



CZAT: a Web Application Based Community Chat App

Kuldeep Gupta, Niraj Srivastava and Vaishnav Goswami

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 20, 2021

CZAT – A Web Application Based RealTime Chat App

Kuldeep Gupta
Btech CSE
Galgotias University
Greater Noida,India
rakeshgupta_1971@rediffmail.com

Niraj Srivastava
Btech CSE
Galgotias University
Greater Noida,India
nirajsrivastava2001@gmail.com

Vaishnav Goswami
Btech CSE
Galgotias University
Greater Noida,India
vaishnav0501@gmail.com

Abstract— This research paper discussing the various useful tools and techniques that are used in a development of a real time chat app. We also discuss about the procedure follow in a chat app, mostly focused on a JavaScript library REACT JS. Next, we compare different chat applications with our real time chat app. This Paper tell about the technologies used in this development, JavaScript library React JS, APIs, CSS, HOOKS and PROPS explained in result its functionality. It is hoped it will give a useful framework for guiding the process.

Keywords—ReactJS, React Hooks, Props, CSS, APIs.

I. INTRODUCTION

The word “CZAT” is derived from the Polish language which means “chat”. The idea of creating this chat app to bring an organization closer, in which there are various departments and locations. So that they can get a vast enclosure. It will also have authentication to maintain user’s security, it also contains various styles which will make our user comfortable.

The chat app will support a user like all other chat boxes but the difference in this chat app is you can like the particular message and we can create as many community as we want. It also tells whether the user is active or not. This chat app also contains various permissions for different user to maintain the privacy and a healthy environment between users. It also contains the photo gallery in which a user can view all the posted photos without scrolling back to the chats.

II. RELATED WORK

A. React Hooks

Hooks are functions that let you “hook into” React state and lifecycle features from function components. Hooks don’t work inside classes — they let you use React without classes. React provides a few built-in Hooks like useState. You can also create your own Hooks to reuse stateful behavior between different components. We’ll look at the built-in Hooks first.

B. APIs

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.

C. ReactJS

React is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.

D. React Props

In some cases, those components need to communicate (send data to each other) and the way to pass data between components is by using props.

“Props” is a special keyword in React, which stands for properties and is being used for passing data from one component to another.

But the important part here is that data with props are being passed in a uni-directional flow.

III. LITERATURE SURVEY

Messaging apps (a.k.a. "social messaging" or "chat applications") are apps and platforms that enable instant messaging. Many such apps have developed into broad platforms enabling status updates, conversation.

Some examples of popular messaging apps include WhatsApp, Facebook Messenger, Telegram and Snapchat. The popularity of certain apps greatly differ between different countries. Certain apps have emphasis on certain uses - for example Skype focuses on video calling, Slack focuses on messaging and file sharing for work teams, and Snapchat focuses on image messages. Some social networking services offer messaging services as a component of their overall platform, such as Facebook's Facebook Messenger, while others have a direct messaging function as an additional adjunct component of their social networking platforms, like Instagram, Reddit and Twitter, either directly or through chat rooms.

Messaging apps are the most widely used smartphone apps, with in 2018 over 1.3 billion monthly users of WhatsApp and Facebook Messenger.

IV. PROJECT OBJECTIVE

This project aim to build an online chat system that has the capability to have real-time communication using web browser. WebRTC approach will be implemented for achieve this real time communication. WebRTC approach does not require any additional plug in to the browser with the constraint that the browser support webRTC approach such as

Google Chrome. However the screens sharing in the WebRTC approach require install a plugin.

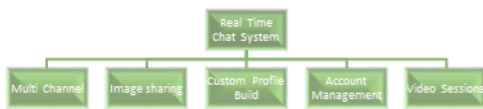
1. To provide an online consultation solution:

The traditional way to have consultation might face some issues such as availability of the student and lecturer in the school.

2. To provide a more systematic appointment system: A systematic appointment system will help in manage the appointment more effective and efficient. This appointment system is convenient because a template is provided to completed and sent.

3. To provide a SMS reminder to remind the lecturer about the consultation: A SMS reminder will be sent to both parties, lecturer and student to remind them about they will going to have a consultation later. This will ensure that both parties will not miss out the appointments that have been made before.

4. Proposed approach



V. SCOPE

This project will be developed in web based. The project is planned to introduce an online web chat system solution for the student and lecturer. The project included an appointment system that will be handling all the appointment between the student and the teacher effectively. Furthermore, a real time communication chat system will be included as the feature in the project to make a face to face communication channel between the student and the teacher. There are additional feature included in the project such as the screen sharing that share the current screen of the user, file sharing and text chat.

VI. FEASIBILITY ANALYSIS

- Many communities and users can access the web-app at the same time
- The CZAT web-app is free of cost
- The user can access their chats from anywhere.
- The web-app is responsive and can fit to any device's browser
- This is a Real-time chat app.

VII. REQUIED TOOLS

- React.js : frontend
- Chat Engine rest Api : backend
- Npm – used to run reactjs
- Visual Studio code – coding platform
- Sockets – for two way interactive communication between user's and a server.

VIII. FEATURES

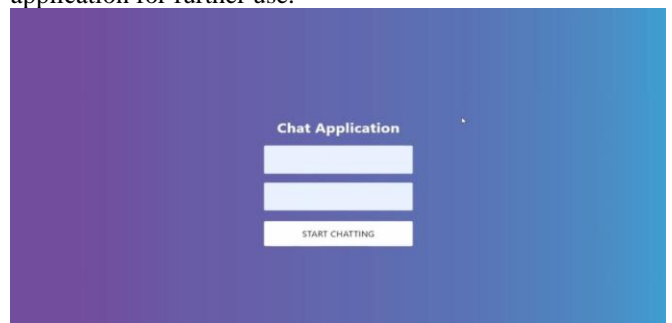
- Many users can access the real time chat web-app at the same time.

- The chat app also consists of login page, sign up page.
- User can also build his profile.
- User can create the community to chat with its like minded people.
- The users can also react on the posts
- The user can also share images to other users.

IX. RESULTS

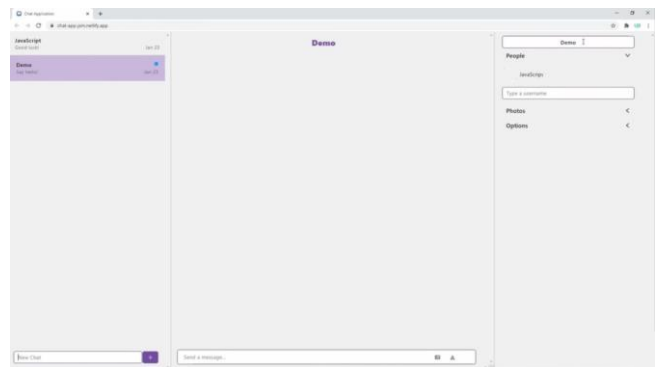
A. Login Page:

In the image, we can provide information such as User Name and password. User can successfully login to chat application for further use.



B. Profile Page

We have added page to add full details of User as their personal.



X. CONCLUSION

Instant Messenger is a proprietary, simplified version of Internet Relay Chat, which allows two or more people to carry on a conversation, in real time, using text based messages with context awareness.

A lot of people are available which provide messages free of charge. Since they are free of charge, they are the preferred services by millions

of people around the world. Some of the Mobile Messaging Applications those are generally used : Hike, Chat On, Whatsapp, E-buddy messenger, Facebook etc.

CZAT is a type of real time chat app which have extra functions which are not used by these existing Mobile Messaging Apps.

XI. REFERENCES

- MDN Web Docs - <https://developer.mozilla.org/en-US/>
- React Hooks - <https://reactjs.org/docs/hooks-overview.html>
- ReactJS - <https://reactjs.org/docs/getting-started.html>
- React Props - <https://reactjs.org/docs/components-and-props.html>
- GitHub - <https://github.com/>
- Sockets- <https://socket.io/docs/v3/index.html>