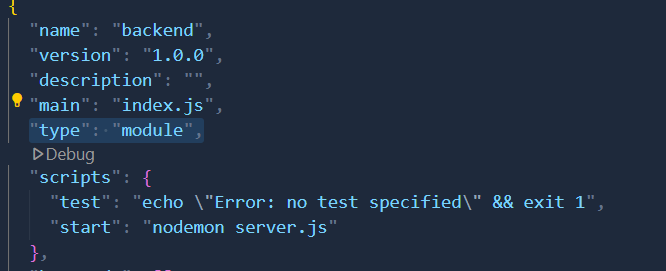
**Backend**

* npm init -y
* npm install express mysql nodemon cookie-parser bcrypt jsonwebtoken cors

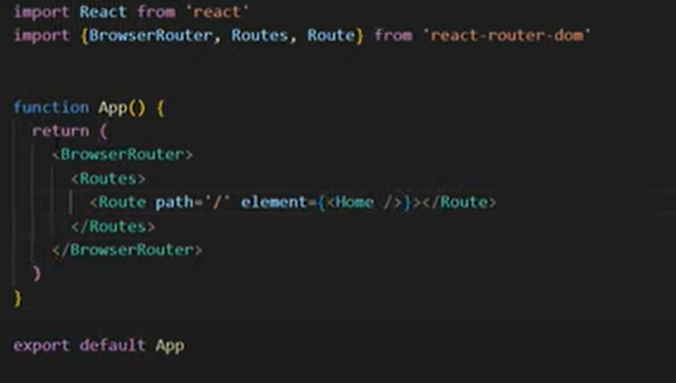
**( mekedi mysql wenuwat mysql2 install krann. mysql eke default authentication eka support kranne caching\_sha2\_password eka node eka support kran na. e nisa mysql2 install kran. [your MySQL user is using an authentication method (like caching\_sha2\_password) that the Node.js mysql library doesn’t support by default. This is common when using MySQL 8.x with older MySQL clients like mysql Node.js package.])**

* create server.js file
* set the scrpit -> "start" : "nodemon server.js"
* set the "type": "module", in side the package.json file
* 

**Frontend**

* npm create vite@latest . --template react ( select framework – react, Select a variant – JavaScript )
* npm install bootstrap axios react-router-dom

**Normal configuration tika hadala backend app eka start krann ( server.js eke import + create app+ use app + data base connection + listen app )**

* Then Create the basic setup in side the front-end
* **Create Register.jsx file**
* **Import the browser router and set the route in App.jsx**
* Then set the route for register page
* <Route path="/register" element={<Register />}></Route>
* Import krann path eka udin
* Run the front-end ( npm run dev )
* CSS load wenne naththam hariyat, main.jsx ekat bootstrap import krann

***import 'bootstrap/dist/css/bootstrap.min.css';***

* CSS wala aulk enwa nam meka ain krala danna

***import './index.css'***

* create the login page and set the route
* **page athara navigate wenn ‘Link ‘ eka import krann**
* import { Link } from 'react-router-dom'
* **then use, when click the create account button it should navigate to the sign-up page ( button ekak nemei Link ekak dann / methan use kranne APP.jsx eke use krapu routes )**
* <Link to="/register" className='btn btn-default border w-100 bg-light rounded-0 text-decoration-none'>Create Account</Link>
* **Propaty ekak thiyawna meke ‘to’ kiyal eken set krann navigate wenn oni page eka**
* **Register krann yanakot eya enter krana data tika api varibl ekak store kran idal, sign-up click kram back end ekat yawann oni. Data tika store kra gann useState ekak use kranwa.**

**const [values, setValues] = useState({**

**name: "",**

**email: "",**

**password: ""**

**});**

**Methana name,email, password widihat ganne input eke thiayan name kiyana eka athule Thiyan variable eka**

<div className="mb-3">

            <label htmlFor="name"><strong>Name</strong></label>

            <input

              type="text"

              placeholder="Enter Name"

              name="name"

              className="form-control rounded-0"

            />

</div>

* **Then we have to set the onchange function input field ekat set krann**

**onChange={e => setValues({ ...values, name: e.target.value })}**

methan setValues kiyanne useState ekat use krapu variable eka.

(…values) kiyanne input eke previous value eka Thiyan thanata name eke thiyana aluth value eka set krann

* **Form ekat ‘ onSubmit ‘ ekak set krann. Ekat set kran function eka useState ekat Yatin implement ktann. Default submit wen eka prevent krann**.

***<form onSubmit={handleSubmit}>***

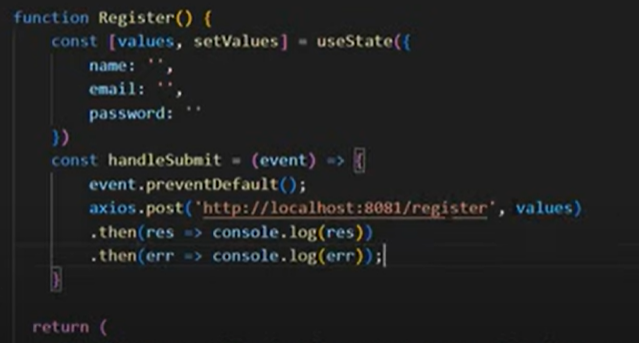
* Implementation of ***handleSubmit***

***Const handleSubmit = (event) =>{***

***Event.preventDefault();***

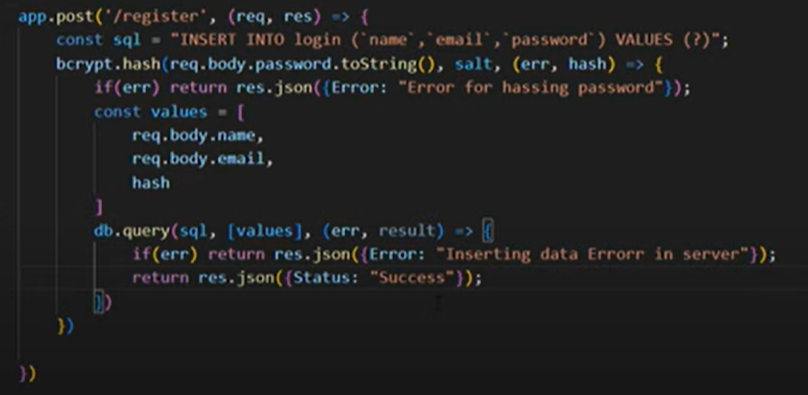
***// methan API end point eka call krann***

***}***



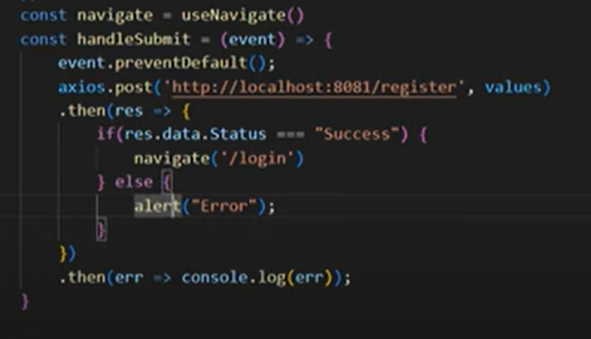
* **Back end UPL ekai end point ekai. E end point ekat useState eke thiyana values tika post krnna. Result eka print krann. Error ekak nam eka print krann**
* **Then create this API endpoint inside the backend server.js**
* *//register API*
* app.post("/register", (req, res )=>{
* const sql = "INSERT INTO login (`name`, `email`, `password`) VALUES (?, ?, ?)";
* })
* **Then set the salt value inside the server.js**

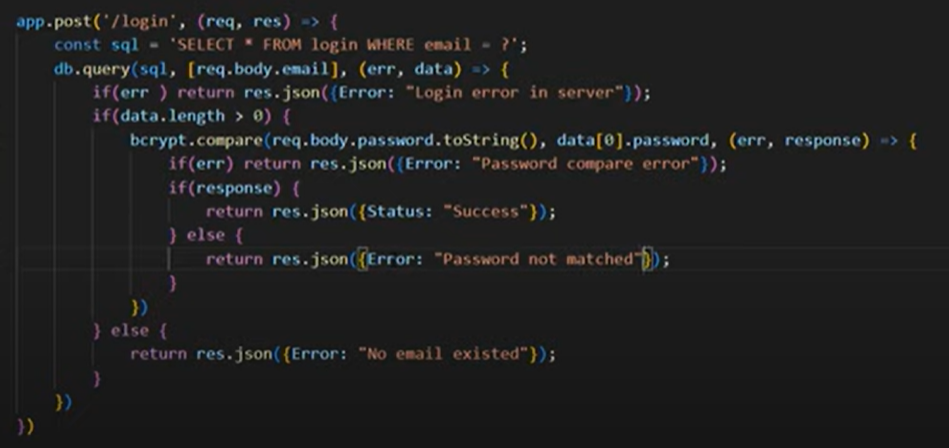
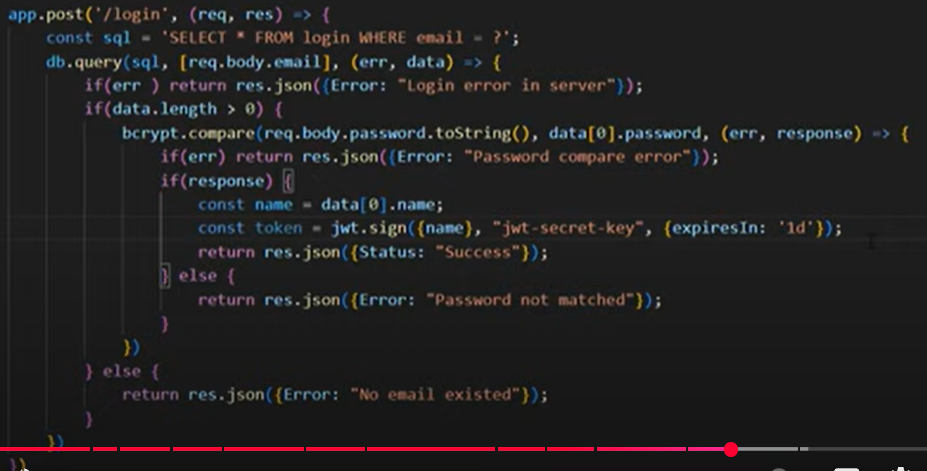
**const salt =10;**



* **Sign up eka success unama login ekat navigate wenna. useNavigate function eka use kranwa. Page walat navigate wenna. React-router-dom walin dena ekak.**

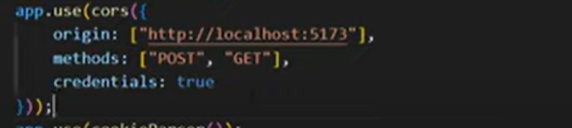
***const navigate = useNavigate()***



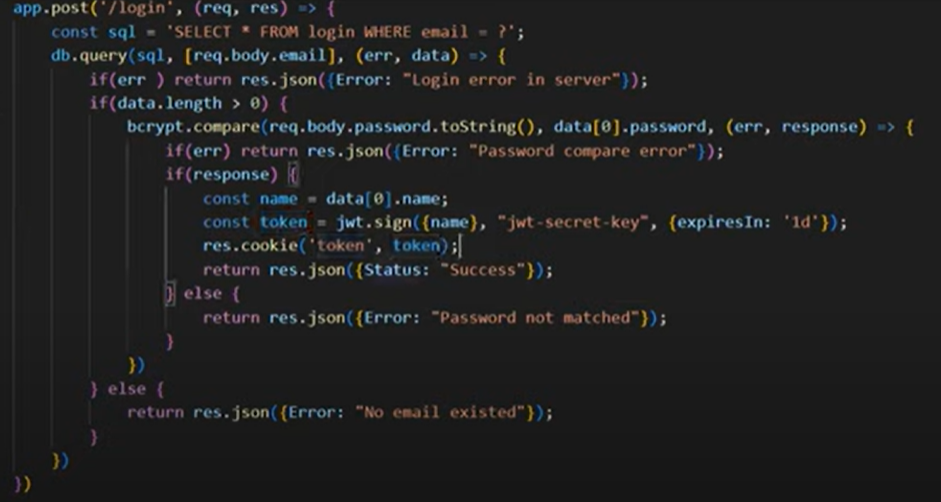
* **Sign-in eka hadnkot register ekat hadapu useState, handleSubmit, onchnage ekai copy kran dann. Signin ekedi name eka ganne na. e nisa useStae eken name eka ain krann. API endpoint eka change krann, Login widihat. Eka backend eke hadanna. Register eke wagem from eketh onSubmit function eka dann**
* **Create the login endpoint in the server.js**
* **Then generate the token and store in the cookie. ( jsonwebtoken and cookie-parsor )**

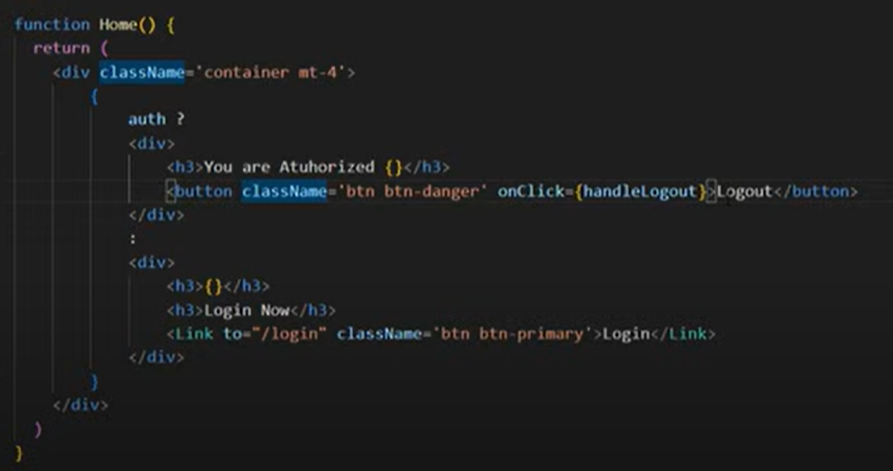
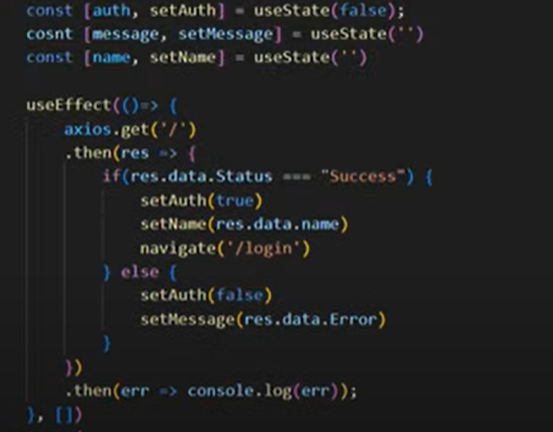
**Successfully log unata passe token ekak generate kranwa. Eka cookie ekak widihata save krnwa. Expire time eka 1 day meke dila thiyenne**

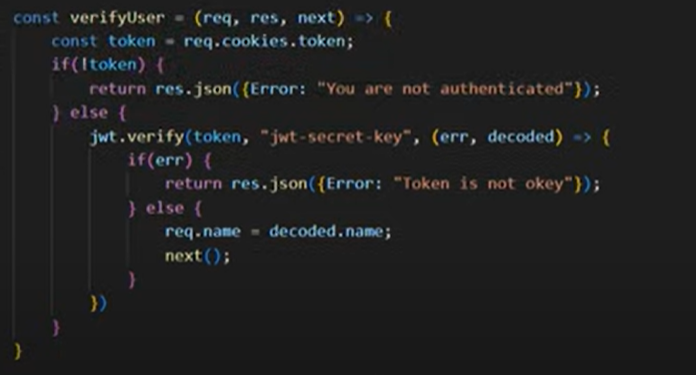
* **Cors ekath update krann cookie walin**

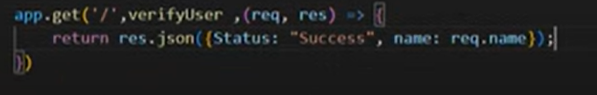


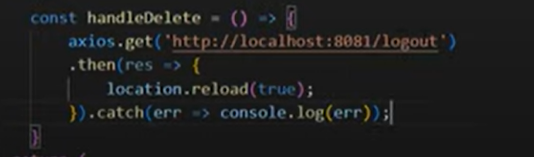
* **Login front page eketh credential set krann**
* **Generate krana token eka cookie ekak widihat store kranwa browser eke. Eka application tab eke cookie wala thaiynwa**



* **Log una kenekt wen message ekak pennai, ehema nathi kenekat wena ekak pennai set kramu.( log wel kiyala dan ganne eyata token ekak thiynwa cookie wala store wela)**
* **Meke ((( auth ? ))) kiyala check kranne, log welad kiyala wela nam eke athule tika wenne. Ehema naththam ithuru tika wenne**
* **Home eke check krala balnwa, log welad nadd kiyala. useState eka default auth eke false. E kiyanne log wela na. it passe useEffect ekak dala loggin eka set kranwa. E wagem log wena kenage name eka aran useState ekak save kragannwa. useEffect eken*****check one is authorize or not ( token eka browser eke cookei eke thiynwad nadd kiyala check kranwa )***
* 
* **Then set the verify use function in the server.js. generate krapu token ekath ekk cookie eka balnwa . *"jwt-secret-key"----meka private key eka pahaldi use krapu token eka generate karnna. Name eka kalin encrypt krapu eka mekedi dcrypt krala gannwa***



* **useEffect eke authorization eka check krann gahpu route eka.**
* **Then logout. Its mean delete the cookie in the browser. Home.jsx eke**



* **Set the API for the clear cookie ( logout ) in the server.js**