**PRACTICE REPORT**

**Subject: Web application**

**Lab 6: Build fake facebook**

*Instructor: Trần Vĩnh Khiêm*

*Report date: 4/1/2024*

1. **GENERAL INFORMATION:**

Class: MSIS207.O12.CTTT.1

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Student ID** | **Email** |
| 1 | Nguyễn Văn Trường Khoa | 20521472 | [20521472@gm.uit.edu.vn](mailto:20521472@gm.uit.edu.vn) |
| 2 |  |  |  |

## DETAILED REPORT

**Link Github contain code:**

<https://github.com/KRocker3512/Web-app-practice-class>

**Note** : If you can’t access link above, please contact to me via email :20521472@gm.uit.edu.vn .

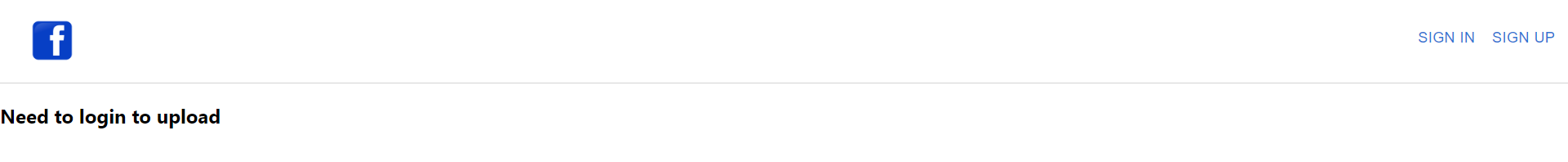
Demo :

A screenshot of a login form

Description automatically generated

A screenshot of a login box

Description automatically generated



A screen shot of a comic book

Description automatically generated

Code :

Front end:

Component

Post.js

|  |
| --- |
| import React from 'react'  import './Post.css'  import {Avatar} from '@mui/material'  const Post = ({username, caption, imageUrl}) => {      return(          <div className='post'>              <div className='post\_header'>                  <Avatar                      className='post\_avatar'                      alt={username}                      src=''                  />                  <h3>{username}</h3>              </div>              <img className='post\_image' src={imageUrl}/>              <h4 className='post\_text'><strong>{username}</strong>{caption}</h4>          </div>      )  }  export default Post |

Post.css

|  |
| --- |
| .post{      background-color: white;      max-width: 1000px;      border: 1px solid;      margin-bottom: 45px;  }  .post\_image{      width: 500px;      height: 500px;      object-fit: contain;      border-top: 1px solid lightgray;      border-bottom: 1px solid lightgray;  }  .post\_text{      font-weight: normal;      padding: 20px;  }  .post\_header{      display: flex;      align-items: center;      padding: 20px;  }  .post\_avatar{      margin-right: 10px;  } |

ImageUpload.js

|  |
| --- |
| import React, {useState} from 'react'  import './ImageUpload.css'  import {storage} from '../firebase';  import axios from 'axios'  const ImageUpload = ({username}) => {      const [image, setImage] = useState(null)      const [progress, setProgress] = useState(0)      const [caption, setCaption] = useState('')      const [url, setUrl] = useState('');      const handleChange = e => {          if(e.target.files[0]) {              setImage(e.target.files[0])          }      }      const handleUpload = () => {          const uploadTask = storage.ref("images/${image.name}").put(image);          uploadTask.on(              'state\_changed',              (snapshot) => {                  const progress = Math.round(                      (snapshot.bytesTransferred/snapshot.totalBytes) \* 100                  );                  setProgress(progress);              },              (error) => {                  console.log(error);              },              () => {                  storage                      .ref('images')                      .child(image.name)                      .getDownloadUrl()                      .then((url) => {                          setUrl(url);                          axios.post('http://localhost:9000/upload', {                              caption: caption,                              user: username,                              image: url                          })                          setProgress(0);                          setCaption('');                          setImage(null);                      })              }          );      };      return(          <div className='imageUpload'>              <progress className='imageUpload\_progress' value={progress} max='100'>                  <input                      type='text'                      placeholder='Enter a caption...'                      className='imageUpload\_input'                      value={caption}                      onChange={e => setCaption(e.target.value)}                  />                  <input className='imageUpload\_file' type='file' onChange={handleChange}/>                  <button className='imageUpload\_button' onClick={handleUpload}>Upload</button>              </progress>          </div>      )  }  export default ImageUpload; |

ImageUpload.css

|  |
| --- |
| .imageUpload{      display: flex;      flex-direction: column;      max-width: 800px;      width: 100%;      margin: 10px auto;  }  .imageUpload\_progress{      width: 100px;      margin-bottom: 10px;  }  .imageUpload\_input{      padding: 10px;      margin-bottom: 10px;  }  .imageUpload\_file{      margin-bottom: 10px;  }  .imageUpload\_button{      border: none;      color: lightgray;      background-color: #6082a3;      cursor: pointer;      padding: 10px;      font-weight: bolder;      font-size: 0.9rem;  }  .imageUpload\_button:hover{      color: #6082a3;      background-color: lightgray;  } |

App.js

|  |
| --- |
| import './App.css';  import Post from './components/Post';  import React, {useState, useEffect} from 'react';  import {makeStyles} from '@mui/styles'  import Modal from '@mui/material/Modal';  import {Button, Input} from '@mui/material';  import {auth} from './firebase';  import ImageUpload from './components/ImageUpload';  import axios from 'axios';  import Pusher from 'pusher-js';  var pusher = new Pusher('6610f8c9f686b6a34312', {    cluster: 'ap1'  });  function getModalStyle() {    const top = 50;    const left = 50;    return{      top: `${top}%`,      left: `${left}%`,      transform: `translate(-${top}%, -${left}%)`    };  }  const useStyles = makeStyles(theme => ({    paper: {      position: 'absolute',      width: 400,      border: '2px solid #000',    }  }));  function App() {    const classes = useStyles();    const [modalStyle] = React.useState(getModalStyle);    const [open, setOpen] = useState(false);    const [username, setUsername] = useState('');    const [email, setEmail] = useState('');    const [password, setPassword] = useState('');    const [user, setUser] = useState('');    const [openSignIn, setOpenSignIn] = useState(false);    const [posts, setPosts] = useState([]);    console.log(posts)    useEffect(() => {      const unsubscribe = auth.onAuthStateChanged(authUser => {        if(authUser) {          setUser(authUser);        } else {          setUser(null);        }      })      return () => {        unsubscribe()      }    }, [user, username])    const fetchPosts = async () =>{      await axios.get('http://localhost:9000/sync').then(response => setPosts(response.data))    }    useEffect(() => {      const channel = pusher.subscribe('my-channel');      channel.bind('inserted', (data) => {        fetchPosts();      });    }, [])    useEffect(() => {      fetchPosts()    }, [])    const signUp = async(e) => {      e.preventDefault();      auth.createUserWithEmailAndPassword(email, password)          .then(authUser => authUser.user.updateProfile({displayName: username}))          .catch(error => alert(error.message))      setOpen(false)    }    const signIn = async(e) => {      e.preventDefault();      auth.signInWithEmailAndPassword(email, password)          .catch(error => alert(error.message))      setOpenSignIn(false)    }    return (      <div className="App">        <Modal open={open} onClose={() => setOpen(false)}>          <div style={modalStyle} className={classes.paper}>            <form className='app\_signup'>              <center>                <img className='app\_headerImage' src='' alt='Header'/>              </center>              <Input placeholder='username'                     type='text'                     value={username}                     onChange={e => setUsername(e.target.value)}              />              <Input placeholder='email'                     type='text'                     value={email}                     onChange={e => setEmail(e.target.value)}              />              <Input placeholder='password'                     type='password'                     value={password}                     onChange={e => setPassword(e.target.value)}              />              <Button type='submit' onClick={signUp}>Sign Up</Button>            </form>          </div>        </Modal>        <Modal open={openSignIn} onClose={() => setOpenSignIn(false)}>          <div style={modalStyle} className={classes.paper}>            <form className = "app\_signup">              <center>                <img className='app\_headerImage'                     src=''                     alt='Header'                />              </center>              <center>                <Input placeholder='email'                       type='text'                       value={email}                       onChange={e => setEmail(e.target.value)}                />                <Input placeholder='password'                       type='password'                       value={password}                       onChange={e => setPassword(e.target.value)}                />                <Button type='submit' onClick={signIn}>Sign In</Button>              </center>            </form>          </div>        </Modal>        <div className='app\_header'>          <img className='app\_headerImage' src='https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/Facebook\_icon.svg/2048px-Facebook\_icon.svg.png' alt='Header'/>          {user ? <Button onClick={() => auth.signOut()}>Logout</Button> : (              <div className='app\_loginContainer'>                <Button onClick={() => setOpenSignIn(true)}>Sign In</Button>                <Button onClick={() => setOpen(true)}>Sign Up</Button>              </div>          )}        </div>        <div>          <div className='app\_posts'>            {              posts.map(post => (                <Post                    key={post.\_id}                    username={post.user}                    caption={post.caption}                    imageUrl={post.image}                />              ))            }          </div>        </div>        {user?.displayName ? <ImageUpload username={user.displayName}/> : <h3 className='app\_notLogin'>Need to login to upload</h3>}      </div>    );  }  export default App; |

App.css

|  |
| --- |
| .app{    display: grid;    place-items: center;    background-color: #fafafa;  }  .app\_header{    display: flex;    justify-content: space-between;    position: sticky;    top: 0;    z-index: 1;    width: 99%;    background-color: white;    padding: 20px;    border-bottom: 1px solid lightgray;    object-fit: contain;  }  .app\_notLogin{    margin-bottom: 20px;  }  .app\_loginContainer{    margin-right: 10px;  }  .app\_posts{    padding: 20px;  }  .app\_headerImage{      margin-left: 10px;    height: 40px;  }  .app\_signup{    display: flex;    flex-direction: column;  } |

Axios.js

|  |
| --- |
| import axios from "axios"  const instance = axios.create({      baseURL: "http://localhost:9000"  })  export default instance |

Firebase.js

|  |
| --- |
| import firebase from 'firebase/compat/app';  import 'firebase/compat/auth';  import 'firebase/compat/firestore';  import 'firebase/compat/storage';  const firebaseConfig = {    apiKey: "AIzaSyCebILJWt7M3LtJLjHNuEUMY\_I9gvKQQSY",    authDomain: "webapppractiselab6.firebaseapp.com",    projectId: "webapppractiselab6",    storageBucket: "webapppractiselab6.appspot.com",    messagingSenderId: "203096143167",    appId: "1:203096143167:web:683be99868acf07ecc0f8a",    measurementId: "G-148DZEPKG6"  };  const firebaseApp = firebase.initializeApp(firebaseConfig)  const db = firebaseApp.firestore()  const auth = firebase.auth()  const storage = firebase.storage()    export {db, auth, storage} |

Back end:

Model.js

|  |
| --- |
| const mongoose = require('mongoose')  const postsModel = mongoose.Schema({      caption: String,      user: String,      image: String  })    module.exports = mongoose.model('posts', postsModel) |

.env

|  |
| --- |
| MONGO\_URL=mongodb+srv://20521472:nguyenvantruongkhoa20521472@cluster0.cz6lnjt.mongodb.net/?retryWrites=true&w=majority |

Index.js

|  |
| --- |
| const express = require("express")  const app = express()  const mongoose = require("mongoose")  const dotenv = require("dotenv")  const cors = require("cors")  const Pusher = require("pusher")  const Posts = require("./models/postModel")  dotenv.config()  app.use(express.json())  app.use(cors())  mongoose.connect(process.env.MONGO\_URL, {      useNewUrlParser: true,      useUnifiedTopology: true,  }).then(() => console.log("MongoDB connected!")).catch((err) => console.log(err))  const PORT = process.env.PORT || 9000  const pusher = new Pusher({    appId: process.env.PUSHER\_ID,    key: process.env.PUSHER\_KEY,    secret: process.env.PUSHER\_SECRET,    cluster: "ap1",    useTLS: true  })  mongoose.connection.once('open', () => {      const changeStream = mongoose.connection.collection('post').watch()      changeStream.on('change', change => {          console.log(change)          if(change.operationType === "insert") {              console.log('Trigerring Pusher')              pusher.trigger('post', 'inserted', {                  change: change              })          } else {              console.log('Error trigerring Pusher')          }      })  })  app.post('/upload', (req, res) => {      const dbPost = req.body      Posts.create(dbPost, (err, data) => {          if(err)              res.status(500).send(err)          else              res.status(201).send(data)      })  })  app.get('/sync', (req, res) => {      Posts.find((err, data) => {          if(err)              res.status(500).send(err)          else              res.status(201).send(data)      })  })  app.get('/', (req, res) => {      res.send('BACKEND IS RUNNING')    })    app.listen(PORT, () => {      console.log("Backend server is running!")  }) |