MAD LAB 14

01-135211-102

Group Member:

Muhammad Hammad 211-102.

Hammad ul Hasan 058.

Q1: Project Document: **[CLO-3]** (10)

Your semester project document contain the following:

1. Project Title

Ans) The title of the project is **Chat Ease.**

2. Project Scope (Functionality you have completed as well as not completed, mention both)

Ans) ChatEase contains the main features such as.

Signin:

Allows users to authenticate and access the application securely using their credentials.

Signup:

Enables users to create new accounts within the application to gain access to its features and functionalities.

ChatpageScreen:

Displays the main interface for users to engage in real-time conversations with other users, fostering communication and interaction.

Firebase As a Database:

Utilizes Firebase as a backend service to store and manage application data, ensuring seamless data synchronization and scalability.

Forgot Password:
Provides functionality for users to recover their account access by resetting their forgotten passwords through email verification or other security measures.

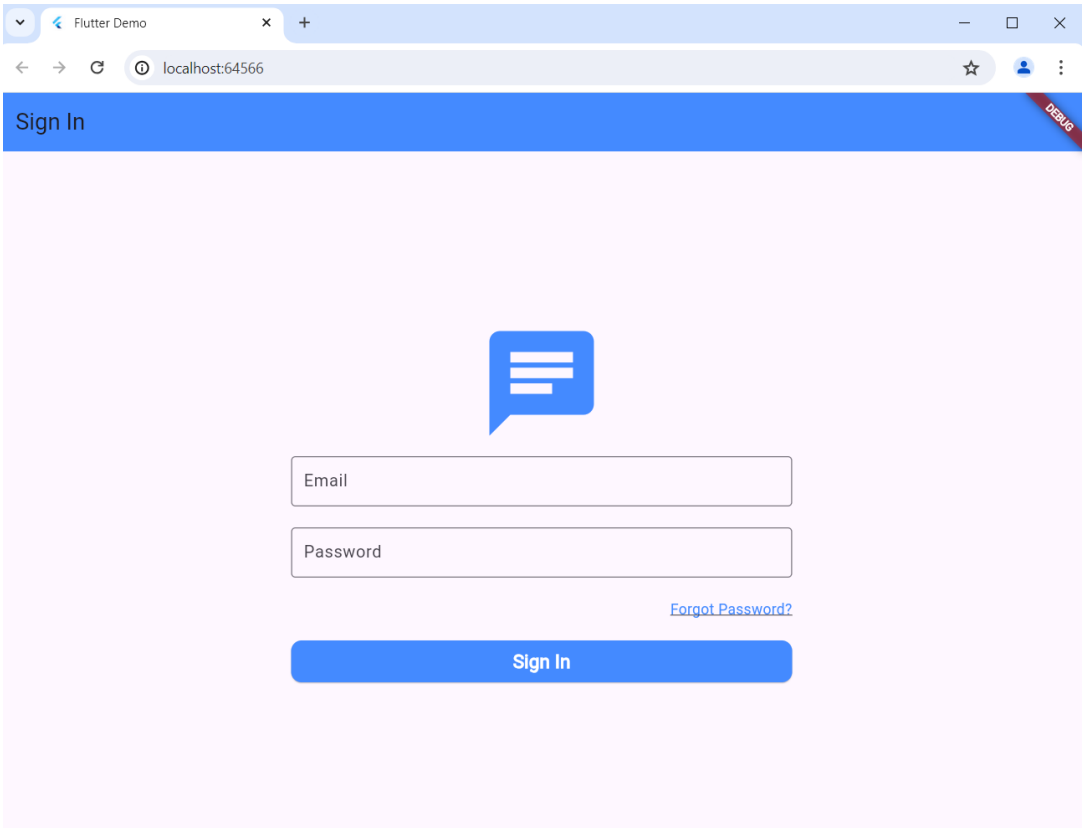
Advanced Search Feature:
Enhances user experience by enabling them to perform complex searches within the application, allowing for refined and targeted results.

Chatrooms:
Organizes conversations into dedicated spaces, facilitating discussions among users with shared interests or purposes.

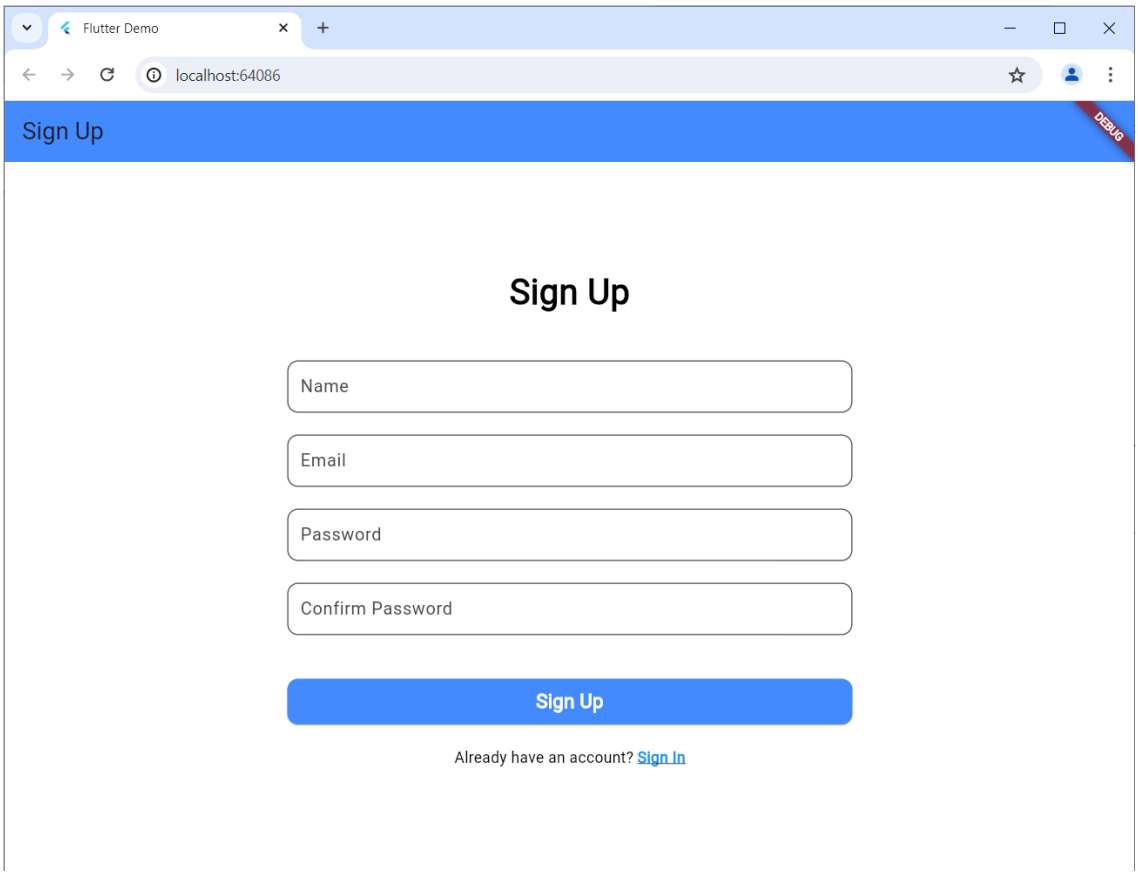
Design of your Application (If you are using database for record keeping, include ER diagram as well)

Design:

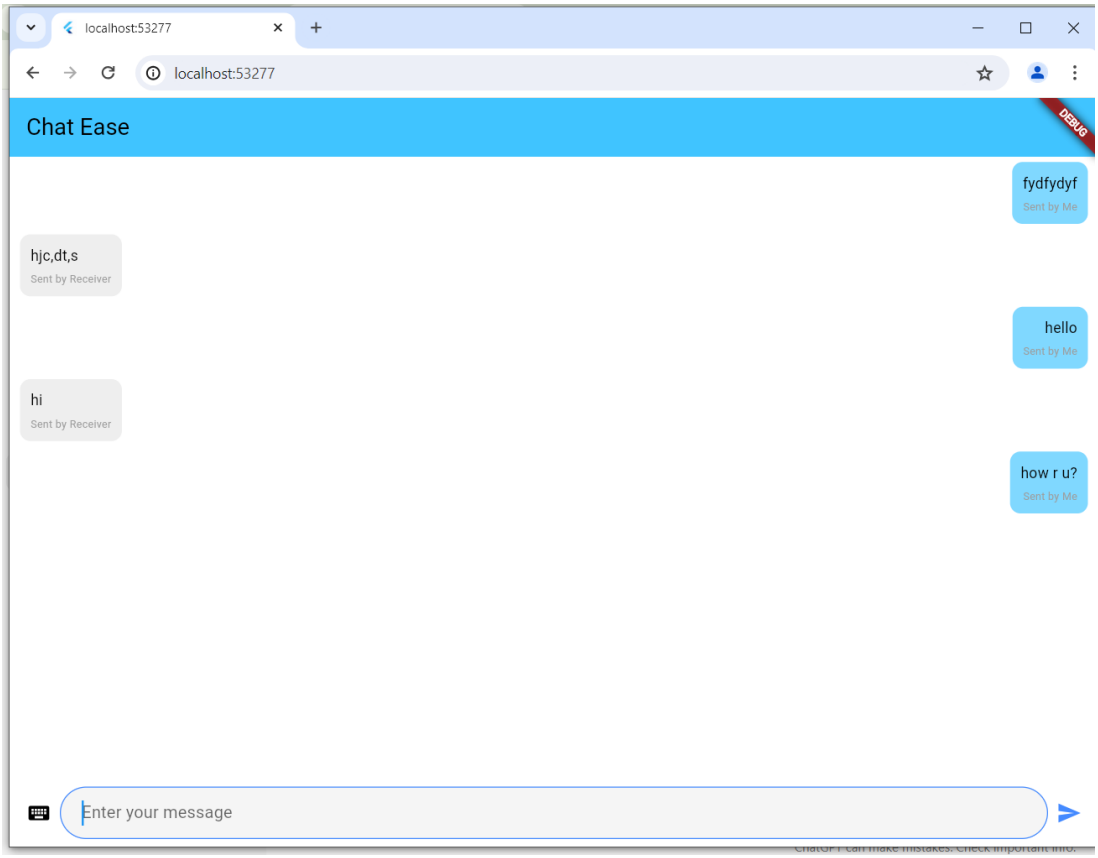
Signin:


Flutter Demo
localhost:64566
Sign In
DEBUG
Email
Password
Forgot Password?
Sign In

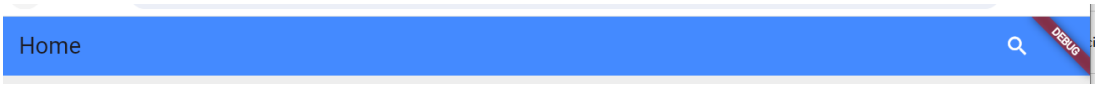
Signup:


Flutter Demo
localhost:64086
Sign Up
DEBUG
Sign Up
Name
Email
Password
Confirm Password
Sign Up
Already have an account? Sign In

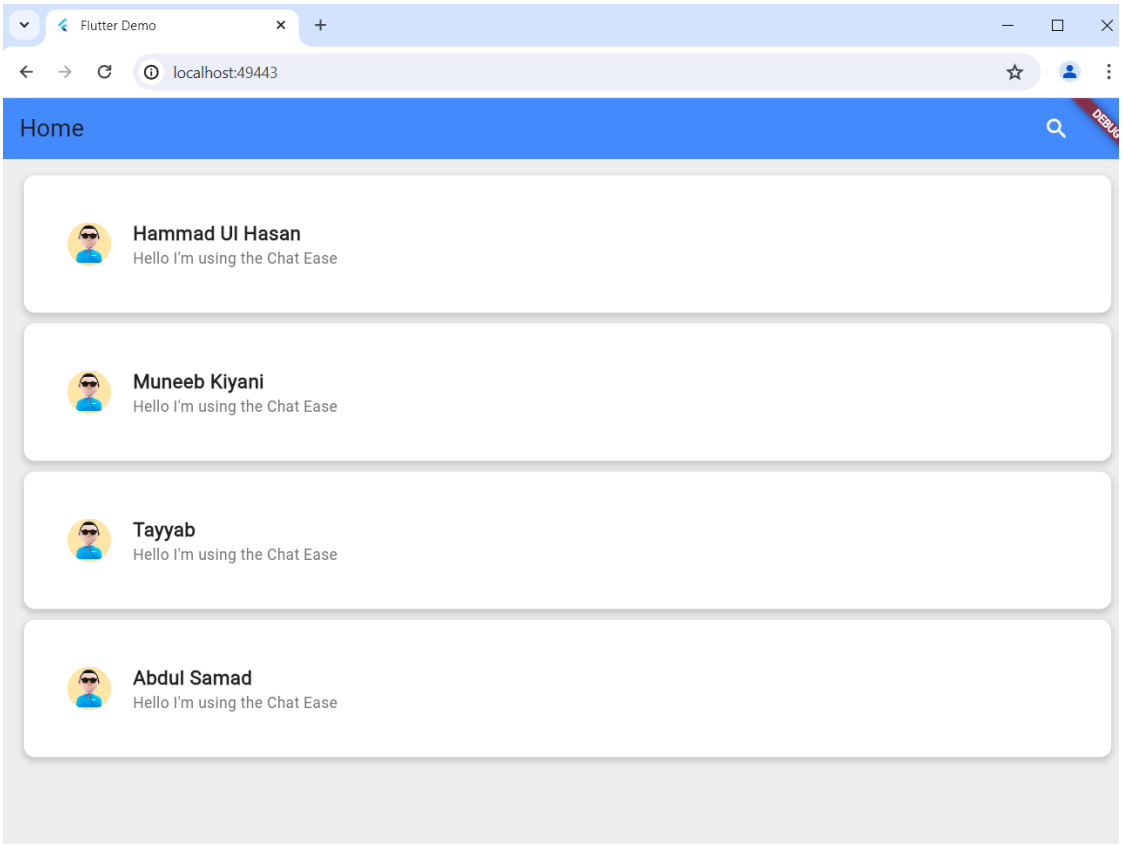
Chatpage:



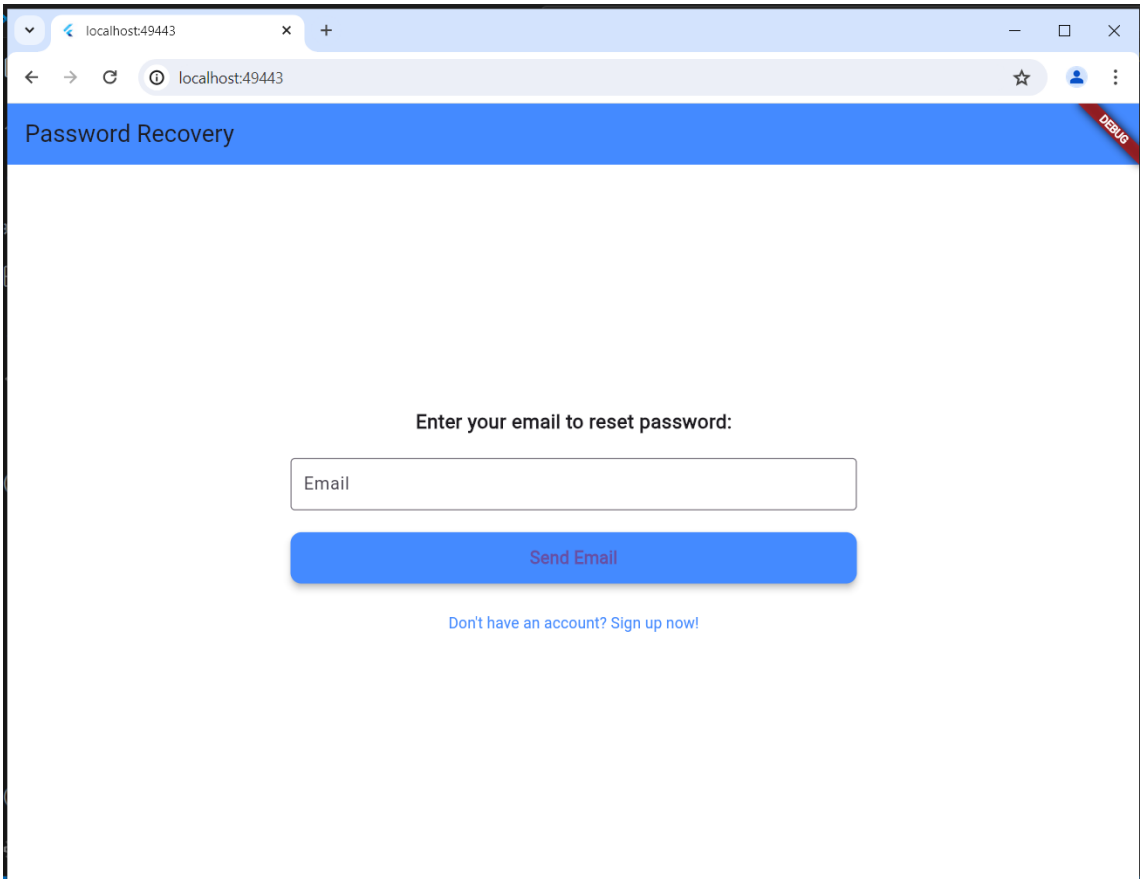
Advanced Search AI powered:



Homepage.dart:

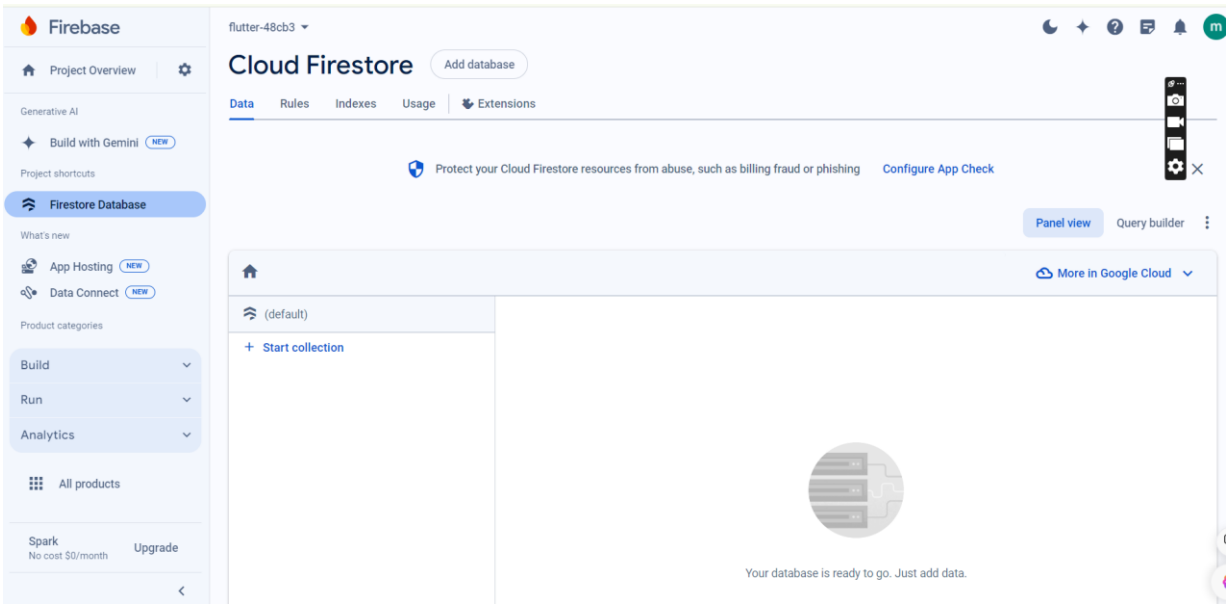


Password Recovery:



Firebase:

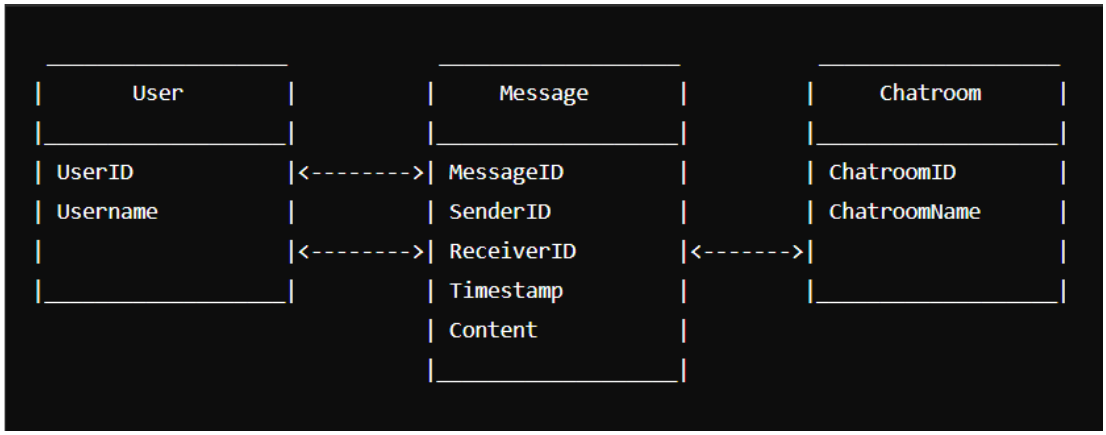
A Firebase database-powered chat application enables real-time communication between users by storing and synchronizing messages and user data across devices instantly. With Firebase, users can seamlessly send and receive messages, join chatrooms, and interact in a dynamic and responsive environment. The database handles data storage, retrieval, and synchronization, ensuring smooth communication experiences for users across platforms.




```
    at _engine.EnginePlatformDispatcher.new.invokeOnPointerDataPacket (http://localhost:62723/dart_sdk.js:178801:15)
    at [_sendToFramework] (http://localhost:62723/dart_sdk.js:180621:49)
    at _engine.ClickDebouncer.new.onPointerData (http://localhost:62723/dart_sdk.js:180532:31)
    at http://localhost:62723/dart_sdk.js:181157:24
    at http://localhost:62723/dart_sdk.js:181117:9
    at loggedHandler (http://localhost:62723/dart_sdk.js:180767:11)
    at Object._checkAndCall (http://localhost:62723/dart_sdk.js:12253:16)
    at Object.dcall (http://localhost:62723/dart_sdk.js:12258:17)
    at ret (http://localhost:62723/dart_sdk.js:68791:21)
Application finished.
PS C:\Users\atl\Desktop\assign\flutter_application_15>
```

bug my code | Search Error | Share Code Link | Explain Code | Comment Code | Find Bugs | Code Chat | Ln 200, Col 1 | Spaces: 2 | UTF-8

ER Diagram:



Complete code

Main. Dart

```dart
// ignore: unused_import
// ignore_for_file: unused_import, duplicate_ignore, depend_on_referenced_packages
import 'package:flutter_application_4/pages/home_page.dart';
import 'package:flutter_application_4/pages/sign_up.dart';
import 'package:flutter_application_4/service/auth.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}
class MyApp extends StatelessWidget
{
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
```

```
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home:
      FutureBuilder(future:Authmethods().getcurrentuser(), builder: (context,
AsyncSnapshot<dynamic> snapshot ){
        if(snapshot.hasData){
          return const  Home();
        }
        else{
          return const Signup();
        }
      }),
    );
  }
}
class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.
```

```dart
  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".
  final String title;
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;
  void _incrementCounter() {
    setState(() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
      // so that the display can reflect the updated values. If we changed
      // _counter without calling setState(), then the build method would not be
      // called again, and so nothing would appear to happen.
      _counter++;
    });
  }
  @override
  Widget build(BuildContext context) {
    // This method is rerun every time setState is called, for instance as done
    // by the _incrementCounter method above.
    //
    // The Flutter framework has been optimized to make rerunning build methods
    // fast, so that you can just rebuild anything that needs updating rather
    // than having to individually change instances of widgets.
```

```
    return Scaffold(
      appBar: AppBar(
        // TRY THIS: Try changing the color here to a specific color (to
        // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
        // change color while the other colors stay the same.
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        // Here we take the value from the MyHomePage object that was created by
        // the App.build method, and use it to set our appbar title.
        title: Text(widget.title),
      ),
      body: Center(
        // Center is a layout widget. It takes a single child and positions it
        // in the middle of the parent.
        child: Column(
          // Column is also a layout widget. It takes a list of children and
          // arranges them vertically. By default, it sizes itself to fit its
          // children horizontally, and tries to be as tall as its parent.
          //
          // Column has various properties to control how it sizes itself and
          // how it positions its children. Here we use mainAxisAlignment to
          // center the children vertically; the main axis here is the vertical
          // axis because Columns are vertical (the cross axis would be
          // horizontal).
          //
          // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
          // action in the IDE, or press "p" in the console), to see the
          // wireframe for each widget.
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
```

```
            const Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headlineMedium,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ), // This trailing comma makes auto-formatting nicer for build methods.
    );
  }
}
```

Signin:

```
// ignore_for_file: unused_import, unnecessary_new,
use_build_context_synchronously, avoid_unnecessary_containers,
sized_box_for_whitespace, prefer_const_constructors,
prefer_const_literals_to_create_immutables


import 'package:cloud_firestore/cloud_firestore.dart';

import 'package:firebase_auth/firebase_auth.dart';

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

```dart
import 'package:flutter_application_4/pages/home_page.dart';
import 'package:flutter_application_4/service/database.dart';
import 'package:flutter_application_4/service/shared_pref.dart';

class Signin extends StatefulWidget{
  const Signin({super.key});
  @override
  State<Signin> createState() => _SigninState();
}
class _SigninState extends State<Signin> {
  String email="",password="",name="",pic="",id="",username="";
  TextEditingController usermailcontroller=new TextEditingController();
  TextEditingController userpasswordcontroller=new TextEditingController();
  final _formkey=GlobalKey<FormState>();
  userlogin() async
  {
    try{
      await FirebaseAuth.instance.signInWithEmailAndPassword(email: email,
password: password);
      QuerySnapshot querySnapshot= await DatabaseMethods().getUserbyemail(email);
      name="${querySnapshot.docs[0] ["name"]}";
      username="${querySnapshot.docs[0] ["username"]}";
      pic="${querySnapshot.docs[0] ["photo"]}";
      id=querySnapshot.docs[0].id;

    await SharedPreferenceHelper().saveUserDisplayName(name);  //save the details
of a users.
    await SharedPreferenceHelper().saveUserName(username); //it saves the result
from the database.
    await SharedPreferenceHelper().saveUserId(id);
```

```
    await SharedPreferenceHelper().saveUserPic(pic);
      Navigator.pushReplacement(context, MaterialPageRoute(builder:
(context)=>Home(),),);
    } on FirebaseAuthException catch(e){
      if(e.code=='user-not-found'){
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          backgroundColor: Colors.redAccent,
          content: Text
        ("No user found for that email",
        style: TextStyle(fontSize: 18, color: Colors.black),)),);
      }
      else if(e.code=="wrong password"){
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          backgroundColor: Colors.redAccent,
          content: Text
        ("Wrong password provided by user",
        style: TextStyle(fontSize: 18, color: Colors.black),)),);
      }
    }

  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        child: Stack(
          children: [
            Container(
              height: MediaQuery.of(context).size.height / 4.0,
```

```
          width: MediaQuery.of(context).size.width,
          decoration: BoxDecoration(
              gradient: LinearGradient(
                colors: [Color(0xFF7F30fe), Color(0xFF6380fb)],
                begin: Alignment.topLeft,
                end: Alignment.bottomRight,
              ),
              borderRadius: BorderRadius.vertical(
                bottom: Radius.elliptical(
                    MediaQuery.of(context).size.width, 105),
              )),
        ),
        Padding(
          padding: const EdgeInsets.only(top: 70.0),
          child: Column(
            children: [
              Center(
                  child: Text(
                "Sign in",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 24.0,
                    fontWeight: FontWeight.bold),
              )),
              Center(
                  child: Text(
                "Login to your account",
                style: TextStyle(
                    color: Color.fromARGB(255, 106, 26, 135),
```

```
            fontSize: 18.0,
            fontWeight: FontWeight.w500),
      )),
      SizedBox(
        height: 20.0,
      ),
      Container(
        margin:
            EdgeInsets.symmetric(vertical: 20.0, horizontal: 20.0),
        child: Material(
          elevation: 5.0,
          borderRadius: BorderRadius.circular(10),
          child: Container(
            padding: EdgeInsets.symmetric(
                vertical: 30.0, horizontal: 20.0),
            height: MediaQuery.of(context).size.height / 2,
            width: MediaQuery.of(context).size.width,
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(10),
            ),
            child: Form(
              key: _formkey,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    "Email",
                    style: TextStyle(
```

```
      color: Colors.black,
      fontSize: 18.0,
      fontWeight: FontWeight.w500),
  ),
  Container(
    padding: EdgeInsets.only(left: 10.0),
    decoration: BoxDecoration(
        border: Border.all(
            width: 1.0, color: Colors.black38)),
    child: TextFormField(
      controller: usermailcontroller,
      validator: (value){
        if(value==null ||value.isEmpty)
        {
          return 'Plaese Enter The Email';

        }
        return null;

      },
      decoration: InputDecoration(
          border: InputBorder.none,
          prefixIcon: Icon(Icons.mail),
          prefixIconColor:
              Color.fromARGB(255, 106, 26, 135)),
    )),
  const SizedBox(
    height: 20.0,
  ),
```

```
Text(
  "Password",
  style: TextStyle(
      color: Colors.black,
      fontSize: 18.0,
      fontWeight: FontWeight.w500),
),
Container(
    padding: EdgeInsets.only(left: 10.0),
    decoration: BoxDecoration(
        border: Border.all(
            width: 1.0, color: Colors.black38)),
    child: TextFormField(
       controller: userpasswordcontroller,
      validator: (value){
        if(value==null ||value.isEmpty)
        {
          return 'Plaese Enter The Password';

        }
        return null;

      },
      decoration: InputDecoration(
          border: InputBorder.none,
          prefixIcon: Icon(Icons.password),
          prefixIconColor:
              Color.fromARGB(255, 106, 26, 135)),
      obscureText: true,
```

```
    )),
SizedBox(
  height: 20.0,
),
Container(
  alignment: Alignment.bottomRight,
  child: Text(
    "Forgot your password",
    style: TextStyle(
        color: Colors.black,
        fontSize: 16.0,
        fontWeight: FontWeight.w500),
  ),
),
SizedBox(
  height: 30.0,
),
GestureDetector(
  onTap: (){
    if(_formkey.currentState!.validate()){
      email=usermailcontroller.text;
      password=userpasswordcontroller.text;
    }
    userlogin();
  },
  child: Center(
    child: Container(
      width: 140,
      child: Material(
```

```
                                    elevation: 5.0,
                                    child: Center(
                                      child: Container(
                                        padding: EdgeInsets.all(10),
                                        decoration: BoxDecoration(
                                          color:
                                              Color.fromARGB(255, 106, 25, 135),
                                          borderRadius:
BorderRadius.circular(10),
                                        ),
                                        child: Center(
                                            child: Text(
                                          "Sign In",
                                          style: TextStyle(
                                              color: Colors.white,
                                              fontSize: 18.0,
                                              fontWeight: FontWeight.bold),
                                        )),
                                      ),
                                    ),
                                  ),
                                ),
                              ),
                            ),
                          ),
                        ],
                      ),
                    ),
                  ),
                ),
              ),
```

```
                ),
                const SizedBox(
                  height: 50.0,
                ),
                Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: const [
                    Text(
                      "Don't have an account! ",
                      style: TextStyle(color: Colors.black, fontSize: 16.0),
                    ),
                    Text(
                      "Sign Up Now",
                      style: TextStyle(
                          color: Colors.purple,
                          fontSize: 14.0,
                          fontWeight: FontWeight.bold),
                    ),
                  ],
                )
              ],
            ),
          )
        ],
      ),
    ),
  );
 }
}
```

Signup:

```dart
// ignore_for_file: unused_import, duplicate_ignore, unnecessary_null_comparison,
unused_local_variable, avoid_print, non_constant_identifier_names,
use_build_context_synchronously, avoid_unnecessary_containers,
use_super_parameters, prefer_const_constructors,
prefer_const_literals_to_create_immutables


import 'package:flutter/material.dart';

import 'package:firebase_auth/firebase_auth.dart';

import 'package:flutter_application_4/pages/home_page.dart';

import 'package:flutter_application_4/service/database.dart';

import 'package:flutter_application_4/service/shared_pref.dart';

// ignore: unused_import

import 'package:shared_preferences/shared_preferences.dart';

import 'package:random_string/random_string.dart';


void main() {

  String id = randomAlphaNumeric(10);

  print(id); // Print the generated random alphanumeric string

}


class Signup extends StatefulWidget {

  const Signup({Key? key}) : super(key: key);

  @override

  State<Signup> createState() => _SignupState();

}
```

```dart
class _SignupState extends State<Signup> {
  String email = "", password = "", name = "", confirmpassword = "";
  TextEditingController mailcontroller = TextEditingController();
  TextEditingController passwordcontroller = TextEditingController();
  TextEditingController namecontroller = TextEditingController();
  TextEditingController confirmpasswordcontroller = TextEditingController();
  final _formkey = GlobalKey<FormState>();

  void registration() async {
    if (_formkey.currentState!.validate()) {
      if (password != null && password == confirmpassword) {
        try {
          UserCredential userCredential = await FirebaseAuth.instance
              .createUserWithEmailAndPassword(email: email, password: password);
          String Id = randomAlphaNumeric(10);
          String user = mailcontroller.text.replaceAll("@gmail.com", "");
          String updateusername = user.replaceAll(user[0], user[0].toUpperCase());
          String firstletter = user.substring(0, 1).toUpperCase();

          Map<String, dynamic> userinfoMap = {
            "name": namecontroller.text,
            "E-mail": mailcontroller.text,
            "Username": updateusername.toUpperCase(),
            "SearchKey": firstletter,
            "photo":
                "https://www.freepik.com/icon/user_1077114#fromView=keyword&page=
1&position=0",
            "id": Id,
          };
```

```
          await DatabaseMethods().addUserDetails(userinfoMap, Id);
          await SharedPreferenceHelper().saveUserId(Id);
          await
SharedPreferenceHelper().saveUserDisplayName(namecontroller.text);
          await SharedPreferenceHelper().saveUserEmail(mailcontroller.text);
          await SharedPreferenceHelper().saveUserPic(
              "https://www.freepik.com/icon/user_1077114#fromView=keyword&page=1&
position=0");
          await SharedPreferenceHelper()
              .saveUserName(mailcontroller.text.replaceAll("Gmail.com",
"").toUpperCase());

          ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            content: const Text(
              "Registered Successfully",
              style: TextStyle(fontSize: 20.0),
            ),
          ));

          Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => Home()),
          );
        } on FirebaseAuthException catch (e) {
          if (e.code == 'weak-password') {
            ScaffoldMessenger.of(context).showSnackBar(SnackBar(
              backgroundColor: Colors.orangeAccent,
              content: Text(
                "Password provided is too weak",
```

```
              style: TextStyle(fontSize: 18.0),
            ),
          ));
        } else if (e.code == 'email-already-in-use') {
          ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "Account already exists",
              style: TextStyle(fontSize: 18.0),
            ),
          ));
        }
      }
    }
  }
}
@override
Widget build(BuildContext context)
{
  return Scaffold(
    body: Container(
      child: Stack(
        children: [
          Container(
            height: MediaQuery.of(context).size.height / 4.0,
            width: MediaQuery.of(context).size.width,
            decoration: BoxDecoration(
                gradient: LinearGradient(
                  colors: [Color(0xFF7F30fe), Color(0xFF6380fb)],
```

```
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
          borderRadius: BorderRadius.vertical(
            bottom: Radius.elliptical(
                MediaQuery.of(context).size.width, 105),
          )),
      ),
      Padding(
        padding: const EdgeInsets.only(top: 70.0),
        child: Column(
          children: [
            Center(
                child: Text(
              "Sign up",
              style: TextStyle(
                  color: Colors.white,
                  fontSize: 24.0,
                  fontWeight: FontWeight.bold),
            )),
            Center(
                child: Text(
              "Create a new account",
              style: TextStyle(
                  color: Color.fromARGB(255, 106, 26, 135),
                  fontSize: 18.0,
                  fontWeight: FontWeight.w500),
            )),
            SizedBox(
```

```
    height: 15.0,
  ),
  Container(
    margin:
        EdgeInsets.symmetric(vertical: 20.0, horizontal: 20.0),
    child: Material(
      elevation: 5.0,
      borderRadius: BorderRadius.circular(10),
      child: Container(
        padding: EdgeInsets.symmetric(
            vertical: 30.0, horizontal: 20.0),
        height: MediaQuery.of(context).size.height / 1.56,
        width: MediaQuery.of(context).size.width,
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(10),
        ),
        key: _formkey,
        child: Form(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                "Name",
                style: TextStyle(
                    color: Colors.black,
                    fontSize: 18.0,
                    fontWeight: FontWeight.w500),
              ),
```

```
                        Container(
                            padding: EdgeInsets.only(left: 10.0),
                            decoration: BoxDecoration(
                                border: Border.all(
                                    width: 1.0, color: Colors.black38)),
                            child: TextFormField(
                              controller: namecontroller,
                              validator: (value) {
                                if (value == null || value.isEmpty) {
                                  return 'Please enter name';
                                }
                                return null;
                              },
                              decoration: InputDecoration(
                                  border: InputBorder.none,
                                  prefixIcon: Icon(Icons.person),
                                  prefixIconColor:
                                      Color.fromARGB(255, 106, 26, 135)),
                            )),
                        const SizedBox(
                          height: 15.0,
                        ),
                        Text(
                          "Email",
                          style: TextStyle(
                              color: Colors.black,
                              fontSize: 18.0,
                              fontWeight: FontWeight.w500),
                        ),
```

```
Container(
    padding: EdgeInsets.only(left: 10.0),
    decoration: BoxDecoration(
        border: Border.all(
            width: 1.0, color: Colors.black38)),
    child: TextFormField(
      controller: namecontroller,
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please enter E-mail';
        }
        return null;
      },
      decoration: InputDecoration(
          border: InputBorder.none,
          prefixIcon: Icon(Icons.password),
          prefixIconColor:
              Color.fromARGB(255, 106, 26, 135)),
      obscureText: true,
    )),
SizedBox(
  height: 15.0,
),
Text(
  "Password",
  style: TextStyle(
      color: Colors.black,
      fontSize: 18.0,
      fontWeight: FontWeight.w500),
```

```
),
Container(
    padding: EdgeInsets.only(left: 10.0),
    decoration: BoxDecoration(
        border: Border.all(
            width: 1.0, color: Colors.black38)),
    child: TextFormField(
      controller: passwordcontroller,
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please enter password';
        }
        return null;
      },
      decoration: InputDecoration(
          border: InputBorder.none,
          prefixIcon: Icon(Icons.mail),
          prefixIconColor:
              Color.fromARGB(255, 106, 26, 135)),
    )),
SizedBox(
  height: 15.0,
),
Text(
  "Confirm Password",
  style: TextStyle(
      color: Colors.black,
      fontSize: 18.0,
      fontWeight: FontWeight.w500),
```

```
),
Container(
    padding: EdgeInsets.only(left: 10.0),
    decoration: BoxDecoration(
        border: Border.all(
            width: 1.0, color: Colors.black38)),
    child: TextFormField(
      controller: confirmpasswordcontroller,
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please confirm password';
        }
        return null;
      },
      decoration: InputDecoration(
          border: InputBorder.none,
          prefixIcon: Icon(Icons.mail),
          prefixIconColor:
              Color.fromARGB(255, 106, 26, 135)),
    )),
SizedBox(
  height: 15.0,
),
GestureDetector(
  onTap: () {
    if (_formkey.currentState!.validate()) {
      setState(() {
        email = mailcontroller.text;
        name = namecontroller.text;
```

```
            password = passwordcontroller.text;
            confirmpassword =
                confirmpasswordcontroller.text;
          });
        }
        registration();
      },
      child: Center(
        child: Container(
          margin:
              EdgeInsets.symmetric(horizontal: 20.0),
          width:
              MediaQuery.of(context).size.width / 1.0,
          child: Material(
            elevation: 5.0,
            child: Center(
              child: Container(
                padding: EdgeInsets.all(10),
                decoration: BoxDecoration(
                  color: Color.fromARGB(
                      255, 106, 25, 135),
                  borderRadius:
                      BorderRadius.circular(10),
                ),
                child: Center(
                    child: Text(
                  "Sign Up",
                  style: TextStyle(
                      color: Colors.white,
```

```
                                    fontSize: 18.0,
                                    fontWeight: FontWeight.bold),
                                ),
                            ),
                        ),
                    ),
                ),
            ),
        ),
      ],
    ),
  ),
),
const SizedBox(
  height: 20.0,
),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: const [
    Text(
      "Don't have an account! ",
      style: TextStyle(color: Colors.black, fontSize: 16.0),
    ),
    Text(
      "Sign Up Now",
      style: TextStyle(
          color: Colors.purple,
```

```
                            fontSize: 14.0,
                            fontWeight: FontWeight.bold),
                      ),
                    ],
                  )
                ],
              ),
            )
          ],
        ),
      ),
    );
  }
}
```

Homepage.dart:

```
// ignore_for_file: non_constant_identifier_names, prefer_const_constructors,
use_build_context_synchronously, sized_box_for_whitespace,
prefer_const_constructors_in_immutables, avoid_unnecessary_containers,
unused_import, duplicate_ignore, unused_local_variable, unnecessary_import,
unused_label, unnecessary_string_escapes, prefer_is_empty,
avoid_function_literals_in_foreach_calls


import 'dart:collection';
import 'dart:ui';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:flutter/painting.dart';
```

```dart
import 'package:flutter_application_4/pages/chat_page.dart';
import 'package:flutter_application_4/service/database.dart';
import 'package:flutter_application_4/service/shared_pref.dart';
class Home extends StatefulWidget {
  const Home({super.key});
  @override
  State<Home> createState() => _HomeState();
}
class _HomeState extends State<Home> {
  bool search=false;
  String? myName,myProfilePic, myUserName, myEmail;
  String? chatRoomStream;
  Stream? chatRoomsStream;
  Future<void> getTheSharedPref() async {


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


    });


  }
  Widget ChatRoomList(){
    return StreamBuilder(
      stream: chatRoomsStream,
      builder: (
```

```
        context,AsyncSnapshot snapshot){
      return snapshot.hasData?ListView.builder(
        padding: EdgeInsets.zero,
        itemCount: snapshot.data.docs.length,
        itemBuilder: (context, index){
        DocumentSnapshot ds=snapshot.data.doc.length;
        shrinkWrap: true;
        return _ChatRoomListTile(lastmessage: ds["lastMesage"], chatRoomId: ds.id,
myUsername: myUserName!, time: ds["lastMessageSendTs"]);
      // ignore: prefer_const_constructors
      }): Center(child: CircularProgressIndicator(),);
    });
  }
  @override
    void initState() {
      super.initState();
      ontheload();
    }
  getChatRoomIdbyUsername(String a, String b){
    if(a.substring(0,1).codeUnitAt(0)>b.substring(0,1).codeUnitAt(0)){
      return"$b\_$a";
    }else{
      return"$a\_$b";
    }
  }
  var querrResultset=[];
  var tempSearchStore=[];
  intiateSearch(value)
  {
```

```
if(value.length==0){
  setState(() {
    querrResultset=[];
    tempSearchStore=[];
  });
}
setState(() {
  search=true;
});
var capitalizedvalue= value.substring(0,1).toUpperCase()+value.substring(1);
if(querrResultset.length==0 && value.length==1)
{
  DatabaseMethods().Search(value).then((QuerySnapshot docs){
    for(int i=0; i<docs.docs.length; i++){
      querrResultset.add(docs.docs[i].data());
    }
  });
}
else{
  tempSearchStore=[];
  querrResultset.forEach((element){
    if(element['username'].startsWith(capitalizedvalue)){
      setState((){
        tempSearchStore.add(element);

      });
    }
  }
);
```

```
    }
    }
  @override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color.fromARGB(255, 174, 149, 241),
    body: Container(
      child: Column(
        children: [
          Padding(
            padding: const EdgeInsets.only(left: 20.0, right: 20.0, top: 50.0,
bottom: 20.0),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                search
                    ? Expanded(
                        child: TextField(
                          onChanged: (value) {
                            intiateSearch(value.toUpperCase());
                          },
                          decoration: InputDecoration(
                            border: InputBorder.none,
                            hintText: 'Search User',
                            hintStyle: TextStyle(color: Colors.black12, fontSize:
20.0, fontWeight: FontWeight.w500),
                          ),
                          style: TextStyle(color: Colors.black12, fontSize: 18.0,
fontWeight: FontWeight.bold),
                        ),
```

```
                )
              : Text(
                  "ChatUp",
                  style: TextStyle(
                    color: Color.fromARGB(255, 17, 4, 19),
                    fontSize: 20.0,
                    fontWeight: FontWeight.bold,
                  ),
                ),
            GestureDetector(
              onTap: () {
                setState(() {
                  search = !search;
                });
              },
              child: Container(
                padding: EdgeInsets.all(8),
                decoration: BoxDecoration(
                  color: Color.fromARGB(255, 228, 249, 249),
                  borderRadius: BorderRadius.circular(20),
                ),
                child: Icon(
                  search ? Icons.close : Icons.search,
                  color: search ? Colors.deepOrange : Colors.deepPurple,
                ),
              ),
            ),
          ],
        ),
```

```
        ),
        Container(
          padding: EdgeInsets.symmetric(horizontal: 30.0, vertical: 20.0),
          width: MediaQuery.of(context).size.width,
          height: search
              ? MediaQuery.of(context).size.height / 1.19
              : MediaQuery.of(context).size.height / 1.18,
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.only(
              topLeft: Radius.circular(20),
              topRight: Radius.circular(20),
            ),
          ),
          child: Column(
            children: [
              if (search)
                Expanded(
                  child: ListView(
                    padding: EdgeInsets.only(left: 10.0, right: 10.0),
                    primary: false,
                    shrinkWrap: true,
                    children: tempSearchStore.map((element) {
                      return buildResultCard(element);
                    }).toList(),),
                ),
                 ChatRoomList(),
      ],
    ),
```

```
    ),
  ]));
}
Widget buildResultCard(Map data){
  return GestureDetector(
    onTap: ()async{
      search=false;
      setState(() {


      });
      var chatRoomId= getChatRoomIdbyUsername(myUserName!,data["username"]);
      Map<String, dynamic>chatRoomInfoMap={
        "users":[myUserName, data["username"]],
      };
      await DatabaseMethods().createChatRoom(chatRoomId, chatRoomInfoMap);
      Navigator.push(context, MaterialPageRoute(builder:
(context)=>chat_page(name: data["Name"], profileurl: data["Photo"], username:
data["username"])));


    },
    child: Container(
      margin: EdgeInsets.symmetric(vertical: 8),
      child: Material(
        elevation: 5.0,
        borderRadius: BorderRadius.circular(10),
        child: Container(
          padding: EdgeInsets.all(18),
          decoration: BoxDecoration(
            color: Colors.white24, // Changed 'colors' to 'Colors'
            borderRadius: BorderRadius.circular(10)
```

```
          ),
          child: Row(children:
          [
            ClipRRect(
              borderRadius: BorderRadius.circular(60),
            child: Image.network(data["Photo"], height: 70, width: 70, fit:
BoxFit.cover,)),
            SizedBox(width: 10.0,),
            Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  data["Name"], // Ensure 'data' contains a 'username' key
                  style: TextStyle(color: Colors.black, // Changed 'colors' to
'Colors'
                   fontWeight: FontWeight.bold,
                   fontSize: 18.0),
                   ),
                   SizedBox(
                    height: 8.0,
                   ),
                   Text(data["Username"],
                   style: TextStyle(color: Colors.black,
                   fontWeight: FontWeight.bold,
                   fontSize: 15.0)
                   ),
                   SizedBox(
                    height: 10.0,
                   ),
```

```
                  Text(data["username"], style: TextStyle(color: Colors.black12,
fontSize: 15.0, fontWeight: FontWeight.bold),),

              ],

            ),

          ],),

        ),

      ),

    ),

  );

}

}


chat_page({required name, required profileurl, required username}) {

}

class _ChatRoomListTile extends StatefulWidget {

  final String  lastmessage, chatRoomId, myUsername, time;

  _ChatRoomListTile({required this.lastmessage, required this.chatRoomId, required
this.myUsername,required this.time});


  @override

  State<_ChatRoomListTile> createState() => __ChatRoomListTileState();

}

class __ChatRoomListTileState extends State<_ChatRoomListTile> {

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

Future<void> getThisUserInfo() async {

  String username = widget.chatRoomId.replaceAll("_", "
").replaceAll(widget.myUsername, "");

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

  String name = querySnapshot.docs[0]["Name"];
```

```
  String profilePicUrl = querySnapshot.docs[0]["Photo"];
  String id = querySnapshot.docs[0]["id"];
  // Removed the redundant name assignment with an empty key
  setState(() {
    // Update the state with the fetched information if needed
  });
}
@override
void initState() {
  super.initState(); // super.initState() should be called at the beginning of initState
  getThisUserInfo();
}

  @override
Widget build(BuildContext context) {
  return Container(
    margin: EdgeInsets.symmetric(horizontal: 20.0, vertical: 20.0),
    child: Row(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        profilePicUrl == ""
            ? CircularProgressIndicator()
            : ClipRRect(
                borderRadius: BorderRadius.circular(60),
                child: Image.network(
                  profilePicUrl,
                  height: 70.0,
                  width: 70.0,
```

```
            fit: BoxFit.cover,
          ),
        ),
      SizedBox(
        width: 20.0,
      ),
      Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          SizedBox(
            height: 10.0,
          ),
          Text(
            username,
            style: TextStyle(
              color: Colors.black,
              fontSize: 18.0,
              fontWeight: FontWeight.bold,
            ),
          ),
          Container(
            width: MediaQuery.of(context).size.width / 2,
            child: Text(
              widget.lastmessage,
              overflow: TextOverflow.ellipsis,
              style: TextStyle(
                color: Color.fromARGB(255, 42, 36, 36),
                fontSize: 12.0,
                fontWeight: FontWeight.bold,
```

```
                ),
              ),
            ),
            Spacer(),
            Text(
              widget.time,
              style: TextStyle(
                color: Color.fromARGB(255, 42, 36, 36),
                fontSize: 12.0,
                fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
      ],
    ),
  );
}
}
```

Chatpage.dart:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:flutter_application_4/pages/home_page.dart';
import 'package:flutter_application_4/service/database.dart';
import 'package:intl/intl.dart';
import 'package:random_string/random_string.dart';
```

```
class Chatpage extends StatefulWidget {
  String name, profileurl, username;
  Chatpage({required this.name, required this.profileurl, required this.username});

  @override
  State<Chatpage> createState() => _ChatpageState();
}

class _ChatpageState extends State<Chatpage> {
  TextEditingController messagecontroller = new TextEditingController();
  String? myUserName, myProfilePic, myName, myEmail, messageId, chatRoomId;
  Stream? messageStream;

  getChatRoomIdbyUsername(String a, String b) {
    if (a.substring(0, 1).codeUnitAt(0) > b.substring(0, 1).codeUnitAt(0)) {
      return "$b\_$a";
    } else {
      return "$a\_$b";
    }
  }

  Widget chatMessageTile(String message, bool sendByMe) {
    return Row(
      mainAxisAlignment: sendByMe ? MainAxisAlignment.end :
MainAxisAlignment.start,
      children: [
        Flexible(
          child: Container(
```

```
          padding: EdgeInsets.all(16),
          margin: EdgeInsets.symmetric(horizontal: 16, vertical: 4),
          decoration: BoxDecoration(
            borderRadius: BorderRadius.only(
              topLeft: Radius.circular(24),
              bottomRight: sendByMe ? Radius.circular(0) : Radius.circular(24),
              topRight: Radius.circular(24),
              bottomLeft: sendByMe ? Radius.circular(24) : Radius.circular(0)),
            color: sendByMe
                ? Color.fromARGB(255, 234, 236, 240)
                : Color.fromARGB(255, 211, 228, 243)),
          child: Text(
            message,
            style: TextStyle(
                color: Colors.black,
                fontSize: 15.0,
                fontWeight: FontWeight.bold),
          ),
        ),
      ),
    ],
  );
}

@override
Widget build(BuildContext context) {
  var messagecontroller;
  return Scaffold(
    backgroundColor: Color.fromARGB(255, 227, 141, 251),
```

```
      body: Container(
        margin: EdgeInsets.only(top: 60.0, left: 20.0, right: 20.0),
        child: Stack(
          children: [
            Container(
              margin: EdgeInsets.only(top: 50.0),
              width: MediaQuery.of(context).size.width,
              height: MediaQuery.of(context).size.height / 1.10,
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.only(
                  topLeft: Radius.circular(30),
                  topRight: Radius.circular(30),
                ),
              ),
              child: chatMessage(),
            ),
            Padding(
              padding: const EdgeInsets.only(left: 10.0),
              child: Row(
                children: [
                  GestureDetector(
                    onTap: () {
                      Navigator.pushReplacement(
                          context, MaterialPageRoute(builder: (context) =>
Home()));
                    },
                    child: Icon(Icons.arrow_back_ios_new_outlined)),
                  SizedBox(
```

```
                width: 40.0,
              ),
              Text(
                widget.name,
                style: TextStyle(
                    color: Colors.black,
                    fontSize: 18.0,
                    fontWeight: FontWeight.w500),
              )
            ],
          ),
        ),
        Container(
          margin: EdgeInsets.only(left: 20.0, right: 20.0, bottom: 20.0),
          alignment: Alignment.bottomCenter,
          child: Material(
            elevation: 5.0,
            borderRadius: BorderRadius.circular(10),
            child: Container(
              padding: EdgeInsets.all(10.0),
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.circular(30),
              ),
              child: TextField(
                controller: messagecontroller,
                decoration: InputDecoration(
                  border: InputBorder.none,
                  hintText: "Type A Message!",
```

```
                    hintStyle: TextStyle(color: Colors.black38),
                    suffixIcon: GestureDetector(
                      onTap: () {
                        addmessage(true);
                      },
                      child: Icon(Icons.send_rounded))
                  ),
                ),
              ),
            ),
          ),
        ],
      ),
    ),
  );
 }

 chatMessage() {}

 void addmessage(bool bool) {}
}
```