Project Report

On

<u>NOVO-CHAT APPLICATION</u>

Submitted by

K.Karthik Babu-R170978

Under the guidance of

Mr K.Vinod Kumar

Assistant Professor

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge and Technologies R.K.Valley, Kadapa

Andhra Pradesh

Rajiv Gandhi University of Knowledge Technologies

RK Valley, Kadapa (Dist), Andhra Pradesh, 516330

DECLARATION

We hereby declare that the report of the B.Tech Major Project Work entitled "**NOVO-CHAT APPLICATION**" which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any university or institution for award of any degree.

K Karthik Babu – R170978

RGUKT

(A.P.Government Act 18 of 2008)

RGUKT, RK VALLEY

Department of Computer Science and Engineering

CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled “**NOVO-CHAT APPLICATION**” submitted by **K Karthik Babu (R170978)** under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic year 2022-2023 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Project Internal Guide	**Head of the Department**
Mr K Vinod Kumar	Mr. N Satyanandaram
Assistant Professor,	HOD Of CSE ,
RGUKT, RK Valley.	RGUKT, RK Valley.

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director,Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr. N Satyanandaram for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Mr K Vinod Kumar for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

Index

Abstract

Chatting is a method of using technology to bring people and ideas together despite of the geographical barriers.The technology has been available for years but the acceptance was quite recent. Our project is an example of a chat server. Itis made up of two applications -the client application, which runs on the user's web browser and server application, runs on any hosting servers on the network. To start chatting client should get connected to server where they can do private and chat.

INTRODUCTION

Today Developers around the world are making efforts to enhance user experience of using application as well as to enhance the developer's workflow of designing applications to deliver projects and rollout change requests under strict timeline. Stacks can be used to build web applications in the shortest span of time. The stacks used in web development are basically the response of software engineers to current demands. They have essentially adopted pre-existing frameworks (including JavaScript) to make their lives easier.

Purpose

The purpose of NOVO-CHAT APPLICATION makes it easy to communicate with people anywhere in the world by sending and receiving messages.

Intended Audience

- ❖ END USERS

Product Vision

Broadcasting Chat Server Application is going to be a text communication software, it will be able to communicate between two computers using point to point communication.

Requirement Specification

Software Requirements:

Front end	Html,Css,javascript,Reactjs
Server side Language	Nodejs
Database Server	MongoDB
Web Browser	Firefox,Chrome, Any other compatible Browser
Operating System	Ubuntu,Windows,Linux

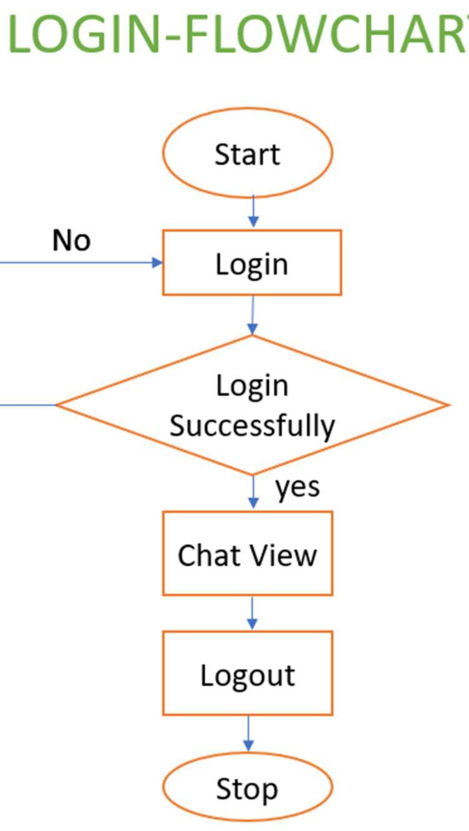
MODULES

- Login
- Registration
- Chat user

Modules Description

- **Login**

Login is a module to enter into the chat for the user.User has to Login with the email and password with captcha verification.



■ Registration

Registration is a module to create a user account and the required inputs are gmail,password,Name,Dp.

■ Chat User

The chat user interface is shown after successful login.User is able to chat with other users to communicate around the world.It allows group chats.

Advantages

- It Makes easy to communicate with people anywhere in the world by sending or receiving messages in the world.
- Instant messaging.
- Group Chat is possible.

Technologies Used

- FRONTEND : HTML ,CSS , JavaScript,ReactJs.
- BACKEND : Nodejs ,MongoDB.
- ENVIRONMENT : Visual Studio Code.

MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License which is deemed non-free by several Distributions.

Nodejs

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine and executes JavaScript code outside a web browser, which was designed to build scalable network applications.

HTML

HTML, or Hypertext Markup Language, is a markup language for the web that defines the structure of web pages.

Hypertext: text (often with embeds such as images, too) that is organized in order to connect related items
Markup: a style guide for typesetting anything to be printed in hardcopy or soft copy format
Language: a language that a computer system understands and uses to interpret commands.

CSS

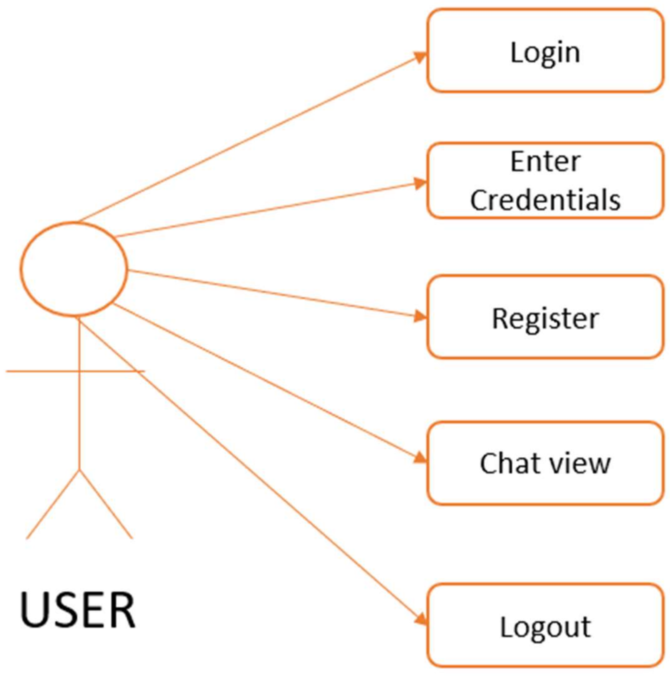
Cascading Style Sheets (**CSS**) is a Style sheet Language used for describing the Presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathMl or Xhtml). CSS is a cornerstone technology of the World wide web, alongside HTML and Javascript(js).

Javascript

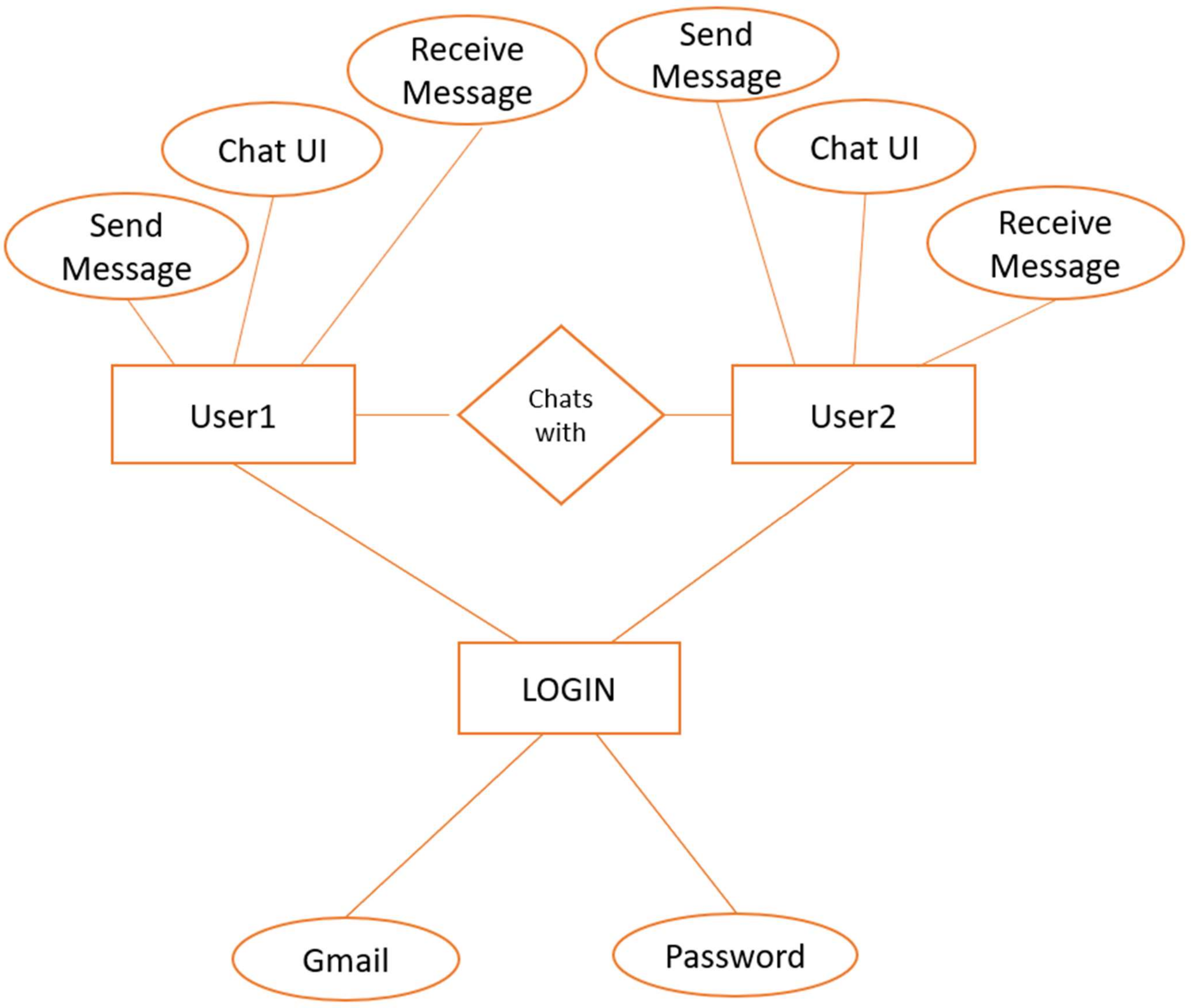
JavaScript is **a dynamic programming language that's used for web development, in web applications, for game development, and lots more**. It allows you to implement dynamic features on web pages that cannot be done with only HTML and CSS.

Use Case Diagram:

User:



ER Diagram:

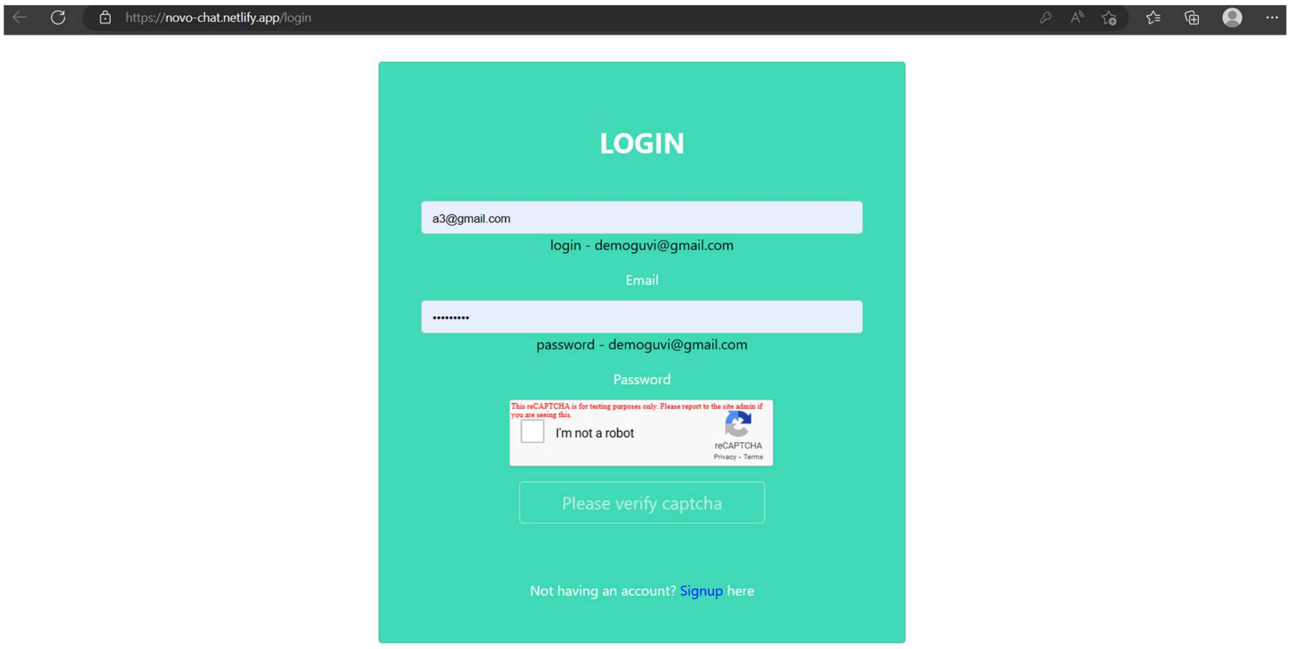

Receive Message
Send Message
Chat UI
Chat UI
Send Message
Receive Message
User1
Chats with
User2
LOGIN
Gmail
Password

SOURCE CODE

Home/Login.js:

```
import React, { useContext, useState } from "react";
import { Link } from 'react-router-dom'
import { useLoginUserMutation } from "../services/appApi";
import { useNavigate } from "react-router-dom";
import "./Login.css";
import { AppContext } from "../context/appContext";
import { Spinner } from "react-bootstrap";
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
import {  Bounce } from 'react-toastify';
import ReCAPTCHA from "react-google-recaptcha";


// Site key: 6LeIxAcTAAAAAJcZVRqyHh71UMIEGNQ_MXjiZKhI
// Secret key: 6LeIxAcTAAAAAGG-vFI1TnRWxMZNFuojJ4WifJWe

const Login = () => {
  const [verified,setVerified] = useState(false)
    const [email, setEmail] = useState("");
    const [password, setPassword] = useState("");
    const navigate = useNavigate();
    const { socket } = useContext(AppContext);
    function onChange(value) {
      console.log("Captcha value:", value);
      setVerified(true)
    }

    const [loginUser, { isLoading, error }] = useLoginUserMutation();
    const handleLogin = (e) => {
        e.preventDefault();
        loginUser({email, password }).then(({ data }) => {
            if (data) {
                socket.emit("new-user");
                navigate("/");
            }
        });
    }
    if(error){
      toast.warn('Invalid email or password😲', {
        position: "top-center",
        autoClose: 5000,
        hideProgressBar: false,
        closeOnClick: true,
        pauseOnHover: true,
        draggable: true,
        progress: true,
        });

    }

  return (
    <>
          <div style={{marginTop:'30px'}} >
          <div className="row d-flex justify-content-center align-items-center h-100">
            <div className="col-12 col-md-8 col-lg-6 col-xl-5">
              <div className="card text-white  custom-card"
style={{backgroundColor:"rgb(64, 218, 182"}}>
                <div className="card-body p-5 text-center">
                  <div className="mb-md-3 mt-md-4 pb-5">
                    <h2 className="fw-bold mb-5 text-uppercase">Login</h2>
                    <div className="mb-1">
                        <input type="email" id="email" name="email" onChange={(e) =>
setEmail(e.target.value)} value={email}  className="form-control " />
                        <p style={{color:"black"}}>login - demoguvi@gmail.com</p>
                        <label className="form-label" >Email</label>
                      </div>
                    <div className="mb-1">
                      <input type="password" id="password"
name="password"  onChange={(e) => setPassword(e.target.value)} value={password}
className="form-control " />
                      <p style={{color:"black"}}>password - demoguvi@gmail.com</p>

                      <label className="form-label">Password</label>
                    </div>
                    <div className="mb-3 d-flex justify-content-center" >
                    <ReCAPTCHA
                      sitekey="6LeIxAcTAAAAAJcZVRqyHh71UMIEGNQ_MXjiZKhI"
                      onChange={onChange}
                    />
                    </div>
                    <button disabled={!verified} className="btn btn-outline-light btn-lg
px-5"  id="btn-submit"  onClick={handleLogin}>
                    {!verified ? "Please verify captcha":"Login"}
                    {/* {isLoading && !verified ? <Spinner animation="grow" /> : "Please
fill the captcha to  Login"} */}
                    </button>
                  </div>
                  <div>
                    <p className="mb-0">Not having an account? <Link
to='/signup'  style={{textDecoration:'none',color:"blue"}} >Signup </Link> here
                    </p>
                  </div>

                </div>
              </div>
            </div>
          </div>
        </div>
        <ToastContainer
    transition={Bounce}
  />
    </>
  )
}

export default Login;
```

OUTPUT:



https://novo-chat.netlify.app/login

LOGIN

a3@gmail.com

login - demoguvi@gmail.com

Email

•••••••••

password - demoguvi@gmail.com

Password

This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this.

I'm not a robot

reCAPTCHA
Privacy - Terms

Please verify captcha

Not having an account? Signup here

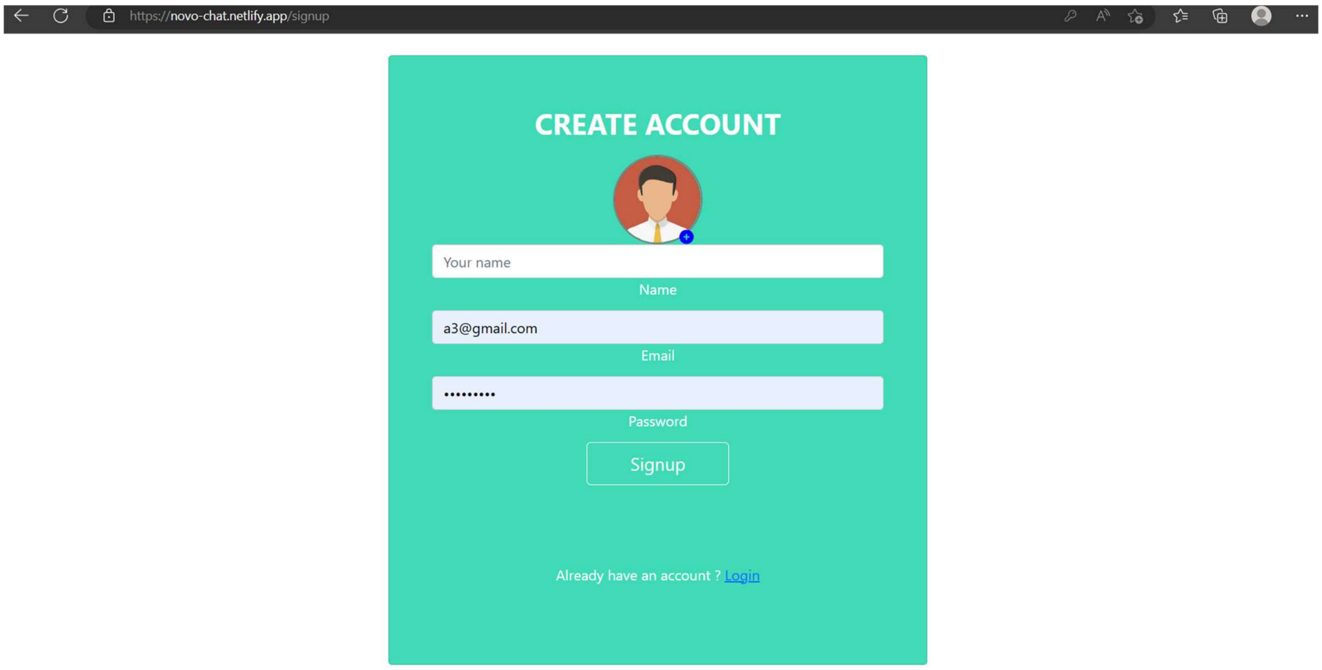
Register.js:

```
import React, { useState } from "react";
import { useSignupUserMutation } from "../services/appApi";
import { Link, useNavigate } from "react-router-dom";
import "./Signup.css";
import profile from "../assets/profile.png";
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
import {  Bounce } from 'react-toastify';
import { Spinner } from "react-bootstrap";
const Signup = () => {
    const [email, setEmail] = useState("");
    const [password, setPassword] = useState("");
    const [name, setName] = useState("");
    const [signupUser, { isLoading, error }] = useSignupUserMutation();
    const navigate = useNavigate();
    //image upload states
    const [image, setImage] = useState(null);
    const [upladingImg, setUploadingImg] = useState(false);
    const [imagePreview, setImagePreview] = useState(null);

    function validateImg(e) {
        const file = e.target.files[0];
        if (file.size >= 1048576) {
            return toast.warn('Please upload less than 1Mb 😣😣😣!', {
              position: "top-center",
              autoClose: 5000,
              hideProgressBar: false,
              closeOnClick: true,
              pauseOnHover: true,
              draggable: true,
              progress: true,
              });

        } else {
            setImage(file);
            setImagePreview(URL.createObjectURL(file));
        }
    }

    async function uploadImage() {
        const data = new FormData();
        data.append("file", image);
        data.append("upload_preset", "unme-chatapp");
        try {
            setUploadingImg(true);
            let res = await
    fetch("https://api.cloudinary.com/v1_1/dvuqiruzf/image/upload", {
                method: "post",
                body: data,
            });
            const urlData = await res.json();
            setUploadingImg(false);
            return urlData.url;
        } catch (error) {
            setUploadingImg(false);
            toast.error('Something went wrong😭', {
              position: "top-center",
              autoClose: 5000,
              hideProgressBar: false,
              closeOnClick: true,
              pauseOnHover: true,
              draggable: true,
              progress: true,
              });

        }
    }
     const  handleSignup =async(e)=> {
        e.preventDefault();
        if (!image) return toast.warn('Please upload profile picture 😅!', {
          position: "top-center",
          autoClose: 5000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: true,
          progress: true,
          });

        else if( !name) return toast.warn('Please enter your name 😅!', {
          position: "top-center",
          autoClose: 5000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: true,
          progress: true,
          });
        else if( !email) return toast.warn('Please Enter your email 😅!', {
          position: "top-center",
          autoClose: 5000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: true,
          progress: true,
          });

        else if( !password) return toast.warn('Oops! You forgot to fill password 😅!', {
          position: "top-center",
          autoClose: 5000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: true,
          progress: true,
          });

        const url = await uploadImage(image);
        signupUser({ name, email, password, picture: url }).then(({ data }) => {
            if (data) {
                navigate("/login");
            }
        });
    }

    if(error){
      toast.error('Already used this email 😠', {
        position: "top-center",
        autoClose: 5000,
        hideProgressBar: false,
        closeOnClick: true,
        pauseOnHover: true,
        draggable: true,
        progress: true,
        });

    }
    return (
      <>

        <div className="mt-4">
          <div className="row d-flex justify-content-center align-items-center h-100">
            <div className="col-12 col-md-8 col-lg-6 col-xl-5">
              <div className="card text-white  custom-card"
  style={{backgroundColor:"rgb(64, 218, 182"}}>
                <div className="card-body p-5 text-center">
```

```
                  <div className="mb-md-3 mt-md-2 pb-5">
                    <h2 className="fw-bold mb-3 text-uppercase">Create Account</h2>
                    <div className="signup-profile-pic__container">
                            <label htmlFor="image-upload" className="image-upload-
label">
                            <img src={imagePreview || profile} className="signup-
profile-pic" />
                                <i className="fas fa-plus-circle add-picture-icon"></i>
                            </label>
                            <input type="file"  id="image-upload" hidden
accept="image/png, image/jpeg" onChange={validateImg} />
                        </div>
                    <div className="mb-1">
                        <input type="email"  placeholder="Your name" onChange={(e) =>
setName(e.target.value)} value={name}  className="form-control " />
                        <label className="form-label" >Name</label>
                      </div>
                    <div className="mb-1">
                        <input type="email" id="email" name="email" onChange={(e) =>
setEmail(e.target.value)} value={email}  className="form-control " />
                        <label className="form-label" >Email</label>
                      </div>
                    <div className="mb-1">
                      <input type="password" id="password"
name="password"  onChange={(e) => setPassword(e.target.value)} value={password}
className="form-control " />
                      <label className="form-label">Password</label>
                    </div>
                    <button className="btn btn-outline-light btn-lg px-5"  id="btn-
submit"  onClick={handleSignup}>
                    {upladingImg || isLoading ? <Spinner animation="grow" /> : "Signup"}
                    </button>
                  </div>
                  <div className="py-4">
                            <p className="text-center">
                                Already have an account ? <Link to="/login">Login</Link>
                            </p>
                        </div>

                </div>
              </div>
            </div>
          </div>
        </div>
        <ToastContainer
      transition={Bounce}
    />
```

```
        </>
    );
}

export default Signup;
```

OUTPUT:



https://novo-chat.netlify.app/signup

CREATE ACCOUNT

Your name

Name

a3@gmail.com

Email

•••••••••

Password

Signup

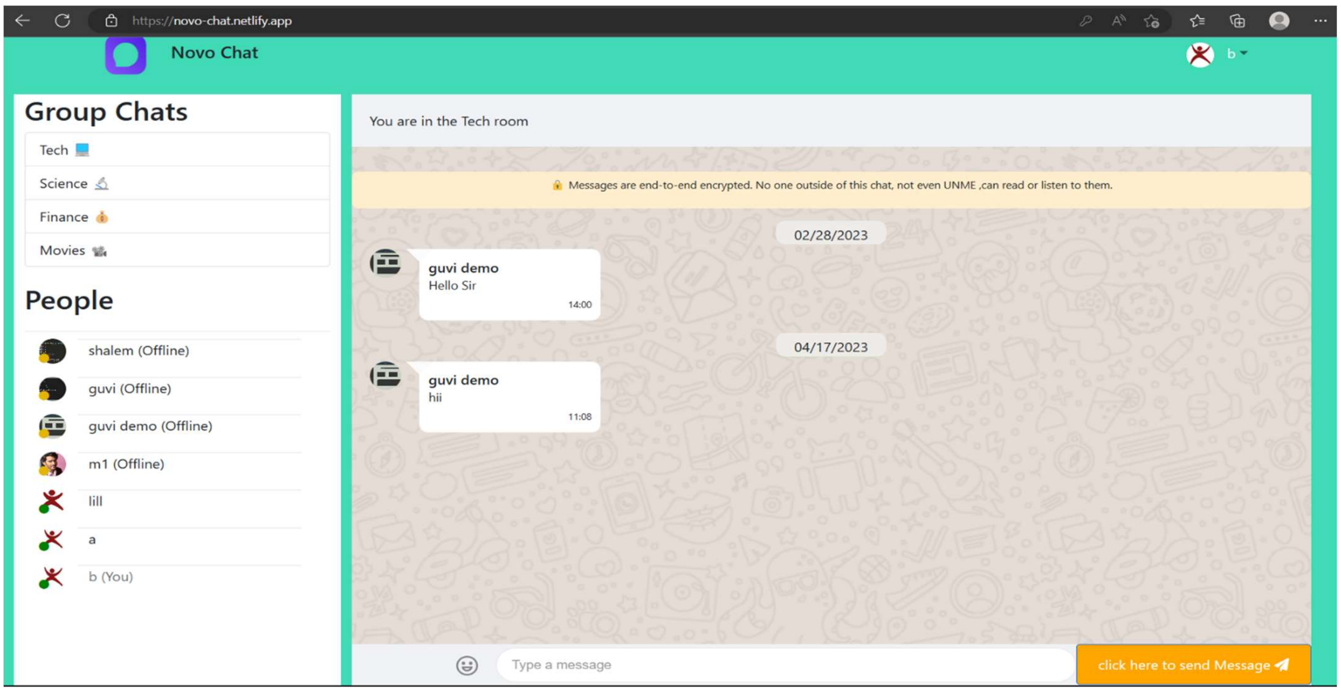
Already have an account ? Login

Chat.js:

```
import React, { useEffect } from "react";
import { Container, Row, Col } from "react-bootstrap";
import Sidebar from "../components/Sidebar";
import MessageForm from "../components/MessageForm";
import { useNavigate } from "react-router-dom";
import { useSelector } from "react-redux";
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
import {  Bounce } from 'react-toastify';
const Chat =()=> {
    const user = useSelector((state) => state.user);
    const navigate= useNavigate()
useEffect(()=>{
    if(!user){
        navigate('/login')
    }

})
```

```
    return (
        <>
        <div style={{ backgroundColor:"rgb(64, 218, 182)",}}>
                <div style={{  marginLeft:"30px", marginRight:"30px",  }}   >
                <Row >
                    <Col md={3}
                        style={{backgroundColor:"white"}}
                    >
                        <Sidebar />
                    </Col>
                    <Col md={9} >
                        <MessageForm />
                    </Col>
                </Row>

            </div>
        </div>
        <ToastContainer
    transition={Bounce}
  />
        </>
    );
}
export default Chat;
```

OUTPUT:



https://novo-chat.netlify.app

Novo Chat

b

Group Chats

- Tech
- Science
- Finance
- Movies

People

- shalem (Offline)
- guvi (Offline)
- guvi demo (Offline)
- m1 (Offline)
- lill
- a
- b (You)

You are in the Tech room

Messages are end-to-end encrypted. No one outside of this chat, not even UNME ,can read or listen to them.

02/28/2023

guvi demo
Hello Sir
14:00

04/17/2023

guvi demo
hii
11:08

Type a message

click here to send Message

App.js

```
import "./App.css";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Navigation from "./components/Navigation";
import Login from "./pages/Login";
import Signup from "./pages/Signup";
import Chat from "./pages/Chat";
import { useSelector } from "react-redux";
import { useState } from "react";
import { AppContext, socket } from "./context/appContext";

function App() {
    const [rooms, setRooms] = useState([]);
    const [currentRoom, setCurrentRoom] = useState([]);
    const [members, setMembers] = useState([]);
    const [messages, setMessages] = useState([]);
    const [privateMemberMsg, setPrivateMemberMsg] = useState({});
    const [newMessages, setNewMessages] = useState({});
    const user = useSelector((state) => state.user);



    return (
        <AppContext.Provider value={{ socket, currentRoom, setCurrentRoom, members,
setMembers, messages, setMessages, privateMemberMsg, setPrivateMemberMsg, rooms,
setRooms, newMessages, setNewMessages }}>
            <BrowserRouter>
                <Navigation />
                <Routes>
                    <Route path="/" element={<Chat />} />
                    <Route path="/login" element={<Login />} />
                    <Route path="/signup" element={<Signup />} />
                </Routes>
            </BrowserRouter>
        </AppContext.Provider>
    );
}

export default App;
```

Implementation and System Testing

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

System Testing:

The goal of the system testing process was to determine all faults in our project. The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing.

1.Unit testing

2.Integration testing

Unit Testing:

Unit testing is commenced when a unit has been created and effectively reviewed. In order to test a single module we need to provide a complete environment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses .A procedure to call the functions of the unit under test with appropriate parameters.

Integration Testing:

In the Integration testing we test various combination of the project module by providing the input. The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

Conclusion

The main objective of this application is to develop a Secure chat application.There would be more chatting experiences enjoyed by the users.A first positive impression is essential in human relationship as well as human in computer interaction.This project hopes to develop a chat service web app with quality user interface.

Future Enhancement

- It would extended to include features such as File Transfer,voice Message.
- Making possibe to include Video call,Audio call,Group call.
- Increasing the effectiveness of the application by providing Voice chat.
- Providing more privacy.

References

- https://www.w3schools.com
- https://reactjs.org/
- https://www.javatpoint.com/nodejs
- https://www.geeksforgeeks.org/