## jwt

* Npm I express cors pg jsonwebtoken bcrypt
* user\_id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

this will generate a random combination of strings to uniquely identify each users. The uuid\_generate\_v4 is the function used to generate the unique combo string id.

* Salt to decrypt the password. genSalt is the function does salt the password
* Npm I dotenv - to store the jwt token (.env)
* Utils folder has the function to generate jwt token to each user.

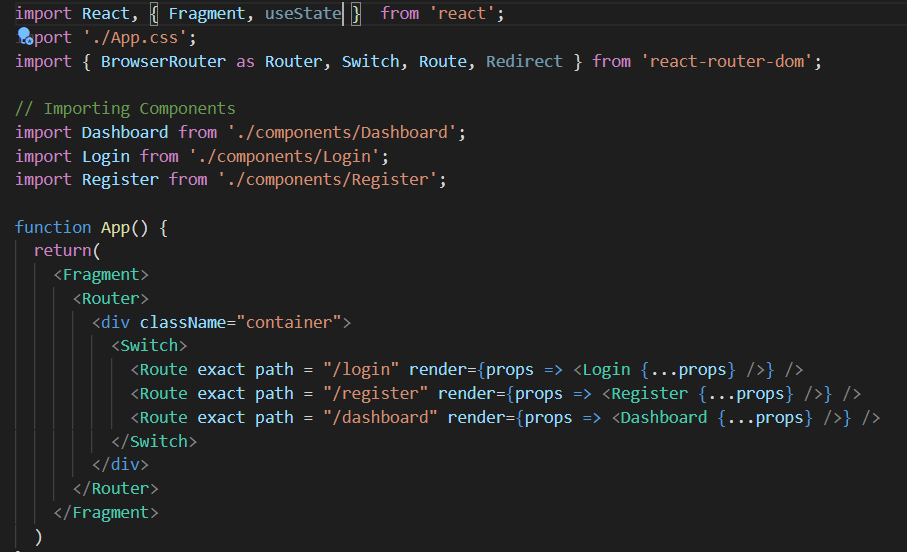
## Postgres

Command prompt controls

* Psql -U username - to login in to the postgres in command prompt
* \l - to list all the databases
* \c - to use one database
* Create extension if not exists “uuid-ossp”; to use the uuid function in the postgres

## React

* Npx install create-react-app app-name
* Npm install react-router-dom - to route between components(navigate)
* Npm install react-toastify - to send push notification (alert)
* This is how we add routes other components



<button onClick={() => setAuth(true)}>Authenticate</button> // login.js

<Route  //app.js

            exact

            path="/login"

            render={props =>

              !isAuthenticated ? /\* if 'isAuth' is not true go to Login, else go to dashboard \*/

              (<Login {...props} setAuth={setAuth} />) : (<Redirect to = "/dashboard"/>)}/>

const [isAuthenticated, setIsAuthenticated] = useState(false);

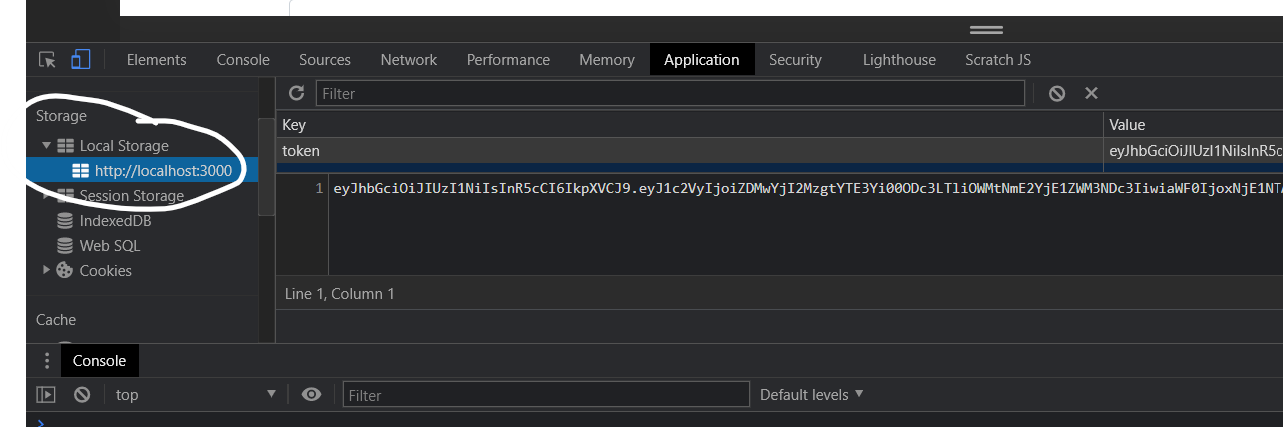
  const setAuth = (boolean) => {

    setIsAuthenticated(boolean); //setIsAuth toggles the isAuth value from F to T

  };

* This is how we toggle states.
* Though we set the value to true, when we refresh we get logged out, cuz the system resets and goes to default value which is false

Localstorage:



<Link to="/register" >Register</Link>

* <a> in react