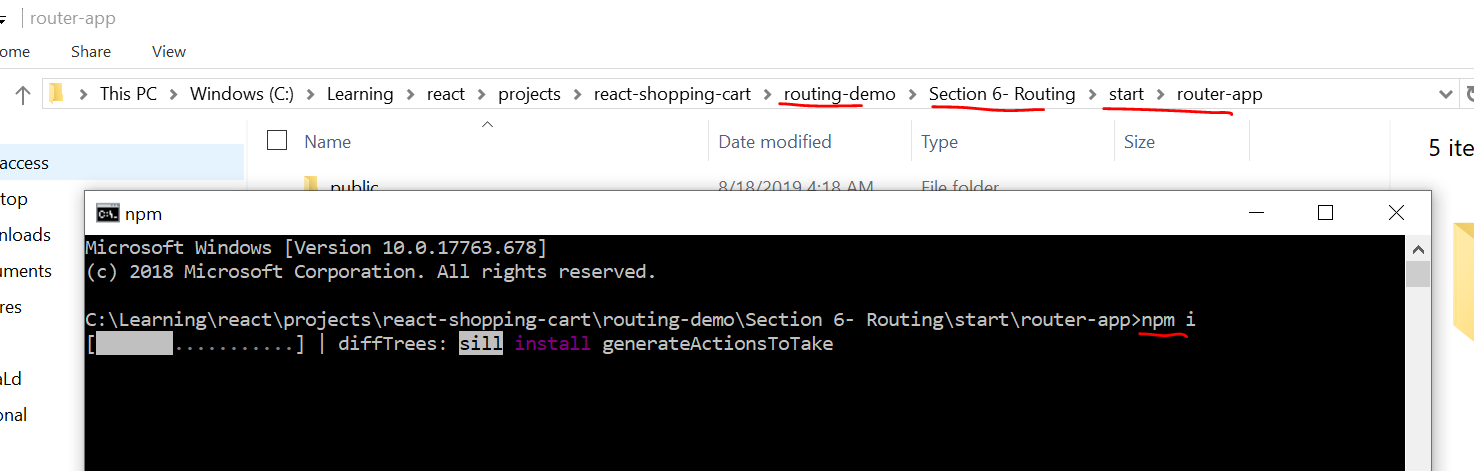


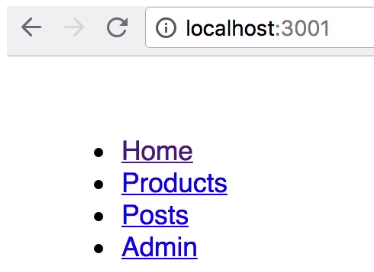
Download Resources.zip from <https://github.com/ashokmahdev/react-shopping-cart>

After downloading the project , go to below project folder( \react-shopping-cart\routing-demo\Section 6- Routing\start\router-app ) and do ***npm i*** to install project dependencies and run ***npm start*** to start the project



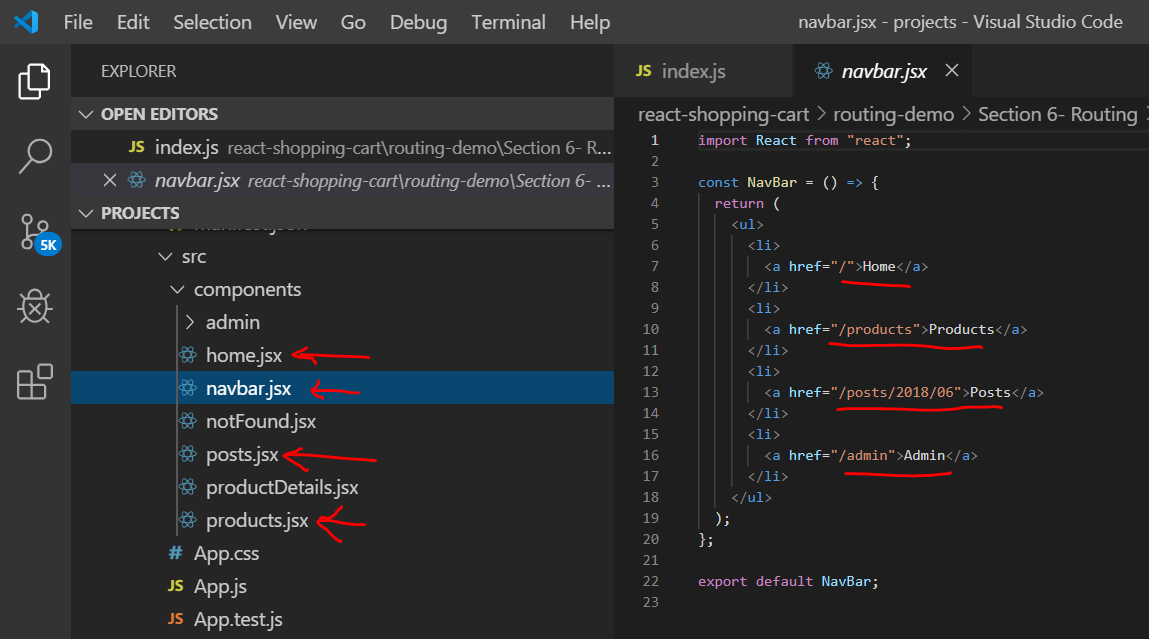
You will see below projects started with few components

We will work on this project to understand routing concepts, there are 4 components



Click on links to see that no routing to components have been implemented



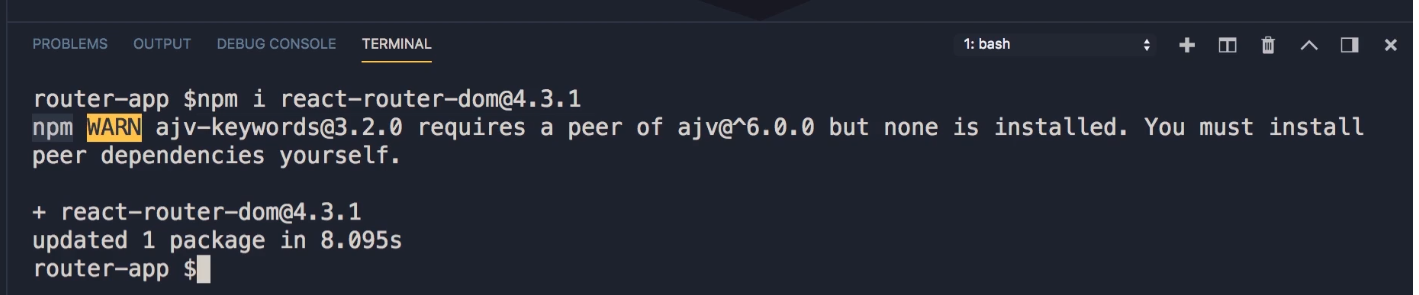


**React is just a simple library and only responsible for rendering view, it will not have a routing module like angular as it is not a framework.**

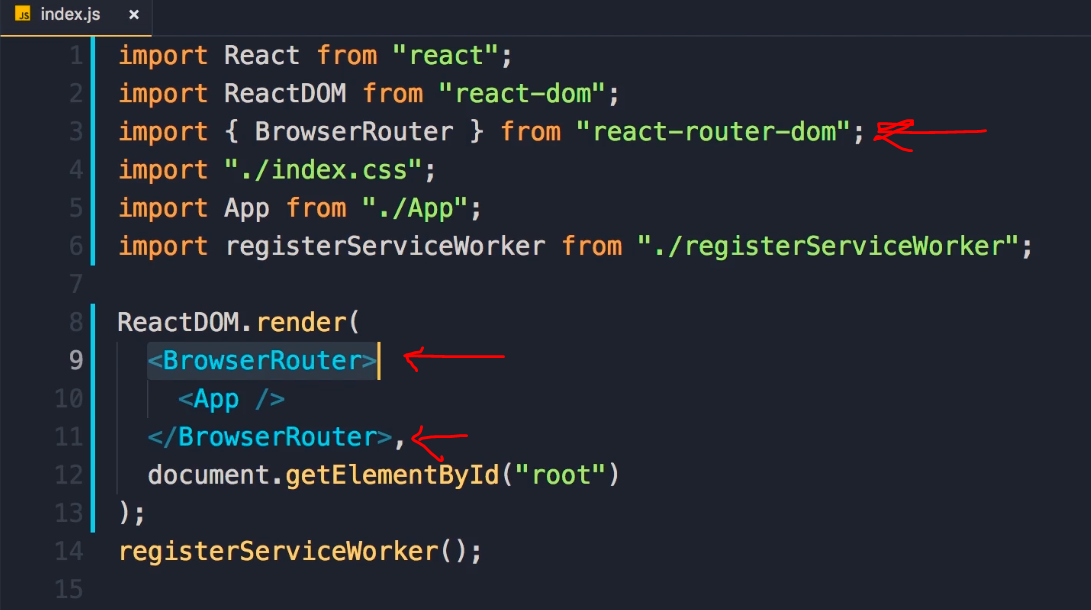
Give examples of other frameworks like .NET core and angular

To achieve routing, we need to install react routing library

Install react-router-dom package



Now to achieve routing 2 steps we need follow

1. Make below changes to index.js
   1. warp app component with browser Router component
   2. Now browser history object will be available for entire component tree in our project
   3. 
   4. Code

import { BrowserRouter } from "react-router-dom";

ReactDOM.render(

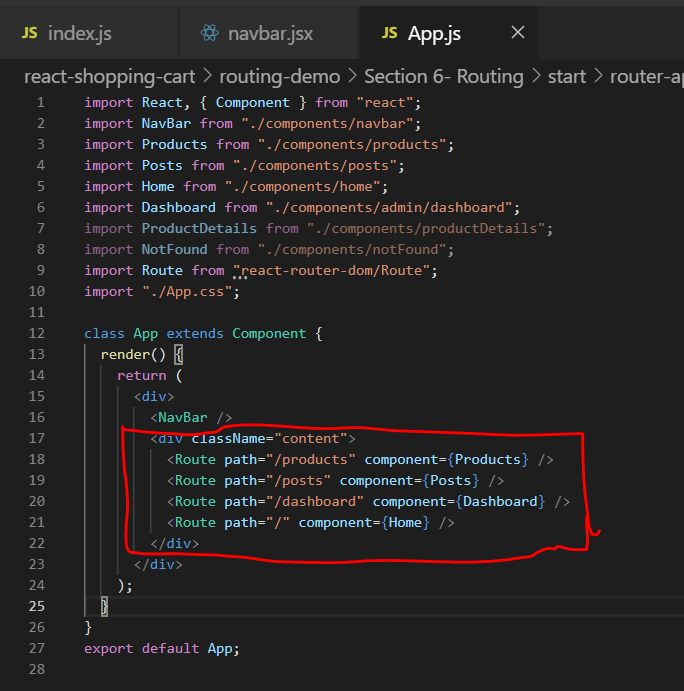
<BrowserRouter>

<App />

</BrowserRouter>,

document.getElementById("root")

);

1. Now we need to tell react what component to render based on URL path by adding routes
   1. Go to App Component and associate Route component to each of the links listed in the navbar component
      1. 
   2. 
   3. Paste the below code ( selectively)

import Route from "react-router-dom/Route";

class App extends Component {

render() {

return (

<div>

<NavBar />

<div className="content">

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

<Route path="/posts" component={Posts} />

<Route path="/admin" component={Dashboard} />

<Route path="/" component={Home} />

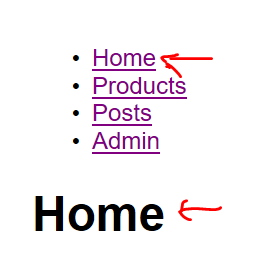
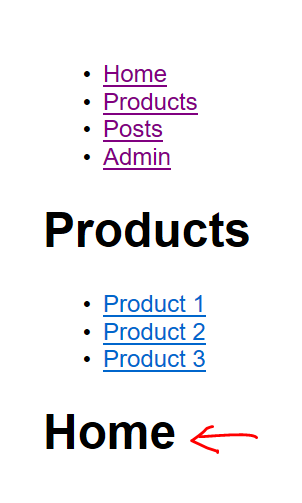
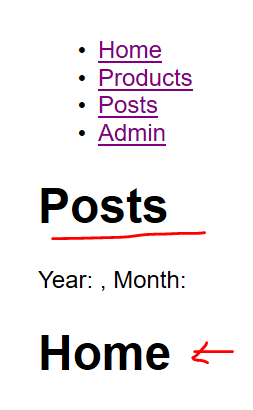
</div>

</div>

);

}

}

1. Now we click on different links we get to see respective components
   1. 
2. One problem is we see home component by default in all component clicks . This is because multiple routes match the URL path . Kind of partial match
   1.  



1. In order to fix the issue we need to add exact match

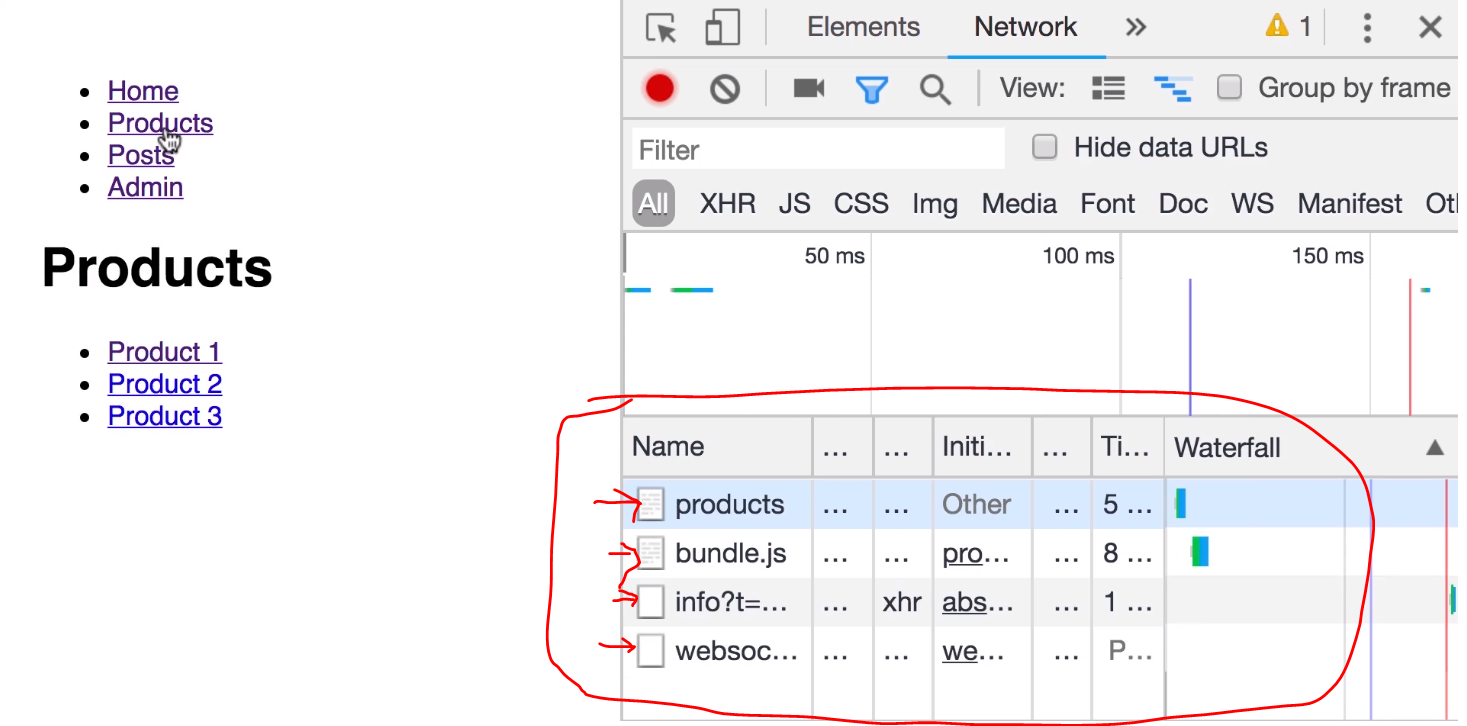
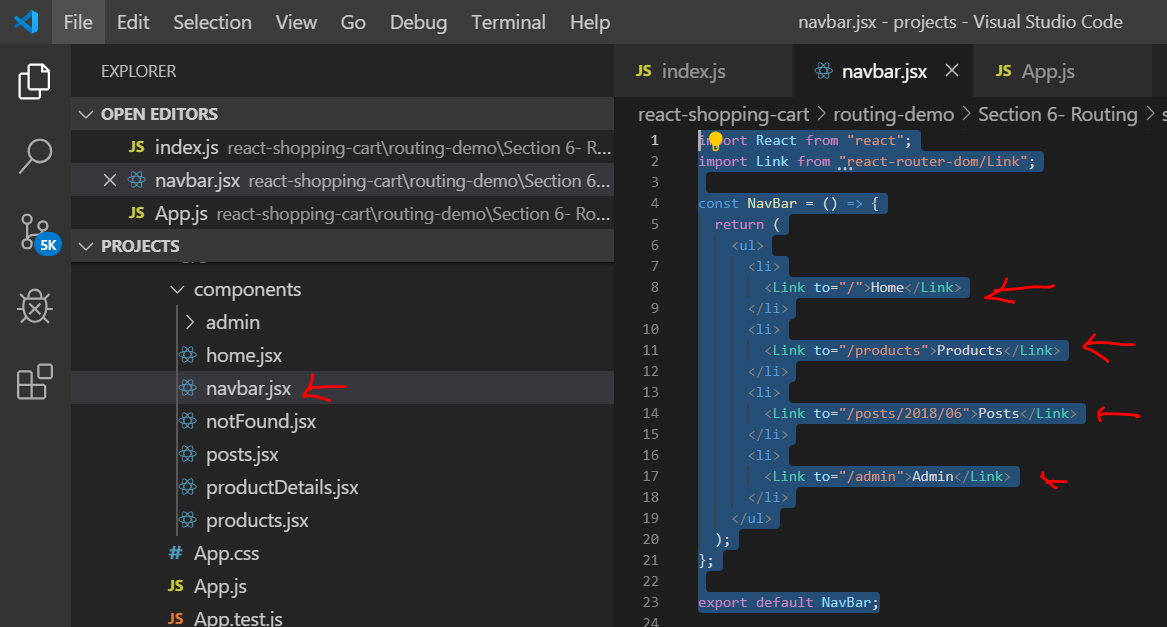
<Route path="/" exact component={Home} />

1. The other solution is to use switch component , this does the first match and ignores the other routes

import Switch from "react-router-dom/Switch";

* 1. 



1. If we notice in developer tools on click on each component the entire page is loaded every time, this is because it is not a single page application , we need to only refresh the content area which is changed and avoid full page refresh
   1. 
2. 

Paste the below code in navbar.jsx

import React from "react";

import Link from "react-router-dom/Link";

const NavBar = () => {

return (

<ul>

<li>

<Link to="/">Home</Link>

</li>

<li>

<Link to="/products">Products</Link>

</li>

<li>

<Link to="/posts/2018/06">Posts</Link>

</li>

<li>

<Link to="/admin">Admin</Link>

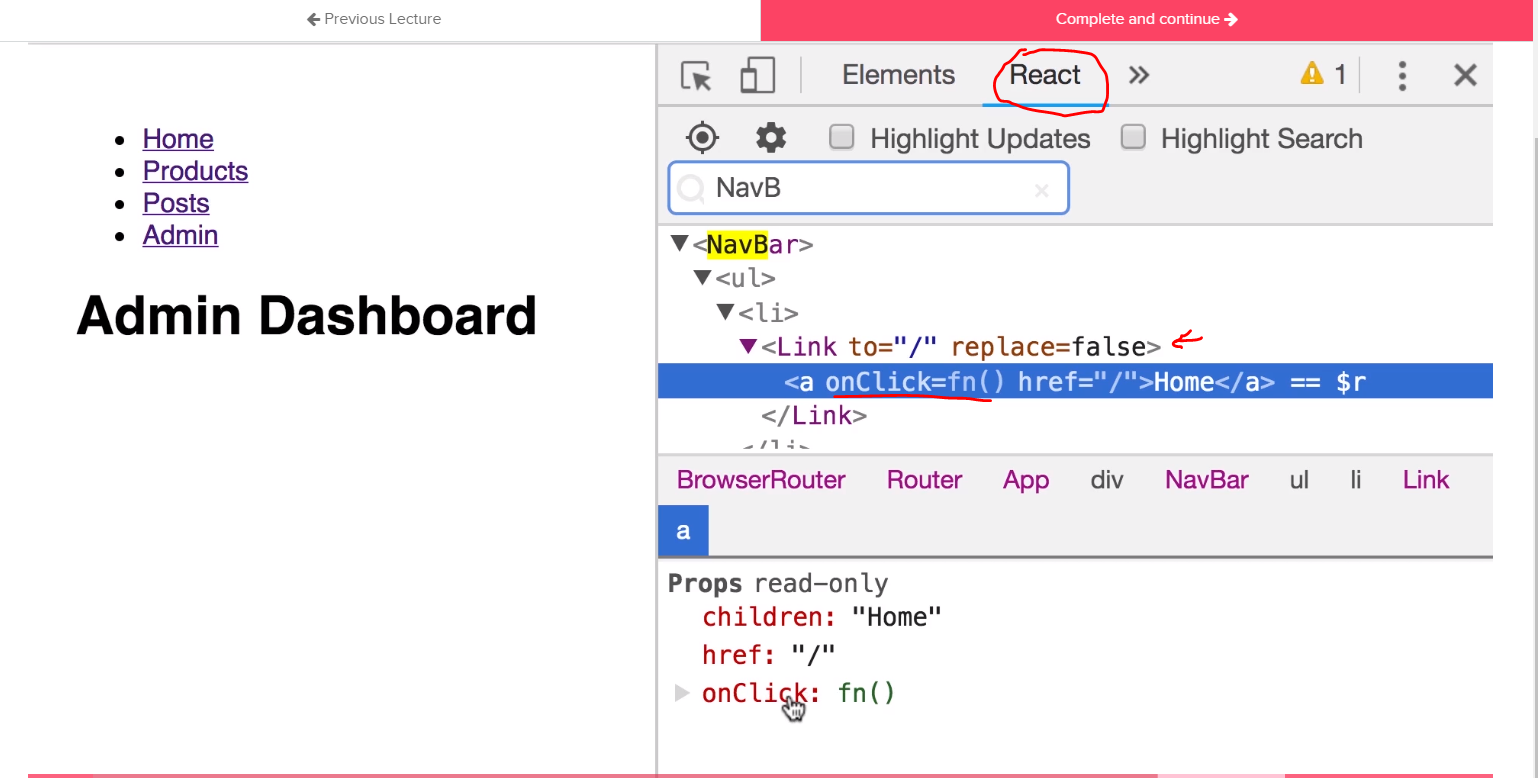
</li>

</ul>

);

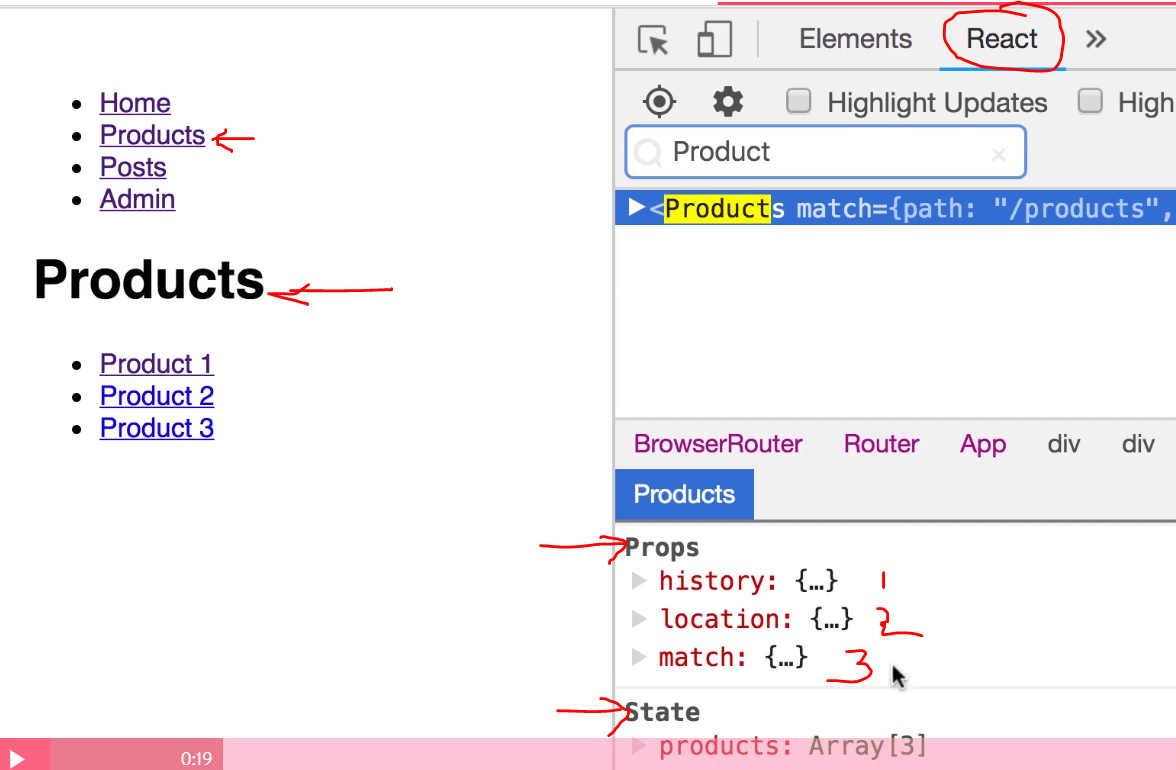
};

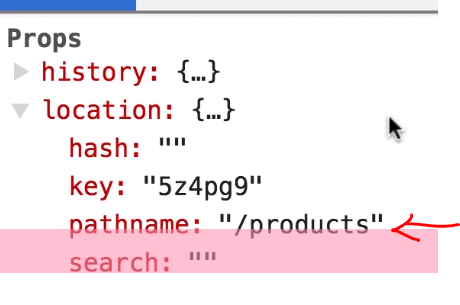
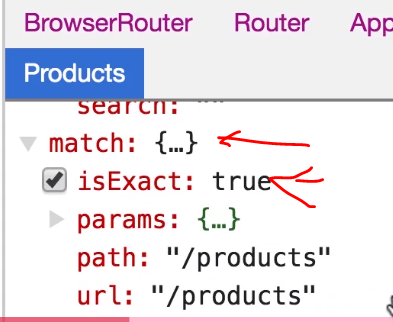
export default NavBar;

1. Observe the network tab when loading each component, now it doesn’t reload the entire page
2. All components are part of the bundle.js , they are loaded ahead of time and only once during the first component load
3. If we are wondering how this is Link component prevented additional request to the server
   1. 
4. Under the hoods it is still an anchor tag with handler , it only updates the browser URL and stops the http request.
5. The router upon change of the URL path loads the requested component matching the URL.

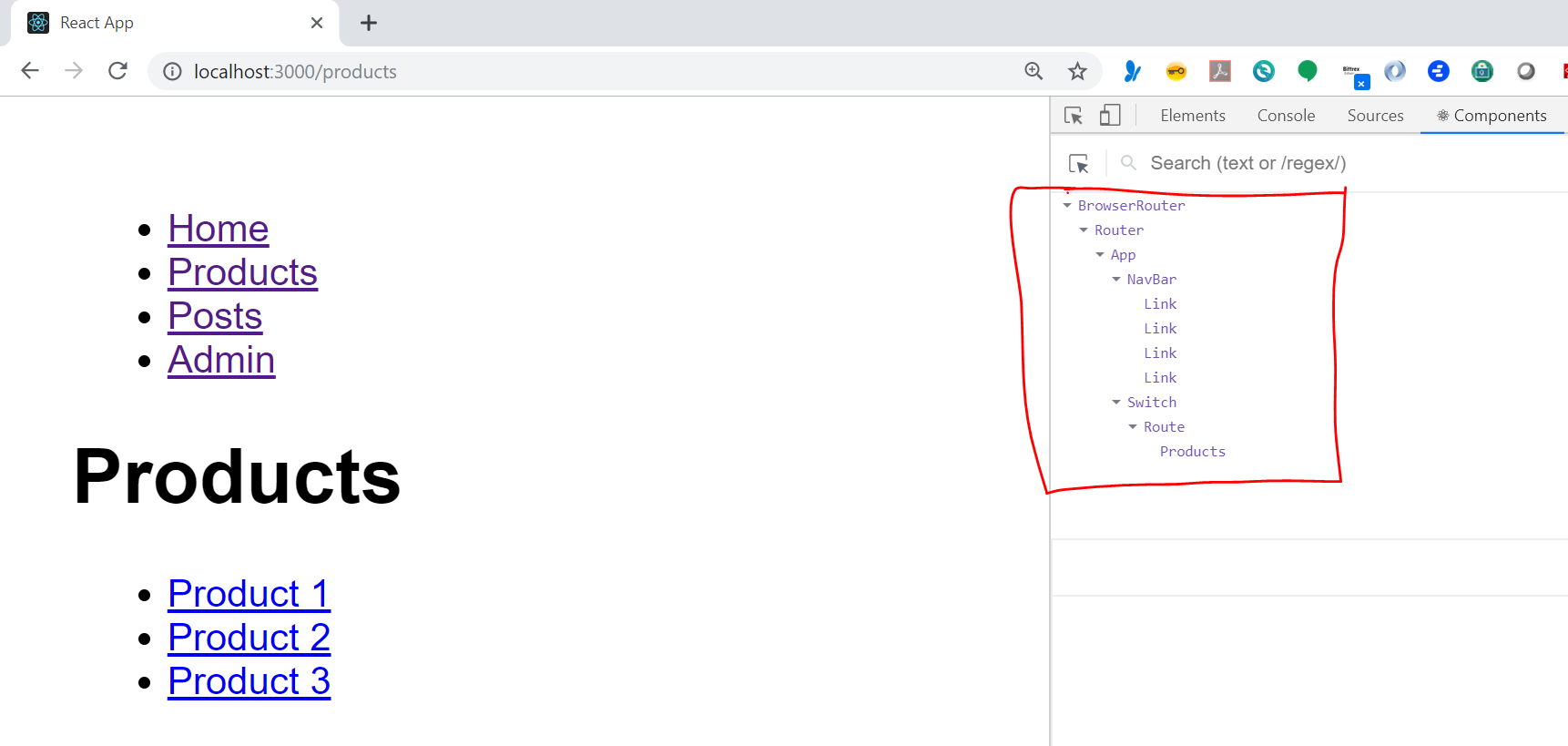
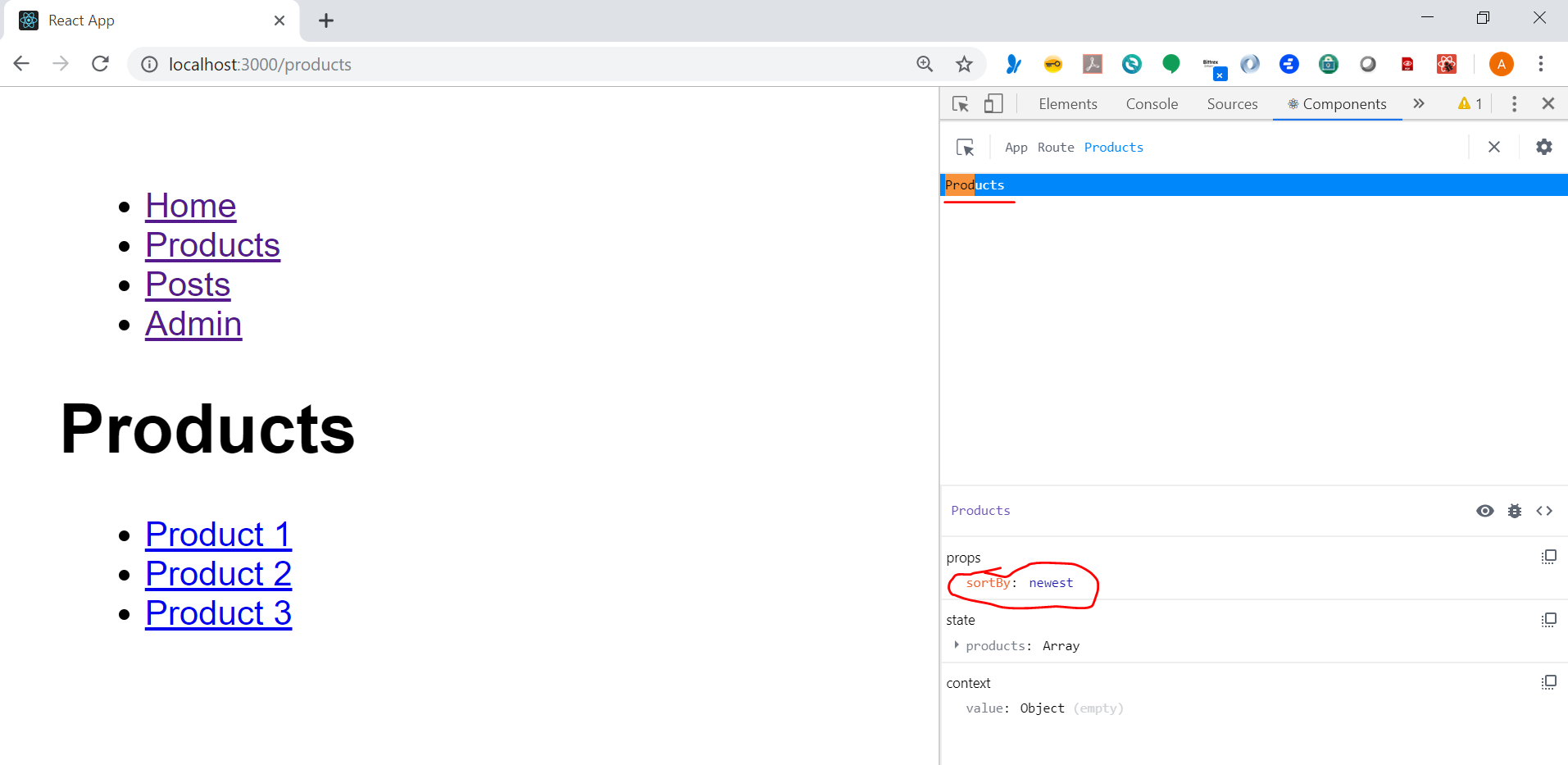
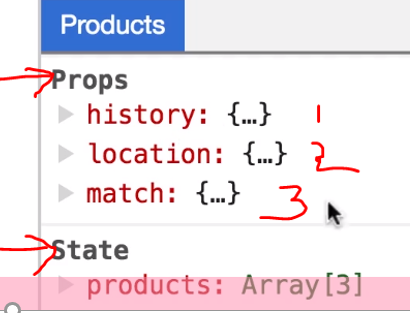


Look at the product component in react developer tools

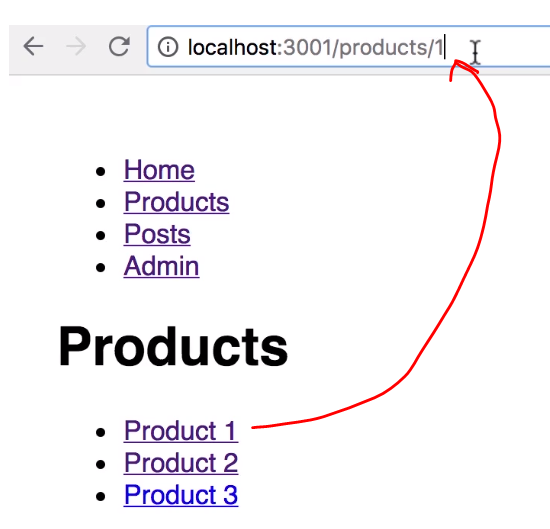
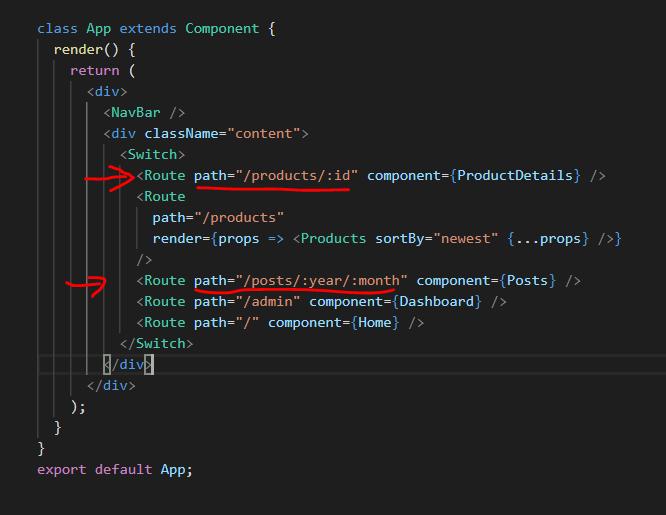


1. If we observe the product component has by default 3 props
   1. History
   2. Location
   3. Match
2. We did not pass them, they have been passed by default by parent wrapper component which is nothing but Route component
3. Observe the details
   1.  
   2. These properies will be user later in the course
   3. Explore more about Route component and ites porps under below link
      1. <https://reacttraining.com/react-router/web/guides/quick-start>



1. If we have to pass additional parameter to the component from parent, we need to use render property with an arrow function to handle as shown below. The arrow function will pass the prop and return the component
2. Replace Products route with below code
   1. <Route
   2. path="/products"
   3. render={() => <Products sortBy="newest" />}
   4. />
3. Observer the Prop passed to product component in developer tools
   1. 
   2. 
4. Below default props which were appearing before disappeared
   1. 
5. To fix this problem add below code , which passes the default props explicity
   1. <Route
   2. path="/products"
   3. render={props => <Products sortBy="newest" {...props} />}
   4. />

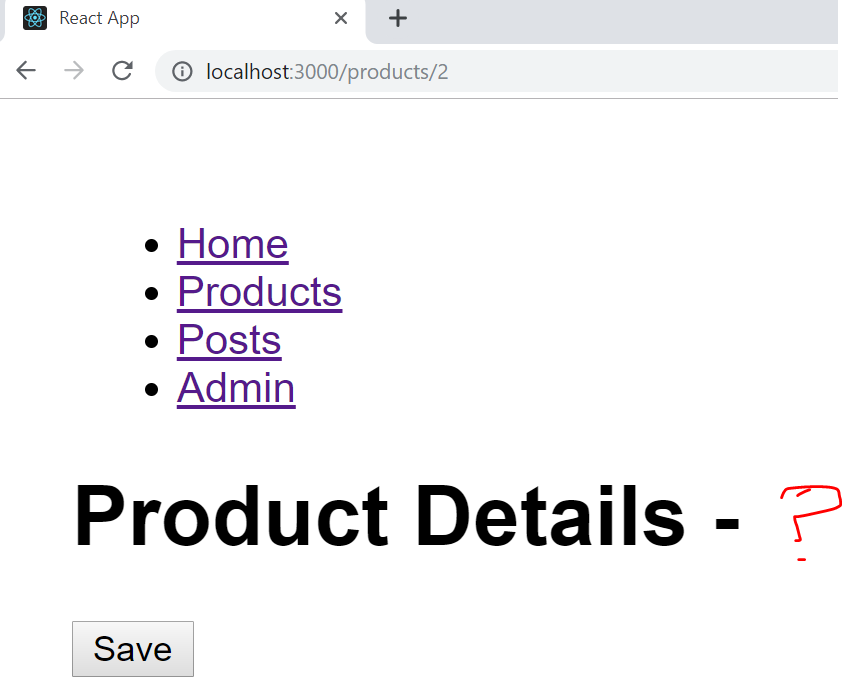
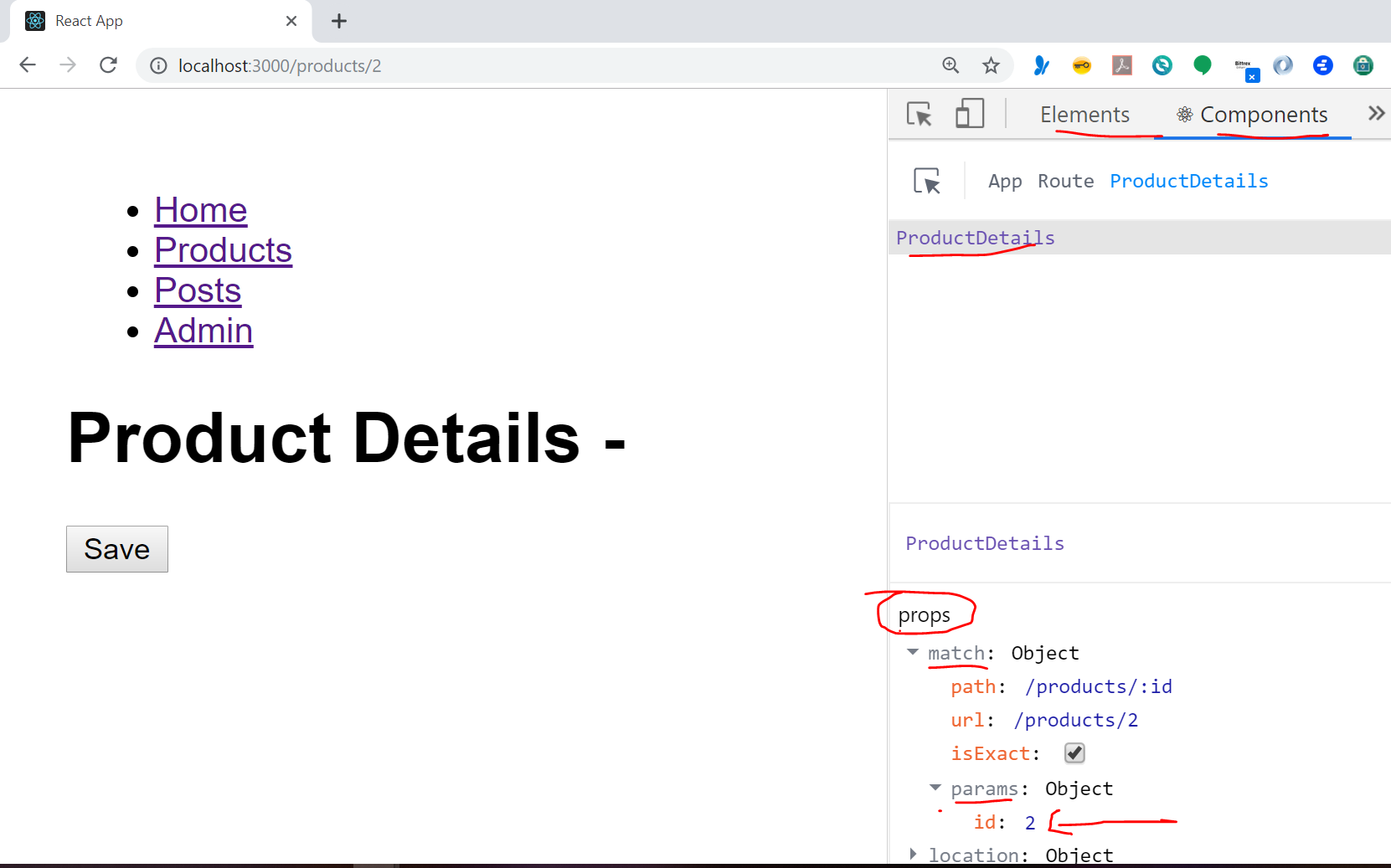
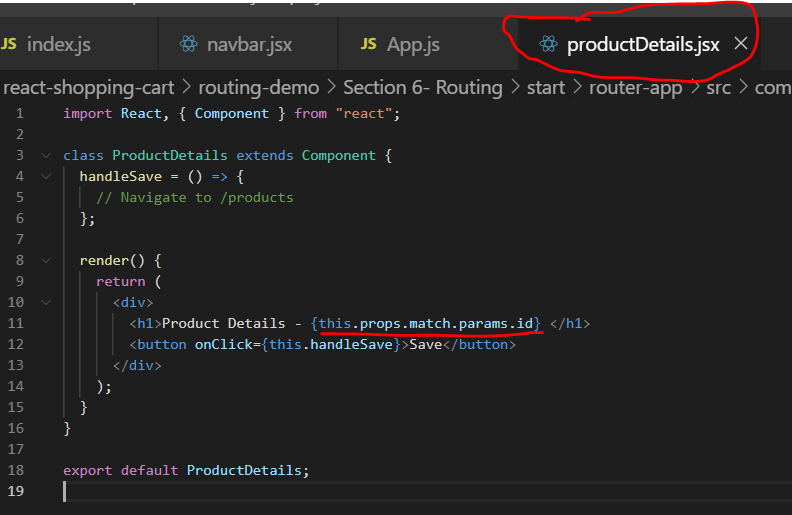
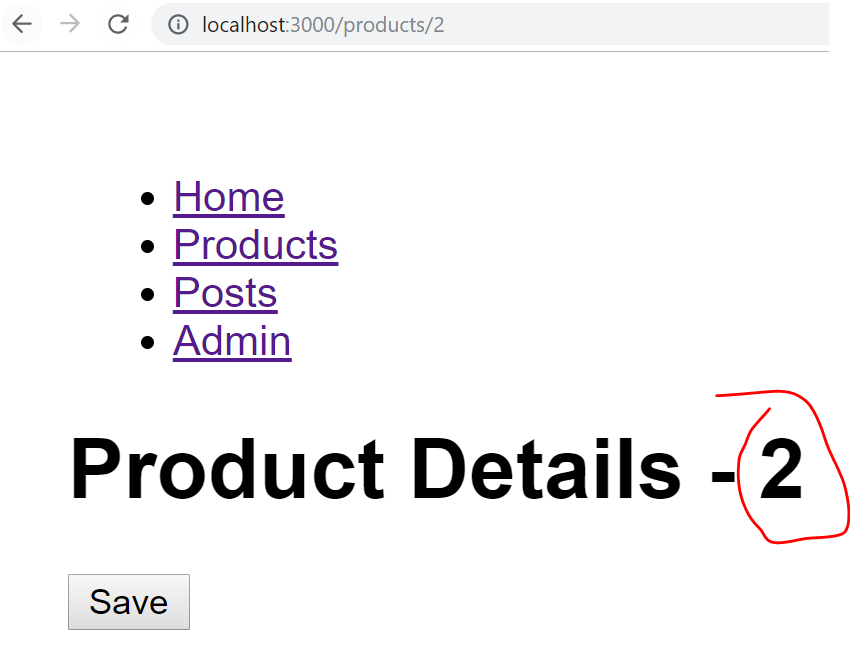
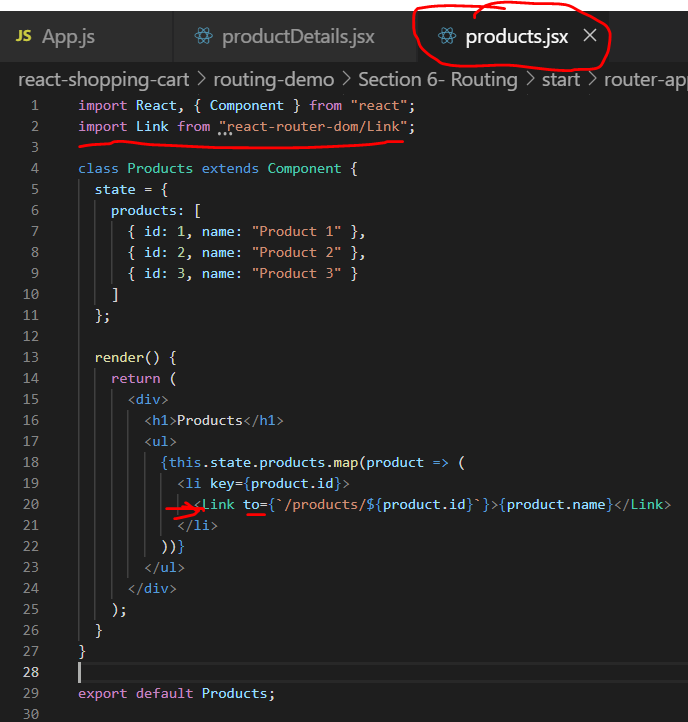


1. Upon selection of product I should see different URL
2. Lets make below changes to code to navigate to product details page
   1. 
   2. Add an additional route to product details

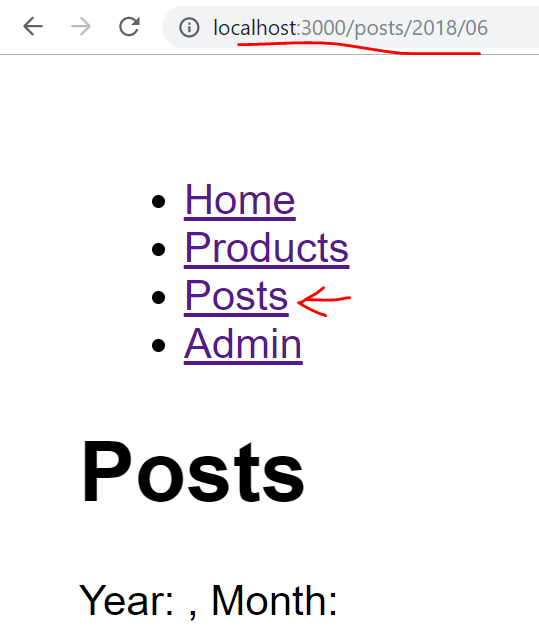
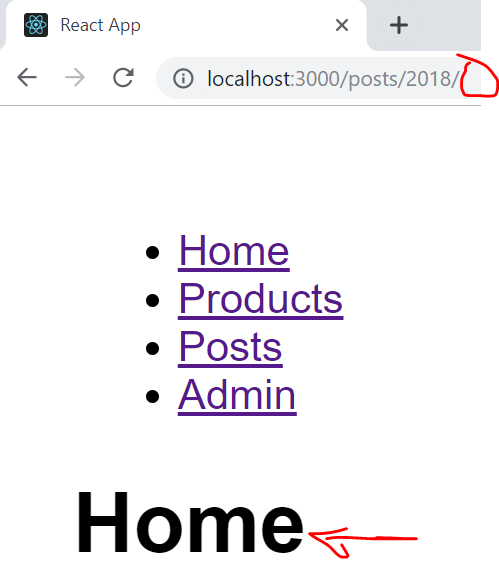
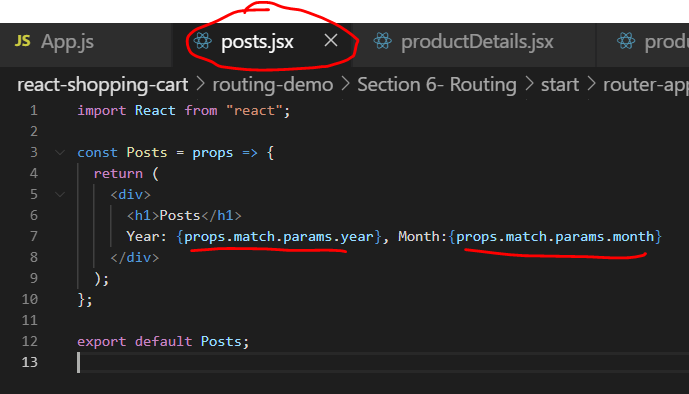
<Route path="/products/:id" component={ProductDetails} />

* 1. Make posts component accept year and month route parameters

<Route path="/posts/:year/:month" component={Posts} />

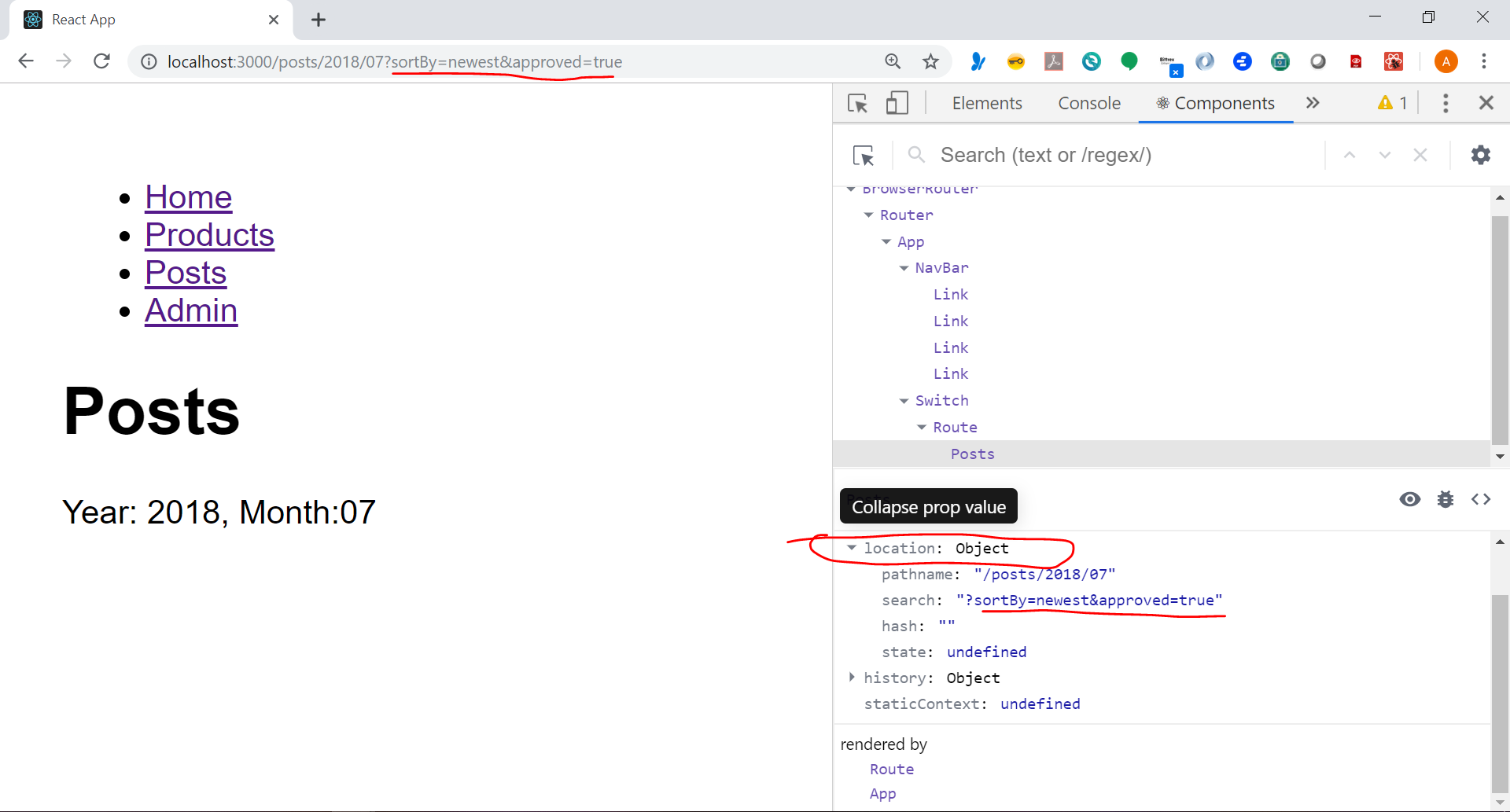
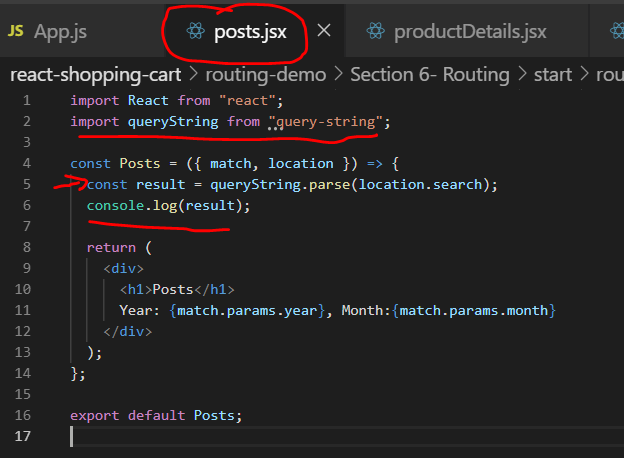
1. Notice new route to product detail is ahead of products route
2. Now to I want to print product id from the URL in child (product detail component)
   1. 
3. It is already available under props - > match -> params
   1. 
4. Add the below code is Product details component
   1. 
      1. 
5. Another issue – if we notice on click of products the entire page is reloaded , to avoid that make below changes in products component
   1. 
   2. Code to paste
      1. import Link from "react-router-dom/Link";
      2. <Link to={`/products/${product.id}`}>{product.name}</Link>



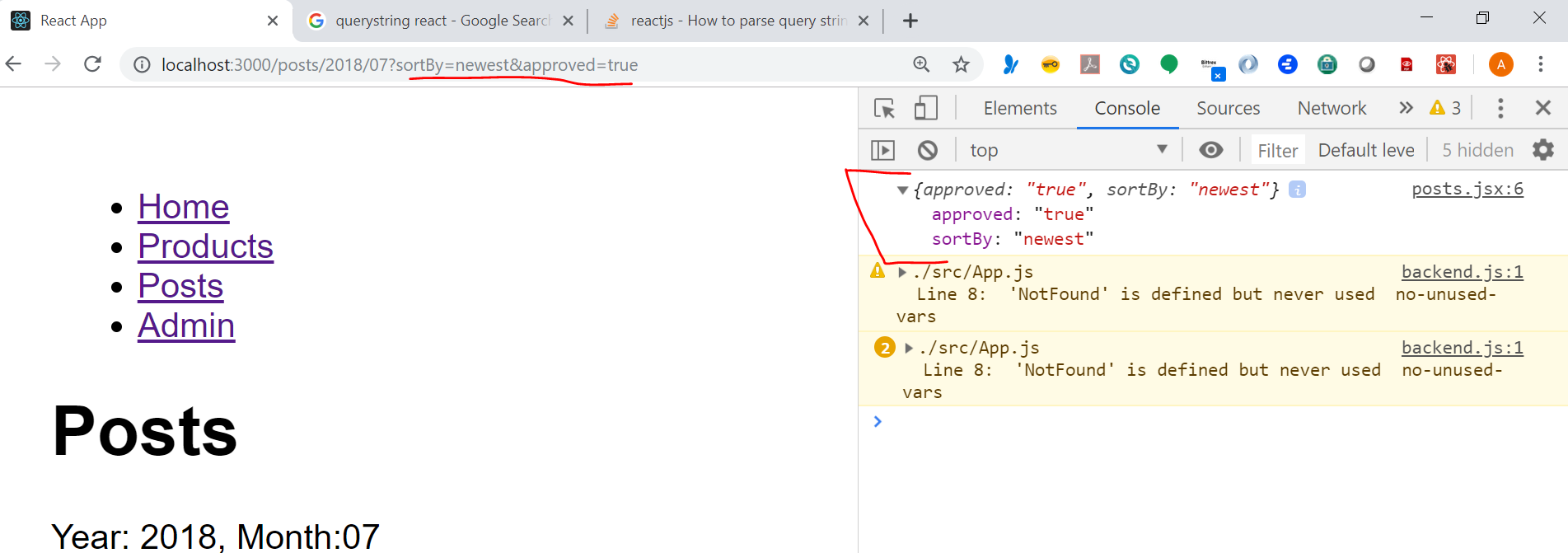
1. When we click on post component link the below is displayed
   1. 
2. When we remove the month param from URL we see home component. Why is this? This is because the routes require exact match and as below URL is not matching any route by default home component is being displayed
   1. 
3. To fix this we need to make URL params options as shown below
   1. 
4. Let’s print/read the year and month params inside posts component
   1. 
5. Now if you optionally provide month as part of the URL and observe the behavior



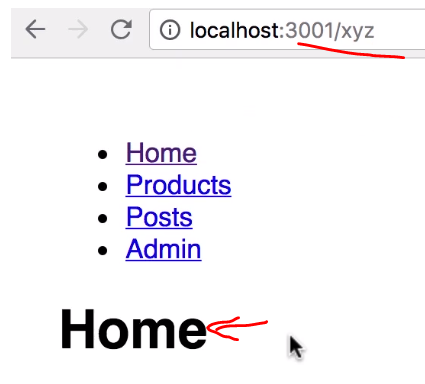


1. It is good practice to include optional parameters as part of URL query string and not as part of URL path ( Except Id parameter)
2. Let’s say, newest approved posts in query string. They are available under porps.location and not under match
   1. 
3. We take help of below package to read the query string parameters
   1. npm i query-string@6.1.0
4. Make below code changes
   1. 
   2. Code
      1. import queryString from "query-string";
      2. const result = queryString.parse(location.search);

console.log(result);

* 1. Observe the change in console
     1. 

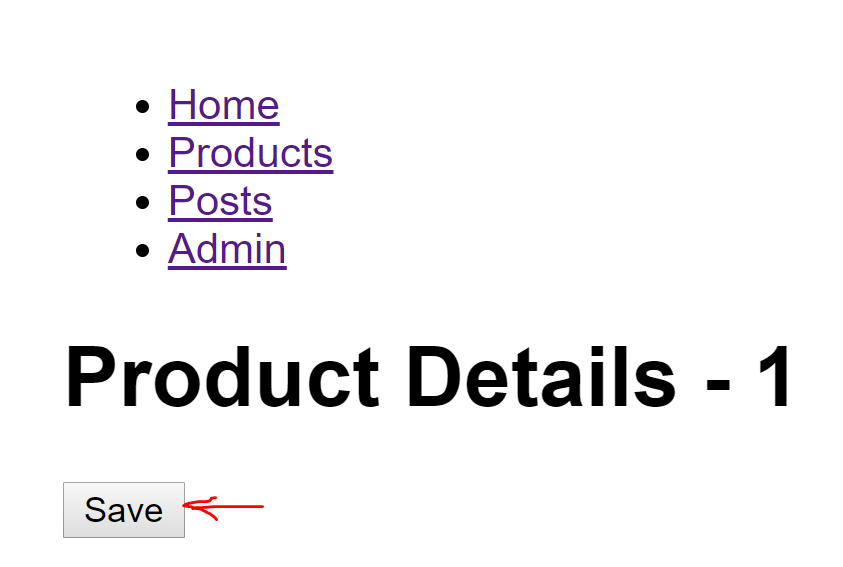
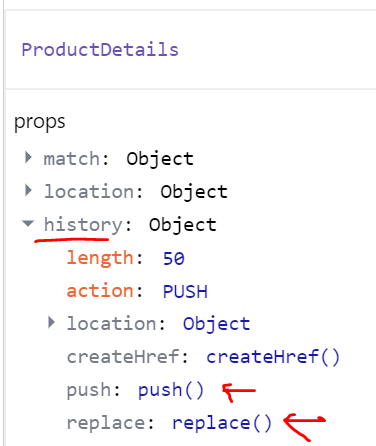
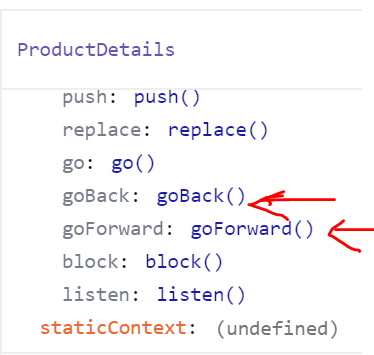
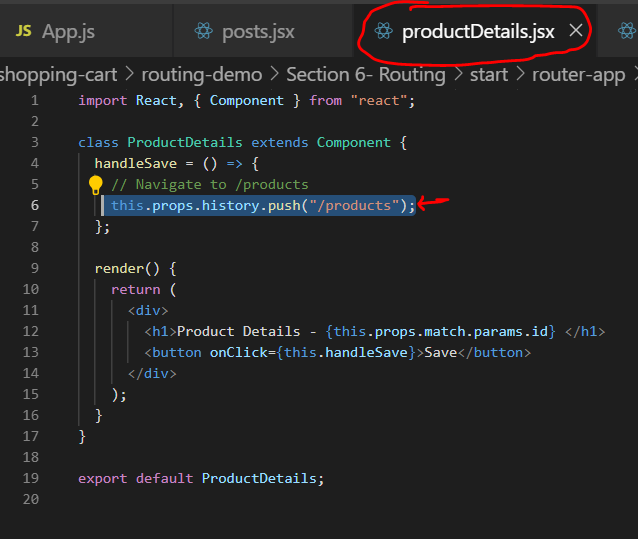


1. If I type an invalid route we get Home component by default,
   1. 
2. That is because we don’t have exact match for the user typed URL and by default home route is at the end of list of routes and it is not having exact match attribute
3. Let’s exact attribute and no component will be displayed for <http://localhost:3000/asdsad>
   1. <Route path="/" exact component={Home} />
4. Next : if not matching route is found we need to redirect to not found page
   1. import Redirect from "react-router-dom/Redirect";
   2. <Redirect to="/not-found" />
   3. Which component to redirect to ? Register a new route to NotFound component

<Route path="/not-found" component={NotFound} />

1. Similarly redirect can be used for redirecting from one URL to another
   1. <Redirect from="/messages" to="/posts" />



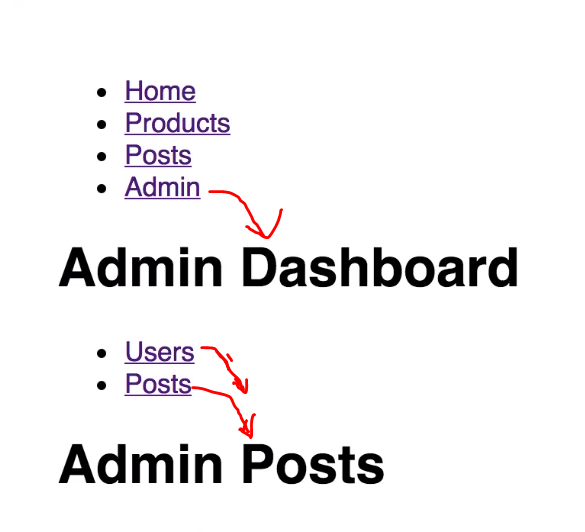
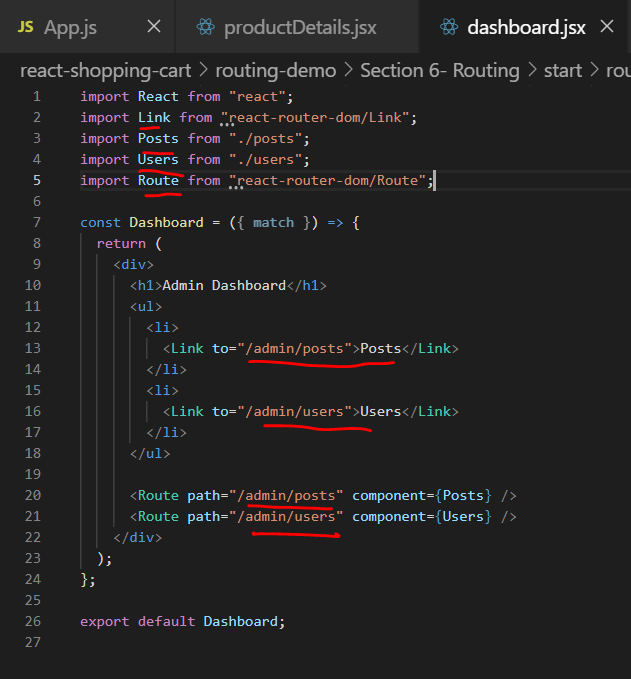
1. Next purpose : When user clicks on save we want to redirect to product list page , we can do this by programmatic navigation
   1. 
2. So far we have seen usage of location and match porps , now we need to use history for this purpose. It has below highlighted methods
   1.  
      1. Push -Adds user action to browser history
      2. Replace – will not add user action to browser history
3. Lets take a look at how both are used
4. Go to save handler in product details page and write below code
   1. 
   2. Code

this.props.history.push("/products");

* 1. Similarly test replace and show how back button will work in both cases. Replace will be used in case of login pages

this.props.history.replace("/products");

NESTED ROUTING

1. Example for nested routing
   1. 
2. Let’s try to implement this
3. Add highlighted code to dashboard component
   1. 
   2. Code
      1. import Posts from "./posts";
      2. import Users from "./users";
      3. import Route from "react-router-dom/Route";
   3. Link Code

<ul>

<li>

<Link to="/admin/posts">Posts</Link>

</li>

<li>

<Link to="/admin/users">Users</Link>

</li>

</ul>

* 1. Route Code

<Route path="/admin/posts" component={Posts} />

<Route path="/admin/users" component={Users} />

1. What we need to note is that route component need not be just used in app.js it can be used like anywhere like other components. It renders the output within the content area it belongs to