Advanced Programming

Lab-09

1. Muneeb Hashmi

#220042

BSCS-7A

**Server.js**

const path = require('path');

const express = require('express');

const app = express();

require("./models/db");

var Product = require('./models/product.model')

var bodyParser = require("body-parser")

var multer = require('multer')

var upload = multer()

app.use(bodyParser.json())

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

app.use(upload.array())

var data;

var fs = require('fs');

var cart = [];

var price = 0;

var productitems;

Product.find(function(err, resp)

{

if (err) throw err;

console.log(resp)

console.log(resp.length)

productitems = resp

console.log(productitems)

})

const cors = require('cors');

app.use(cors())

app.get('/api/products', (req, res)=>{

Product.find(function(err, response)

{

if (err) throw err;

console.log(response)

console.log(response.length)

productitems = response

data = [productitems, cart, {price:price}]

res.send(data)

console.log(data)

})

})

app.get('/api/products/cart/:name',(req, res)=>

{

var pname= req.params.name;

var quantity=0;

console.log(pname);

Product.findOne({name:pname},function(err, response)

{

if (err) throw err;

console.log(response)

console.log(response.length)

price += response.price;

console.log(price)

var getitem = cart.find((item, i)=>{

if (item.name === pname)

{

quantity =cart[i].quantity +1;

cart[i] = {name:pname, quantity:quantity};

}

})

if (quantity == 0)

{

quantity +=1

var it = {

name: pname,

quantity:quantity,

}

cart.push(it)

}

data = [productitems, cart, {price:price}]

res.send(data)

})

})

app.get('/api/products/cart/delete/:name',(req, res)=>

{

var pname= req.params.name;

console.log(pname);

var itemp;

Product.findOne({name:pname}, function(err, response)

{

if (err) throw err;

console.log(response)

console.log(response.length)

itemp = response.price;

price = price - itemp;

console.log(price)

var newarray;

var index;

var getitem = cart.find((item, i)=>{

if (item.name === pname)

{

console.log(item.name)

if (item.quantity == 1 )

{

index=i;

cart.splice(index, 1)

}

else

{

var quantity =cart[i].quantity -1;

cart[i] = {name:pname, quantity:quantity};

}

}

})

data = [productitems, cart, {price:price}]

res.send(data)

})

})

app.get('/api/products/delete/:name', (req, res)=>{

var pname = req.params.name;

console.log(pname)

Product.deleteOne({name:pname}, function(err, response)

{

if (err) throw err;

console.log(response.deletedCount)

res.redirect("/api/products")

})

})

app.get('/api/products/asc', (req, res)=>{

Product.find().sort({price:1}).exec(function(err, response)

{

if (err) throw err;

console.log(response)

console.log(response.length)

productitems = response

data = [productitems, cart, {price:price}]

res.send(data)

console.log(data)

})

})

app.get('/api/products/dsc', (req, res)=>{

Product.find().sort({price:-1}).exec(function(err, response)

{

if (err) throw err;

console.log(response)

console.log(response.length)

productitems = response

data = [productitems, cart, {price:price}]

res.send(data)

console.log(data)

})

})

app.post('/api/products/create', function(req, res){

console.log("helloo");

var name = req.body.name;

var category = req.body.category;

var image= req.body.image;

var desc = req.body.desc

var price= parseFloat(req.body.price);

var newProduct = {

name:name,

category:category,

price:price,

image:image,

desc:desc

}

var product = new Product(newProduct)

product.save(function(err, Product)

{

if (err) throw err;

console.log("product saved in database")

console.log(Product)

})

console.log("added")

res.redirect('/api/products')

})

const port = 5000;

app.listen(port, ()=>console.log('server started on ${port}'))

**App.js**

import React, {Component} from 'react';

import './App.css';

import Product from './components/products/Product';

import Add from './components/products/Add'

class App extends Component {

constructor()

{

super()

}

render()

{

return (

<div className="App">

<Product />

<Add />

</div>

);

}

}

export default App;

**Product Component (product.js)**

import React, {Component} from 'react';

import M from 'materialize-css';

//import 'materialize-css/dist/css/materialize.min.css';

//import 'react-materialize';

const axios = require('axios').default;

class Product extends Component {

constructor()

{

super()

this.state ={

products:[],

cart:[],

price:0

}

this.handleChangeA = this.handleChangeA.bind(this);

this.handleChangeD = this.handleChangeD.bind(this);

this.handleDelete = this.handleDelete.bind(this);

this.handleCart = this.handleCart.bind(this);

this.handleDeleteItem = this.handleDeleteItem.bind(this);

}

handleDeleteItem(name)

{

console.log("delete"+name)

axios.get('/api/products/cart/delete/'+name).then(response=>

this.setState({products:response.data[0], cart:response.data[1], price:response.data[2].price},()=>console.log("products fetched", response.data))

)

}

handleCart(name)

{

console.log(name)

axios.get('/api/products/cart/'+name).then(response=>

this.setState({products:response.data[0], cart:response.data[1], price:response.data[2].price},()=>console.log("products fetched", response.data))

)

}

handleChangeA(event)

{

axios.get('/api/products/asc').then(response=>

this.setState({products:response.data[0], cart:response.data[1], price:response.data[2].price},()=>console.log("products fetched", response.data))

)

}

handleDelete(name)

{

console.log(name)

axios.get('/api/products/delete/'+name).then(response=>

this.setState({products:response.data[0], cart:response.data[1], price:response.data[2].price},()=>console.log("products fetched", response.data))

)

}

handleChangeD(event)

{

axios.get('/api/products/dsc').then(response=>

this.setState({products:response.data[0], cart:response.data[1], price:response.data[2].price},()=>console.log("products fetched", response.data))

)

}

componentDidMount()

{

M.AutoInit()

axios.get('/api/products').then(response=>

this.setState({products:response.data[0], cart:response.data[1], price:response.data[2].price},()=>console.log("products fetched", response.data))

)

}

render()

{

return (

<div>

<h1>Products</h1>

<button className="btn waves-effect waves-light" onClick={this.handleChangeA}>Ascending Order

</button><br /><br />

<button className="btn waves-effect waves-light" onClick={this.handleChangeD}>Descending Order

</button>

<div>

<div className="col s4 m4">

<h2 className="header">Your Cart</h2>

<div className="card horizontal">

<div className="card-image">

<img src="https://lorempixel.com/100/190/nature/6" />

</div>

<div className="card-stacked">

{ this.state.cart.map(item=>

<div>

<div className="card-content">

Name: {item.name} Quantity : {item.quantity} <a class="btn-floating waves-effect waves-light red" onClick={()=>this.handleDeleteItem(item.name)}><i class="material-icons">delete</i></a>

</div>

</div>

)}

<div className="card-content">

Total Price : {this.state.price}

</div>

</div>

</div>

</div>

</div>

<div className="row">

{

this.state.products.map(product =>

<div className="col s12 m4">

<div className="card">

<div className="card-image">

<img src={product.image} height="200px" />

<span className="card-title blue-text text-darken-2">{product.name}</span>

<span style={{marginLeft: "250px"}}>Delete item</span><a class="btn-floating halfway-fab waves-effect waves-light red" onClick={()=>this.handleDelete(product.name)}><i class="material-icons">delete</i></a>

</div>

<div className="card-content">

{product.desc}

</div>

<div className="card-action">

Add to cart<a style={{marginLeft: "20px"}} class="btn-floating waves-effect waves-light red" onClick={()=>this.handleCart(product.name)}><i class="material-icons">add</i></a>

</div>

<div className="card-action">

Price ${product.price}

</div>

</div>

</div>

)

}

</div></div>

);

}

}

export default Product;

**Add Component (add.js)**

import React, {Component} from 'react';

import M from 'materialize-css';

class Add extends Component {

constructor()

{

super()

this.state ={

products:[],

isHidden:false,

}

this.buttonValue="Add Items"

this.handleChange = this.handleChange.bind(this);

}

handleChange(){

this.setState({

isHidden: !(this.state.isHidden)

})

if (this.buttonValue=="Add Items")

{

this.buttonValue = "Cancel";

}

else

{

this.buttonValue="Add Items"

}

}

componentDidMount()

{

M.AutoInit();

}

render()

{

return (

<div>

<button className="btn waves-effect waves-light" onClick={()=>this.handleChange()}>{this.buttonValue}</button>

{this.state.isHidden && <div><h1>Add product here</h1>

<div className="row">

<form action='/api/products/create' method="post" className="col s12">

<div className="row">

<div className="col s6">

<label for="name">Product Name</label>

<input type="text" name="name" placeholder="Enter product name here" />

</div>

<div className="col s6">

<label for="category">Product Category</label>

<input type="text" name="category" placeholder="Enter product id here" />

</div>

</div>

<div className="row">

<div className="col s6">

<label for="image">Image Url</label>

<input type="text" name="image" placeholder="Enter Image Url" />

</div>

<div className="col s6">

<label for="price">Price</label>

<input type="number" name="price" placeholder="Enter price of the product" step="0.01" min="0" />

</div>

</div>

<div className="row">

<div className="col s12">

<label for="desc">Description</label>

<textarea name="desc" className="materialize-textarea"></textarea><br /><br />

</div>

</div>

<button type="submit" className="btn waves-effect waves-light">Done</button>

</form>< /div>

< /div>}

</div>

);

}

}

export default Add;

**Product Model (product.model.js)**

const mongoose = require('mongoose');

mongoose.set('useCreateIndex', true);

mongoose.set('useNewUrlParser', true);

mongoose.set('useUnifiedTopology', true);

var productSchema = mongoose.Schema(

{

name:

{

type:String,

required:true,

},

category:

{

type:String,

required:true,

},

desc:

{

type:String,

},

price:

{

type:Number,

required:true

},

image:

{

type:String,

}

})

var Product = mongoose.model("Product",productSchema);

module.exports = Product;

**Database (db.js)**

const mongoose = require('mongoose');

mongoose.set('useNewUrlParser', true);

mongoose.set('useUnifiedTopology', true);

mongoose.connect('mongodb://localhost/productdb', function(err)

{

if (err) throw err;

console.log("Mongodb Database my db connected")

});

var Product = require('./product.model')