INSTALL REACT ROUTER

npm install history react-router-dom@next

WRAP <APP/> IN INDEX.JS UNDER ROUTER TAG

import {BrowserRouter as Router} from 'react-router-dom';

ReactDOM.render(

  <React.StrictMode>

    <Router>

      <App />

    </Router>

  </React.StrictMode>,

  document.getElementById('root')

);

CREATE DIFFERENT COMPONENTS AND ROUTES:

import { Routes, Route } from 'react-router-dom';

//importing components

import { Header } from './components/Header';

import { ProductDetails } from './components/ProductDetails';

import { Products } from './components/Products';

import { Home } from './components/Home';

function App() {

  return (

    <div>

      <Header />

      <Routes>

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

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

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

          <Route path=":productId" element={<ProductDetails />}></Route>

        </Route>

      </Routes>

    </div>

  );

MAKE SIMPLE NAV BAR IN HEADER.JS:

import React from 'react'

import { Link } from 'react-router-dom'

export const Header = () => {

    return (

        <nav>

            <Link to="/">Home</Link>

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

        </nav>

    )

}

SOME PRODUCT:

Now in order to have some data, we create a show object in products.js and pass productId, name and image to each object. Then to get data, we iterate over all of the data in shoe object using Object.entries.map which takes productId and destructure name and image of each data so we can use.

export const Products = () => {

    const shoes = {

        "grey-simple-men-shoe": {

            name: "Grey Simple Men Shoe",

            img: greymen

        },

        "grey-stylish-men-shoe": {

            name: "Grey Stylish Men Shoe",

            img: grey

        },

        "dark-blue-leather-sneakers": {

            name: "Dark Blue-Leather Sneakers",

            img: darkblue

        },

        "pink-sneaker": {

            name: "Pink Sneaker",

            img: pink

        },

    }

    return (

        <div>

            <h1>Products page</h1>

            <ul>

                {Object.entries(shoes).map(([productId, {name, img}]) =>

                    (

                       <li key={productId}>

                           <h2>{name}</h2>

                           <img src={img} alt={name}/>

                       </li>

                    )

                )}

            </ul>

        </div>

    )

}

Now the below part is like it take productId as key so that we can use name and img of each productId

                       <li key={productId}>

                           <h2>{name}</h2>

                           <img src={img} alt={name}/>

                       </li>

Now after that, to show products on the main page and and to show details of the product whenever someone click on the product page, we do some of the changes.

First we list all of the products in ProductIndex.js. Next when someone click on any of the product, all details related to that product will be provided in ProductDetails.js. But how product details can be shown on same product page, we use <Outlet /> in our main product page. So summarizing

Route for main product page. In that, one for the productIndex which contains all of the products, and it is generally like a product home page which contains all of the product. Then in that, route for the ProductDetails which contain details of every single product. So till now;

Main products.js

import React from 'react'

import { Outlet } from 'react-router-dom';

export const Products = () => {

    return (

        <div>

            <h1>Products page</h1>

            <Outlet />

        </div>

    )

}

ProductIndex.js:

import React from 'react'

import { Link } from 'react-router-dom'

//importing some of the images from the system

import greymen from "./images/grey.jpg"

import grey from "./images/4.jpg";

import darkblue from "./images/5.jpg";

import pink from "./images/pink shoes.jpeg"

export const ProductIndex = () => {

    const shoes = {

        "grey-simple-men-shoe": {

            name: "Grey Simple Men Shoe",

            img: greymen

        },

        "grey-stylish-men-shoe": {

            name: "Grey Stylish Men Shoe",

            img: grey

        },

        "dark-blue-leather-sneakers": {

            name: "Dark Blue-Leather Sneakers",

            img: darkblue

        },

        "pink-sneaker": {

            name: "Pink Sneaker",

            img: pink

        },

    }

    return (

        <div>

             <ul>

                {Object.entries(shoes).map(([productId, {name, img}]) =>

                    (

                        <li key={productId}>

                            <Link to={productId}>

                                <h2>{name}</h2>

                                <img src={img} alt={name}/>

                             </Link>

                         </li>

                    )

                )}

            </ul>

        </div>

    )

}

Here in li tag, we have also place link tag, so when ever some click on the product, it redirects to product id which is ultimately the ProductDetails.js

ProductDetails.js(not made till now):

import React from 'react'

export const ProductDetails = () => {

    return (

        <div>

            Product Details

        </div>

    )

}

App.js

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

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

          <Route path=":productId" element={<ProductDetails />}></Route>

        </Route>

Now Its time to handle product details.

As we want to show product details on the same page, so we use useParams which handle all our parameters(productId) in this case, so when we click on any of the product, it redirects us to /products/productId and the details of that product can be find on productDetails. So we just need to use useParams to handle productId and information related to productId like its name img and we can display them and other stuff like that.

import React from 'react'

//importing some of the images from the system

import greymen from "./images/grey.jpg"

import grey from "./images/4.jpg";

import darkblue from "./images/5.jpg";

import pink from "./images/pink shoes.jpeg"

import { useParams } from 'react-router-dom';

export const ProductDetails = () => {

    const shoes = {

        "grey-simple-men-shoe": {

            name: "Grey Simple Men Shoe",

            img: greymen

        },

        "grey-stylish-men-shoe": {

            name: "Grey Stylish Men Shoe",

            img: grey

        },

        "dark-blue-leather-sneakers": {

            name: "Dark Blue-Leather Sneakers",

            img: darkblue

        },

        "pink-sneaker": {

            name: "Pink Sneaker",

            img: pink

        },

    };

    const {productId} = useParams();

    const product = shoes[productId];

    const {name, img} = product;

    return (

        <div>

            <h4>{name}</h4>

            <img src={img} alt={name}/>

        </div>

    )

}

CREATE DUMMY ABOUT US AND CONTACT US.js:

Put data later

FOR NAVBAR:

Watch this tutorial: <https://www.youtube.com/watch?v=fL8cFqhTHwA>

First create button.js and button.css in components folder to handle buttons.

Then create Navbar folder in components folder. With in the Navbar folder, created Navbar.js, Navbar.css and MenuItems.js

In MenuItems, placed all the items used in Navbar in the form of array, each item is the object containing name, url and the className. We can give any name to key we want. Then in Navbar.js, we will create Navbar.

For that we first made left side heading, then icon, and then use unordered list <ul> to create menu links

As we have created the array of items in MenuItems, so for each item, we need to iterate over the MenuItems using map function and we pass item and index to it, so we can extract each items information like its title url and className(cName) using item.title, item.url and item.cName. And we know <ul> uses <li> so we used them after iterating over MenuItems using map, and we used index of each item to extract or use each item, then to create links, we used <a href></a> and pass className, url and title to it with the help of item.

ICONS AND FONTS:

Use montserrate font, write google font montserrate in google, click first link and select regular

<link href="https://fonts.googleapis.com/css2?family=**Montserrat**&display=swap" rel="stylesheet">

For icons, create font awesome account, clicked on font awesome cdn and use this link and place in index.html

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.1/css/all.css" integrity="sha384-vp86vTRFVJgpjF9jiIGPEEqYqlDwgyBgEF109VFjmqGmIY/Y4HV4d3Gp2irVfcrp" crossorigin="anonymous">

For bar and cross of navbar:

 state = { clicked: false};

     handleClick = () => {

         this.setState({clicked:!this.state.clicked})

     };

    render () {

        return (

            <nav className="NavbarItems">

                <h1 className="navbar-logo">React <i className="fab fa-react"></i></h1>

                <div className="menu-icon" onClick={this.handleClick}>

                    <i className={this.state.clicked ? 'fas fa-times': 'fas fa-bars'}></i>

              </div>

We created a state and set clicked to false. Then we create function to handle click, so when someone click, it changes. Like if it is 1, then 0 or if 0 then 1. In this case, by default click is false. So on false or when no one clicks on it, we see bars. When onClick(someone click), we call handleClick function that see if clicked true, then changes bars with cross and if not clicked, then maintain bars.

Then we made code for button, where we define different styles and size, and in return it check if checkButtonStyles available(means in css) then it uses this other wise it use STYLES[0]-> which is btn—primary and same with size. Then do button css. In the end did Navbaar css.

This is the brian repo whose tutorial for navbar I followed.

<https://github.com/briancodex/react-navbar-v1/blob/master/src/components/Navbar/Navbar.css>