Experiment - 6

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

Aim: To Connect Flutter UI with fireBase database

Theory:
Connecting a Flutter app to Firebase involves several steps, including setting up a Firebase project, configuring the Flutter app to use Firebase, and integrating Firebase SDKs into your Flutter code.

1. Firebase Setup: First, you need to create a Firebase project in the Firebase console (https://console.firebase.google.com/). This project will host your app's data and services.
2. Add Firebase to Flutter App: You'll need to add the Firebase SDK to your Flutter app. This involves adding Firebase configuration files to your project and updating your Flutter app's dependencies to include Firebase SDKs.
3. Initialize Firebase: In your Flutter app, you'll initialize Firebase by calling Firebase.initializeApp(). This should typically be done at the beginning of your app's lifecycle, such as in the main() function or in the initState() method of your main widget.
4. Use Firebase Services: Once Firebase is initialized, you can use various Firebase services in your Flutter app, such as Authentication, Firestore, Realtime Database, Cloud Storage, Cloud Messaging, and more.

Syntax:
```
WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
```

And import necessary packages. Add firebase_core and firebase_auth packages in pubsec.yaml.

Widget and Properties:
TextButton, TextField, AppBar, Container, Scaffold are the widgets used in this experiment.

Code and Output:
```
import 'dart:io';
```

```
import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:telegram/model/message.dart';
import 'package:telegram/services/chat/chat_service.dart';
import 'package:intl/intl.dart';
import 'package:telegram/mediaScreen.dart';

class ChatPage extends StatefulWidget {
 final String receivedUserPhoneNumber;
 const ChatPage({
   Key? key,
   required this.receivedUserPhoneNumber,
 }) : super(key: key);

 // Hardcoded set of usernames for receiver user's phone number
 static Map<String, String> hardcodedUsernames = {
   '+911234567890': 'Percy',
   '+919867506341': 'Preeti',
   '+911234567891': 'Annabeth',
   '+911987654321': 'Priya',
   '+911987654321': 'Gargi'
   // Add more mappings as needed
 };

 @override
 State<ChatPage> createState() => _ChatPageState();
}

class _ChatPageState extends State<ChatPage> {
 final TextEditingController _messageController = TextEditingController();
 final ChatService _chatService = ChatService();
 final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;

 void sendMessage() async {
   if (_messageController.text.isNotEmpty) {
     await _chatService.sendMessage(
       widget.receivedUserPhoneNumber,
       _messageController.text,
     );
     _messageController.clear();
   }
```

```
}

 Future<void> _selectImageFromGallery() async {
  final pickedFile = await ImagePicker().pickImage(source: ImageSource.gallery);
  if (pickedFile != null) {
   String imageUrl = await _uploadImage(File(pickedFile.path));
   await _chatService.sendMessage(
    widget.receivedUserPhoneNumber,
    '',
    imageUrl: imageUrl,
   );
  }
 }

 Future<String> _uploadImage(File imageFile) async {
  Reference storageReference =
FirebaseStorage.instance.ref().child('chat_images/${DateTime.now().millisecondsSinceEpoch}');
  UploadTask uploadTask = storageReference.putFile(imageFile);
  TaskSnapshot snapshot = await uploadTask;
  String downloadUrl = await snapshot.ref.getDownloadURL();
  return downloadUrl;
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    backgroundColor: const Color(0xFF588FBA),
    title: Row(
     children: [
      GestureDetector(
       onTap: () {
        // Navigate to Media Screen
        Navigator.push(
         context,
         MaterialPageRoute(builder: (context) => MediaScreen(userPhoneNumber:
widget.receivedUserPhoneNumber,)),
        );
       },
       child: Row(
        children: [
         // Add profile icon here, for example:
         CircleAvatar(
          radius: 20,
```

```
                backgroundColor: Colors.orange, // Default background color
                child: Text(
                  ChatPage.hardcodedUsernames[widget.receivedUserPhoneNumber]?.substring(0, 1) ?? '', //
Display the first letter of the username
                  style: TextStyle(
                    color: Colors.white,
                    fontWeight: FontWeight.bold,
                  ),
                ),
              ),
              SizedBox(width: 13), // Add some spacing between icon and text
              // Add username text here
              Text(
                ChatPage.hardcodedUsernames[widget.receivedUserPhoneNumber] ?? 'Preeti',
                style: TextStyle(color: Colors.white),
              ),
            ],
          ),
        ),
        SizedBox(width: 110),
        Icon(Icons.call, color: Colors.white),
        SizedBox(width: 15),
        Icon(Icons.more_vert, color: Colors.white)
      ],
    ),
    iconTheme: IconThemeData(color: Colors.white), // Change back arrow color to white
  ),
  body: Container(
    decoration: BoxDecoration(
      image: DecorationImage(
        image: AssetImage("assets/images/chatBg.png"), // Replace "assets/background_image.jpg" with
your image asset path
        fit: BoxFit.cover,
      ),
    ),
    child: Column(
      children: [
        Expanded(
          child: _buildMessageList(),
        ),
        _buildMessageInput()
      ],
    ),
  ),
```

```
  );
}



Widget _buildMessageList() {
  return StreamBuilder(
    stream: _chatService.getMessages(widget.receivedUserPhoneNumber,
_firebaseAuth.currentUser!.phoneNumber.toString()),
    builder: (context, snapshots) {
      if (snapshots.hasError) {
        print("Error fetching messages: ${snapshots.error}");
        return Text('Error: ${snapshots.error}');
      }
      if (snapshots.connectionState == ConnectionState.waiting) {
        print("Waiting for messages...");
        return const Text('Loading...');
      }
      if (!snapshots.hasData) {
        print("No message data available.");
        return const Text('No messages yet.');
      }
      return ListView(
        children: snapshots.data!.docs
            .map((document) => _buildMessageItem(document))
            .toList(),
      );
    },
  );
}

Widget _buildMessageItem(DocumentSnapshot document) {
  Map<String, dynamic> data = document.data() as Map<String, dynamic>;

  var alignment = (data['senderPhoneNumber'] == _firebaseAuth.currentUser!.phoneNumber)
      ? Alignment.centerRight
      : Alignment.centerLeft;

  if (data['imageUrl'] != null && data['imageUrl'] != '') {
    return Container(
      alignment: alignment,
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Column(
```

```
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          FutureBuilder(
            future: _getUserDisplayName(data['senderPhoneNumber']),
            builder: (context, snapshot) {
              if (snapshot.connectionState == ConnectionState.waiting) {
                return CircularProgressIndicator();
              } else {
                return Text(snapshot.data.toString());
              }
            },
          ),
          Container(
            constraints: BoxConstraints(maxWidth: 250.0), // Adjust the max width as needed
            child: ClipRRect(
              borderRadius: BorderRadius.circular(8.0),
              child: Image.network(
                data['imageUrl'],
                width: 250.0, // Adjust the width as needed
                height: 200.0, // Adjust the height as needed
                fit: BoxFit.cover,
              ),
            ),
          ),
          Text(
            _formatTimestamp(data['timestamp']),
            style: TextStyle(color: Colors.grey),
          ), // Add this line to display the timestamp
        ],
      ),
    ),
  );
} else {
  return Container(
    alignment: alignment,
    child: Padding(
      padding: const EdgeInsets.all(8.0),
      child: Stack(
        children: [
          Container(
            padding: EdgeInsets.symmetric(vertical: 10, horizontal: 15),
            decoration: BoxDecoration(
              color: alignment == Alignment.centerRight ? Color(0xFFE1FFC6) : Colors.grey[100],
              borderRadius: BorderRadius.circular(20),
```

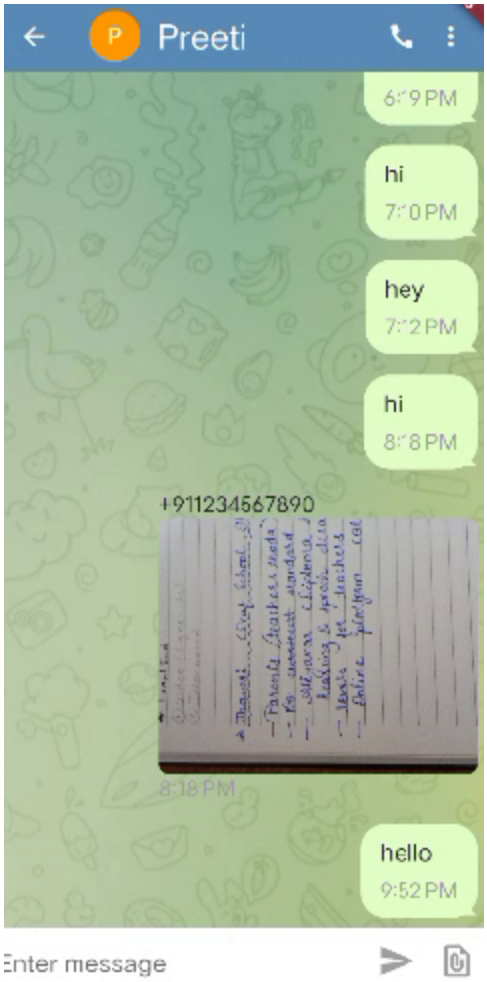
```
            ),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  data['message'],
                  style: TextStyle(fontSize: 16),
                ),
                SizedBox(height: 5),
                Text(
                  _formatTimestamp(data['timestamp']),
                  style: TextStyle(color: Colors.grey),
                ), // Add this line to display the timestamp
              ],
            ),
          ),
          Positioned(
            bottom: -10,
            right: alignment == Alignment.centerRight ? -8 : null,
            left: alignment == Alignment.centerLeft ? -8 : null,
            child: Icon(
              Icons.arrow_drop_up,
              color: alignment == Alignment.centerRight ? Color(0xFFE1FFC6) : Colors.grey[100],
              size: 30,
            ),
          ),
        ],
      ),
    ),
  );
 }
}

String _formatTimestamp(Timestamp timestamp) {
 DateTime dateTime = timestamp.toDate();
 String formattedTime = DateFormat.jm().format(dateTime); // Format the time as desired
 return formattedTime;
}

Widget _buildMessageInput() {
 return Container(
   color: Colors.white, // Set background color to white
   child: Row(
     children: [
```

```
        Expanded(
          child: TextField(
            controller: _messageController,
            decoration: const InputDecoration(
              hintText: 'Enter message',
            ),
            obscureText: false,
          ),
        ),
        IconButton(
          onPressed: sendMessage,
          icon: const Icon(
            Icons.send,
            size: 30,
            color: Colors.grey,
          ),
        ),
        IconButton(
          onPressed: _selectImageFromGallery,
          icon: const Icon(
            Icons.file_present_outlined,
            size: 30,
            color: Colors.grey,
          ),
        )
      ],
    ),
  );
}


Future<String> _getUserDisplayName(String phoneNumber) async {
  var snapshot = await FirebaseFirestore.instance.collection('users').doc(phoneNumber).get();
  var userData = snapshot.data();
  if (userData != null) {
    return userData['username'] ?? userData['phone'];
  }
  return phoneNumber;
}
}
```


Preeti
6:19 PM
hi
7:10 PM
hey
7:12 PM
hi
8:18 PM
+911234567890
8:18 PM
hello
9:52 PM
Enter message