gitignore

# See https://help.github.com/articles/ignoring-files/ for more about ignoring files.

# dependencies

/node\_modules

/.pnp

.pnp.js

# testing

/coverage

# production

/build

# misc

.DS\_Store

.env.local

.env.development.local

.env.test.local

.env.production.local

npm-debug.log\*

yarn-debug.log\*

yarn-error.log\*

package

{

  "name": "cvms",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.6.7",

    "cors": "^2.8.5",

    "db": "^5.0.0",

    "express": "^4.18.2",

    "express-fileupload": "^1.4.3",

    "file-upload": "^0.0.0",

    "mongo": "^0.1.0",

    "mongodb": "^6.3.0",

    "myprofile": "^1.0.0",

    "nodemailer": "^6.9.11",

    "nodemon": "^3.1.0",

    "react": "^18.2.0",

    "react-dom": "^18.2.0",

    "react-router-dom": "^6.21.0",

    "react-scripts": "^5.0.1",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  },

  "devDependencies": {

    "babel-loader": "^9.1.3",

    "webpack": "^5.89.0"

  }

}

package.js

{

  "name": "client",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@emotion/react": "^11.11.3",

    "@emotion/styled": "^11.11.0",

    "@mui/icons-material": "^5.15.6",

    "@mui/material": "^5.15.6",

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.6.7",

    "react": "^18.2.0",

    "react-dom": "^18.2.0",

    "react-router-dom": "^6.22.3",

    "react-scripts": "5.0.1",

    "redux": "^5.0.1",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

README

**# Getting Started with Create React App**

This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).

**## Available Scripts**

In the project directory, you can run:

**### `npm start`**

Runs the app in the development mode.\

Open [http://localhost:3000](http://localhost:3000) to view it in your browser.

The page will reload when you make changes.\

You may also see any lint errors in the console.

**### `npm test`**

Launches the test runner in the interactive watch mode.\

See the section about [running tests](https://facebook.github.io/create-react-app/docs/running-tests) for more information.

**### `npm run build`**

Builds the app for production to the `build` folder.\

It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.\

Your app is ready to be deployed!

See the section about [deployment](https://facebook.github.io/create-react-app/docs/deployment) for more information.

**### `npm run eject`**

**\*\*Note: this is a one-way operation. Once you `eject`, you can't go back!\*\***

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

**## Learn More**

You can learn more in the [Create React App documentation](https://facebook.github.io/create-react-app/docs/getting-started).

To learn React, check out the [React documentation](https://reactjs.org/).

**### Code Splitting**

This section has moved here: [https://facebook.github.io/create-react-app/docs/code-splitting](https://facebook.github.io/create-react-app/docs/code-splitting)

**### Analyzing the Bundle Size**

This section has moved here: [https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size](https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size)

**### Making a Progressive Web App**

This section has moved here: [https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app](https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app)

**### Advanced Configuration**

This section has moved here: [https://facebook.github.io/create-react-app/docs/advanced-configuration](https://facebook.github.io/create-react-app/docs/advanced-configuration)

**### Deployment**

This section has moved here: [https://facebook.github.io/create-react-app/docs/deployment](https://facebook.github.io/create-react-app/docs/deployment)

**### `npm run build` fails to minify**

This section has moved here: [https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify](https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify)

Index

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/fav.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="description" content="KL UNIVERSITY"/>

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <title>KLU</title>

  </head>

  <body>

    <div id="root" class="fullheight"></div>

  </body>

</html>

manifest

{

  "short\_name": "React App",

  "name": "Create React App Sample",

  "icons": [

    {

      "src": "favicon.ico",

      "sizes": "64x64 32x32 24x24 16x16",

      "type": "image/x-icon"

    },

    {

      "src": "logo192.png",

      "type": "image/png",

      "sizes": "192x192"

    },

    {

      "src": "logo512.png",

      "type": "image/png",

      "sizes": "512x512"

    }

  ],

  "start\_url": ".",

  "display": "standalone",

  "theme\_color": "#000000",

  "background\_color": "#ffffff"

}

robots

# https://www.robotstxt.org/robotstxt.html

User-agent: \*

Disallow:

home.js

import React from 'react'

import './home.css'

import logouticon from './images/logout.png'

import { callApi, errorResponse, getSession, setSession } from './main';

import menuicon from './images/menu.png'

const logoutstyle = {"float" : "right", "padding-right" : "5px", "cursor" : "pointer"};

const titlestyle = {"padding-left" : "5px", "font-weight" : "bold"};

const userlabelstyle = {"float" : "right", "padding-right" : "20px"};

export function loadUname(res)

{

   var data = JSON.parse(res);

   var HL1 = document.getElementById("HL1");

   HL1.innerText = `${data[0].firstname} ${data[0].lastname}`;

}

export function loadMenu(res)

{

    var data = JSON.parse(res);

    var list = "";

    for(var x in data)

    {

        list += `<li>

                    <label id='${data[x].mid}L'>${data[x].mtitle}</label>

                    <div id='${data[x].mid}'></div>

                 </li>`;

    }

    var mlist = document.getElementById("mlist");

    mlist.innerHTML = list;

    for(x in data)

    {

        document.getElementById(`${data[x].mid}L`).addEventListener("click", showSMenu.bind(null, data[x].mid));

    }

}

export function loadSMenu(res)

{

    var data = JSON.parse(res);

    var slist = "";

    for(var x in data)

    {

        slist += `<label id='${data[x].smid}'>${data[x].smtitle}</label>`;

    }

    var smenu = document.getElementById(`${data[0].mid}`);

    smenu.innerHTML = slist;

    for(x in data)

    {

        document.getElementById(`${data[x].smid}`).addEventListener("click", loadModule.bind(null, data[x].smid));

    }

}

export function loadModule(smid)

{

    var titlebar = document.getElementById('titlebar');

    var module = document.getElementById('module');

    switch(smid)

    {

        case "M10102":

            module.src = "/myprofile";

            titlebar.innerText = "My Profile";

            break;

        case "M10103":

            module.src = "/changepassword";

            titlebar.innerText = "Change Password";

            break;

        default:

            module.src="";

            titlebar.innerText="";

    }

}

export function showSMenu(mid)

{

    var surl = "http://localhost:5000/home/menus";

    var ipdata = JSON.stringify({

        mid: mid

    });

    callApi("POST", surl, ipdata, loadSMenu, errorResponse);

    var smenu = document.getElementById(mid);

    if(smenu.style.display === 'block')

        smenu.style.display = 'none';

    else

        smenu.style.display = 'block';

}

class Home extends React.Component

{

    constructor()

    {

        super();

        this.sid = getSession("sid");

        //alert(this.sid);

        if(this.sid === "")

            window.location.replace("/");

        var url = "http://localhost:5000/home/uname";

        var data = JSON.stringify({

            emailid : this.sid

        });

        callApi("POST", url, data, loadUname, errorResponse);

        url = "http://localhost:5000/home/menu";

        callApi("POST", url, "", loadMenu, errorResponse);

    }

    logout()

    {

        setSession("sid", "", -1);

        window.location.replace("/");

    }

    render()

    {

        return(

            <div className='fullheight'>

                <div className='header'>

                    <label style={titlestyle}>KL University | Counselling and Visitors Management System</label>

                    <label style={logoutstyle} onClick={this.logout}>Logout</label>

                    <img src={logouticon} alt='' className='logouticon' onClick={this.logout} />

                    <label id='HL1' style={userlabelstyle}></label>

                </div>

                <div className='content'>

                    <div className='menubar'>

                        <div className='menuheader'>

                            <img src={menuicon} alt='' />

                            <label>Menu</label>

                        </div>

                        <div className='menu'>

                            <nav><ul id='mlist' className='mlist'></ul></nav>

                        </div>

                    </div>

                    <div className='outlet'>

                        <div id='titlebar'></div>

                        <iframe id='module' src="" title='module'></iframe>

                    </div>

                </div>

                <div className='footer'>Copyright @ KL University. All rights reserved.</div>

            </div>

        );

    }

}

export default Home;

index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import Login from './login';

import Home from './home';

import ChangePassword from './changepassword';

import AxiosTesting from './axiostesting';

import MyProfile from './myprofile';

import {BrowserRouter, Routes, Route} from 'react-router-dom';

function Website(){

  return(

    <BrowserRouter>

      <Routes>

        <Route path="/" element={<Login/>} />

        <Route path='/home' element={<Home/>} />

        <Route path='/changepassword' element={<ChangePassword/>} />

        <Route path='/axiostesting' element={<AxiosTesting/>} />

        <Route path='/myprofile' element={<MyProfile/>} />

      </Routes>

    </BrowserRouter>

  );

}

ReactDOM.render(<Website/> , document.getElementById('root'));

login.js

import React from 'react';

import logo from './images/logo.png'

import { callApi, errorResponse, setSession } from './main';

const popupwindowstyle={width: '300px', height:'450px', background: 'white'}

const logostyle={width:'75px', height:'75px', left:'113px', position:'absolute', top:'10px'}

const logodiv={height:'100px'}

const space={height: '20px'}

function Login(){

    window.onload=function(){

        var login = document.getElementById('login');

        login.style.display = 'block';

    }

    function validate()

    {

        var T1=document.getElementById('T1');

        var T2=document.getElementById('T2');

        var url = "http://localhost:5000/login/signin";

        var data = JSON.stringify({

            emailid : T1.value,

            pwd : T2.value

        });

        callApi("POST", url, data, loginSuccess, errorResponse);

    }

    function loginSuccess(res)

    {

        var data = JSON.parse(res);

        if(data === 1)

        {

            var T1=document.getElementById('T1');

            setSession("sid", T1.value, (24 \* 60));

            window.location.replace("/home");

        }

        else

            alert("Invalid Credentials!");

    }

    function registration(){

        var login = document.getElementById('login');

        var registration = document.getElementById('registration');

        var T1 = document.getElementById('T1');

        var T2 = document.getElementById('T2');

        T1.value = "";

        T2.value = "";

        login.style.display = 'none';

        registration.style.display = 'block';

    }

    function signup(){

        var RT1 = document.getElementById('RT1');

        var RT2 = document.getElementById('RT2');

        var RT3 = document.getElementById('RT3');

        var RT4 = document.getElementById('RT4');

        var RT5 = document.getElementById('RT5');

        RT1.style.border="";

        RT2.style.border="";

        RT3.style.border="";

        RT4.style.border="";

        RT5.style.border="";

        if(RT1.value === "")

        {

            RT1.style.border="1px solid red";

            RT1.focus();

            return;

        }

        if(RT2.value === "")

        {

            RT2.style.border="1px solid red";

            RT2.focus();

            return;

        }

        if(RT3.value === "")

        {

            RT3.style.border="1px solid red";

            RT3.focus();

            return;

        }

        if(RT4.value === "")

        {

            RT4.style.border="1px solid red";

            RT4.focus();

            return;

        }

        if(RT5.value === "")

        {

            RT5.style.border="1px solid red";

            RT5.focus();

            return;

        }

        var data = JSON.stringify({

            firstname : RT1.value,

            lastname : RT2.value,

            contactno : RT3.value,

            emailid : RT4.value,

            pwd : RT5.value

        });

        var url = "http://localhost:5000/registration/signup";

        callApi("POST", url, data, signupSuccess, errorResponse);

        //alert("Registered successfully...");

        RT1.value = "";

        RT2.value = "";

        RT3.value = "";

        RT4.value = "";

        RT5.value = "";

        var login = document.getElementById('login');

        var registration= document.getElementById('registration');

        registration.style.display = 'none';

        login.style.display = 'block';

    }

    //signupSuccess

    function signupSuccess(res)

    {

        var data = JSON.parse(res);

        alert(data);

    }

    return(

        <div className='fullheight'>

            <div id='loginheader' className='loginheader'>KL University</div>

            <div id='logincontent'>

                <div id='login' className='popup'>

                    <div id='loginwindow' className='popupwindow' style={popupwindowstyle}>

                        <div id='logintxt'>Login</div>

                        <div id='logininnerwindow' className='innerwindow'>

                            <div style={logodiv}> <img src={logo} alt='' style={logostyle} /> </div>

                            <div>Username\*</div>

                            <div> <input type='text' id='T1' className='textbox' autoComplete='off' /> </div>

                            <div style={space}></div>

                            <div>Password\*</div>

                            <div> <input type='password' id='T2' className='textbox' /> </div>

                            <div style={space}></div>

                            <div> <button className='btn' onClick={validate} >Sign In</button> </div>

                            <div style={space}></div>

                            <div style={space}></div>

                            <div>New user? <label className='linklabel' onClick={registration} >Register here</label></div>

                        </div>

                    </div>

                </div>

                <div id="registration" className='popup'>

                    <div id='registrationwindow' className='popupwindow' style={popupwindowstyle}>

                        <div id='regtxt'>New Registration</div>

                        <div id='reginnerwindow' className='innerwindow'>

                            <div>First Name\*</div>

                            <div> <input type='text' id='RT1' className='textbox' autoComplete='off' /> </div>

                            <div style={space}></div>

                            <div>Last Name\*</div>

                            <div> <input type='text' id='RT2' className='textbox' autoComplete='off' /> </div>

                            <div style={space}></div>

                            <div>Contact Number\*</div>

                            <div> <input type='text' id='RT3' className='textbox' autoComplete='off' /> </div>

                            <div style={space}></div>

                            <div>Email ID\*</div>

                            <div> <input type='text' id='RT4' className='textbox' autoComplete='off' /> </div>

                            <div style={space}></div>

                            <div>Password\*</div>

                            <div> <input type='password' id='RT5' className='textbox' autoComplete='off' /> </div>

                            <div style={space}></div>

                            <div> <button className='btn' onClick={signup} >Sign Up</button> </div>

                        </div>

                    </div>

                </div>

            </div>

            <div id='loginfooter'>Copyright @ KL University. All rights reserved.</div>

        </div>

    );

}

export default Login;

main.js

//callApi

export function callApi(method, url, data, callbackSuccess, callbackError)

{

    var xhttp = new XMLHttpRequest();

    xhttp.open(method, url, true);

    xhttp.setRequestHeader('Content-Type', 'application/json');

    xhttp.send(data);

    xhttp.onreadystatechange = function(){

        if(this.readyState === 4)

        {

            if(this.status === 200)

                callbackSuccess(this.responseText);

            else

                callbackError("Error: 404 - Service not available");

        }

    };

}

//callApiFormData

export function callApiFormData(method, url, data, callbackSuccess, callbackError)

{

    var xhttp = new XMLHttpRequest();

    xhttp.open(method, url, true);

    //xhttp.setRequestHeader('Content-Type', 'application/json');

    xhttp.send(data);

    xhttp.onreadystatechange = function(){

        if(this.readyState === 4)

        {

            if(this.status === 200)

                callbackSuccess(this.responseText);

            else

                callbackError("Error: 404 - Service not available");

        }

    };

}

export function errorResponse(res)

{

    alert(res);

}

export function setSession(sname, svalue, exp)

{

    var d = new Date();

    d.setTime(d.getTime() + (exp \* 60000));

    document.cookie = sname + "=" + svalue + ";expires=" + d.toUTCString() + ";path=/";

}

export function getSession(sname)

{

    sname = sname + "=";

    var decoddedcookie = decodeURIComponent(document.cookie);

    var sn = decoddedcookie.split(";");

    for(var i=0; i<sn.length ; i++)

    {

        var s = sn[i];

        while(s.charAt(0) === ' ')

            s=s.substring(1);

        if(s.indexOf(sname) === 0)

            return s.substring(sname.length, s.length);

    }

    return "";

}

myprofile.js

import React from 'react';

import {callApiFormData, errorResponse, getSession} from './main';

import './myprofile.css';

import Axios from 'axios';

export function userInfo()

{

    var url = "http://localhost:5000/myprofile/userinfo";

    // var data = JSON.stringify({

    //     emailid : getSession("sid")

    // });

    // callApi("POST", url, data, userData, errorResponse);

    Axios.post(url, {emailid: getSession("sid")})

        .then(res => userData(res.data))

        .catch(err => errorResponse(err));

}

export function userData(res)

{

    var data = res;

    // var data = JSON.parse(res);

    var L1 = document.getElementById("L1");

    var L2 = document.getElementById("L2");

    var L3 = document.getElementById("L3");

    var L4 = document.getElementById("L4");

    var IM1 = document.getElementById('IM1');

    //L1.innerText = data[0].firstname;

    L1.innerHTML = `<b style='color:red'>${data[0].firstname}</b>`;

    L2.innerText = data[0].lastname;

    L3.innerText = data[0].contactno;

    L4.innerText = data[0].emailid;

    IM1.src = require('./images/photo/' + data[0].imgurl);

}

export function uploadPhoto()

{

    var FU1 = document.getElementById('FU1');

    var url = "http://localhost:5000/uploaddp";

    var data = new FormData();

    data.append("fname", getSession("sid"));

    data.append("myfile", FU1.files[0]);

    callApiFormData("POST", url, data, uploadSuccess, errorResponse);

}

export function uploadSuccess(res)

{

    var data = JSON.parse(res);

    alert(data);

}

class MyProfile extends React.Component

{

    constructor()

    {

        super();

        this.sid = getSession("sid");

        if(this.sid === "")

            window.location.replace("/");

        userInfo();

    }

    render()

    {

        return(

            <div className='fullheight'>

                <table className='tablestyle'>

                    <tr>

                        <td colSpan={2}><h3>User Information</h3></td>

                    </tr>

                    <tr>

                        <td className='firstcolumn'>First Name</td>

                        <td><label id='L1'></label></td>

                    </tr>

                    <tr>

                        <td className='firstcolumn'>Last Name</td>

                        <td><label id='L2'></label></td>

                    </tr>

                    <tr>

                        <td className='firstcolumn'>Contact No.</td>

                        <td><label id='L3'></label></td>

                    </tr>

                    <tr>

                        <td className='firstcolumn'>Email ID</td>

                        <td><label id='L4'></label></td>

                    </tr>

                    <tr>

                        <td className='firstcolumn'>Upload Photo</td>

                        <td><input type='file' id='FU1' accept='image/jpeg' onChange={uploadPhoto}  /> </td>

                    </tr>

                    <tr>

                        <td className='firstcolumn'>Photo</td>

                        <td><img src='#' id='IM1' alt='' className='imgstyle' />  </td>

                    </tr>

                </table>

            </div>

        );

    }

}

export default MyProfile;

app.js

const express = require('express');

const cors = require('cors');

const {MongoClient} = require('mongodb');

const fileupload = require('express-fileupload');

const nodemailer = require('nodemailer');

const app = express();

app.use(express.json());

app.use(cors());

app.use(fileupload());

const PORT = process.env.PORT || 5000;

app.listen(PORT, console.log('Server running on the port number ${PORT}'));

//Configuration for MongoDB

const url = 'mongodb://localhost:27017/';

const client = new MongoClient(url);

//Simple Web API

app.get('/klef/cse', async function(req, res){

    //res.send("Computer Science and Engineering");

    res.json("Computer Science");

});

app.post('/klef/klu', async function(req, res){

    //res.send("Computer Science and Engineering");

    res.json("K L University");

});

//REGISTRATION MODULE

app.post('/registration/signup', async function(req, res){

    try

    {

        conn = await client.connect();

        db = conn.db('s22');

        users = db.collection('users');

        data = await users.insertOne(req.body);

        res.json("Registered Successfully...");

        conn.close();

    }catch(err)

    {

        res.json(err).status(404);

    }

    //res.json(req.body);

});

//LOGIN MODULE

app.post('/login/signin', async function(req, res){

    try

    {

        conn = await client.connect();

        db = conn.db('s22');

        users = db.collection('users');

        data = await users.count(req.body);

        conn.close();

        res.json(data);

    }catch(err)

    {

        res.json(err).status(404);

    }

});

//HOME MODULE

app.post('/home/uname', async function(req, res){

    try

    {

        conn = await client.connect();

        db = conn.db('s22');

        users = db.collection('users');

        data = await users.find(req.body, {projection : {firstname: true, lastname : true }}).toArray();

        conn.close();

        res.json(data);

    }catch(err)

    {

        res.json(err).status(404);

    }

});

app.post('/home/menu', async function(req, res){

    try

    {

        conn = await client.connect();

        db = conn.db('s22');

        menu = db.collection('menu');

        data = await menu.find({}).sort({mid:1}).toArray();

        conn.close();

        res.json(data);

    }catch(err)

    {

        res.json(err).status(404);

    }

});

app.post('/home/menus', async function(req, res){

    try

    {

        conn = await client.connect();

        db = conn.db('s22');

        menu = db.collection('menus');

        data = await menu.find(req.body).sort({smid:1}).toArray();

        conn.close();

        res.json(data);

    }catch(err)

    {

        res.json(err).status(404);

    }

});

//CHANGE PASSWORD

app.post('/cp/updatepwd', async function(req, res){

    try

    {

        conn = await client.connect();

        db = conn.db('s22');

        users = db.collection('users');

        data = await users.updateOne({emailid:req.body.emailid}, {$set:{pwd: req.body.pwd}});

        conn.close();

        res.json("Password has been Updated..");

    }catch(err)

    {

        res.json(err).status(404);

    }

});

//MyPROFILE

app.post('/myprofile/userinfo', async function(req, res){

    try

    {

        conn = await client.connect();

        db = conn.db('s22');

        users = db.collection('users');

        data = await users.find(req.body).toArray();

        conn.close();

        res.json(data);

    }catch(err)

    {

        res.json(err).status(404);

    }

});

//FILE UPLOAD

app.post('/uploaddp', async function(req, res){

    try

    {

        if(!req.files)

            return res.json("File not found");

        let myfile = req.files.myfile;

        var fname = req.body.fname;

        myfile.mv('../src/images/photo/'+ fname +'.jpg', function(err){

            if(err)

                return res.json("File upload operation failed!");

            res.json("File uploaded successfully...");

        });

        conn = await client.connect();

        db = conn.db('s22');

        users = db.collection('users');

        data = await users.updateOne({emailid: req.body.fname},{$set:{imgurl: fname + '.jpg'}});

        conn.close();

    }catch(err)

    {

        res.json(err).status(404);

    }

});

app.post('/sendemail', async function(req,res){

    try

    {

        var transport = nodemailer.createTransport({

            service: "Gmail",

            host: "smtp.gmail.com",

            port: 445,

            secure: true,

            auth: {user: "karisaiganesh04@gmail.com", pass: "sscksznmidwwyide"}

        });

        var maildata={

            from:"karisaiganesh04@gmail.com",

            to: "2200032461@kluniversity.in",

            subject: "Testing Email",

            text: "This is testing email."

        };

        transport.sendMail(maildata, function(err){

            if(err)

                return res.json("Error: Email sending failed");

            res.json("Email sent successfully");

        });

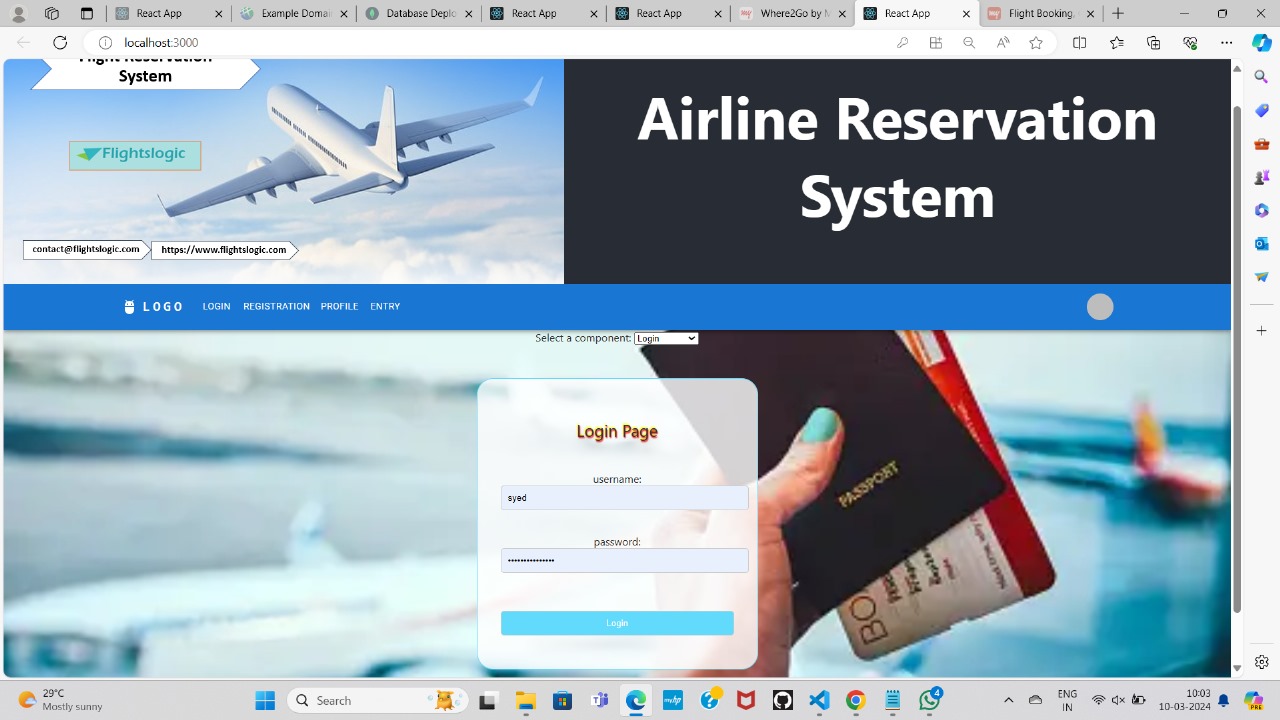
    }catch(err)

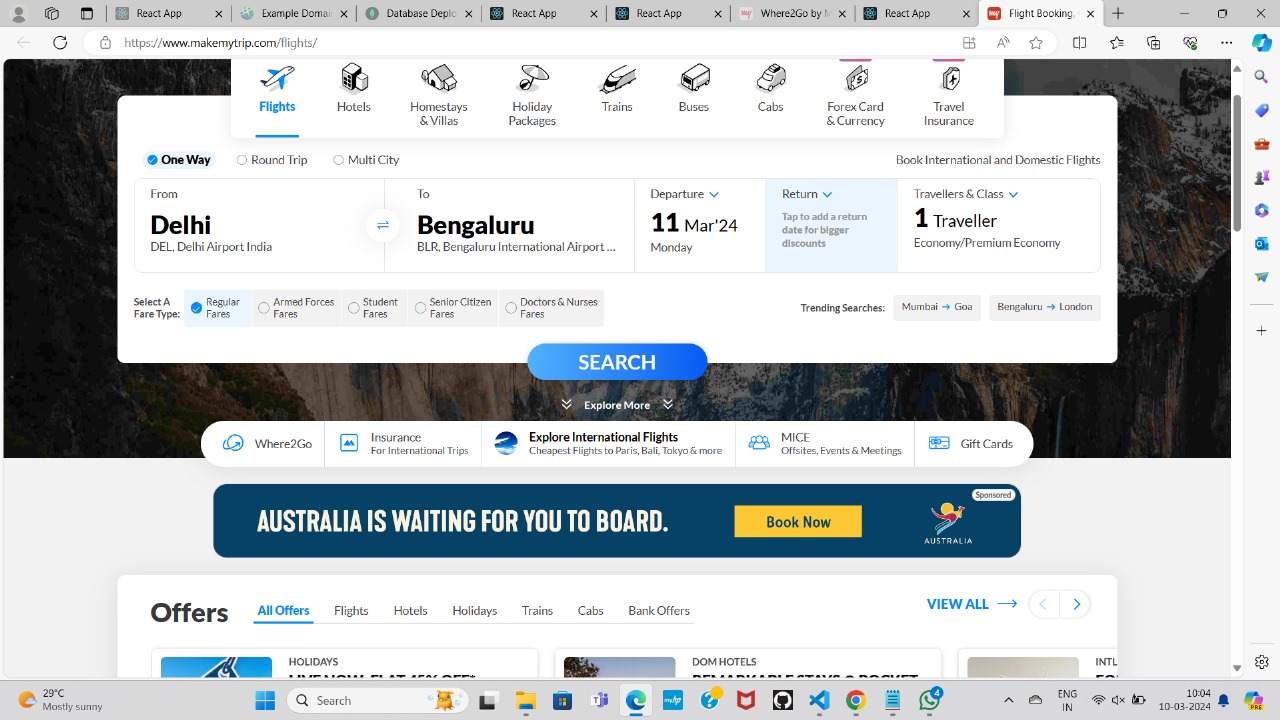
{

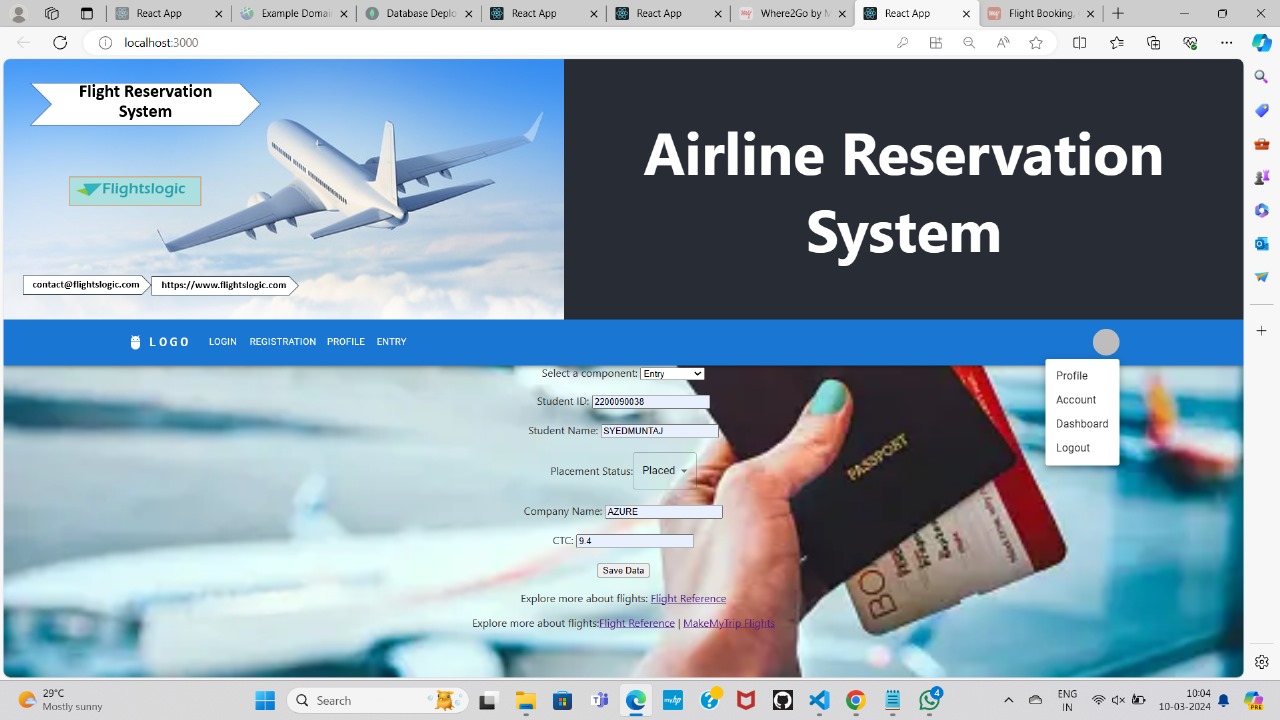
        res.json(err).status(404);

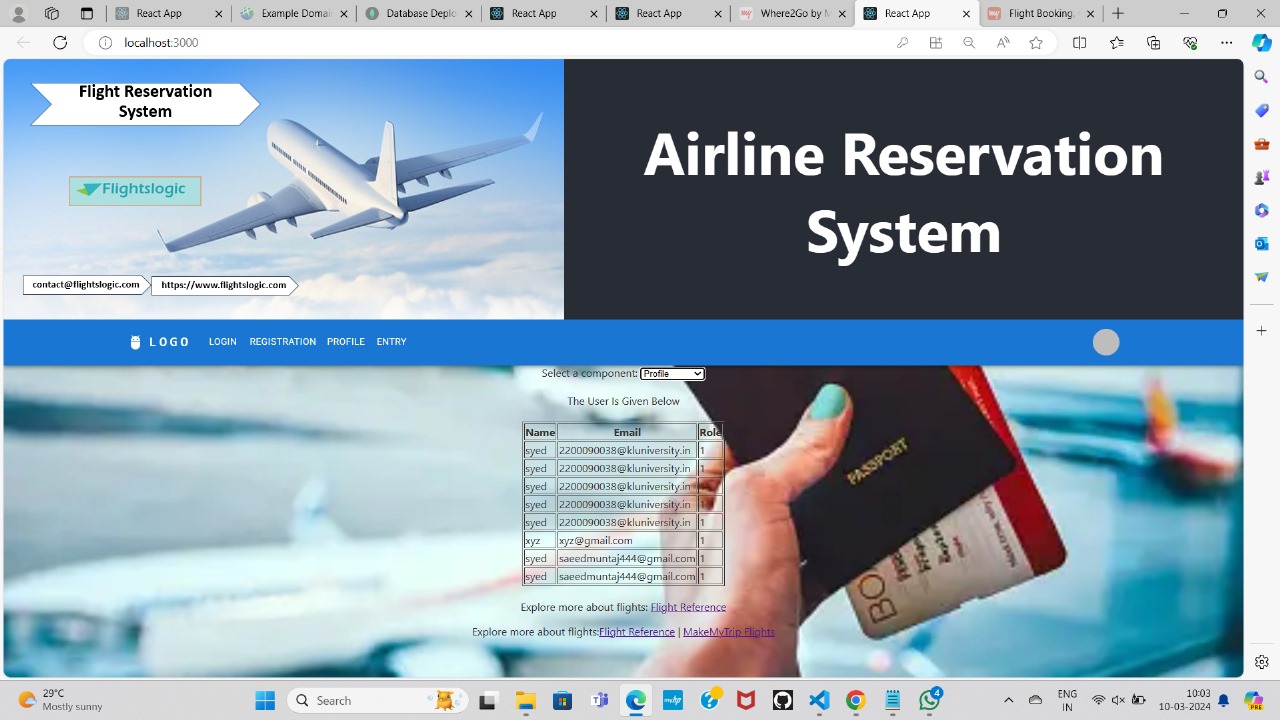
}

});









2200033057\_Ananya Kasim

2200032133\_Akshitha Seeram

2200090038\_Syed Muntaz