**Tip: Try to create reusable components**

User: niwatesaurabh pass: ecommerce

mongodb+srv://niwatesaurabh:<db\_password>@ecommerce.ueo8d.mongodb.net/

mongodb+srv://niwatesaurabh:ecommerce@ecommerce.ueo8d.mongodb.net/

**Planning:**

-Before login show the body page, product details page even cart page also but when user click on checkout if not user login then show popup of login or register.

-if user not logged in then add product in session storage and when user logged in add the all products in cart [ first add in database create endpoint and then fetch the data]. Add product in cart enable option of add multiple product in cart at once.

Frontend:

-Login [via email or phone which register]

-Sign up [register fields like Name, phone, email, password and verify send otp to user on email]

-Forgot password [via email]

-**for not logged in user check on which route user going and stored in session storage and when user login in redirect him to that route where he is last time**

-Cart Logic

-Give option of add address to user before checkout [give option of edit, delete]

-In profile section give option to user update their details like email id or mobile no.

But when update email id o mobile no again verify by otp.

-Give the option of choosing payment like Cart, UPI id, COD. [Implement Payment Integration].

-Show the successful orders.

-implement pagination for products and data [frontend and backend]

**Frontend Routes:**

/app.jsx file contain Navbar, footer and Outlet

/login login route

/signup sign up page

/forgotPassword forgot password page

/ Home page <App> in App Navbar is constant and then use Outlet so Childrens routes [ Body, /product/:prodId, /cart, /wishlist, /update address/:id, /userProfile,

]

/support or /contact

/payment

**Schemas:**

**User Schema:**

FirstName

LastName

Profile Picture [ not while login but give option there in while updating address]

Email

Password

Address [Country, State, City, landmark, Pincode, Phone no]

[in feature add google authentication by google via firebase]

User: /api/v1/user [/login, /register, /logout, /forgotPassword]

 **Non-logged-in user**:

* **Add to cart**: Store the product in **session storage**.

 **If the user immediately logs in**:

* Fetch the products from **session storage**.
* Add those products to the **database** for persistence.
* After storing the products in the database, fetch the updated cart data from the **backend** (since the cart is now managed in the database).

 **Logged-in user**:

* **Add to cart**: Directly store the product in the **database**.
* Fetch the cart data from the **backend** to keep the frontend updated.
* **Adding a Product to the Cart**: **POST** /cart, **GET** /cart, **PUT** /cart/:productid, **PUT** /cart/:productid

<https://youtu.be/sR5Z8AJ-zRU?si=ZB_Wt4xW3sPvW8Rr> -> react caraousel videos

Edge case: merge the session storage product with actual **challenging**

const router = createBrowserRouter([

{

path: "/",

element: <App />,

children: [

{

index: true, // Default route for "/"

element: <Body />,

},

{

path: "home",

element: <Body />,

},

{

path: "cart",

element: <Cart />,

},

],

},

{

path: "/login",

element: <Login />,

},

{

path: "/register",

element: <Register />,

},

{

path: "/forgotpassword",

element: <ForgotPassword />,

},

]);