**How to connect Frontend to Backend:**

1) Create .env file and store the URL in a environment variable => REACT\_APP\_BACKEND=http://localhost:8000/api/

2) Make another file like Backend.js and in that file store that environment variable into a normal variable and export it

=> var API = process.env.REACT\_APP\_BACKEND;

3) Import the "API" variable into the main file : => import { API } from "../backend";

4) use console.log to print the variable on the Frontend => console.log("API IS", API);

5) Make sure the DB is connected

**Main file which is being rendered is:** Home.js and Base.js (main)

* In Base.js, we have used something called as “children”, what this does is that it makes Base component act as an enclosing parent tag. So, when we use this Base tag <Base> in Home.js, all the code written inside the <Base> tag will be imported to Base.js in “children” variable.
* So, wherever this “children” is used, it will render the code in Home.js file under the Base tag

**Routes in Menu.js:** In the navigation bar, routes are mentioned which will be used to redirect the user to certain pages e.g.: /**signin**, /**signout for signin and signout pages.**

These links are written under the Link tag <Link> under “to=””. To make all these given paths compatible and working, all we need to do is to export the Menu component using “withRouter from react-router-dom”. This makes it compatible and will allow it to fetch all the mentioned routes from Routes.js

* export default withRouter(Menu)

Menu.js is imported in Base.js because we want the Menu bar to be visible everywhere.

**Nav Bar styling in Menu.js:**

const currentTab =  (history, path) => {

    if(history.location.pathname===path){

        console.log(history.location.pathname, path)

        return {color:"#FFFFFF"}

    } else {

        return({color:"#F4C724"})

    }

}

<Link style={currentTab(history,"/cart")} className="nav-link" to="/cart">

                    Cart

                </Link>

History is given to us by Link from React-router-dom.

The **History** interface allows manipulation of the browser session history, that is the pages visited in the tab or frame that the current page is loaded in.

History.location.pathnam =>

location - (object) The current location. May have the following properties:

* pathname - (string) The path of the URL

For link to cart page:

History.location.pathname will return = “/cart”

And path is what we are passing in the method call of “currentTab” inside the link tag after the history => “/cart”

* history.push(“/”) in Index.js for signout method.

push(path, [state]) - (function) Pushes a new entry onto the history stack, this means that the path mentioned in history.push(“/”) which is route to Home page will be loaded.

**Route.js:**

<Route path="/signup" exact component={Signup}/>

Redirect to which page Is done by the path mentioned “/signup” and which component is to be loaded on that redirected page is obtained by “exact component={Signup}”.

The keyword **“exact”** makes sure that only the mentioned component is loaded and not any other component.

A common use of JSON is to exchange data to/from a web server. When sending data to a web server, the data has to be a string. Convert a JavaScript object into a string with JSON.stringify().

where to store data coming from the form before submitting data to DB -> state

when somebody types something in form and we want to get exactly what he is typing -> handlechanges the hit submit -> submitting what user entered and getting a response from backend