A Mini Project Report

On

CHAT ROOM USING REACT - FIREBASE

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Course

Cloud Computing Laboratory
In
Computer Engineering (VIII SEM)

Submitted By

17102071 Vishnu Velukutty
1702035 Parth Shah
17102033 Khush Shah

Subject Incharge

Prof. Deepak Khachane

Department Of Computer Engineering
A. P. SHAH INSTITUTE OF TECHNOLOGY
THANE – 400 615
UNIVERSITY OF MUMBAI

Academic Year 2021 – 22

Department of Computer Engineering
A. P. Shah Institute of Technology
Thane – 400 615

CERTIFICATE

This is to certify that the requirements for the project report entitled '**Chat room using React-Firebase**' have been successfully completed by the following students:

Name	Roll No.
Vishnu Velukutty	9
Khush Shah	53
Parth Shah	56

in partial fulfillment of the course Cloud Computing Laboratory in Computer Engineering (VIII SEM) of Mumbai University in the Department of Computer Engineering, A. P. Shah Institute of Technology during the Academic Year 2019 – 20.

External Examiner

_____ _____

(Prof. Deepak Khachane)
Subject In-charge

Date:

Place: Thane

Department of Computer Engineering
A. P. Shah Institute of Technology
Thane – 400 615

PROJECT APPROVAL

This project entitled "**Chat room using React-Firebase**" by Vishnu Velukutty, Parth Shah, Khush Shah are approved for the course Cloud Computing Laboratory in Computer Engineering (VIII sem) of Mumbai University in the Department of Computer Engineering.

Subject In-charge:

_____

Date:

Place: Thane

Department of Computer Engineering
A. P. Shah Institute of Technology
Thane - 400 615

DECLARATION

We declare that this written submission for Cloud Computing Laboratory mini project entitled "Chat Room using React-Firebase" 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 ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not
been taken when needed.

Project Group Members:

Vishnu Velukutty & Sign:

_____

Parth Shah & Sign:

_____

Khush Shah & Sign:

_____

_____

Date:

Place:

Table of Contents

Abstract

The term chat room, or chatroom (and sometimes group chat; abbreviated as GC), is primarily used to describe any form of synchronous conferencing, occasionally even asynchronous conferencing. The term can thus mean any technology ranging from real-time online chat and online interaction with strangers (e.g. online forums) to fully immersive graphical social environments.

The primary use of a chat room is to share information via text with a group of other users. Generally speaking, the ability to converse with multiple people in the same conversation differentiates chat rooms from instant messaging programs, which are more typically designed for one-to-one communication. The users in a particular chat room are generally connected via a shared internet or other similar connection, and chat rooms exist catering for a wide range of subjects. New technology has enabled the use of file sharing and webcams.

List of Figures

Chapter 1

Introduction

1.1 Fundamentals

Cloud computing, sometimes referred to simply as "cloud," is the use of computing resources — servers, database management, data storage, networking, software applications, and special capabilities such as blockchain and artificial intelligence (AI) — over the internet, as opposed to owning and operating those resources on premise.

Cloud computing's ability to provide on-demand availability of computing resources as services over the internet eliminates the need for enterprises to procure, configure, or manage resources themselves, and they only pay for what they use.

There are three types of cloud computing service models:

Infrastructure as a service: Offers compute and storage services

Platform as a service: Offers a develop-and-deploy environment to build cloud apps

Software as a service: Delivers apps as services.

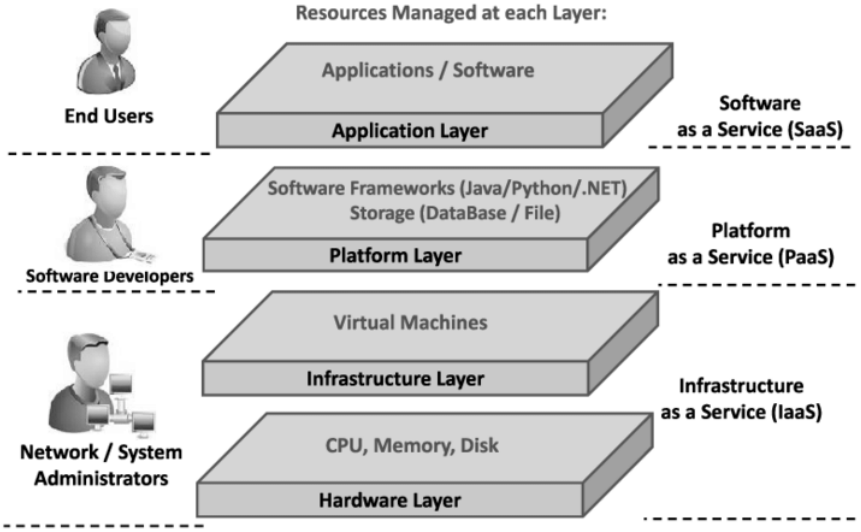

Fig 1.1: Basic Cloud Computing Architecture

The Figure 1.1 shows the Basic cloud computing architecture and the services managed by each individual in each model

1.2 Objectives

The objective of this work is as follows:

1. To build an app having to invest less in hardware resources
2. To enable chatting between anyone having access to personal computer, laptop, or phone and decent internet connection
3. To reduce hassles of registration using single authentication method for access the chatroom
4. To chat in any language

1.3 Organization of the Report

The report is organized as follows:

The Chapter 1 gives introduction to the topic, it describes the fundamental terms used in this project. It motivates to study and understand the different techniques used in this work. This chapter also presents the outline of the objective of the report.

The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique.

The Chapter 3 presents the Theory and proposed work. It describes the major approaches used in this work.

The Chapter 4 describes the societal and technical applications are mentioned.

The Chapter 5 presents the summary of the report.

Chapter 2
Literature Survey

2.1 Introduction

Chat room is a Web site, part of a Web site, or part of an online service such as America Online, that provides a venue for communities of users with a common interest to communicate in real time. Forums and discussion groups, in comparison, allow users to post messages but don't have the capacity for interactive messaging. Most chat rooms don't require users to have any special software; those that do, such as Internet Relay Chat (IRC) allow users to download it from the Internet. Chat room users register for the chat room of their choice, choose a user name and password, and log into a particular room (most sites have multiple chat rooms). Inside the chat room, generally there is a list of the people currently online, who also are alerted that another person has entered the chat room. To chat, users type a message into a text box. The message is almost immediately visible in the larger communal message area and other users respond. Users can enter chat rooms and read messages without sending any, a practice known as lurking.

Because chat room messages are spontaneous and instantly visible, there is a potential for abuse, which may or may not be intentional. Site hosts typically post a frequently asked questions (FAQ) list to guide users to appropriate chat room behavior, such as introducing yourself when you enter a room, making it clear when you are directing a question or response to a specific user, and reporting disruptive users, for example. Disruptive users may verbally abuse other chatters, monopolize the conversation, or even just disable it by repeatedly typing the same word or phrase into the conversation, a practice (much frowned upon) known as scrolling.Chat rooms can be found that focus on virtually any aspect of human endeavor or interest: there are current communities based on classic movies, Irish ancestry, baton twirling, and psychic readings, for example. Various sites, such as Yahoo, provide a directory of chat sites. Others, such as MSN Web Communities, guide users through the steps required to create their own chat room.

2.2 Literature Review

Chat rooms are Web sites or programs that allow people to send text messages to one another in real time. The chat room works as a virtual room, where groups of people send messages that others can read instantaneously. Often, people in chat rooms will use aliases or nicknames to provide some anonymity, and will use the chat room to meet others. If individual visitors to the chat room wish to talk privately, they can enter a private chat room, which allows two or more people to send messages privately to one another.

Cloud computing has received increasing interest from enterprises since its inception. With its innovative information technology (IT) services delivery model, cloud computing could add technical and strategic business value to enterprises. However, cloud computing poses highly concerning internal (e.g., Top management and experience) and external issues (e.g., regulations and

standards). This presents a systematic literature review to explore the current key issues related to cloud computing adoption. Using the grounded theory approach, articles are classified into eight main categories: internal, external, evaluation, proof of concept, adoption decision, implementation and integration, IT governance, and confirmation. Then, the eight categories are divided into two abstract categories: cloud computing adoption factors and processes, where the former affects the latter. The results of this review indicate that enterprises face serious issues before they decide to adopt cloud computing. Based on the findings, the paper provides a future information systems (IS) research agenda to explore the previously under-investigated areas regarding cloud computing adoption factors and processes.

Other use of chatrooms are by developers, since communication is critical for the software development teams to maintain project awareness, facilitate project co-ordination and avoid misunderstandings. The features offered in the chatrooms, such as private messaging, group conversations, and code sharing help accommodate the communication needs of the software development teams. Therefore, chatrooms have been increasingly adopted among the developers. Since the last study on Slack performed in 2016, the audience of Slack has more than doubled possibly leading to an evolution of the ways Slack is used; while another rich community formed around Gitter and remains unstudied. In this paper, we perform an investigative study using qualitative and quantitative techniques to gain insights on the use of popular modern chatrooms, specifically Slack and Gitter. Based on the survey responses from 163 developers, the interviews with 21 developers, and the chatroom data collected from 11 Slack and 770 Gitter rooms, we are able to uncover the reasons behind the use of Slack and Gitter, the perceived impact on the associated projects, and the quality determinants of the two chatrooms. We find that the developers seek knowledge from the chatrooms to obtain timely feedback from experts, and in return share their expertise to build the project community and their reputations. Furthermore, it is perceived by the Gitter developers that the chatrooms have an impact on prioritizing the new features and the bug fixes. In Slack, the most reported impact concerns an increased project awareness, in terms of a better tracking of the work progress. As reported on the developers survey, both Slack and Gitter chat services have a visible impact on mentoring developers, and sharing the best practices. In terms of quality determinants, a non-ephemeral history and a better history management (e.g., advanced search) could be keys for both chat services to reach their full potential.

Among other groups, chat rooms appeal to adolescents and teenagers, who communicate with others and use features to send files to one another online. Because so many young people enjoy chat rooms and instant messaging, these features are also being incorporated into other tools. For example, a game used on an Xbox or PlayStation may provide features so that you can chat with people you're playing against online. Although this can be fun for those with a real interest in the game, it can also be used by pedophiles to connect with children and adolescents who are trying to connect with young people in new ways.

Chapter 3
Design Phase

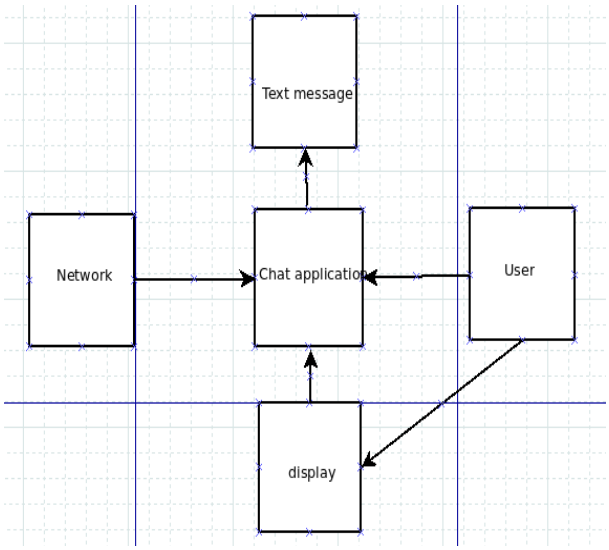

Fig 2.1 Block Diagram

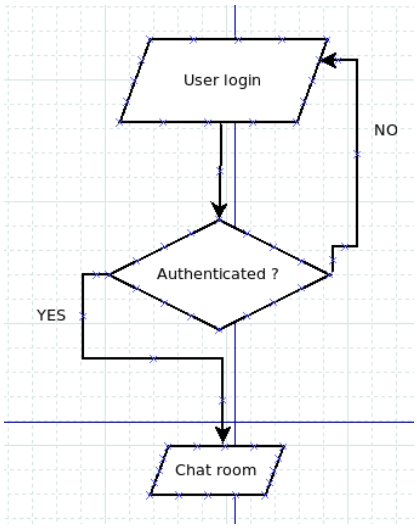

Fig 2.2 Flowchart

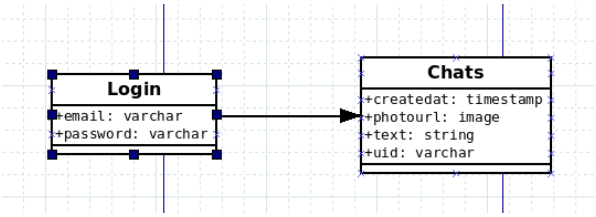

Fig 3.3 UML Diagram

Chapter 4
Implementation Details

4.1 Overview

The User Interface (Front-end) is created with JavaScript using React framework and its Back-end is exclusively based on Firebase

4.2 Implementation Details

The User Interface (Front-end) is created with JavaScript using React framework since it is a free and open-source front-end development

- Declarative: React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable, simpler to understand, and easier to debug.
- Component-Based: Build encapsulated components that manage their state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep the state out of the DOM.
- Learn Once, Write Anywhere: We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using Node

and its Back-end is exclusively based on Firebase

This helps in building apps by reducing the need of managing servers. Helps in effortless scaling up to support millions of users with Firebase databases, machine learning infrastructure, hosting and storage solutions, and Cloud Functions. Additionally Firebase Authentication and Security Rules are helpful to create a secure app.

The data is stored in Firebase Firestore which can take in real-time data and store and order (limited to 25 in our app). Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. It is an alternative to Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity.

Steps for front-end :

- The User Interface is created in React Framework using JavaScript

Steps for back-end :

- In the frontend code import all Firebase SDK plugins and Hooks

- To Setup the backend first login to Firebase account
    1. Click on Console
    2. Click on add project
    3. Give a name to your project (in this case backend)

- Setup up authentication methods by selecting Google sign-in method
- Go to left pane select Project Overview then add Firebase to app option
- Select Web as option
- Register your app by putting in a name
- Check the option that says “Also setup Firebase Hosting” and register the app
- This will give you the keys required to connect your app to Firebase add them in the front-end code (in the initializeApp function)

Steps for setting-up database :

- Select the Cloud Firestore
- Update the existing rules to match your requirements and save

Steps for hosting :

- In the root of your project folder run `npm run build`
- After this login into your firebase account which is used for hosting
- Type `firebase init` and select option Hosting
- Update the details of backend in the site parameter accordingly (backend)
- Re-run build command
- Run `firebase deploy`

3.2.2 Details of packages, data set

React Firebase Hooks library explores how React Hooks can work to make integration with Firebase even more straightforward than it already is. It takes inspiration for naming from RxFire and is based on an internal library that we had been using in a number of apps prior to the release of React Hooks. The implementation with hooks is 10x simpler than our previous implementation.

Chapter 5
Project Inputs and Outputs

5.1 Inputs Details

The inputs for the project is basically a chat message that a logged-in user will enter and it will be displayed on the screen for another logged-in user to view. This should be visible to another user if he/she logs in after some time.

4.2 Output Details and Screenshots

Fig 3.1 Chatroom login

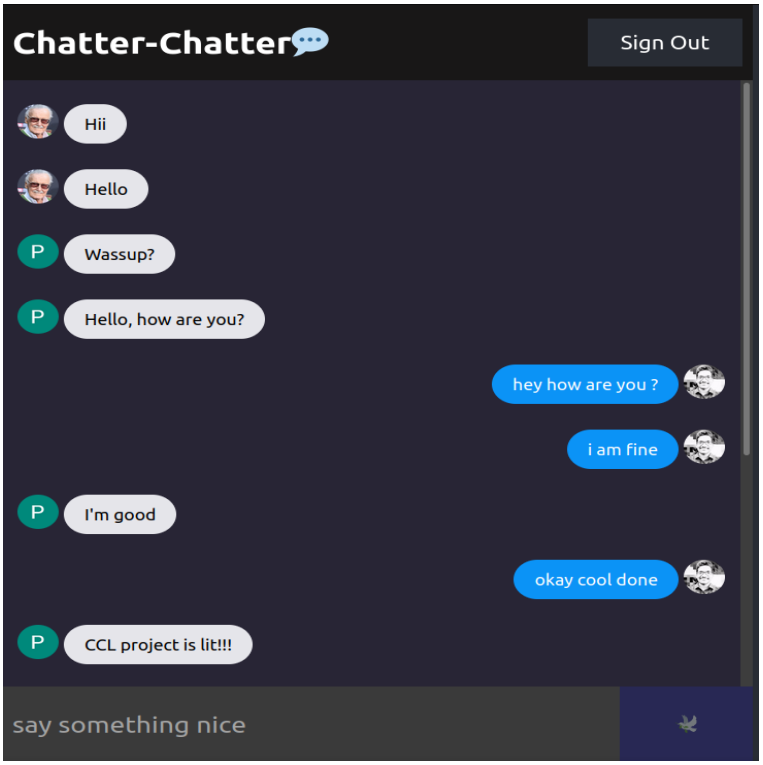

Fig 3.2 Chatroom Window

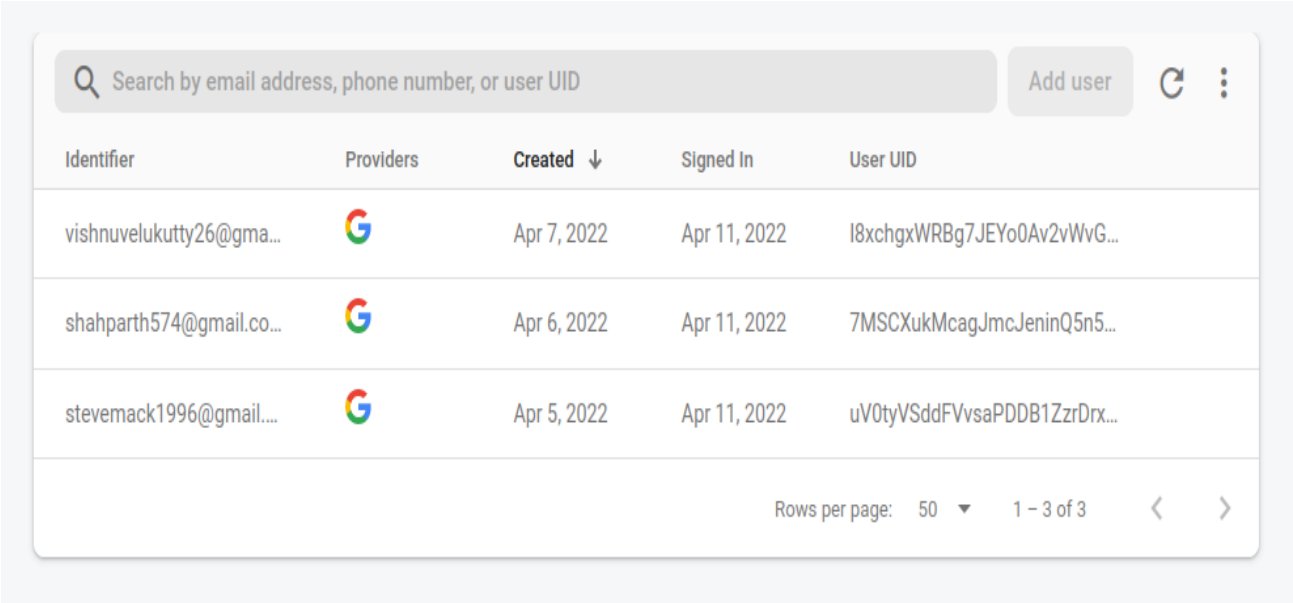

Fig 3.3 Chatroom Authenticated users

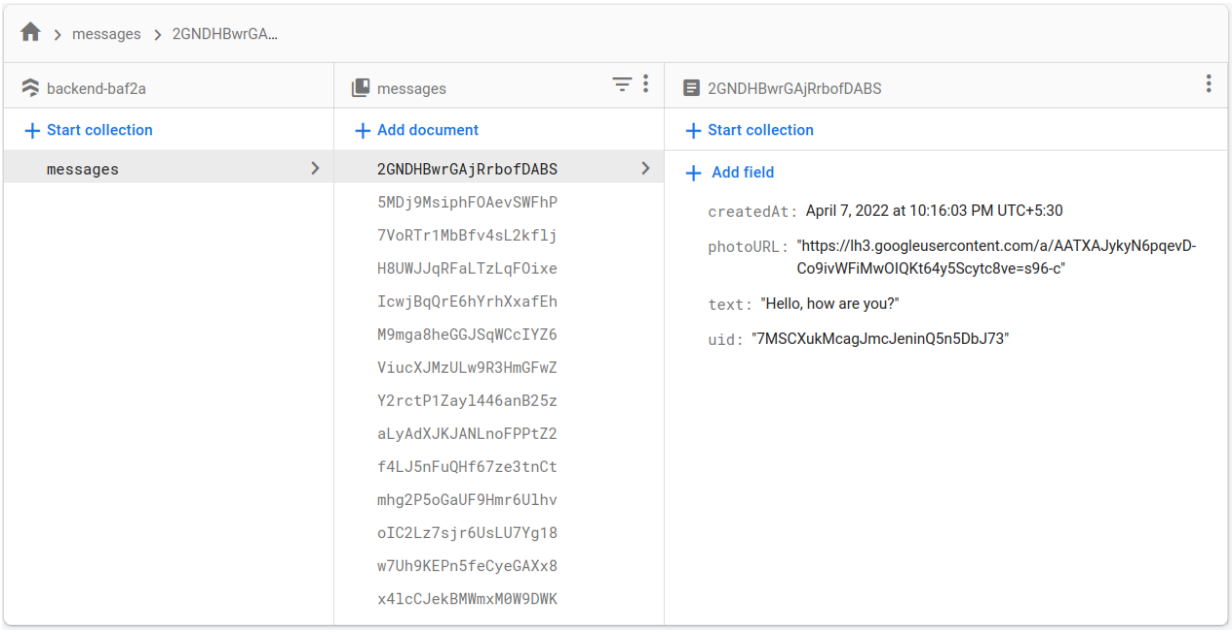

Fig 3.4 Chatroom Messages

Chapter 6
Result and Future Scope

6.1 Result

A Chat room was successfully developed using React framework, deployed and hosted on Firebase and can accessed from anywhere around the world using a mobile, laptop or desktop. User can login using any Google account and type up-to 255 characters in the input field. All users can view previous chats irrespective of login time. Additionally a user can chat in any language since the application supports all languages irrespective of the Unicode.

The data is stored and ordered according to the time-stamp and can be accessed via Firebase console. Additionally the users that have logged in can also be viewed by the Root/Admin user from Firebase console login this data is ordered sequentially with date, time and day along with unique id and can store data irrespective of Unicode.

The app itself is hosted in Firebase which enables user from any part of the world to access it via web and start chatting after logging into google account.

The app has it drawback first and foremost no multiple ways of authentication except for google, second there is no way of preventing when a user tries to use offensive language in the chats. Lastly this chatroom can't implement sharing of images and creation of groups for specific communities.

6.2 Future Scope

1. Implementation bad word filtering and banning user.
2. Implementation of groups creation.
3. Addition of more authentication methods.
4. Implementation of sharing images.

References

Firebase Documentations. https://firebase.google.com/docs

React Documentations. https://reactjs.org/docs/getting-started.html

Npm Documentations. https://docs.npmjs.com/

Understanding Chatrooms. https://www.sciencedirect.com/topics/computer-science/chat-room

Literature Review on Cloud computation adoption
https://www.researchgate.net/publication/278698086_A_Literature_Review_on_Cloud_Computing_Adoption_Issues_in_Enterprises

Github : https://github.com/ParthBShah/Chat-Room