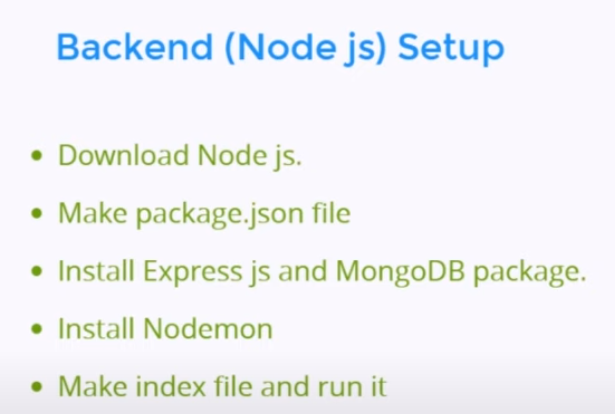
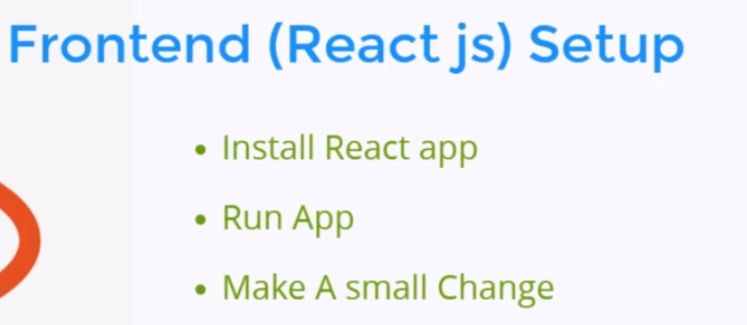


# First Video:- Setting up backend

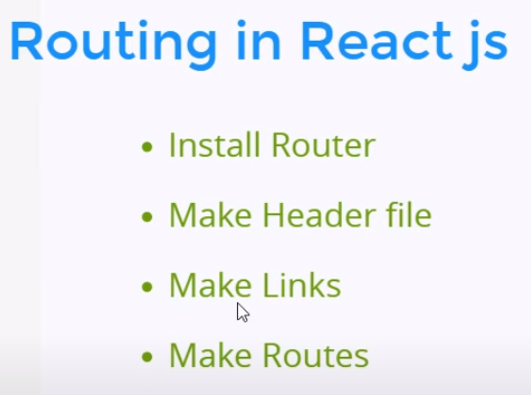


Install nodemon, express and mongoose

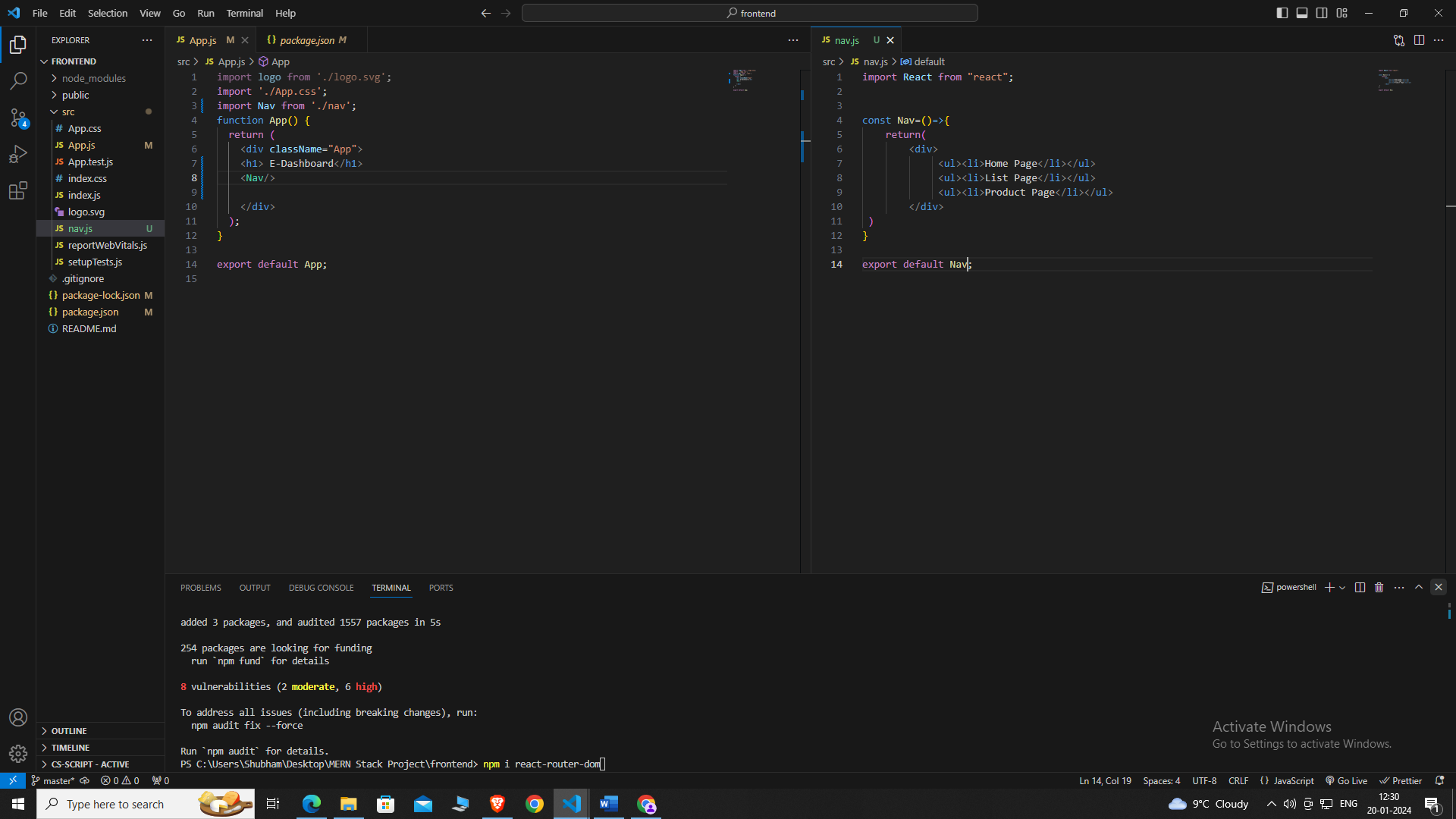
* npm install nodemon express mongoose



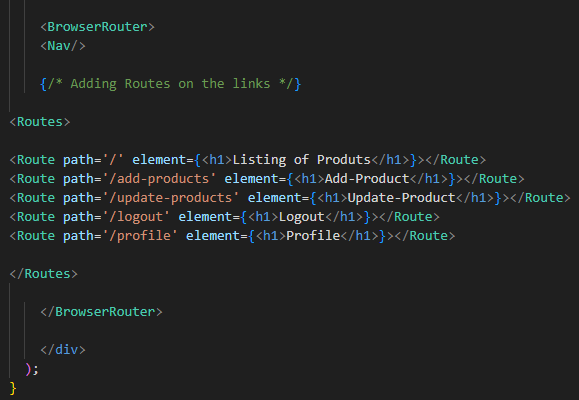
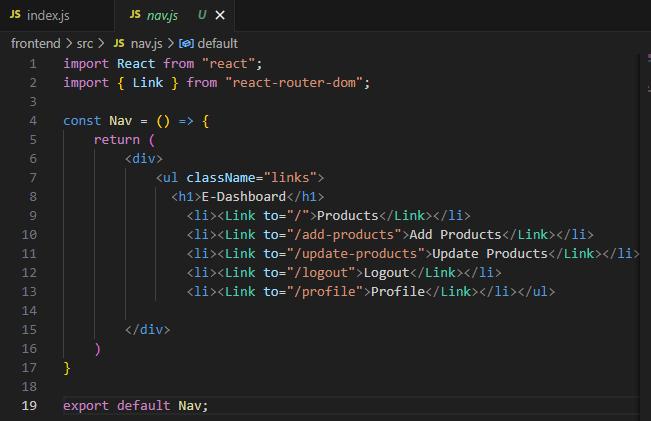
Run npx create-react-app folder\_name

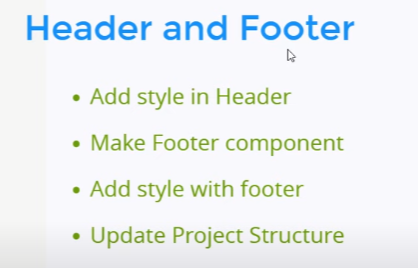


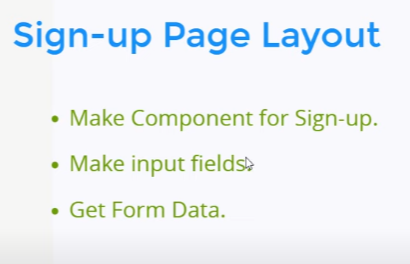
Installing router: npm i react-router-dom

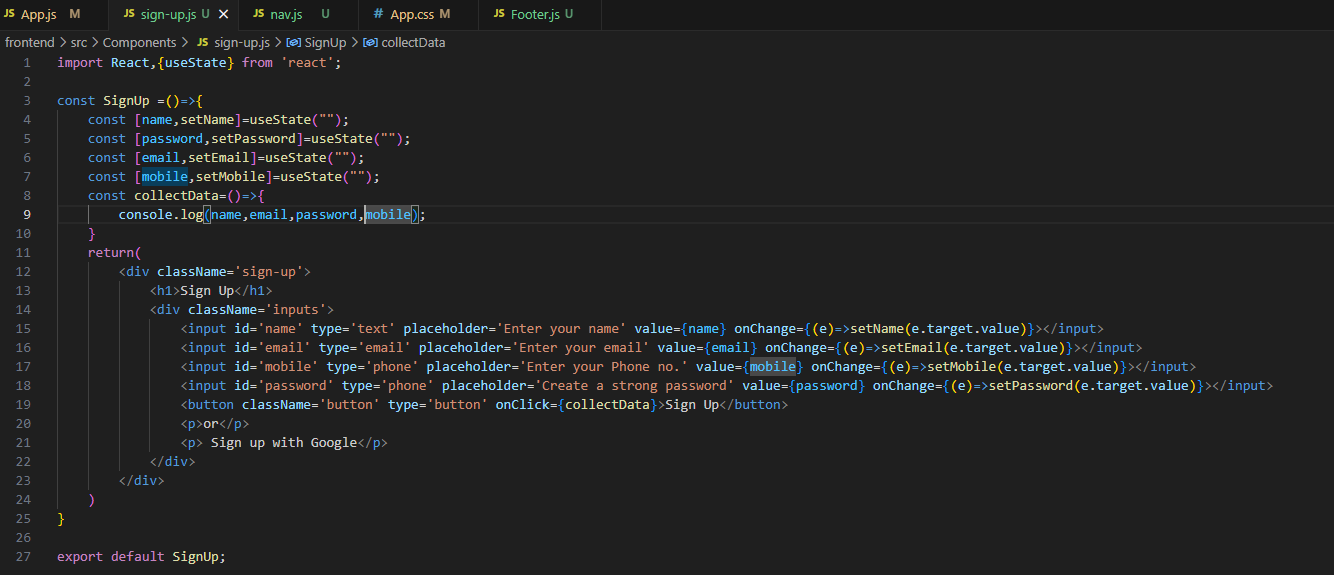


After creating a nav component and importing that into app.js now it is the time to add routing on the links using our router dom







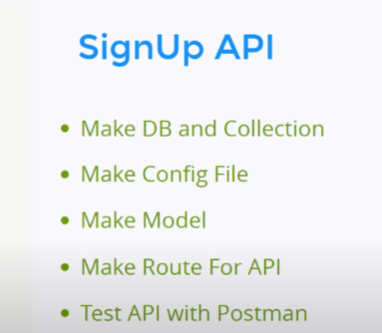


Here is a module used for getting the data of input field which is useState.

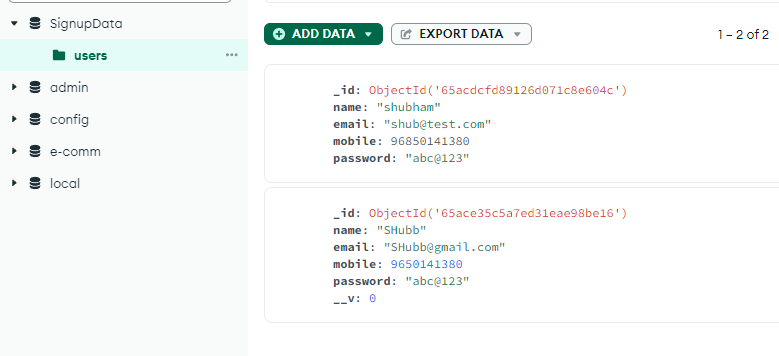


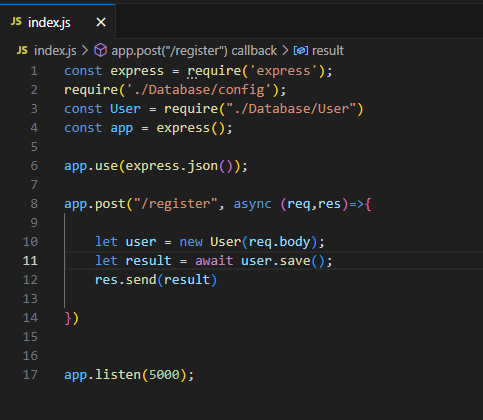
Install mongoose

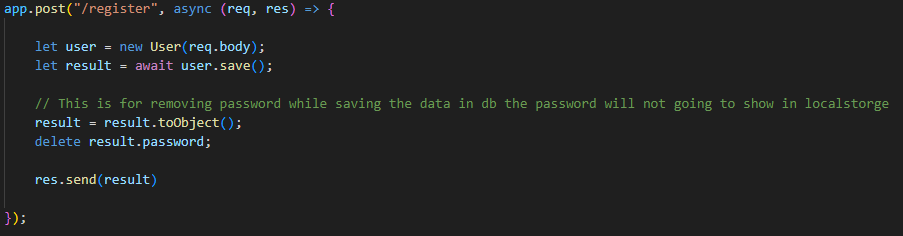




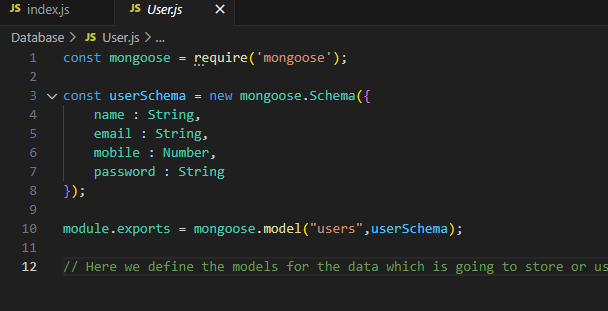
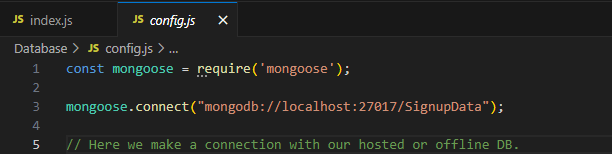
Create a new DB and a collection







We add something which is .toObject() this is to remove password from localstorage.





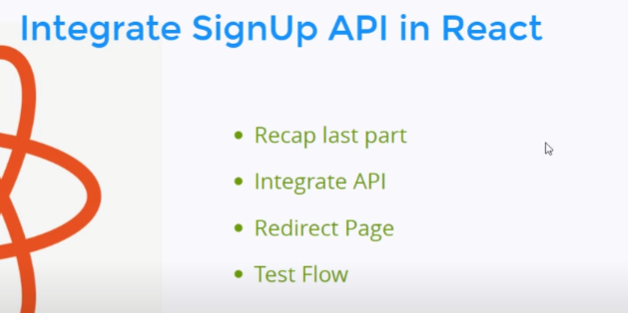
This can be a interview question. This is a backend issue.

Install npm i cors

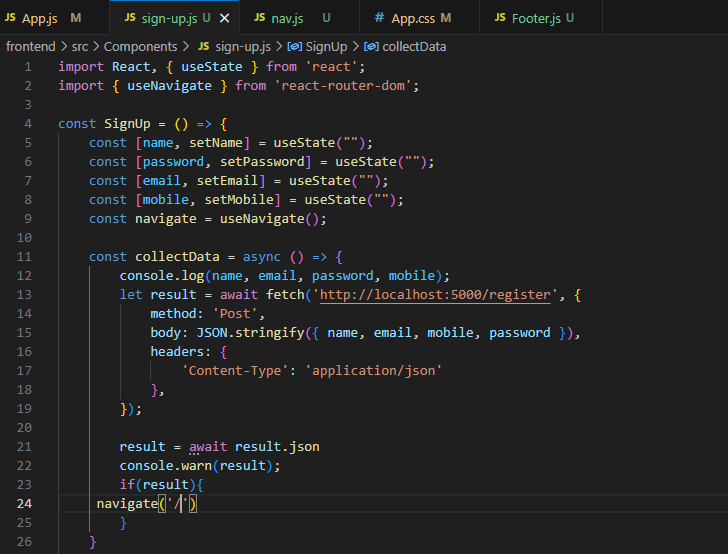




Install cors and import cors then use it as cors.



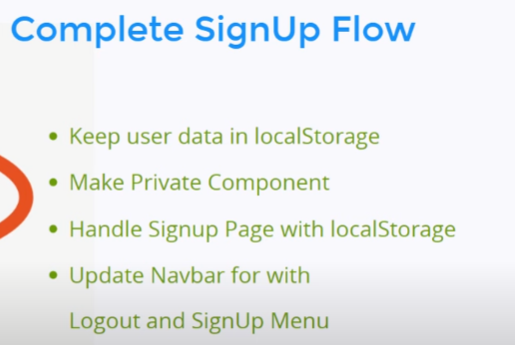
In frontend part in signup.js component we create an api to store the data on db.



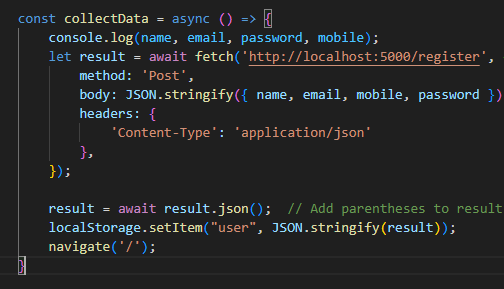
In the function collectData() 🡪

From line 13 that is the api used to store the user data in DB.

And when everthing is fine we will redirecting the user on home page for doing this we are using useNavigate which is a function used to redirecting.



First let’s store the data in localstorage:-



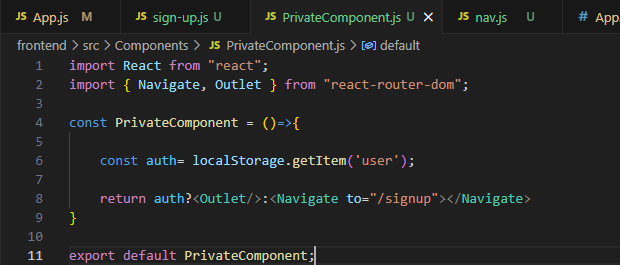
        localStorage.setItem("user", JSON.stringify(result));

This helps us to store the input field data into local storage.

**Q. What is private Components?**

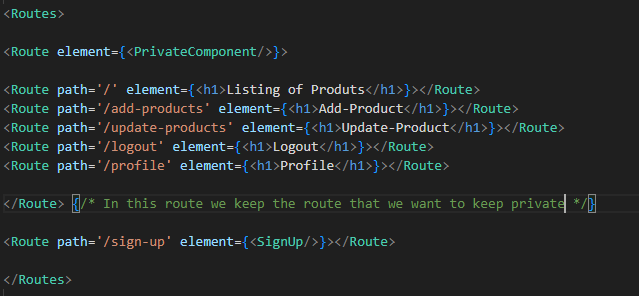
Ans. Private components are those components which is protected by any unauthorised access until or unless any certain condition passed. Like we don’t want that any user can access our product, update page or any other page without login or signup so that page or component called private component.

Now to apply or make the route Private we need to create privatecomponent.js file (name can be anything as per your convenience).



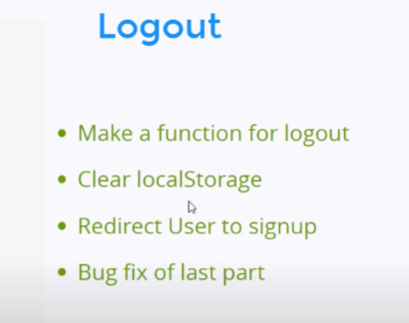
Now Import this into App.js



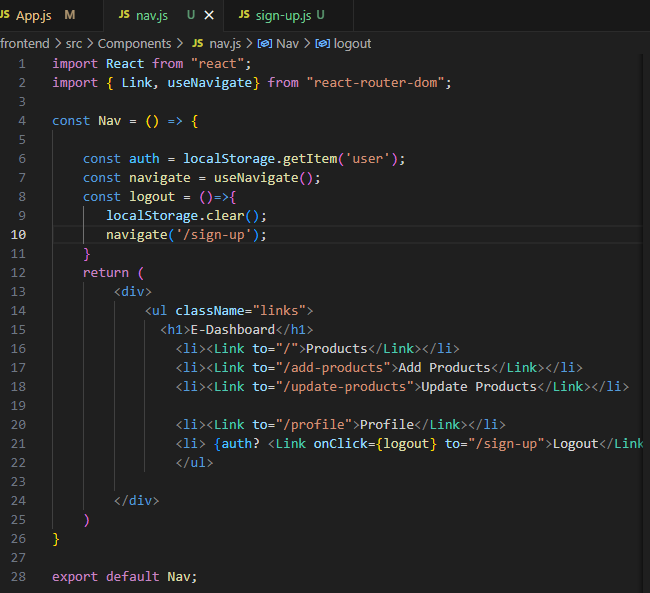


Then make a new route instance where PrivateComponent function will be place and take all the components that should be private in that routes.

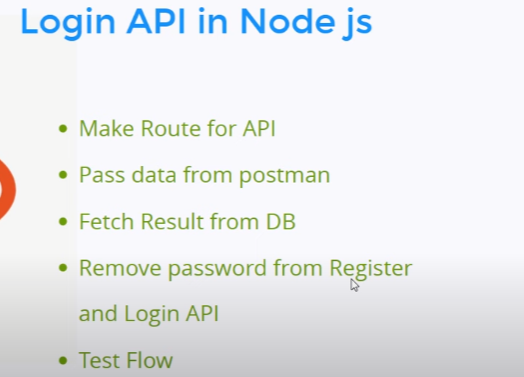
This is how we will create private components now no one can access that components without signup or login

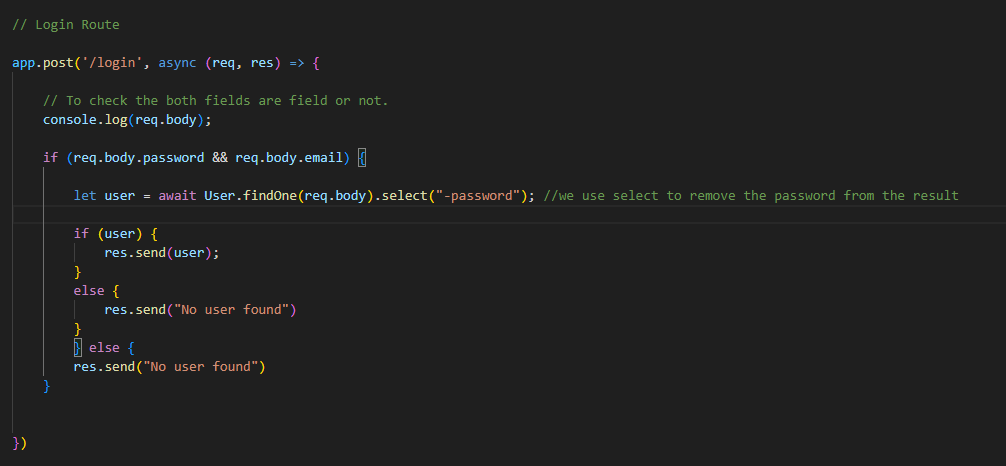


To implement logout feature go to nav.js make a function which usually clears the data on localstorage.



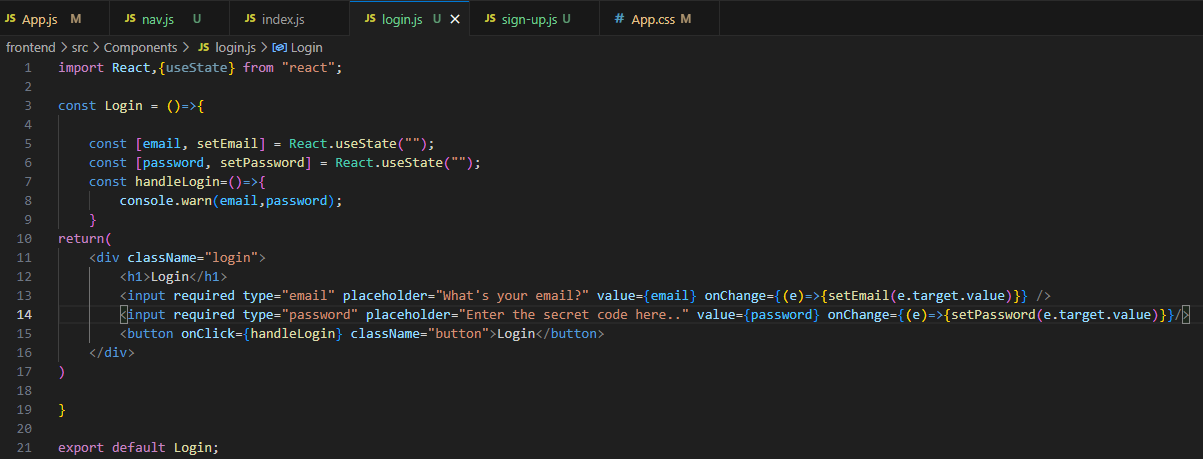
As, shown in line no. 8 there is a function named logout.



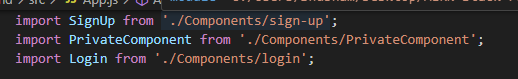


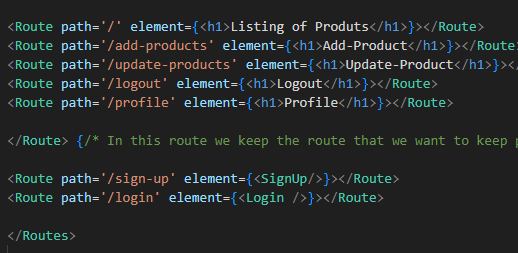
This is the api for login there is something in line no. 33 we have use select(“-password”) because of removing password while logging in the pass will shown on localstorage.

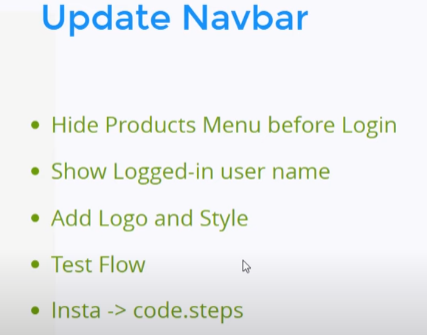
This is the login form:

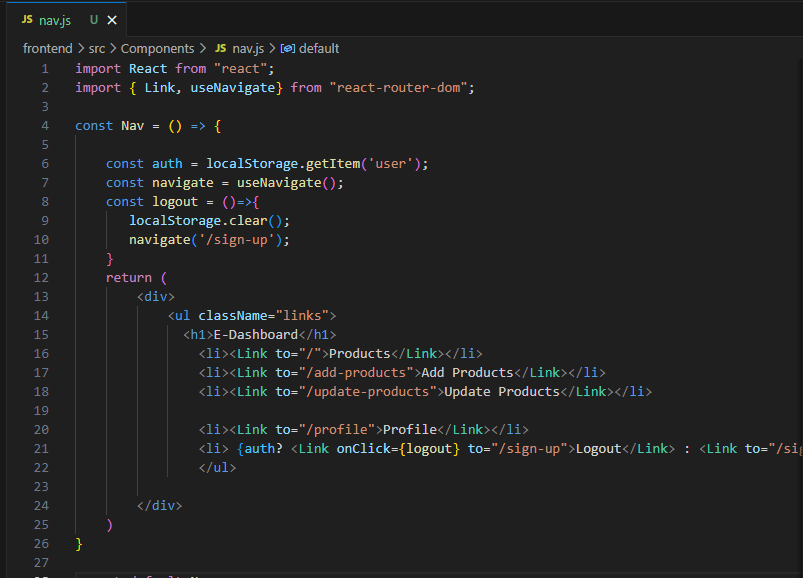


Import this into app.js



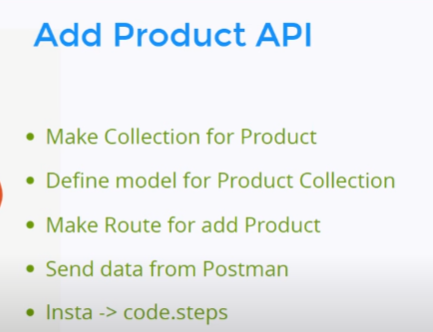
 Add a route

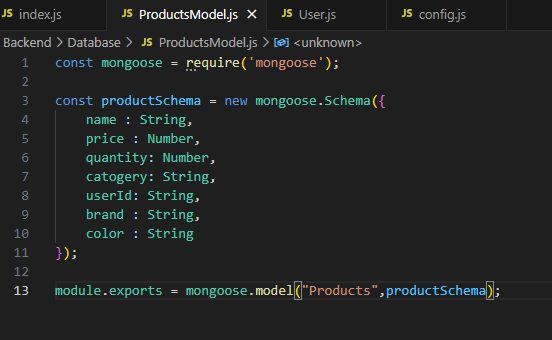




Before updation navbar.

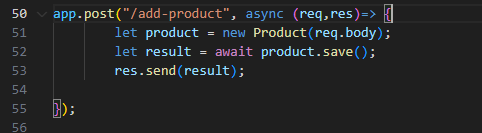
this is after updation





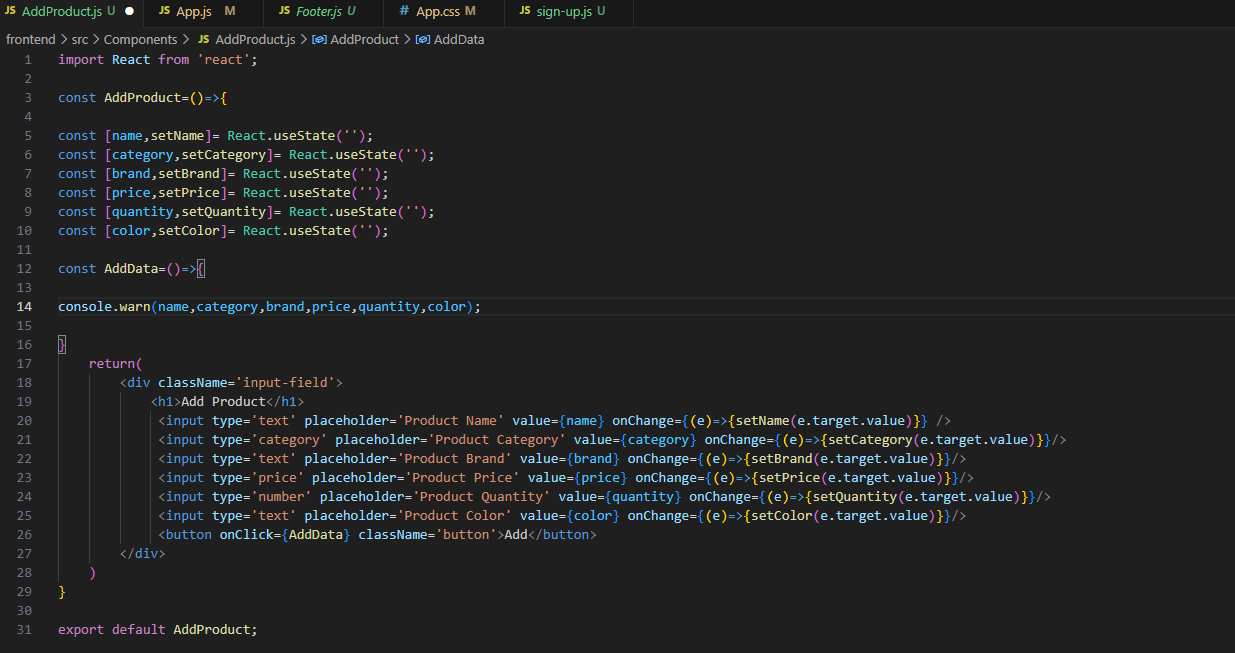
This is our product schema.

Make new route in index.js file



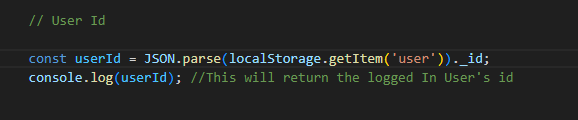
Note: Make sure to restart the project completely with mongodbcompass.

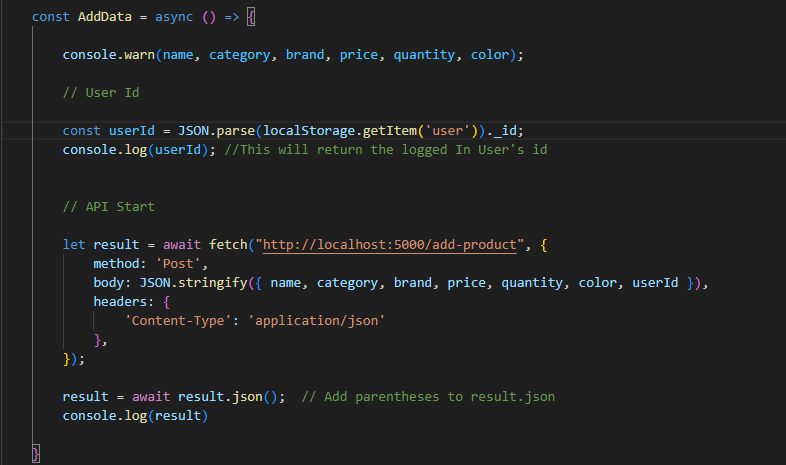


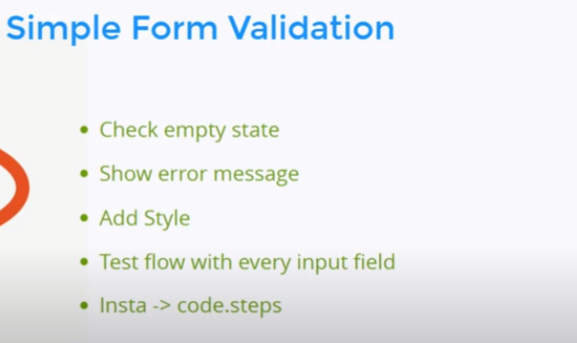


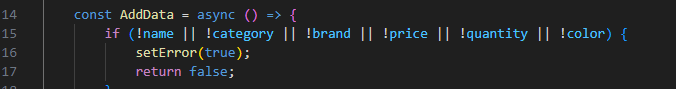
Import this into App.js

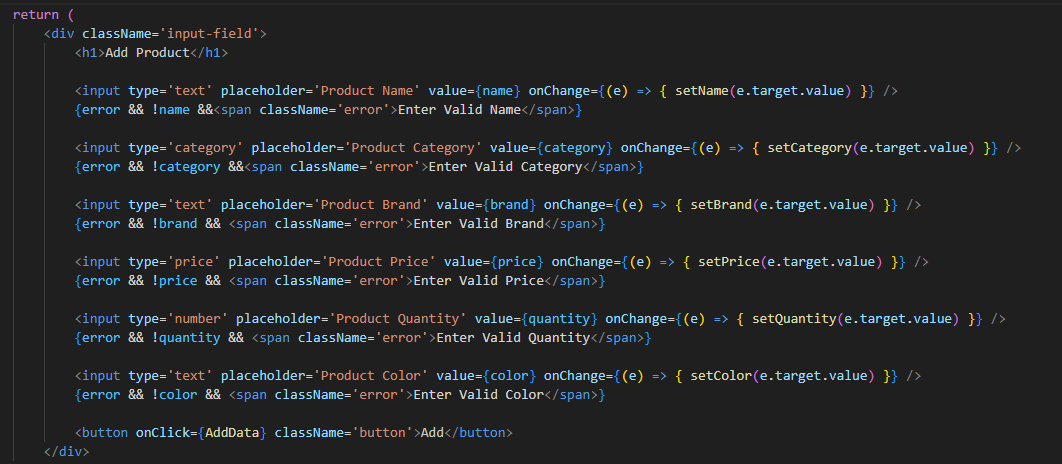


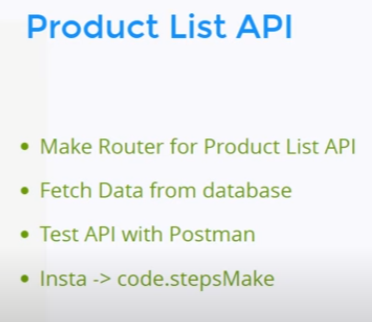
There is catch that the product is going to be added in db is adding by an user to remark that whose user is adding that product will be identified by the id. So, we will write some code to identify the user 

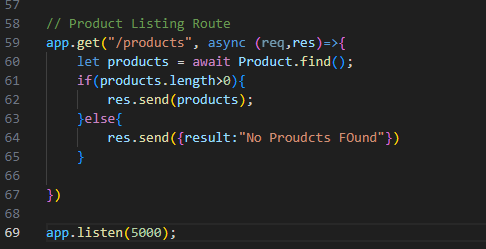


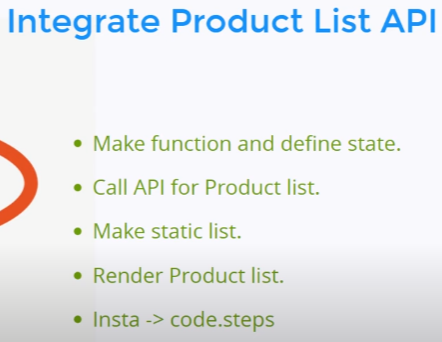






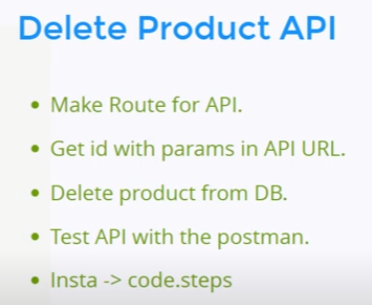


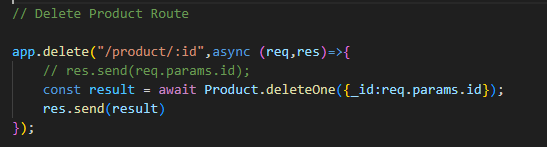


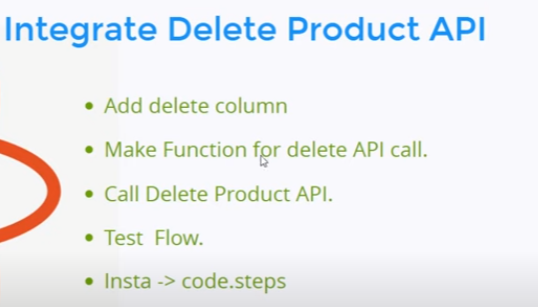


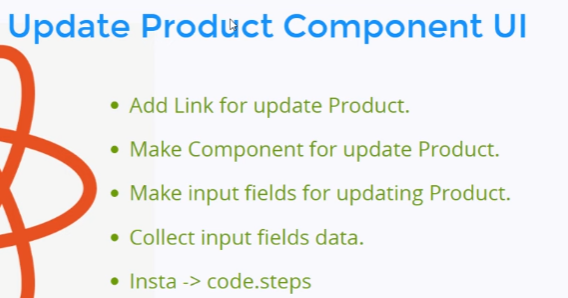


This is the api and the integration of products listing page.

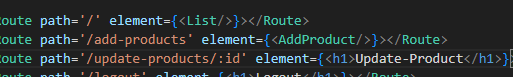




This is the api for deleting products. 



To make updates in our products we have to fetch the exact data of that product which will be possible using ID. To identify the id we have to pass id parameter through the link like in app.js

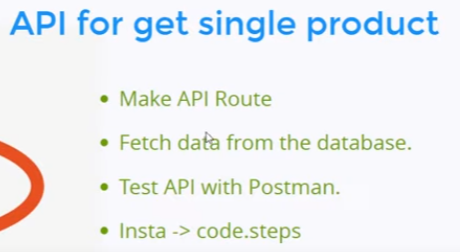


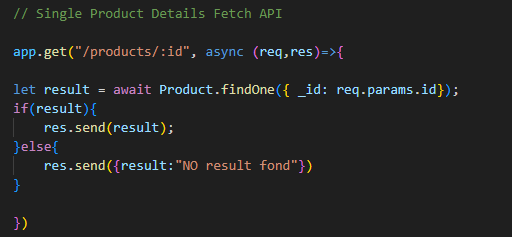
In, Product Listing page we have to make the id dynamic so when we click on any product updation button it will fetch the details of that product.



We can also use template to make it dynamic.

Check the UpdateProduct.js file for the code





Now Integration-

