

FLASHCHAT – AN ONLINE CHATTING APPLICATION

Project Report Submitted in partial fulfilment of the

Requirements for the award of credits for

Mobile Application Development Laboratory Course (19CS3751)

in

IV Year I Semester

By

A. Yaswanth (19501A0505)

B. Deva Karthik (19501A0518)

K. Bhanu Prasanna (19501A0527)

B. Satish (19501A0530)

Submitted to

Mr. B. Vishnu Vardhan (Ph. D)

Assistant Professor



Department of Computer Science and Engineering

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Approved by AICTE and NBA Accredited)

Kanuru, Vijayawada, AP – 520007

(2019 - 23)

PROJECT ABSTRACT

Project Abstract:

The ease of real-time chat communication and messaging's user-friendly capabilities have contributed to its prevalence in our daily lives. With a chat app, users are able to receive the same engaging and lively interactions through custom messaging features, just as they would in person. Sending and receiving messages in real time using a messaging service facilitates communication with individuals located anywhere in the world. **FlashChat** is an online chatting application which enables group of users to chat on a single channel. It provides a fast and interactive way to send and receive messages. The users can register with their email id and login instantly without any latency.

Strategy:

We intend to develop Chatting App for Mobile Platform. The development of the application is done using **Flutter** – an open-source UI software development kit developed by Google. As the development is done in Flutter, the application can be used cross-platform for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase. **Firebase Authentication** is used to authenticate the users via a registration page. The authentication is done using an email and password option. As soon as the users register, it brings them to the chat page where they can send messages and emojis. Registered users can login via a login page. The messages are stored in a cloud database so that other users can receive messages remotely. Google Firebase **Cloud Firestore** database is used to exchange messages that are stored as a key-value pairs in the database. It contains fields such as, sender, text, and timestamp.

Further Plans:

This app is deployed locally and we intend to deploy the application in Google Play Store and iOS App Store. The application only has Email/Password as a Sign-in method but for future use, we intend to give more options for the users, such as phone number, google OAuth, twitter, etc. We also intend to develop individual channels for the user end-to-end communication.

CODE

The app is started using **flutter run** command.

main.dart:

```
import 'package:flutter/material.dart';
import 'package:flash_chat/screens/welcome_screen.dart';
import 'package:flash_chat/screens/login_screen.dart';
import 'package:flash_chat/screens/registration_screen.dart';
import 'package:flash_chat/screens/chat_screen.dart';
import 'package:firebase_core/firebase_core.dart';
```

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(FlashChat());
}
```

```
class FlashChat extends StatelessWidget {
  static const String id = 'login_screen';

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      initialRoute: WelcomeScreen.id,
      routes: {
        WelcomeScreen.id: (context) => WelcomeScreen(),
        LoginScreen.id: (context) => LoginScreen(),
        RegistrationScreen.id: (context) => RegistrationScreen(),
        ChatScreen.id: (context) => ChatScreen(),
      },
    );
  }
}
```

```
}  
}
```

welcome_screen.dart:

```
import 'package:flash_chat/screens/login_screen.dart';  
import 'package:flash_chat/screens/registration_screen.dart';  
import 'package:flutter/material.dart';  
import 'package:animated_text_kit/animated_text_kit.dart';  
import 'package:flash_chat/components/rounded_button.dart';  
  
class WelcomeScreen extends StatefulWidget {  
  static const String id = 'welcome_screen';  
  
  @override  
  _WelcomeScreenState createState() => _WelcomeScreenState();  
}  
  
class _WelcomeScreenState extends State<WelcomeScreen>  
  with SingleTickerProviderStateMixin {  
  AnimationController controller;  
  Animation animation;  
  
  @override  
  void initState() {  
    super.initState();  
  
    controller = AnimationController(  
      duration: Duration(seconds: 1),  
      vsync: this,  
    );  
    animation = ColorTween(begin: Colors.blueGrey, end: Colors.white)  
      .animate(controller);
```

```
controller.forward();
controller.addListener(() {
  setState(() {});
});
}
```

```
@override
void dispose() {
  controller.dispose();
  super.dispose();
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: animation.value,
    body: Padding(
      padding: EdgeInsets.symmetric(horizontal: 24.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: <Widget>[
          Row(
            children: <Widget>[
              Hero(
                tag: 'logo',
                child: Container(
                  child: Image.asset('images/logo.png'),
                  height: 60.0,
                ),
              ),
            ],
          ),
        ],
      ),
    ),
  );
}
```

```
AnimatedTextKit(animatedTexts: [
  TypewriterAnimatedText(
    'Flash Chat',
    speed: Duration(milliseconds: 600),
    textStyle: TextStyle(
      fontSize: 45.0,
      fontWeight: FontWeight.w900,
      color: Colors.black,
    ),
  ),
]),
],
),
 SizedBox(
  height: 48.0,
),
 RoundedButton(
  title: 'Log In',
  color: Colors.lightBlueAccent,
  onPressed: () {
    Navigator.pushNamed(context, LoginScreen.id);
  },
),
 RoundedButton(
  title: 'Register',
  color: Colors.blueAccent,
  onPressed: () {
    Navigator.pushNamed(context, RegistrationScreen.id);
  },
),
],
```

```
    ),  
    ),  
  );  
}  
}
```

login_screen.dart:

```
import 'package:flash_chat/components/rounded_button.dart';  
import 'package:flash_chat/screens/chat_screen.dart';  
import 'package:flutter/material.dart';  
import 'package:flash_chat/constants.dart';  
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:modal_progress_hud/modal_progress_hud.dart';
```

```
class LoginScreen extends StatefulWidget {  
  static const String id = 'login_screen';  
  
  @override  
  _LoginScreenState createState() => _LoginScreenState();  
}
```

```
class _LoginScreenState extends State<LoginScreen> {  
  final _auth = FirebaseAuth.instance;
```

```
  bool showSpinner = false;
```

```
  String email;  
  String password;
```

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

```

```
backgroundColor: Colors.white,
body: ModalProgressHUD(
  inAsyncCall: showSpinner,
  child: Padding(
    padding: EdgeInsets.symmetric(horizontal: 24.0),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: <Widget>[
        Flexible(
          child: Hero(
            tag: 'logo',
            child: Container(
              height: 200.0,
              child: Image.asset('images/logo.png'),
            ),
          ),
        ),
        ),
        SizedBox(
          height: 48.0,
        ),
        TextField(
          keyboardType: TextInputType.emailAddress,
          textAlign: TextAlign.center,
          onChanged: (value) {
            email = value;
          },
          decoration:
            kTextFieldDecoration.copyWith(hintText: 'Enter your email'),
        ),
        SizedBox(
```



```

        height: 8.0,
      ),
      TextField(
        obscureText: true,
        textAlign: TextAlign.center,
        onChanged: (value) {
          password = value;
        },
        decoration: kTextFieldDecoration.copyWith(
          hintText: 'Enter your password'),
      ),
      SizedBox(
        height: 24.0,
      ),
      RoundedButton(
        color: Colors.lightBlueAccent,
        title: 'Log In',
        onPressed: () async {
          setState(() {
            showSpinner = true;
          });

          try {
            final user = _auth.signInWithEmailAndPassword(
              email: email, password: password);

            if (user != null) {
              Navigator.pushNamed(context, ChatScreen.id);
            }
          }
          setState(() {
            showSpinner = false;
          });
        },
      ),
    ),
  ),
);

```

```

        });
      } catch (e) {
        print("Wrong" + e);
      }
    }},
  ],
),
),
),
);
}
}

```

registration_screen.dart:

```

import 'package:flash_chat/constants.dart';
import 'package:flash_chat/screens/chat_screen.dart';
import 'package:flutter/material.dart';
import 'package:flash_chat/components/rounded_button.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:modal_progress_hud/modal_progress_hud.dart';

class RegistrationScreen extends StatefulWidget {
  static const String id = 'registration_screen';

  @override
  _RegistrationScreenState createState() => _RegistrationScreenState();
}

class _RegistrationScreenState extends State<RegistrationScreen> {
  final _auth = FirebaseAuth.instance;

  bool showSpinner = false;

```

String email;

String password;

@override

```
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: Colors.white,  
    body: ModalProgressHUD(  
      inAsyncCall: showSpinner,  
      child: Padding(  
        padding: EdgeInsets.symmetric(horizontal: 24.0),  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          crossAxisAlignment: CrossAxisAlignment.stretch,  
          children: <Widget>[  
            Flexible(  
              child: Hero(  
                tag: 'logo',  
                child: Container(  
                  height: 200.0,  
                  child: Image.asset('images/logo.png'),  
                ),  
              ),  
            ),  
            SizedBox(  
              height: 48.0,  
            ),  
            TextField(  
              keyboardType: TextInputType.emailAddress,  
              textAlign: TextAlign.center,
```

```
onChanged: (value) {  
  email = value;  
},  
decoration:  
  kTextFieldDecoration.copyWith(hintText: 'Enter your email'),  
,  
SizedBox(  
  height: 8.0,  
,  
TextField(  
  obscureText: true,  
  textAlign: TextAlign.center,  
  onChanged: (value) {  
    password = value;  
  },  
  decoration: kTextFieldDecoration.copyWith(  
    hintText: 'Enter your password'),  
,  
SizedBox(  
  height: 24.0,  
,  
RoundedButton(  
  color: Colors.blueAccent,  
  title: 'Register',  
  onPressed: () async {  
    setState(() {  
      showSpinner = true;  
    });  
  
    // print(email);  
    // print(password);
```

```

    try {
      final newUser =
        await _auth.createUserWithEmailAndPassword(
          email: email, password: password);
      if (newUser != null) {
        Navigator.pushNamed(context, ChatScreen.id);
      }
      setState(() {
        showSpinner = false;
      });
    } catch (e) {
      print(e);
    }
  })
],
),
),
),
);
}
}

```

chat_screen.dart:

```

import 'package:flutter/material.dart';
import 'package:flash_chat/constants.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

final _firestore = FirebaseFirestore.instance;
User loggedInUser;

```

```
class ChatScreen extends StatefulWidget {  
  static const String id = 'chat_screen';  
  @override  
  _ChatScreenState createState() => _ChatScreenState();  
}
```

```
class _ChatScreenState extends State<ChatScreen> {  
  final messageTextController = TextEditingController();  
  final _auth = FirebaseAuth.instance;
```

```
  String messageText;
```

```
  @override
```

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

```
    getCurrentUser();  
  }
```

```
  void getCurrentUser() {  
    try {  
      final user = _auth.currentUser;  
      if (user != null) {  
        loggedInUser = user;  
        print(loggedInUser.email);  
      }  
    } catch (e) {  
      print(e);  
    }  
  }
```

```

// void getMessages() async {
//   final messages = await _firestore.collection('messages').get();
//   for (var message in messages.docs) {
//     print(message.data());
//   }
// }

// void messagesStream() async {
//   await for (var snapshot in _firestore.collection('messages').snapshots()) {
//     for (var message in snapshot.docs) {
//       print(message.data());
//     }
//   }
// }

```

@override

```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: null,
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.close),
          onPressed: () {
            _auth.signOut();
            Navigator.pop(context);
            // messagesStream();
          },
        ),
      ],
      title: Text(' ⚡ Chat'),
      backgroundColor: Colors.lightBlueAccent,
    ),
  );
}

```

```

),
body: SafeArea(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: <Widget>[
      MessagesStream(),
      Container(
        decoration: kMessageContainerDecoration,
        child: Row(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: <Widget>[
            Expanded(
              child: TextField(
                controller: messageTextController,
                onChanged: (value) {
                  messageText = value;
                },
                decoration: kMessageTextFieldDecoration,
              ),
            ),
          ],
          FlatButton(
            onPressed: () {
              messageTextController.clear();
              _firestore.collection('messages').add({
                'text': messageText,
                'sender': loggedInUser.email,
                'timestamp': FieldValue.serverTimestamp(),
              });
            },
            child: Text(

```



```

        'Send',
        style: kSendButtonTextStyle,
    ),
),
],
),
),
],
),
),
);
}
}

```

```

class MessagesStream extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream:
        _firebase.collection('messages').orderBy('timestamp').snapshots(),
      builder: (context, snapshot) {
        if (snapshot.hasData) {
          final messages = snapshot.data.docs.reversed;
          List<MessageBubble> messageBubbles = [];
          for (var message in messages) {
            final messageText = message['text'];
            final messageSender = message['sender'];

            final currentUser = loggedInUser.email;

            final messageBubble = MessageBubble(

```

```

        sender: messageSender,
        text: messageText,
        isMe: currentUser == messageSender,
    );

    messageBubbles.add(messageBubble);
}
return Expanded(
  child: ListView(
    reverse: true,
    padding: EdgeInsets.symmetric(horizontal: 10.0, vertical: 20.0),
    children: messageBubbles,
  ),
);
} else {
  return Center(
    child: CircularProgressIndicator(
      backgroundColor: Colors.lightBlueAccent,
    ),
  );
}
},
);
}
}
}

```

```

class MessageBubble extends StatelessWidget {
  MessageBubble({this.sender, this.text, this.isMe});

  final String sender;
  final String text;

```

```
final bool isMe;
```

```
@override
```

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

```
  return Padding(
```

```
    padding: EdgeInsets.all(10.0),
```

```
    child: Column(
```

```
      crossAxisAlignment:
```

```
        isMe ? CrossAxisAlignment.end : CrossAxisAlignment.start,
```

```
      children: [
```

```
        Text(
```

```
          sender,
```

```
          style: TextStyle(
```

```
            fontSize: 12.0,
```

```
            color: Colors.black54,
```

```
        ),
```

```
      ),
```

```
      Material(
```

```
        elevation: 5.0,
```

```
        borderRadius: isMe
```

```
          ? BorderRadius.only(
```

```
            topLeft: Radius.circular(30.0),
```

```
            bottomLeft: Radius.circular(30.0),
```

```
            bottomRight: Radius.circular(30.0))
```

```
          : BorderRadius.only(
```

```
            topRight: Radius.circular(30.0),
```

```
            bottomLeft: Radius.circular(30.0),
```

```
            bottomRight: Radius.circular(30.0)),
```

```
        color: isMe ? Colors.lightBlueAccent : Colors.white,
```

```
        child: Padding(
```

```
          padding: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),
```

```

        child: Text(
          text,
          style: TextStyle(
            fontSize: 15.0,
            color: isMe ? Colors.white : Colors.black54),
        ),
      )),
    ],
  ),
);
}
}

```

rounded_button.dart:

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

```

class RoundedButton extends StatelessWidget {
  final Color color;
  final Function onPressed;
  final String title;
  RoundedButton({@required this.color, @required this.title, this.onPressed});

```

```
@override
```

```

Widget build(BuildContext context) {
  return Padding(
    padding: EdgeInsets.symmetric(vertical: 16.0),
    child: Material(
      color: color,
      borderRadius: BorderRadius.circular(30.0),
      elevation: 5.0,
      child: MaterialButton(

```

```
        onPressed: onPressed,  
        minWidth: 200.0,  
        height: 42.0,  
        child: Text(  
          title,  
          style: TextStyle(color: Colors.white),  
        ),  
      ),  
    ),  
  );  
}  
}
```

constants.dart:

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

```
const kSendButtonTextStyle = TextStyle(  
  color: Colors.lightBlueAccent,  
  fontWeight: FontWeight.bold,  
  fontSize: 18.0,  
);
```

```
const kMessageTextFieldDecoration = InputDecoration(  
  contentPadding: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),  
  hintText: 'Type your message here...',  
  border: InputBorder.none,  
);
```

```
const kMessageContainerDecoration = BoxDecoration(  
  border: Border(  
    top: BorderSide(color: Colors.lightBlueAccent, width: 2.0),
```

```

    ),
  );

const kTextFieldDecoration = InputDecoration(
  hintText: 'Enter a value',
  contentPadding: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),
  border: OutlineInputBorder(
    borderRadius: BorderRadius.all(Radius.circular(32.0)),
  ),
  enabledBorder: OutlineInputBorder(
    borderSide: BorderSide(color: Colors.blueAccent, width: 1.0),
    borderRadius: BorderRadius.all(Radius.circular(32.0)),
  ),
  focusedBorder: OutlineInputBorder(
    borderSide: BorderSide(color: Colors.blueAccent, width: 2.0),
    borderRadius: BorderRadius.all(Radius.circular(32.0)),
  ),
);

```

pubspec.yaml:

```

name: flash_chat
description: A new Flutter application.

```

```

version: 1.0.0+1

```

```

environment:
  sdk: ">=2.1.0<3.0.0"

```

dependencies:

```

flutter:
  sdk: flutter

```

```
cupertino_icons: ^0.1.2
animated_text_kit: ^4.2.1
firebase_core: ^1.14.1
firebase_auth: ^3.3.14
cloud_firestore: ^3.1.12
modal_progress_hud: ^0.1.3
```

dev_dependencies:

```
flutter_test:
  sdk: flutter
```

flutter:

```
uses-material-design: true
```

assets:

```
- images/
```

main/AndroidManifest.xml:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="co.appbrewery.flash_chat">
```

```
<!-- io.flutter.app.FlutterApplication is an android.app.Application that
     calls FlutterMain.startInitialization(this); in its onCreate method.
```

```
     In most cases you can leave this as-is, but you if you want to provide
     additional functionality it is fine to subclass or reimplement
```

```
     FlutterApplication and put your custom class here. -->
```

```
<application
  android:name="${applicationName}"
  android:label="flash_chat"
```

```

        android:icon="@mipmap/ic_launcher">
        <activity
            android:name="io.flutter.embedding.android.FlutterActivity"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"

            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale|layoutDirec
            tion|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <!-- This keeps the window background of the activity showing
                until Flutter renders its first frame. It can be removed if
                there is no splash screen (such as the default splash screen
                defined in @style/LaunchTheme). -->

            <meta-data
                android:name="io.flutter.app.android.SplashScreenUntilFirstFrame"
                android:value="true" />

            <meta-data
                android:name="flutterEmbedding"
                android:value="2" />

            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>

```

images/logo.png:



OUTPUT

- Firstly, we need to install flutter and inspect whether you have the prerequisites using **flutter doctor** command.

```
E:\Flutterfiles\flash-chat-flutter>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[V] Flutter (Channel stable, 2.10.5, on Microsoft Windows [Version 10.0.22000.856], locale en-IN)
[V] Android toolchain - develop for Android devices (Android SDK version 32.1.0-rc1)
[V] Chrome - develop for the web
[V] Visual Studio - develop for Windows (Visual Studio Community 2022 17.3.4)
[V] Android Studio (version 2021.1)
[V] VS Code (version 1.73.1)
[V] Connected device (3 available)
[V] HTTP Host Availability

• No issues found!
```

This shows that the requirements are fulfilled and the app is ready to run. If there are any issues, it is displayed also with the fix.



The app is installed when flutter run command is executed with a logo.

- Welcome screen is rendered with a typing animation when the application is started.
- New users need to Register with their email and new password in the registration page.

Flash Chat_

Log In

Register



karthik03102001ece@gmail.com

.....

Register



karthik03102001ece@gmail.com

.....

Log In

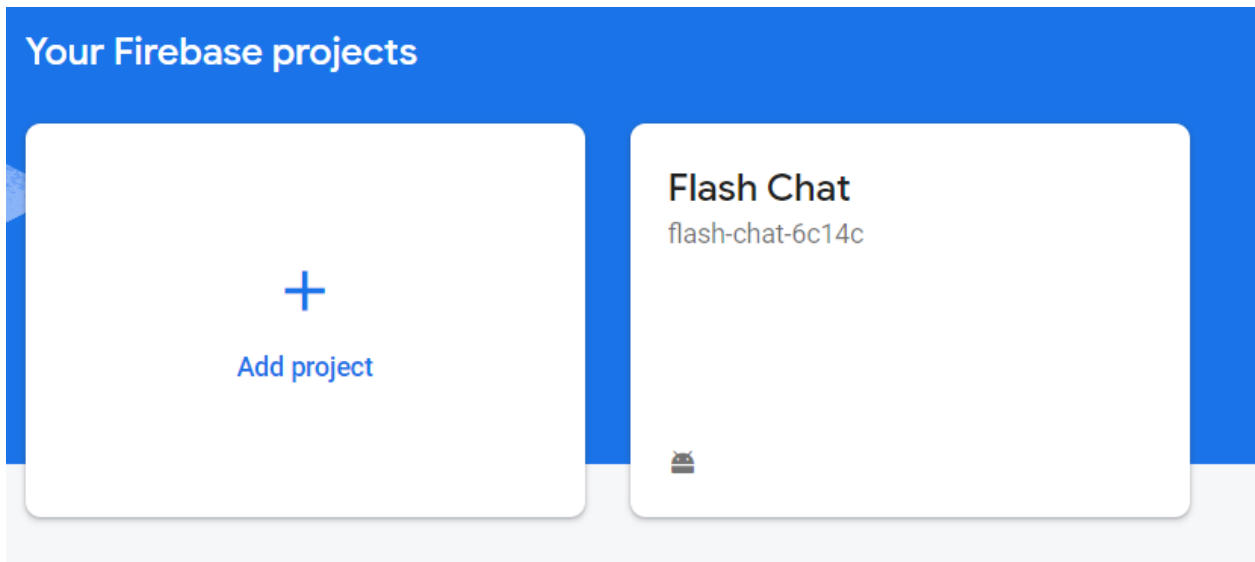
<  Chat



Type your message here...

Send

- After the user registered, the user can login anytime and exchange messages in the chat page. All the firebase build services such as database, authentication, and extensions can be used by creating a firebase project.



- After adding multiple users, the firebase authentication adds the users with a User UID and signed-in date as shown below. The app administrator can add, remove or disable users.

Flash Chat ▾ Go to

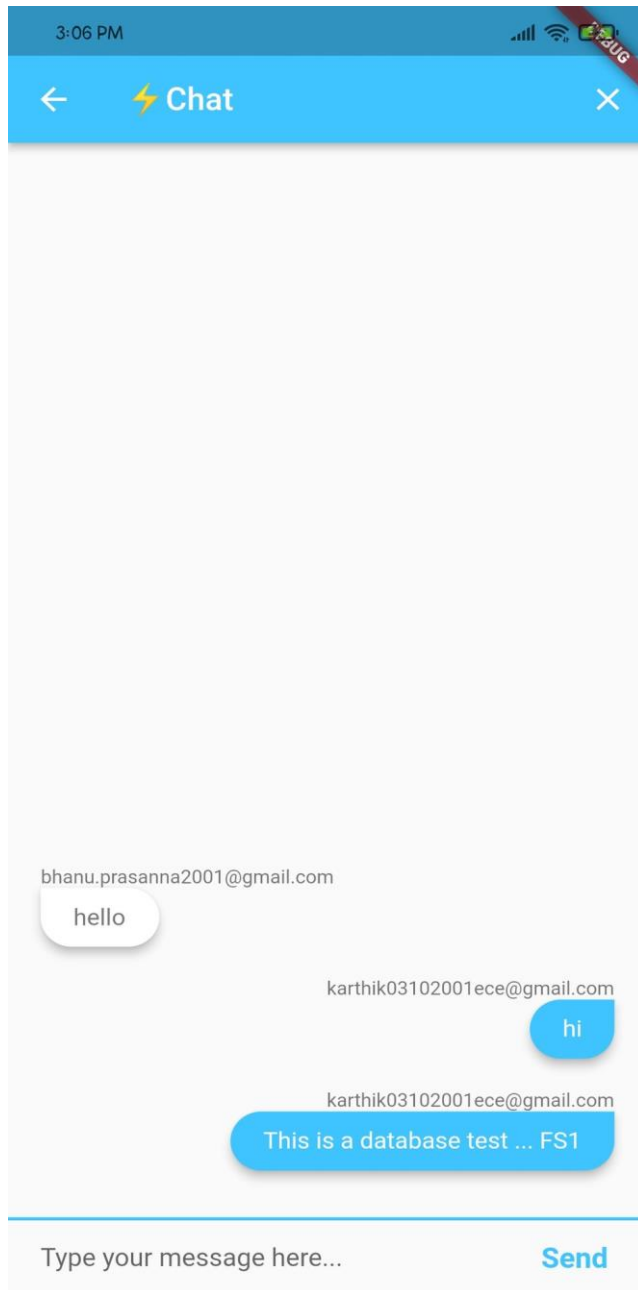
Authentication

[Users](#) [Sign-in method](#) [Templates](#) [Usage](#) [Settings](#)

[Add user](#)
↺
⋮

Identifier	Providers	Created ↓	Signed in	User UID
bhanu.prasanna2001@gm...	✉	4 Dec 2022	4 Dec 2022	0Ux11TVFnJfbvwMX5HaqhSsHEg...
yaswanthakula02@gmail.c...	✉	4 Dec 2022	4 Dec 2022	ZB2QIHdJ1bMny3KEmEwZo8Lyv8...
karthik03102001ece@gma...	✉	4 Dec 2022	4 Dec 2022	nKEOnedPdnTY5VeF0qDaaK94cN...
devakarthik03102001ece...	✉	7 Nov 2022	8 Nov 2022	NKdZZUk9czP9BhyKrWEBKA1S75...

Rows per page
50 ▾
1 – 4 of 4
<
>



Multiple users can exchange messages and the messages are displayed with the user email specified above so that others can know the sender. The messages data is stored in a **messages** collection inside the Firestore database as show in the below figure.

Flash Chat ▾ Cloud Firestore		
🏠 > messages > HFsrCH32BD2s...		
📁 flash-chat-6c14c	📁 messages	📄 HFsrCH32BD2snry1v9Wu
+ Start collection	+ Add document	+ Start collection
messages >	HFsrCH32BD2snry1v9Wu >	+ Add field
	jh0BhqxrdgW414M8rUo1 z8qQWmYpm58ip005KSsb	sender: "karthik03102001ece@gmail.com" text: "This is a database test ... FS1" timestamp: 4 December 2022 at 15:06:38 UTC+5:30