Experiment - 2

Name: Preeti Khamkar **Division:** D15A **Roll no. :** 28

Aim: To design Flutter UI by including common widgets.

Theory:

In Flutter, there are several common widgets that are used to build user interfaces. These widgets serve as the building blocks for creating various UI elements. Here are some of the most commonly used ones:

1. Container: A versatile widget that can contain other widgets and provides properties for styling, padding, margin, alignment, and more.
2. Text: Displays a string of text with properties for styling such as font size, color, alignment, and font weight.
3. Image: Displays an image from various sources such as assets, the internet, or memory.
4. Row: Arranges its children widgets horizontally in a row.
5. Column: Arranges its children widgets vertically in a column.
6. Stack: Stacks its children widgets on top of each other, allowing for overlapping UI elements.
7. ListView: Displays a scrollable list of children widgets.
8. GridView: Displays a scrollable grid of children widgets arranged in rows and columns.
9. AppBar: A material design app bar that typically contains a title, actions, and/or navigation elements.
10. Scaffold: Implements the basic material design visual layout structure, providing a framework for the app's UI.
11. IconButton: A button widget that displays an icon and triggers an action when pressed.
12. TextField: Allows users to input text and provides properties for customization such as keyboard type, input validation, and styling.
13. RaisedButton and FlatButton: Button widgets that trigger an action when pressed, with raised and flat styles respectively.
14. AlertDialog: Displays a modal dialog box with a message and optional actions.
15. BottomNavigationBar: Provides navigation between top-level views in the app, typically displayed at the bottom of the screen.

These are just a few examples of the many widgets available in Flutter. You can combine these widgets and customize their properties to create complex and beautiful user interfaces for your Flutter applications.

Syntax:

1. Container Widget

```
Container(
  // Properties

  // Child widget
  child: Text(),
)
```

2. Text Widget

```
Text(
  style: TextStyle(
  ),
)
```

3. RaisedButton Widget

```
RaisedButton(
  onPressed: () {
    // Action to perform when the button is pressed
  },
)
```

4. Scaffold Widget

```
Scaffold(
  appBar: AppBar(
  ),
)
```

Widget and Properties:

TextButton, TetField, AppBar, Container, Scaffold are the widgets used in this experiment.

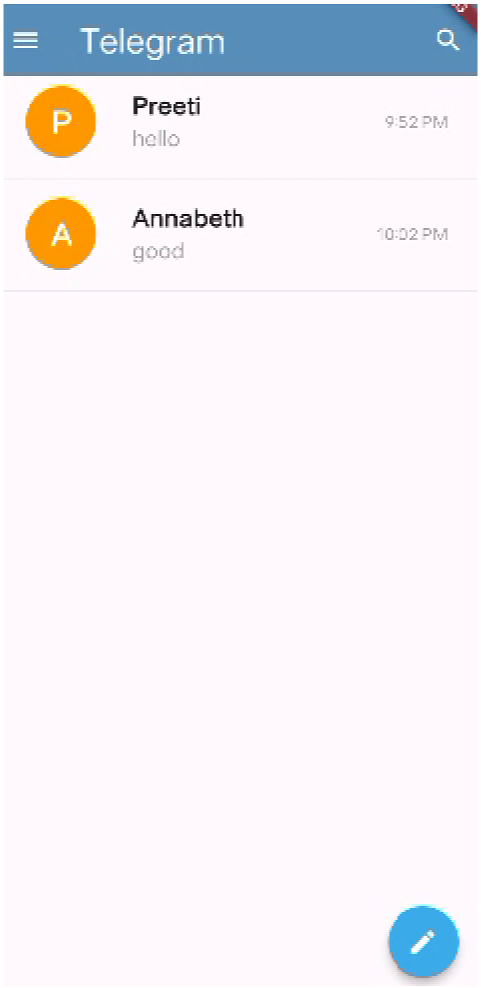
Code and Output:

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:telegram/screen/chatPage.dart';
import '../contacts.dart';
import '../search.dart';
import 'drawer/settings.dart';
import 'package:intl/intl.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key});

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

  // Make the method public
  static String constructChatRoomId(String phoneNumber1, String phoneNumber2) {
    List<String> phoneNumbers = [phoneNumber1, phoneNumber2];
    phoneNumbers.sort();
    return phoneNumbers.join("");
  }
}

class _HomePageState extends State<HomePage> {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  void signOut() async {
    await _auth.signOut();
    // Navigate to login screen
    Navigator.pushReplacementNamed(context, '/login');
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Telegram', style: TextStyle(color: Colors.white)),
        backgroundColor: Color(0xFF588FBA),
        actions: [
          IconButton(
            icon: Icon(Icons.search),
            onPressed: () {
              // Navigate to search screen
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => Search(),
                ),
              );
            },
          ),
        ],
        iconTheme: IconThemeData(color: Colors.white), // Set the color of drawer icon to white
      ),

      drawer: Drawer(
        child: StreamBuilder<DocumentSnapshot>(
          stream: FirebaseFirestore.instance
              .collection('users')
              .doc(_auth.currentUser!.uid)
              .snapshots(),
          builder: (context, snapshot) {
            if (snapshot.hasError) {
              return Text('Error: ${snapshot.error}');
            }
            if (snapshot.connectionState == ConnectionState.waiting) {
              return Center(child: CircularProgressIndicator());
            }

            var userData = snapshot.data!.data() as Map<String, dynamic>;
            var profileImageUrl = userData['profileImageUrl'] ?? '';
            var username = userData['username'] ?? '';
            var phoneNumber = userData['phone'] ?? '';

            return ListView(
              padding: EdgeInsets.zero,
              children: [
                DrawerHeader(
                  decoration: BoxDecoration(color: Color(0xFF588FBA)), // Changed to #588FBA
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      CircleAvatar(
                        backgroundImage: profileImageUrl.isNotEmpty
                            ? NetworkImage(profileImageUrl)
                            : null,
                        backgroundColor:
                        Colors.orange, // Default background color
              radius: 30, // Adjust as needed
              child: profileImageUrl.isEmpty
                  ? Text(
                username[0].toUpperCase(), // Display the first letter of the username
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 24, // Adjust the font size as needed
                  fontWeight: FontWeight.bold, // Make the font bold
                ),
              )
                  : null,
            ),
            SizedBox(height: 8),
            Row(
              children: [
                Expanded(
                  child: Text(
                    username,
                    style:
                    TextStyle(color: Colors.white, fontSize: 18),
                  ),
                ),
                Icon(Icons.keyboard_arrow_down,
                    color: Colors.white), // Arrow color set to white
              ],
            ),
            Text(
              phoneNumber,
              style: TextStyle(color: Colors.white),
            ),
          ],
        ),
      ),
      ListTile(
        leading:
        Icon(Icons.group, color: Colors.grey), // Changed icon color
        title: Text('New Group'),
        onTap: () {
          // Navigate to new group screen
        },
      ),
      ListTile(
        leading: Icon(Icons.contacts,
            color: Colors.grey), // Changed icon color
        title: Text('Contacts'),
        onTap: () {
          // Navigate to Contacts screen
        },
      ),
      ListTile(
        leading: Icon(Icons.call,
            color: Colors.grey), // Changed icon color
        title: Text('Call'),
        onTap: () {
          // Navigate to Contacts screen
        },
      ),
      ListTile(
        leading: Icon(Icons.emoji_people_outlined,
            color: Colors.grey), // Changed icon color
        title: Text('People Nearby'),
        onTap: () {
          // Navigate to People Nearby screen
        },
      ),
      ListTile(
        leading: Icon(Icons.save,
            color: Colors.grey), // Changed icon color
        title: Text('Saved Messages'),
        onTap: () {
          // Navigate to Saved Messages screen
        },
      ),
      ListTile(
        leading: Icon(Icons.settings,
            color: Colors.grey), // Changed icon color
        title: Text('Settings'),
        onTap: () {
          // Navigate to settings screen
          Navigator.push(
              context,
              MaterialPageRoute(
                  builder: (context) => ProfilePage(
                      userId: _auth.currentUser!.uid)));
        },
      ),
      Divider(
        color: Colors.grey[300], // Added a light grey line separator
```

```
            thickness: 1,
          ),
          ListTile(
            leading: Icon(Icons.person_add_outlined,
                color: Colors.grey), // Changed icon color
            title: Text('Invite Friends'),
            onTap: () {
              // Invite Friends
            },
          ),
          ListTile(
            leading: Icon(Icons.help_outline,
                color: Colors.grey), // Changed icon color
            title: Text('Telegram Features'),
            onTap: () {
              signOut();
            },
          ),
          ListTile(
            leading: Icon(Icons.logout,
                color: Colors.grey), // Changed icon color
            title: Text('Logout'),
            onTap: () {
              signOut();
            },
          ),
        ],
      );
    },
  ),
),
body: _buildUserList(),
floatingActionButton: FloatingActionButton(
  onPressed: () {
    // Navigate to ContactsPage
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => ContactsPage(), // Replace ContactsPage with the actual page
      ),
    );
  },
  child: Icon(Icons.edit, color: Colors.white,),
  backgroundColor: Color(0xFF3BACEB),
      shape: CircleBorder(),// Adjust the color to match the theme
    ),
    floatingActionButtonLocation: FloatingActionButtonLocation.endFloat,
  );
}


// build list of users except the current user
Widget _buildUserList() {
  return StreamBuilder<QuerySnapshot>(
    stream: FirebaseFirestore.instance.collection('users').snapshots(),
    builder: (context, snapshot) {
      if (snapshot.hasError) {
        return Text('Error: ${snapshot.error}');
      }
      if (snapshot.connectionState == ConnectionState.waiting) {
        return Center(child: CircularProgressIndicator());
      }

      return ListView(
        children: snapshot.data!.docs
            .map<Widget>((doc) => _buildUserListItem(doc))
            .toList(),
      );
    },
  );
}

// build individual user list items
Widget _buildUserListItem(DocumentSnapshot document) {
  Map<String, dynamic>? data = document.data() as Map<String, dynamic>?;

  if (_auth.currentUser != null &&
      _auth.currentUser!.phoneNumber != null &&
      data != null) {
    if (data['phone'] != null &&
        _auth.currentUser!.phoneNumber != data['phone']) {
      String displayName = data['username'] ?? data['phone'];

      return StreamBuilder<QuerySnapshot>(
        stream: FirebaseFirestore.instance
            .collection('chat_rooms')
            .doc(HomePage.constructChatRoomId(_auth.currentUser!.phoneNumber.toString(),
data['phone']))
```

```
    .collection('messages')
    .orderBy('timestamp', descending: true)
    .limit(1)
    .snapshots(),
builder: (context, snapshot) {
  if (snapshot.hasData &&
      snapshot.data!.docs.isNotEmpty) {
    var lastMessage = snapshot.data!.docs.first;
    var timestamp = lastMessage['timestamp'];
    var messageTimestamp = DateTime.fromMillisecondsSinceEpoch(timestamp.seconds * 1000);
    var formattedTimestamp = DateFormat.jm().format(messageTimestamp);

    return Column(
      children: [
        ListTile(
          title: Text(
            displayName,
            style: TextStyle(fontWeight: FontWeight.bold),
          ), // Make the username bold
          subtitle: Text(
            lastMessage['message'],
            maxLines: 1,
            overflow: TextOverflow.ellipsis,
            style: TextStyle(color: Colors.grey), // Set text color to grey
          ),
          trailing: Text(
            formattedTimestamp,
            style: TextStyle(color: Colors.grey), // Set text color to grey
          ),
          leading: CircleAvatar(
            backgroundImage: data['profileImageUrl'] != null
                ? NetworkImage(data['profileImageUrl'])
                : null,
            backgroundColor: Colors.orange, // Default background color
            radius: 40, // Increased the radius to make the profile icon bigger
            child: data['profileImageUrl'] == null
                ? Text(
              displayName[0].toUpperCase(), // Display the first letter of the username
              style: TextStyle(
                color: Colors.white,
                fontSize: 24, // Adjust the font size as needed
                fontWeight: FontWeight.bold, // Make the font bold
              ),
            )
```

```
                         : null, // Show the first letter only when no profile image is available
                   ),
                   onTap: () {
                     Navigator.push(
                       context,
                       MaterialPageRoute(
                         builder: (context) => ChatPage(
                           receivedUserPhoneNumber: data['phone'] ?? '',
                         ),
                       ),
                     );
                   },
                 ),
                 Divider(
                   color: Colors.grey[300], // Added a light grey line separator
                   thickness: 1,
                 ),
               ],
             );
           } else {
             return SizedBox.shrink(); // Return an empty SizedBox if there are no messages
           }
         },
       );
     }
   }
   return SizedBox.shrink(); // Return an empty SizedBox if user data is invalid
 }
}
```



Preeti
hello
9:52 PM

Annabeth
good
10:02 PM