

# **INNOVA CLOUD APPLICATION**

## **CS 5525 - Cloud Computing Project**

### **Team Members Pictures:**



### **Team Members:**

Bharath Kumar Pulindram - 16337663

Abhinay Reddy Polimera - 16334737

Gayathri Sarapu - 16344170

**Application Link:** <https://cloud-native-chat.web.app>

**Github code link:** <https://github.com/BharathPulindram/cc-cloud-native>

### **App video demo overview:**

[https://umsystem.zoom.us/rec/share/PfDH8S9lkgcNvFXcl-5V7nSeb8kdznr4wLCbCx6VmhdJCNzmz7mXMPjw-fGwhrzJ.03fj63YS7pq8ur\\_p?startTime=1683945911000](https://umsystem.zoom.us/rec/share/PfDH8S9lkgcNvFXcl-5V7nSeb8kdznr4wLCbCx6VmhdJCNzmz7mXMPjw-fGwhrzJ.03fj63YS7pq8ur_p?startTime=1683945911000)

## **Motivation / Purpose:**

This is a cross-platform cloud-native dynamic web application, deployed on Google Firebase and accessible via Play Store. Technologies include React JS, Firebase authentication, Firebase hosting, Firebase storage.

This application can be used by teams to collaborate and share information. It features the use of a chat engine, which synchronizes messages from multiple users in real time. Users can message using text, emojis, GIFs, etc. It acts as a shared workspace where discussions and digital content can be shared. The application also sends broadcast notifications to users, when the content needs to be shared across large groups of participants. The real time database is used as a web socket to transfer real time data effectively among users.

The technology stack includes – Firebase authentication, firebase storage, firebase hosting, React JS, Next JS. This application isolates the work among individuals by creating a room for their content and information. Application is developed in React JS, Next JS, firebase authentication and storage are built from Google's firebase, and the complete application would be deployed using Firebase hosting. Data visualization is also achieved using firebase google analytics and fire store which is a real time database

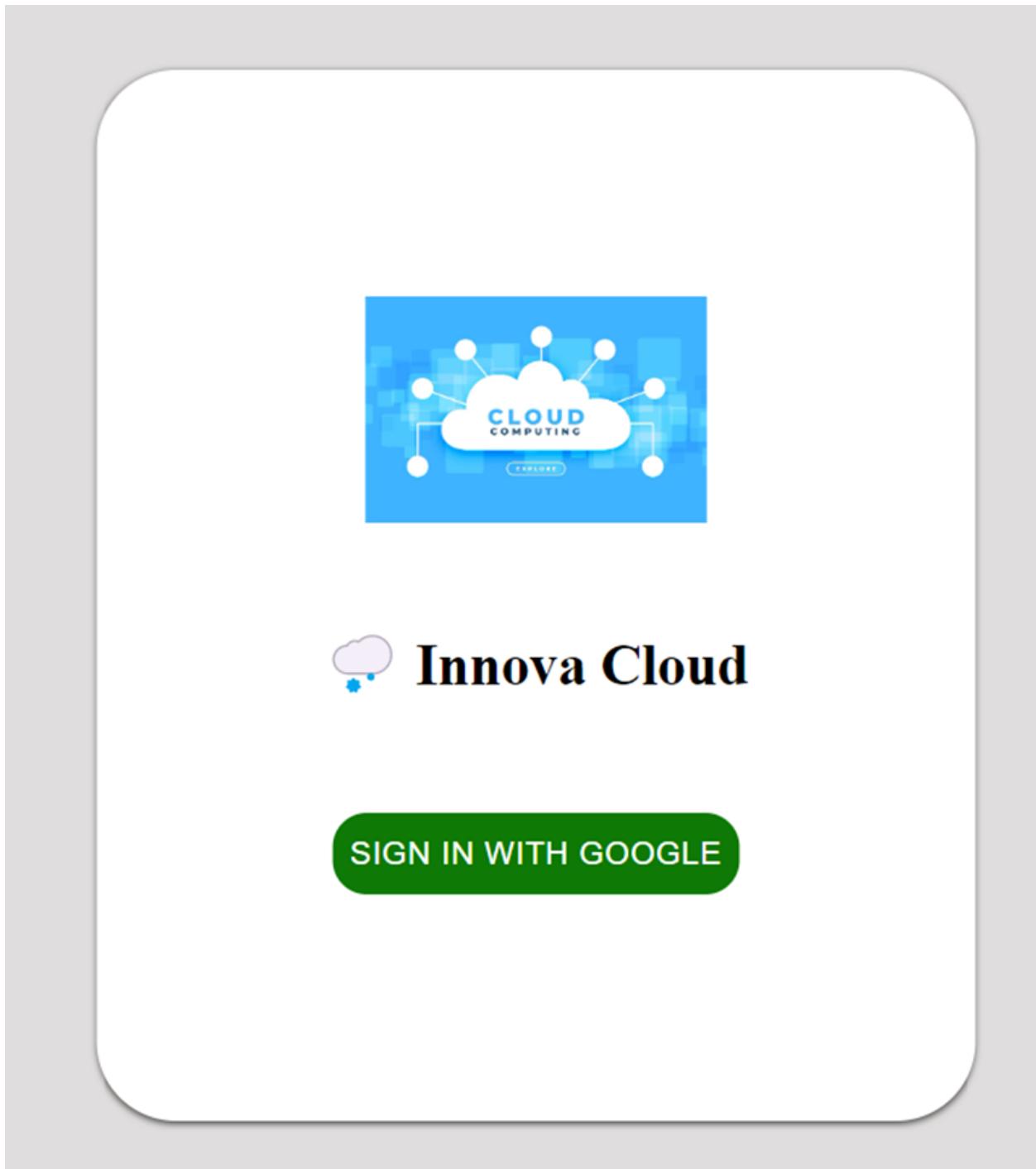
## **Technologies Used:**

### **Cloud Technologies:** Google Firebase

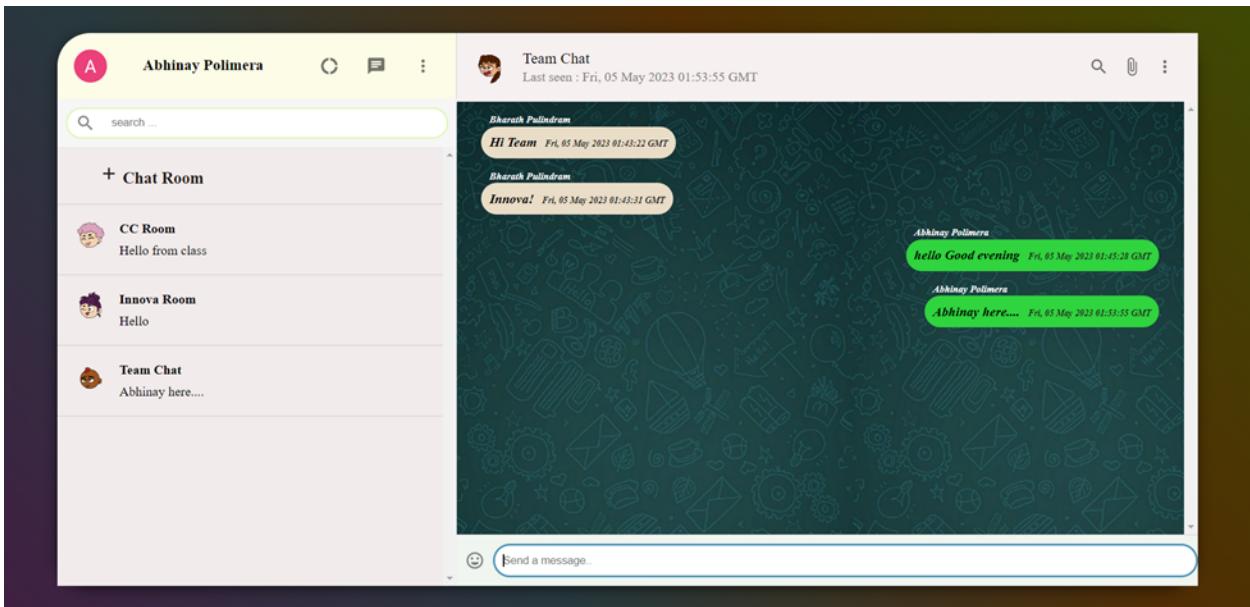
Firebase is used as a cloud service for real time cloud database, hosting, google authentication and Google analytics for data visualization.

**Technologies:** React JS is used for frontend development in which the entire UI application part is developed. The NextJS framework of React is used for server side rendering and data operations.

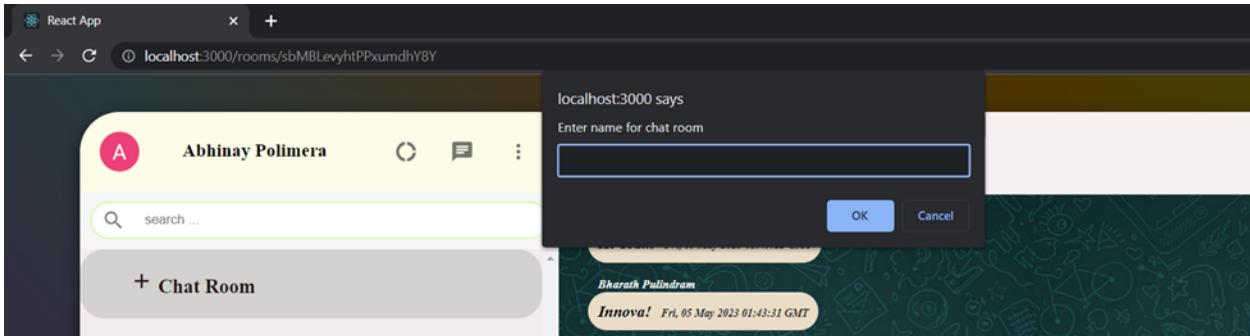
**Step1:** Login Page with Google Authentication:



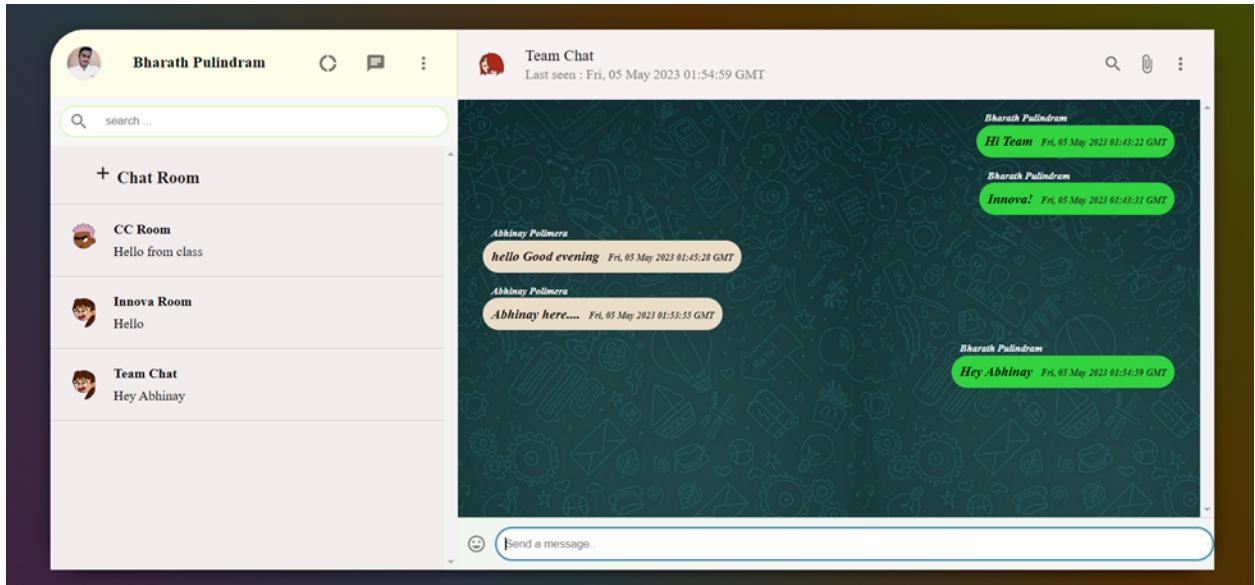
## Step2: Homepage with chat rooms:



## Step3: To create a new chat room:



## Step4: Multiple users accessing and messaging at the same time:



## Step5: Firebase Authentication

The screenshot shows the Firebase Authentication screen in the Firebase console. The left sidebar includes Project Overview, Firestore Database, and Authentication. The Authentication section has tabs for Users, Sign-in method, Templates, Usage, Settings, and Extensions. The Users tab is selected, displaying a table of users:

Identifier	Providers	Created	Signed In	User UID
polimeraabhinay@gmail.co...		May 4, 2023	May 4, 2023	DiQ3LGYiFwMr2JXXe00mJNG8tEk1
bharath.pulindram@gmail...		Apr 28, 2023	May 4, 2023	Ik303qWOSGMepWQBY6fb8S4uL...
pbkbharath1707@gmail.co...		Apr 28, 2023	May 4, 2023	UgxICegF7VdeShVbF34peINJXsE2

At the bottom, there are pagination controls: "Rows per page: 50", "1 = 3 of 3", and navigation arrows.

## Step6: Firebase Realtime Database

The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation bar includes 'Project Overview', 'Firestore Database', 'Authentication', 'Analytics', 'Engage', and 'All products'. A sidebar message says 'Customize your nav!' and 'Spark No-cost \$0/month Upgrade'. The main area is titled 'Cloud Firestore' with tabs for 'Data', 'Rules', 'Indexes', 'Usage', and 'Extensions'. A banner at the top right says 'Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing' with a 'Configure App Check' button. Below, a 'Panel view' tab is selected, showing a hierarchical tree structure: 'cloud-native-chat' > 'rooms' > 'I4T28td3FopuBJwoYdZR'. This document contains fields 'oBh6SnZsYp4b3Gn04pcP' and 'sbMBLevyhtPPxumdhY8Y'. To the right, a 'Query builder' panel shows a collection named 'messages' with a single document 'I4T28td3FopuBJwoYdZR' containing the field 'name: "CC Room"'. A 'More in Google Cloud' link is at the top right of the builder.

## Step7: Chat messages schema and real time data

This screenshot shows the 'messages' collection within the 'rooms' document of the 'sbMBLevyhtPPxumdhY8Y' document. The 'messages' collection has a single document 'sbMBLevyhtPPxumdhY8Y' with the field 'name: "Team Chat"'. The 'More in Google Cloud' link is visible at the top right.

This screenshot shows the 'messages' collection within the 'rooms' document of the 'sbMBLevyhtPPxumdhY8Y' document. The 'messages' collection has multiple documents: '6FT1Ix3ynmtoP7yzp8r', '7tdFY9DkasNLDgt1zot1', 'V0o9ySJ9eITR9mJmttyrw', 'ZzfKhNo930yNPkly1MgU', and 'gwV2xgMuY0u4jc7fa0JU'. The 'More in Google Cloud' link is visible at the top right.

## Step8: Added Story Posting functionality

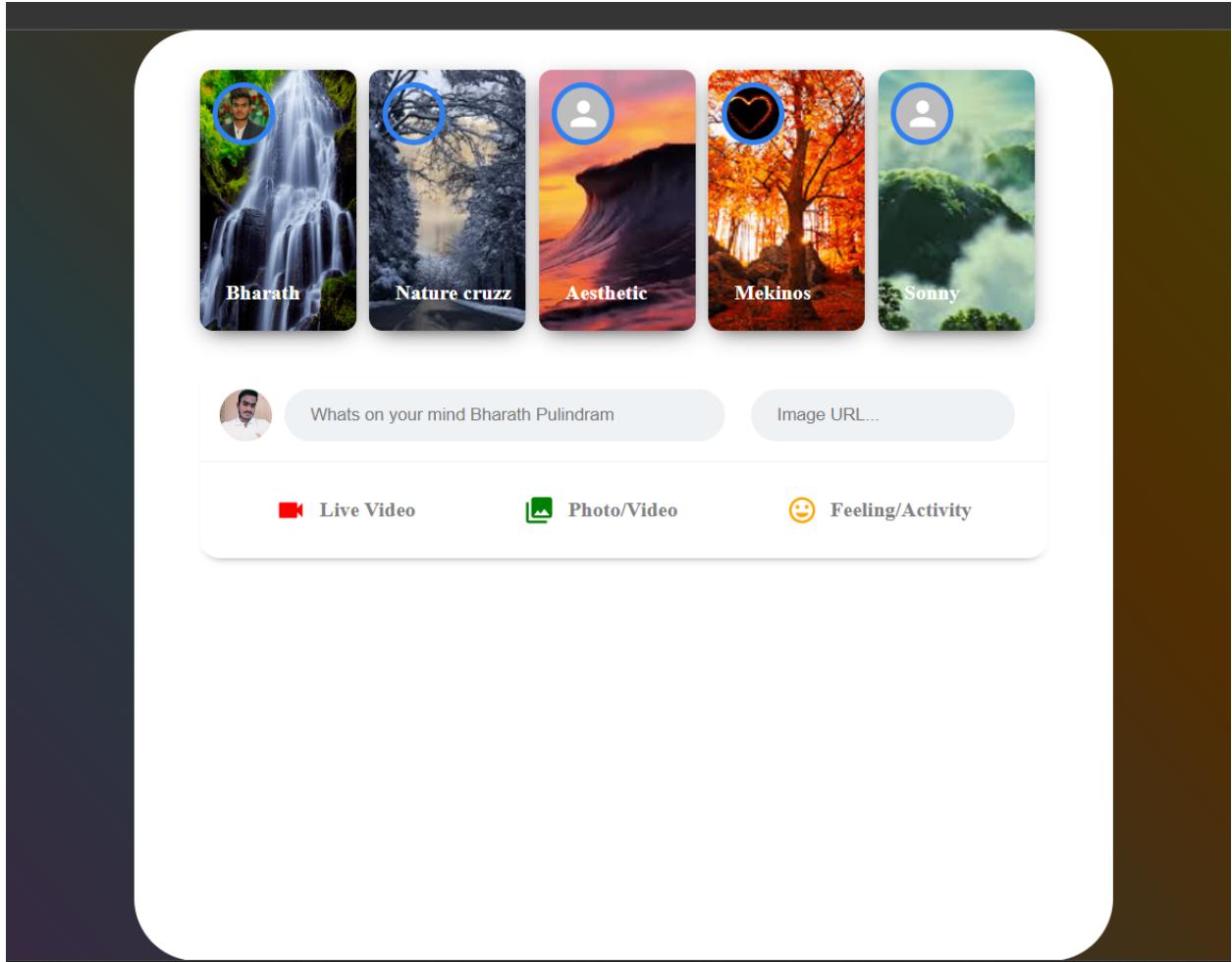
## Step9: Homepage after logged in



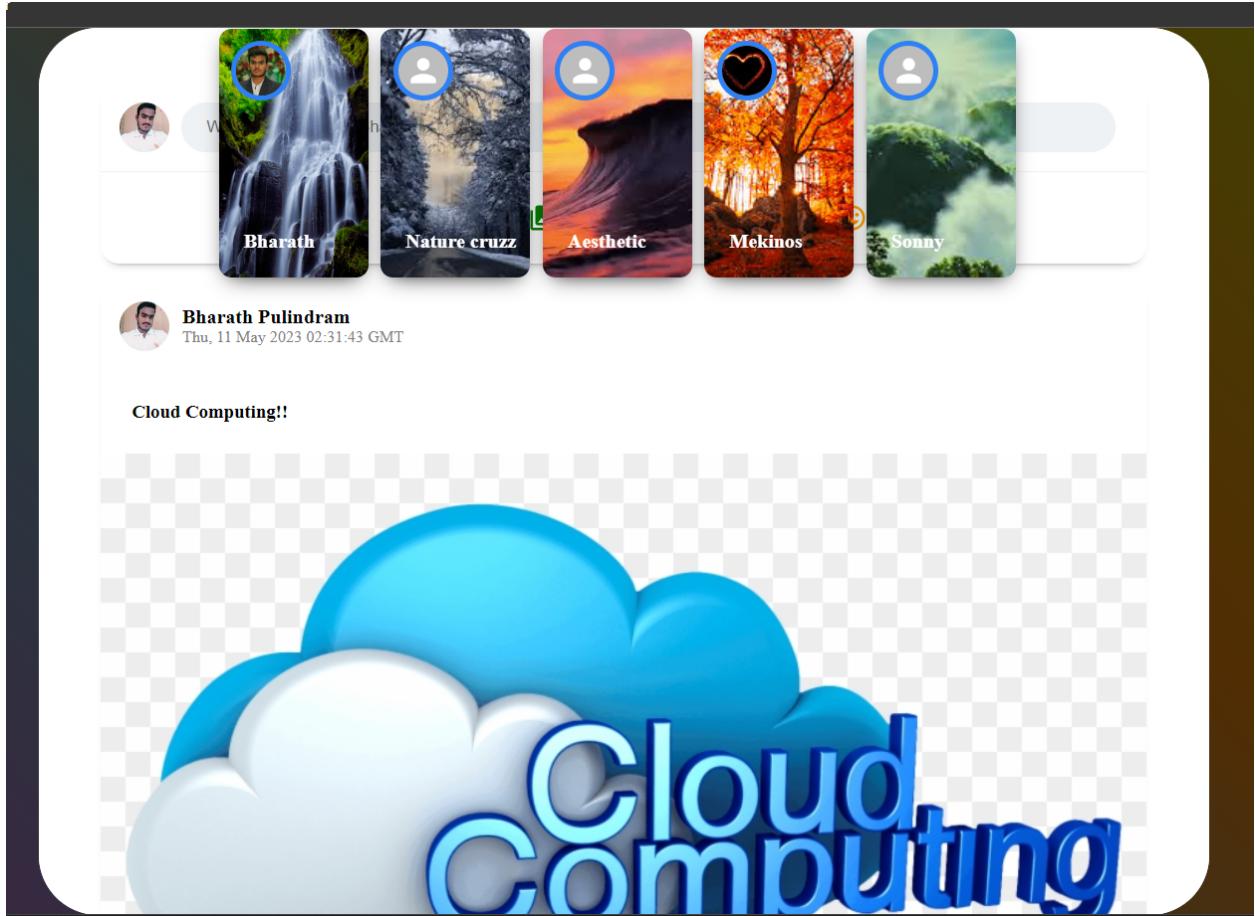
On clicking Chat Engine we navigate to chat rooms

The image shows a screenshot of a mobile application interface. On the left, there is a sidebar with a user profile picture and the name "Bharath Pulindram". Below the profile are four chat room entries: "CC Room" (with message "Hi room"), "Innova Room" (with message "good to see you"), "Team Chat" (with message "Hey Abhinay"), and a "+ Chat Room" button. On the right, there is a main chat screen for the "CC Room". The header shows the room name "CC Room" and the last seen time "Thu, 11 May 2023 02:20:58 GMT". The body of the screen shows a conversation between Bharath and Abhinay Polimera. Bharath's messages are in green bubbles, and Abhinay's message is in a grey bubble. The messages are: "Hello Guys!" (Fri, 28 Apr 2023 05:00:00 GMT), "Hello from cloud app" (Sat, 29 Apr 2023 02:40:24 GMT), "Good" (Sat, 29 Apr 2023 04:25:01 GMT), "Hello from class" (Thu, 04 May 2023 19:24:49 GMT), and "Hello" (Thu, 11 May 2023 02:17:32 GMT). At the bottom of the screen is a message input field with a placeholder "Send a message..".

On selecting Post Stories you would redirect to Post stories page



**After posting message and image:**



## Firebase db collection - post stories

posts > qWCCWxR97td...

[More in Google Cloud](#)

cloud-native-chat	posts	qWCCWxR97td1U0rAxixC
+ Start collection	+ Add document	+ Start collection
posts >	qWCCWxR97td1U0rAxixC >	+ Add field
rooms		image: "https://www.clipartmax.com/png/middle/227-2278606_cloud-computing-clipart-cluster-computing-cloud-computing-logo-png.png" message: "Cloud Computing!" profilePic: "https://lh3.googleusercontent.com/a/AGNmyxa-TQXbtXkUi5ZUKkxG869ZwGgb9eazbM2I57s6tQ=s96-c" timestamp: May 10, 2023 at 9:31:43PM UTC-5 username: "Bharath Pulindram"

Sample code image:

The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor pane on the right.

**EXPLORER**

- OPEN EDITORS: Sidebarjsx 1
- CLOUD-CHAT
  - node\_modules
  - public
  - src
    - Chat
    - Login
      - Login.css
      - Login.jsx
    - Sidebar
      - Sidebar.css
      - Sidebar.jsx 1
    - SidebarChat
      - firebase.js
      - reducer.js
      - StateProvider.js
    - Facebook
      - App.css
      - App.js
      - index.css

**Sidebarjsx 1**

```
You, last week | 1 author (You)
1 import React, { useEffect, useState } from "react";
2 import Avatar from "@mui/material/Avatar"; You, 2 weeks ago • sidebar
3 import "./Sidebar.css";
4 import DonutLargeIcon from "@mui/icons-material/DonutLarge";
5 import ChatIcon from "@mui/icons-material/Chat";
6 import MoreVertIcon from "@mui/icons-material/MoreVert";
7 import { IconButton } from "@mui/material";
8 import SearchOutlinedIcon from "@mui/icons-material/SearchOutlined";
9 import SidebarChat from "../SidebarChat/SidebarChat";
10 import db from "../../firebase";
11 import { useStateValue } from "../../StateProvider";
12
13 function Sidebar() {
14   const [rooms, setRooms] = useState([]);
15   const [{ user }, dispatch] = useStateValue();
16
17   useEffect(() => {
18     const unSubscribe = db.collection("rooms").onSnapshot((snapshot) =>
19       setRooms(
20         snapshot.docs.map((doc) => ({
21           id: doc.id,
22           data: doc.data(),
23         })
24       )
25     )
26   })
27 }
```

## RESULT:

The Application is deployed on firebase Hosting.

<https://cloud-native-chat.web.app>

We are able to chat with multiple users in real time using chat rooms and also post stories in the app. We can also track and access user data in the firestore database.