npm install react-router-dom

in the App.js we provide

    <BrowserRouter>

      <div>Content for all pages</div>

      <Routes>

        <Route path="homepage" element={<Home />}>

         <Route path="about" element={<About />} />

      </Routes>

    </BrowserRouter>

  );

To create a link (btn) to a component

      <Link to="about">

        About

      </Link>

To create shared component

    <BrowserRouter>

      <Routes>

        <Route path="/homepage" element={<SharedLayout />}>

          <Route index element={<Home />}></Route> // index will match path specified in the SharedLayout (“/homepage”)

          <Route path="about" element={<About />} />

          <Route path="dashboard" element={<Dashboard />} />

          <Route path="login" element={<Login />} />

          <Route path="products" element={<Products />} />

          <Route path="\*" element={<Error />} />

        </Route>

      </Routes>

    </BrowserRouter>

const SharedLayout = () => {

  return (

    <Fragment>

      <StyledNavbar /> // Shared component for all specified components

      <Outlet /> // specific component that will be shown depending on the path.

    </Fragment>

  );

};

To set unique ID path

<Route path="products/:productId" element={<SingleProduct />} /> // at this point any path after /products/ will get us to SingleProduct page

const SingleProduct = () => {

  const { productId } = useParams(); // to get the product id (path after /products/)

  const selectedProduct = products.find((prod) => prod.id === productId);

  const handleRenderProduct = () => {

    if (selectedProduct) {

      return (

        <div>Product name</div>

      );

    } else {

      return <h2>Product not found</h2>;

    }

  };

To navigate to certain path (programmatically, not using <Link ></Link>).

  const navigate = useNavigate();

  const handleSubmit = async (e) => {

    e.preventDefault();

    navigate("/homepage/dashboard");

  };

To set up protected route

          <Route

            path="dashboard"

            element={

              <ProtectedRoute> // This Component will be shown instead of Dashboard

                <Dashboard user={user} />

              </ProtectedRoute>

            }

          />

const ProtectedRoute = ({ children, user }) => {

  if (!user) return <Navigate to="/homepage/login" />; // check certain conditions, in our case we check if user exists, if does not, we navigate to other page

  return children; // if user exist, we navigate to the page which was initially protected, Dashboard in our case.

};