##### CONTENT

|  |  |  |
| --- | --- | --- |
| SL NO | **TITLE** | **PAGE NO** |
| 1 | ABSTRACT |  |
| 2 | INTRODUCTION |  |
| 3 | REQUIREMENT SPECIFICATION |  |
| 4 | TECHNOLOGY STACK |  |
| 5 | ANALYSIS AND DESIGN |  |
| 6 | IMPLEMENTATION AND SYSTEM TESTING |  |
| 7 | CODING |  |
| 8 | CONCLUSION |  |
| 9 | REFERENCES |  |

### ABSTRACT

The "College grievance application" is a module that facilitates student queries resolution by allowing them to submit their concerns and allowing teachers to respond to them. This module is an integral part of a larger college application and provides a streamlined process for grievance resolution. The module provides a user-friendly interface for students to submit their queries and tracks the progress of their queries until resolution. Teachers can access the submitted queries, respond to them, and close them once resolved. The module ensures that all queries are addressed in a timely and efficient manner, ensuring student satisfaction and maintaining the reputation of the college.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | INTRODUCTIONThe project module we are discussing is an efficient and user-friendly application that allows students to submit their queries and enables teachers to respond to them. The module is built using popular web development technologies like React, Express, and MySQL, making it a robust and reliable platform for students and teachers alike.The module is designed to cater to the needs of both client users (students) and admin users (teachers). The client interface provides students with an easy-to-use platform to submit their queries, track the progress of their queries, and receive updates on their resolution. The admin interface, on the other hand, allows teachers to view the queries, respond to them, and close them once resolved.This module streamlines the entire grievance resolution process, ensuring that queries are addressed in a timely and efficient manner. With its intuitive design and powerful features, this application is an essential tool for any college looking to maintain a high level of student satisfaction and academic excellence.ADMIN USERS  * **Login Page**: It authenticates admin users by using user ID and password. * **Dashboard**: It provides quick access to most commonly used services in an application. It displays total number of students and staffs registered in the application. * **Manage Staff:**   + **Add Staff:** allows admin to add new staff to database.   + **Remove Staff:** allows user to remove an existing staff from database. * **Manage Student:**   + **Add Student**: allows admin to add new student to database.   + **Remove Student:** allows user to remove an existing student from database. * **Queries:**   + **Active Queries:** List of all active, yet to be resolved queries.   + **Resolved Queries:** List of resolved queries.  CLIENT USERS  * **Login Page**: It authenticates students by using register ID and password. * **Register Page**: Allows students to register to start using the application. * **Home**: Contains college logo and other useful information. * **Queries:** Allows student to post a query, which will further be resolved by any authorized staff or admin.  REQUIREMENT SPECIFICATION  1. **User Interface:** The module should have separate user interfaces for client users (students) and admin users (teachers) with an intuitive and user-friendly design. 2. **Query Submission:** The client users should be able to submit their queries through the application with relevant details and attachments if necessary. 3. **Query Tracking:** The client users should be able to track the status of their queries and receive updates on their resolution. 4. **Query Viewing:** The admin users should be able to view all the submitted queries along with their status and relevant details. 5. **Query Resolution:** The admin users should be able to respond to queries, communicate with clients, and close queries once they are resolved. 6. **User Authentication:** The module should have a secure login system with user authentication to prevent unauthorized access to the system. 7. **Database Management:** The module should have a MySQL database for storing all the submitted queries, their status, and relevant details. 8. **Notification System:** The module should have a notification system that sends email notifications to clients and admins when a query is submitted, updated, or resolved. 9. **Performance Optimization:** The module should be optimized for performance and should be able to handle a large number of queries simultaneously without any delays or crashes. 10. **Data Security:** The module should ensure the confidentiality, integrity, and availability of all the data stored in the system by implementing appropriate security measures such as encryption, access controls  TECHNOLOGY STACKFRONT-END  1. **React js**   ReactJS (or simply React) is a JavaScript library for building user interfaces. It was developed by Facebook and is now open source, meaning that anyone can use it for their projects. ReactJS uses a declarative programming style, which allows developers to describe how they want their UI to look based on the current state of their application, and React takes care of rendering the UI efficiently.  ReactJS works by breaking down a user interface into small, reusable components, which can be combined to create complex UIs. These components can be thought of as building blocks that can be used to create a complete user interface. Each component is responsible for rendering a part of the UI based on the data it receives as input. When the data changes, React updates the UI automatically, which makes it very efficient.  ReactJS also uses a virtual DOM (Document Object Model), which is a lightweight copy of the actual DOM. React updates the virtual DOM based on changes in the data, and then efficiently updates the actual DOM by only making the necessary changes. This approach improves the performance of the application and makes it faster than other approaches that update the entire DOM every time the data changes.  Overall, ReactJS has become a popular choice for building web applications due to its flexibility, efficiency, and large community support.   1. **HTML**   HTML (HyperText Markup Language) is a standard markup language used to create web pages and applications. It is a core technology of the World Wide Web and is used to define the structure and content of web pages. HTML is written in plain text and includes tags and attributes that describe the structure and content of a web page.  HTML documents are made up of various tags and elements that define the content and layout of a web page. These tags can be used to define headings, paragraphs, links, images, tables, forms, and other elements that make up a web page. The tags are used to describe the structure of the content, while the attributes are used to provide additional information or modify the behavior of the elements.  When a web browser loads an HTML page, it parses the HTML code and uses it to display the web page. The browser reads the tags and elements and uses them to create a visual representation of the web page that can be viewed by the user.  HTML has evolved over the years, with new versions and features being added to support modern web development practices. HTML5 is the latest version of HTML, which includes new features such as video and audio elements, canvas, and new input types for forms.   1. **CSS**   CSS stands for Cascading Style Sheets, which is a language used to describe the presentation of HTML or XML documents. It allows developers to separate the structure and content of a web page from its appearance and layout. CSS is used to define various styles and visual elements of a web page, such as colors, fonts, layout, and positioning of elements. It can be used to control the style of text, images, backgrounds, borders, and other elements of a web page, making it an essential tool for web developers and designers. CSS also makes it easier to maintain and update the design of a website by allowing changes to be made to the style sheet, rather than to each individual page. BACK-END  1. **Node js**   Node.js is an open-source, cross-platform, server-side JavaScript runtime environment that executes JavaScript code outside a web browser. It allows developers to build scalable, high-performance applications using JavaScript on both the client and server sides of the web application.  Node.js was built on top of Google's V8 JavaScript engine, which is also used in Google Chrome. It provides an event-driven, non-blocking I/O model that makes it lightweight and efficient, allowing developers to build fast and scalable applications.  Node.js is commonly used for building server-side applications, real-time applications, and network applications. It has a large and active community that provides a vast number of modules and packages to extend its functionality, making it a popular choice for web developers.   1. **Express js**   Express.js is a popular open-source web application framework built for Node.js. It provides a robust set of features and tools for building web applications and APIs, including routing, middleware support, and templating engines.  Express.js allows developers to easily create web applications by providing a simple and intuitive API that abstracts away many of the low-level details of web development. It offers a flexible routing system that allows developers to easily define the routes and methods used by their application. It also includes middleware support, which enables developers to add functionality to their application such as authentication, logging, and error handling.  Express.js is a lightweight framework that is easy to learn and use. It has a large and active community of developers that provides a wide range of plugins and modules, making it a popular choice for building web applications and APIs with Node.js. DATABASE  1. **MySQL**   MySQL is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for managing and manipulating data. It is one of the most widely used databases in the world, powering many popular websites and applications.  MySQL is a client-server database system, which means that the database runs on a server and can be accessed by client applications using various programming languages. It provides a variety of features such as support for transactions, multiple storage engines, and data replication.  MySQL can be used for a wide range of applications, from small personal projects to large enterprise-level applications. It is well-known for its scalability, reliability, and ease of use. It is also compatible with a wide range of platforms and programming languages, making it a versatile choice for developers.  MySQL is free and open-source software, released under the GNU General Public License. It is supported by a large and active community of developers, who contribute to its development and provide support through online forums and documentation. ANALYSIS AND DESIGNANALYSIS The "college grievance application" project aims to provide a platform for college students to report and resolve grievances related to academic, administrative, or other issues. Here is a brief analysis of the project:  **Functionality:**  The application allows students to submit grievances and track their status through a web-based interface. It also enables the college administration to review and respond to grievances, assign them to relevant authorities, and track their resolution status. The system can send automated notifications to both the students and the administration, keeping them updated on the progress of grievances.  **Usability:**  The application has a user-friendly interface that is easy to navigate, and it provides clear instructions on how to use the system. Students can submit grievances with minimum effort, and the administration can quickly review and respond to them. The application is designed to be accessible on multiple devices, making it convenient for users to access it from anywhere.  **Security:**  The system has robust security measures in place to protect the sensitive data of students and the administration. It uses encryption to secure data transmission, and access to the system is restricted through authentication and authorization controls. The application also provides auditing and logging functionalities to track system activity and identify any security breaches.  **Scalability:**  The application has been designed to scale as the number of users and grievances increase. It can handle a large number of requests simultaneously, and the database can store and retrieve grievances quickly. The system can also be easily upgraded and maintained as per the changing needs of the college.  **Potential challenges:**  One of the challenges that the project may face is the integration of the application with the existing college systems. The project team needs to ensure that the application can seamlessly integrate with the college's IT infrastructure, and there are no compatibility issues. The team also needs to ensure that the application is accessible to all students, including those with disabilities.  **Conclusion:**  Overall, the "college grievance application" project is a useful tool for addressing grievances in a college setting. It provides an efficient and effective way for students to report grievances and for the administration to respond to them. The project has the potential to make the college environment more transparent, accountable, and responsive to the needs of the students. However, the project team needs to ensure that the application is scalable, secure, and user-friendly, and that it can integrate with the college's IT infrastructure. DIAGRAMS  1. **ER Diagram**   An Entity-Relationship (ER) diagram is a visual representation of entities and the relationships between them in a database. ER diagrams are used to design and model relational databases, and they are an important tool in database design.  The main components of an ER diagram are entities, attributes, and relationships. Entities are the things or concepts that the database will store information about. Attributes are characteristics or properties of entities that are used to describe or identify them. Relationships describe the connections between entities, such as how they are related to each other or how they interact.  ER diagrams use symbols to represent entities, attributes, and relationships. Entities are represented by rectangles, and attributes are represented by ovals connected to the entity by lines. Relationships are represented by diamonds, with lines connecting the diamonds to the entities that are related.  There are three types of relationships that can be represented in an ER diagram: one-to-one, one-to-many, and many-to-many. A one-to-one relationship means that each instance of an entity is related to only one instance of another entity. A one-to-many relationship means that each instance of an entity can be related to multiple instances of another entity, but each instance of the other entity can be related to only one instance of the first entity. A many-to-many relationship means that each instance of an entity can be related to multiple instances of another entity, and each instance of the other entity can be related to multiple instances of the first entity.  ER diagrams are useful in database design because they provide a clear, concise representation of the structure of the database. They help designers to identify potential problems with the design, such as redundant data or inconsistent relationships, and to make adjustments before the database is implemented    **1. Client Entity Relationship Diagram**    **2. Admin Entity Relationship Diagram**   1. **User Flow Diagram**   A user flow diagram is a visual representation of the steps a user takes to complete a specific task or achieve a goal on a website, application, or other digital product. It is also sometimes referred to as a user journey or user task flow.  The diagram typically includes a series of connected boxes or shapes that represent each step in the user's journey, with arrows indicating the direction of the flow. It may also include annotations or notes to describe each step in more detail.  User flow diagrams are often used in the early stages of the design process to help designers and developers understand how users will interact with the product and identify potential pain points or areas for improvement. They can also be used to communicate the user experience to stakeholders or team members who are not familiar with the product.  An example of a user flow diagram might show the steps a user takes to make a purchase on an e-commerce website, starting with browsing products, adding items to the cart, entering payment and shipping information, and completing the purchase. By mapping out the user's journey in this way, designers can identify any areas where the user might get stuck or confused and make changes to improve the overall user experience.    **3. User Flow Diagram** SQL TABLES **Admins:**    **All\_staffs:**    **All\_students:**    **Registered\_staffs:**    **Registered\_students:**    **Queries:**   IMPLIMENTATON AND SYSTEM TESTING  * **Plan and Design the App:**   a. Determine the features and functionality of the app.  b. Create a user interface design.  c. Plan the data structure and database design.   * **Set Up the Development Environment:**   a. Install Node.js and NPM on your machine.  b. Install the React framework and any additional libraries you will need.  c. Set up a code editor like VS Code or Atom.   * **Create a New React App:**   a. Use the create-react-app command to create a new React app.  b. Open the app in your code editor.   * **Build the User Interface:**   a. Use React components to build the user interface.  b. Use HTML, CSS, and JavaScript to style the app and make it responsive.   * **Add Functionality:**   a. Implement the app's features and functionality using React components and JavaScript.  b. Use APIs to connect to a database and retrieve or update data.  c. Use React Router to handle navigation between pages.   * **Test and Debug:**   a. Test the app to ensure that it works as expected.  b. Use debugging tools to fix any issues.  c. Make sure the app is responsive and works well on different devices and browsers.   * **Deploy the App:**   a. Choose a hosting platform like Heroku or Netlify.  b. Set up a production environment and deploy the app.  c. Test the app in the production environment to ensure that it works correctly. **CLIENT UI**Login Page  Register Page  Home Page  Side Panel  **ADMIN UI**Login Page  Dashboard  Sidebar  Sidebar sub-menus  Add staff  Remove staff  Add student  Remove student  CODINGAdmin application **./package.json**  {    "name": "admin",    "version": "0.1.0",    "private": true,    "dependencies": {      "@testing-library/jest-dom": "^5.16.5",      "@testing-library/react": "^13.4.0",      "@testing-library/user-event": "^13.5.0",      "react": "^18.2.0",      "react-dom": "^18.2.0",      "react-router": "^6.8.2",      "react-router-dom": "^6.8.2",      "react-scripts": "5.0.1",      "react-toastify": "^9.1.1",      "styled-components": "^5.3.9",      "universal-cookie": "^4.0.4",      "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"      ]    }  }  **./public/index.html**  <!DOCTYPE html>  <html lang="en">    <head>      <meta charset="utf-8" />      <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />      <meta name="viewport" content="width=device-width, initial-scale=1" />      <meta name="theme-color" content="#000000" />      <meta        name="description"        content="Web site created using create-react-app"      />      <link href="./fontawesome-icons/css/all.css" rel="stylesheet">      <title>Silicon Admin Panel</title>    </head>    <body>      <div id="root"></div>      <div id="portal"></div>    </body>  </html>  **./src/index.js**  import React from 'react';  import {BrowserRouter} from 'react-router-dom';  import ReactDOM from 'react-dom/client';  import App from './App';  import { AuthProvider } from './context/AuthProvider';  const root = ReactDOM.createRoot(document.getElementById('root'));  root.render(    <React.StrictMode>      <AuthProvider>        <BrowserRouter>          <App />        </BrowserRouter>      </AuthProvider>    </React.StrictMode>  );  **./src/app.js**  import './App.css';  import { Route, Routes } from 'react-router-dom';  import { ToastContainer } from 'react-toastify';  import 'react-toastify/dist/ReactToastify.css';  import Login from './pages/Login';  import Layout from './components/globals/Layout';  import Dashboard from './pages/Dashboard';  import RequireAuth from './utils/RequireAuth';  import Notification from './pages/Notification';  import Event from './pages/Event';  import Update from './pages/Update';  import AddStaff from './pages/AddStaff';  import RemoveStaff from './pages/RemoveStaff';  import EditStaff from './pages/EditStaff';  import AddStudent from './pages/AddStudent';  import RemoveStudent from './pages/RemoveStudent';  import EditStudent from './pages/EditStudent';  function App() {    return (      <div className="App">        <Routes>          <Route path="/login" element={<Login />} />          <Route element={<RequireAuth/>}>            <Route path="/" element={<Layout><Dashboard/></Layout>}/>            <Route path="/dashboard" element={<Layout><Dashboard/></Layout>}/>            <Route path="/post/notification" element={<Layout><Notification/></Layout>}/>            <Route path="/post/event" element={<Layout><Event/></Layout>}/>            <Route path="/post/update" element={<Layout><Update/></Layout>}/>            <Route path="/addStaff" element={<Layout><AddStaff/></Layout>}/>            <Route path="/removeStaff" element={<Layout><RemoveStaff/></Layout>}/>            <Route path="/editStaff" element={<Layout><EditStaff/></Layout>}/>            <Route path="/addStudent" element={<Layout><AddStudent/></Layout>}/>            <Route path="/removeStudent" element={<Layout><RemoveStudent/></Layout>}/>            <Route path="/editStudent" element={<Layout><EditStudent/></Layout>}/>          </Route>        </Routes>        <ToastContainer          position="top-center"          autoClose={3000}          hideProgressBar={false}          newestOnTop={false}          closeOnClick          rtl={false}          pauseOnFocusLoss          draggable          pauseOnHover          theme="colored"        />      </div>    );  }  export default App;  **./src/app.css**  \*{    box-sizing: border-box;    margin: 0;    padding: 0;  }  .App {    text-align: center;  }  **./src/components/globals/Layout.jsx**  import React from 'react';  import Navbar from './Navbar';  const Layout = ({children}) => {    return (      <>          <Navbar />          {children}      </>    )  }  export default Layout  **./src/components/globals/Navbar.jsx**  import React, { useState } from 'react';  import styled from 'styled-components';  import { SidebarData } from './SidebarData';  import SubMenu from './SubMenu';  const Nav = styled.div`    background: #15171c;    height: 80px;    display: flex;    justify-content: flex-start;    align-items: center;  `;  const Header = styled.h1`    color: white;    display: flex;    width: 100%;    align-items: center;    justify-content: center;  `;  const NavIcon = styled.i`    color: white;    margin-left: 2rem;    font-size: 2rem;    height: 80px;    display: flex;    justify-content: flex-start;    align-items: center;  `;  const SidebarNav = styled.nav`    background: #15171c;    width: 250px;    height: 100vh;    display: flex;    justify-content: center;    position: fixed;    top: 0;    left: ${({ sidebar }) => (sidebar ? '0' : '-100%')};    transition: 350ms;    z-index: 10;    overflow-y: scroll;    ::-webkit-scrollbar {      display: none;    }    -ms-overflow-style: none;  /\* IE and Edge \*/    scrollbar-width: none;  /\* Firefox \*/  `;  const SidebarWrap = styled.div`    width: 100%;  `;  const Sidebar = () => {    const [sidebar, setSidebar] = useState(false);    const showSidebar = () => setSidebar(!sidebar);    return (      <>          <Nav>            <NavIcon to='#'>              <i class="fa-solid fa-bars" onClick={showSidebar} />            </NavIcon>            <Header>              <span>SILICON CITY COLLEGE</span>            </Header>          </Nav>          <SidebarNav sidebar={sidebar}>            <SidebarWrap>              <NavIcon to='#'>                <i class="fa-solid fa-square-xmark" onClick={showSidebar} />              </NavIcon>              {SidebarData.map((item, index) => {                return <SubMenu item={item} key={index} />;              })}            </SidebarWrap>          </SidebarNav>      </>    );  };  export default Sidebar;  **./src/components/globals/SidebarData.js**  import React from 'react';  export const SidebarData = [    {      title: 'Dashboard',      path: '/dashboard',      icon: <i class="fa fa-home" aria-hidden="true"/>,      iconClosed: <i class="fa-solid fa-sort-down" />,      iconOpened: <i class="fa-solid fa-sort-up" />    },    {      title: 'Post',      path: '',      icon: <i class="fa-solid fa-envelope"/>,      iconClosed: <i class="fa-solid fa-sort-down" />,      iconOpened: <i class="fa-solid fa-sort-up"/>,      subNav: [        {          title: 'Notification',          path: '/post/notification',          icon: <i class="fa-solid fa-message"/>,          cName: 'sub-nav'        },        {          title: 'Event',          path: '/post/event',          icon: <i class="fa-solid fa-calendar"/>,          cName: 'sub-nav'        },        {          title: 'Update',          path: '/post/update',          icon: <i class="fa-solid fa-square-pen"/>        }      ]    },    {      title: 'Manage Staffs',      path: '',      icon: <i class="fa-solid fa-person-chalkboard"/>,      iconClosed: <i class="fa-solid fa-sort-down" />,      iconOpened: <i class="fa-solid fa-sort-up"/>,      subNav: [        {          title: 'Add staff',          path: '/addstaff',          icon: <i class="fa-solid fa-user-plus"/>,          cName: 'sub-nav'        },        {          title: 'Remove staff',          path: '/removestaff',          icon: <i class="fa-solid fa-user-minus"/>,          cName: 'sub-nav'        },        /\* {          title: 'Edit staff',          path: '/editstaff',          icon: <i class="fa-solid fa-user-pen"/>        } \*/      ]    },    {      title: 'Manage Students',      path: '',      icon: <i class="fa-solid fa-graduation-cap"/>,      iconClosed: <i class="fa-solid fa-sort-down" />,      iconOpened: <i class="fa-solid fa-sort-up"/>,      subNav: [        {          title: 'Add student',          path: '/addstudent',          icon: <i class="fa-solid fa-user-plus"/>,          cName: 'sub-nav'        },        {          title: 'Remove student',          path: '/removestudent',          icon: <i class="fa-solid fa-user-minus"/>,          cName: 'sub-nav'        },        /\* {          title: 'Edit student',          path: '/editstudent',          icon: <i class="fa-solid fa-user-pen"/>        } \*/      ]    },    {      title: 'Queries',      path: '',      icon: <i class="fa-solid fa-clipboard-question"/>,      iconClosed: <i class="fa-solid fa-sort-down" />,      iconOpened: <i class="fa-solid fa-sort-up"/>,      subNav: [        {          title: 'Active Queries',          path: '/activequeries',          icon: <i class="fa-brands fa-creative-commons-share"/>,          cName: 'sub-nav'        },        {          title: 'Resolved Queries',          path: '/resolvedqueries',          icon: <i class="fa-sharp fa-solid fa-circle-check"/>,          cName: 'sub-nav'        }      ]    }  ];  **./src/components/SubMenu.jsx**  import React, { useState } from 'react';  import { Link } from 'react-router-dom';  import styled from 'styled-components';  const SidebarLink = styled(Link)`    display: flex;    color: #e1e9fc;    justify-content: space-between;    align-items: center;    padding: 20px;    list-style: none;    height: 60px;    text-decoration: none;    font-size: 18px;    &:hover {      background: #252831;      border-left: 4px solid #d8cdf3;      cursor: pointer;    }  `;  const SidebarLabel = styled.span`    margin-left: 16px;  `;  const DropdownLink = styled(Link)`    background: #414757;    height: 60px;    padding-left: 3rem;    display: flex;    align-items: center;    text-decoration: none;    color: #f5f5f5;    font-size: 18px;    &:hover {      background: #241e34;      cursor: pointer;    }  `;  const SubMenu = ({ item }) => {    const [subnav, setSubnav] = useState(false);    const showSubnav = () => setSubnav(!subnav);    return (      <>        <SidebarLink to={item.path} onClick={item.subNav && showSubnav}>          <div>            {item.icon}            <SidebarLabel>{item.title}</SidebarLabel>          </div>          <div>            {item.subNav && subnav              ? item.iconOpened              : item.subNav              ? item.iconClosed              : null}          </div>        </SidebarLink>        {subnav &&          item.subNav.map((item, index) => {            return (              <DropdownLink to={item.path} key={index}>                {item.icon}                <SidebarLabel>{item.title}</SidebarLabel>              </DropdownLink>            );          })}      </>    );  };  export default SubMenu;  **./src/conext/AuthProvider.js**  import { createContext, useState } from 'react';  import Cookies from 'universal-cookie';    const AuthContext = createContext({})  export const AuthProvider = ({ children }) => {      const cookies = new Cookies();      const [auth, setAuth] = useState(cookies.get('admin'));      return (          <AuthContext.Provider value={{auth, setAuth}}>              {children}          </AuthContext.Provider>      )  }  export default AuthContext;  **./src/modals/popupModal.jsx**  import React from 'react'  import ReactDom from 'react-dom'    const PopupModal = ({ toggleActive, bg, active, children }) => {    if(!active) return null;    const overlay = {      position: "fixed",      top: "0",      bottom: "0",      left: "0",      right: "0",      backgroundColor: "rgba(0,0,0,.8)",      zIndex: "999"  }  const container = {      position: "fixed",      top: "50%",      left: "50%",      transform: "translate(-50%,-50%)",      padding: "5px",      backgroundColor: bg,      zIndex: "1000",      borderRadius: "6px"  }  const closeBtn = {      fontSize: "36px",      color: "orangered",      margin: "8px"  }  const header = {      width: "100%",      display: "inline-flex",      justifyContent: "right"  }  const body = {      textAlign: "center",      padding: "24px"  }    return ReactDom.createPortal(      <div style={overlay}>      <div style={container}>          <div style={header}>              <i class="fa-solid fa-rectangle-xmark" style={closeBtn} onClick={() => toggleActive(({active: false, prompt: ""}))}></i>          </div>          <div style={body}>              {children}          </div>      </div>      </div>,      document.getElementById("portal")    )  }  export default PopupModal  **./src/pages/AddStaff.jsx**  import React, { useState } from 'react';  import "./addStaff.css"  const AddStaff = () => {    const [regId,setRegId] = useState('');    const [name,setName] = useState('');    const [designation,setDesignation] = useState('');    const [gender,setGender] = useState('');    const handleSubmit = (e) => {      e.preventDefault();      fetch(`http://localhost:8080/manage-users/addstaff`, {            method: 'POST',            headers: {              'Content-Type': 'application/json',            },            body: JSON.stringify({              regId,              name,              designation,              gender            })          })          .then(response => response.json())          .then(data => {            if (data.error) {              console.log(data.error.code)              alert("Server Error!")            }else {              console.log(data,"here")            }          })    }    return (      <div className="page-body">        <h1>Add a new staff</h1>        <div className="form-container">          <form onSubmit={handleSubmit}>            <label for="regId">Registration ID</label>            <input type="text" id="regId" name="regId" required onChange={(e) => setRegId(e.target.value)} value={regId}/>            <label for="name">Full Name</label>            <input type="text" id="name" name="name" required onChange={(e) => setName(e.target.value)} value={name}/>            <label for="designation">Designation</label>            <select id="designation" name="designation" required onChange={(e) => setDesignation(e.target.value)} value={designation}>              <option value="">Choose Designation</option>              <option value="male">Principle</option>              <option value="female">Senior Professor</option>              <option value="female">Assistant Professor</option>            </select>            <label for="gender">Gender</label>            <select id="gender" name="gender" required onChange={(e) => setGender(e.target.value)} value={gender}>              <option value="">Select Gender</option>              <option value="male">Male</option>              <option value="female">Female</option>            </select>            <input type="submit" value="Add Staff" />          </form>        </div>      </div>    )  }  export default AddStaff  **./src/pages/AddStudent.jsx**  import React from 'react'  import "./addStaff.css"  const AddStudent = () => {      const handleSubmit = (e) => {      e.preventDefault();    }    return (      <div className="page-body">        <h1>Add a new student</h1>        <div className="form-container">          <form onSubmit={handleSubmit}>            <label for="regId">Registration ID</label>            <input type="text" id="regId" name="regId" required />            <label for="name">Full Name</label>            <input type="text" id="name" name="name" required />            <label for="section">Section</label>            <select id="section" name="section" required >              <option value="">Choose section</option>              <option value="BCOM">BCOM</option>              <option value="BBA">BBA</option>              <option value="BCA">BCA</option>            </select>            <label for="gender">Gender</label>            <select id="gender" name="gender" required >              <option value="">Select Gender</option>              <option value="Male">Male</option>              <option value="Female">Female</option>            </select>            <input type="submit" value="Add Staff" />          </form>        </div>      </div>    )  }  export default AddStudent  **./src/pages/Dashboard.jsx**  import React, {useState,useEffect} from 'react'  import './dashboard.css';  import PopupModal from '../modals/popupModal';  const Dashboard = () => {    const [allStaffs, setAllStaffs] = useState(200);    const [regStaffs, setRegStaffs] = useState(20);    const [allStudents, setAllStudents] = useState(14);    const [regStudents, setRegStudents] = useState(34);    const [active, setActive] = useState(true);    const handleClick = () => {      setActive(!active);    }    return (      <div className="container">        <div className="card ">          <span>Number of staffs in database</span><br/>          {allStaffs}        </div>        <div className="card ">          <span>Registered staffs</span><br/>          {regStaffs}        </div>        <div className="card ">          <span>Number of students in database</span><br/>          {allStudents}        </div>        <div className="card ">          <span>Registered students</span><br/>          {regStudents}        </div>        {active ? <PopupModal active bg="yellow" toggleActive={handleClick}>          <h1>wwe</h1>        </PopupModal>:null}      </div>    )  }  export default Dashboard  **./src/pages/Login.jsx**  import React, {useState,useEffect} from 'react'  import './dashboard.css';  import PopupModal from '../modals/popupModal';  const Dashboard = () => {    const [allStaffs, setAllStaffs] = useState(200);    const [regStaffs, setRegStaffs] = useState(20);    const [allStudents, setAllStudents] = useState(14);    const [regStudents, setRegStudents] = useState(34);    const [active, setActive] = useState(true);    const handleClick = () => {      setActive(!active);    }    return (      <div className="container">        <div className="card ">          <span>Number of staffs in database</span><br/>          {allStaffs}        </div>        <div className="card ">          <span>Registered staffs</span><br/>          {regStaffs}        </div>        <div className="card ">          <span>Number of students in database</span><br/>          {allStudents}        </div>        <div className="card ">          <span>Registered students</span><br/>          {regStudents}        </div>        {active ? <PopupModal active bg="yellow" toggleActive={handleClick}>          <h1>wwe</h1>        </PopupModal>:null}      </div>    )  }  export default Dashboard  **./src/pages/RemoveStaff.jsx**  import React, {useState, useEffect} from 'react'  import './removeStaff.css'  const RemoveStaff = () => {    const [tableData, setTableData] = useState([]);      useEffect(() => {      fetch('http://localhost:8080/manage-users/allstaffs')        .then(response => response.json())        .then(data => console.log(data))        .catch(error => console.error(error));    }, []);    const [search, setSearch] = useState('');      const handleSearch = (e) => {      e.preventDefault();      console.log(search)      fetch('http://localhost:8080/manage-users/removestaff', {        method: 'POST',        headers: {          'Content-Type': 'application/json'        },          body: JSON.stringify({ regno: search })      })        .then(response => response.json())        .then(data => console.log(data))        .catch(error => console.error(error))    }    return (      <div className="remove-staff-page">        <div class="search-local">          <div class="icon">            <i class="fa-solid fa-magnifying-glass"/>          </div>          <input type="text" placeholder="Enter Register ID" onChange={(e) => setSearch(e.target.value)} value={search} />          <button onClick={handleSearch}>            <span>Search</span>          </button>        </div>        <div className="table-container">          <table>            <tr>              <th>Name</th>              <th>Register ID</th>              <th>Designation</th>              <th>Gender</th>            </tr>            <tr>              <td>Rohan</td>              <td>R2103511</td>              <td>Professer</td>              <td>Male</td>            </tr>          </table>        </div>      </div>    )  }  export default RemoveStaff  **./src/pages/RemoveStudent.jsx**  import React, {useState, useEffect} from 'react'  import './removeStaff.css'  const RemoveStudent = () => {    const [tableData, setTableData] = useState([]);      useEffect(() => {      fetch('http://localhost:8080/manage-users/allstudent')        .then(response => response.json())        .then(data => console.log(data))        .catch(error => console.error(error));    }, []);    const [search, setSearch] = useState('');      const handleSearch = (e) => {      e.preventDefault();      console.log(search)      fetch('http://localhost:8080/manage-users/removestudent', {        method: 'POST',        headers: {          'Content-Type': 'application/json'        },          body: JSON.stringify({ regno: search })      })        .then(response => response.json())        .then(data => console.log(data))        .catch(error => console.error(error))    }    return (      <div className="remove-staff-page">        <div class="search-local">          <div class="icon">            <i class="fa-solid fa-magnifying-glass"/>          </div>          <input type="text" placeholder="Enter Register ID" onChange={(e) => setSearch(e.target.value)} value={search} />          <button onClick={handleSearch}>            <span>Search</span>          </button>        </div>        <div className="table-container">          <table>            <tr>              <th>Name</th>              <th>Register ID</th>              <th>Section</th>              <th>Gender</th>            </tr>            <tr>              <td>Kiran</td>              <td>R2103705</td>              <td>BCA</td>              <td>Male</td>            </tr>            </table>        </div>      </div>    )  }  export default RemoveStudent  **./src/utils/RequireAuth.js**  import { useLocation, Navigate, Outlet } from "react-router";  import useAuth from "./hooks/useAuth";  const RequireAuth = () => {      const {auth} = useAuth();      const location = useLocation();      return(          auth?.name              ? <Outlet />              : <Navigate to="/login" state={{ from: location }} replace />      )  }  export default RequireAuth;  **./src/utils/hooks/useAuth.js**  import { useContext } from "react";  import AuthContext from "../../context/AuthProvider";  const useAuth = () => {      return useContext(AuthContext);  }  export default useAuth; Client application **./package.json**  {    "name": "silicon-app",    "version": "0.1.0",    "private": true,    "dependencies": {      "@fortawesome/fontawesome-svg-core": "^6.2.1",      "@fortawesome/free-solid-svg-icons": "^6.2.1",      "@fortawesome/react-fontawesome": "^0.2.0",      "@testing-library/jest-dom": "^5.16.5",      "@testing-library/react": "^13.4.0",      "@testing-library/user-event": "^13.5.0",      "axios": "^1.2.6",      "react": "^18.2.0",      "react-dom": "^18.2.0",      "react-router": "^6.8.0",      "react-router-dom": "^6.8.0",      "react-scripts": "5.0.1",      "react-toastify": "^9.1.1",      "styled-components": "^5.3.6",      "universal-cookie": "^4.0.4",      "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"      ]    }  }  **./public/index.html**  <!DOCTYPE html>  <html lang="en">    <head>      <meta charset="utf-8" />      <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />      <meta name="viewport" content="width=device-width, initial-scale=1" />      <meta name="theme-color" content="#000000" />      <meta        name="Silicon City College Application"        content="web application for students and staffs of silicon city college"      />      <title>Silicon App</title>      <link href="./fontawesome-icons/css/all.css" rel="stylesheet">    </head>    <body>      <noscript>You need to enable JavaScript to run this app.</noscript>      <div id="root"></div>      <div id="portal"></div>    </body>  </html>  **./src/index.js**  import React from 'react';  import ReactDOM from 'react-dom/client';  import { BrowserRouter } from 'react-router-dom';  import LoadingScreen from './utils/LoadingScreen';  import App from './App';  import { AuthProvider } from './context/AuthProvider';    const root = ReactDOM.createRoot(document.getElementById('root'));  root.render(<LoadingScreen />);  window.addEventListener('load', () => {    root.render(      <React.StrictMode>        <AuthProvider>          <BrowserRouter>            <App />          </BrowserRouter>        </AuthProvider>      </React.StrictMode>    );  })    **./src/app.js**  import './App.css';  import {React} from 'react';  import { Route, Routes } from 'react-router-dom';  import { ToastContainer } from 'react-toastify';  import 'react-toastify/dist/ReactToastify.css';  import LogoAnimation from './utils/LogoAnimation';  import Login from './pages/Login';  import Register from './pages/Register';  import Layout from './components/global/Layout';  import Home from './pages/Home';  import Notifications from './pages/Notifications';  import Queries from './pages/Queries';  import RequireAuth from './utils/RequireAuth';  import Profile from './pages/Profile';  import SocialMedia from './pages/SocialMedia';  import Notes from './pages/Notes';  function App() {    return (        <div className="App">          <LogoAnimation />          <Routes>            <Route path="/login" element={<Login />} />            <Route path="/register" element={<Register />} />            <Route path="/resetpassword" element={<div>Reset Password Page(not yet finished)</div>} />            <Route element={<RequireAuth/>}>              <Route path="/" element={<Layout><Home/></Layout>}/>              <Route path="/home" element={<Layout><Home/></Layout>}/>              <Route path="/notifications" element={<Layout><Notifications/></Layout>}/>              <Route path="/queries" element={<Layout><Queries/></Layout>}/>              <Route path="/profile" element={<Layout><Profile/></Layout>}/>              <Route path="/notes" element={<Layout><Notes/></Layout>}/>              <Route path="/social" element={<Layout><SocialMedia/></Layout>}/>            </Route>          </Routes>          <ToastContainer            position="top-center"            autoClose={3000}            hideProgressBar={false}            newestOnTop={false}            closeOnClick            rtl={false}            pauseOnFocusLoss            draggable            pauseOnHover            theme="colored"          />        </div>    );  }  export default App;  **./src/app.css**  \*{    margin: 0;    padding: 0;    box-sizing: border-box;    /\*font-family: 'Poppins', sans-serif;\*/    overflow: hidden;  }  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;  }  .App {    text-align: center;    overflow: hidden;  }  **./src/components/QuoteBox.jsx**  import React, { useEffect, useState } from 'react';  import './QuoteBox.css';  const QuoteBox = () => {    const [quote, setQuote] = useState({})    useEffect(() => {      fetch("https://type.fit/api/quotes")        .then(function(response) {          return response.json();        })        .then(function(data) {          setQuote(data[Math.floor(Math.random() \* 1642) + 1]);        });    },[])    return (      <div class="card">        <blockquote>          <p>{quote.text}</p>        </blockquote>        <h3 class="name">{quote.author}</h3>      </div>    )  }  export default QuoteBox  **./src/components/quoteBox.css**  .card {    font-size: 14pt;    position: relative;    margin: 6px auto;    max-width: 600px;    width: 96vw;    background-color: #fff;    border-radius: 5px;    box-shadow: 1px 2px 2px 0 rgba(0, 0, 0, 0.2);    padding: 0.75rem 1rem;    z-index: -1;  }  .card p {    margin-top: 0;  }  blockquote {    margin: 0;    font-style: italic;  }  blockquote p::before {    content: "“";    font-family: serif;    font-size: 15rem;    color: #134ead;    position: absolute;    top: -50px;    left: -35px;    opacity: 0.1;  }  .name {    margin: 0;    color: #e0597c;  }  .title {    font-size: 0.75rem;    margin: 0;  }  **./src/components/globals/Layout.jsx**  import React from 'react';  import NavBar from './NavBar';  const Layout = ({children}) => {    return (      <>        <NavBar/>        {children}      </>    )  }  export default Layout  **./src/components/globals/Navbar.jsx**  import React, {useState} from 'react'  import { Link } from "react-router-dom";  import './navbar.css';  import {toast} from 'react-toastify'  import userLogo from '../../assets/user.png';  import useAuth from '../../utils/hooks/useAuth';  import Cookies from 'universal-cookie';    const SidebarData = [    {      title: "Home",      path: "/",      icon: <i class="fa-solid fa-house"></i>,      cName: "nav-text",    },    {      title: "Notifications",      path: "/notifications",      icon: <i class="fa-solid fa-pen-to-square"></i>,      cName: "nav-text",    },    {      title: "Silicon Connect",      path: "/social",      icon: <i class="fa-sharp fa-solid fa-circle-nodes"></i>,      cName: "nav-text",    },    {      title: "Notes",      path: "/notes",      icon: <i class="fa-solid fa-book"></i>,      cName: "nav-text",    },    {      title: "Queries",      path: "/queries",      icon: <i class="fa-solid fa-clipboard-question"></i>,      cName: "nav-text",    },  ];  const NavBar = () => {      const { auth } = useAuth();    const [sidebar, setSidebar] = useState(false);    const [currentPage, setCurrentPage] = useState("Home")    const showSidebar = () => setSidebar(!sidebar);    const cookies = new Cookies();      const logoutHandler = () => {      cookies.remove('user');      toast.info("User logger out successfully");    }    return (      <div>          <div className="navbar">            <Link to="#" className="menu-bars">            <i class="fa-solid fa-bars-staggered" onClick={showSidebar}></i>            </Link>            <div className="nav-title">{currentPage}</div>          </div>          <nav className={sidebar ? "nav-menu active" : "nav-menu"}>            <ul className="nav-menu-items" onClick={showSidebar}>              <li className="navbar-toggle">                <Link to="#" className="menu-bars">                  <i class="fa-solid fa-xmark"></i>                </Link>              </li>              <li>                <Link to="/profile" className="profile-box" onClick={() => setCurrentPage("Profile")}>                  <div className="profile-icon">                    <img src={userLogo} alt="Profile"/>                  </div>                  <span className="profile-name">{auth.name}</span>                </Link>              </li>              <hr/>              {SidebarData.map((item, index) => {                return (                  <li key={index} className={item.cName}>                    <Link to={item.path} onClick={() => setCurrentPage(item.title)}>                      {item.icon}                      <span>{item.title}</span>                    </Link>                  </li>                );              })}            </ul>            <div className="nav-footer">              <Link to="/login" onClick={logoutHandler}>                <div className="logout-btn">                  <span className="btn-text">Log Out</span>                  <i className="fa-solid fa-right-from-bracket"></i>                </div>              </Link>            </div>          </nav>      </div>    )  }  export default NavBar  **./src/components/globals/navbar.css**  hr{    margin: 16px;  }  /\* Navbar CSS \*/  .navbar {    background-color: #ffffff;    height: 80px;    display: flex;    justify-content: flex-start;    align-items: center;  }  .nav-title {    font-size: 2rem;    font-weight: 600;    padding-bottom: 8px;    padding-left: 22px;  }  .menu-bars {    margin-left: 2rem;    font-size: 2rem;    background: none;    color: #000;  }  .nav-menu {    background-color: #ffffff;    width: 250px;    height: 100vh;    display: flex;    justify-content: center;    position: fixed;    top: 0;    left: -100%;    transition: 850ms;  }  .nav-menu.active {    left: 0;    transition: 450ms;  }  .nav-text {    display: flex;    justify-content: flex-start;    align-items: center;    padding: 8px 0px 8px 16px;    list-style: none;    height: 60px;  }  .nav-text a {    text-decoration: none;    color: #151414;    font-size: 18px;    width: 95%;    height: 100%;    display: flex;    align-items: center;    padding: 0 16px;    border-radius: 4px;  }  .nav-text a:hover {    background-color: #000000;    color: #fff;  }  .nav-menu-items {    width: 100%;    overflow-y: scroll;    padding-bottom: 60px;  }  .nav-menu-items::-webkit-scrollbar {    width: 0;  }  .navbar-toggle {    background-color: #ffffff;    width: 100%;    height: 80px;    display: flex;    justify-content: flex-start;    align-items: center;  }  span {    margin-left: 16px;  }  /\*    Profile icon styling \*/  .profile-box {    width: 100%;    display:flex;    flex-direction: column;    align-items: center;    justify-content: center;  }  .profile-icon {    background-color: #000;    width: 90px;    height: 90px;    border-radius: 50%;    overflow: hidden;  }  .profile-icon img{    width: 90px;    height: 90px;  }  .profile-name {    margin: 10px;    font-size: 22px;    color: rgb(66, 66, 236);  }                    /\* Footer \*/  .nav-footer{    background-color: #fff;    width: 100%;    padding: 6px;    margin: 8px;    display: flex;    align-items: center;    justify-content: center;    position: absolute;    bottom: 0;  }  /\*  Log out button styling \*/  .logout-btn {    height: 38px;    width: 120px;    background-color: #1f2335;    border-radius: 6px;    display: flex;    align-items: center;    justify-content: center;  }  .logout-btn i {    color: whitesmoke;    font-size: 24px;    padding: 9px;  }  .btn-text {    font-size: 15px;    color:#fff;    font-weight: 700;  }  **./src/context/AuthProvider.js**  import { createContext, useState } from 'react';  import Cookies from 'universal-cookie';    const AuthContext = createContext({})  export const AuthProvider = ({ children }) => {      const cookies = new Cookies();      const [auth, setAuth] = useState(cookies.get('user'));      return (          <AuthContext.Provider value={{auth, setAuth}}>              {children}          </AuthContext.Provider>      )  }  export default AuthContext;  **./src/pages/Home.jsx**  import React from 'react'  import QuoteBox from '../components/QuoteBox';  import './home.css';  const Home = () => {    return (      <>        <div>          <img src="../scc-logo-nobg.png" alt="clg-logo" />          <div>            <span className="clg-name">SILICON CITY COLLEGE</span><br/>            <span className="desc">(A Unit of MJ Education Trust)</span><br/>            <span className="sub-desc">Re-Accredited by NAAC with 'A' Grade</span><br/>            <span className="sub-desc">Recognised by UGC under section 2(f) & 12(b), Affiliated to Bengaluru North University</span><br/>          </div>        </div>        <hr/>        <QuoteBox/>      </>    )  }  export default Home  **./src/pages/home.css**    .clg-name {      font-size: 46px;      font-weight: 800;      color: rgb(31, 9, 117);  }  .desc {      font-size: 22px;      font-weight: 700;  }  .sub-desc {      font-size: 22px;      font-weight: 700;      color: rgb(31, 9, 117);  }  **./src/pages/Login.jsx**  import React, {useState} from 'react';  import './Login.css';  import { Link } from 'react-router-dom';  import { toast } from 'react-toastify';  import useAuth from '../utils/hooks/useAuth';  import { useNavigate } from 'react-router-dom';  import Cookies from 'universal-cookie';  const Login = () => {    const { setAuth } = useAuth();    const [userId, setUserId] = useState("");    const [password, setPassword] = useState("");    const navigate = useNavigate();    const cookies = new Cookies();    const handleLogin = (e) => {      e.preventDefault();      fetch(`http://localhost:8080/authentication/login`, {            method: 'POST',            headers: {              'Content-Type': 'application/json',              /\* 'authorization': `Bearer ${auth.accessToken}` \*/            },            body: JSON.stringify({              userId,              password            })          })          .then(response => response.json())          .then(data => {            if (data.error) {              console.log(data.error.code)              toast.error("Server Error!")            }else if(data.result){              setAuth({...data.result, accessToken: data.accessToken})              cookies.set('user',{...data.result, accessToken: data.accessToken})              toast.success("Logged in as " + data.result.name)              navigate("/home")            }else{              toast.warning("Wrong credentials! Try again.")            }          })    }    return (      <div className="login-page">          <div className="login-card">              <div className="login-card-content">                  <div className="header">                      <div className="logo">                          <div><img src="./scc-logo-nobg.png" alt="logo-img"></img></div>                      </div>                      <h2>Silicon City College</h2>                      <h3>Empowerment through Knowledge</h3>                  </div>                  <div className="form">                      <form onSubmit={handleLogin}>                          <div className="form-field username">                              <div className="icon">                                  <i className="fa-sharp fa-solid fa-user"></i>                              </div>                              <input type="text" value={userId} onChange={(e) => setUserId(e.target.value)} placeholder="User ID" required />                          </div>                          <div className="form-field password">                              <div className="icon">                                  <i className="fa-sharp fa-solid fa-lock"></i>                              </div>                              <input type="password" value={password} onChange={(e) => setPassword(e.target.value)} placeholder="Password" required />                          </div>                          <button className="button" type="submit">                              Login                          </button>                          <div>                              Don't have an account? <Link to="/register">Sign Up Now</Link>                          </div>                      </form>                  </div>              </div>              <div className="login-card-footer">                  <Link to="/resetpassword">Forgot password?</Link>              </div>          </div>      </div>    )  }  export default Login;  **./src/pages/Login.css**  .login-page {      width: 100vw;      height: 100vh;      background-image: linear-gradient( #0A2647, #03001C);      background-repeat: no-repeat;      background-attachment: fixed;      margin: 0;      display: flex;      align-items: center;      color: white;      font-family: "Montserrat", sans-serif;      font-size: 14px;      justify-content: center;    }      a {      color: white;      text-decoration: none;      font-weight: bold;      outline: none;      transition: all 0.2s;    }      a:hover,    a:focus {      color: #fdc654;      transition: all 0.2s;    }        .logo img {      width: 100%;      height: 100%;      padding: 18px;      margin: 0;    }    .login-card {      padding: 32px 32px 0;      box-sizing: border-box;      text-align: center;      width: 100%;      display: flex;      height: 100%;      max-height: 740px;      max-width: 350px;      flex-direction: column;    }      .login-card-content {      flex-grow: 2;      justify-content: center;      display: flex;      flex-direction: column;    }      .login-card-footer {      padding: 32px 0;    }        h2 {      font-size: 26px;      margin: 0;      white-space: nowrap;    }      h3 {      color: #d61e2d;      font-size: 14px;      line-height: 18px;      margin: 4px;    }      .header {      margin-bottom: 50px;    }      .logo {      border-radius: 40px;      width: 200px;      height: 200px;      display: flex;      justify-content: center;      margin: 0 auto 16px;      background: rgba(255, 255, 255, 0.1);      align-items: center;    }      .button {      background: white;      display: block;      color: #d61e2d;      width: 100%;      border: none;      border-radius: 40px;      padding: 12px 0;      text-transform: uppercase;      font-weight: bold;      margin-bottom: 32px;      outline: none;    }      .form-field {      margin-bottom: 16px;      width: 100%;      position: relative;    }      .form-field .icon {      position: absolute;      background: white;      color: #d61e2d;      left: 0;      top: -2px;      display: flex;      align-items: center;      height: 100%;      width: 40px;      height: 40px;      justify-content: center;      border-radius: 20px;    }      .form-field .icon:after {      content: "";      display: block;      width: 0;      height: 0;      border: 12px solid transparent;      border-left: 12px solid white;      position: absolute;      top: 8px;      right: -20px;    }      .form-field input {      border: 1px solid rgba(255, 255, 255, 0.2);      text-align: center;      width: 100%;      border-radius: 16px;      height: 36px;      background: rgba(255, 255, 255, 0.1);      color: white;      outline: none;      transition: all 0.2s;    }      .form-field input::placeholder {      color: white;    }      .form-field input:hover,    .form-field input:focus {      background: white;      color: #d61e2d;      transition: all 0.2s;    }      .form-field input:hover::placeholder {      color: #d61e2d;    }    **./src/pages/Register.jsx**  import {React, useState} from 'react';  import { Link, useNavigate } from 'react-router-dom';  import { toast } from 'react-toastify';  import { RegisterPage, Container, SubmitButtonBox, SubmitButton, UserDetails, FormInput, Details, VerifyButtonBox, VerifyButton, FormInputBox, Title, RadioInput, AccTypeTitle, DotOne, DotTwo, Category, AccTypeLabel, LinkButton} from "./Register.style";    const Register = () => {      const [idNumber, setIdNumber] = useState('');      const [fullName, setFullName] = useState('');      const [section, setSection] = useState('');      const [designation, setDesignation] = useState('');      const [gender, setGender] = useState('');      const [email, setEmail] = useState('');      const [phoneNumber, setPhoneNumber] = useState('');      const [password, setPassword] = useState('');      const [confirmPassword, setConfirmPassword] = useState('');      const [accType, setAccType] = useState('');      const navigate = useNavigate();        async function handleSubmit(event){        event.preventDefault();        if(document.querySelector('#readOnlyInput').value === ''){          toast.warning("Fill all input fields!")        }else{          if(password === confirmPassword){            fetch("http://localhost:8080/authentication/register", {              method: "POST",              headers: {                'Content-Type': 'application/json'              },              body: JSON.stringify({                idNumber: idNumber,                name: fullName,                gender: gender,                section: section,                designation: designation,                email: email,                phone: phoneNumber,                password: password,                type: accType              })            })            .then(response => response.json())            .then(data => {                if(data.message) {                  toast.success(data.message)                  navigate("/login")                }                else if(data.err.code === "ER\_DUP\_ENTRY") {                  toast.warning("User with this ID number already exists.")                }else {                  toast.error("There was an error, please try again!")                }            })          }else{            toast.warning("Passwords do not match!")          }        }      }      async function verifyId(){        if(accType) {          fetch(`http://localhost:8080/authentication/verifyid?idNumber=${idNumber}&accType=${accType}`, {            method: 'GET',            headers: {              'Content-Type': 'application/json'            }          })          .then(response => response.json())          .then(data => {            if (data.error) {              toast.error(data.error.code)            }else {              if(data.result[0].section){                setIdNumber(data.result[0].student\_id);                setFullName(data.result[0].name);                setSection(data.result[0].section);                setGender(data.result[0].gender);              }else{                setIdNumber(data.result[0].staff\_id);                setFullName(data.result[0].name);                setDesignation(data.result[0].designation);                setGender(data.result[0].gender);              }            }          })        }else{          toast.warning("Please select an account type.")        }      }        return (        <RegisterPage>          <Container>            <Title>Registration</Title>            <form onSubmit={handleSubmit}>              <div>                <RadioInput type="radio" name="accType" id="dot-1" onChange={(e) => setAccType(e.target.value)} value="student" required checked={accType === 'student'}/>                <RadioInput type="radio" name="accType" id="dot-2" onChange={(e) => setAccType(e.target.value)} value="staff" required checked={accType === 'staff'}/>                <Category>                    <AccTypeLabel for="dot-1">                        <DotOne accType={accType}/>                        <AccTypeTitle>Student</AccTypeTitle>                    </AccTypeLabel>                    <AccTypeLabel for="dot-2">                        <DotTwo accType={accType}/>                        <AccTypeTitle>Staff</AccTypeTitle>                    </AccTypeLabel>                </Category>              </div>              <UserDetails>                <FormInputBox>                  <Details>ID Number</Details>                  <FormInput id="user\_id\_input" type="text" placeholder="R1234567" required onChange={(e) => setIdNumber(e.target.value)} value={idNumber}/>                  <VerifyButtonBox>                    <VerifyButton onClick={verifyId}>Verify</VerifyButton>                  </VerifyButtonBox>                </FormInputBox>                <FormInputBox>                  <Details>Full Name</Details>                  <FormInput type="text" placeholder="E.g: John Smith" required readOnly onChange={(e) => setFullName(e.target.value)} value={fullName}/>                </FormInputBox>                {                  accType === "student" ?                    <FormInputBox>                      <Details>Section</Details>                      <FormInput type="text" id="readOnlyInput" placeholder="E.g: B.Com" required readOnly onChange={(e) => setSection(e.target.value)} value={section}/>                    </FormInputBox> :                    <FormInputBox>                      <Details>Designation</Details>                      <FormInput type="text" id="readOnlyInput" placeholder="E.g: Professor" required readOnly onChange={(e) => setDesignation(e.target.value)} value={designation}/>                    </FormInputBox>                }                <FormInputBox>                  <Details>Gender</Details>                  <FormInput type="text" placeholder="Male/Female" required readOnly onChange={(e) => setGender(e.target.value)} value={gender}/>                </FormInputBox>                <FormInputBox>                  <Details>Email</Details>                  <FormInput type="email" placeholder="johnsmith@email.com" required onChange={(e) => setEmail(e.target.value)} value={email}/>                </FormInputBox>                <FormInputBox>                  <Details>Phone Number</Details>                  <FormInput type="tel" pattern="[0-9]{10}" placeholder="0123456789" required onChange={(e) => setPhoneNumber(e.target.value)} value={phoneNumber}/>                </FormInputBox>                <FormInputBox>                  <Details>Password</Details>                  <FormInput type="password" placeholder="\*\*\*\*\*\*\*\*" required onChange={(e) => setPassword(e.target.value)} value={password}/>                </FormInputBox>                <FormInputBox>                  <Details>Confirm Password</Details>                  <FormInput type="password" placeholder="\*\*\*\*\*\*\*\*" required onChange={(e) => setConfirmPassword(e.target.value)} value={confirmPassword}/>                </FormInputBox>              </UserDetails>              <SubmitButtonBox>                <SubmitButton type="submit" value="Register" />              </SubmitButtonBox>            </form>            <LinkButton><Link to="/">Back to Login</Link></LinkButton>          </Container>        </RegisterPage>    )  }  export default Register;  **./src/pages/Register.style.js**  import styled from "styled-components";  export const RegisterPage = styled.div`      display: flex;      height: 100vh;      width: 100vw;      justify-content: center; /\*center vertically \*/      align-items: center; /\* center horizontally \*/      background: linear-gradient( #0A2647, #03001C);      padding: 10px;  `;  export const Container = styled.div`      max-width: 700px;      width: 100%;      background: #fff;      padding: 25px 30px;      border-radius: 5px;      @media only screen and (max-width: 584px) {          max-width: 100%;      }  `;  export const Title = styled.div`      font-size: 25px;      font-weight: 500;      position: relative;        &::before {          content: "";          position: absolute;          height: 3.5px;          width: 30px;          background: linear-gradient(135deg, #71b7e6, #9b59b6);          left: 0;          bottom: 0;      }  `;  export const RadioInput = styled.input`      display: none;  `;  export const Category = styled.div`      display: flex;      width: 80%;      margin: 15px 0;      justify-content: space-between;      @media only screen and (max-width: 584px) {          width: 100%;      }  `;  export const AccTypeLabel = styled.label`      display: flex;      align-items: center;  `;  export const AccTypeTitle = styled.div`      font-size: 20px;      font-weight: 500;  `;  export const DotOne = styled.span`      height: 18px;      width: 18px;      border-radius: 50%;      margin: 10px;      border: 5px solid transparent;      transition: all 0.3s ease;      border-color: ${props => (props.accType === "student" ? "#d9d9d9" : "none")};      background: ${props => (props.accType === "student" ? "#9b59b6" : "#d9d9d9")};  `;  export const DotTwo = styled.span`      height: 18px;      width: 18px;      border-radius: 50%;      margin: 10px;      border: 5px solid transparent;      transition: all 0.3s ease;      border-color: ${props => (props.accType === "staff" ? "#d9d9d9" : "none")};      background: ${props => (props.accType === "staff" ? "#9b59b6" : "#d9d9d9")};  `  export const UserDetails = styled.div`      display: flex;      flex-wrap: wrap;      justify-content: space-between;      margin: 20px 0 12px 0;      @media only screen and (max-width: 584px) {          max-height: 300px;          overflow-y: scroll;      }      &::-webkit-scrollbar {          width: 0;      }  `;  export const FormInputBox = styled.div`      width: calc(100% / 2 - 20px);      margin-bottom: 15px;      @media only screen and (max-width: 584px) {          width: 100%;      }  `;  export const Details = styled.span`      font-weight: 500;      margin-bottom: 5px;      display: block;  `;  export const FormInput = styled.input`      height: 45px;      width: 100%;      outline: none;      border-radius: 5px;      border: 1px solid #d9d9d9;      padding-left: 15px;      font-size: 16px;      border-bottom-width: 2px;      transition: all 0.3s ease;        &:focus {          border-color: #9b59b6;      }      &:valid {          border-color: #9b59b6;      }  `;  export const VerifyButtonBox = styled.div`      margin: 0;      padding: 6px 100px;  `;  export const VerifyButton = styled.div`      margin: 0;      padding: 4px;      color:#ffffff;      background: linear-gradient(135deg, #71b7e6, #9b59b6);      border-radius: 6px;      &:hover {          background: linear-gradient(-135deg, #71b7e6, #9b59b6);          box-shadow: .2em .2em rgb(108, 108, 108);          transition: all .25s ease-in-out;      }      &:active {          box-shadow: 2px 2px 5px #333;          box-shadow: .05em .05em rgb(77, 77, 77);          transform: translate(.25em, .25em);      }  `;  export const SubmitButtonBox = styled.div`      height: 45px;      margin: 45px 0;  `;  export const SubmitButton = styled.input`      height: 100%;      width: 100%;      outline: none;      color: #fff;      border: none;      font-size: 18px;      font-weight: 500;      border-radius: 5px;      background: linear-gradient(135deg, #71b7e6, #9b59b6);      transition: all 0.3s ease;      &:hover {          background: linear-gradient(-135deg, #71b7e6, #9b59b6);      }  `;  export const LinkButton = styled.div`      color: darkcyan;      &:hover {          color: darkslateblue;      }  `;  **./src/utils/LogoAnimation.jsx**  import styled from "styled-components";  export const RegisterPage = styled.div`      display: flex;      height: 100vh;      width: 100vw;      justify-content: center; /\*center vertically \*/      align-items: center; /\* center horizontally \*/      background: linear-gradient( #0A2647, #03001C);      padding: 10px;  `;  export const Container = styled.div`      max-width: 700px;      width: 100%;      background: #fff;      padding: 25px 30px;      border-radius: 5px;      @media only screen and (max-width: 584px) {          max-width: 100%;      }  `;  export const Title = styled.div`      font-size: 25px;      font-weight: 500;      position: relative;        &::before {          content: "";          position: absolute;          height: 3.5px;          width: 30px;          background: linear-gradient(135deg, #71b7e6, #9b59b6);          left: 0;          bottom: 0;      }  `;  export const RadioInput = styled.input`      display: none;  `;  export const Category = styled.div`      display: flex;      width: 80%;      margin: 15px 0;      justify-content: space-between;      @media only screen and (max-width: 584px) {          width: 100%;      }  `;  export const AccTypeLabel = styled.label`      display: flex;      align-items: center;  `;  export const AccTypeTitle = styled.div`      font-size: 20px;      font-weight: 500;  `;  export const DotOne = styled.span`      height: 18px;      width: 18px;      border-radius: 50%;      margin: 10px;      border: 5px solid transparent;      transition: all 0.3s ease;      border-color: ${props => (props.accType === "student" ? "#d9d9d9" : "none")};      background: ${props => (props.accType === "student" ? "#9b59b6" : "#d9d9d9")};  `;  export const DotTwo = styled.span`      height: 18px;      width: 18px;      border-radius: 50%;      margin: 10px;      border: 5px solid transparent;      transition: all 0.3s ease;      border-color: ${props => (props.accType === "staff" ? "#d9d9d9" : "none")};      background: ${props => (props.accType === "staff" ? "#9b59b6" : "#d9d9d9")};  `  export const UserDetails = styled.div`      display: flex;      flex-wrap: wrap;      justify-content: space-between;      margin: 20px 0 12px 0;      @media only screen and (max-width: 584px) {          max-height: 300px;          overflow-y: scroll;      }      &::-webkit-scrollbar {          width: 0;      }  `;  export const FormInputBox = styled.div`      width: calc(100% / 2 - 20px);      margin-bottom: 15px;      @media only screen and (max-width: 584px) {          width: 100%;      }  `;  export const Details = styled.span`      font-weight: 500;      margin-bottom: 5px;      display: block;  `;  export const FormInput = styled.input`      height: 45px;      width: 100%;      outline: none;      border-radius: 5px;      border: 1px solid #d9d9d9;      padding-left: 15px;      font-size: 16px;      border-bottom-width: 2px;      transition: all 0.3s ease;        &:focus {          border-color: #9b59b6;      }      &:valid {          border-color: #9b59b6;      }  `;  export const VerifyButtonBox = styled.div`      margin: 0;      padding: 6px 100px;  `;  export const VerifyButton = styled.div`      margin: 0;      padding: 4px;      color:#ffffff;      background: linear-gradient(135deg, #71b7e6, #9b59b6);      border-radius: 6px;      &:hover {          background: linear-gradient(-135deg, #71b7e6, #9b59b6);          box-shadow: .2em .2em rgb(108, 108, 108);          transition: all .25s ease-in-out;      }      &:active {          box-shadow: 2px 2px 5px #333;          box-shadow: .05em .05em rgb(77, 77, 77);          transform: translate(.25em, .25em);      }  `;  export const SubmitButtonBox = styled.div`      height: 45px;      margin: 45px 0;  `;  export const SubmitButton = styled.input`      height: 100%;      width: 100%;      outline: none;      color: #fff;      border: none;      font-size: 18px;      font-weight: 500;      border-radius: 5px;      background: linear-gradient(135deg, #71b7e6, #9b59b6);      transition: all 0.3s ease;      &:hover {          background: linear-gradient(-135deg, #71b7e6, #9b59b6);      }  `;  export const LinkButton = styled.div`      color: darkcyan;      &:hover {          color: darkslateblue;      }  `;  **./src/utils/LogoAnimation.css**  #logo{      height: 100vh;      width: 100vw;      display: flex;      flex-direction: column;      align-items: center;      justify-content: center;      opacity:1;      transition: opacity 1.5s ease-in;      background: white;  }  .animate {      font-size: 86px;      font-weight: 900;      text-transform: uppercase;      margin: 100px 0 0;  }  .animate span {      display: inline-block;  }  .animate span:nth-of-type(2) {      animation-delay: .1s;  }  .animate span:nth-of-type(3) {      animation-delay: .2s;  }  .animate span:nth-of-type(4) {      animation-delay: .3s;  }  .animate span:nth-of-type(5) {      animation-delay: .4s;  }  .animate span:nth-of-type(6) {      animation-delay: .5s;  }  .animate span:nth-of-type(7) {      animation-delay: .6s;  }  .animate span:nth-of-type(8) {      animation-delay: .7s;  }  .animate span:nth-of-type(9) {      animation-delay: .8s;  }  .animate span:nth-of-type(10) {      animation-delay: .9s;  }  .animate span:nth-of-type(11) {      animation-delay: 1.0s;  }  .animate span:nth-of-type(12) {      animation-delay: 1.1s;  }  .animate span:nth-of-type(13) {      animation-delay: 1.2s;  }  .animate span:nth-of-type(14) {      animation-delay: 1.3s;  }  .animate span:nth-of-type(15) {      animation-delay: 1.4s;  }  .animate span:nth-of-type(16) {      animation-delay: 1.5s;  }  .animate span:nth-of-type(17) {      animation-delay: 1.6s;  }  .animate span:nth-of-type(18) {      animation-delay: 1.7s;  }  .animate span:nth-of-type(19) {      animation-delay: 1.8s;  }  .animate span:nth-of-type(20) {      animation-delay: 1.9s;  }  .seven span {      color: #1c145f;      opacity: 0;      transform: translate(-150px, 0) scale(.3);      animation: leftRight .5s forwards;  }  @keyframes leftRight {      40% {          transform: translate(50px, 0) scale(.7);          opacity: 1;          color: #c6e4ef;      }      60% {          color: #0f40ba;      }      80% {          transform: translate(0) scale(2);          opacity: 0;      }      100% {          transform: translate(0) scale(1);          opacity: 1;      }  }  img{      width: 250;      height: 250px;      opacity: 1;      transition: opacity 1s ease-in;  }   /\*   For smaller screens \*/  @media (max-width: 600px) {      .animate {          font-size: 24px;          font-weight: 900;          margin: 40px 0 0;          padding-bottom: 30%;      }      img{          width: 180px;          height: 180px;      }  }  **./src/utils/hooks/useAuth.js**  import { useContext } from "react";  import AuthContext from "../../context/AuthProvider";  const useAuth = () => {      return useContext(AuthContext);  }  export default useAuth;  **./src/utils/modals/PopupModal.jsx**  import React from 'react'  import ReactDom from 'react-dom'    const PopupModal = ({ toggleActive, bg, active, children }) => {    if(!active) return null;    const overlay = {      position: "fixed",      top: "0",      bottom: "0",      left: "0",      right: "0",      backgroundColor: "rgba(0,0,0,.8)",      zIndex: "999"  }  const container = {      position: "fixed",      top: "50%",      left: "50%",      transform: "translate(-50%,-50%)",      padding: "5px",      backgroundColor: bg,      zIndex: "1000",      borderRadius: "6px"  }  const closeBtn = {      fontSize: "36px",      color: "orangered",      margin: "8px"  }  const header = {      width: "100%",      display: "inline-flex",      justifyContent: "right"  }  const body = {      textAlign: "center",      padding: "24px"  }    return ReactDom.createPortal(      <div style={overlay}>      <div style={container}>          <div style={header}>              <i class="fa-solid fa-rectangle-xmark" style={closeBtn} onClick={() => toggleActive(({active: false, prompt: ""}))}></i>          </div>          <div style={body}>              {children}          </div>      </div>      </div>,      document.getElementById("portal")    )  }  export default PopupModal Client application **./package.json**  {    "name": "silicon\_app\_server",    "version": "1.0.0",    "description": "",    "main": "index.js",    "scripts": {      "test": "echo \"Error: no test specified\" && exit 1",      "start": "nodemon ./src/index.js"    },    "author": "",    "license": "ISC",    "dependencies": {      "body-parser": "^1.20.1",      "cors": "^2.8.5",      "dotenv": "^16.0.3",      "express": "^4.18.2",      "jsonwebtoken": "^9.0.0",      "mysql": "^2.18.1",      "nodemon": "^2.0.20"    }  }  **./src/index.js**  const express = require('express');  const bodyParser = require('body-parser');  const database = require('./config/database');  const authentication = require('./routes/authentication');  const manageusers = require('./routes/manage-users');  const cors = require('cors');  const app = express();  // parse application/json  app.use(bodyParser.json());  app.use(cors());  app.use(express.json());  // connect to the database  database.connect();  app.use('/authentication', authentication);  app.use('/manage-users', manageusers);  // start the server  app.listen(8080, () => {      console.log('Server listening on port 8080');  });  **./src/config/Database.js**  const mysql = require('mysql');  // create a connection to the SQL database  const database = mysql.createConnection({      host: 'localhost',      user: 'root',      password: 'password',      database: 'silicon\_app\_database'  });  module.exports = database;  **./src/routes/Authentication.js**  const express = require('express');  const database = require('../config/database');  const router = express.Router();  const jwt = require('jsonwebtoken');  require('dotenv').config();                                     /\* user registration \*/  router.post('/register', (req, res) => {      const { idNumber, name, gender, section, designation, email, phone, password, type } = req.body;      // insert the data into the database      if(type === "student"){          database.query(              'INSERT INTO registered\_students (student\_id, name, section, gender, email, phone, password) VALUES (?, ?, ?, ?, ?, ?, ?)',              [ idNumber, name, section, gender, email, phone, password ],              (err, results) => {                  if (err) {                      console.log(err);                      res.status(500).send({ err });                  } else {                      res.status(200).send({ message: 'User registered successfully' });                  }              }          );      }else{          database.query(              'INSERT INTO registered\_staffs (staff\_id, name, designation, gender, email, phone, password) VALUES (?, ?, ?, ?, ?, ?, ?)',              [ idNumber, name, designation, gender, email, phone, password ],              (error, results) => {                  if (error) {                      console.log(error);                      res.status(500).json({ error });                  }else {                      console.log('User registered successfully');                      res.status(200).json({ message: 'User registered successfully' });                  }              }          );      }      });  router.get('/verifyid', (req,res) => {      const idNumber = req.query.idNumber;      const accType = req.query.accType;      if(accType === "student"){          database.query(`SELECT \* FROM all\_students WHERE student\_id = '${idNumber}'`,(err, result) => {              if(err){                  res.status(500).json({ error: err });              }else if(result.length === 0){                  res.status(404).json({ error: {code: 'ID number not found in database'} });              }else{                  res.send({ result });              }          });      }else{          database.query(`SELECT \* FROM all\_staffs WHERE staff\_id = ${idNumber}`,(err, result) => {              if(err){                  res.status(500).json({ error: err });              }else if(result.length === 0){                  res.status(404).json({ error: {code: 'ID number not found in database'} });              }else{                  res.send({ result });              }          });      }  });                                 /\*  user login \*/  router.post('/login', (req,res) => {      const {userId, password} = req.body;      //search for the user in database      console.log(req.body)      database.query(`          SELECT \*          FROM          (SELECT \* FROM registered\_students          WHERE student\_id = '${userId}' AND password = '${password}') AS student          UNION          SELECT \*          FROM          (SELECT \* FROM registered\_staffs          WHERE staff\_id = '${userId}' AND password = '${password}') AS staff          LIMIT 1      `,(err, result) =>{          let data = result[0] ? result[0] : null          if(err){              res.status(500).json({ error: err });          }else if(data === null){              res.status(401).json("Not Authorized");          }else {              const userId = data.student\_id || data.staff\_id              const accessToken = jwt.sign({ id: userId }, process.env.ACCESS\_TOKEN\_SECRET\_CODE)              console.log(result[0])              res.send({ result: result[0], accessToken: accessToken });          }      })  })  function verifyToken(req,res,next) {      console.log(req)      const authHeader = req.headers.authorization;      if(authHeader){          const token = authHeader.split(" ")[1];          jwt.verify(token, process.env.ACCESS\_TOKEN\_SECRET\_CODE, (err, user) => {              if(err){                  console.log(err);                  return res.status(403).json("Token is not valid!");              }              req.user = user              next()          })      }else{          res.status(401).json("You are not authenticated!");      }  }                                          /\* ADMIN ROUTES \*/  router.post('/login/admin', (req,res) => {      const {userId, password} = req.body;        console.log(req.body)      database.query(`          SELECT \*          FROM admins          WHERE admin\_id = '${userId}' AND password = '${password}'      `,(err, result) =>{          console.log(result[0])          if(err){              res.status(500).json({ error: err });          }if(!result[0]){              res.status(400).json("User not authorized");          }else{              res.status(200).json(result[0]);          }      })  })    module.exports = router; 8. CONCLUSION In conclusion, the development of a college grievance application can significantly enhance the quality of student life on campus. The application provides students with an accessible and convenient platform to voice their grievances and concerns and seek appropriate solutions. It streamlines the process of addressing student complaints and helps ensure that the college administration is aware of issues affecting students' well-being. This project can bring about positive changes that can benefit both the students and the college as a whole. By facilitating communication between students and the administration, the college grievance application can promote a more harmonious and productive campus environment. Overall, the successful implementation of this project has the potential to create a more inclusive and supportive college community. 8. REFERENCES  * [**React js documentation**](https://react.dev/learn) * [**HTML documentation**](https://developer.mozilla.org/en-US/docs/Web/HTML) * [**CSS documentation**](https://developer.mozilla.org/en-US/docs/Web/CSS) * [**Node js documentation**](https://nodejs.org/en/docs) * [**Express js documentation**](https://expressjs.com/en/guide/routing.html) * [**MySQL documentation**](https://docs.oracle.com/en-us/iaas/mysql-database/doc/getting-started.html) | |  |
|  |  |  |  |