Backend Code:

Models:

const mongoose = require('mongoose');  
  
const Schema = mongoose.Schema;  
  
const contactFormSchema = new Schema({  
 firstName: String,  
 lastName: String,  
 email: String,  
 phone: String,  
 message: String,  
 propertyId: String,  
})  
  
const contactform = mongoose.model('contactform', contactFormSchema, 'contactforms')  
  
module.exports = contactform;

const mongoose = require('mongoose');  
  
const Schema = mongoose.Schema;  
  
const propertySchema = new Schema({  
 name: String,  
 price: String,  
 propertyArea: String,  
 street: String,  
 bedrooms: String,  
 date: String,  
 occupied: String,  
 areaStatus: String,  
})  
  
const property = mongoose.model('property', propertySchema, 'properties')  
  
module.exports = property;

const express = require('express');  
const router = express.Router();  
var nodemailer = require('nodemailer');  
const jwt = require('jsonwebtoken');  
const Area = require('../models/areaModel');  
const Contactform = require('../models/contactFormModel');  
const Property = require('../models/propertyModel');  
const User = require('../models/userModel');  
  
  
router.post('/create', (req, res) => {  
 console.log(req.body)  
 Contactform.insertMany(req.body, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.get('/viewAll', (req, res) => {  
 Contactform.find({}, (err, data) => {  
 if (err) {  
 console.log(err)   
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.post('/delete/:id', (req, res) => {  
 Contactform.findOneAndDelete({ \_id: req.params.id }, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.post('/sendEmail', (req, res) => {  
 console.log(req.body)  
 var transporter = nodemailer.createTransport({  
 service: 'gmail',  
 auth: {  
 user: 'webdevumkc@gmail.com',  
 pass: 'webdevumkc2021'  
 }  
 });  
   
 var emailBody = `<body style="background-color:grey">  
 <table align="center" border="0" cellpadding="0" cellspacing="0"  
 width="550" bgcolor="white" style="border:2px solid black">  
 <tbody>  
 <tr>  
 <td align="center">  
 <table align="center" border="0" cellpadding="0"   
 cellspacing="0" class="col-550" width="550">  
 <tbody>  
 <tr>  
 <td align="center" style="background-color: #4cb96b;  
 height: 50px;">  
   
 <a href="#" style="text-decoration: none;">  
 <p style="color:white;  
 font-weight:bold;">  
 Thank you for contacting us  
 </p>  
 </a>  
 </td>  
 </tr>  
 </tbody>  
 </table>  
 </td>  
 </tr>  
 <tr style="height: 300px;">  
 <td align="center" style="border: none;  
 border-bottom: 2px solid #4cb96b;   
 padding-right: 20px;padding-left:20px">  
   
 <p style="font-weight: bolder;font-size: 42px;  
 letter-spacing: 0.025em;  
 color:black;">  
 <p>Hi, we received a request that you had enquired for the below property</p>  
 <p>http://localhost:3000/property/${req.body.propertyId} </p>  
 <p>Inorder to reserve the above property, please visit the below link</p>  
 <p>http://localhost:3000/customer\_enroll/${req.body.propertyId} </p>  
 <br>   
 </p>  
 </td>  
 </tr>  
 </tbody>  
 </table>  
</body>`  
  
 var mailOptions = {  
 from: 'webdevumkc@gmail.com',  
 to: req.body.email,  
 subject: 'Response to enquiry about property',  
 text: 'Test',  
 html: emailBody  
 };  
   
 transporter.sendMail(mailOptions, function(error, info){  
 if (error) {  
 console.log(error);  
 } else {  
 console.log('Email sent: ' + info.response);  
 }  
 });  
})  
  
module.exports = router;

const mongoose = require('mongoose');  
  
const Schema = mongoose.Schema;  
  
const areaSchema = new Schema({  
 name: String,  
})  
  
const area = mongoose.model('area', areaSchema, 'areas')  
  
module.exports = area;

const mongoose = require('mongoose');  
  
const Schema = mongoose.Schema;  
  
const customerEnrollSchema = new Schema({  
 name: String,  
 email: String,  
 phone: String,  
 password: String,  
 propertyId: String,  
})  
  
const customerEnroll = mongoose.model('customerEnroll', customerEnrollSchema, 'customerEnrolls')  
  
module.exports = customerEnroll;

const mongoose = require('mongoose');  
  
const Schema = mongoose.Schema;  
  
const userSchema = new Schema({  
 username: String,  
 email: String,  
 password: String,  
})  
  
const user = mongoose.model('user', userSchema, 'users')  
  
module.exports = user;

Routes:

const express = require('express');  
const router = express.Router();  
const jwt = require('jsonwebtoken');  
const Area = require('../models/areaModel');  
const User = require('../models/userModel');  
  
function verifyToken(req, res, next) {  
 console.log(req.headers.authorization)  
 if (!req.headers.authorization) {  
 return res.status(401).send('Unauthorized request')  
 }  
 let token = req.headers.authorization.split(' ')[1]  
 if (token === 'null') {  
 return res.status(401).send('Unauthorized request')  
 }  
 let payload = jwt.verify(token, 'secretKey')  
 if(!payload) {  
 return res.status(401).send('Unauthorized request')  
 }  
 req.userId = payload.subject;  
 next()  
 }  
  
router.post('/create', verifyToken, (req, res) => {  
 Area.insertMany(req.body, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 // console.log(data)  
 res.send(data)  
 }  
 })  
})  
  
router.get('/viewAllData', verifyToken, (req, res) => {  
 Area.find({}, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 res.send(data)  
 }  
 })  
})  
  
router.get('/specificArea/:id', verifyToken, (req, res) => {  
 Area.findOne({ \_id: req.params.id }, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 res.send(data)  
 }  
 })  
})  
  
router.post('/updateAreaData/:id', verifyToken, (req, res) => {  
 Area.findOneAndUpdate({ \_id: req.params.id }, req.body, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 res.send(data)  
 }  
 })  
})  
  
router.post('/delete/:id', verifyToken, (req, res) => {  
 Area.findOneAndDelete({ \_id: req.params.id}, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 res.send(data)  
 }  
 })  
})  
  
module.exports = router;

const express = require('express');  
const router = express.Router();  
var nodemailer = require('nodemailer');  
const jwt = require('jsonwebtoken');  
const Area = require('../models/areaModel');  
const Contactform = require('../models/contactFormModel');  
const Property = require('../models/propertyModel');  
const User = require('../models/userModel');  
  
  
router.post('/create', (req, res) => {  
 console.log(req.body)  
 Contactform.insertMany(req.body, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.get('/viewAll', (req, res) => {  
 Contactform.find({}, (err, data) => {  
 if (err) {  
 console.log(err)   
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.post('/delete/:id', (req, res) => {  
 Contactform.findOneAndDelete({ \_id: req.params.id }, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.post('/sendEmail', (req, res) => {  
 console.log(req.body)  
 var transporter = nodemailer.createTransport({  
 service: 'gmail',  
 auth: {  
 user: 'webdevumkc@gmail.com',  
 pass: 'webdevumkc2021'  
 }  
 });  
   
 var emailBody = `<body style="background-color:grey">  
 <table align="center" border="0" cellpadding="0" cellspacing="0"  
 width="550" bgcolor="white" style="border:2px solid black">  
 <tbody>  
 <tr>  
 <td align="center">  
 <table align="center" border="0" cellpadding="0"   
 cellspacing="0" class="col-550" width="550">  
 <tbody>  
 <tr>  
 <td align="center" style="background-color: #4cb96b;  
 height: 50px;">  
   
 <a href="#" style="text-decoration: none;">  
 <p style="color:white;  
 font-weight:bold;">  
 Thank you for contacting us  
 </p>  
 </a>  
 </td>  
 </tr>  
 </tbody>  
 </table>  
 </td>  
 </tr>  
 <tr style="height: 300px;">  
 <td align="center" style="border: none;  
 border-bottom: 2px solid #4cb96b;   
 padding-right: 20px;padding-left:20px">  
   
 <p style="font-weight: bolder;font-size: 42px;  
 letter-spacing: 0.025em;  
 color:black;">  
 <p>Hi, we received a request that you had enquired for the below property</p>  
 <p>http://localhost:3000/property/${req.body.propertyId} </p>  
 <p>Inorder to reserve the above property, please visit the below link</p>  
 <p>http://localhost:3000/customer\_enroll/${req.body.propertyId} </p>  
 <br>   
 </p>  
 </td>  
 </tr>  
 </tbody>  
 </table>  
</body>`  
  
 var mailOptions = {  
 from: 'webdevumkc@gmail.com',  
 to: req.body.email,  
 subject: 'Response to enquiry about property',  
 text: 'Test',  
 html: emailBody  
 };  
   
 transporter.sendMail(mailOptions, function(error, info){  
 if (error) {  
 console.log(error);  
 } else {  
 console.log('Email sent: ' + info.response);  
 }  
 });  
})  
  
module.exports = router;

const express = require('express');  
const router = express.Router();  
const jwt = require('jsonwebtoken');  
const Area = require('../models/areaModel');  
const Contactform = require('../models/contactFormModel');  
const CustomerEnroll = require('../models/customerEnrollModel');  
const Property = require('../models/propertyModel');  
const User = require('../models/userModel');  
  
function verifyToken(req, res, next) {  
 console.log(req.headers.authorization)  
 if (!req.headers.authorization) {  
 return res.status(401).send('Unauthorized request')  
 }  
 let token = req.headers.authorization.split(' ')[1]  
 if (token === 'null') {  
 return res.status(401).send('Unauthorized request')  
 }  
 let payload = jwt.verify(token, 'secretKey')  
 if(!payload) {  
 return res.status(401).send('Unauthorized request')  
 }  
 req.userId = payload.subject;  
 next()  
 }  
  
router.post('/create', (req, res) => {  
 console.log(req.body)  
 CustomerEnroll.insertMany(req.body, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 Property.findOneAndUpdate({ \_id: req.body.propertyId }, { $set: { occupied: 'true'}}, (err, dataupdate) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (dataupdate) {  
 res.send(dataupdate)  
 }  
 }  
 })  
 }  
 }  
 })  
})  
  
router.get('/viewAll', (req, res) => {  
 CustomerEnroll.find({}, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.post('/delete/:id', (req, res) => {  
 CustomerEnroll.findOneAndDelete({ \_id: req.params.id }, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
router.post('/login', (req, res) => {  
 console.log(req.body, 'req body')  
 let userData = req.body;  
  
 CustomerEnroll.findOne({ email: req.body.email}, (err, user) => {  
 if(err) {  
 console.log(err);  
 } else {  
 if(!user) {  
 res.status(401).send('Invalid email')  
 } else {  
 if(user.password !== userData.password) {  
 res.status(401).send('Invalid Password')  
 } else {  
 let payload = { subject: user.\_id}  
 let token = jwt.sign(payload, 'secretKey')  
 res.status(200).send({token: token, userData: user})  
 }  
 }  
 }  
 })  
})  
  
router.post('/specificCustomer', verifyToken ,(req, res) => {  
 CustomerEnroll.findOne({ \_id: req.userId }, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 })  
})  
  
module.exports = router;

const express = require('express');  
const router = express.Router();  
const jwt = require('jsonwebtoken');  
const Area = require('../models/areaModel');  
const Property = require('../models/propertyModel');  
const User = require('../models/userModel');  
  
function verifyToken(req, res, next) {  
 console.log(req.headers.authorization)  
 if (!req.headers.authorization) {  
 return res.status(401).send('Unauthorized request')  
 }  
 let token = req.headers.authorization.split(' ')[1]  
 if (token === 'null') {  
 return res.status(401).send('Unauthorized request')  
 }  
 let payload = jwt.verify(token, 'secretKey')  
 if(!payload) {  
 return res.status(401).send('Unauthorized request')  
 }  
 req.userId = payload.subject;  
 next()  
 }  
  
router.post('/create', verifyToken, (req, res) => {  
 console.log('create property')  
 Property.insertMany(req.body, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 if (!data) {  
 console.log('no data')  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 }  
 })  
})  
  
router.get('/viewAllProperties', verifyToken, (req, res) => {  
 Property.find({}, (err, data) => {  
 if (err) {  
 console.log(err)  
 } else {  
 if (!data) {  
 console.log('no data found')  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 }  
 })  
})  
  
router.post('/getSpecificProperty/:id', (req, res) => {  
 console.log('in specific')  
 Property.findOne({ \_id: req.params.id }, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 if (!data) {  
 console.log('no data')  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 }  
 })  
})  
  
router.post('/updateProperty', (req, res) => {  
 Property.findOneAndUpdate({ \_id: req.body.\_id }, req.body, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 if (!data) {  
 console.log('no data')  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 }  
 })  
})  
  
router.post('/delete/:id', (req, res) => {  
 Property.findOneAndDelete({ \_id: req.params.id }, (err, data) => {  
 if (err) {  
 res.send(err)  
 } else {  
 if (!data) {  
 console.log('no data')  
 } else {  
 if (data) {  
 res.send(data)  
 }  
 }  
 }  
 })  
})  
  
router.post('/findFilteredProperties', (req, res) => {  
 console.log(req.body)  
 Property.find({}, (err, allData) => {  
 if (err) {  
 res.send(err)  
 } else {  
 if (!allData) {  
 console.log('no data')  
 } else {  
 if (allData) {  
 console.log(allData)  
 let dateFilteredData = [];  
 allData.forEach((v,i) => {  
 let reqDate = req.body.date;  
 console.log(req.body)  
 let reqDateDay = req.body.date.split('-')[2].split('T')[0];  
 let reqDateMonth = req.body.date.split('-')[1];  
 let reqDateYear = req.body.date.split('-')[0];  
 let vDateDay = v.date.split('-')[2].split('T')[0];  
 let vDateMonth = v.date.split('-')[1];  
 let vDateYear = v.date.split('-')[0];  
 // console.log(reqDateDay, 'req day')  
 // console.log(reqDateMonth, 'req month')  
 // console.log(reqDateYear, 'req year')  
 // console.log(vDateDay, 'v day')  
 // console.log(vDateMonth, 'v Month')  
 // console.log(vDateYear, 'v year')  
 if (reqDateYear >= vDateYear) {  
 // console.log('req year greater than or equal to')  
 if (reqDateMonth >= vDateMonth) {  
 // console.log('req month greater than or equal to')  
 if (reqDateDay >= vDateDay) {  
 // console.log('req day greater than or equal to')  
 dateFilteredData.push(v)  
 } else {  
 // console.log('req month only greater')  
 dateFilteredData.push(v)  
 }  
 }  
 } else {  
 console.log('false')  
 }  
  
 })  
 let reqPrice = req.body.price;  
 let priceFilteredData = [];  
 // console.log(dateFilteredData, 'datefiltereddata')  
 dateFilteredData.forEach((v,i) => {  
 console.log(req.body.price)  
 if (reqPrice >= Number(v.price)) {  
 priceFilteredData.push(v)  
 // console.log('reqprice less than or equal to v')  
 } else {  
 // priceFilteredData.push(v)  
 console.log(reqPrice, v.price,'reqPrice lesser than v')  
 }  
 })  
 console.log(priceFilteredData, 'priceFilteredData')  
 let reqBedrooms = req.body.bedrooms;  
 let bedroomsFilteredData = [];  
 priceFilteredData.forEach((v, i) => {  
 if (Number(reqBedrooms) === Number(v.bedrooms)) {  
 bedroomsFilteredData.push(v)  
 console.log(v, 'samebedrooms')  
 } else {  
 console.log('bedrooms didnt match')  
 }  
 })  
 let reqStreet = req.body.street;  
 let streetFilteredData = [];  
 bedroomsFilteredData.forEach((v, i) => {  
 if (reqStreet == v.street) {  
 streetFilteredData.push(v)  
 console.log(v, 'street matched')  
 } else {  
 console.log('street didnt match')  
 }  
 })  
 res.send(bedroomsFilteredData)  
 }  
 }  
 }  
 })  
})  
  
module.exports = router;

const express = require('express');  
const router = express.Router();  
const jwt = require('jsonwebtoken');  
const User = require('../models/userModel');  
  
router.post('/create', (req,res) => {  
 // console.log(req.body, 'body')  
 // return false  
 let userData = req.body;  
 let user = new User(userData)  
 user.save((error, data) => {  
 if (error) {  
 console.log(error)  
 } else {  
 console.log(data, 'data after response')  
 // res.send(data)  
 let payload = { subject: data.\_id.toString() }  
 console.log(payload, 'payload')  
 let token = jwt.sign(payload, 'secretKey');  
 res.status(200).send({token:token, userData: data})  
 }  
 })  
})  
  
router.post('/login', (req, res) => {  
 console.log(req.body, 'req body')  
 let userData = req.body;  
  
 User.findOne({ email: req.body.email}, (err, user) => {  
 if(err) {  
 console.log(err);  
 } else {  
 if(!user) {  
 res.status(401).send('Invalid email')  
 } else {  
 if(user.password !== userData.password) {  
 res.status(401).send('Invalid Password')  
 } else {  
 let payload = { subject: user.\_id}  
 let token = jwt.sign(payload, 'secretKey')  
 res.status(200).send({token: token, userData: user})  
 }  
 }  
 }  
 })  
})  
  
function verifyToken(req, res, next) {  
 console.log(req.headers.authorization)  
 if (!req.headers.authorization) {  
 return res.status(401).send('Unauthorized request')  
 }  
 let token = req.headers.authorization.split(' ')[1]  
 if (token === 'null') {  
 return res.status(401).send('Unauthorized request')  
 }  
 let payload = jwt.verify(token, 'secretKey')  
 if(!payload) {  
 return res.status(401).send('Unauthorized request')  
 }  
 req.userId = payload.subject;  
 next()  
 }  
  
  
module.exports = router;

Front End:

import {  
 BrowserRouter as Router,  
 Switch,  
 Route,  
 Link  
} from "react-router-dom";  
import PublicRoute from "./routes/publicRoute";  
import PrivateRoute from './routes/privateRoute';  
import Login from "./pages/login/Login";  
import Register from "./pages/register/Register";  
import Home from "./pages/home/Home";  
import PropertySearch from "./pages/propertySearch/PropertySearch";  
import Property from "./pages/property/Property";  
import AdminDashboard from "./pages/adminDashboard/AdminDashboard";  
import CustomerEnroll from "./components/customerEnroll/CustomerEnroll";  
import CustomerLogin from "./pages/customerLogin/CustomerLogin";  
import CustomerDashboard from "./pages/customerDashboard/CustomerDashboard";  
  
function App() {  
 return (  
 <>  
 <Router>  
 <Switch>  
 <PublicRoute path="/login" restricted={true} component={Login} exact />  
 <PublicRoute path="/register" component={Register} exact />  
 <PublicRoute path="/home" component={Home} exact />  
 <PublicRoute path="/propertysearch" component={PropertySearch} exact />  
 <PublicRoute path="/property/:id" component={Property} exact />  
 <PrivateRoute path="/admin\_dashboard" component={AdminDashboard} exact />  
 <PublicRoute path="/customer\_enroll/:id" component={CustomerEnroll} exact />  
 <PublicRoute path="/customerlogin" component={CustomerLogin} exact />  
 <PublicRoute path="/customerDashboard" component={CustomerDashboard} exact />  
 </Switch>  
 </Router>  
 </>  
 );  
}  
  
export default App;

import React from 'react';  
import ReactDOM from 'react-dom';  
import './index.css';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
  
ReactDOM.render(  
 <React.StrictMode>  
 <App />  
 </React.StrictMode>,  
 document.getElementById('root')  
);  
  
// If you want to start measuring performance in your app, pass a function  
// to log results (for example: reportWebVitals(console.log))  
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals  
reportWebVitals();

body {  
 margin: 0;  
 font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',  
 'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',  
 sans-serif;  
 -webkit-font-smoothing: antialiased;  
 -moz-osx-font-smoothing: grayscale;  
}  
  
code {  
 font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',  
 monospace;  
}

import React from 'react';  
import './footer.css';  
  
const Footer = () => {  
  
 return (  
 <div className="FooterContainer">  
 <div className="footerLeft">  
 <h2>Property Rental</h2>  
 <p>This application is all about properties and rental management</p>  
 </div>  
 <div className="footerCenter">  
 <h4>Nav Links</h4>  
 <ul>  
 <li>Services</li>  
 <li>About</li>  
 </ul>  
 </div>  
 <div className="footerRight">  
 <h4>Contact</h4>  
 <ul>  
 <li>Email: bhuvankumarchennoju@gmail.com</li>  
 <li>Address:</li>  
 <li>Kansas city, United States of America</li>  
 </ul>  
 </div>  
 </div>  
 )  
}  
  
export default Footer;

.contactFormContainer {  
 display: flex;  
 margin: 10%;  
}  
  
.deleteButton {  
 background-color: red;  
 color: white;  
 border: none;  
 cursor: pointer;  
 font-size: 16px;  
}

.contactFormContainer {  
 display: flex;  
 margin: 10%;  
}  
  
.deleteButton {  
 background-color: red;  
 color: white;  
 border: none;  
 cursor: pointer;  
 font-size: 16px;  
}

import React, { useEffect, useState } from 'react';  
import './customers.css';  
import Table from '@mui/material/Table';  
import TableBody from '@mui/material/TableBody';  
import TableCell from '@mui/material/TableCell';  
import TableContainer from '@mui/material/TableContainer';  
import TableHead from '@mui/material/TableHead';  
import TableRow from '@mui/material/TableRow';  
import Paper from '@mui/material/Paper';  
import axios from 'axios';  
import { useHistory, useLocation } from "react-router-dom";  
  
const Customers = () => {  
 let [customerEnrollData, setCustomerEnrollData] = useState({});  
 let [loading, setLoading] = useState(true)  
 let history = useHistory();  
  
 let CustomersDataFn = () => {  
 axios.get('http://localhost:8000/customerEnroll/viewAll').then(  
 data => {  
 console.log(data)  
 setCustomerEnrollData(data.data)  
 setLoading(false)  
 },   
 err => console.log(err)  
 )  
 }  
 useEffect(() => {  
 CustomersDataFn();  
 }, [])  
  
 let handleDelete = (v) => {  
 axios.post(`http://localhost:8000/customerEnroll/delete/${v}`).then(  
 data => {  
 console.log(data)  
 CustomersDataFn()  
 },  
 err => {  
 console.log(err)  
 }  
 )  
 }   
  
 return (  
 <>  
 <div className="contactFormContainer">  
 <TableContainer component={Paper}>  
 <Table sx={{ minWidth: 650 }} aria-label="simple table">  
 <TableHead>  
 <TableRow>  
 <TableCell>Sr.No</TableCell>  
 <TableCell align="right">Name</TableCell>  
 <TableCell align="right">Email</TableCell>  
 <TableCell align="right">Phone</TableCell>  
 {/\* <TableCell align="right"></TableCell> \*/}  
 </TableRow>  
 </TableHead>  
 <TableBody>  
 {  
 (loading) ? (  
 ''  
 ) : (  
 (customerEnrollData) ? (  
 customerEnrollData.map((v,i) => (   
 <TableRow  
 key={i}  
 sx={{ '&:last-child td, &:last-child th': { border: 0 } }}  
 >  
 <TableCell component="th" scope="row">  
 {i+1}  
 </TableCell>  
 <TableCell align="right">{v.name}</TableCell>  
 <TableCell align="right">{v.email}</TableCell>  
 <TableCell align="right">{v.phone}</TableCell>  
 <TableCell align="right"><button onClick={() => history.push(`/property/${v.propertyId}`)}>View Property</button></TableCell>  
 <TableCell align="right"><button className="deleteButton" onClick={() => handleDelete(v.\_id)}>Delete</button></TableCell>  
   
 </TableRow>  
 ))  
 ) : (  
 ''  
 )  
 )  
 }  
 </TableBody>  
 </Table>  
 </TableContainer>  
 </div>  
 </>  
 )  
}  
  
export default Customers;

.customerEnroll {  
 height: 600px;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 align-items: center;  
 background: linear-gradient(  
 rgba(255, 255, 255, 0.5),  
 rgba(255, 255, 255, 0.5)  
 ),   
 url('../../images/customerEnroll.jpg');  
 background-size: cover;  
}  
  
.customerEnrollTitle {  
 font-size: 50px;  
 font-weight: bold;  
 color: white;  
}  
  
.customerEnrollForm {  
 margin-top: 20px;  
 display: flex;  
 flex-direction: column;  
}  
  
.customerEnrollForm label {  
 margin: 10px 0;  
 font-weight: 700;  
}  
  
.customerEnrollInput {  
 padding: 10px;  
 background-color: white;  
 border: none;  
 border-radius: 5px;  
}  
  
.customerEnrollButton {  
 margin-top: 20px;  
 cursor: pointer;  
 background-color: black;  
 border: none;  
 color: white;  
 border-radius: 10px;  
 padding: 10px;  
 font-weight: 700;  
}  
  
.customerEnrollLoginButton {  
 position: absolute;  
 top: 60px;  
 right: 20px;  
 background-color: lightcoral;  
 cursor: pointer;  
 border: none;  
 padding: 10px;  
 color: white;  
 border-radius: 10px;  
}

import React, { useEffect, useState } from 'react';  
import './customerEnroll.css';  
import 'react-notifications/lib/notifications.css';  
import {NotificationContainer, NotificationManager} from 'react-notifications';  
import { useHistory } from "react-router-dom";  
import axios from 'axios';  
  
const CustomerEnroll = () => {  
 let history = useHistory();  
 const [formData, setFormData] = useState({ propertyId: window.location.pathname.split('/')[2] });  
 const handleChange = (e) => {  
 e.preventDefault();  
 console.log(e.target.value,'e')  
 console.log(e.target.name, 'name')  
 setFormData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 // console.log(formData, 'formData in change')  
 }  
 const handleSubmit = () => {  
 console.log(formData, 'formData')  
 // dispatch(registerUser(formData));  
 axios.post('http://localhost:8000/customerEnroll/create', formData).then(  
 data => {  
 console.log(data, 'data in register')  
 NotificationManager.success('User Enrolled successfully');  
 setTimeout(() => {  
 history.push('/customerlogin')  
 }, 2000)  
 },  
 err => {  
 console.log(err)  
 }  
 )  
 }  
 return (  
 <div className="customerEnroll">  
 <NotificationContainer/>  
 <pan className="customerEnrollTitle">Customer Enroll</pan>  
 <div className="customerEnrollForm">  
 <label>Name</label>  
 <input className="customerEnrollInput" name="name" type="text" placeholder="Enter your name...." onChange={handleChange} />  
 <label>Email</label>  
 <input className="customerEnrollInput" name="email" type="text" placeholder="Enter your email...." onChange={handleChange}/>  
 <label>Phone</label>  
 <input className="customerEnrollInput" name="phone" type="text" placeholder="Enter your phone number...." onChange={handleChange}/>  
 <label>Password</label>  
 <input className="customerEnrollInput" name="password" type="password" placeholder="Enter your password...." onChange={handleChange} />  
 <button className="customerEnrollButton" onClick={() => handleSubmit() }>Submit</button>  
 </div>  
 {/\* <button className="registerLoginButton" onClick={() => handleClick('login')}>Login</button> \*/}  
 </div>  
 )  
}  
  
export default CustomerEnroll;

.contactFormContainer {  
 display: flex;  
 margin: 10%;  
}  
  
.deleteButton {  
 background-color: red;  
 color: white;  
 border: none;  
 cursor: pointer;  
 font-size: 16px;  
}

import React, { useEffect, useState } from 'react';  
import './contactform.css';  
import Table from '@mui/material/Table';  
import TableBody from '@mui/material/TableBody';  
import TableCell from '@mui/material/TableCell';  
import TableContainer from '@mui/material/TableContainer';  
import TableHead from '@mui/material/TableHead';  
import TableRow from '@mui/material/TableRow';  
import Paper from '@mui/material/Paper';  
import axios from 'axios';  
import { useHistory, useLocation } from "react-router-dom";  
  
const ContactForm = () => {  
 let [contactFormData, setContactFormData] = useState({});  
 let [loading, setLoading] = useState(true)  
 let history = useHistory();  
  
 let ContactDataFn = () => {  
 axios.get('http://localhost:8000/contactform/viewAll').then(  
 data => {  
 console.log(data)  
 setContactFormData(data.data)  
 setLoading(false)  
 },   
 err => console.log(err)  
 )  
 }  
 useEffect(() => {  
 ContactDataFn();  
 }, [])  
  
 let handleDelete = (v) => {  
 axios.post(`http://localhost:8000/contactform/delete/${v}`).then(  
 data => {  
 console.log(data)  
 ContactDataFn()  
 },  
 err => {  
 console.log(err)  
 }  
 )  
 }  
 let handleSubmit = (v) => {  
 axios.post('http://localhost:8000/contactform/sendEmail', v).then(  
 data => {  
 console.log(data)  
 },  
 err => {  
 console.log(err)  
 }  
 )  
 }   
  
 return (  
 <>  
 <div className="contactFormContainer">  
 <TableContainer component={Paper}>  
 <Table sx={{ minWidth: 650 }} aria-label="simple table">  
 <TableHead>  
 <TableRow>  
 <TableCell>Sr.No</TableCell>  
 <TableCell align="right">Name</TableCell>  
 <TableCell align="right">Email</TableCell>  
 <TableCell align="right">Phone</TableCell>  
 {/\* <TableCell align="right"></TableCell> \*/}  
 </TableRow>  
 </TableHead>  
 <TableBody>  
 {  
 (loading) ? (  
 ''  
 ) : (  
 (contactFormData) ? (  
 contactFormData.map((v,i) => (   
 <TableRow  
 key={i}  
 sx={{ '&:last-child td, &:last-child th': { border: 0 } }}  
 >  
 <TableCell component="th" scope="row">  
 {i+1}  
 </TableCell>  
 <TableCell align="right">{v.firstName}</TableCell>  
 <TableCell align="right">{v.email}</TableCell>  
 <TableCell align="right">{v.phone}</TableCell>  
 <TableCell align="right"><button onClick={() => history.push(`/property/${v.propertyId}`)}>View Property</button></TableCell>  
 <TableCell align="right"><button onClick={() => handleSubmit(v)}>Send Email</button></TableCell>  
 <TableCell align="right"><button className="deleteButton" onClick={() => handleDelete(v.\_id)}>Delete</button></TableCell>  
   
 </TableRow>  
 ))  
 ) : (  
 ''  
 )  
 )  
 }  
 </TableBody>  
 </Table>  
 </TableContainer>  
 </div>  
 </>  
 )  
}  
  
export default ContactForm;

import React from 'react';  
import './homeContentOne.css';  
  
const HomeContentOne = () => {  
  
 return (  
 <div className="HomeContentOneContainer">  
 <div className="HomeContentOneContainerTitle">  
 <h2>Does new appeal to you? We’re proud to present</h2>  
 </div>  
 <div className="HomeContentOneContainerContent">  
 <div>  
 <div className="content1image"></div>  
 <h4>St louis</h4>  
 </div>  
 <div>  
 <div className="content2image"></div>  
 <h4>Kansas city</h4>  
 </div>  
 <div>  
 <div className="content3image"></div>  
 <h4>Wichita</h4>  
 </div>  
 {/\* <div className="content2image"></div> \*/}  
 {/\* <div className="content3image"></div> \*/}  
 </div>  
 </div>  
 )  
}  
  
export default HomeContentOne;

.HomeContentOneContainer {  
 height: 500px;  
 background: #ffffff;  
 display: flex;  
 flex-direction: column;  
 /\* align-items: center; \*/  
 /\* justify-content: center; \*/  
}  
  
.HomeContentOneContainerContent {  
 display: flex;  
 align-items: center;  
 justify-content: space-around;  
}  
  
.HomeContentOneContainerTitle h2 {  
 margin: 30px;  
 font-size: 22px;  
}  
  
.HomeContentOneContainerContent h4 {  
 margin-left: 20px;  
 font-size: 18px;  
}  
  
.content1image {  
 background: url('../../images/homecontent1.jpg');  
 height: 250px;  
 /\* width: 30%; \*/  
 width: 400px;  
 background-size: cover;  
}  
  
.content2image {  
 background: url('../../images/homecontent2.jpg');  
 height: 250px;  
 width: 400px;  
 background-size: cover;  
}  
  
.content3image {  
 background: url('../../images/homecontent3.jpg');  
 height: 250px;  
 width: 400px;  
 background-size: cover;  
}

import React from 'react';  
import './homeContentTwo.css';  
  
const HomeContentTwo = () => {  
  
 return (  
 <div className="HomeContentTwoContainer">  
 <div className="HomeContentTwoContainerTitle">  
 <h2>Or is classic more your style? See the winners of our resident apartment design contest</h2>  
 </div>  
 <div className="HomeContentTwoContainerContent">  
 <div>  
 <div className="content1imageTwo"></div>  
 <h4>St louis</h4>  
 </div>  
 <div>  
 <div className="content2imageTwo"></div>  
 <h4>Kansas city</h4>  
 </div>  
 <div>  
 <div className="content3imageTwo"></div>  
 <h4>Wichita</h4>  
 </div>  
 </div>  
 </div>  
 )  
}  
  
export default HomeContentTwo;

.HomeContentTwoContainer {  
 height: 500px;  
 background: #ffffff;  
 display: flex;  
 flex-direction: column;  
 /\* align-items: center; \*/  
 /\* justify-content: center; \*/  
}  
  
.HomeContentTwoContainerContent {  
 display: flex;  
 align-items: center;  
 justify-content: space-around;  
}  
  
.HomeContentTwoContainerTitle h2 {  
 margin: 30px;  
 font-size: 22px;  
}  
  
.HomeContentTwoContainerContent h4 {  
 margin-left: 20px;  
 font-size: 18px;  
}  
  
.content1imageTwo {  
 background: url('../../images/room1.jpg');  
 height: 250px;  
 /\* width: 30%; \*/  
 width: 400px;  
 background-size: cover;  
}  
  
.content2imageTwo {  
 background: url('../../images/room2.jpg');  
 height: 250px;  
 width: 400px;  
 background-size: cover;  
}  
  
.content3imageTwo {  
 background: url('../../images/room3.jpg');  
 height: 250px;  
 width: 400px;  
 background-size: cover;  
}

.homeSearchContainer {  
 background: url('../../images/homesearch.jpg');  
 /\* object-fit: cover; \*/  
 background-position: center;  
 height: 500px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
}  
  
.homeSearchInsideBox {  
 background-color: rgba(255,255,255,0.6);;  
 /\* opacity: 0.6; \*/  
 height: 300px;  
 width: 90%;  
 display: flex;  
 flex-wrap: wrap;  
 align-items: center;  
 justify-content: space-evenly;  
}  
  
.homeSearchInsideBox select {  
 /\* margin-left: 50px; \*/  
 padding: 15px 10px;  
 background:#ffffff;  
 border-radius: 5px;  
 border-color: 1px black;  
 font-size: 12px;  
 text-align: center;  
}  
  
.homeSearchInsideBox option {  
   
}  
  
.datepickerr {  
 padding: 15px 10px;  
 /\* margin-left: 50px; \*/  
 background: #ffffff;  
 border-radius: 5px;  
 border-color: 1px black;  
 font-size: 12px;  
}  
  
.submitButtonHome {  
 padding: 15px 20px;  
 /\* margin-left: 50px; \*/  
 background: #00adef;  
 border-radius: 5px;  
 border: none;  
 font-size: 12px;  
 /\* margin-right: 150px; \*/  
 color: #ffffff;  
 cursor: pointer;  
}  
  
/\* .react-datepicker-wrapper {  
 width: 160px;  
 padding: 0;  
} \*/

import React, { useEffect, useState } from 'react';  
import DatePicker from 'react-datepicker';  
import './homesearch.css';  
import "react-datepicker/dist/react-datepicker.css";  
import axios from 'axios';  
import { useHistory } from "react-router-dom";  
  
const HomeSearch = () => {  
 // const [value, onChange] = useState(new Date());  
 const [startDate, setStartDate] = useState(new Date());  
 const [areaStreetData, setAreaStreetData] = useState({})  
 const [loadingArea, setLoadingArea] = useState(true)  
 const [homeSearchData, setHomeSearchData] = useState({ date: new Date()})  
 const history = useHistory();  
  
 let GetAreaAllDataFn = () => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.get('http://localhost:8000/area/viewAllData', config).then(  
 data => {  
 console.log(data, 'area view all data')  
 setAreaStreetData(data.data)  
 setLoadingArea(false)  
 }  
 )  
 }  
  
 let handleChange = (v, e) => {  
 // setAreaStreetData(prevState => ({  
 // ...prevState,  
 // [e.target.name] : e.target.value  
 // }))  
 console.log(e)  
 // console.log(e)  
 if (v === "homesearch") {  
 setHomeSearchData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 }  
 if (v === "date") {  
 setHomeSearchData(prevState => ({  
 ...prevState,  
 date : e  
 }))  
 }  
 console.log(homeSearchData)  
 }  
  
 let handlesubmit = () => {  
 console.log(homeSearchData)  
 axios.post('http://localhost:8000/property/findFilteredProperties', homeSearchData).then(  
 data => {  
 console.log(data)  
 history.push('/propertysearch', data.data)  
 },  
 err => console.log(err)  
 )  
 }  
  
 useEffect(() => {  
 GetAreaAllDataFn();  
 }, [])  
  
 return (  
 <div className="homeSearchContainer">  
 {/\* {console.log(areaStreetData, 'areaStreetData')} \*/}  
 <div className="homeSearchInsideBox">  
 { (loadingArea) ? (  
 ''  
 ) : (  
 (areaStreetData) ? (  
 <select name="street" onChange={(e) => handleChange('homesearch', e)} type="text">  
 { areaStreetData.map((v,i) => {  
 return <option key={i} value={v.name}>{v.name}</option>  
 }) }  
 </select>   
 ) : (  
 ''  
 )  
 )}  
 <select name="bedrooms" type="text" onChange={(e) => handleChange('homesearch', e)}>  
 <option value="--">Bedrooms</option>  
 <option value="0">Studio</option>  
 <option value="1">1+</option>  
 <option value="2">2+</option>  
 <option value="3">3+</option>  
 </select>  
 <select name="price" type="text" onChange={(e) => handleChange('homesearch', e)}>  
 <option value="--">Price</option>  
 <option value="600">Less than $600</option>  
 <option value="1200">Less than $1200</option>  
 <option value="2000">Less than $2000</option>  
 <option value="3000">Less than $3000</option>  
 </select>  
 <div>  
 <DatePicker name="date" className="datepickerr" selected={homeSearchData.date} onChange={(e) => handleChange('date', e)} />  
 </div>  
 <button type="button" className="submitButtonHome" onClick={handlesubmit}>Submit</button>  
 </div>  
 </div>  
 )  
}  
  
export default HomeSearch;

import React, { useEffect, useState } from 'react';  
import './properties.css';  
import DatePicker from 'react-datepicker';  
import "react-datepicker/dist/react-datepicker.css";  
import Table from '@mui/material/Table';  
import TableBody from '@mui/material/TableBody';  
import TableCell from '@mui/material/TableCell';  
import TableContainer from '@mui/material/TableContainer';  
import TableHead from '@mui/material/TableHead';  
import TableRow from '@mui/material/TableRow';  
import Paper from '@mui/material/Paper';  
import axios from 'axios';  
  
const Properties = () => {  
 let [showPropertiesViewAllData, setShowPropertiesViewAllData] = useState(false)  
 let [showPropertiesAddProperty, setShowPropertiesAddProperty] = useState(false)  
 let [showPropertiesEditProperty, setShowPropertiesEditProperty] = useState(false)  
 let [addpropertyData, setAddpropertyData] = useState({ date: new Date(), occupied: 'false'})  
 let [streetData, setStreetData] = useState({})  
 let [allPropertiesData, setAllPropertiesData] = useState({})  
 let [editPropertyData, setEditPropertyData] = useState({})  
  
 let handleClick = (v, id) => {  
 if (v === "addPropertyButton") {  
 setShowPropertiesAddProperty(true)  
 setShowPropertiesViewAllData(false)  
 }  
 if (v === "propertiesEditButton") {  
 setShowPropertiesEditProperty(true)  
 setShowPropertiesViewAllData(false)  
 GetSpecificPropertyFn(id)  
 }  
 if (v === "propertiesDeleteButton") {  
 axios.post(`http://localhost:8000/property/delete/${id}`, {}).then(  
 data => {  
 console.log(data, 'data after delete')  
 GetAllPropertiesFn()  
 },  
 err => console.log(err)  
 )  
 }  
 }  
  
 let handleCancel = (v) => {  
 if (v === "propertiesAddPropertyCancelButton") {  
 setShowPropertiesAddProperty(false)  
 setShowPropertiesViewAllData(true)  
 }  
 if (v === "propertiesEditPropertyCancelButton") {  
 setShowPropertiesEditProperty(false)  
 setShowPropertiesViewAllData(true)  
 }  
 }  
  
 let handleChange = (v, e) => {  
 if (v === "addProperty") {  
 setAddpropertyData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 }  
 if (v === "date") {  
 setAddpropertyData(prevState => ({  
 ...prevState,  
 date : e  
 }))  
 }  
 if (v === "EditProperty") {  
 setEditPropertyData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 }  
 }  
  
 let handleSubmit = (v) => {  
 if (v === "propertiesAddPropertySubmitButton") {  
 console.log(addpropertyData, 'add property data')  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.post('http://localhost:8000/property/create', addpropertyData , config).then(  
 data => {  
 console.log(data, 'property data')  
 GetAllPropertiesFn()  
 setShowPropertiesViewAllData(true)  
 setShowPropertiesAddProperty(false)  
 }  
 )  
 }  
 if (v === "propertiesEditPropertySubmitButton") {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.post('http://localhost:8000/property/updateProperty', editPropertyData, config).then(  
 data => {  
 console.log(data, ' edit property data')  
 GetAllPropertiesFn()  
 setShowPropertiesViewAllData(true)  
 setShowPropertiesEditProperty(false)  
 }  
 )  
 }  
 }  
  
 let GetAreaAllDataFn = () => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.get('http://localhost:8000/area/viewAllData', config).then(  
 data => {  
 console.log(data, 'area view all data')  
 setStreetData(data.data)  
 }  
 )  
 }  
  
 let GetAllPropertiesFn = () => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.get('http://localhost:8000/property/viewAllProperties', config).then(  
 data => {  
 console.log(data, 'property view all data')  
 setAllPropertiesData(data.data)  
 }  
 )  
 }  
  
 let GetSpecificPropertyFn = (id) => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.post(`http://localhost:8000/property/getSpecificProperty/${id}`, {} ,config).then(  
 data => {  
 console.log(data, 'specific property data')  
 setEditPropertyData(data.data)  
 }  
 )  
 }  
  
 useEffect(() => {  
 GetAreaAllDataFn()  
 GetAllPropertiesFn()  
 setTimeout(() => {  
 setShowPropertiesViewAllData(true)  
 }, 1000)  
   
 }, [])  
  
 return (  
 <>  
 <div className="propertiesContainer">  
 <button className="addPropertyButton" onClick={() => handleClick('addPropertyButton')} type="button">Add Property</button>  
 { (showPropertiesViewAllData) ? (  
 <TableContainer component={Paper}>  
 <Table sx={{ minWidth: 650 }} aria-label="simple table">  
 <TableHead>  
 <TableRow>  
 <TableCell>S.no</TableCell>  
 <TableCell align="right">Name</TableCell>  
 <TableCell align="right">Price</TableCell>  
 </TableRow>  
 </TableHead>  
 { (allPropertiesData) ? (  
 <TableBody>  
 {allPropertiesData.map((row, i) => (  
 <TableRow  
 key={row.name}  
 sx={{ '&:last-child td, &:last-child th': { border: 0 } }}  
 >  
 <TableCell component="th" scope="row">  
 {i+1}  
 </TableCell>  
 <TableCell align="right">{row.name}</TableCell>  
 <TableCell align="right">{row.price}</TableCell>  
 <TableCell align="right" onClick={() => handleClick('propertiesEditButton', row.\_id)}><button className="propertiesEditButton">Edit</button></TableCell>  
 <TableCell align="right"><button className="propertiesDeleteButton" onClick={() => handleClick('propertiesDeleteButton', row.\_id)}>Delete</button></TableCell>  
 {/\* <TableCell align="right">{row.protein}</TableCell> \*/}  
 </TableRow>  
 ))}  
 </TableBody>  
 ) : (  
 ''  
 )}  
   
 </Table>  
 </TableContainer>  
 ) : (  
 ''  
 )}  
 { (showPropertiesAddProperty) ? (  
 <div className="propertiesAddPropertyContainer">  
 <label>Name</label>  
 <input type="text" name="name" onChange={(e) => handleChange('addProperty', e)} placeholder="Enter name" />  
 <label>Price</label>  
 <input type="text" name="price" onChange={(e) => handleChange('addProperty', e)} placeholder="Enter Price" />  
 <label>Area of the property(sqft)</label>  
 <input type="text" name="propertyArea" onChange={(e) => handleChange('addProperty', e)} placeholder="Enter area of the property" />  
 { (streetData) ? (  
   
 <select className="propertiesSelectOne" name="street" onChange={(e) => handleChange('addProperty', e)} type="text">  
 { streetData.map((v,i) => {  
 return <option value={v.name}>{v.name}</option>  
 }) }  
 </select>   
 ) : (  
 ''  
 )}  
 {/\* {console.log(streetData, 'streetdata')} \*/}  
 {/\* <select className="propertiesSelectOne" name="street" onChange={(e) => handleChange('addProperty', e)} type="text">  
 <option value="--">Street</option>  
 <option value="chicago">21st jump street</option>  
 <option value="kansascity">22nd jump street</option>  
 <option value="stlousis">23rd jump street</option>  
 </select> \*/}  
 <select className="propertiesSelectTwo" name="bedrooms" onChange={(e) => handleChange('addProperty', e)} type="text">  
 <option value="--">Bedrooms</option>  
 <option value="0">Studio</option>  
 <option value="1">1+</option>  
 <option value="2">2+</option>  
 <option value="3">3+</option>  
 </select>  
 <select className="propertiesSelectTwo" name="areaStatus" onChange={(e) => handleChange('addProperty', e)} type="text">  
 <option value="--">Area Status</option>  
 <option value="safe">Safe</option>  
 <option value="studentfriendly">Student Friendly</option>  
 <option value="notsafe">Not safe</option>  
 </select>  
 <DatePicker name="date" className="datepickerr" selected={addpropertyData.date} onChange={(e) => handleChange('date', e)} />  
 <div>  
 <button type="button" onClick={() => handleSubmit('propertiesAddPropertySubmitButton')}>Submit</button>  
 <button type="button" onClick={() => handleCancel('propertiesAddPropertyCancelButton')}>Cancel</button>  
 </div>  
 </div>  
 ) : (  
 ''  
 )}  
 { (showPropertiesEditProperty) ? (  
 <div className="propertiesEditPropertyContainer">  
 <label>Name</label>  
 <input type="text" name="name" placeholder="Enter name" onChange={(e) => handleChange('EditProperty', e)} value={editPropertyData.name} />  
 <label>Price</label>  
 <input type="text" name="price" placeholder="Enter Price" onChange={(e) => handleChange('EditProperty', e)} value={editPropertyData.price} />  
 <label>Area of the property(sqft)</label>  
 <input type="text" name="propertyArea" placeholder="Enter area of the property" onChange={(e) => handleChange('EditProperty', e)} value={editPropertyData.propertyArea}/>  
 { (streetData) ? (  
   
 <select className="propertiesSelectOne" name="street" onChange={(e) => handleChange('EditProperty', e)} value={editPropertyData.street} type="text">  
 { streetData.map((v,i) => {  
 return <option value={v.name}>{v.name}</option>  
 }) }  
 </select>   
 ) : (  
 ''  
 )}  
 <select className="propertiesSelectTwo" name="bedrooms" type="text" onChange={(e) => handleChange('EditProperty', e)} value={editPropertyData.bedrooms}>  
 <option value="--">Bedrooms</option>  
 <option value="0">Studio</option>  
 <option value="1">1+</option>  
 <option value="2">2+</option>  
 <option value="3">3+</option>  
 </select>  
 <select className="propertiesSelectTwo" name="areaStatus" onChange={(e) => handleChange('EditProperty', e)} value={editPropertyData.areaStatus} type="text">  
 <option value="--">Area Status</option>  
 <option value="safe">Safe</option>  
 <option value="studentfriendly">Student Friendly</option>  
 <option value="notsafe">Not safe</option>  
 </select>  
 <p className="datepickerr">Date: {editPropertyData.date}</p>  
 <div>  
 <button type="button" onClick={() => handleSubmit('propertiesEditPropertySubmitButton')} >Submit</button>  
 <button type="button" onClick={() => handleCancel('propertiesEditPropertyCancelButton')}>Cancel</button>  
 </div>  
 </div>  
 ) : (  
 ''  
 )}  
 </div>  
 </>  
 )  
}  
  
export default Properties;

.propertiesContainer {  
 /\* background: blue; \*/  
 height: 500px;  
 margin: 5% 10%;  
}  
  
.addPropertyButton {  
 padding: 15px 20px;  
 /\* border: none; \*/  
 border-radius: 10px;  
 margin: 10%;  
}  
  
.propertiesAddPropertyContainer {  
 /\* background: blueviolet; \*/  
 height: 300px;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
}  
  
.propertiesAddPropertyContainer input {  
 padding: 10px 20px;  
 margin: 10px;  
 border-radius: 10px;  
 /\* border: none; \*/  
}  
  
.propertiesAddPropertyContainer label {  
 color: black;  
 font-weight: 800;  
}  
  
.propertiesSelectOne {  
 margin-top: 20px;  
 padding: 10px 40px;  
 text-align: center;  
 border-radius: 10px;  
}  
  
.propertiesSelectTwo {  
 margin-top: 20px;  
 padding: 10px 60px;  
 text-align: center;  
 border-radius: 10px;  
}  
  
.propertiesAddPropertyContainer button {  
 padding: 10px 20px;  
 border-radius: 10px;  
 margin: 20px;  
 color: #ffffff;  
 background: black;  
}  
  
.propertiesEditPropertyContainer {  
 /\* background: blueviolet; \*/  
 height: 300px;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
}  
  
.propertiesEditPropertyContainer button {  
 padding: 10px 20px;  
 border-radius: 10px;  
 margin: 20px;  
 color: #ffffff;  
 background: black;  
}  
  
.propertiesEditPropertyContainer label {  
 color: black;  
 font-weight: 800;  
}  
  
.propertiesEditPropertyContainer input {  
 padding: 10px 20px;  
 margin: 10px;  
 border-radius: 10px;  
 /\* border: none; \*/  
}  
  
.propertiesDeleteButton {  
 background: red;  
 padding: 10px 15px;  
 border-radius: 10px;  
 color: white;  
}  
  
.propertiesEditButton {  
 background: blue;  
 padding: 10px 15px;  
 border-radius: 10px;  
 color: white;  
}  
  
.propertiesAddPropertyContainer .datepickerr {  
 margin-left: 38%;  
 padding: 10px 30px;  
}  
  
.propertiesEditPropertyContainer .datepickerr {  
 margin-left: 2%;  
 padding: 10px 10px;  
 color: black;  
 font-weight: 900;  
}

import React, { useEffect, useState } from 'react';  
import './topbar.css';  
  
const TopBar = () => {  
  
 return (   
 <>  
 <div className="topbarContainer">  
 <div className="topbarLeft">  
 <h2>Property and Rent</h2>  
 </div>  
 <div className="topbarRight">  
 <ul>  
 <li>Home</li>  
 <li>About</li>  
 <li>Services</li>  
 <li>Contact</li>  
 </ul>   
 </div>  
 </div>  
 </>  
 )  
}  
  
export default TopBar;

.topbarContainer {  
 background-color: #f8f9fa;  
 height: 100px;  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
}  
  
.topbarLeft {  
 /\* flex: 4; \*/  
}  
  
.topbarLeft h2 {  
 font-size: 25px;  
 margin-left: 20px;  
 padding-left: 50px;  
 cursor: pointer;  
}  
  
.topbarRight {  
 /\* flex: 8; \*/  
}  
  
.topbarRight ul {  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 list-style: none;  
 margin: 30px;  
 font-weight: 500;  
 font-family: 'MarkOT-Medium';  
 font-size: 15px;  
}  
  
.topbarRight li {  
 margin-right: 30px;  
 padding-right: 30px;  
 cursor: pointer;  
}

import React from 'react';  
import { Route, Redirect } from 'react-router-dom';  
import { isLogin } from './utils';  
  
const PrivateRoute = ({ component: Component, ...rest }) => {  
 return (  
 <Route  
 {...rest}  
 render={(props) =>   
 isLogin() ? <Component /> : <Redirect to={{ pathname: '/login' }} />  
 }   
 />  
 )  
}  
  
export default PrivateRoute;

import React from 'react';  
import { Route, Redirect } from 'react-router-dom';  
import { isLogin } from "./utils";  
  
const PublicRoute = ({ component: Component, restricted, ...rest }) => {  
 return (  
 <Route  
 {...rest}  
 render={(props) => isLogin() && restricted ? (  
 <Redirect to ='/home' />  
 ) : (  
 <Component {...props} />  
 )  
 }  
 />  
 )  
}  
  
export default PublicRoute;

const TOKEN\_KEY = "token";  
  
export const logout = () => {  
 localStorage.removeItem(TOKEN\_KEY);  
}  
  
export const isLogin = () => {  
 if (localStorage.getItem(TOKEN\_KEY)) {  
 return true;  
 }  
 return false;  
}

.customerDashboardContainer {  
 height: 700px;  
 position: relative;  
}  
  
.containerbox {  
 position: absolute;  
 background-color: gray;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
 top: 30%;  
 left: 30%;  
 color: #ffffff;  
 padding: 20px 40px;  
}  
  
.containerbox p {  
 font-size: 18px;  
}  
  
.logoutButton {  
 background: red;  
 color: #ffffff;  
}

import React, { useState, useEffect } from 'react';  
import './adminDashboard.css';  
import Table from '@mui/material/Table';  
import TableBody from '@mui/material/TableBody';  
import TableCell from '@mui/material/TableCell';  
import TableContainer from '@mui/material/TableContainer';  
import TableHead from '@mui/material/TableHead';  
import TableRow from '@mui/material/TableRow';  
import Paper from '@mui/material/Paper';  
import axios from 'axios';  
import Properties from '../../components/properties/Properties';  
import ContactForm from '../../components/contactForm/ContactForm';  
import Customers from '../../components/customers/Customers';  
import { useHistory } from "react-router-dom";  
  
const AdminDashboard = () => {  
 let [showAreaCreateForm, setShowAreaCreateForm] = useState(false)  
 let [showAreaViewAll, setShowAreaViewAll] = useState(false)  
 let [showAreaEditData, setShowAreaEditData] = useState(false)  
 let [areaCreateFormData, setAreaCreateFormData] = useState({})  
 let [areaViewAllData, setAreaViewAllData] = useState({})  
 let [areaViewAllDataLoading, setAreaViewAllDataLoading] = useState(true)  
 let [areaEditDataApi, setAreaEditDataApi] = useState({})  
 let [showAreas, setShowAreas] = useState(false)  
 let [showProperties, setShowProperties] = useState(false)  
 let [showContactForm, setShowContactForm] = useState(false)  
 let [showCustomers, setShowCustomers] = useState(false)  
 let history = useHistory();  
  
 let handleClick = (v, id) => {  
 console.log(v, 'v')  
 if (v === "AreaCreateForm") {  
 setShowAreaCreateForm(true)  
 setShowAreaViewAll(false)  
 setShowAreaEditData(false)  
 }  
 if (v === "AreaEditData") {  
 setShowAreaEditData(true)  
 setShowAreaViewAll(false)  
 GetAreaSpecificIdFn(id)  
 }  
 }  
  
 let GetAreaSpecificIdFn = (id) => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.get(`http://localhost:8000/area/specificArea/${id}`, config).then(  
 data => {  
 console.log(data, 'area specific data')  
 setAreaEditDataApi(data.data)  
 },  
 err => console.log(err)  
 )  
 }  
  
 let GetAreaViewAllDataFn = () => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.get('http://localhost:8000/area/viewAllData', config).then(  
 data => {  
 console.log(data, 'area view all data')  
 setAreaViewAllData(data.data)  
 setAreaViewAllDataLoading(false)  
 }  
 )  
 }  
   
 let handleCancel = (v) => {  
 console.log(v, 'v')  
 if (v === "AreaCreateFormCancel") {  
 GetAreaViewAllDataFn()  
 setShowAreaCreateForm(false)  
 setShowAreaViewAll(true)  
 }  
 if (v === "areaEditDataCancel") {  
 setShowAreaEditData(false)  
 setShowAreaViewAll(true)  
 }  
 }  
  
 let handleDelete = (v, id) => {  
  
 if (v === "AreaEditData") {  
 console.log('delete')  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.post(`http://localhost:8000/area/delete/${id}`, {}, config).then(  
 data => {  
 console.log(data)  
 GetAreaViewAllDataFn()  
 },  
 err => console.log(err)  
 )  
 }  
 }  
  
 let handleChange = (v, e) => {  
 if (v === "areaCreateFormChange") {  
 setAreaCreateFormData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 }  
 if (v === "AreaEditDataChange") {  
 setAreaEditDataApi(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 }  
 }  
  
 let handleSubmit = (v, id) => {  
 if (v === "areaCreateFormSubmit") {  
 console.log(areaCreateFormData,'area submit')  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.post('http://localhost:8000/area/create', areaCreateFormData, config).then(  
 areaCreateFormData => {  
 console.log(areaCreateFormData, 'data in area create form')  
 setShowAreaViewAll(true)  
 setShowAreaCreateForm(false)  
 GetAreaViewAllDataFn()  
 },  
 err => console.log(err)  
 )  
 }  
 if (v === "areaEditDataSubmit") {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.post(`http://localhost:8000/area/updateAreaData/${id}`, areaEditDataApi, config).then(  
 data => {  
 console.log(data, 'after edit submit')  
 setShowAreaViewAll(true)  
 setShowAreaEditData(false)  
 GetAreaViewAllDataFn()  
 },  
 err => {  
 console.log(err)  
 }  
 )  
 }   
 }  
  
 useEffect(() => {  
 GetAreaViewAllDataFn();  
 setShowAreaViewAll(true)  
 }, [])  
  
 let handleComponents = (v) => {  
 if (v === 'areas') {  
 setShowAreas(true)  
 setShowProperties(false)  
 setShowContactForm(false)  
 setShowCustomers(false)  
 }  
 if (v === 'properties') {  
 setShowAreas(false)  
 setShowProperties(true)  
 setShowContactForm(false)  
 setShowCustomers(false)  
 }  
 if (v === 'contactForm') {  
 setShowContactForm(true)  
 setShowProperties(false)  
 setShowAreas(false)  
 setShowCustomers(false)  
 }  
 if (v === 'customers') {  
 setShowContactForm(false)  
 setShowProperties(false)  
 setShowAreas(false)  
 setShowCustomers(true)  
 }  
 }  
  
 let handleLogout = () => {  
 localStorage.removeItem('token')  
 history.push('/login')  
 }  
  
  
 return (  
 <>  
 <div className="adminDashboardContainer">  
 <div className="leftContainer">  
 <div className="topHeading">Admin Dashboard</div>  
 <div className="selection" onClick={() => handleComponents('areas')}>Areas</div>  
 <div className="selection" onClick={() => handleComponents('properties')}>Properties</div>  
 <div className="selection" onClick={() => handleComponents('contactForm')}>Contact form</div>  
 <div className="selection" onClick={() => handleComponents('customers')}>Customers</div>  
 <div className="selection"><button onClick={handleLogout}>Logout</button></div>  
 </div>  
 <div className="rightContainer">  
 { (showAreas) ? (  
 <>  
 <button className="AreaCreateFormButton" onClick={() => handleClick('AreaCreateForm')}>Create</button>  
 { (showAreaCreateForm) ? (  
 <div className="AreaCreateForm">  
 <label>Name</label>  
 <input name="name" type="text" onChange={(e) => handleChange('areaCreateFormChange', e)} />  
 <button className="createFormsubmitButton" type="button" onClick={() => handleSubmit('areaCreateFormSubmit')}>Submit</button>  
 <button className="createFormcancelButton" type="button" onClick={() => handleCancel('AreaCreateFormCancel')}>Cancel</button>  
 </div>  
 ) : ''}  
 { (showAreaViewAll) ? (  
 <div className="AreaViewAll">  
 <TableContainer className="tableContainer" component={Paper}>  
 <Table sx={{ minWidth: 650 }} aria-label="simple table">  
 <TableHead>  
 <TableRow>  
 <TableCell>No</TableCell>  
 <TableCell align="right">Name</TableCell>  
 </TableRow>  
 </TableHead>  
 <TableBody>  
 { (areaViewAllDataLoading) ? (  
 ''  
 ) : (  
 (areaViewAllData) ? (  
 areaViewAllData.map((v,i) => (  
 <TableRow  
 key={i}  
 sx={{ '&:last-child td, &:last-child th': { border: 0 } }}  
 >  
 <TableCell component="th" scope="row">  
 {i+1}  
 </TableCell>  
 <TableCell align="right">{v.name}</TableCell>  
 <TableCell align="right"><button type="button" className="AreaEditButton" onClick={() => handleClick('AreaEditData', v.\_id)}>edit</button></TableCell>  
 <TableCell align="right"><button type="button" className="AreaDeleteButton" onClick={() => handleDelete('AreaEditData', v.\_id)}>Delete</button></TableCell>  
 {/\* <TableCell align="right">{row.protein}</TableCell> \*/}  
 </TableRow>  
 ))  
 ) : (  
 ''  
 )  
 )}  
 </TableBody>  
 </Table>  
 </TableContainer>  
 </div>  
 ) : ''}  
 { (showAreaEditData) ? (  
  
 (areaEditDataApi) ? (  
 <div className="AreaEditData">  
 <p>edit data</p>  
 <label>Name</label>  
 <input name="name" type="text" onChange={(e) => handleChange('AreaEditDataChange', e)} value={areaEditDataApi.name} />  
 <button className="areaEditDatasubmitButton" type="button" onClick={() => handleSubmit('areaEditDataSubmit', areaEditDataApi.\_id)}>Submit</button>  
 <button className="areaEditDatacancelButton" type="button" onClick={() => handleCancel('areaEditDataCancel')}>Cancel</button>  
 {/\* <button type="button" onClick={() => handleCancel('editData')}>Cancel</button> \*/}  
 </div>  
 ) : (  
 ''  
 )  
 ) : ''}  
 </>  
 ): (  
 ''  
 )}  
 { (showProperties) ? (  
 <Properties />  
 ) : (  
 ''  
 )}  
 { (showContactForm) ? (  
 <ContactForm />  
 ) : (  
 ''  
 )}  
 { (showCustomers) ? (  
 <Customers />  
 ) : (  
 ''  
 )}  
   
 </div>  
 </div>  
 </>  
 )  
}  
  
export default AdminDashboard;

.adminDashboardContainer {  
 /\* background: black; \*/  
 height: 780px;  
 display: flex;  
 align-items: center;  
 /\* justify-content: space-around; \*/  
}  
  
.leftContainer {  
 color: #ffffff;  
 flex: 2;  
 border-right: 5px solid gray;  
 height: 780px;  
}  
  
.rightContainer {  
 color: #ffffff;  
 flex: 10  
}  
  
.topHeading {  
 background: black;  
 height: 80px;  
 /\* margin: 10px; \*/  
 text-align: center;  
 box-sizing: border-box;  
 padding-top: 22px;  
 font-size: 22px;  
 font-weight: 900;  
 /\* border-radius: 10px; \*/  
 border-bottom: 5px gray solid;  
}  
  
.selection {  
 background: #ffffff;  
 color: black;  
 height: 50px;  
 margin: 20px;  
 text-align: center;  
 padding-top: 12px;  
 box-sizing: border-box;  
 border-radius: 20px;  
 font-size: 18px;  
 font-weight: 500;  
 letter-spacing: 2px;  
 cursor: pointer;  
}  
  
.rightContainer {  
 height: 780px;  
}  
  
.AreaCreateFormButton {  
 margin-top: 10%;  
 margin-left: 10%;  
 padding: 10px 20px;  
 font-size: 18px;  
 border-radius: 10px;  
}  
  
.AreaCreateForm {  
 background: #D3D3D3;  
 height: 100px;  
 margin: 10% 15%;  
 /\* text-align: center; \*/  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 border-radius: 20px;  
}  
  
.AreaCreateForm label {  
 margin: 10px;  
 font-size: 18px;  
 color: black;  
 font-weight: 800;  
}  
  
.AreaCreateForm input {  
 padding: 15px 50px;  
 border-radius: 10px;  
 margin: 10px;  
}  
  
.createFormsubmitButton {  
 padding: 10px 20px;  
 margin: 30px;  
 font-size: 18px;  
 border-radius: 10px;  
}  
  
.createFormcancelButton {  
 padding: 10px 20px;  
 margin: 30px;  
 font-size: 18px;  
 border-radius: 10px;  
}  
  
.AreaViewAll {  
 /\* background: #D3D3D3; \*/  
 height: 400px;  
 margin: 5% 10%;  
 border-radius: 20px;  
}  
  
.AreaEditData {  
 /\* background: rgb(218, 206, 198); \*/  
 background: #D3D3D3;  
 height: 100px;  
 margin: 10% 15%;  
 /\* text-align: center; \*/  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 border-radius: 20px;  
}  
  
.areaEditDatasubmitButton {  
 padding: 10px 20px;  
 margin: 30px;  
 font-size: 18px;  
 border-radius: 10px;  
}  
  
.areaEditDatacancelButton {  
 padding: 10px 20px;  
 margin: 30px;  
 font-size: 18px;  
 border-radius: 10px;  
}  
  
.AreaEditData label {  
 margin: 10px;  
 font-size: 18px;  
 color: black;  
 font-weight: 800;  
}  
  
.AreaEditData input {  
 padding: 15px 50px;  
 border-radius: 10px;  
 margin: 10px;  
}  
  
.tableContainer {  
 color: white;  
 /\* height: 100px; \*/  
}  
  
.AreaEditButton {  
 background: blue;  
 color: white;  
 padding: 10px 20px;  
 border: none;  
 border-radius: 10px;  
}  
  
.AreaDeleteButton {  
 background: red;  
 color: white;  
 padding: 10px 20px;  
 border: none;  
 border-radius: 10px;  
}

import axios from 'axios';  
import React, { useEffect, useState } from 'react';  
import './customerDashboard.css';  
import { useHistory } from "react-router-dom";  
  
const CustomerDashboard = () => {  
 let [userData, setUserData] = useState({});  
 let [loading, setLoading] = useState(true);  
 let history = useHistory();  
  
 useEffect(() => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.post(`http://localhost:8000/customerEnroll/specificCustomer`, {}, config).then(  
 data => {  
 console.log(data, 'after submit')  
 setUserData(data.data)  
 setLoading(false)  
 },  
 err => {  
 console.log(err)  
 }  
 )  
 }, [])  
  
 let handleClick = () => {  
 localStorage.removeItem('token')  
 history.push('/customerlogin')  
 }  
 return (  
 <>  
 <div className="customerDashboardContainer">  
 <div className="containerbox">  
 { (loading) ? (  
 ''  
 ) : (  
 (userData) ? (  
 <>  
 <p>Name: {userData.name}</p>  
 <p>Email: {userData.email}</p>  
 <p>Phone: {userData.phone}</p>  
 <p><button onClick={() => history.push(`/property/${userData.propertyId}`)}>View Property</button></p>  
 <p><button className="logoutButton" onClick={handleClick}>Logout</button></p>  
 </>  
 ) : (  
 ''  
 )  
 )}  
 </div>  
 </div>  
 </>  
 )  
}  
  
export default CustomerDashboard;

.login {  
 height: 780px;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 align-items: center;  
 background: linear-gradient(  
 rgba(255, 255, 255, 0.5),  
 rgba(255, 255, 255, 0.5)  
 ),   
 url('../../images/login.jpg');  
 background-size: cover;  
}  
  
.loginTitle {  
 font-size: 50px;  
 color: #ffffff;  
 font-weight: 900;  
}  
  
.loginForm {  
 margin-top: 20px;  
 display: flex;  
 flex-direction: column;  
}  
  
.loginForm label {  
 margin: 10px 0;  
 font-weight: 700;  
}  
  
.loginInput {  
 padding: 10px;  
 background-color: white;  
 border: none;  
 border-radius: 5px;  
}  
  
.loginInput:focus {  
 outline: none;  
}  
  
.loginButton {  
 margin-top: 20px;  
 cursor: pointer;  
 background-color: black;  
 border: none;  
 color: white;  
 border-radius: 10px;  
 padding: 10px;  
 font-weight: 800;  
}  
  
.loginRegisterButton {  
 position: absolute;  
 top: 60px;  
 right: 20px;  
 background-color: white;  
 cursor: pointer;  
 border: none;  
 padding: 10px;  
 color: black;  
 border-radius: 10px;  
 font-weight: bold;  
 font-size: 24px;  
}

import './customerLogin.css';  
import { useHistory } from "react-router-dom";  
import { useState } from 'react';  
import axios from 'axios';  
// import { useDispatch } from 'react-redux';  
// import { loginUser } from '../../redux/actions/loginAction';  
  
export default function CustomerLogin() {  
 let history = useHistory();  
 // let dispatch = useDispatch();  
 let [formData, setFormData] = useState([])  
 const handleClick = (arg) => {  
 history.push(`/${arg}`)  
 }  
 const handleChange = (e) => {  
 console.log(e.target.name, 'name')  
 console.log(e.target.value, 'value')  
 setFormData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }  
 ))  
 }  
 const handleSubmit = () => {  
 // dispatch(loginUser(formData))  
 axios.post('http://localhost:8000/customerEnroll/login', formData).then(  
 response => {  
 console.log(response, 'response in login')  
 localStorage.setItem('token', response.data.token)  
 setTimeout(() => {  
 history.push('/customerDashboard')  
 }, 1000)  
 }  
 )  
 }  
 return (  
 <div className="login">  
 <pan className="loginTitle"> Customer Login</pan>  
 <div className="loginForm">  
 <label>Email</label>  
 <input className="loginInput" name="email" type="text" placeholder="Enter your email...." onChange={handleChange} />  
 <label>Password</label>  
 <input className="loginInput" name="password" type="password" placeholder="Enter your password...." onChange={handleChange} />  
 <button className="loginButton" onClick={handleSubmit}>Submit</button>  
 </div>  
 {/\* <button className="loginRegisterButton" onClick={() => handleClick('register')}>Register</button> \*/}  
 </div>  
 )  
}

import React, { useState, useEffect } from 'react';  
import Footer from '../../components/footer/Footer';  
import HomeContentOne from '../../components/homeContentOne/HomeContentOne';  
import HomeContentTwo from '../../components/homeContentTwo/HomeContentTwo';  
import HomeSearch from '../../components/homeSearch/HomeSearch';  
import TopBar from '../../components/topbar/TopBar';  
  
const Home = () => {  
  
 return (  
 <>  
 <TopBar />  
 <HomeSearch />  
 <HomeContentOne />  
 <HomeContentTwo />  
 <Footer />  
 </>  
 )  
}  
  
export default Home;

.login {  
 height: 780px;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 align-items: center;  
 background: linear-gradient(  
 rgba(255, 255, 255, 0.5),  
 rgba(255, 255, 255, 0.5)  
 ),   
 url('../../images/login.jpg');  
 background-size: cover;  
}  
  
.loginTitle {  
 font-size: 50px;  
 color: #ffffff;  
 font-weight: 900;  
}  
  
.loginForm {  
 margin-top: 20px;  
 display: flex;  
 flex-direction: column;  
}  
  
.loginForm label {  
 margin: 10px 0;  
 font-weight: 700;  
}  
  
.loginInput {  
 padding: 10px;  
 background-color: white;  
 border: none;  
 border-radius: 5px;  
}  
  
.loginInput:focus {  
 outline: none;  
}  
  
.loginButton {  
 margin-top: 20px;  
 cursor: pointer;  
 background-color: black;  
 border: none;  
 color: white;  
 border-radius: 10px;  
 padding: 10px;  
 font-weight: 800;  
}  
  
.loginRegisterButton {  
 position: absolute;  
 top: 60px;  
 right: 20px;  
 background-color: white;  
 cursor: pointer;  
 border: none;  
 padding: 10px;  
 color: black;  
 border-radius: 10px;  
 font-weight: bold;  
 font-size: 24px;  
}

import './login.css';  
import { useHistory } from "react-router-dom";  
import { useState } from 'react';  
import axios from 'axios';  
// import { useDispatch } from 'react-redux';  
// import { loginUser } from '../../redux/actions/loginAction';  
  
export default function Login() {  
 let history = useHistory();  
 // let dispatch = useDispatch();  
 let [formData, setFormData] = useState([])  
 const handleClick = (arg) => {  
 history.push(`/${arg}`)  
 }  
 const handleChange = (e) => {  
 console.log(e.target.name, 'name')  
 console.log(e.target.value, 'value')  
 setFormData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }  
 ))  
 }  
 const handleSubmit = () => {  
 // dispatch(loginUser(formData))  
 axios.post('http://localhost:8000/user/login', formData).then(  
 response => {  
 console.log(response, 'response in login')  
 localStorage.setItem('token', response.data.token)  
 setTimeout(() => {  
 history.push('/admin\_dashboard')  
 }, 1000)  
 }  
 )  
 }  
 return (  
 <div className="login">  
 <pan className="loginTitle">Login</pan>  
 <div className="loginForm">  
 <label>Email</label>  
 <input className="loginInput" name="email" type="text" placeholder="Enter your email...." onChange={handleChange} />  
 <label>Password</label>  
 <input className="loginInput" name="password" type="password" placeholder="Enter your password...." onChange={handleChange} />  
 <button className="loginButton" onClick={handleSubmit}>Submit</button>  
 </div>  
 {/\* <button className="loginRegisterButton" onClick={() => handleClick('register')}>Register</button> \*/}  
 </div>  
 )  
}

.propertyContainer {  
 background: #ffffff;  
 height: 1100px;  
 display: flex;  
 align-items: center;  
 justify-content: space-around;  
}  
  
.aboutProperty {  
 flex: 7;  
 margin: 30px;  
 padding: 30px;  
}  
  
.aboutProperty p {  
 font-size: 20px;  
 line-height: 1.6;  
}  
  
.contactproperty {  
 flex: 5;  
 display: flex;  
 align-items: center;  
 flex-direction: column;  
}  
  
.contactproperty h2 {  
 color: #262729;  
}  
  
.contactproperty input {  
 padding: 10px;  
 width: 80%;  
 margin: 10px 30px;  
 /\* border: none; \*/  
 border-radius: 10px;  
 border: 1px solid #a38c8c;  
 background: #fff;  
 font-size: 18px;  
 line-height: 1.6;  
}  
  
.contactproperty textarea {  
 padding: 10px;  
 width: 80%;  
 margin: 10px 30px;  
 /\* border: none; \*/  
 border-radius: 10px;  
 border: 1px solid #a38c8c;  
 background: #fff;  
 font-size: 18px;  
 line-height: 1.6;  
}  
  
.contactproperty input:focus {  
 outline: none;  
}  
  
.contactproperty textarea:focus {  
 outline: none;  
}  
  
.contactproperty button {  
 padding: 10px;  
 width: 84%;  
 margin: 10px 30px;  
 border-radius: 10px;  
 border: 1px solid #a38c8c;  
 background: #fff;  
 font-size: 18px;  
 line-height: 1.6;  
 background: #007bff;  
 color: #ffffff;  
 font-size: 24px;  
 cursor: pointer;  
}  
  
.aboutProperty h4 {  
 font-size: 24px;  
}  
  
.aboutProperty li {  
 font-size: 20px;  
 line-height: 1.6;  
}  
  
.amenities {  
 display: flex;  
 align-items: center;  
 justify-content: flex-start;  
}  
  
.amenities ul {  
 margin-right: 80px;  
 padding-right: 60px;  
}  
  
.propertyContainerPropertyImage {  
 /\* height: 100px; \*/  
}

import React, { useEffect, useState } from 'react';  
import TopBar from '../../components/topbar/TopBar';  
import 'react-notifications/lib/notifications.css';  
import {NotificationContainer, NotificationManager} from 'react-notifications';  
import './property.css';  
import image1 from '../../images/room1.jpg';  
import Footer from '../../components/footer/Footer';  
import axios from 'axios';  
  
const Property = () => {  
 let [propertyData, setPropertyData] = useState({})  
 let [loading, setLoading] = useState(true)  
 let [contactFormData, setContactFormData] = useState({propertyId: window.location.pathname.split('/')[2]})  
  
 useEffect(() => {  
 console.log(window.location.pathname.split('/')[2])  
 let propertyId = window.location.pathname.split('/')[2];  
 axios.post(`http://localhost:8000/property/getSpecificProperty/${propertyId}`).then(  
 data => {  
 setPropertyData(data.data)  
 console.log(data.data)  
 setLoading(false)  
 }  
 )  
 }, [])  
  
 let handleChange = (e) => {  
 setContactFormData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 }  
  
 let handleSubmit = () => {  
 axios.post('http://localhost:8000/contactform/create', contactFormData).then(  
 data => {  
 console.log(data)  
 NotificationManager.success('Form submitted successfully');  
 },  
 err => console.log(err)  
 )  
 }  
 return (  
 <>  
 <TopBar />  
 <div className="propertyContainer">  
 <NotificationContainer/>  
 <div className="aboutProperty">  
 {  
 (loading) ? (  
 ''  
 ) : (  
 (propertyData) ? (  
 <>  
 <p>Name: {propertyData.name}</p>  
 <p>Price: {propertyData.price}</p>  
 <p>Property Area: {propertyData.propertyArea}</p>  
 <p>Area Status: {propertyData.areaStatus}</p>  
 <p>Bedrooms: {propertyData.bedrooms}</p>  
 <p>Occupied: {(propertyData.occupied == "true") ? 'Yes occupied' : 'Not occupied'}</p>  
 </>   
 ) : (  
 ''  
 )  
 )  
 }  
 <h2>About Community</h2>  
 <p>1/2 month FREE on studios. 1 month FREE on select one bedrooms & all two bedrooms. Lease must start on or before December 1st, 2021. Terms apply. Nestled in the urban KC neighborhood of Hyde Park on Armour Blvd, 520 East embodies the modern apartment community with plenty of space to relax, work, learn, or play. This 7-story brick apartment was constructed in 2021 and features a top-floor indoor/outdoor pool lounge that will elevate the living experience and entice residents to never leave home. Speaking of home, each resident will enjoy a washer and dryer, smart entry locks, sleek two-toned cabinets, wood-style flooring, LED lighting, an icemaker, and no-fee access to a smart package collection room, a state-of-the-art fitness center, study rooms, and bike storage. If residents feel the need to venture, the surrounding area of Midtown has plenty to offer with a wide variety local restaurants, parks, and shopping. Some of the best Kansas City has to offer! 520 East is just a 5-minute drive or short bike ride to downtown Kansas City, The Plaza, and Westport. Now Renting!</p>  
 <h4>Building Amenities</h4>  
 <div className="amenities">  
 <ul>  
 <li>Bike Storage</li>  
 <li>Fitness Center</li>  
 <li>Package Service</li>  
 <li>Pets Allowed</li>  
 <li>Rooftop Deck</li>  
 </ul>  
 <ul>  
 <li>Business Center</li>  
 <li>Outdoor Grill</li>  
 <li>Parking Available</li>  
 <li>Swimming Pool</li>  
 <li>Smoke Free</li>  
 </ul>  
 </div>  
 </div>  
 <div className="contactproperty">  
 <img src={image1} className="propertyContainerPropertyImage" />  
 <h2>Message our leasing office</h2>  
 <input type="text" name="firstName" placeholder="First Name\*" onChange={handleChange} />  
 <input type="text" name="lastName" placeholder="Last Name\*" onChange={handleChange} />  
 <input type="text" name="email" placeholder="Email\*" onChange={handleChange} />  
 <input type="text" name="phone" placeholder="Phone(optional)" onChange={handleChange} />  
 <textarea type="text" name="message" placeholder="What can we help you with?" onChange={handleChange} />  
 <button type="button" onClick={handleSubmit}>Send Message</button>  
 </div>  
 </div>  
 <Footer />  
 </>  
 )  
}  
  
export default Property;

.searchContainerPropertySearch {  
 background: #ffffff;  
 height: 200px;  
 display: flex;  
 align-items: center;  
 justify-content: space-evenly;  
}  
  
.searchContainerPropertySearch select {  
 /\* margin-left: 80px; \*/  
 padding: 15px 10px;  
 background:#ffffff;  
 border-radius: 5px;  
 border-color: 1px black;  
 font-size: 18px;  
 text-align: center;  
}  
  
.datepicker {  
 /\* margin-left: 80px; \*/  
 padding: 15px 10px;  
 text-align: center;  
 font-size: 12px;  
}  
  
.searchContainerPropertySearch option {  
 text-align: center;  
}  
  
.searchedResultsContainer {  
 background: #ffffff;  
 height: 400px;  
 display: flex;  
 align-items: center;  
 justify-content: space-around;  
 flex-wrap: wrap;  
 margin: 20px;  
}  
  
.searchedResultsImage {  
 background-size: cover;  
 height: 200px;  
 cursor: pointer;  
}  
  
.singleresultContent {  
 margin: 20px 20px;  
 padding: 20px;  
 background: #C0C0C0;  
}  
  
.searchedResultsTitle {  
 font-size: 22px;  
 font-weight: 800;  
}  
  
.searchedResultsPrice {  
 /\* color:#ffffff; \*/  
}  
  
.submitButton {  
 padding: 15px 20px;  
 /\* margin-left: 50px; \*/  
 background: #00adef;  
 border-radius: 5px;  
 border: none;  
 font-size: 12px;  
 /\* margin-right: 150px; \*/  
 color: #ffffff;  
 cursor: pointer;  
}  
  
.areaStatusLabel {  
 height: 40px;  
 width: 80px;  
 background: red;  
 border-radius: 10px;  
 text-align: center;  
 padding-top: 7px;  
 box-sizing: border-box;  
 color: #ffffff;  
 font-weight: 700;  
}

import React, { useEffect, useState } from 'react';  
import HomeSearch from '../../components/homeSearch/HomeSearch';  
import TopBar from '../../components/topbar/TopBar';  
import './propertysearch.css';  
import "react-datepicker/dist/react-datepicker.css";  
import DatePicker from 'react-datepicker';  
import image1 from '../../images/room1.jpg';  
import Footer from '../../components/footer/Footer';  
import { useHistory, useLocation } from "react-router-dom";  
import axios from 'axios';  
  
const PropertySearch = () => {  
 const [startDate, setStartDate] = useState(new Date());  
 let location = useLocation();  
 let history = useHistory();  
 let [searchedData, setSearchedData] = useState({});  
 let [loading, setLoading] = useState(true)  
 const [areaStreetData, setAreaStreetData] = useState({})  
 const [loadingArea, setLoadingArea] = useState(true)  
 const [homeSearchData, setHomeSearchData] = useState({ date: new Date()})  
  
 let GetAreaAllDataFn = () => {  
 let config = {  
 headers: {  
 authorization: `Bearer ${localStorage.getItem('token')}`  
 }  
 }  
 axios.get('http://localhost:8000/area/viewAllData', config).then(  
 data => {  
 console.log(data, 'area view all data')  
 setAreaStreetData(data.data)  
 setLoadingArea(false)  
 }  
 )  
 }  
  
 useEffect(() => {  
 console.log(location.state, 'location state')  
 setSearchedData(location.state)  
 setLoading(false)  
 GetAreaAllDataFn();  
 }, [])  
  
 let handlesubmit = () => {  
 console.log(homeSearchData)  
 axios.post('http://localhost:8000/property/findFilteredProperties', homeSearchData).then(  
 data => {  
 console.log(data)  
 setSearchedData(data.data)  
 // history.push('/propertysearch', data.data)  
 },  
 err => console.log(err)  
 )  
 }  
  
 let handleChange = (v, e) => {  
 // setAreaStreetData(prevState => ({  
 // ...prevState,  
 // [e.target.name] : e.target.value  
 // }))  
 console.log(e)  
 // console.log(e)  
 if (v === "homesearch") {  
 setHomeSearchData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 }  
 if (v === "date") {  
 setHomeSearchData(prevState => ({  
 ...prevState,  
 date : e  
 }))  
 }  
 console.log(homeSearchData)  
 }  
  
 return (  
 <>  
 <TopBar />  
 <div className="searchContainerPropertySearch">  
 { (loadingArea) ? (  
 ''  
 ) : (  
 (areaStreetData) ? (  
 <select name="street" onChange={(e) => handleChange('homesearch', e)} type="text">  
 { areaStreetData.map((v,i) => {  
 return <option key={i} value={v.name}>{v.name}</option>  
 }) }  
 </select>   
 ) : (  
 ''  
 )  
 )}  
 <select name="bedrooms" type="text" onChange={(e) => handleChange('homesearch', e)}>  
 <option value="--">Bedrooms</option>  
 <option value="0">Studio</option>  
 <option value="1">1+</option>  
 <option value="2">2+</option>  
 <option value="3">3+</option>  
 </select>  
 <select name="price" type="text" onChange={(e) => handleChange('homesearch', e)}>  
 <option value="--">Price</option>  
 <option value="600">Less than $600</option>  
 <option value="1200">Less than $1200</option>  
 <option value="2000">Less than $2000</option>  
 <option value="3000">Less than $3000</option>  
 </select>  
 <div>  
 <DatePicker name="date" className="datepicker" selected={startDate} onChange={(date) => setStartDate(date)} />  
 </div>  
 <button type="button" className="submitButton" onClick={handlesubmit}>Submit</button>  
 </div>  
 <div className="searchedResultsContainer">  
 { (loading) ? (  
 ''  
 ) : (  
 (searchedData) ? (  
 searchedData.map((v, i) => {  
 return <div className="singleresultContent">  
 <img className="searchedResultsImage" src={image1} onClick={() => history.push(`/property/${v.\_id}`)} />  
 <h4 className="searchedResultsTitle">{v.name}</h4>  
 <p className="searchedResultsPrice">$ {v.price}</p>  
 <p className="searchedResultsSft">{v.propertyArea} sqft</p>  
 <p className="areaStatusLabel">{v.areaStatus}</p>  
 <p className="searchedResultsSft">Occupied: {(v.occupied == "true") ? 'Yes occupied' : 'Not occupied'}</p>  
 </div>  
 })  
 ) : (  
 <h4>No Data Found</h4>  
 )  
 )  
 }  
 </div>  
 {/\* <Footer /> \*/}  
 </>  
 )  
}  
  
export default PropertySearch;

.register {  
 height: 780px;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 align-items: center;  
 background: linear-gradient(  
 rgba(255, 255, 255, 0.5),  
 rgba(255, 255, 255, 0.5)  
 ),   
 url('../../images/register.jpg');  
 background-size: cover;  
}  
  
.registerTitle {  
 font-size: 50px;  
 font-weight: bold;  
 color: white;  
}  
  
.registerForm {  
 margin-top: 20px;  
 display: flex;  
 flex-direction: column;  
}  
  
.registerForm label {  
 margin: 10px 0;  
 font-weight: 700;  
}  
  
.registerInput {  
 padding: 10px;  
 background-color: white;  
 border: none;  
 border-radius: 5px;  
}  
  
.registerButton {  
 margin-top: 20px;  
 cursor: pointer;  
 background-color: black;  
 border: none;  
 color: white;  
 border-radius: 10px;  
 padding: 10px;  
 font-weight: 700;  
}  
  
.registerLoginButton {  
 position: absolute;  
 top: 60px;  
 right: 20px;  
 background-color: lightcoral;  
 cursor: pointer;  
 border: none;  
 padding: 10px;  
 color: white;  
 border-radius: 10px;  
}

import './register.css';  
import 'react-notifications/lib/notifications.css';  
import {NotificationContainer, NotificationManager} from 'react-notifications';  
import { useState } from 'react';  
import { useHistory } from "react-router-dom";  
// import { registerUser } from '../../redux/actions/registerAction';  
// import { useDispatch, useSelector } from 'react-redux'  
import { useEffect } from 'react';  
import axios from 'axios';  
  
export default function Register() {  
 let history = useHistory();  
 // const dispatch = useDispatch()  
 const [formData, setFormData] = useState([]);  
 // const counter = useSelector((state) => console.log(state))  
 const handleClick = (arg) => {  
 history.push(`/${arg}`)  
 }  
 const handleChange = (e) => {  
 e.preventDefault();  
 console.log(e.target.value,'e')  
 console.log(e.target.name, 'name')  
 setFormData(prevState => ({  
 ...prevState,  
 [e.target.name] : e.target.value  
 }))  
 // console.log(formData, 'formData in change')  
 }  
 const handleSubmit = () => {  
 console.log(formData, 'formData')  
 // dispatch(registerUser(formData));  
 axios.post('http://localhost:8000/user/create', formData).then(  
 data => {  
 console.log(data, 'data in register')  
 NotificationManager.success('Registered successfully');  
 setTimeout(() => {  
 history.push('/login')  
 }, 2000)  
 },  
 err => {  
 console.log(err)  
 }  
 )  
 }  
  
 useEffect(() => {  
 // console.log(counter, 'useEffect')  
 }, [])  
 return (  
 <div className="register">  
 <pan className="registerTitle">Register</pan>  
 <div className="registerForm">  
 <label>Username</label>  
 <input className="registerInput" name="username" type="text" placeholder="Enter your username...." onChange={handleChange} />  
 <label>Email</label>  
 <input className="registerInput" name="email" type="text" placeholder="Enter your email...." onChange={handleChange}/>  
 <label>Password</label>  
 <input className="registerInput" name="password" type="password" placeholder="Enter your password...." onChange={handleChange} />  
 <button className="registerButton" onClick={() => handleSubmit() }>Submit</button>  
 </div>  
 {/\* <button className="registerLoginButton" onClick={() => handleClick('login')}>Login</button> \*/}  
 <NotificationContainer/>  
 </div>  
 )  
}