MES COLLEGE OF ENGINEERING, KUTTIPURAM

DEPARTMENT OF COMPUTER SCIENCE

CSL-333

DATABASE MANAGEMENT SYSTEM LAB

BLOGGING APP

TEAM MEMBERS:

AHMED LASIN(MES21CS009)

ALIF FIAZ MUHAMMED(MES21CS011)

AKIF KIZHISSERI SALIM(MES21CS010)

AMAN ROSHAN(MES21CS012)

**ABSTRACT**

Blogging app allows users to write and publish a blog.

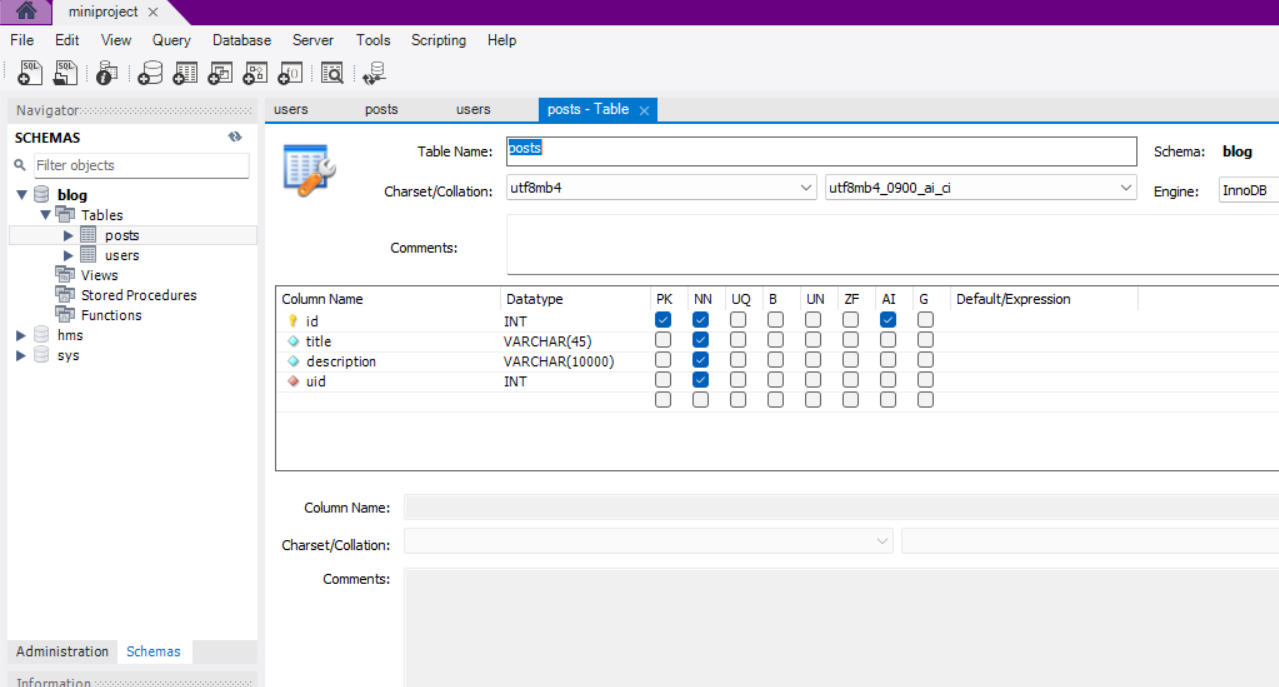
The web application allows user to create a account in the application and later log in with the registered data. After logging in the user can write, edit or delete his blogs.

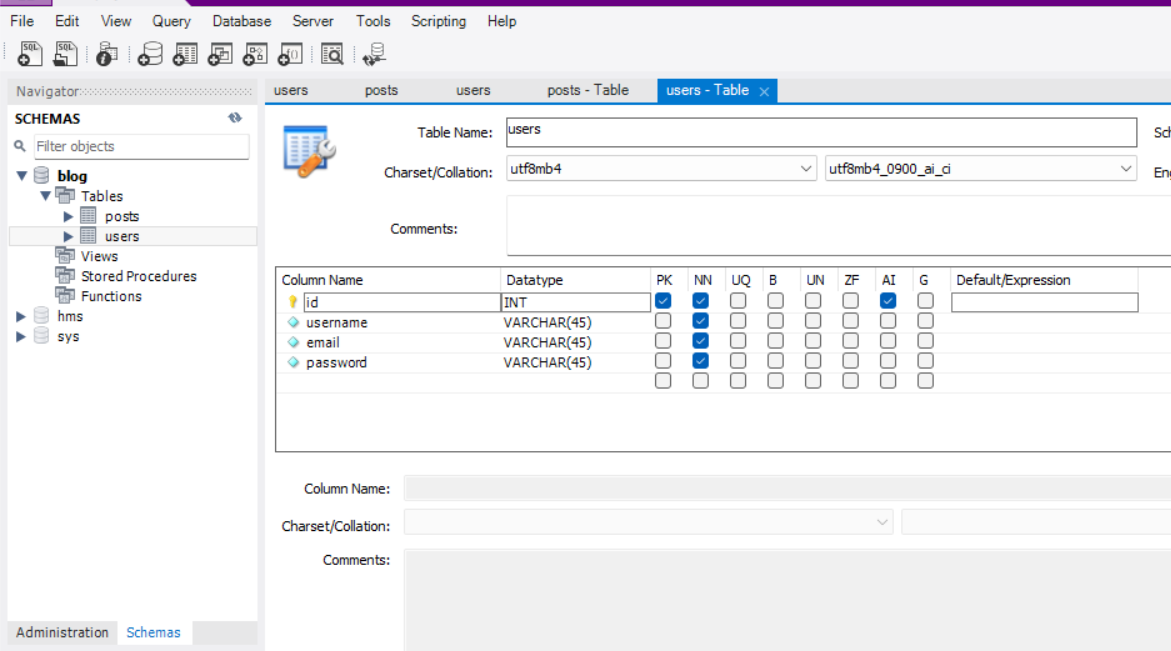
The registration process includes validation of user input and login process uses session variables to track the users session and redirect to their page. The page contain their data that they have published.

**IMPLEMENTATION**

**DATABASE CREATION**

Open SQL workbench and create a table for users with id as primary key ,username , email and password. Then create a table for posts with id as primary key, title ,description and uid which is an foreign key which references the id in the user database.





**CONNECT TO THE DATABASE**

After creating the table, we have to create a mysql connector script to connect the MYSQL database server.Create file named db.js and put the following code in it.

**db.js**

import mysql from "mysql"

export const db = mysql.createConnection({

    host:"localhost",

    port:"3306",

    user:"root",

    password:"alif",

    database:"blog"

})

db.connect((err) => {

    if (err) {

        console.error('Error connecting to MySQL:', err);

        return;

    }

    console.log('Connected to MySQL database');

});

**USER AUTHENTICATION(REGISTRATION AND LOG IN)**

After connecting to database the user data from the front-end has to be validated using the authentication functionalities register and login. Create auth.js file and put the following code in it.

import { db } from "../db.js";

export const register = (req, res) => {

    // CHECK EXISTING USER

    const q = "SELECT \* FROM users WHERE email = ? OR username = ?";

    db.query(q, [req.body.email, req.body.username], (err, data) => {

      if (err) return res.status(500).json(err);

      if (data.length) return res.status(409).json({msg:"User already exists!",status:false});

      const insertQuery = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";

      db.query(insertQuery, [req.body.username, req.body.email,req.body.password], (err, data) => {

        if (err) return res.status(500).json(err);

        return res.status(200).json({msg:"User has been created.",status:true});

      });

    });

  };

export const login = (req,res)=>{

    console.log(req.body.username)

    const q = "select \* from users where username = ? and password = ?";

    db.query(q,[req.body.username, req.body.password],(err,data)=>{

        if(err) return res.status(500).json(err)

        if(data.length === 0){ return res.status(404).json({status:false})

        }else{

            return res.json({status:true,user:data[0]})

        }

    })

}

**USER BLOG MANAGEMENT**

The blog data has to be verified using the addpost ,get post and showpost functionalities. Create post.js file and paste the following code.

import { db } from "../db.js";

export const addpost = (req,res)=>{

    const insertQuery = "INSERT INTO posts (title, description,uid) VALUES (?, ?, ?)";

    db.query(insertQuery, [req.body.title, req.body.description,req.body.uid], (err, data) => {

        if (err) return res.status(500).json(err);

        return res.status(200).json({msg:"post added",status:true});

      });

}

export const getpost = (req, res) => {

    const getquery = "select \* from posts where uid =?";

    db.query(getquery, [req.query.uid], (err, data) => {

      if (err) return res.status(500).json(err);

      return res.status(200).json({ msg: "got data", status: true, dat: data });

    });

  };

export const showpost=(req,res)=>{

    const getquery = "select \* from posts where id =?"

    db.query(getquery, [req.query.id], (err, data) => {

        if (err) return res.status(500).json(err);

        return res.status(200).json({ msg: "got data", status: true, dat: data });

      });

}

export const deletepost=(req,res)=>{

    const deletequery = "delete from posts where id =?"

    db.query(deletequery, [req.body.id], (err, data) => {

        if (err) return res.status(500).json(err);

        return res.status(200).json({ msg: "data deleted", status: true, dat: data });

      });}

**CREATING REGISTRATION FORM**

Create a JS file signup.js and paste the following code.This will create a registration form.

import React, { useState } from 'react'

import axios from "axios"

import {useNavigate} from 'react-router-dom'

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

function Signup() {

  const vallidationFormat = {

    position: "top-right",

    autoClose: 8000,

    hideProgressBar: false,

    closeOnClick: true,

    pauseOnHover: true,

    draggable: true,

    progress: undefined,

    theme: "colored",

    }

  const navigate = useNavigate();

  const [input,setInput] = useState({

    username:'',

    email:'',

    password:''

  });

  const handleSubmit = async(e) => {

    e.preventDefault();

    if(input.password===''||input.submit===''||input.email===''){

      toast.error("Field empty", vallidationFormat );

    }else{

    try {

      const res = await axios.post("http://localhost:8800/api/auth/register", input);

      console.log(res);

      if(res.status){

        navigate('/login')

      }

    } catch (error) {

      toast.error("user already exist", vallidationFormat );

      console.error('Axios Error alif:', error);

    }}

  };

  const handlechange =(e)=>{

    setInput({ ...input, [e.target.name]: e.target.value });

    console.log(input)

  }

  return (

    <>

    <div>

      <div className='login'>

        <div className='innerbox'>

        <h1>USER SIGN UP</h1>

        <form className='form' >

            <input className='formin' name="username"

                type='text'

                placeholder='username' onChange={handlechange} autoComplete='off'/ >

                <input className='formin' name="email"

                type='email'

                placeholder='email' onChange={handlechange} autoComplete='off'/ >

            <input className='formin' type='password'

                name='password'

                placeholder='password' onChange={handlechange} autoComplete='off'/>

                <p>already <a href='/login'>registered</a>?</p>

            <button className='lbutton' onClick={handleSubmit}>submit</button>

        </form>

        </div>

    </div>

    </div>

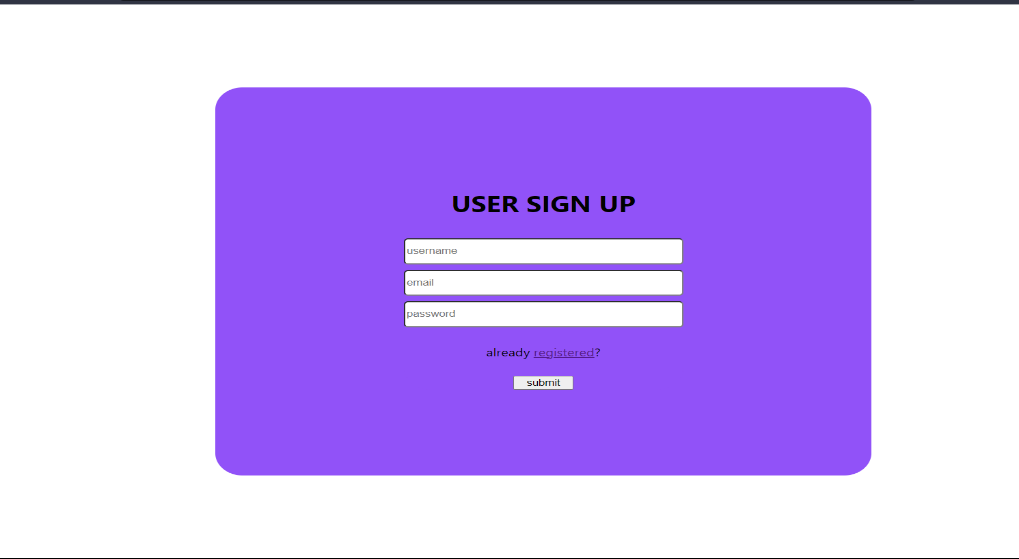
    <ToastContainer/>

    </>

  )

}

export default Signup



**CREATING LOGIN FORM**

Create a JS file login.js and paste the following code.This will create a login form.

import React,{ useState,useReducer } from 'react'

import {useNavigate} from 'react-router-dom'

import axios from 'axios';

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

function Login() {

  const vallidationFormat = {

    position: "top-right",

    autoClose: 8000,

    hideProgressBar: false,

    closeOnClick: true,

    pauseOnHover: true,

    draggable: true,

    progress: undefined,

    theme: "colored",

    }

  const navigate = useNavigate();

  const [input,setInput] = useState({

    username:'',

    password:''

  });

  const handleSubmit = async(e) => {

    e.preventDefault();

    try {

      const res = await axios.post("http://localhost:8800/api/auth/login", input);

      if(res.data.status){

        localStorage.setItem('blog-app',JSON.stringify(res.data.user));

        navigate('/')

        console.log(res.data.user.id)

      }

    } catch (error) {

      toast.error("invalid login", vallidationFormat );

      console.log(error)

    }

  };

  const handlechange =(e)=>{

    setInput({ ...input, [e.target.name]: e.target.value });

    console.log(input)

  }

  return (

<>

    <div className='login'>

        <div className='innerbox'>

        <h1>USER LOG IN</h1>

        <form className='form' >

            <input className='formin' name="username"

                type='text'

                placeholder='username' onChange={handlechange} autoComplete='off'/ >

            <input className='formin' type='password'

                name='password'

                placeholder='password' onChange={handlechange} autoComplete='off'/>

                <p>not yet <a href='/signup'>registred</a>?</p>

                <button className='lbutton' onClick={handleSubmit}>submit</button>

        </form>

        </div>

    </div>

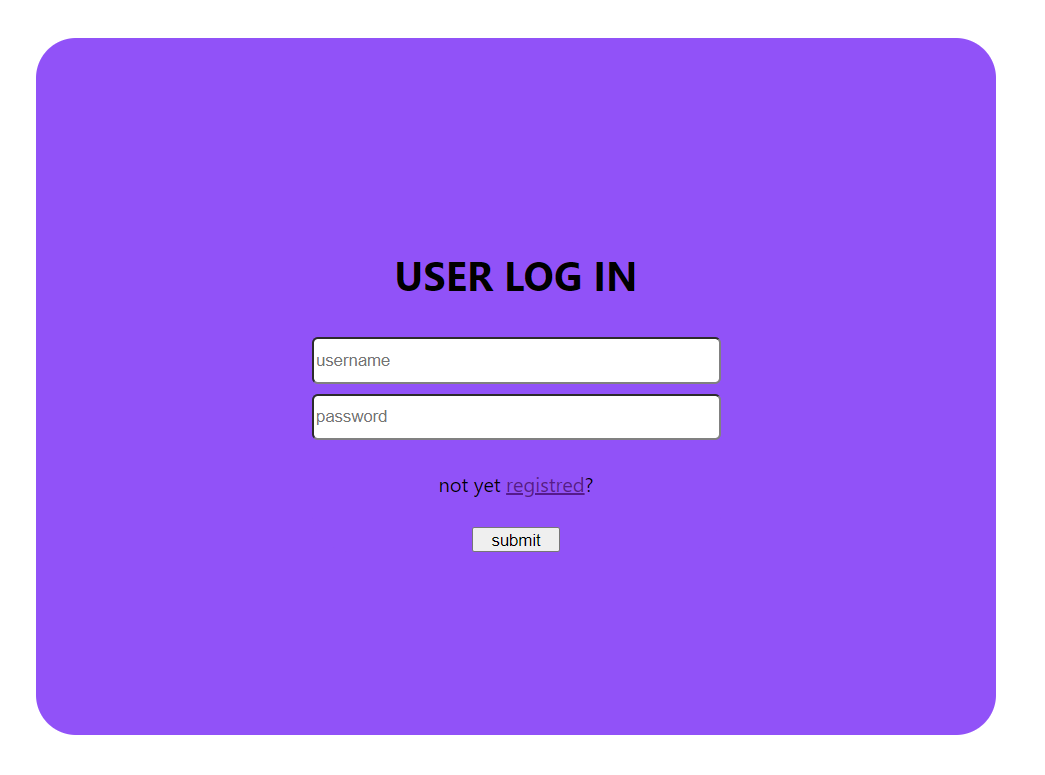
    <ToastContainer/>

    </>

  )

}

export default Login

****

**CREATING HOME PAGE**

Create a JS file home.js and paste the following code.This will create a home page.

import React, { useEffect,useState } from 'react'

import Navbar from '../components/navbar'

import HomeContent from '../components/HomeContent'

import {useNavigate} from 'react-router-dom'

function Home() {

  const [user,setUser] = useState({});

  const navigate = useNavigate();

  useEffect(()=>{

    if(!localStorage.getItem('blog-app')){

      navigate('/login');

    }

  },[])

  useEffect(()=>{

    const getuserdata = async()=>{

      try {

        const users=await JSON.parse(localStorage.getItem('blog-app'));

        setUser(users)

      } catch (error) {

        console.log(error)

      }

    }

    getuserdata();

  },[])

  return (

    <div className='home'>

      <Navbar username={user.username}/>

      <HomeContent userid={user.id}/>

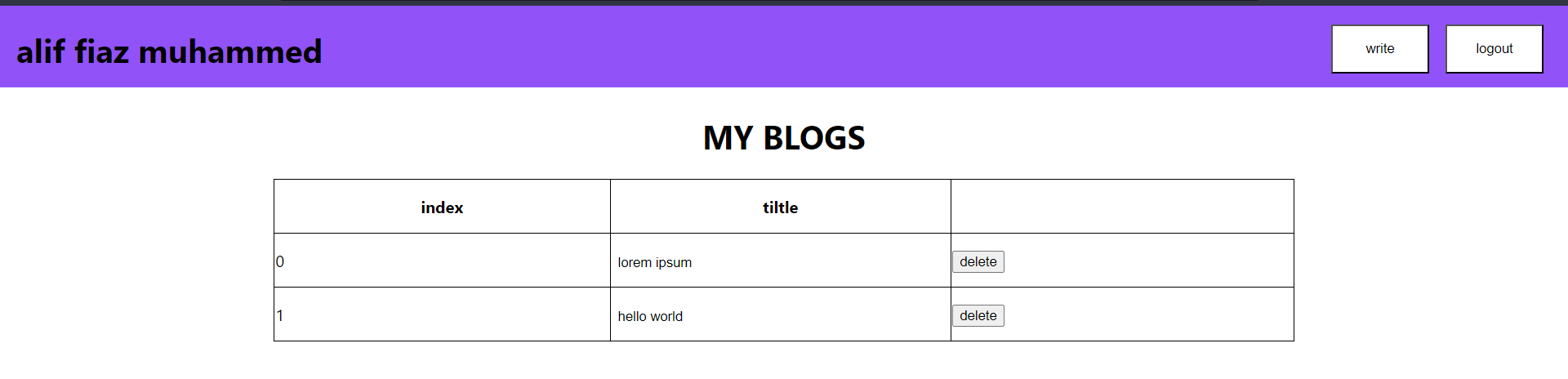
      <h1></h1>

    </div>

  )

}

export default Home

****

**CREATING WRITE PAGE**

Create a JS file write.js and paste the following code.This will create a write page.

import React,{useState,useEffect} from 'react';

import axios from 'axios';

import {useNavigate} from 'react-router-dom';

function Write() {

  const navigate = useNavigate();

    const [input,setInput] = useState({

        title:'',

        description:'',

      });

    const [user,setUser] = useState({});

    useEffect(()=>{

        const getuserdata = async()=>{

          try {

            const users=await JSON.parse(localStorage.getItem('blog-app'));

            setUser(users)

          } catch (error) {

            console.log(error)

          }

        }

        getuserdata();

      },[])

      useEffect(() => {

        // Update the input state with the correct uid when user state changes

        setInput((prevInput) => ({

          ...prevInput,

          uid: user.id,

        }));

      }, [user]);

      const handleSubmit = async(e) => {

        e.preventDefault();

        try {

          console.log(input)

          const res = await axios.post("http://localhost:8800/api/posts/write-post", input);

          console.log(res)

          if(res.data.status){

            navigate('/')

          }

        } catch (error) {

          console.log(error) }

      };

    const handlechange =(e)=>{

    setInput({ ...input, [e.target.name]: e.target.value });

    console.log(input)

  }

  return (

    <div className='write'>

      <div className='innerbox'>

        <h1>WRITE BLOG</h1>

        <form className='form' >

            <input className='formin' name="title"

                type='text'

                placeholder='title' onChange={handlechange}/ >

            <textarea className='formin' type='textarea'

                name='description' maxLength='10000'

                placeholder='description' onChange={handlechange}/>

                <button className='lbutton' onClick={handleSubmit}>submit</button>

        </form>

        </div>

    </div>

  )

}

export default Write

****

**CREATING BLOG PAGE**

Create a JS file blog.js and paste the following code.This will create a blog page.

import React,{useEffect,useState} from 'react'

import { useParams } from 'react-router-dom'

import axios from 'axios';

function Post() {

    const { id } = useParams();

    const [data,setdata] = useState([])

    useEffect(() => {

        const getuserdata = async () => {

          try {

            const idata = await axios.get("http://localhost:8800/api/posts/show-post", {

              params: {id},

            });

            setdata(idata.data.dat);

          } catch (error) {

            console.log(error);

          }

        };

        getuserdata();

      }, [id]);

      useEffect(() => {

        console.log(data); // Ensure this is placed correctly

      }, [data]);

    return (

      <div>

        {

                data.map((d,index)=>{

                    return(

                        <div className='post' key={d.uid}>

                            <div className='title'>

                            <h1>{d.title}</h1>

                            </div>

                            <div className='description'>

                            <p>{d.description}</p>

                            </div>

                        </div>

                    )

                })

            }

      </div>

    );

}

export default Post

**CSS FILE CREATE**

Finally, important step for user experience perspective. Create CSS file style.css and put the below code.

body {

  margin: 0;

  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

    sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

  overflow: hidden;

/\* login \*/

  .login{

    text-align: center;

  display: flex;

  justify-content: center;

  align-items: center;

  background-color: rgba(255,255,255,255);

  height: 100vh;

  width: 100vw;

  cursor: pointer;

  }

  .innerbox{

    display: flex;

    justify-content: center;

    align-items: center;

    flex-direction:column;

    height: 70vh;

    width: 50vw;

    gap: 0.5rem;

    background: #9152f8;

    border-radius: 2rem;

  }

  .form{

    display: flex;

    justify-content: center;

    align-items: center;

    flex-direction:column;

    gap: 0.5rem;

    background: #9152f8;

    border-radius: 2rem;

  }

  .formin{

    width: 20rem;

    height: 2rem;

    border-radius: 0.3rem;

    border-color: gray;

  }

  .lbutton{

    width: 70px;

    height: 20px;

    cursor: pointer;

  }

/\* homecontent \*/

.homecontent{

  padding-top: 5px;

  display: flex;

  flex-direction: column;

  .heading{

    display: flex;

    justify-content: center;

  }

  .table{

    display: flex;

    justify-content: center;

    .tab, th, td {

      width: 1000px;

      height: 50px;

      border: 1px solid black;

      border-collapse: collapse;

    }

  }

}

/\* post \*/

.post{

  width: 100vw;

  height: 100vh;

  display: grid;

  grid-template-rows: 10% 90%;

  justify-items: center;

  gap: 20px;

}

.title{

  display: flex;

  justify-content: center;

  align-items: center;

  font-size: 40px;

}

.description{

  width: 60vw;

  background-color: antiquewhite;

  font-size: 20px;

}

/\* write \*/

.write{

  height: 100vh;

  width: 100vw;

  display: flex;

  justify-content: center;

  align-items: center;

}

/\* home \*/

.home{

  display: grid;

  height: 100vh;

  width: 100vw;

  grid-template-rows: 10% 90%;

  .hbutton{

    opacity:1;

    border:none;

    cursor: pointer;

    background-color: #ffff;

  }

  .navbar{

    background-color: #9152f8;

    width: 100vw;

    height: 100%;

    display: grid;

    grid-template-columns: 50% 50%;

    .fbox{

      display: flex;

      align-items: center;

      padding-left: 1rem;

    }

    .sbox{

      display: flex;

      align-items: center;

      flex-direction: row;

      justify-content: flex-end;

      padding: 0.5rem;

     .login{

      display: flex;

      align-items: center;

      height: 3rem;

      width: 6rem;

      margin-right: 1rem;

     }

    }

  }

}

.h2button{

  cursor: pointer;

}

}

code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

    monospace;

}

**CONCLUSION**

A BLOG application was created using react ,express ,MySQL Workbench and Node as part of this project .The system allows user to register ,login and write and publish his blogs in the application. The application stores the personal details like name and email along with password in a Mysql database. The application also stores user posts.

During the development of the application, Node was used to handle server-side processing and MYSQL was used to store the user data. The system was tested using a variety of user inputs to ensure its functionality and security.

Overall ,the BLOG APPLICATION using react ,express ,MySQL Workbench and Node was successful in meeting its objectives. It provides a secure and convenient way for users to register,login and publish a blog using the application.