A Project Report on

Cloud Chat - Messaging in the Cloud

Submitted in partial fulfillment of the requirements for the award
of the degree of

Bachelor of Engineering

in

Computer Engineering

by

Harvinder Singh(18102018)
Royston Rodrigues(18102003)
Parth Vora(18102037)
Lavleen Jain(18102040)

Under the Guidance of

Prof. Deepak Khachane

Department of Computer Engineering
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2021-2022

Approval Sheet

This Project Report entitled *"Cloud Chat - Messaging in the cloud"* Submitted by *"Harvinder Singh"(18102018),"Royston Rodrigues"(18102003),"Parth Vora"(18102037),"Lavleen Jain"(18102040)*is approved for the partial fulfillment of the requirement for the award of the degree of *Bachelor of Engineering* in *Computer Engineering* from *University of Mumbai*.

(Prof. Deepak Khachane)
Guide

Prof. Sachin Malave
Head Department of Computer Engineering

Place:A.P.Shah Institute of Technology, Thane
Date: 21/4/2022

CERTIFICATE

This is to certify that the project entitled **"Cloud Chat - Messaging in the Cloud"** submitted by **"Harvinder Singh" (18102018), "Royston Rodrigues" (18102003), "Parth Vora" (18102037), "Lavleen Jain" (18102040)** for the partial fulfillment of the requirement for award of a degree **Bachelor of Engineering** in **Computer Engineering**, to the University of Mumbai, is a bonafide work carried out during academic year 2021-2022.

(Prof. Deepak Khachane)
Guide

Prof. Sachin Malave   Dr. Uttam D.Kolekar
Head Department of Computer Engineering   Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date: 21/4/2022

Acknowledgement

We have great pleasure in presenting the report on **Cloud Chat - Messaging in the cloud.** We take this opportunity to express our sincere thanks towards our guide **Prof. Deepak Khachane** from the Department of Computer Engineering, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Sachin Malave** Head of Department,Computer Engineering, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Harvinder Singh
18102018

Royston Rodrigues
18102003

Parth Vora
18102037

Lavleen Jain
18102040

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

_____

Harvinder Singh (18102018)
Royston Rodrigues (18102003)
Parth Vora (18102037)
Lavleen Jain (18102040)

Date: 21/4/2022

Abstract

Cloud computing has transformed the computer era by removing hardware limitations and giving access to near infinite and scalable resources , which help businesses expand across all departments wihout having to bear the costs of physical expansion in terms of hardware and logistics. Cloud service providers today offer a variety of services ranging from compute engines to serverless data warehouses. Clearly the possibilities and the opporturnities that cloud computing offers is endless. Legacy systems can benefit greatly by moving to the cloud which is secure and easily scalable as the demand grows but most importantly , any cloud service , right down to the hardware configuration can be easily configured according to the size of your business , giving you full control of what you use and how you use and also paying for what you use. Cloud Computing has emerged as a superior form of computing which allows unparalled flexibility to its consumers and allowing them to quickly switch between services if the need arises and not having to worry about security issues like data corruption or loss of sensitive data.

Contents

List of Figures

List of Abbreviations

UI: User Interface
JSON: Javascript object notation
ACID :   atomicity, consistency, isolation, and durability.

Chapter 1

Introduction

'Cloud Chat' is a chat application made using the Node.js and Firebase for the backend and Svelte + TailwindCSS for the frontend. Cloud Chat allows you to chat with other users by creating chat rooms. The concept of private rooms have been widely used in chat applications and IRC chats. Private chatrooms are secure and Cloud Chat generates room codes that you can then share with your friends or family so that they can enter the personal chatroom and start chatting with you. This is all possible through the powerful firebase backend which allows login through a google account and all messages are stored in Firestore - a powerful database offered by firebase , and a part of a wide range of services given by them.

Chapter 2

Literature Review

Cloud storage is a cloud computing model that stores data on the Internet through a cloud computing provider who manages and operates data storage as a service. It's delivered on demand with just-in-time capacity and costs, and eliminates buying and managing your own data storage infrastructure. This gives you agility, global scale and durability, with "anytime, anywhere" data access.
Recently with the Rise of NOSQL models , Cloud services providers have started providing cloud database services.
NoSQL database technology stores information in JSON documents instead of columns and rows used by relational databases. To be clear, NoSQL stands for "not only SQL" rather than "no SQL" at all. This means a NoSQL JSON database can store and retrieve data using literally "no SQL." Or you can combine the flexibility of JSON with the power of SQL for the best of both worlds.
Consequently, NoSQL databases are built to be flexible, scalable, and capable of rapidly responding to the data management demands of modern businesses.

The following defines the four most-popular types of NoSQL database.

Document databases are primarily built for storing information as documents, including, but not limited to, JSON documents. These systems can also be used for storing XML documents.

Key-value stores group associated data in collections with records that are identified with unique keys for easy retrieval.Key-value stores have just enough structure to mirror the value of relational databases while still preserving the benefits of NoSQL.

Wide-column databases use the tabular format of relational databases yet allow a wide variance in how data is named and formatted in each row, even in the same table. Like key-value stores, wide-column databases have some basic structure while also preserving a lot of flexibility

Graph databases use graph structures to define the relationships between stored data points. Graph databases are useful for identifying patterns in unstructured and semi-structured information.

Firestore, offered by google as a part of the firebase cloud services is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps - at global scale.

Firestore is serverless and has a powerful query engine that supports complicated ACID transcations and most importantly the feature set that firestore offers easily supersedes the requirements for this project.

The Firestore database architecture makes it easy to quickly prototype and scale a chat application that can be eventually expanded upon and made into a more architecturally sound application complete with room-based chats with sharable codes and Single Sign-On.

Chapter 3

Project Design

3.1 Proposed System
Core Web Application powered by Firebase in the backend saving data to a firestore database.The application uses firestore for all core chat related functionalities including saving and retreieving messages in a room and tagging them with the appropriate user.

3.2 Design (Core Architecture)

FrontEnd
(1) The application relies on the Svelte frontend framework which is a frontend compiler.The app is built in a modularised widget based approach and with the help of svelte it is compiled down to a relatively simple public folder to be served on the web.
(2) Svelte widgets have the .svlete extension and support standard html syntax with the addition of a templating engine to extend functionality in the frontend code.
(3) TailwindCSS , an utility CSS framework has been used to create the styling for the app by directly calling the relevant classes in the HTML code.

Backend
(1) The application relies on the Firebase firestore database which is used to store all the chats and chat room codes from the application.
(2) Authentication is provided by firebase auth and users can sign in to the application using a google account.
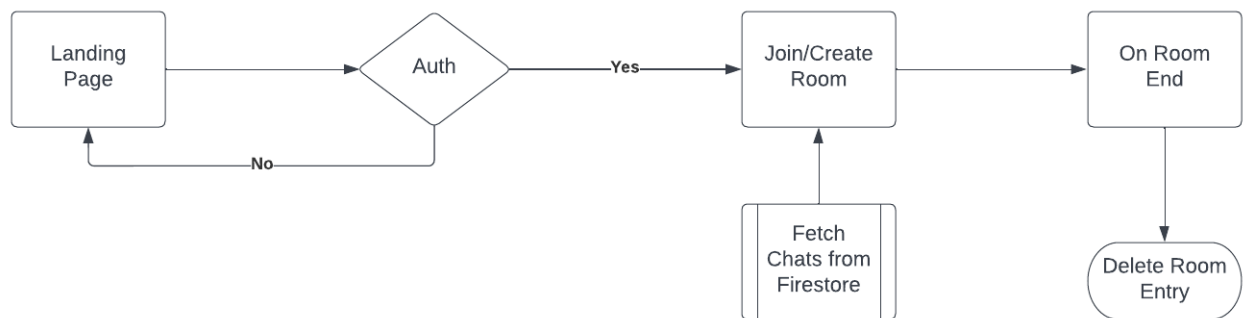
3.3 Class Diagram

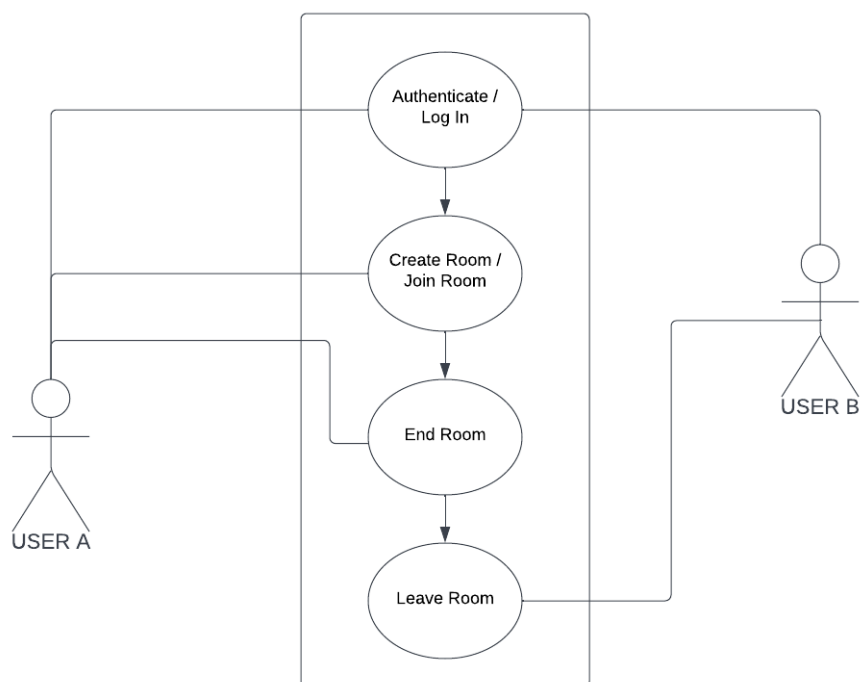Figure 3.1: Class diagram of the application

3.4 Use Case Diagram

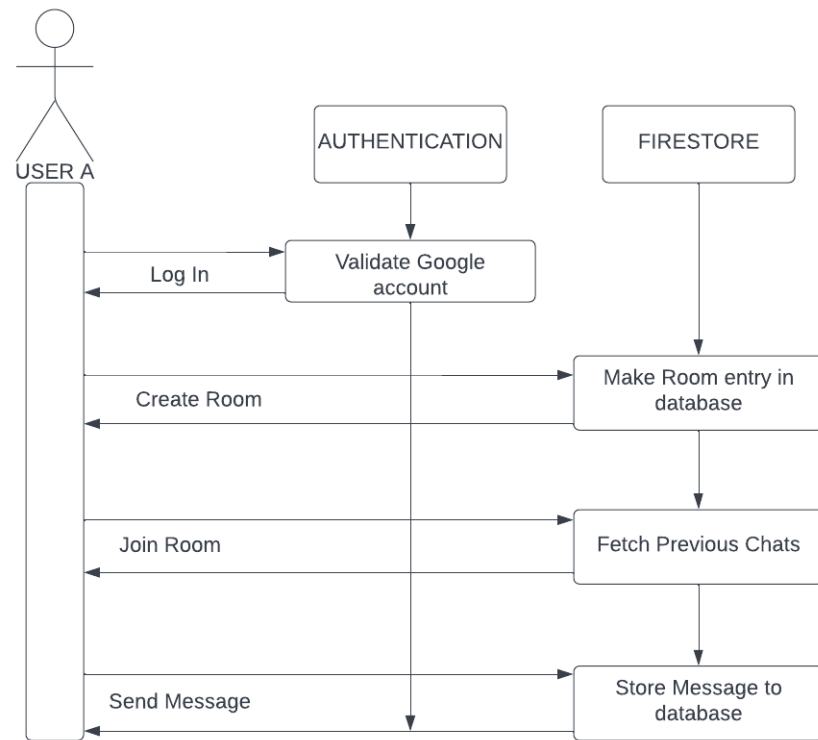Figure 3.2: Use Case diagram of the application

3.5 Sequence Diagram

Figure 3.3: Sequence diagram of the application

Chapter 4

Project Implementation

The screenshot below is a representation of how the final app would look like. The design and themes are subject to change.
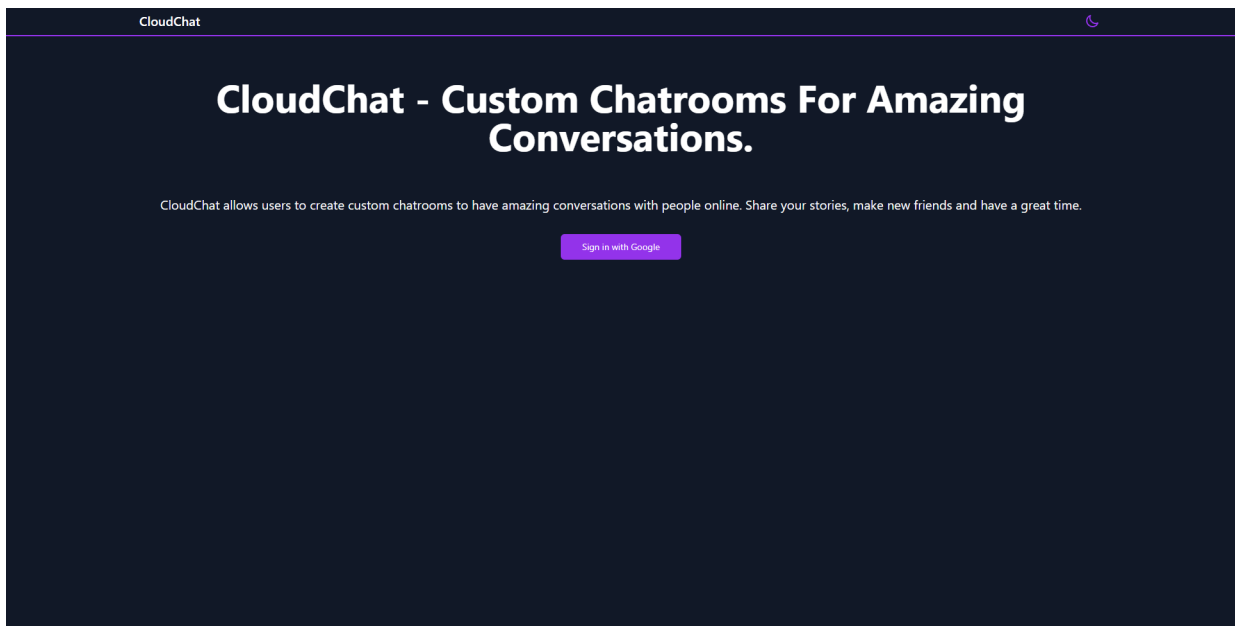

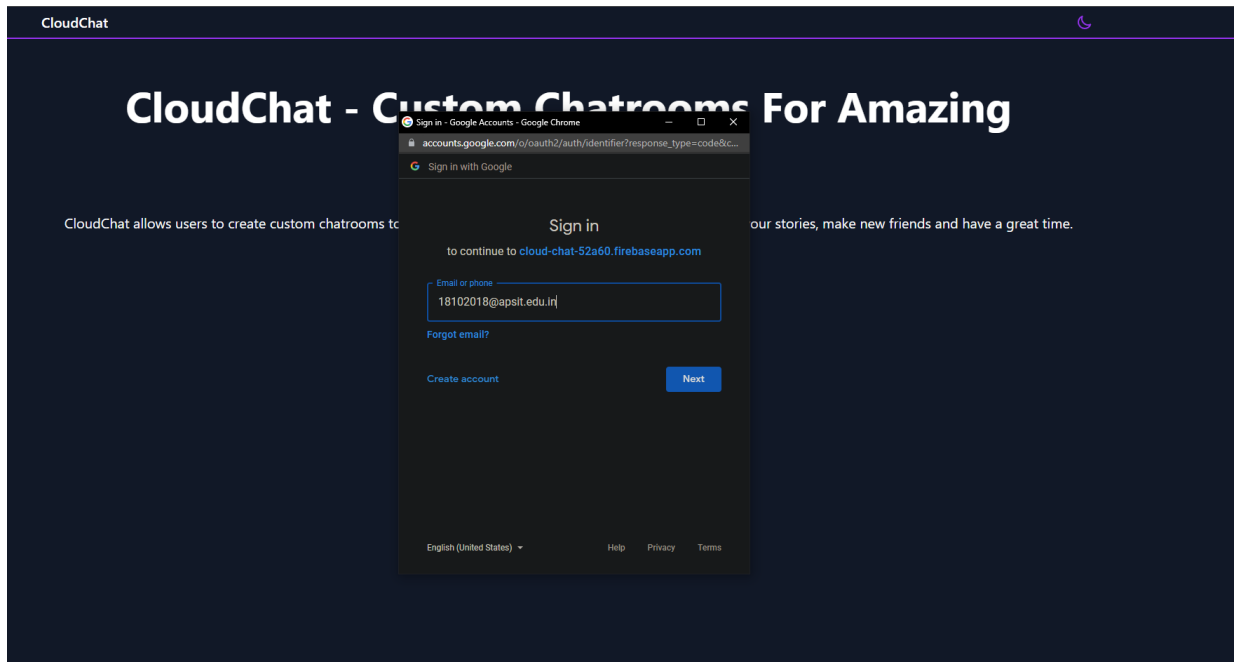Figure 4.1: Application UI / Home Page

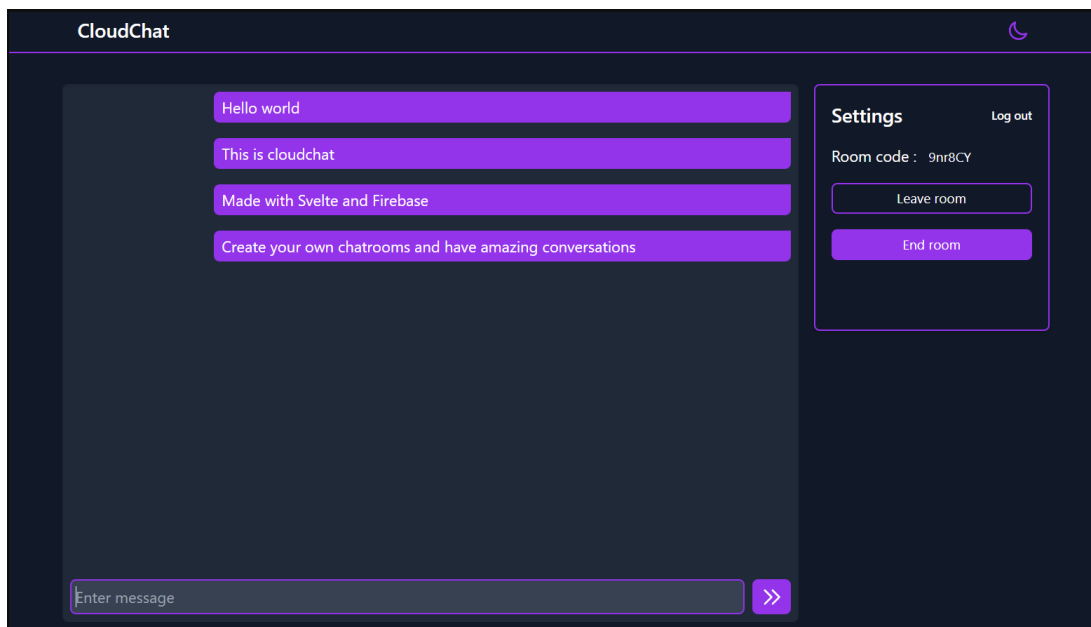Figure 4.2: Application Sign on w/ Google account
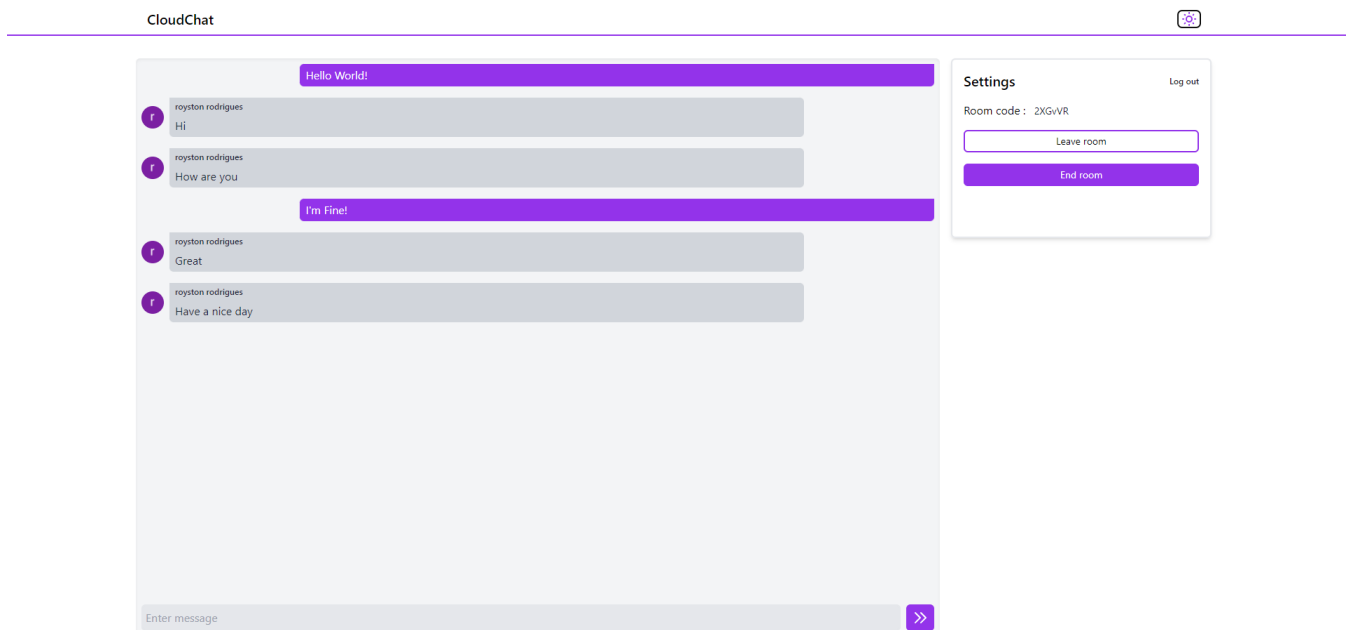

Figure 4.3: Chat Room Demo w/ Dark Mode

Figure 4.4: Chat room demo w/ Light Mode

Chapter 5

Testing

Testing was conducted as follows

1. Cross Platform testing: Application was tested on three different Operating systems-Windows , Ubuntu Android in the OS native browsers and the application worked error-free.

2. UI Testing - Responsiveness was tested using the chrome developer tools and after deployment the application was tested on different browsers and on four different devices with different sizes and it was found that the web app is responsive and resizes itself according to changing screen bounds.

3. Core application testing: The Application logic - Room creation , Chatting , Room deletion and theme changing was thoroughly tested and the application worked without any fatal errors.

Chapter 6

Result

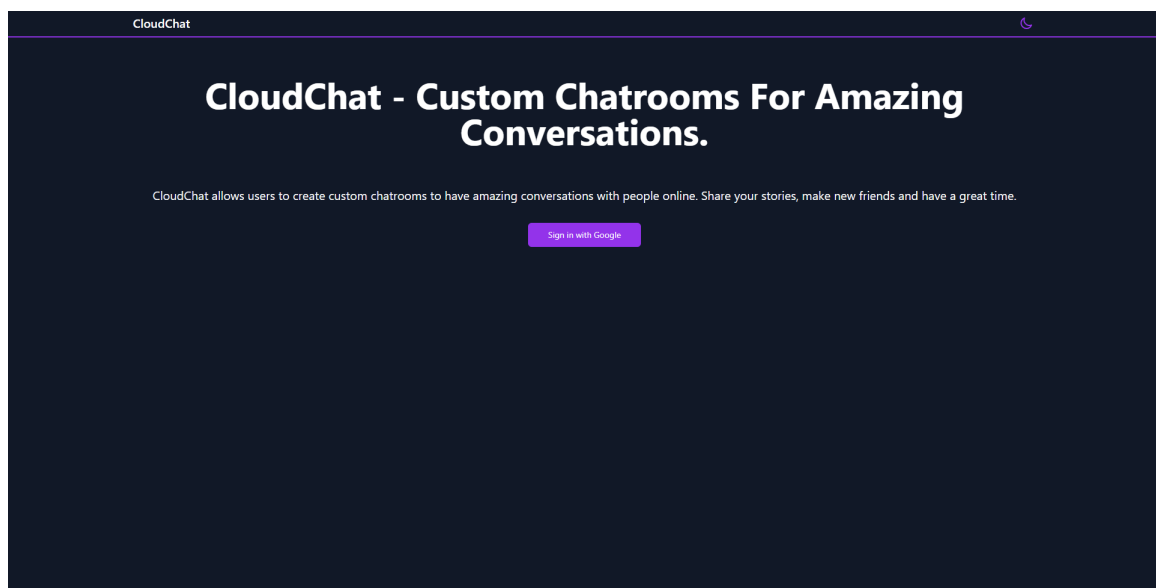The deployed app can be viewed at https://cloud-chat-52a60.web.app/
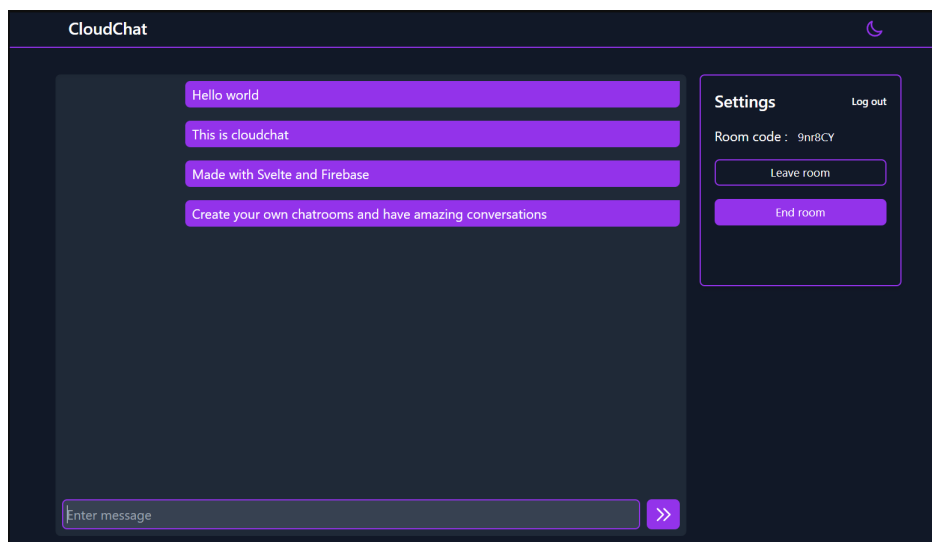

Figure 6.1: Landing Page

Figure 6.2: Chat Room Window

Chapter 7

Conclusions and Future Scope

Therefore , a foundational and functional web chat application has been built , complete with the ability to make custom chatrooms and have private conversations , without having to worry about manual authentication. Github link of the repository: https://github.com/ NoobProgrammerRoy/cloudchat-firebase-app

Future Scope

1. Adding File Upload support.

2. Adding Thread based replies.

3. Custom profile page for general settings and adding a status.

Bibliography

[1] S. Shukla, S. C. Gupta and P. Mishra, "Android-Based Chat Application Using Firebase," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021.

[2] M. A. Mokar, S. O. Fageeri and S. E. Fattoh, "Using Firebase Cloud Messaging to Control Mobile Applications," 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2019.

[3] Xiaoping Huang , "Research and Application of Node.js Core Technology" , ICHCI 2020.