import React, { Component } from "react";

import axios from 'axios';

// const BASE\_URL = 'http://localhost:3000/';

class Product extends Component {

constructor(props) {

super(props);

this.state = {

name:"",

price:"",

modelno:"",

warranty:"",

file: null,

productImage: '',

imagePath: '',

images: [],

imageUrls: [],

}

// this.handleOnChange = this.handleOnChange.bind(this);

// this.clickedButton = this.clickedButton.bind(this)

//StackOverflow

this.onChange = this.onChange.bind(this);

this.onSubmit = this.onSubmit.bind(this);

}

onChange = (event) => {

switch (event.target.name) {

case 'productImage':

this.setState({ productImage: event.target.files[0]});

// this.state.productImage.map(name => {

// return (console.log(name, "Name"))

// })

let images = []

for (var i = 0; i < event.target.files.length; i++) {

images[i] = event.target.files.item(i);

}

images = images.filter(image => image.name.match(/\.(jpg|jpeg|png|gif)$/))

this.setState({imagePath : images.filter(image => {this.setState({imagePath: image.name})})

})

let message = `${images.length} valid image(s) selected`

this.setState({ images, message })

// this.state.images.map(name => {

// return (console.log(name, "Name"))

// })

break;

default:

this.setState({ [event.target.name]: event.target.value });

}

}

onSubmit = (e) => {

e.preventDefault();

const { name, price, modelno, warranty, productImage, imagePath } = this.state;

let formData = new FormData();

this.state.images.map(image => {

this.setState({ImageUrls: image.name})

console.log(image.name, "Name")

})

formData.append('name', name);

formData.append('price', price);

formData.append('modelno', modelno);

formData.append('warranty', warranty);

formData.append('productImage', productImage);

formData.append('imageName', imagePath);

console.log(...formData)

axios.post('http://localhost:3000/api/product/add', formData)

.then((result) => {

console.log('>> (onSubmit) file upload result = ',result);

// access results...

})

.then((result) => {

this.setState({imageUrls: [ result.data.imageUrls]})

console.log(result.data.imageUrls)

})

.catch(function (error) {

console.log(error);

})

// axios.all(uploaders).then(() => {

// console.log('done');

// }).catch(err => alert(err.message));

}

componentWillMount (){

// console.log(this.props.location.state.referrer);

}

handleOnChange (data) {

console.log('entered');

switch(data.target.id){

case "Pname":

this.setState({

name: data.target.value

})

break

case "Price":

this.setState({

price: data.target.value

})

break

case "Mnumber":

this.setState({

modelno: data.target.value

})

break

case "Warrenty":

this.setState({

warranty: data.target.value

})

break

case "Ufile":

this.setState({

file: data.target.files[0]

});

console.log()

break

default:

break

}

}

// topClicked = () => {

// return console.log('hi')

// }

// clickedButton = () => {

// const fd =new FormData();

// fd.append('name', this.state.name);

// fd.append('price', this.state.price);

// fd.append('modelno', this.state.modelno);

// fd.append('warranty', this.state.warrenty);

// fd.append('productImage', this.state.file);

// // let newPost = {

// // name: this.state.name,

// // price: this.state.price,

// // modelno: this.state.modelno,

// // warranty: this.state.warranty,

// // }

// console.log(this.state.warrenty)

// fetch(`http://localhost:3000/api/product/add`, {

// method: 'post',

// body:fd

// })

// .then (function(response){

// return response.json()

// })

// .then(function (data){

// console.log('post request response data', data)

// })

// }

render() {

const { name, price, modelno, warranty } = this.state;

return (

<div>

<form onSubmit={this.onSubmit}>

<input

type="text"

name="name"

value={name}

onChange={this.onChange}

/>

<input

type="text"

name="price"

value={price}

onChange={this.onChange}

/>

<input

type="text"

name="modelno"

value={modelno}

onChange={this.onChange}

/>

<input

type="text"

name="warranty"

value={warranty}

onChange={this.onChange}

/>

<input

type="file"

name="productImage"

onChange={this.onChange}

/>

<button type="submit">Submit</button>

</form>

</div>

);

}

}

export default Product;

import React, { Component } from "react";

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

// const BASE\_URL = 'http://localhost:3000/'

class List extends Component {

constructor(props) {

super(props);

this.state = {

productData : [],

pass: true,

orderData : [],

subOrderData: [],

editId : "",

path: "",

productImage: '',

imagePath: '',

}

this.handelDeleteProducts = this.handelDeleteProducts.bind(this);

this.handelDeleteOrders = this.handelDeleteOrders.bind(this);

this.handelEdit = this.handelEdit.bind(this);

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Data Will Be Loaded After Page Loads\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

componentWillMount(){

fetch('http://localhost:3000/api/product')

.then(res => res.json()

.then(res.data))

.then(product1 => this.setState({productData: product1}));

fetch('http://localhost:3000/api/order')

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

.then(order1 => this.setState({orderData: order1}))

console.log(this.state.path)

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*After Button Clicked Data Will Be Updated\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

componentDidUpdate(){

fetch('http://localhost:3000/api/product')

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

.then(product1 => this.setState({productData: product1}));

fetch('http://localhost:3000/api/order')

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

.then(order1 => this.setState({orderData: order1}))

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Delete Button Clicked\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

handelDeleteProducts = (data) => {

console.log('Deleted Successfully')

fetch("http://localhost:3000/api/product/delete/" + data.target.id, {

method: 'delete',

headers: {'Content-Type': 'application/json'}

})

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

.then(res => console.log(res))

.catch(err => console.log(err))

}

handelDeleteOrders = (data) => {

console.log('Deleted Successfully')

console.log(data.target.id)

fetch("http://localhost:3000/api/order/delete/" + data.target.id, {

method: 'delete',

headers: {'Content-Type': 'application/json'}

})

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

.then(res => console.log(res))

.catch(err => console.log(err) )

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Edit Button Clicked\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

checkimage = () =>{

}

handelEdit(data){

if(this.state.pass=== true)

{

console.log('entered');

return(

<Redirect to={{

pathname: '/',

state: { referrer: data.target.id }

}} />)

}

// console.log(data.target.id);

// fetch('http://localhost:3000/api/product/edit/' + data.target.id, {

// method: 'get',

// headers: {'Content-Type': 'application/json'}

// })

// .then(res => res.json({msg: "Edit Page"}))

// .then(product => this.setState({productEditData: product}));

// console.log(this.props.productEditData);

}

render() {

const productItems = this.state.productData.map(productData =>(

<div style={{textAlign:'left'}} >

<p style={{fontSize: '17px'}}>Name: {productData.name}</p>

<p style={{fontSize: '17px'}}>Price: {productData.price}</p>

<p style={{fontSize: '17px'}}>Model Number: {productData.modelno}</p>

<p style={{fontSize: '17px'}}>Warranty: {productData.warranty}</p>

<p style={{fontSize: '17px'}}>Entry Date: {productData.entrydate}</p>

<br />

<img src={ JSON.stringify(productData.productImage).replace(/"client\\/g, "..").replace(/src\\/g, "").replace(/"/g, "")} alt="new" width="100px" height="100px"/>

<input type="button" value="Edit" id={productData.\_id} onClick={this.handelEdit}>

</input>

<input type="button" value="Delete" id={productData.\_id} onClick={this.handelDeleteProducts} />

</div>

));

const orderItems = this.state.orderData.map(orderData1 =>(

List.js

<div style={{textAlign:'left'}} id={orderData1.\_id}>

<p style={{fontSize: '17px'}}>Customer Name: {orderData1.customername}</p>

<p style={{fontSize: '17px'}}>Quantity: {orderData1.quantity}</p>

<p style={{fontSize: '17px'}}>Phone Number: {orderData1.phoneno}</p>

<p style={{fontSize: '17px'}}>Product Details:

<br/> a) Name:

{Object.entries(orderData1.productdetail)[1].slice(1).map(function(item, i){return(<p>{item}</p>)})}

<br/> b)Price:

{Object.entries(orderData1.productdetail)[2].slice(1).map(function(item, i){return(<p>{item}</p>)})}

</p>

<br />

<input type="button" value="Delete" id={orderData1.\_id} onClick={this.handelDeleteOrders} />

</div>

));

return (

<div className="container" style={{marginTop: "5%"}}>

<h3 className="text-center">List</h3>

<hr/>

<div className="row">

<div className="col-6">

<h4 style={{fontWeight: "bold"}}>Product Details</h4>

{productItems}

</div>

<div className="col-6">

<h4 style={{fontWeight: "bold"}}>Order Details</h4>

{orderItems}

</div>

</div>

</div>

);

}

}

export default List;

const express = require("express");

const router = express.Router();

const mongoose = require('mongoose');

var multer = require('multer');

const path = require("path");

//const upload = multer({dest: "uploads/"});

const Product = require("../../model/Product");

//File Upload

// var storage = multer.diskStorage({

// destination: function (req, file, cb) {

// // cb(null, "./uploads/");

// cb(null, 'client/src/component/uploads/products')

// },

// filename: function(req, file, cb) {

// cb(null, new Date().toISOString() + file.originalname);

// }

// });

// const fileFilter = (req, file, cb) => {

// // reject a file

// if (file.mimetype === 'image/jpeg' || file.mimetype === 'image/png') {

// cb(null, true);

// } else {

// cb(null, false);

// }

// };

// const upload = multer({

// storage: storage,

// limits: {

// fileSize: 1024 \* 1024 \* 5

// },

// fileFilter: fileFilter

// });

var storage = multer.diskStorage({

destination: (req, file, cb) => {

cb(null, 'client/public/images/uploads')

},

filename: (req, file, cb) => {

cb(null, Date.now() + '-' + file.originalname)

}

});

Product.js api/Routes

const upload = multer({ storage })

//Create Product

router.post("/add",upload.single('productImage'), (req, res) => {

console.log(req.file)

const newProduct = new Product({

\_id: new mongoose.Types.ObjectId(),

name: req.body.name,

price: req.body.price,

modelno: req.body.modelno,

warranty: req.body.warranty,

productImage: req.file.path,

imageName: req.body.imageName

});

newProduct.save().then(product => res.json(product));

});

//Read all the Product data

router.get("/", (req, res) => {

Product.find()

.sort({ date: 1 })

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

});

//Read a particular product data using id to edit

router.get("/edit/:id", (req, res) => {

Product.findById(req.params.id)

.exec()

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

.catch(err => res.status(404).json({ msg: "Product Not Found" }));

});

//Update product using their ids

router.post("/update/:id", (req, res) => {

Product.findByIdAndUpdate(req.params.id, req.body)

.then(product => {

res.status(200).json({ msg: "Product Updated Successfully" });

})

.catch(product => {

res.status(400).json({ msg: "Error in updating Product" });

});

});

//Delete Product Data using their ids

router.delete("/delete/:id", (req, res) => {

Product.findById(req.params.id)

.then(product =>

product

.remove()

.then(product =>

res.status(200).json({ msg: "Product Deleted Successfully" })

)

)

.catch(product =>

res.status(400).json({ msg: "Error in deleting Product" })

);

});

module.exports = router;

Product.js Model

const mongoose = require('mongoose');

const productSchema = mongoose.Schema({

\_id: mongoose.Schema.Types.ObjectId,

name: { type: String },

price: { type: String },

modelno: { type: String },

warranty: { type: String },

productImage: { type: String },

imageName: {type: String},

date: {

type: Date,

default: Date.now

}

});

module.exports = mongoose.model('Product', productSchema);

Server.js

const express= require("express");

const mongoose = require("mongoose");

const bodyParser = require("body-parser");

const app = express();

const cors = require("cors");

const db = require("./database/keys").mongoURI;

const product = require("./routes/api/product");

const order = require("./routes/api/order");

// Bodyparser Middleware

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

// Connect to Mongo

mongoose

.connect(db, { useCreateIndex: true, useNewUrlParser: true }) // Adding new mongo url parser

.then(() => console.log("MongoDB Connected..."))

.catch(err => console.log(err));

mongoose.set("useFindAndModify", false);

// Use Routes

app.use(cors());

app.use("/api/product", product);

app.use("/api/order", order);

const port = process.env.PORT || 8080;

app.listen(port, () => console.log(`Server started on port ${port}`));