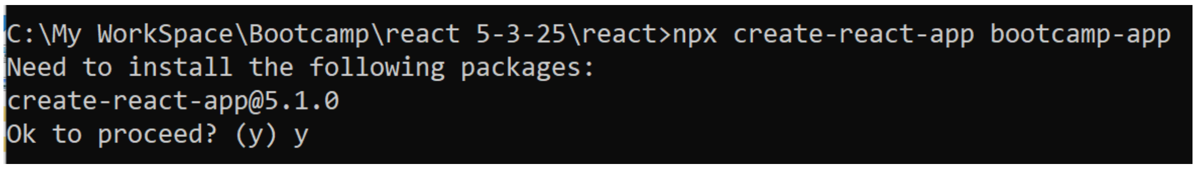
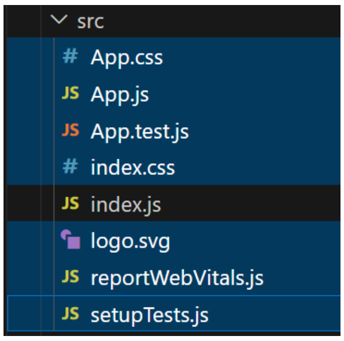
1. Creating app in react



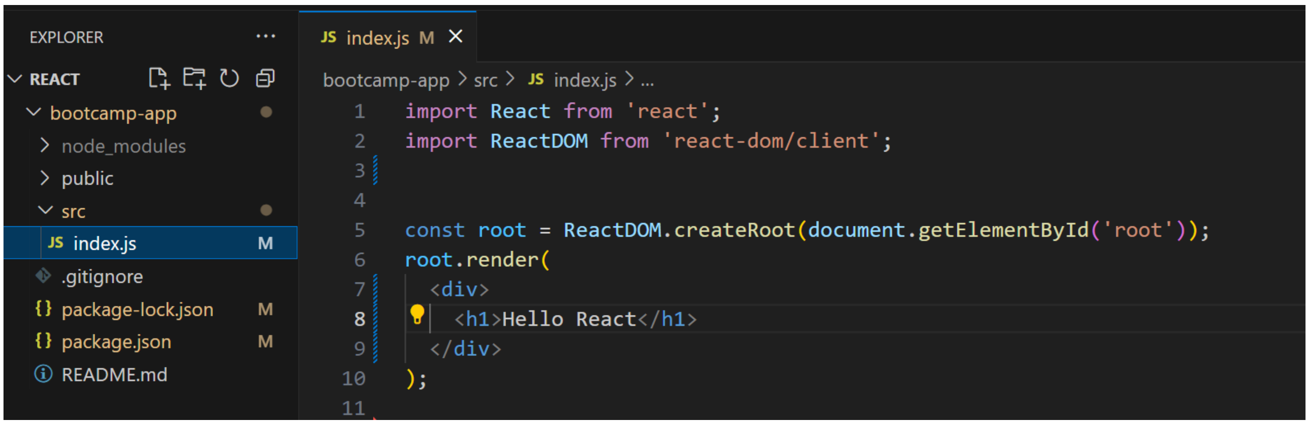
```
C:\My WorkSpace\Bootcamp\react 5-3-25\react>npx create-react-app bootcamp-app
Need to install the following packages:
create-react-app@5.1.0
Ok to proceed? (y) y
```

2. Install required packages
   a. C:\My WorkSpace\Bootcamp\react 5-3-25\react>cd bootcamp-app
   b. C:\My WorkSpace\Bootcamp\react 5-3-25\react\bootcamp-app>npm i react-router-dom
   c. C:\My WorkSpace\Bootcamp\react 5-3-25\react\bootcamp-app>npm i bootstrap@5.3.3
3. Run project
   a. C:\My WorkSpace\Bootcamp\react 5-3-25\react\bootcamp-app>npm start
4. Open project in vs code and delete all files inside src excluding index.js



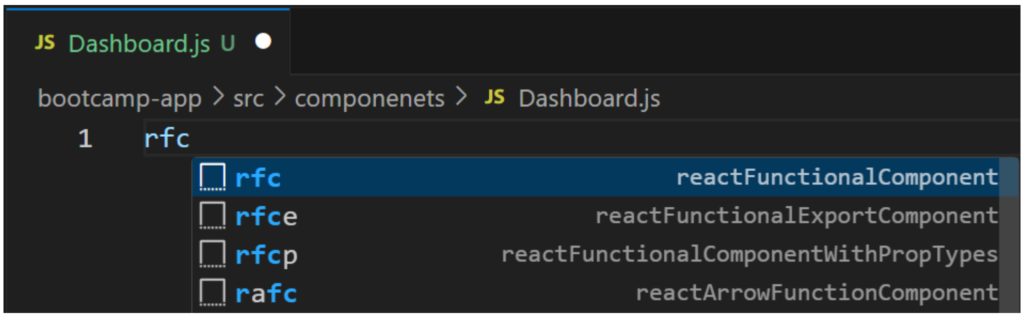
```
src
  App.css
  App.js
  App.test.js
  index.css
  index.js
  logo.svg
  reportWebVitals.js
  setupTests.js
```

5. Make index.js look like



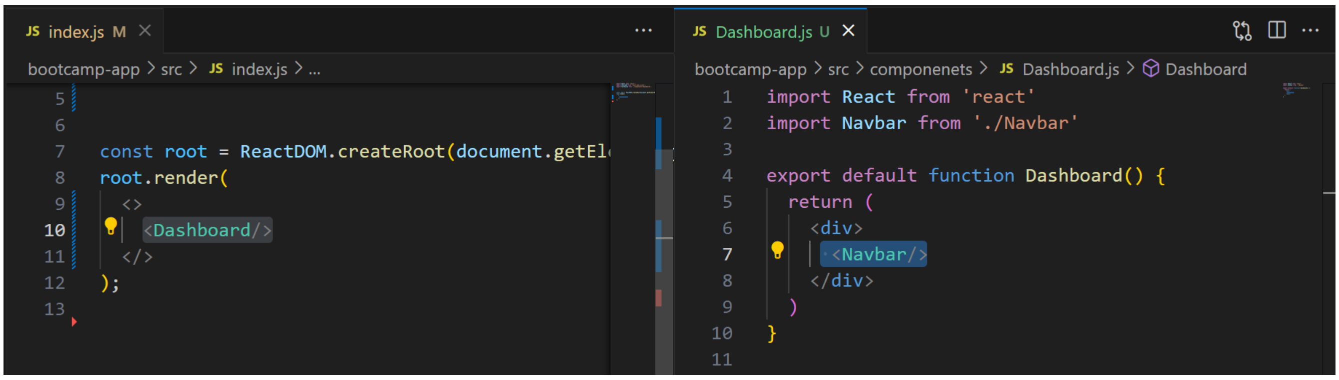
```
import React from 'react';
import ReactDOM from 'react-dom/client';


const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <div>
    <h1>Hello React</h1>
  </div>
);

```

6. Create all required components in react application
7. Create code for all component using rfc



```
rfc
rfc     reactFunctionalComponent
rfce    reactFunctionalExportComponent
rfcp    reactFunctionalComponentWithPropTypes
rafc    reactArrowFunctionComponent
```

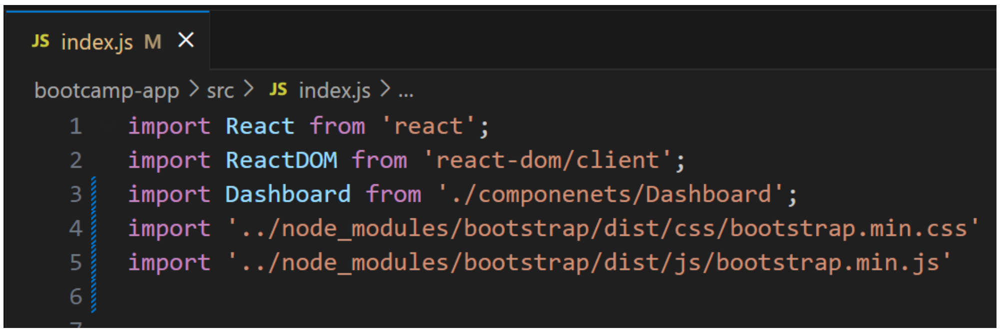
8. Create project wireframe
9. According to wireframe add components



```
index.js M ×
bootcamp-app > src > index.js > ...


const root = ReactDOM.createRoot(document.getEl
root.render(
  <>
    <Dashboard/>
  </>
);

```

```
Dashboard.js U ×
bootcamp-app > src > componenets > Dashboard.js > Dashboard
import React from 'react'
import Navbar from './Navbar'

export default function Dashboard() {
  return (
    <div>
      <Navbar/>
    </div>
  )
}

```

10. Add bootstraps css and js reference inside index.js file



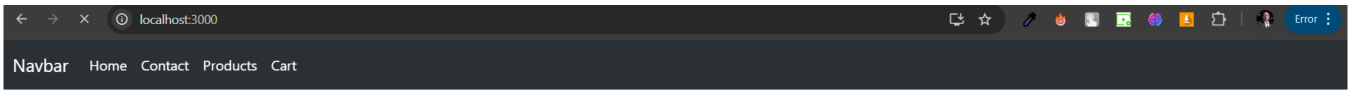
```
index.js M ×
bootcamp-app > src > index.js > ...
import React from 'react';
import ReactDOM from 'react-dom/client';
import Dashboard from './componenets/Dashboard';
import '../node_modules/bootstrap/dist/css/bootstrap.min.css'
import '../node_modules/bootstrap/dist/js/bootstrap.min.js'

```

11. Code for navbar
    a. Copy navbar code from bootstrap and add inside Navbar.js
    b. You might get errors – just close the tags like – hr, input etc and change class to className
    c. Code

```
import React from 'react'

export default function Navbar() {
  return (
    <div>
      <nav className="navbar navbar-expand-lg bg-body-tertiary" data-bs-
theme="dark">
        <div className="container-fluid">
          <a className="navbar-brand" href="#">Navbar</a>
          <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
            <span className="navbar-toggler-icon"></span>
          </button>
          <div className="collapse navbar-collapse" id="navbarSupportedContent">
            <ul className="navbar-nav me-auto mb-2 mb-lg-0">
              <li className="nav-item">
                <a className="nav-link active" aria-current="page"
href="#">Home</a>
              </li>
              <li className="nav-item">
                <a className="nav-link active" aria-current="page"
href="#">Contact</a>
              </li>
              <li className="nav-item">
                <a className="nav-link active" aria-current="page"
href="#">Products</a>
```

```
              </li>
              <li className="nav-item">
                <a className="nav-link active" aria-current="page"
href="#">Cart</a>
              </li>
            </ul>
          </div>
        </div>
      </nav>
    </div>
  )
}
```

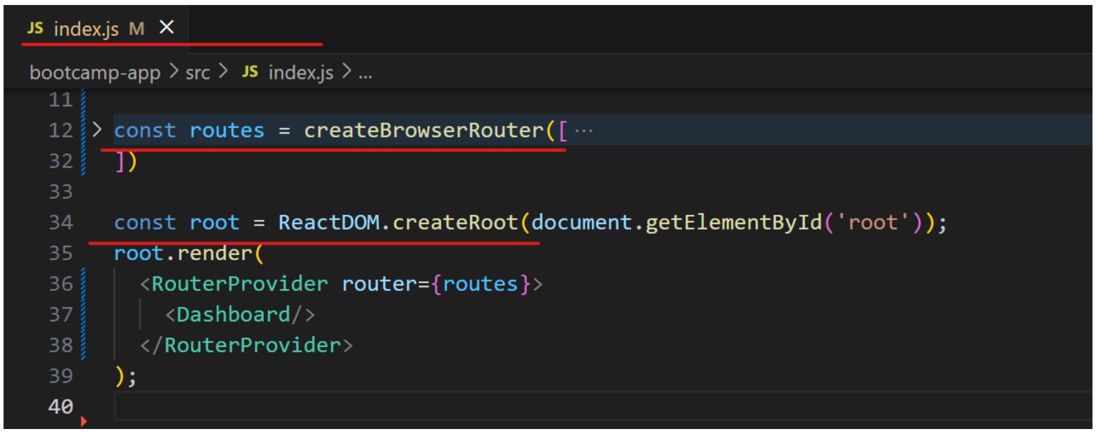
d. Output



Navbar Home Contact Products Cart

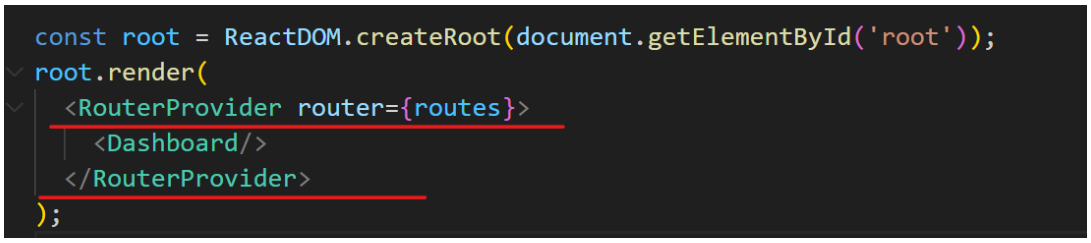
12. Implementing routing
    a. Create routes array inside index.js using createBrowserRouter

```
const routes = createBrowserRouter([
  {
    path:'/',
    element:<Dashboard/>,
    errorElement:<ErrorPage/>,
    children:[
      {
        path:'/contact',
        element:<Contact/>
      },
      {
        path:'/product',
        element:<Product/>
      },
      {
        path:'/product/:id',
        element:<ProductDetails/>
      }
    ]
  }
])
```



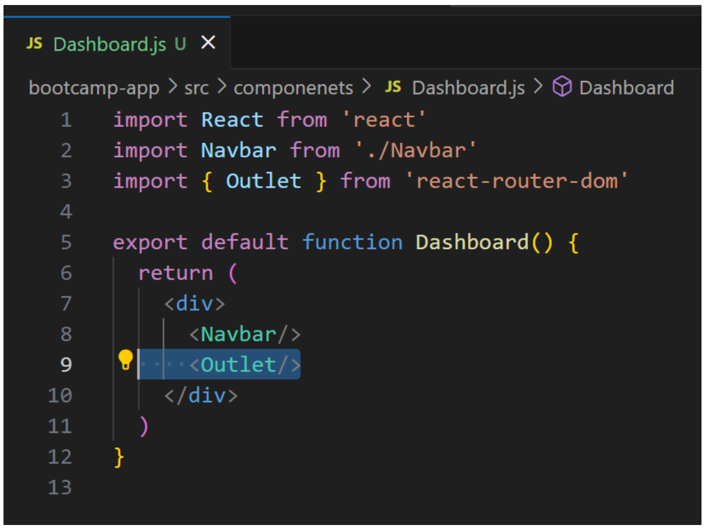
```
index.js M
bootcamp-app > src > index.js > ...
const routes = createBrowserRouter([ ···
])

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <RouterProvider router={routes}>
    <Dashboard/>
  </RouterProvider>
);
```

b. Add above routes array on RouterProvider as –



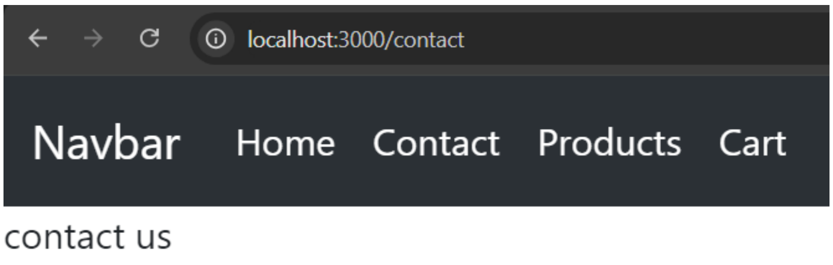
```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <RouterProvider router={routes}>
    <Dashboard/>
  </RouterProvider>
);
```

c. Add <Outlet/> below Navbar inside Dashboard.js to render components dynamically



```
Dashboard.js U
bootcamp-app > src > componenets > Dashboard.js > Dashboard
import React from 'react'
import Navbar from './Navbar'
import { Outlet } from 'react-router-dom'

export default function Dashboard() {
  return (
    <div>
      <Navbar/>
      <Outlet/>
    </div>
  )
}
```

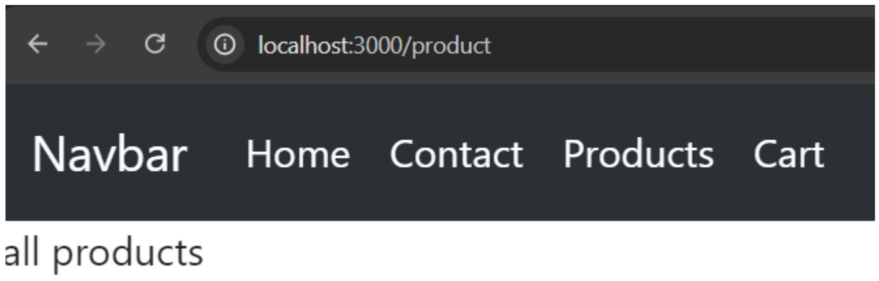
d. Check output



localhost:3000/contact

Navbar Home Contact Products Cart

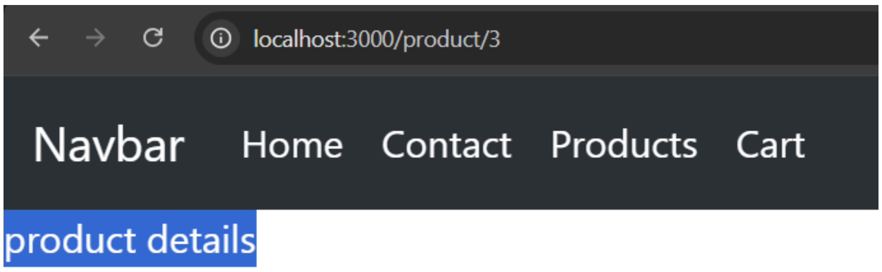
contact us



localhost:3000/product

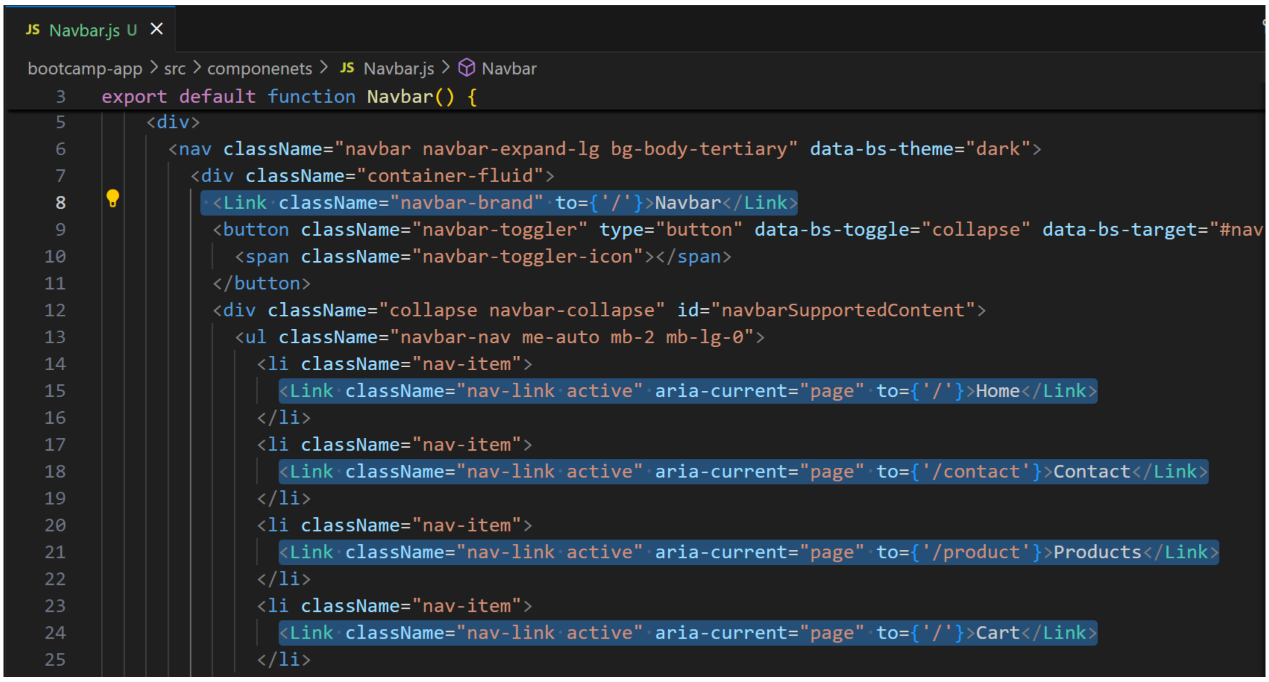
Navbar Home Contact Products Cart

all products



```
localhost:3000/product/3
Navbar   Home   Contact   Products   Cart
product details
```

13. Implementing routing inside navbar
    a. Instead of a tag, we will use Link tag to add routing as -



```
Navbar.js U
bootcamp-app > src > componenets > Navbar.js > Navbar
export default function Navbar() {
    <div>
      <nav className="navbar navbar-expand-lg bg-body-tertiary" data-bs-theme="dark">
        <div className="container-fluid">
          <Link className="navbar-brand" to={'/'}>Navbar</Link>
          <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#nav
            <span className="navbar-toggler-icon"></span>
          </button>
          <div className="collapse navbar-collapse" id="navbarSupportedContent">
            <ul className="navbar-nav me-auto mb-2 mb-lg-0">
              <li className="nav-item">
                <Link className="nav-link active" aria-current="page" to={'/'}>Home</Link>
              </li>
              <li className="nav-item">
                <Link className="nav-link active" aria-current="page" to={'/contact'}>Contact</Link>
              </li>
              <li className="nav-item">
                <Link className="nav-link active" aria-current="page" to={'/product'}>Products</Link>
              </li>
              <li className="nav-item">
                <Link className="nav-link active" aria-current="page" to={'/'}>Cart</Link>
              </li>
```

    b. DONE !!
    c. Go to navbar and click and check whether components are rendering

14. Create card inside Product.js
    Product.js

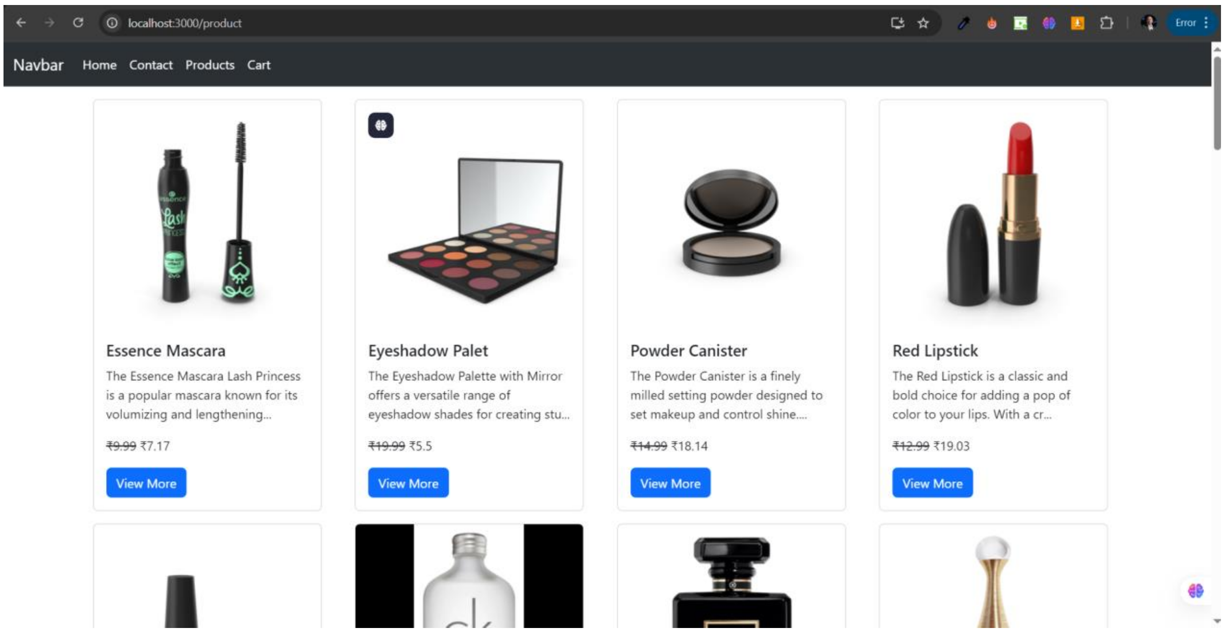
```
import React from 'react'
const products =
[{name:'mobile',price:200},{name:'laptop',price:600},{name:'mobile',price:200},{name:'laptop',price:600},
{name:'mobile',price:200},{name:'laptop',price:600}]
export default function Product() {
  return (
    <div className='container'>
      <div className='row'>
        {
          products.map(product => {
            return (
              <div className='col-3'>
                  <h1>{product.name}</h1>
              </div>
            )
          })
        }
      </div>
    </div>
  )
}
```

15. Fetching products data from api - https://dummyjson.com/products and displaying into Product.js component
    a. Hook in react are nothing but built in functions of react
    b. What is useState() hook?
       When state(default value) of variable changes, we want to render component again, here useState() help us to do that!

    c. Code for Product.js to display all products in card using fetch api

```
import React, { use, useEffect, useState } from 'react'
// const products = [{name:'mobile',price:200},{name:'laptop',price:600}]
export default function Product() {

  //let products = [{name:'mobile',price:200},{name:'laptop',price:600}]
  //let[variableToStoreDefaultvalue, functionToUpdateVariable]=useState(default state/value)
  let[products, setProducts ]=useState(null)
  //let [counter, setCounter]=useState(0)
  async function fetchProucts()
  {
    let response=await fetch("https://dummyjson.com/products",{method:'GET'})
    let data=await response.json()
    console.log(data);
    console.log(data.products);
    //products = data.products
    setProducts(data.products)
  }
  //fetchProucts()
  // useEffect(()=>{
  //   fetchProucts()
  // },[counter])
  useEffect(()=>{
    fetchProucts()
  },[])
  return (

    <div className='container mt-3'>
      {/* <p>{counter}</p>
      <button onClick={()=>{setCounter(counter+1)}}>change counter</button> */}
      <div className='row'>
        {
          products && products.map(product => {
            return (
              <div className='col-3 mb-3' key={product.id}>
                  <div className="card" style={{width:18+"rem"}}>
                      <img src={product.thumbnail} className="card-img-top" alt="..."/>
                      <div className="card-body">
                        <h5 className="card-title">{product.title.slice(0,15)}</h5>
                        <p className="card-text">{product.description.slice(0,95)}...</p>
                        <p className="card-text bg-warning">{product.category}</p>
                        <p><del>&#8377;{product.price}</del>  &#8377;{product.discountPercentag
e}</p>

                        <a href="#" className="btn btn-primary">View More</a>
                      </div>
                    </div>
                </div>
            )
          })
        }
      </div>
    </div>
  )

}
```

d. Output



localhost:3000/product

Navbar Home Contact Products Cart

Essence Mascara

The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening...

~~₹9.99~~ ₹7.17

View More

Eyeshadow Palet

The Eyeshadow Palette with Mirror offers a versatile range of eyeshadow shades for creating stu...

~~₹19.99~~ ₹5.5

View More

Powder Canister

The Powder Canister is a finely milled setting powder designed to set makeup and control shine....

~~₹14.99~~ ₹18.14

View More

Red Lipstick

The Red Lipstick is a classic and bold choice for adding a pop of color to your lips. With a cr...

~~₹12.99~~ ₹19.03

View More

e.

16. Parameterise routing

a. Here we have to display details of particular product when we click on View more button

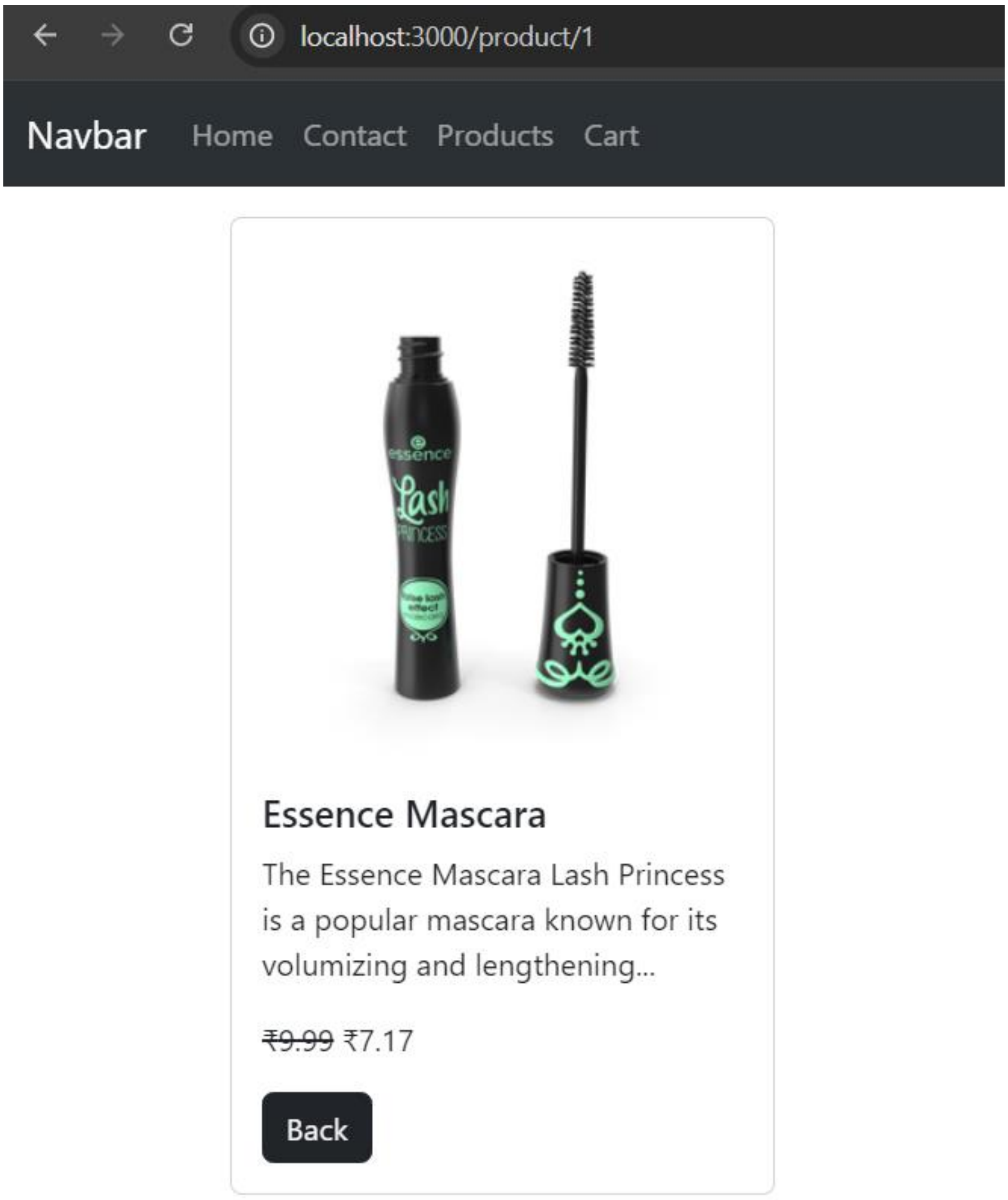
b. Add link to view more button

```
 <div className='col-3 mb-3' key={product.id}>
     <div className="card" style={{width:18+"rem"}}>
         <img src={product.thumbnail} className="card-img-top" alt="..."/>
         <div className="card-body">
           <h5 className="card-title">{product.title.slice(0,15)}</h5>
           <p className="card-text">{product.description.slice(0,95)}...</p>
           <p><del>&#8377;{product.price}</del>  &#8377;{product.discountPercentage}</p>
           <Link to={`/product/${product.id}`} className="btn btn-primary">View More</Link>
         </div>
       </div>
   </div>
```

c. Code inside ProductDetails.js to fetch product from its id and also implementing back button

```
import React, { useEffect, useState } from 'react'
import { Link, useNavigate, useParams } from 'react-router-dom'

export default function ProductDetails() {
  let urlParams=useParams()
  let navigate=useNavigate()
  console.log(urlParams);
  console.log(urlParams.id);
  let id = urlParams.id
  let[product, setProduct]=useState(null)
  async function fetchProuctById()
  {
    let response=await fetch(`https://dummyjson.com/products/${id}`,{method:'GET'})
    let data=await response.json()
    console.log(data);
    setProduct(data)
  }
  useEffect(()=>{fetchProuctById()},[])
  return (
    product &&
   <div className='container mt-3'>
        <div className='row'>
            <div className='col-3 mb-3' key={product.id}>
                  <div className="card" style={{width:18+"rem"}}>
                      <img src={product.thumbnail} className="card-img-top" alt="..."/>
                      <div className="card-body">
                        <h5 className="card-title">{product.title.slice(0,15)}</h5>
                        <p className="card-text">{product.description.slice(0,95)}...</p>
                        <p><del>&#8377;{product.price}</del>  &#8377;{product.discountPercentag
e}</p>
                        <button className='btn btn-dark text-light'
onClick={()=>{navigate('/product')}}>Back</button>
                      </div>
                  </div>
              </div>
        </div>
    </div>
  )
}
```

d. Output

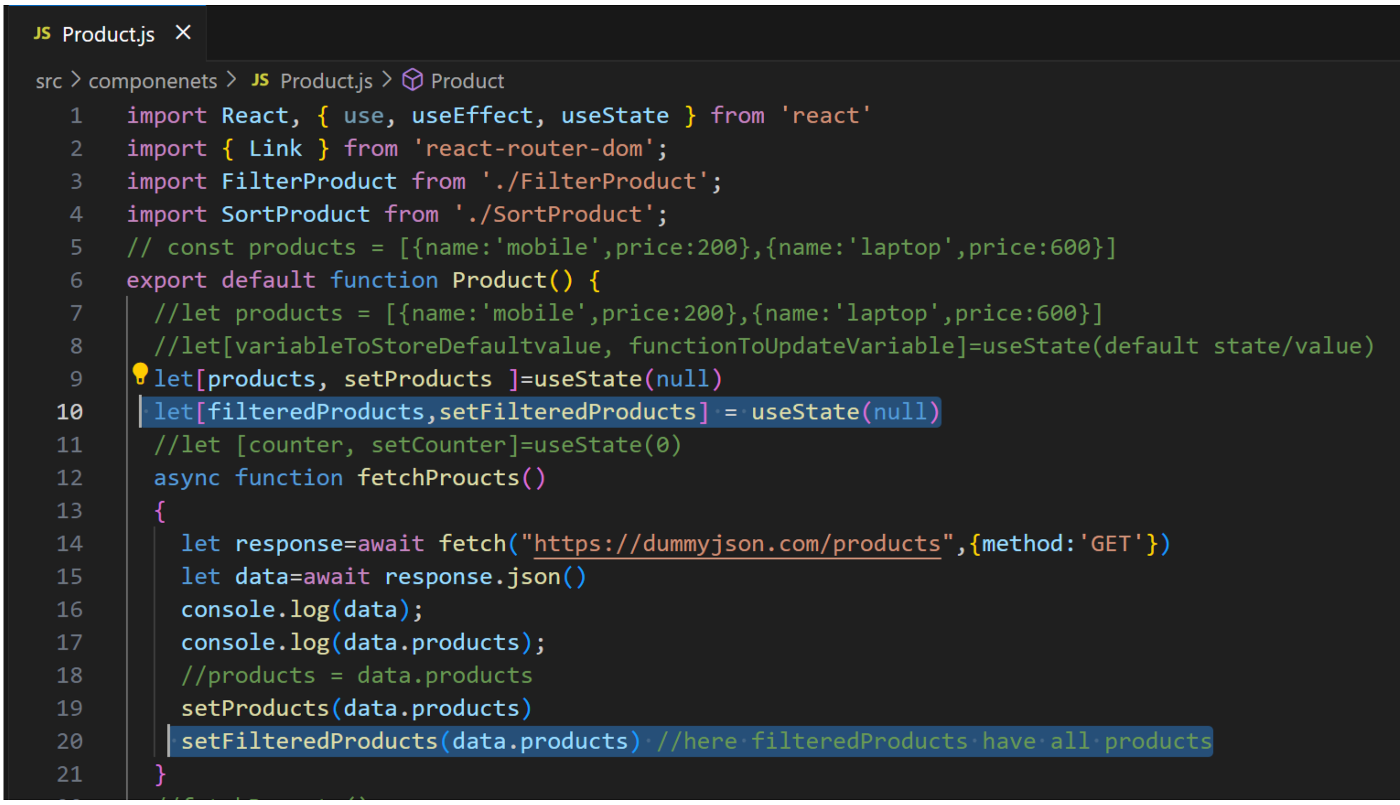

localhost:3000/product/1

Essence Mascara

The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening...

~~₹9.99~~ ₹7.17

17. Create useState to manage state of filtered products (in Product.js)

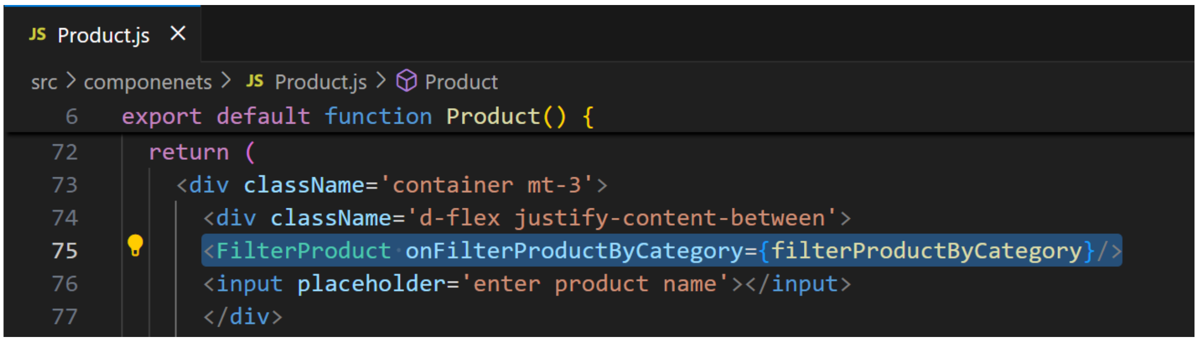


```
import React, { use, useEffect, useState } from 'react'
import { Link } from 'react-router-dom';
import FilterProduct from './FilterProduct';
import SortProduct from './SortProduct';
// const products = [{name:'mobile',price:200},{name:'laptop',price:600}]
export default function Product() {
  //let products = [{name:'mobile',price:200},{name:'laptop',price:600}]
  //let[variableToStoreDefaultvalue, functionToUpdateVariable]=useState(default state/value)
  let[products, setProducts ]=useState(null)
  let[filteredProducts,setFilteredProducts] = useState(null)
  //let [counter, setCounter]=useState(0)
  async function fetchProucts()
  {
    let response=await fetch("https://dummyjson.com/products",{method:'GET'})
    let data=await response.json()
    console.log(data);
    console.log(data.products);
    //products = data.products
    setProducts(data.products)
    setFilteredProducts(data.products) //here filteredProducts have all products
  }
```

18. Filter by product category functionality
    a. Create file FilterProduct.js 🡒 create component using rfc
    b. Create filterProductByCategory() function inside Product.js

```
  function filterProductByCategory(categoryName)
  {

      let filtredProductsBasedOnCategory=products.filter(product =>
        {
          return
product.category.toLowerCase().includes(categoryName.toLowerCase())
        })

      setFilteredProducts(filtredProductsBasedOnCategory)

  }
```

    c. Pass above function to FilterProduct component as –



```
export default function Product() {
  return (
    <div className='container mt-3'>
      <div className='d-flex justify-content-between'>
      <FilterProduct onFilterProductByCategory={filterProductByCategory}/>
      <input placeholder='enter product name'></input>
      </div>
```

    d. Final code for FilterProduct component –

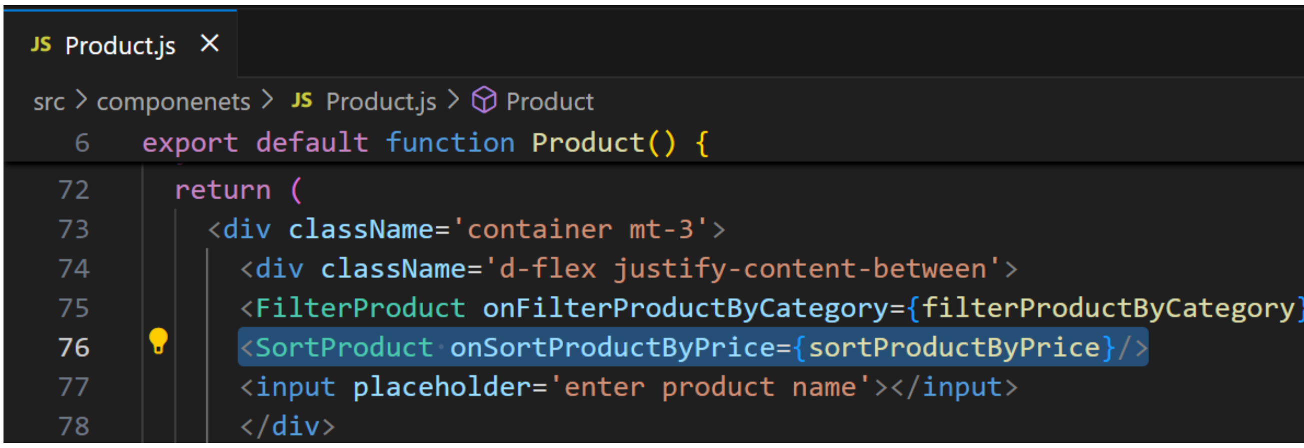
```
import React, { useEffect, useState } from 'react'

export default function FilterProduct(props) {
    let [categories, setCategories]=useState(null)
    async function fetchAllCategories()
    {
        let response =await fetch("https://dummyjson.com/products/categories")
        let data= await response.json()
        setCategories(data)
    }
    useEffect(()=>{fetchAllCategories()},[])
return (
    <div className='mb-3'>
        <div className="dropdown">
            <button className="btn btn-secondary dropdown-toggle" type="button" data-bs-
toggle="dropdown" aria-expanded="false">
                Select Category
            </button>
            <ul className="dropdown-menu">
                <li> <button className='btn'
onClick={()=>{props.onFilterProductByCategory("all")}}>All</button> </li>
                {
                   categories && categories.map(category => {
                    return (
                            <li> <button className='btn'
onClick={()=>{props.onFilterProductByCategory(category.name)}}>{category.name}</button>
</li>
                    )
                   })
                }

            </ul>
        </div>
    </div>
  )
}
```

e. DONE !!!!

19. Sort product functionality
    a. Create SortProduct.js file ➔ create component using rfc
    b. Create sortProductByPrice() function inside Product.js

```
 function sortProductByPrice(sortingType)
{
  let data = [...filteredProducts];
  if(sortingType==='asc')
  {
      data.sort((obj1, obj2)=> obj1.price-obj2.price)
  }
  else
  {
    data.sort((obj1, obj2)=> obj2.price-obj1.price)
  }

  setFilteredProducts(data)
}
```

    c. Render above component inside Product.js



```
export default function Product() {
  return (
    <div className='container mt-3'>
      <div className='d-flex justify-content-between'>
      <FilterProduct onFilterProductByCategory={filterProductByCategory
      <SortProduct onSortProductByPrice={sortProductByPrice}/>
      <input placeholder='enter product name'></input>
      </div>
```

d. Final code for SortProduct component

```
import React from 'react'

export default function SortProduct(props) {
  return (
    <div className='mb-3'>
        <div className="dropdown">
            <button className="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown"
aria-expanded="false">
                Sort Product
            </button>
            <ul className="dropdown-menu">
                <li> <button className='btn' onClick={()=>{props.onSortProductByPrice("asc")}}>Low to
High</button> </li>
                <li> <button className='btn' onClick={()=>{props.onSortProductByPrice("desc")}}>High to
Low</button> </li>
            </ul>
        </div>
    </div>
  )
}
```

e. DONE !!!

20. Handling form in react

We are going to use library – React Hook Form to handle form in react

```
^C
C:\My WorkSpace\Bootcamp\react 5-3-25\react\bootcamp-app>npm install react-hook-form
```

There are 3 important things in form that we have to manage –

Individual field with its data, entire form data, and errors on each field

a. Install RHF as - npm install react-hook-form
b. Code for Contact.js

```
import React from 'react'
import { useForm } from 'react-hook-form'

export default function Contact() {
  let {register, handleSubmit, formState}=useForm()
  function collectFormData(formData)
  {
    console.log(formData);
  }
  return (
    <div className='container d-flex justify-content-center mt-3 border border-3 p-5'>
      <form className='w-50' onSubmit={handleSubmit(collectFormData)}>
        <div className="mb-3">
          <label htmlFor="username" className="form-label">User Name:</label>
          <input type="text" className="form-control" id="username"
          {...register('username',{required:true,minLength:4,maxLength:10})}/>
        </div>
        <p className='text-danger'>
          {formState.errors && formState.errors.username &&
formState.errors.username.type=='required' && 'UserName is required'}
          {formState.errors && formState.errors.username &&
formState.errors.username.type=='minLength' && 'min 4 characters required in username'}
          {formState.errors && formState.errors.username &&
formState.errors.username.type=='maxLength' && 'max 10 characters required in username'}
        </p>
        <div className="mb-3">
          <label htmlFor="userage" className="form-label">User Age:</label>
          <input type="number" className="form-control" id="userage"
          {...register('userage',{required:true,min:18,max:80})}/>
        </div>
        <p className='text-danger'>
          {formState.errors && formState.errors.userage &&
formState.errors.userage.type=='required' && 'UserAge is required'}
          {formState.errors && formState.errors.userage &&
formState.errors.userage.type=='min' && 'min age 18 required'}
          {formState.errors && formState.errors.userage &&
formState.errors.userage.type=='max' && 'max age 80 required'}
        </p>
        <button type="submit" className="btn btn-primary">Submit</button>
      </form>
    </div>
  )
}
```

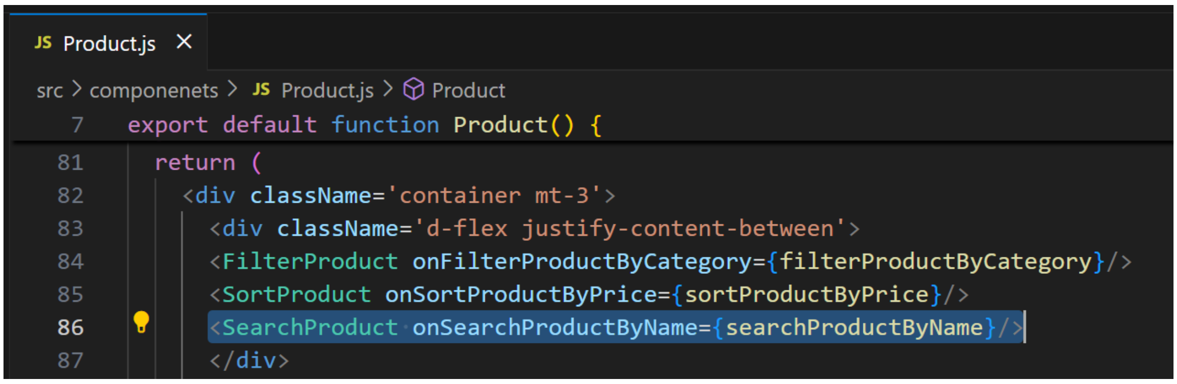
c. DONE!

21. Implementing search functionality
    data.filter(name => {return name.toLowerCase().includes('A'.toLowerCase())})

   a. Create SearchProduct.js file ➔ create component using rfc
   b. Create function searchProductByName() in Product.js

```
  function searchProductByName(productName)
  {

    let data = [...filteredProducts];
    let newdata = data.filter(product => {return
product.title.toLowerCase().includes(productName.toLowerCase())})
    setFilteredProducts(newdata)
  }
```

   c. Render SearchProduct component in Product.js



```
export default function Product() {
  return (
    <div className='container mt-3'>
      <div className='d-flex justify-content-between'>
      <FilterProduct onFilterProductByCategory={filterProductByCategory}/>
      <SortProduct onSortProductByPrice={sortProductByPrice}/>
      <SearchProduct onSearchProductByName={searchProductByName}/>
      </div>
```

   d. Code for SearchProduct.js

```
import React from 'react'
import { useForm } from 'react-hook-form';

export default function SearchProduct(props) {
    let {register, handleSubmit, formState}=useForm()
      function collectFormData(formData)
      {
        console.log(formData);
        props.onSearchProductByName(formData.productName)
      }
  return (
    <div >
      <form className='d-flex' onSubmit={handleSubmit(collectFormData)}>
        <div className='me-3'>
          <input type="text" className="form-control" id="productName"
          {...register('productName',{required:true,minLength:4,maxLength:10})}/>
          <span className='text-danger'>
          {formState.errors && formState.errors.productName &&
formState.errors.productName.type=='required' && 'productName is required'}
          {formState.errors && formState.errors.productName &&
formState.errors.productName.type=='minLength' && 'min 4 characters required in productName'}
          {formState.errors && formState.errors.productName &&
formState.errors.productName.type=='maxLength' && 'max 10 characters required in productName'}
        </span>
        </div>

        <button type="submit" className="btn btn-primary">Search</button>
      </form>
    </div>
  )
}
```

   e. DONE !

22. Context