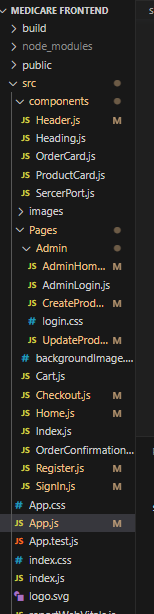
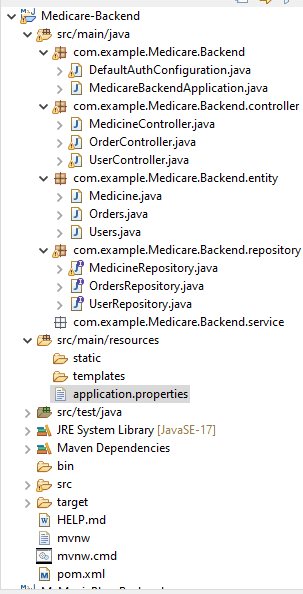
**Capstone Project**

**Medicare**

**Source code**

**Project Structure**

**FrontEnd BackEnd**

**Medicare-FrontEnd**

Package.json

{

  "name": "Medicare-FrontEnd",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@reach/router": "^1.3.4",

    "@testing-library/jest-dom": "^5.16.4",

    "@testing-library/react": "^13.3.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.4.0",

    "bootstrap": "^5.2.0",

    "cors": "^2.8.5",

    "formik": "^2.2.9",

    "react": "^18.2.0",

    "react-bootstrap": "^2.4.0",

    "react-dom": "^18.2.0",

    "react-filter-search": "^1.1.11",

    "react-icons": "^4.4.0",

    "react-lightbox-component": "^1.2.1",

    "react-phone-input-2": "^2.15.1",

    "react-scripts": "5.0.1",

    "react-switch": "^7.0.0",

    "react-use-cart": "^1.13.0",

    "web-vitals": "^2.1.4",

    "yup": "^1.1.1"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

App.js

import React from "react";

import './App.css';

import 'bootstrap/dist/css/bootstrap.min.css';

import Header from './components/Header';

import { Router } from "@reach/router";

//Pages

import Home from './Pages/Home';

import Cart from './Pages/Cart';

import SignIn from "./Pages/SignIn";

import Register from "./Pages/Register";

import Index from "./Pages/Index";

import Checkout from "./Pages/Checkout";

import OrderConfirmation from "./Pages/OrderConfirmation";

import AdminLogin from "./Pages/Admin/AdminLogin";

import AdminHome from "./Pages/Admin/AdminHome";

import CreateProduct from "./Pages/Admin/CreateProduct";

import UpdateProduct from "./Pages/Admin/UpdateProduct";

function App() {

  return (

    <main className="bg-light-2" style={{ height: '100vh', overflowY: 'auto'}}>

      <Router>

        <Index path="/"/>

        <Home path="/home" />

        <SignIn path="sign-in"/>

        <Register path="register"/>

        {/\* <ProductDetails path="product-details/:productId"/> \*/}

        <Cart path="/cart" />

        <Checkout path="/checkout"/>

        <OrderConfirmation path="/order"/>

        {/\* Admin Components \*/}

        <AdminLogin path="/admin\_login"/>

        <AdminHome path="/admin\_home"/>

        <CreateProduct path="/createProduct"/>

        <UpdateProduct path="/updateProduct/:id"/>

      </Router>

    </main>

  );

}

export default App;

components/Header.js

import React, { useContext, useState, useEffect } from 'react';

import { Container, Navbar, Nav } from 'react-bootstrap';

import { BiSun, BiMoon, BiCart } from 'react-icons/bi';

import { VscAccount } from 'react-icons/vsc';

import { Link, useNavigate } from "@reach/router";

import { useCart } from "react-use-cart";

import {Button} from 'react-bootstrap';

const Header = () => {

  const [session, SetSession] = useState(sessionStorage.getItem("name"));

  const navigate=useNavigate();

  const navmenu = () => {

    if (session === null) {

      return (

        <>

        <Link to="/sign-in" className={`nav-link `}>

          Sign in

        </Link>

        <Link

            to="/cart"

            className={` d-flex align-items-center`}

          >

            <BiCart size="2rem" />

            {!isEmpty && <span style={{ position: 'relative', left: '-21px', top: '-18px' }}>{totalItems}</span>}

            <span style={{ marginLeft: !isEmpty ? '-13px' : 0 }}>&nbsp;Cart</span>

          </Link>

          <Link className='btn btn-success' style={{padding:"6px",margin:"0px 19px"}} to='/admin\_login'>

          Admin Login

          </Link>

        </>

      );

    }

    else {

      return (

        <>

          <Link to="home" className={`nav-link `}>

            Signed in As:<strong> {session}</strong>

          </Link>

          <Link

            to="/cart"

            className={` d-flex align-items-center`}

          >

            <BiCart size="2rem" />

            {!isEmpty && <span style={{ position: 'relative', left: '-21px', top: '-18px',color:'black' }}>{totalItems}</span>}

            <span style={{ marginLeft: !isEmpty ? '-13px' : 0 }}>&nbsp;Cart</span>

          </Link>

          <button

           onClick={()=>{

            sessionStorage.removeItem("name");

            navigate("/sign-in");

        }}

          className={`mx-3 btn btn-danger `}>

            <VscAccount size="1.8rem" />

            &nbsp;Logout

          </button>

        </>

      );

    }

  }

  const {

    isEmpty,

    totalItems,

  } = useCart();

  return (

    <Navbar collapseOnSelect expand="md"

      variant={'light'}

      className={'texy border-bottom'}

      style={{ width: '100%', position: 'fixed', zIndex: 100 ,top:"0px", color:'black',   background: "cadetblue"}}

    >

      <Container>

        <Link to="/home">

          <Navbar.Brand className={'text-success'} style={{fontFamily:'fantasy'}}>

            Medicare

          </Navbar.Brand>

        </Link>

        <Navbar.Toggle aria-controls="basic-navbar-nav" />

        <Navbar.Collapse id="basic-navbar-nav">

          <Nav className="ms-auto">

            {

              navmenu()

            }

          </Nav>

        </Navbar.Collapse>

      </Container>

    </Navbar>

  );

};

export default Header;

components/ProductCard.js

import React from 'react';

import { Button, Card} from 'react-bootstrap';

import { useCart } from 'react-use-cart';

import { BsCartPlus } from 'react-icons/bs';

import { Link } from  "@reach/router";

const ProductCard = (props) => {

    let { image, price, name,seller,description, id} = props.data;

    const { addItem } = useCart();

    const addToCart = () =>{

        addItem(props.data);

    }

    return (

        <Card

            style={{ width: '18rem', height: 'auto'}}

            className={`'bg-dark text-black'  p-0 overflow-hidden shadow mx-auto mb-4`}

        >

            {/\* <Link to={`/product-details/${id}`}>

                <div style={{ background: 'white', height: '15rem', overflow: 'hidden', display: 'flex',

                justifyContent: 'center', alignItems: 'center', marginBottom: 'inherit' }}>

                    <div style={{ width: '9rem'}}>

                        <Card.Img variant="top" src={image} className="img-fluid" />

                    </div>

                </div>

            </Link> \*/}

            <Card.Body>

                <Card.Title style={{ textOverflow: 'ellipsis', overflow: 'hidden', whiteSpace: 'nowrap'}}>

                   <u><b>Name:</b></u> {name}

                </Card.Title>

                <Card.Title>

                <u><b>Price:</b></u> &nbsp; Rs. <span className="">{price}</span>

                </Card.Title>

                <Card.Title>

                <u><b>Seller:</b></u>&nbsp;  <span className="">{seller}</span>

                </Card.Title>

                <Card.Title>

                <u><b>Description:</b></u> <br></br>  <span className="">{description}</span>

                </Card.Title>

                <Button

                    onClick={()=> addToCart()}

                    className={`bg-light-primary  d-flex align-item-center m-auto border-0`}

                >

                    <BsCartPlus size="1.8rem" />

                    Add to cart

                </Button>

            </Card.Body>

        </Card>

    );

};

export default ProductCard;

pages/Home.js

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

import { Container, Row, Col, InputGroup, FormControl } from 'react-bootstrap';

import { BiSearch } from 'react-icons/bi';

import SearchFilter from 'react-filter-search';

import ProductCard from '../components/ProductCard';

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

const Home = () => {

    const [searchInput, setSearchInput] = useState('');

    const [productData, setProductData] = useState([]);

    const [Final\_data,SetFinalData]=useState([]);

    const name=sessionStorage.getItem("name");

    // sessionStorage.removeItem("name");

    const getResponse= async ()=>{

        const res = await fetch('http://18.204.215.152:8080/getProducts')

                          .then(res=> res.json());

                          setProductData(await res);

    }

    useEffect(()=>{

        getResponse();

    },[]);

    return (

        <>

        <Header/>

        <img src="https://www.hopkinsmedicine.org/-/media/images/health/3\_-wellness/living-with-a-chronic-disease/help-for-managing-multiple-medications-hero.ashx?h=500&iar=0&mh=500&mw=1300&w=1297&hash=122DB3C1DCAD683F2F7CAE139A67613C" style={{position:'absolute', width:"100%",opacity:'.7'}} height={657    } />

        <Container className="py-4">

            <Row className="justify-content-center">

                <Col xs={10} md={7} lg={6} xl={4} className="mb-3 mx-auto text-center">

                    <h1 className={'text-black my-5'}>Search products</h1>

                    <InputGroup className="mb-3">

                        <InputGroup.Text className={ 'bg-light text-light-primary'}>

                            <BiSearch size="2rem" />

                        </InputGroup.Text>

                        <FormControl

                            placeholder="Search"

                            value={searchInput}

                            onChange={(e)=> setSearchInput(e.target.value)}

                            className={ 'bg-light text-black'}

                        />

                    </InputGroup>

                </Col>

                <SearchFilter

                    value={searchInput}

                    data={productData}

                    renderResults={results =>(

                        <Row className="justify-content-center">

                            {results.map((item, i)=>(

                                <ProductCard data={item} key={i} />

                            ))}

                        </Row>

                    )}

                />

            </Row>

        </Container>

        </>

    );

};

export default Home;

pages/Sign-in.js

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

import { Container, Row, Col, Button, Form, Spinner, InputGroup} from 'react-bootstrap';

import { Link, useNavigate } from "@reach/router";

//icons

import { AiOutlineUser } from 'react-icons/ai';

import { VscKey } from 'react-icons/vsc';

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

import "./backgroundImage.css";

const SignIn = () => {

    const [loading, setLoading] = useState(false);

    const navigate = useNavigate();

    const [users,SetUsers]=useState([])

    async function getResponse(){

        const res = await fetch('http://18.204.215.152:8080/getUsers')

                          .then(res=> res.json());

                          SetUsers(await res);

                          console.log(res);

    }

    useEffect(()=>{

        getResponse();

    },[])

    const handleSubmit = (event)=>{

        const form = event.currentTarget;

        event.preventDefault();

        const username = form.username.value;

        const password = form.password.value;

        if(username && password){

            users.map((item,index)=>{

                if(item.number===username && item.password===password){

                    sessionStorage.setItem("name", item.name)

                    sessionStorage.setItem("number", item.number)

                    sessionStorage.setItem("email", item.email)

                    sessionStorage.setItem("address", item.address)

                    navigate("/home");

                }

            })

            // fetch('https://fakestoreapi.com/auth/login',{

            //     method: 'POST',

            //     headers: {

            //         'Content-Type': 'application/json'

            //     },

            //     body:JSON.stringify({

            //         username: username,

            //         password: password

            //     })

            // }).then(res=>res.json())

            // .then(json=>sessionStorage.setItem("token", json.token))

            // .catch(error=> console.error(error))

            // .finally(()=>{

            //     setLoading(false);

            //     navigate('/', {replace: true})

            //     alert('Login successfully');

            // })

        }

    }

    return (

        <div className='containerUnique2'>

          <Header/>

                <Container className=" py-5 mt-5">

            <Row className="justify-content-center mt-5">

                <Col xs={11} sm={10} md={8} lg={4} className={`p-4 rounded text-black bg-light`}>

                    <h1 className={`text-center border-bottom pb-3 'text-light-primary'`}>

                        Sign in

                    </h1>

                    <Form onSubmit={handleSubmit}>

                        <InputGroup className="mb-4 mt-5">

                            <InputGroup.Text>

                                <AiOutlineUser size="1.8rem" />

                            </InputGroup.Text>

                            <Form.Control name="username" type="text" placeholder="Username" minLength={3} required />

                        </InputGroup>

                        <InputGroup className="mb-4">

                            <InputGroup.Text>

                                <VscKey size="1.8rem" />

                            </InputGroup.Text>

                            <Form.Control name="password" type="password" placeholder="Password"  required />

                        </InputGroup>

                        <Button

                            type="submit"

                            className={`bg-light-primary m-auto d-block`}

                            disabled={loading}

                            style={{border: 0}}

                        >

                        {loading?

                            <>

                                <Spinner

                                    as="span"

                                    animation="grow"

                                    size="sm"

                                    role="status"

                                    aria-hidden="true"

                                />

                                &nbsp;Loading...

                            </> : 'Sign in'

                        }

                        </Button>

                        <Form.Group className="mt-3 text-center">

                            <Form.Text className="text-muted fw-bold">

                                New to Medicare?

                            </Form.Text>

                            <Row className="py-2 border-bottom mb-3"/>

                            <Link to='/register' className="btn btn-info rounded-0">

                                Create your Medicare account

                            </Link>

                        </Form.Group>

                    </Form>

                </Col>

            </Row>

       </Container>

       </div>

    );

};

export default SignIn;

pages/Register.js

import React, { useState } from 'react';

import { Container, Row, Col, Button, Form, Spinner, InputGroup } from 'react-bootstrap';

import PhoneInput from 'react-phone-input-2';

import 'react-phone-input-2/lib/high-res.css';

import axios from 'axios';

import { useNavigate } from '@reach/router';

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

import "./backgroundImage.css";

const Register = () => {

    const [loading, setLoading] = useState(false);

    const [number, setNumber] = useState(null);

    const navigate=useNavigate();

    const handleSubmit = (event) => {

        const form = event.currentTarget;

        event.preventDefault();

        const password = form.password.value;

        const firstname = form.firstName.value;

        const lastname = form.lastName.value;

        const email = form.email.value;

        const address=form.address.value;

        if (password && firstname && lastname && email && number) {

            setLoading(true);

            const data = {

                "name":firstname+" "+lastname,

                "email":email,

                "number":number,

                "address":address,

                "password":password

             };

            axios.post('http://18.204.215.152:8080/addUser', data)

                .then(response => alert("data addred"));

            setLoading(false);

            navigate('/sign-in')

        }

    }

    return (

        <div className='containerUnique3'>

        <Header/>

        <Container className="py-3">

            <Row className="justify-content-center mt-5">

                <Col xs={11} sm={10} md={8} lg={4} className={`p-4 rounded text-black bg-light`}>

                    <h1 className={`text-center border-bottom pb-3  'text-light-primary'`}>

                        Create Account

                    </h1>

                    <Form onSubmit={handleSubmit}>

                        <Row>

                            <Form.Group className="mb-3 col-lg-6">

                                <Form.Label>First name</Form.Label>

                                <Form.Control name="firstName" type="text" placeholder="First name" required />

                            </Form.Group>

                            <Form.Group className="mb-3 col-lg-6">

                                <Form.Label>Last name</Form.Label>

                                <Form.Control name="lastName" type="text" placeholder="Last name" required />

                            </Form.Group>

                        </Row>

                        <Form.Group className="mb-3">

                            <Form.Label>Email</Form.Label>

                            <Form.Control name="email" type="email" placeholder="Email" required />

                        </Form.Group>

                        <Form.Group className="mb-3">

                            <Form.Label>Mobile number</Form.Label>

                            <PhoneInput

                                country={'in'}

                                value={number}

                                onChange={phone => setNumber(phone)}

                                className="text-dark"

                            />

                            <Form.Group className="mb-3">

                                <Form.Label>Address</Form.Label>

                                <Form.Control name="address" type="text" placeholder="Address"  minLength={3} required />

                            </Form.Group>

                        </Form.Group>

                        <Form.Group className="mb-3">

                            <Form.Label>Password</Form.Label>

                            <Form.Control name="password" type="password" placeholder="Password" minLength={6} required />

                        </Form.Group>

                        <Button

                            type="submit"

                            className={`'bg-light-primary' m-auto d-block`}

                            disabled={loading}

                            style={{ border: 0 }}

                        >

                            {loading ?

                                <>

                                    <Spinner

                                        as="span"

                                        animation="grow"

                                        size="sm"

                                        role="status"

                                        aria-hidden="true"

                                    />

                                    &nbsp;Loading...

                                </> : 'Continue'

                            }

                        </Button>

                    </Form>

                </Col>

            </Row>

        </Container>

        </div>

    );

};

export default Register;

pages/OrderConformation.js

import Button from 'react-bootstrap/Button';

import Col from 'react-bootstrap/Col';

import Form from 'react-bootstrap/Form';

import InputGroup from 'react-bootstrap/InputGroup';

import Row from 'react-bootstrap/Row';

import \* as formik from 'formik';

import \* as yup from 'yup';

import { Container, Table } from 'react-bootstrap';

import { useCart } from 'react-use-cart';

import PhoneInput from 'react-phone-input-2';

import { BsCartCheck, BsCartX } from 'react-icons/bs';

import { Link, useLocation, useNavigate } from '@reach/router';

import { useState } from 'react';

import Modal from 'react-bootstrap/Modal'

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

import { useEffect } from 'react';

// import { CurrencyDollar } from 'react-bootstrap-icons'

function OrderConfirmation() {

  const navigate=useNavigate();

  const location=useLocation();

  const access=location.state;

  const date=new Date();

let day = date.getDate();

let month = date.getMonth() + 1;

let year = date.getFullYear();

const number=sessionStorage.getItem('number');

const email=sessionStorage.getItem('email');

const address=sessionStorage.getItem('address');

// This arrangement can be altered based on how we want the date's format to appear.

let currentDate = `${day}/${month}/${year}`;

  console.log(access);

  useEffect(()=>{

    if (sessionStorage.getItem("name") === null) {

      navigate("/sign-in");

  }

},[])

  return (

    <div>

      <Header/>

      <div className="container mt-5 mb-5">

                    <div className="row d-flex justify-content-center">

                        <div className="col-md-8">

                            <div className="card">

                                <div className="invoice p-5">

                                    <h5>Your order Confirmed!</h5>

                                    <span className="font-weight-bold d-block mt-4">Hello, </span>

                                    <span>You order has been confirmed and will be shipped in next two days!</span>

                                    <div className="payment border-top mt-3 mb-3 border-bottom table-responsive">

                                        <table className="table table-borderless">

                                            <tbody>

                                                <tr>

                                                    <td>

                                                        <div className="py-2">

                                                            <span className="d-block text-muted">Order Date</span>

                                                            <span>{currentDate}</span>

                                                        </div>

                                                    </td>

                                                    <td>

                                                        <div className="py-2">

                                                            <span className="d-block text-muted">Contact Number</span>

                                                            <span>{number}</span>

                                                        </div>

                                                    </td>

                                                    <td>

                                                        <div className="py-2">

                                                            <span className="d-block text-muted">Contact Email</span>

                                                            <span>{email}</span>

                                                        </div>

                                                    </td>

                                                    <td>

                                                        <div className="py-2">

                                                            <span className="d-block text-muted">Shiping Address</span>

                                                            <span>{address}</span>

                                                        </div>

                                                    </td>

                                                </tr>

                                            </tbody>

                                        </table>

                                    </div>

                                    <div className="product border-bottom table-responsive">

                <thead>

                        <tr>

                            <th>Id</th>

                            <th>Name</th>

                            <th>Seller</th>

                            <th>Quantity</th>

                            <th>Description</th>

                            <th>Price</th>

                        </tr>

                    </thead>

                    <tbody>

                        {access.Items.map((item, index)=>{

                            return(

                              <tr key={index}>

                              <td>{item.id}</td>

                              <td>

                                  <h6 style={{ whiteSpace: 'nowrap', width: '10rem', overflow: 'hidden', textOverFlow: 'ellipsis'}}>

                                      {item.name}

                                  </h6>

                              </td>

                              <td><h6 style={{ whiteSpace: 'nowrap', width: '8rem', overflow: 'hidden', textOverFlow: 'ellipsis'}}>

                                      {item.seller}

                                  </h6></td>

                              <td>

                              <h6 style={{ whiteSpace: 'nowrap', width: '8rem', overflow: 'hidden', textOverFlow: 'ellipsis'}}>

                                      {item.quantity}

                                  </h6>

                              </td>

                              <td> <h6 style={{ whiteSpace: 'nowrap', width: '9.5rem', overflow: 'hidden', textOverFlow: 'ellipsis'}}>

                                      {item.description}

                                  </h6></td>

                                <td><h6 style={{ whiteSpace: 'nowrap', width: '2rem', overflow: 'hidden', textOverFlow: 'ellipsis'}}>

                                      {item.price}

                                  </h6></td>

                          </tr>

                            )

                        })}

                    </tbody>

                                    </div>

                                    <div className="row d-flex justify-content-end">

                                        <div className="col-md-5">

                                            <table className="table table-borderless">

                                                <tbody className="totals">

                                                    <tr>

                                                        <td>

                                                            <div className="text-left">

                                                                <span className="text-muted">Total</span>

                                                            </div>

                                                        </td>

                                                        <td>

                                                            <div className="text-right">

                                                                <span>Rs/-{access.total}</span>

                                                            </div>

                                                        </td>

                                                    </tr>

                                                </tbody>

                                            </table>

                                        </div>

                                    </div>

                                    <p>We will be sending shipping confirmation email when the item shipped successfully!</p>

                                    <p className="font-weight-bold mb-0">Thanks for Choosing us!</p>

                                    <span>Medicare</span>

                                </div>

                                <Link to="/" className='btn btn-success'>Back to Home</Link>

                            </div>

                        </div>

                    </div>

                </div>

      {/\* <Container className='bg-light py-3 border rounded my-5 ' style={{position:'relative',top:'82px',padding:'110px'}}>

        <h3 className='text-center '> Order Summary</h3>

      </Container> \*/}

    </div>

  )

}

export default OrderConfirmation;

pages/index.js

import Card from 'react-bootstrap/Card';

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

import { Button } from 'react-bootstrap';

import {AiOutlineFileSearch} from "react-icons/ai"

import { Link } from '@reach/router';

const Index=()=> {

  return (

    <>

        <Header/>

    <Card className="bg-white text-dark" style={{position:'relative',top:"54px" , textAlign:'center'}}>

      <Card.Img src="https://www.hopkinsmedicine.org/-/media/images/health/3\_-wellness/living-with-a-chronic-disease/help-for-managing-multiple-medications-hero.ashx?h=500&iar=0&mh=500&mw=1300&w=1297&hash=122DB3C1DCAD683F2F7CAE139A67613C" style={{position:'relative', width:"100%" }} height={600}  alt="Card image" />

      <Card.ImgOverlay>

        <Card.Title><b className='text-dark h1' style={{fontFamily:'serif',fontWeight:'bolder'}}>Medicare  <br></br></b> A Place to Cure All Your Diseases</Card.Title>

        <Card.Text>

          This is a Medical E-commerce Websit from which you can Order all your Medicines Online

        </Card.Text>

        <Link to="/home" className='btn btn-primary '><AiOutlineFileSearch size="1.8rem"/>  Search Medicines</Link>

      </Card.ImgOverlay>

    </Card>

    </>

  );

}

export default Index;

// width: 87%;

//     position: relative;

//     left: 6%;

pages/Checkout.js

import Button from 'react-bootstrap/Button';

import Col from 'react-bootstrap/Col';

import Form from 'react-bootstrap/Form';

import InputGroup from 'react-bootstrap/InputGroup';

import Row from 'react-bootstrap/Row';

import \* as formik from 'formik';

import \* as yup from 'yup';

import { Container, Table } from 'react-bootstrap';

import { useCart } from 'react-use-cart';

import PhoneInput from 'react-phone-input-2';

import { BsCartCheck, BsCartX } from 'react-icons/bs';

import { Link, useNavigate } from '@reach/router';

import { useState } from 'react';

import axios from 'axios';

import Modal from 'react-bootstrap/Modal'

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

// import { CurrencyDollar } from 'react-bootstrap-icons'

import "./backgroundImage.css";

import { useDeferredValue } from 'react';

import { useEffect } from 'react';

function Checkout() {

  const [data\_items,SetItems]=useState([]);

  const [totalPrice,SetTotalPrice]=useState(0);

  useEffect(()=>{

    if (sessionStorage.getItem("name") === null) {

      navigate("/sign-in");

  }

  },[])

  const navigate=useNavigate();

  const [payment,SetPayment]=useState("");

  const {

    isEmpty,

    items,

    cartTotal,

    updateItemQuantity,

    removeItem,

    emptyCart,

} = useCart();

  const [milestone, setMilestone] = useState({

    name: sessionStorage.getItem("name"),

    number:sessionStorage.getItem("number") ,

    email: sessionStorage.getItem("email"),

    address: sessionStorage.getItem("address")

  })

  const handleSubmit = () => {

    if(milestone.name==="" && milestone.number==="" && milestone.email==="" && milestone.address==="" ){

      alert('Invalid');

    }

    else{

      const date = new Date();

let day = date.getDate();

let month = date.getMonth() + 1;

let year = date.getFullYear();

// This arrangement can be altered based on how we want the date's format to appear.

let currentDate = `${day}-${month}-${year}`;

      var items\_list="";

      items.map((item,index)=>{

        let a=item.id+"-"+item.quantity

        items\_list=items\_list.concat(",",a)

      });

      const data={

        "user\_name":milestone.name,

        "user\_number":milestone.number,

        "user\_email":milestone.email,

        "user\_address":milestone.address,

        "items":items\_list.slice(1),

        "date":currentDate

      }

      axios.post('http://18.204.215.152:8080/addOrder', data)

                .then(response => alert("Order Placed"));

      SetItems(items);

      SetTotalPrice(cartTotal);

      console.log(data\_items)

      emptyCart();

      navigate("/order",{state:{total:totalPrice,Items:data\_items}});

    }

  }

  const setpayment = () => {

    if(payment==='1'){

      return (

        <h5 className="text-center text-danger mt-4">\*\*Selected Cash on Delivery as Payment Method\*\*</h5>

      );

    }

    else if(payment==='2'){

      return (

        <>

        <h4 className='text-center mt-4 mb-4'><u>Enter Debit Card Details</u></h4>

        <Row className="mb-3">

                <Form.Group as={Col} md="4" controlId="validationFormik01">

                  <Form.Label>Debit Card Number</Form.Label>

                  <Form.Control

                    type="text"

                    name="cardnumber"

                    placeholder="XXXX-XXXX-XXXX-XXXX"

                  />

                </Form.Group>

                <Form.Group as={Col} md="4" controlId="validationFormik02">

                  <Form.Label>Expiry Date</Form.Label>

                  <Form.Control

                    type="text"

                    name="expiry"

                    placeholder="--/----"

                  />

                </Form.Group>

                <Form.Group as={Col} md="4" controlId="validationFormikUsername">

                  <Form.Label>CVV</Form.Label>

                    <Form.Control

                      type="text"

                      placeholder="XXX"

                      aria-describedby="inputGroupPrepend"

                      name="cvv "

                    />

                </Form.Group>

              </Row>

              </>

      );

    }

    else if(payment==='3'){

       return (

        <>

        <h4 className='text-center mt-4 mb-4'><u>Enter Credit Card Details</u></h4>

        <Row className="mb-3">

                <Form.Group as={Col} md="4" controlId="validationFormik01">

                  <Form.Label>Credit Card Number</Form.Label>

                  <Form.Control

                    type="text"

                    name="cardnumber"

                    placeholder="XXXX-XXXX-XXXX-XXXX"

                  />

                </Form.Group>

                <Form.Group as={Col} md="4" controlId="validationFormik02">

                  <Form.Label>Expiry Date</Form.Label>

                  <Form.Control

                    type="text"

                    name="expiry"

                    placeholder="--/----"

                  />

                </Form.Group>

                <Form.Group as={Col} md="4" controlId="validationFormikUsername">

                  <Form.Label>CVV</Form.Label>

                    <Form.Control

                      type="text"

                      placeholder="XXX"

                      aria-describedby="inputGroupPrepend"

                      name="cvv "

                    />

                </Form.Group>

              </Row>

              </>

      );

    }

  }

  const updateMilestone = (event, property) => {

    const target = event.target

    console.log(target)

    event.preventDefault()

    setMilestone((prevState) => ({

      ...prevState,

      [property]: event.target.value,

    }))

  }

  return (

    <div className='' >

               <img src="https://www.hopkinsmedicine.org/-/media/images/health/3\_-wellness/living-with-a-chronic-disease/help-for-managing-multiple-medications-hero.ashx?h=500&iar=0&mh=500&mw=1300&w=1297&hash=122DB3C1DCAD683F2F7CAE139A67613C" style={{position:'absolute', width:"100%",opacity:'.7'}} height={600    } />

        <Header/>

      <Container className=' bg-light py-3 border w-75 rounded my-5 ' style={{position:'relative',top:'31px',padding:'50px',overflowY:"scroll",height:'32pc'}}>

        <h3 className='text-center '> Check Out</h3>

        <hr></hr>

      <Form>

      <Row className="mb-3">

      <h4 className='text-center '> <u>User Details</u></h4>

                 <Form.Group as={Col} md="4" controlId="validationFormik01">

                   <Form.Label>Full Name</Form.Label>

                   <Form.Control

                    type="text"

                    onChange={(e) => updateMilestone(e, 'name')}

                    value={milestone.name}

                    name="firstName"

                    required

                  />

                  <Form.Control.Feedback>Looks good!</Form.Control.Feedback>

                </Form.Group>

                <Form.Group as={Col} md="4" controlId="validationFormik02">

                  <Form.Label>Mobile number</Form.Label>

                  {/\* <PhoneInput

 value={milestone.number}

                    name="number"

                    onChange={(e) => updateMilestone(e, 'number')}

                    className="text-dark"

                  /> \*/}

                   <Form.Control

                    type="text"

                    onChange={(e) => updateMilestone(e, 'number')}

                    value={milestone.number}

                    name="firstName"

                    required

                  />

                  <Form.Control.Feedback>Looks good!</Form.Control.Feedback>

                </Form.Group>

                <Form.Group as={Col} md="4" controlId="validationFormikUsername">

                  <Form.Label>Email</Form.Label>

                  <InputGroup hasValidation>

                    <InputGroup.Text id="inputGroupPrepend">@</InputGroup.Text>

                    <Form.Control

                      type="text"

                      value={milestone.email}

                      onChange={(e) => updateMilestone(e, 'email')}

                      placeholder="Email"

                      aria-describedby="inputGroupPrepend"

                      name="email"

                      required

                    />

                  </InputGroup>

                </Form.Group>

              </Row>

                <Form.Group controlId="validationFormik03">

                 <Form.Label>Address</Form.Label>

                 <Form.Control

                  type="text"

                  onChange={(e) => updateMilestone(e, 'address')}

                  value={milestone.address}

                  placeholder="Address"

                  name="address"

                  required

                />

                </Form.Group>

                <Form.Group className='w-25 text-center' style={{position:'relative',left:'38%'}} required>

                <Form.Label>Payment Type</Form.Label>

                <Form.Select onChange={(e) => {SetPayment(e.target.value) }} aria-label="Default select example">

                  <option value="0" > Open this select menu</option>

                  <option value="1">Cash on Delivery</option>

                  <option value="2">Debit Card</option>

                  <option value="3">Credit Card</option>

                </Form.Select>

              </Form.Group>

            {

              setpayment()

            }

            <Container className="py-4 mt-2 border border-4 border-dark">

            <h4 className={` text-center`}>

               <u> {isEmpty? 'Your Cart is Empty' : 'Review Cart Details'}</u>

            </h4>

            <Row className="justify-content-center">

                <Table responsive="sm" striped bordered hover variant={'light'} className="mb-5">

                <thead>

                        <tr>

                            <th>Id</th>

                            <th>Name</th>

                            <th>Price</th>

                            <th>Seller</th>

                            <th>Quantity</th>

                            <th>Description</th>

                            <th>Actions</th>

                        </tr>

                    </thead>

                    <tbody>

                        {items.map((item, index)=>{

                            return(

                              <tr key={index}>

                              <td>{item.id}</td>

                              <td>

                                  <h6 style={{ whiteSpace: 'nowrap', width: '14rem', overflow: 'hidden', textOverFlow: 'ellipsis'}}>

                                      {item.name}

                                  </h6>

                              </td>

                              <td>Rs. {item.price}</td>

                              <td>Seller ({item.seller})</td>

                              <td>Quantity ({item.quantity})</td>

                              <td> {item.description}</td>

                              <td>

                                  <Button onClick={()=> updateItemQuantity(item.id, item.quantity - 1)} className="ms-2">-</Button>

                                  <Button onClick={()=> updateItemQuantity(item.id, item.quantity + 1)} className="ms-2">+</Button>

                                  <Button variant="danger" onClick={()=> removeItem(item.id)} className="ms-2">Remove Item</Button>

                              </td>

                          </tr>

                            )

                        })}

                    </tbody>

                </Table>

                {!isEmpty &&

                    <Row

                        style={{ position: 'fixed', bottom: 0}}

                        className={`bg-light text-balck justify-content-center w-100`}

                    >

                        <Col className="py-2">

                            <h4>Total Price: Rs. {cartTotal}</h4>

                        </Col>

                        <Col className="p-0" md={4}>

                            <Button variant="danger"

                                className="m-2"

                                onClick={()=> emptyCart()}

                            >

                                <BsCartX size="1.7rem" />

                                Clear Cart

                            </Button>

                            <Button onClick={handleSubmit}

                                className="btn btn-success m-2"

                            >

                                <BsCartCheck size="1.7rem" />

                               Checkout

                            </Button>

                        </Col>

                    </Row>}

            </Row>

        </Container>

      </Form>

      </Container>

    </div>

  )

}

export default Checkout;

pages/Cart.js

import React from 'react';

import { Button, Container, Col, Row, Table} from 'react-bootstrap';

import { useCart } from 'react-use-cart';

import { BsCartCheck, BsCartX} from 'react-icons/bs';

// import {Link} f

import { Link } from "@reach/router";

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

import { useState } from 'react';

import "./backgroundImage.css";

const Cart = () => {

    const [session, SetSession] = useState(sessionStorage.getItem("name"));

    const checkSignin=()=>{

        // alert("Sign in To Continue");

        if(session===null){

            return(

                <Link to='/sign-in'

                                className="btn btn-success m-2"

                            >

                                <BsCartCheck size="1.7rem" />

                               Checkout

                            </Link>

            );

        }

        else{

            return(

                <Link to='/checkout'

                                className="btn btn-success m-2"

                            >

                                <BsCartCheck size="1.7rem" />

                               Checkout

                            </Link>

            );

        }

    }

    const {

        isEmpty,

        items,

        cartTotal,

        updateItemQuantity,

        removeItem,

        emptyCart,

    } = useCart();

    return (

        <div className='containerUnique3'>

        <Header/>

        {/\* <img src="https://www.hopkinsmedicine.org/-/media/images/health/3\_-wellness/living-with-a-chronic-disease/help-for-managing-multiple-medications-hero.ashx?h=500&iar=0&mh=500&mw=1300&w=1297&hash=122DB3C1DCAD683F2F7CAE139A67613C" style={{position:'absolute', width:"100%",opacity:'.7' ,zIndex:'-1'}} height={657} /> \*/}

        <Container className=" py-4 mt-5">

            <h1 className={`text-light-primary my-5 text-center`}>

                {isEmpty? 'Your Cart is Empty' : 'The Cart'}

            </h1>

            <Row className="justify-content-center">

                <Table responsive="sm" striped bordered hover variant={'light'} className="mb-5">

                    <thead>

                        <tr>

                            <th>Id</th>

                            <th>Name</th>

                            <th>Price</th>

                            <th>Seller</th>

                            <th>Quantity</th>

                            <th>Description</th>

                            <th>Actions</th>

                        </tr>

                    </thead>

                    <tbody>

                        {items.map((item, index)=>{

                            return(

                                <tr key={index}>

                                    {/\* <td>

                                        <div style={{ background: 'white', height: '8rem', overflow: 'hidden', display: 'flex',

                                        justifyContent: 'center', alignItems: 'center' }}>

                                            <div style={{ padding: '.5rem'}}>

                                                <img src={item.image} style={{ width: '4rem'}} alt={item.title} />

                                            </div>

                                        </div>

                                    </td> \*/}

                                    <td>{item.id}</td>

                                    <td>

                                        <h6 style={{ whiteSpace: 'nowrap', width: '14rem', overflow: 'hidden', textOverFlow: 'ellipsis'}}>

                                            {item.name}

                                        </h6>

                                    </td>

                                    <td>Rs. {item.price}</td>

                                    <td>Seller ({item.seller})</td>

                                    <td>Quantity ({item.quantity})</td>

                                    <td> {item.description}</td>

                                    <td>

                                        <Button onClick={()=> updateItemQuantity(item.id, item.quantity - 1)} className="ms-2">-</Button>

                                        <Button onClick={()=> updateItemQuantity(item.id, item.quantity + 1)} className="ms-2">+</Button>

                                        <Button variant="danger" onClick={()=> removeItem(item.id)} className="ms-2">Remove Item</Button>

                                    </td>

                                </tr>

                            )

                        })}

                    </tbody>

                </Table>

                {!isEmpty &&

                    <Row

                        style={{ position: 'fixed', bottom: 0}}

                        className={`bg-light text-balck justify-content-center w-100`}

                    >

                        <Col className="py-2">

                            <h4>Total Price: Rs. {cartTotal}</h4>

                        </Col>

                        <Col className="p-0" md={4}>

                            <Button variant="danger"

                                className="m-2"

                                onClick={()=> emptyCart()}

                            >

                                <BsCartX size="1.7rem" />

                                Clear Cart

                            </Button>

                            {

                                checkSignin()

                            }

                        </Col>

                    </Row>}

            </Row>

        </Container>

        </div>

    );

};

export default Cart;

pages/Admin/AdminLogin.js

import React, { useState } from 'react'

import "../Admin/login.css";

import { useNavigate } from '@reach/router';

// import "../Admin/login";

function AdminLogin() {

        const [username,Setusename]=useState("");

    const [password,Setpassword]=useState("");

    const navigate=useNavigate();

    const handleSubmit=()=>{

        if(username==="admin" && password==="admin"){

            sessionStorage.setItem("admin", true);

            navigate("/admin\_home");

        }

    }

    // JSX code for login form

    const renderForm = (

      <div className="form">

        <form onSubmit={handleSubmit}>

          <div className="input-container">

            <label>Username </label>

            <input type="text"  onChange={(e)=>{Setusename(e.target.value)}} placeholder="Username" required/>

          </div>

          <div className="input-container">

            <label>Password </label>

          <input type="password" onChange={(e)=>{Setpassword(e.target.value)}} placeholder="Password"  required/>

          </div>

          <div className="button-container">

            <input  className='btn btn-success my-2' type="button" onClick={handleSubmit} value="Login"/>

          </div>

        </form>

      </div>

    );

    return (

      <div className="app" style={{background:'bisque'}}>

        <div className="login-form">

          <div className="title text-center">Admin Login</div>

          {renderForm}

        </div>

      </div>

    );

  }

  export default AdminLogin;

pages/Admin/AdminHome.js

import { Link, useNavigate } from '@reach/router'

import axios from 'axios';

import React, { useEffect } from 'react'

import { useState } from 'react';

import "./login.css";

import Switch from "react-switch";

function AdminHome() {

    const navigate = useNavigate();

    var data = []

    const [checkedValue, SetcheckedValue] = useState(false);

    const [productData, setProductData] = useState([]);

    // const deleteProduct = (pid) => {

    //     axios.get("http://localhost:8081/deleteProduct/" + pid)

    //         .then(response => alert("Medicine Data Deleted"));

    //     window.location.reload();

    // }

    // const changeStatus = (pid) => {

    //     axios.get("http://localhost:8081/updateStatus/" + pid)

    //         .then(response => console.log("A"));

    //         window.location.reload();

    // }

    // const getData=()=>{

    //     axios.get("http://localhost:8081/getAllProducts")

    //     .then(response => setProductData(response.data));

    // }

    const deleteProduct = (pid) => {

        axios.get("http://18.204.215.152:8080/deleteProduct/" + pid)

            .then(response => alert("Medicine Data Deleted"));

        window.location.reload();

    }

    const changeStatus = (pid) => {

        axios.get("http://18.204.215.152:8080/updateStatus/" + pid)

            .then(response => console.log("A"));

            window.location.reload();

    }

    const getData=()=>{

        axios.get("http://18.204.215.152:8080/getAllProducts")

        .then(response => setProductData(response.data));

    }

    useEffect(() => {

        getData();

        console.log(productData.length);

        if (sessionStorage.getItem("admin") === null) {

            navigate("/admin\_login");

        }

    }, [])

    return (

<>

<nav className="navbar navbar-expand-lg navbar-light text-center" style={{background:'#ffbc6c'}}>

  <Link to="/admin\_home" className="navbar-brand mx-5 " href="#" style={{position:'relative',left:'7.5pc'}}><b>Medicare Admin</b></Link>

  <button className="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

    <span className="navbar-toggler-icon"></span>

  </button>

  <div className="collapse navbar-collapse" id="navbarNav">

    <ul className="navbar-nav">

      <li className="nav-item "  style={{position:'relative',right:'-51.8pc' ,color:'black'}}>

      <button className='btn btn-danger'  onClick={() => {

                    sessionStorage.removeItem("admin");

                    navigate("/admin\_login");

                }}>Logout</button>

      </li>

    </ul>

  </div>

</nav>

        <div className='d-flex bg-light justify-content-center align-items-center' style={{ height: '37pc' }}>

            <div className='w-75 bg-white rounded' style={{ padding: '8px',border:'2px solid', overflowY: "scroll", height: "573px",position:'relative',top:'-0px' }}>

                <h3 className='text-center'>List of Medicines</h3>

                <Link to='/createProduct' className='btn btn-success ' style={{ textAlign: 'right', position: 'relative', right: "-86%" }}>Add +</Link >

                <table className='table table-striped ' style={{}}>

                    <thead>

                        <tr>

                            <th>Sno</th>

                            <th>Product Name</th>

                            <th>Price</th>

                            <th>Seller</th>

                            <th>Description</th>

                            <th>Status</th>

                            <th>Actions</th>

                        </tr>

                    </thead>

                    <tbody>

                        {

                            productData.map((item, index) => (

                                <tr key={index}>

                                    <td>{item.id}</td>

                                    <td>{item.name}</td>

                                    <td>{item.price}</td>

                                    <td>{item.seller}</td>

                                    <td>{item.description}</td>

                                    <td>                                       <Switch onChange={()=>changeStatus(item.id)} checked={item.status} />

                                    </td>

                                    <td>

                                        <Link className='btn btn-success mx-3' to={`/updateProduct/${item.id}`}>Update</Link>

                                        <button

                                            onClick={() => { deleteProduct(item.id) }}

                                            className='btn btn-danger'>Delete</button>

                                    </td>

                                </tr>

                            ))

                        }

                    </tbody>

                </table>

            </div>

        </div>

        </>

    )

}

export default AdminHome

pages/Admin/CreateProduct.js

import { useNavigate } from '@reach/router';

import axios from 'axios';

import React, { useState } from 'react'

import { useEffect } from 'react';

import { Link } from '@reach/router';

function CreateProduct() {

    const navigate=useNavigate();

    const [data, setData] = useState({

        name: '',

        price:'',

        seller: '',

        description:'',

        status: ''

      })

    const handleSubmit=(e)=>{

        e.preventDefault();

        if(data.name==="" && data.name==="" && data.name==="" && data.name==="" && data.name==="" ){

            alert("Fill all the details");

        }

        else{

            // console.log(typeof(data.sta))

            const data1={

                "name":data.name,

                "price":data.price,

                "seller":data.seller,

                "description":data.description,

                "status":data.status

            }

            axios.put('http://18.204.215.152:8080/addProduct', data1)

            .then(response => alert("Medicine Added"));

            navigate("/admin\_home");

                }

    }

    useEffect(()=>{

        if(sessionStorage.getItem("admin")===null){

            navigate("/admin\_login");

        }

},[]);

      const updateData = (event, property) => {

        const target = event.target

        console.log(target)

        event.preventDefault()

        setData((prevState) => ({

          ...prevState,

          [property]: event.target.value,

        }))

      }

    return (

        <>

<nav className="navbar navbar-expand-lg navbar-light text-center" style={{background:'#ffbc6c'}}>

<Link to="/admin\_home" className="navbar-brand mx-5 " href="#" style={{position:'relative',left:'7.5pc'}}><b>Medicare Admin</b></Link>

  <button className="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

    <span className="navbar-toggler-icon"></span>

  </button>

  <div className="collapse navbar-collapse" id="navbarNav">

    <ul className="navbar-nav">

      <li className="nav-item"  style={{position:'relative',right:'-51.8pc'}}>

      <button className='btn btn-danger'  onClick={() => {

                    sessionStorage.removeItem("admin");

                    navigate("/admin\_login");

                }}>Logout</button>

      </li>

    </ul>

  </div>

</nav>

        <div className='d-flex  justify-content-center align-items-center' style={{ height: '37pc' }}>

            <div className='w-50 bg-white rounded' style={{ padding: '23px',border:'2px solid', overflowY: "scroll", height: "59%",position:'relative',top:'-30px' }}>

                <form>

                    <h2>Add Medicine</h2>

                    <div className='row'>

                    <div className='col mb-2'>

                        <label >Name</label>

                        <input type='text' placeholder='Enter Medicine Name'

                         onChange={(e) => updateData(e, 'name')}

                        className='form-control'/>

                    </div>

                    <div className=' col mb-2'>

                        <label >Seller</label>

                        <input type='text' placeholder='Enter Medicine Seller'

                         onChange={(e) => updateData(e, 'seller')}

                        className='form-control'/>

                    </div>

                    </div>

                    <div className='row'>

                    <div className='col mb-2'>

                        <label >Price</label>

                        <input type='text' placeholder='Enter Medicine Price'

                         onChange={(e) => updateData(e, 'price')}

                        className='form-control'/>

                    </div>

                        <div className='col mb-2'>

                            <label >Description</label>

                            <input type='text' placeholder='Enter Medicine Description'

                             onChange={(e) => updateData(e, 'description')}

                            className='form-control'/>

                        </div>

                        </div>

                    <div className='mb-2'>

                        <label >Status</label>

                        <select onChange={(e)=>{ updateData(e, 'status');console.log(data);}} className='form-control'>

                            <option defaultValue={0}>Select</option>

                            <option value={true}>Available</option>

                            <option value={false}>Not Available</option>

                        </select>

                        {/\* <input type='text' placeholder='Enter Medicine Status'

                         onChange={(e) => updateData(e, 'status')}

                        className='form-control'/> \*/}

                    </div>

                    <button onClick={handleSubmit} className='btn btn-success'>Submit</button>

                </form>

            </div>

        </div>

        </>

    )

}

export default CreateProduct

pages/Admin/UpdateProduct.js

import { useNavigate, useParams } from '@reach/router';

import axios from 'axios';

import React, { useState } from 'react'

import { useEffect } from 'react';

import { Link } from '@reach/router';

function UpdateProduct() {

    const [productData, setProductData] = useState({});

    const [data2,SetData]=useState([]);

    const navigate=useNavigate();

    const {id}=useParams();

    var dd={

        id:"",

        name: '',

        price:'',

        seller: '',

        description:'',

        status: ''

    }

    useEffect(()=>{

        if(sessionStorage.getItem("admin")===null){

            navigate("/admin\_login");

        }

        axios.get("http://localhost:8081/getProduct/"+id)

        .then(resp=>setData({...data,id:resp.data.id,name:resp.data.name,price:resp.data.price,seller:resp.data.seller,description:resp.data.description,status:resp.data.status}));

        // console.log(data);

    },[])

    const [data, setData] = useState({

        id:"",

        name: "",

        price:"",

        seller: "",

        description:"",

        status: ""

      })

      const updateData = (event, property) => {

        const target = event.target

        console.log(target)

        event.preventDefault()

        setData((prevState) => ({

          ...prevState,

          [property]: event.target.value,

        }))

      }

      const handleSubmit=(event)=>{

        event.preventDefault();

        console.log(data);

        axios.put("http://18.204.215.152:8080/addProduct",data)

        .then(resp=>alert("Medicine Updated"))

        navigate('/admin\_home')

      }

    return (

        <>

<nav className="navbar navbar-expand-lg navbar-light text-center" style={{background:'#ffbc6c'}}>

<Link to="/admin\_home" className="navbar-brand mx-5 " href="#" style={{position:'relative',left:'7.5pc'}}><b>Medicare Admin</b></Link>

<button className="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

    <span className="navbar-toggler-icon"></span>

  </button>

  <div className="collapse navbar-collapse" id="navbarNav">

    <ul className="navbar-nav">

      <li className="nav-item"  style={{position:'relative',right:'-51.8pc'}}>

      <button className='btn btn-danger'  onClick={() => {

                    sessionStorage.removeItem("admin");

                    navigate("/admin\_login");

                }}>Logout</button>

      </li>

    </ul>

  </div>

</nav>

        <div className='d-flex justify-content-center align-items-center' style={{ height: '37pc' }}>

        <div className='w-50 bg-white rounded' style={{ padding: '23px',border:'2px solid', overflowY: "scroll", height: "58%",position:'relative',top:'-30px' }}>

   <form>

                    <h2>Update Medicine</h2>

                    {/\* <div className='mb-2'>

                        <label >Name</label>

                        <input type='text' placeholder='Enter Medicine Name'

                         onChange={(e) => updateData(e, 'name')}

                         value={data.name}

                        className='form-control'/>

                    </div>

                    <div className='mb-2'>

                        <label >Seller</label>

                        <input type='text' placeholder='Enter Medicine Seller'

                         onChange={(e) => updateData(e, 'seller')}

                         value={data.seller}

                        className='form-control'/>

                    </div>

                    <div className='mb-2'>

                        <label >Price</label>

                        <input type='text' placeholder='Enter Medicine Price'

                         value={data.price}

                         onChange={(e) => updateData(e, 'price')}

                        className='form-control'/>

                    </div> \*/}

                    <div className='row'>

                    <div className='col mb-2'>

                        <label >Name</label>

                        <input type='text' placeholder='Enter Medicine Name'

                         onChange={(e) => updateData(e, 'name')}

                         value={data.name}

                        className='form-control'/>

                    </div>

                    <div className=' col mb-2'>

                        <label >Seller</label>

                        <input type='text' placeholder='Enter Medicine Seller'

                         onChange={(e) => updateData(e, 'seller')}

                         value={data.seller}

                        className='form-control'/>

                    </div>

                    </div>

                    <div className='row'>

                    <div className='col mb-2'>

                        <label >Price</label>

                        <input type='text' placeholder='Enter Medicine Price'

                         onChange={(e) => updateData(e, 'price')}

                         value={data.price}

                        className='form-control'/>

                    </div>

                        <div className='col mb-2'>

                            <label >Description</label>

                            <input type='text' placeholder='Enter Medicine Description'

                             onChange={(e) => updateData(e, 'description')}

                             value={data.description}

                            className='form-control'/>

                        </div>

                        </div>

                    <div className='mb-2'>

                        <label >Status</label>

                        <input type='text' placeholder='Enter Medicine Status'

                         value={data.status}

                         onChange={(e) => updateData(e, 'status')}

                        className='form-control'/>

                    </div>

                    {/\* <div className='mb-2'>

                        <label >Description</label>

                        <input type='text' placeholder='Enter Medicine Description'

                         value={data.description}

                         onChange={(e) => updateData(e, 'description')}

                        className='form-control'/>

                    </div> \*/}

                    <button onClick={handleSubmit} className='btn btn-success'>Submit</button>

                </form>

            </div>

        </div>

        </>

    )

}

export default UpdateProduct;

**Medicare-BackEnd**

**Main java File**

package com.example.Medicare.Backend;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class MedicareBackendApplication {

public static void main(String[] args) {

SpringApplication.run(MedicareBackendApplication.class, args);

}

}

**Default Auth Configuration file**

package com.example.Medicare.Backend;

import java.util.Arrays;

import org.apache.catalina.filters.CorsFilter;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.cors.CorsConfiguration;

import org.springframework.web.cors.reactive.UrlBasedCorsConfigurationSource;

import org.springframework.web.servlet.config.annotation.CorsRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration

public class DefaultAuthConfiguration implements WebMvcConfigurer {

@Override

public void addCorsMappings(CorsRegistry myCorsRegistry){

myCorsRegistry.addMapping("/\*\*")

.allowedOrigins("http://localhost:3000") //frontend's link

.allowedHeaders("\*")

.allowedMethods("GET","PUT")

.allowCredentials(true)

.maxAge(4800);

}

}

**Controllers**

MedicalController.java

package com.example.Medicare.Backend.controller;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.example.Medicare.Backend.entity.Medicine;

import com.example.Medicare.Backend.repository.MedicineRepository;

@RestController()

@CrossOrigin

@RequestMapping("")

public class MedicineController {

@Autowired

private MedicineRepository medicineRepo;

@GetMapping("/getAllProducts")

public Iterable<Medicine> getAll() {

Iterable<Medicine> data=medicineRepo.findAll();

System.out.println(data);

return data;

}

@GetMapping("/getProducts")

public Iterable<Medicine> getProducts() {

Iterable<Medicine> data=medicineRepo.findByStatus(true);

System.out.println(data);

return data;

}

@GetMapping("/getProduct/{id}")

public Optional<Medicine> getproduct(@PathVariable("id") int id) {

Optional<Medicine> data=medicineRepo.findById(id);

System.out.println(id);

return data;

}

@PutMapping("/addProduct")

public String addData(@RequestBody Medicine medicine ) {

medicineRepo.save(medicine);

return "Data Saved";

}

@GetMapping("/deleteProduct/{pid}")

public String deleteData(@PathVariable("pid") String pid) {

System.out.println(pid);

medicineRepo.deleteById(Integer.parseInt(pid));

return "Data Deleted";

}

@GetMapping("/updateStatus/{id}")

public String UpdateStatus(@PathVariable("id") int id) {

Medicine med=medicineRepo.findById(id).get();

// med.setStatus(status);

if(med.getStatus()==true) {

med.setStatus(false);

}

else if(med.getStatus()==false) {

med.setStatus(true);

}

medicineRepo.save(med);

return "status changed";

//

}

}

**OrdersController.java**

package com.example.Medicare.Backend.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.example.Medicare.Backend.entity.Orders;

import com.example.Medicare.Backend.repository.OrdersRepository;

@RestController()

@CrossOrigin

@RequestMapping(value="")

public class OrderController {

@Autowired

public OrdersRepository orderRepo;

@GetMapping("/getOrders")

public Iterable<Orders> getAll() {

Iterable<Orders> data=orderRepo.findAll();

return data;

}

@PostMapping("/addOrder")

public String addOrder(@RequestBody Orders order) {

orderRepo.save(order);

return "Order Updated";

}

}

**UsersController.java**

package com.example.Medicare.Backend.controller;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

import java.util.Optional;

import java.util.stream.StreamSupport;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.PatchMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

import com.example.Medicare.Backend.entity.Users;

import com.example.Medicare.Backend.repository.UserRepository;

@RestController()

@CrossOrigin

@RequestMapping(value="")

public class UserController {

@Autowired

public UserRepository userRepo;

@GetMapping("/getUsers")

public Iterable<Users> getAll() {

Iterable<Users> data=userRepo.findAll();

return data;

}

@PostMapping("/addUser")

public String sayHello(@RequestBody Users user ) {

userRepo.save(user);

return "Data Saved";

}

}

**Entitys**

**Medicine.java**

**package** com.example.Medicare.Backend.entity;

**import** jakarta.persistence.\*;

@Entity

**public** **class** Medicine {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** **int** id;

**private** String name;

**private** String price;

**private** String seller;

**private** String description;

**private** **boolean** status;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getPrice() {

**return** price;

}

**public** **void** setPrice(String price) {

**this**.price = price;

}

**public** String getSeller() {

**return** seller;

}

**public** **void** setSeller(String seller) {

**this**.seller = seller;

}

**public** String getDescription() {

**return** description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** **boolean** getStatus() {

**return** status;

}

**public** **void** setStatus(**boolean** status) {

**this**.status = status;

}

@Override

**public** String toString() {

**return** "Medicine [id=" + id + ", name=" + name + ", price=" + price + ", seller=" + seller + ", description="

+ description + ", status=" + status + "]";

}

**public** Medicine() {

}

}

**Orders.java**

package com.example.Medicare.Backend.entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.\*;

import jakarta.persistence.Id;

@Entity

public class Orders {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String user\_name;

private String user\_number;

private String user\_email;

private String user\_address;

private String items;

private String date;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getUser\_name() {

return user\_name;

}

public void setUser\_name(String user\_name) {

this.user\_name = user\_name;

}

public String getUser\_number() {

return user\_number;

}

public void setUser\_number(String user\_number) {

this.user\_number = user\_number;

}

public String getUser\_email() {

return user\_email;

}

public void setUser\_email(String user\_email) {

this.user\_email = user\_email;

}

public String getUser\_address() {

return user\_address;

}

public void setUser\_address(String user\_address) {

this.user\_address = user\_address;

}

public String getItems() {

return items;

}

public void setItems(String items) {

this.items = items;

}

public String getDate() {

return date;

}

public void setDate(String date) {

this.date = date;

}

public Orders() {

}

}

**Users.java**

**package** com.example.Medicare.Backend.entity;

**import** jakarta.persistence.\*;

@Entity

**public** **class** Users {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** **int** id;

**private** String name;

**private** String number;

**private** String email;

**private** String address;

**private** String password;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getNumber() {

**return** number;

}

**public** **void** setNumber(String number) {

**this**.number = number;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** Users() {

}

@Override

**public** String toString() {

**return** "Users [id=" + id + ", name=" + name + ", number=" + number + ", email=" + email + ", address=" + address

+ ", password=" + password + "]";

}

}

**Repository**

**MedicineRepository.java**

package com.example.Medicare.Backend.repository;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;

import com.example.Medicare.Backend.entity.Medicine;

public interface MedicineRepository extends CrudRepository<Medicine, Integer> {

Iterable<Medicine> findByStatus(boolean i);

Iterable<Medicine> findByName(String name);

}

**OrdersRepository.java**

package com.example.Medicare.Backend.repository;

import org.springframework.data.repository.CrudRepository;

import com.example.Medicare.Backend.entity.Orders;

public interface OrdersRepository extends CrudRepository<Orders,Integer>{

}

**UsersRepository.java**

package com.example.Medicare.Backend.repository;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;

import com.example.Medicare.Backend.entity.Users;

public interface UserRepository extends CrudRepository<Users, Integer> {

Iterable<Users> findByName(String name);

Optional<Users> findByNumberAndPassword(String number, String password);

}

**Application.properties**

spring.datasource.url=jdbc:mysql://localhost:3306/medicare

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.jpa.properties.hibernate.format\_sql=true

spring.jpa.hibernate.ddl-auto=update

server.port=8080