**Mini Project Report on**



**Mini Chat App in React.js**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name :**  **University Roll No.:**

**Kushagra Sharma 2019678**

***Under the Mentorship of***

**Ms. Tanisha Mittal**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Mini Chat app in React.js”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Ms. Tanisha Mittal, Designation**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name : Kusharga Sharma University Roll no. : 2019678

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-2** |
| Chapter 2 | Literature Survey |  |
| Chapter 3 | Methodology |  |
| Chapter 4 | Result and Discussion |  |
| Chapter 5 | Conclusion and Future Work |  |
|  | References |  |

**Chapter 1**

**Introduction**

With the advancement of internet, more and more people have been using online chatting tools for communication. These applications allow real-time communication all across the globe.

This programming tools used in this application are React.js, firebase and tailwind css.

* 1. **Introduction**

In this report we will be discussing a chat application that was developed using React.js, Firebase and Tailwind CSS. The goal of this project was to create a user friendly and efficient chat application that could be used by the people from all over the world. The web app is designed to be responsive and work seamlessly for the, mobile and desktop devices.

The application is built in React.JS, a popular JavaScript library for building user interfaces and it was chosen due to its ability to efficiently handle user interactions and real-time updates. It allows development of reusable components that can be easily integrated into the application, making it easy to maintain and scale.

Firebase on the other hand, is a Backend-as-a-Service (BaaS) platform that provides a wide range of services for building web and mobile applications. For this chat application, we used Firebase’s Real time Database ( Firestore database) and authentication services to handle data storage and user management. Firestore database allows for real time data syncing across all connected clients, ensuring that the users are always viewing the most up-to-date information. Authentication provides easy-to-use user management, including g-mail and password based authentication, as well as integration with popular platforms like Facebook, GitHub.

The chat application has several key features that make it suitable for a wide range of use cases and user groups. One of the main features is real-time messaging, which allows users to communicate with each other in real-time, without any delays or lag. Users can send and receive messages in real-time, with notifications appearing instantly on their screens. This makes the chat app perfect for use cases such as team collaboration, customer support, and online communities.

Another important feature is user authentication, which ensures that only authorized users can access the chat app. This feature ensures that users can only view and participate in conversations that they have been granted access to, making it suitable for use cases such as private messaging and group chats.

Lastly, Tailwind CSS was used to style the application. Tailwind CSS is a utility-first CSS framework that provides a set of classes that can be used to style HTML elements. The benefits of using Tailwind CSS include the ability to quickly style elements and the ability to easily create a consistent design across the entire application.

The scope of the project should be broken-down and the system should be declared before advancing further. The scope are as follows:

1. The design and construction of this application is aimed at building a web-based application and mobile application

2. This system is developed using React.js.

3. The database of this system implemented using Firebase

In conclusion, the chat application developed using React.js, Firebase, and Tailwind CSS is a powerful and user-friendly tool that can be used by people from all over the world. Its real-time capabilities and seamless integration with Firebase makes it a perfect choice for chatting. It is also responsive and can be used on both desktop and mobile devices, making it accessible to a wide range of users.

**Chapter 2**

**Literature Survey**

**2.1 React.Js –**

React.js is a popular JavaScript library for building user interfaces. It allows for efficient updates and rendering of components, making it a good choice for building a chat app. Many existing chat apps, such as Facebook Messenger and WhatsApp, have been built using React.js.

Features of React.Js –

**2.11 JSX-JavaScript Syntax Extension-**

JSX is a syntax extension to JavaScript. It is used with react to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code. This makes the code easier to understand and debug, as it avoids the usage of complex JavaScript DOM structures.

**2.12 Virtual DOM –**

React keeps a lightweight representation of the real DOM in the memory, and that is known as “Virtual DOM(VDOM).

**2.2 Firebase-**

Firebase is a mobile and web application development platform that provides a number of services, including real-time databases and authentication. It is a useful tool for building chat apps because it allows for easy synchronization of messages between multiple users. Some examples of chat apps that use Firebase include Slack and GroupMe.

**2.22 Cloud Firestore –**

NoSql database built for global apps, Cloud Firestore is a NoSql document database that lets you easily store, sync, and query data for your mobile and web apps-at global scale

**2.23 Authentication –**

Firebase Authentication provides backend services, easy-to-use SDKs, and readymade UI libraries to authenticate users to your app. Normally, it would take you months

to set up your own authentication system. And even after that, you would need to keep

a dedicated team to maintain that system. But if you use Firebase, you can set up the

entire system in under 10 lines of code that will handle everything for you, including

complex operations like account merging.

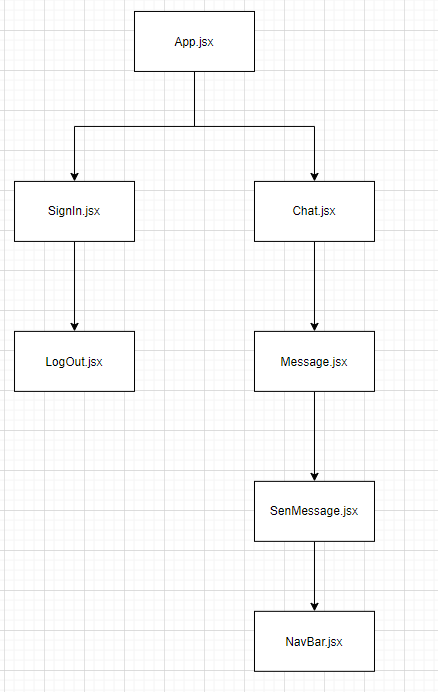
**2.3 Tailwind CSS-**

Tailwind CSS is a utility-first CSS framework that allows for easy styling of web pages. It is a popular choice for building chat apps because it provides a wide range of pre-defined classes that can be used to quickly style elements. Some examples of chat apps that use Tailwind CSS include Discord and Slack.

**Chapter 3**

**Methodology**

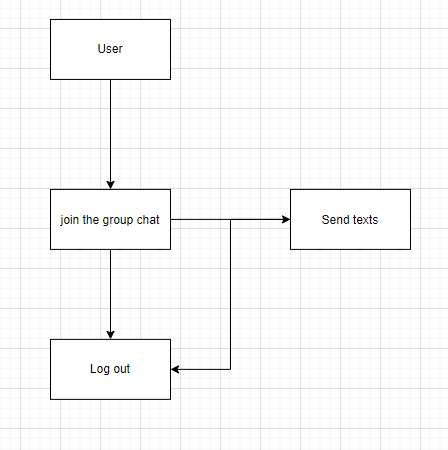
The System Design is based on web application design which allows users to chat remotely and get connected with others. This system design contains main modules such as app.js which includes login module signup module and Home module where users are landed once they login securely. We briefly explain these modules below



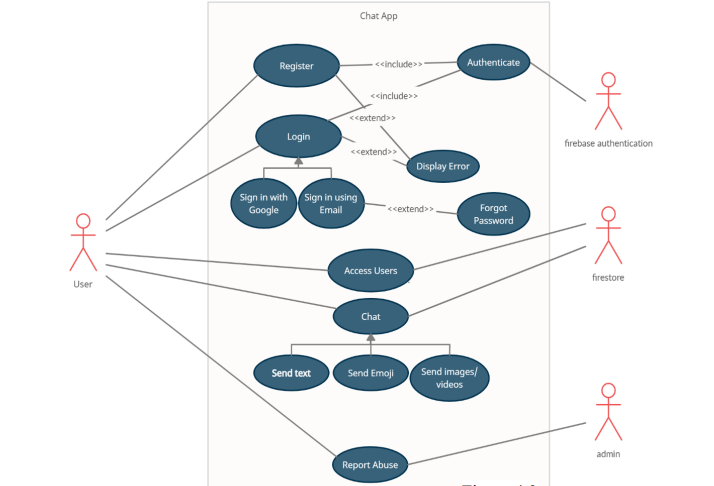
**Login/Signup Module –**

The login module allows to the user to sign in or sign up using the google account and all the details are stored via the auth tokens.

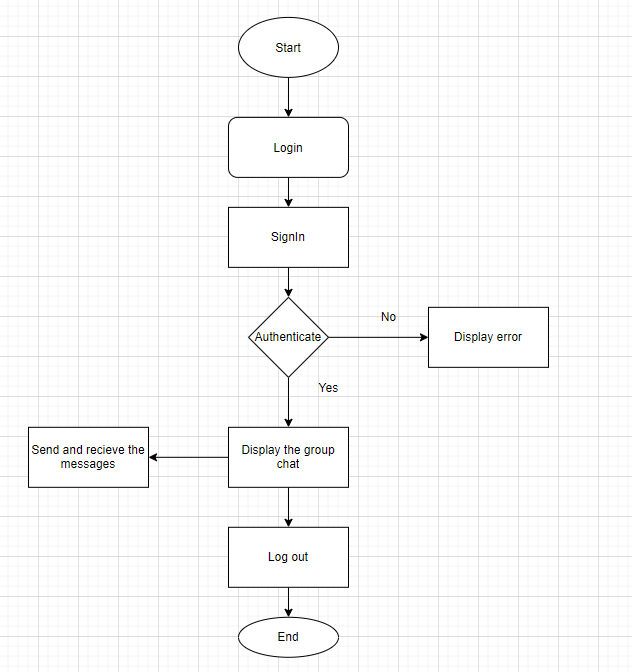
**Flow diagram –**

****

**Use Case Diagram –**

****

**Flow chart –**

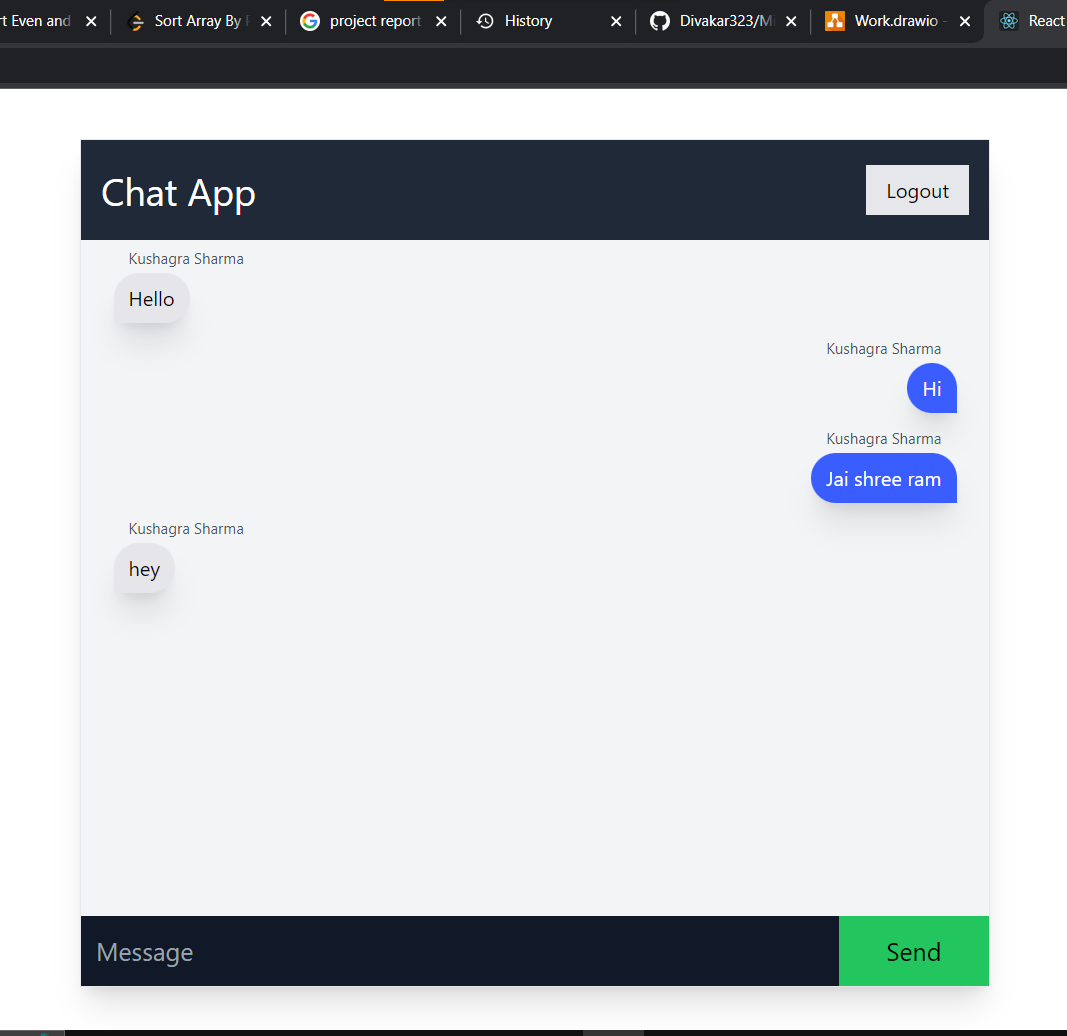
****

**Chapter 4**

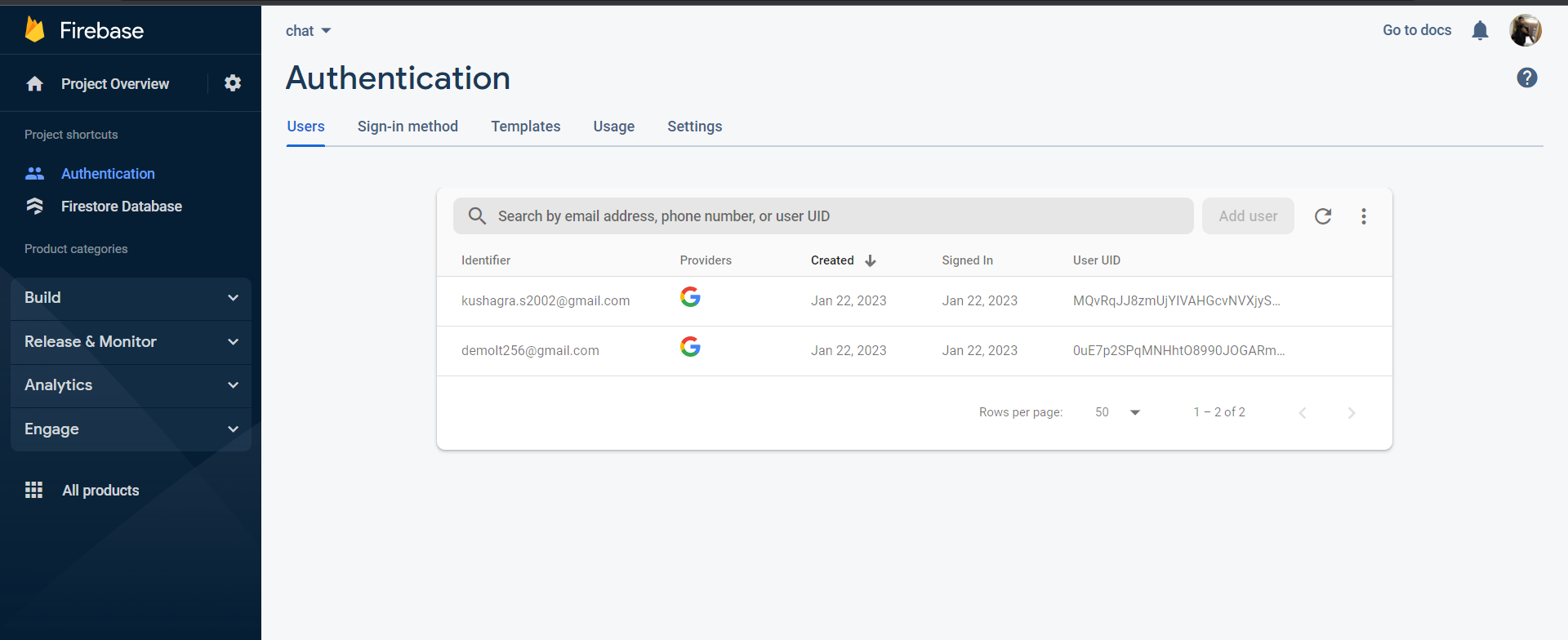
**Result and Discussion**

The above mentioned methodology resulted in the development of a single page react app.

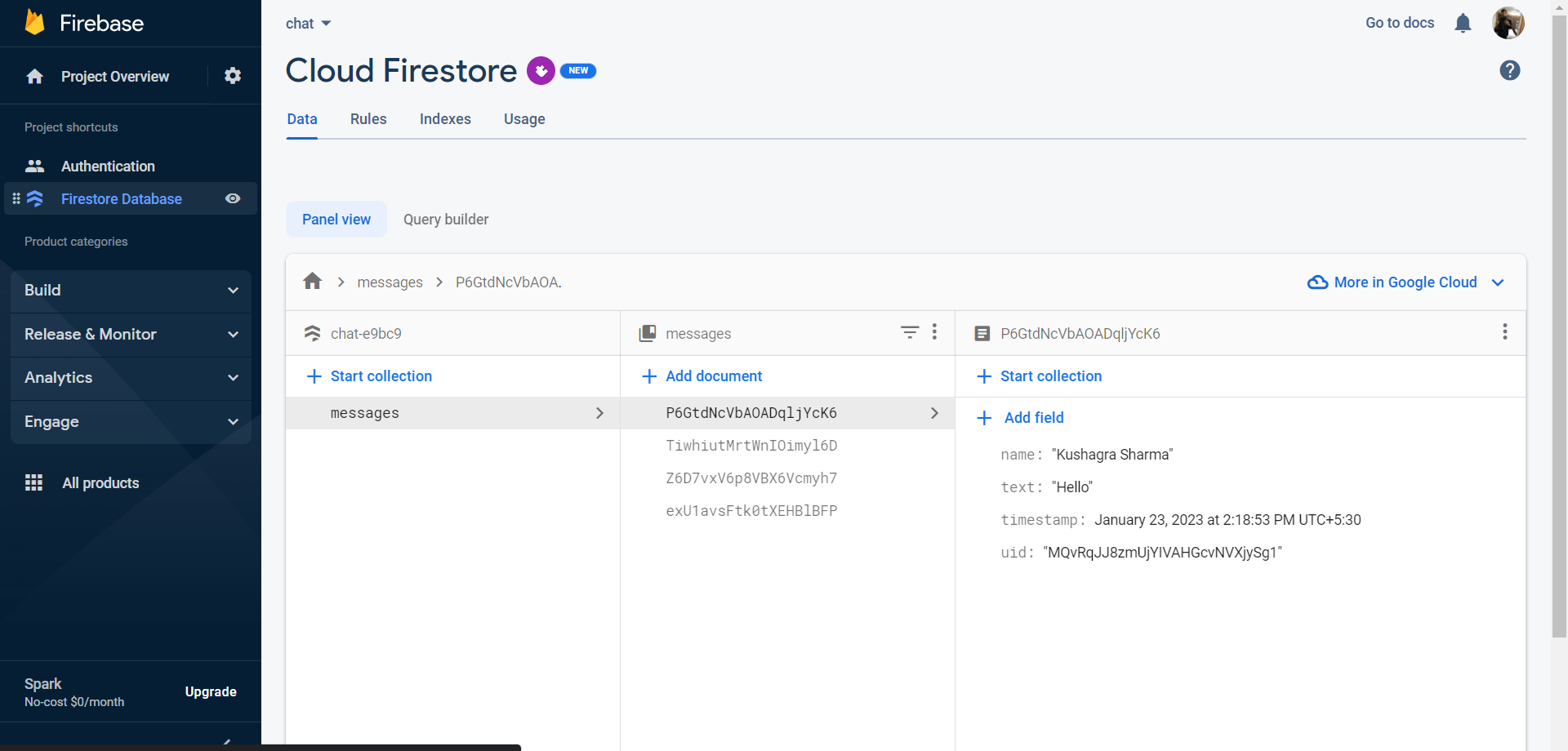
This web app allows the user to sign in to join the group chat. The user interface is inspired by a minimalistic design, with easy to access and understand usage.

****

The back-end managed by firebase stores the auth tokens of the accounts that have been logged in in the web app. The users can be manually removed in case of any security issues by the admins or automatically by the server in case of any profanity.



All the data in the web app is stored in real-time on the server/database using firestore databse. This includes the message id, sender id and its timestamp.



**Chapter 5**

**Conclusion and Future Work**

The project mini chat app with reac.js was completed successfully. On running the tests, it can be inferred that the chat application developed using React and Firebase is faster in real-time with a speed less than a second compared to the application developed using PHP and MySQL, since Firebase uses NoSQL database . This web app connects all the people who like to chat together in a community.

The future work may include the addition od one-one conversations, the ability to allow users send pictures, voice notes and other forms of media other than just text, and find other people to have a conversation with their unique id or username.

**Reeferences**

[1] Croucher, T.H., & Wilson, M. (2012). Node: Up and Running. United States

[2] Keissling, Manuel. 2012. The Node Beginner Book. Lulu.com, United States

[3] Teixeira, Pedro. 2012. Hands-on Node.js. Wrox.

[4] Sidik, B. (2011). JavaScript. Bandung: Informatika

[5] Purnomosidi, B. (2013). Penbangan Sistem Informas Penegelolaan Inventaris Barang Divisi Pustekin Berbasis Web. Bandung: Politeknik Telkom.

[6] Tim A. Majchrzak University of Agder, Kristiansand, Norway Andreas Biørn-Hansen Westerdals, Oslo, Norway Tor-Morten Grønli Westerdals, Oslo, Norway

[7] Tim A. Majchrzak, Benjamin Ruland and Till Weber ¨ Department of Information Systems, University of Munster, M ¨ unster, Germany

[8] A study of internet instant messaging and chat protocols published on 14 August 2006 by R. B.

Jennings, E. M Nahum, D. P Olshefski, D Saha, Zon-yin Shae, Christopher J. Waters (http://ieeexplore.ieee.org/document/1668399/)

[9] Robert W. Sebesta: Programming the World WideWeb, 8th Edition Pearson Education, 2015