!.in react the navigation b/t 2 pages is achvied with react-router-dom library

Npm I react-router-dom – version – 6

1.configuration setup

Import {createBrowserRoute} from ‘react-router’

Const approuter = createBrowerRouter([

{

Path:’/’,

Element:<App/>,

errorElement:</ErrorElement/> ..> when we hit some page other than /

}])

2. we need to provide the configuration to react with router

<RouterProvider router={approuter}/> 🡪 this gives access to all route in above createBrowser

3. we have a hook from react router that is used to get the error status of page

Import {useRouteError} from react-router;

Const error = useRouteError ();

Console.log(err)

When we need to keep the header alone static and on clicking different page we need to render the pages alone how we can do??

1.We can use concept of children route

const appRouter = createBrowserRouter([

  {

    path: "/",

    element: <AppLayout />,

    children: [

      {

        path: "/",

        element: <Body />,

      },

      {

        path: "/about",

        element: <About />,

      },

      {

        path: "/contact",

        element: <ContactUs />,

      },

    ],

    errorElement: <Error />,

  },

]);

Here Applayout is main route and others are sub children of it ..

How to render children inside Applayour .. we have component called **outlet** in react-router

const AppLayout = () => {

  return (

    <div className="app">

      <Header />

      <Outlet />

    </div>

  );

};

When using react and to navigate to other page we should not use <a href=””> why?

**It will reload the entire page 🡪 this will lead to performance issue**

To solve this we use

**<Link to=’/contact’>contact</Link>**

When we use link the page is not refreshed this is called single page application .. the components are just replaced and not entire page is reloaded

There are 2 types of routing in web app:

1.**client side routing** 🡪 we don’t do any server call and all the components already exist.. new page is not fetched . . why single page application .. we implement route with client side routing

2**.server side routing** 🡪 it reloads the entire page and makes network call and gets the entire page from server.. The html is coming from server

Dynamic routing

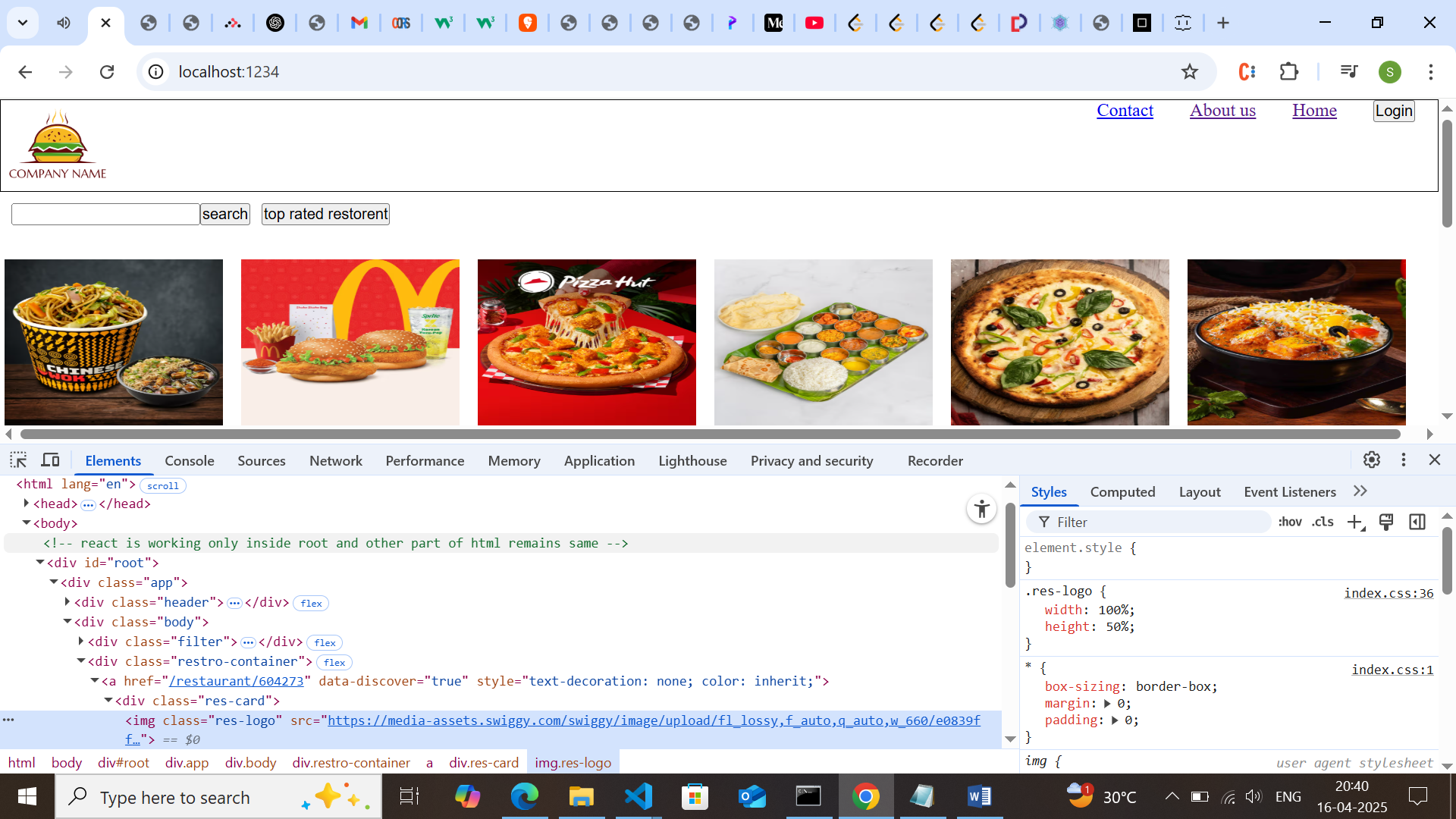
1.we have dynamic route for each page when we click a list

To read params from /:id we use useParams hook from react-router

Const {id} = useParams()

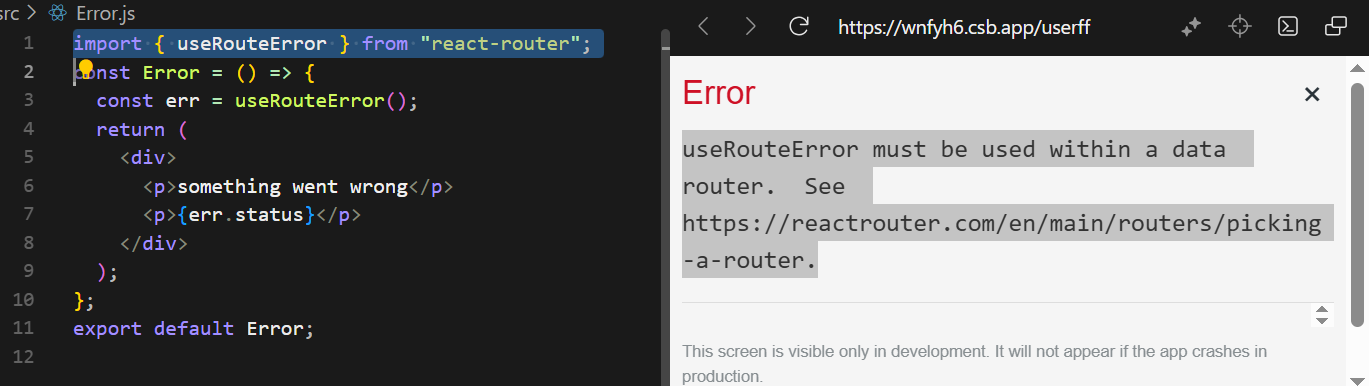
**Link is component from router dom and link is using <a href=””> behind the screen**

**In html it will be considered as <a href=’’>**



There are 3 different ways the react router dom renders routes in ui

* Data mode(createBrowserRouter mentioned above) 🡪 it is most modern method which supports errorElements , useRouteError hook to get router specific error , loader etc
* Declarative approach (browserRouters,Routes,Route) 🡪 used when we need to render route directly in jsx and used in single page applications but for big app it will be messy to handle route and child routes
  + It does support special errorElement props or useRouteError
  + The 404 error of route is handled wit path=’\*’



  <BrowserRouter>

      <Routes>

        <Route path="/" element={<App />} />

        <Route path="/user" element={<User />}>

          {/\* act as default route for /user \*/}

          <Route index element={<IndexUser />} />

          {/\* nested route \*/}

          <Route path="child" element={<Child />} />

        </Route>

        {/\* catch 404 pages  \*/}

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

      </Routes>

    </BrowserRouter>

* Framework mode 🡪 it is not mostly used it defines how framework like nextjs works with react route