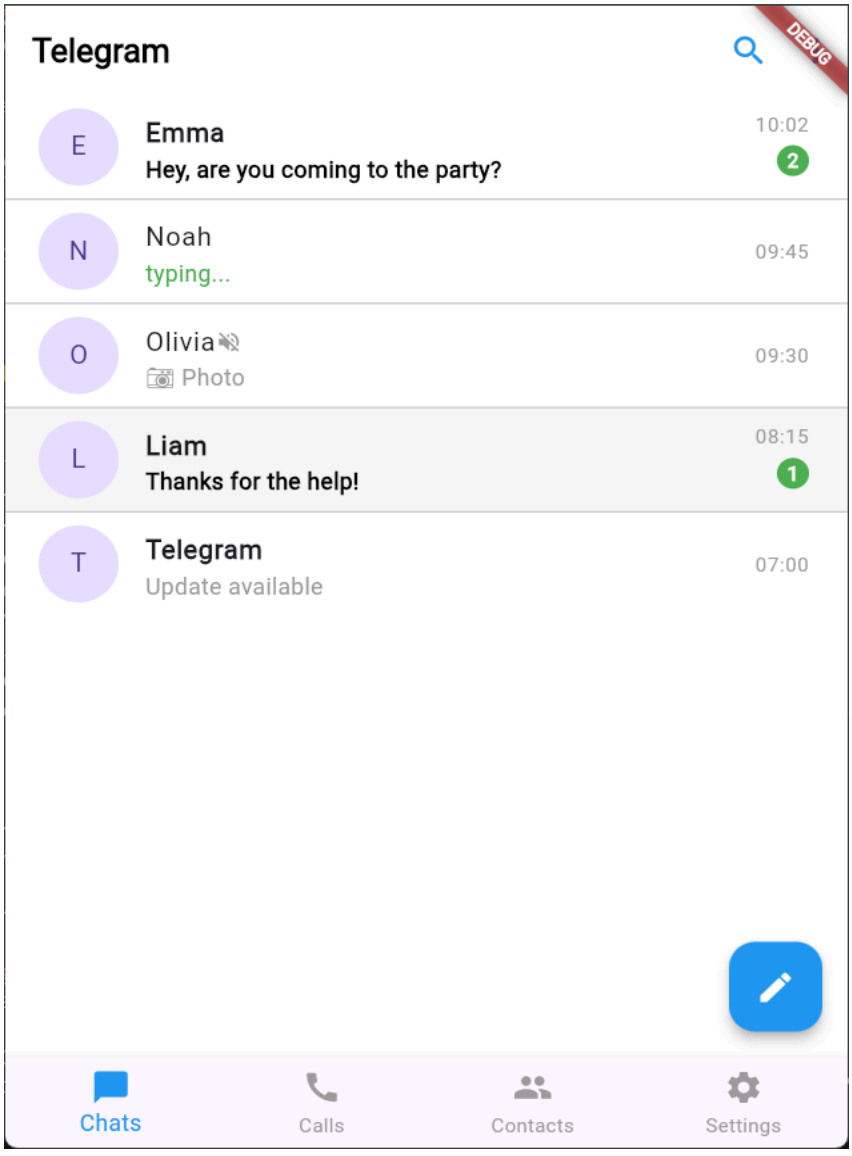
```
Widget _buildAvatar(Chat chat) {

  return CircleAvatar(

    radius: 28,

    child: Text(chat.name[0]),

  );

}



Widget _buildTrailing(Chat chat) {

  return Column(

    mainAxisAlignment: MainAxisAlignment.center,
```

```
    crossAxisAlignment: CrossAxisAlignment.end,

    children: [

      Text(

        chat.time,

        style: const TextStyle(

          color: Colors.grey,

          fontSize: 12,

        ),

      ),

      if (chat.unreadCount > 0)
```

```
        Container(

          padding: const EdgeInsets.all(6),

          decoration: const BoxDecoration(

            color: Colors.green,

            shape: BoxShape.circle,

          ),

          child: Text(

            chat.unreadCount.toString(),

            style: const TextStyle(

              color: Colors.white,
```

```
                fontSize: 12,


                fontWeight: FontWeight.bold,


              ),


            ),


          ),


      ],


    );


  }


}
```

OUTPUT:


Telegram
DEBUG
E
Emma
Hey, are you coming to the party?
10:02
2
N
Noah
typing...
09:45
O
Olivia
Photo
09:30
L
Liam
Thanks for the help!
08:15
1
T
Telegram
Update available
07:00
Chats
Calls
Contacts
Settings