

**Prachit P. Paralikar**  
**D15A**  
**43**

## **MAD PWA LAB 2**

### **AIM:**

To design Flutter UI by including common widgets.

### **THEORY:**

**Some common widgets used in Flutter are as follows:**

#### **MaterialApp:**

Represents the overall structure of a Flutter application, providing settings for the app.

#### **Container:**

A box model allowing customization of size, padding, margin, and decoration for its child widget.

#### **Column and Row:**

Column: Arranges children vertically.

Row: Arranges children horizontally.

#### **Text:**

Displays text with specific styles using the TextStyle class.

#### **Image:**

Displays images from various sources, including network URLs or local assets.

#### **ListView:**

Displays a scrolling list of widgets, allowing users to scroll through content.

#### **GestureDetector:**

Captures user gestures like taps, double taps, and swipes, enabling interactive behavior.

#### **Stack:**

Overlaps widgets, allowing them to be positioned on top of each other.

#### **AppBar:**

Represents the app bar at the top of the screen, typically containing the app's title and actions.

#### **TextField:**

Allows users to input text, providing a UI for text entry.

### **SYNTAX:**

**Syntaxes for some of common widgets are as follows:**

### **1.Text**

```
Text(  
  "Text Content",  
  style:TextStyle(  
    //style properties  
  )  
)
```

### **2.Container**

```
Container(  
  //style like  
  Width: 100,  
  Height: 100,  
  child://child widget to which container contains  
)
```

### **3.TextButton**

```
TextButton(  
  Event: action,  
  child: Text('Button placeholder'),  
)
```

### **WIDGET AND PROPERTIES:**

The widgets used in the following code are MaterialApp, Scaffold, Padding, Column, TextField, SizedBox, TextButton, Icon, Text, Edgelnsets.

- Scaffold is used to define the basic visual structure of the screen.
- SizedBox creates a box of a specified height.
- Edgelnsets provide padding between content and the box.

### **CODE:**

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

```
import 'package:messenger_clone/helperfunctions/sharedpref_helper.dart';
```

```
import 'package:messenger_clone/services/auth.dart';
```

```
import 'package:messenger_clone/services/database.dart';
```

```
import 'package:messenger_clone/views/chatscreen.dart';
```

```
import 'package:messenger_clone/views/signin.dart';
```

```
import 'package:messenger_clone/views/stories_screen.dart'; // Import the Stories  
screen
```

```
import 'package:cloud_firestore/cloud_firestore.dart';
```

```
class Home extends StatefulWidget {
```

```
  @override
```

```
  _HomeState createState() => _HomeState();
```

```
}
```

```
class _HomeState extends State<Home> {
```

```
  bool isSearching = false;
```

```
  String? myName, myProfilePic, myUserName, myEmail;
```

```
Stream? usersStream, chatRoomsStream;
```

```
TextEditingController searchUsernameEditingController =
```

```
TextEditingController();
```

```
getMyInfoFromSharedPreferences() async {
```

```
myName = await SharedPreferencesHelper().getDisplayName();
```

```
myProfilePic = await SharedPreferencesHelper().getUserProfileUrl();
```

```
myUserName = await SharedPreferencesHelper().getUserName();
```

```
myEmail = await SharedPreferencesHelper().getUserEmail();
```

```
setState(() {});
```

```
}
```

```
getChatRoomIdByUsernames(String a, String b) {
```

```
if (a.substring(0, 1).codeUnitAt(0) > b.substring(0, 1).codeUnitAt(0)) {  
  
    return "$b\_a";  
  
} else {  
  
    return "$a\_b";  
  
}  
  
}
```

```
onSearchBtnClick() async {  
  
    isSearching = true;  
  
    setState(() {});  
  
    usersStream = await DatabaseMethods()  
  
        .getUserByUsername(searchUsernameEditingController.text);  
  
  
    setState(() {});  
  
}
```

```
}
```

```
Widget chatRoomsList() {
```

```
  return StreamBuilder(  
    stream: chatRoomsStream,  
    builder: (context, snapshot) {  
      return snapshot.hasData  
        ? ListView.builder(  
          itemCount: snapshot.data!.docs.length,  
          shrinkWrap: true,  
          itemBuilder: (context, index) {  
            DocumentSnapshot ds = snapshot.data!.docs[index];  
            return ChatRoomListTile(  
              ds["lastMessage"], ds.id, myUserName!);  
          }  
        )  
        : Center(  
          child: Text("No chat rooms found")  
        );  
    }  
  );  
}
```

```
  })
```

```
    : Center(child: CircularProgressIndicator());
```

```
  },
```

```
);
```

```
}
```

```
Widget searchListUserTile(
```

```
  {required String profileUrl,
```

```
  required String name,
```

```
  required String username,
```

```
  required String email}) {
```

```
  return GestureDetector(
```

```
    onTap: () {
```

```
      var chatRoomId = getChatRoomIdByUsernames(myUserName!, username);
```

```
Map<String, dynamic> chatRoomInfoMap = {
```

```
  "users": [myUserName, username]
```

```
};
```

```
DatabaseMethods().createChatRoom(chatRoomId, chatRoomInfoMap);
```

```
Navigator.push(
```

```
  context,
```

```
  MaterialPageRoute(
```

```
    builder: (context) => ChatScreen(username, name)));
```

```
},
```

```
child: Container(
```

```
  margin: EdgeInsets.symmetric(vertical: 8),
```

```
  child: Row(
```

```
    children: [
```

```
      CircleAvatar(
```



```
        radius: 20,

        backgroundImage: NetworkImage(profileUrl),

      ),

      SizedBox(width: 12),

      Column(

        crossAxisAlignment: CrossAxisAlignment.start,

        children: [Text(name), Text(email)])

    ],

  ),

),

);

}
```

```
Widget searchUsersList() {
```

```
return StreamBuilder(

  stream: usersStream,

  builder: (context, snapshot) {

    return snapshot.hasData

      ? ListView.builder(

        itemCount: snapshot.data!.docs.length,

        shrinkWrap: true,

        itemBuilder: (context, index) {

          DocumentSnapshot ds = snapshot.data!.docs[index];

          return searchListUserTile(

            profileUrl: ds["imgUrl"],

            name: ds["name"],

            email: ds["email"],

            username: ds["username"]);
```

```
        },  
      )  
      : Center(  
        child: CircularProgressIndicator(),  
      );  
    },  
  );  
}
```

```
getChatRooms() async {  
  
  chatRoomsStream = await DatabaseMethods().getChatRooms();  
  
  setState(() {});  
}
```

```
onScreenLoaded() async {  
  
    await getMyInfoFromSharedPreference();  
  
    getChatRooms();  
  
}
```

```
@override
```

```
void initState() {  
  
    onScreenLoaded();  
  
    super.initState();  
  
}
```

```
@override
```

```
Widget build(BuildContext context) {  
  
    return Scaffold(  

```

**appBar: AppBar(**

**title: Text("Chats"),**

**actions: [**

**InkWell(**

**onTap: () {**

**AuthMethods().signOut(context).then((s) {**

**Navigator.pushReplacement(**

**context, MaterialPageRoute(builder: (context) => SignIn()));**

**});**

**},**

**child: Padding(**

**padding: const EdgeInsets.all(8.0),**

**child: Icon(Icons.exit\_to\_app),**

**),**

)

],

),

body: Container(

margin: EdgeInsets.symmetric(horizontal: 20),

child: Column(

children: [

Row(

children: [

isSearching

? GestureDetector(

onTap: () {

isSearching = false;

searchUsernameEditingController.text = "";

```
setState(() {});
```

```
},
```

```
child: Padding(
```

```
padding: EdgeInsets.only(right: 12),
```

```
child: Icon(Icons.arrow_back)),
```

```
)
```

```
: Container(),
```

```
Expanded(
```

```
child: Container(
```

```
margin: EdgeInsets.symmetric(vertical: 16),
```

```
padding: EdgeInsets.symmetric(horizontal: 16),
```

```
decoration: BoxDecoration(
```

```
border: Border.all(
```

```
color: Colors.grey,
```

**width: 1,**

**),**

**borderRadius: BorderRadius.circular(24),**

**),**

**child: Row(**

**children: [**

**Expanded(**

**child: TextField(**

**controller: searchUsernameEditingController,**

**decoration: InputDecoration(**

**border: InputBorder.none,**

**hintText: "Search",**

**),**

**),**



),

**GestureDetector(**

**onTap: () {**

**if (searchUsernameEditingController.text != "") {**

**onSearchBtnClick();**

**}**

**},**

**child: Icon(Icons.search),**

**),**

**],**

**),**

**),**

**),**

**],**

),

isSearching ? searchUsersList() : chatRoomsList(),

],

),

),

bottomNavigationBar: BottomAppBar(

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

IconButton(

icon: Icon(Icons.chat),

onPressed: () {

// Add functionality for chat

},

),

IconButton(

icon: Icon(Icons.view\_headline),

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (context) => StoriesScreen()),

);

},

),

],

),

),

);

```
}
```

```
}
```

```
class ChatRoomListTile extends StatefulWidget {
```

```
  final String lastMessage, chatRoomId, myUsername;
```

```
  ChatRoomListTile(this.lastMessage, this.chatRoomId, this.myUsername);
```

```
  @override
```

```
  _ChatRoomListTileState createState() => _ChatRoomListTileState();
```

```
}
```

```
class _ChatRoomListTileState extends State<ChatRoomListTile> {
```

```
  String profilePicUrl = "", name = "", username = "";
```

```
getThisUserInfo() async {
```

```
    username =
```

```
        widget.chatRoomId.replaceAll(widget.myUsername, "").replaceAll("_", "");
```

```
    QuerySnapshot querySnapshot = await DatabaseMethods().getUserInfo(username);
```

```
    print(
```

```
        "something bla bla ${querySnapshot.docs[0].id} ${querySnapshot.docs[0]["name"]} ${querySnapshot.docs[0]["imgUrl"]}");
```

```
    name = "${querySnapshot.docs[0]["name"]}";
```

```
    profilePicUrl = "${querySnapshot.docs[0]["imgUrl"]}";
```

```
    setState(() {});
```

```
}
```

```
@override
```

```
void initState() {
```

```
    getThisUserInfo();
```

```
super.initState();  
  
}
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
  return GestureDetector(
```

```
    onTap: () {
```

```
      Navigator.push(
```

```
        context,
```

```
        MaterialPageRoute(
```

```
          builder: (context) => ChatScreen(username, name)));
```

```
    },
```

```
    child: Container(
```

```
      margin: EdgeInsets.symmetric(vertical: 8),
```

**child: Row(**

**children: [**

**CircleAvatar(**

**radius: 20,**

**backgroundImage: NetworkImage(profilePicUrl),**

**),**

**SizedBox(width: 12),**

**Column(**

**crossAxisAlignment: CrossAxisAlignment.start,**

**children: [**

**Text(**

**name,**

**style: TextStyle(fontSize: 16),**

**),**

```
        SizedBox(height: 3),

        Text(widget.lastMessage)

      ],

    ),

    ],

  ),

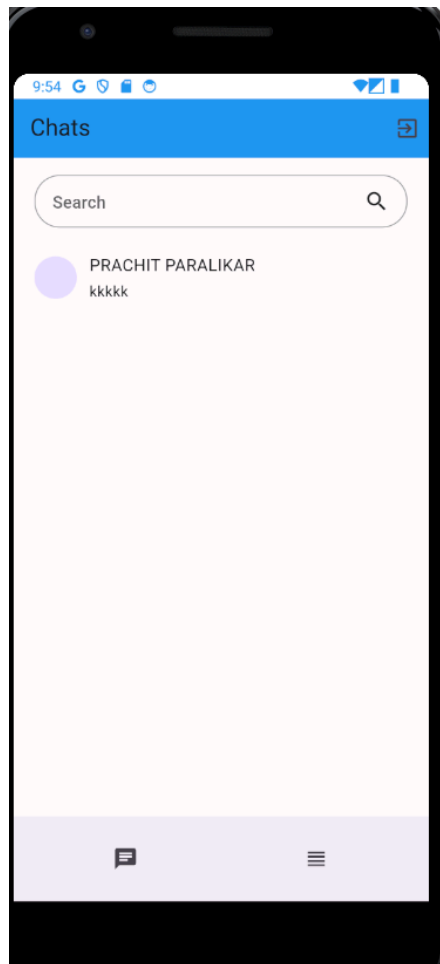
);

}

}
```

**OUTPUT:**





**CONCLUSION:**

Learned about basic widgets used in flutter.

Learned about syntaxes for the same.