# Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7A**

# Lab 09: React JS

**Date: 14 November, 2019**

**Time: 10:00-01:00pm & 02:00-05:00pm**

**Hafiz Muhammad Aafaq (212419)**

# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

# 

# Lab 09: React JS

**Lab Tasks**

Build a simple shopping cart prototype that shows how React components can be used to build a friendly user experience with instant visual updates and scalable code in ecommerce applications.

**Features**

* Add and remove products from the floating cart
* Sort products by highest to lowest and lowest to highest price

**React Code:**

**App.js:**

import React, { useState, useEffect,Component } from "react";

import Cart from './Components/AddToCart/Cart';

// SERVICES

import productService from './services/productService';

import Card from 'react-bootstrap/Card';

import Button from 'react-bootstrap/Button';

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

import Header from '../src/Components/Header/header';

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

class App extends Component {

state = {

products: this.products,

Cart: []

}

addCart = [];

componentDidMount(){

this.getProducts();

}

getProducts = async () => {

let res = await productService.getAll();

console.log(res);

this.setState({

products:res

})

}

Compare = (a,b) =>{

if ( a.name < b.name ){

return -1;

}

if ( a.name > b.name ){ // Sorting Algorithm

return 1;

}

return 0;

}

addToCart = (...product)=>{

this.state.Cart.sort(this.Compare); //Sort

this.setState(prevState => ({

Cart: [...prevState.Cart, ...product] //Add to Cart

}))

// console.log(this.state.Cart);

}

renderProduct = product => {

return (

<div key={product.\_id} className="row" >

<Card style={{ width: '18rem', margin:'10px'}}>

<Card.Img variant="top" src={product.image}/>

<Card.Body>

<Card.Title>{product.name}</Card.Title>

<Card.Text>

{product.description}

</Card.Text>

<Button variant="success" style={{ margin:'10px'}}>{product.price}</Button>

<Button variant="primary" onClick={()=>this.addToCart(product)}>Add to Cart</Button>

</Card.Body>

</Card>

</div>

);

};

deleteCart = (id) =>{

this.setState(prevState => {

const cart = this.state.cart.filter((item,j) => id !== j);

return{

cart,

}

})

}

render(){

return (

<div className="App">

<Header/>

<Container>

<Row>

{(this.state.products && this.state.products.length > 0) ? (

this.state.products.map(product => this.renderProduct(product))

) : (

<p>No products found</p>

)}

</Row>

<Cart products={this.state.Cart} delete={(id)=>this.deleteCart(id)}/>

</Container>

</div>

);

}

}

export default App;

**Route.js:**

import React from 'react';

import { Component } from 'react';

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

import App from './App';

import Cart from './Components/AddToCart/Cart';

class Routes extends Component{

render(){

return(

<Switch>

<Route path="/" exact component={App}/>

<Route path="/cart" exact component={Cart}/>

</Switch>

)

}

}

export default Routes;

**Index.js:**

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import \* as serviceWorker from './serviceWorker';

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

import Routes from './routes';

const App = () =>{

return(

<BrowserRouter>

<Routes/>

</BrowserRouter>

)

}

ReactDOM.render(<App />, document.getElementById('root'));

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

serviceWorker.unregister();

**Cart.js:**

import React, { Component } from 'react';

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

import styles from './style.module.css';

import productService from '../../services/productService';

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

import FontAwesome from 'react-fontawesome';

class Cart extends Component{

render(){

return(

<div>

<h1 className={styles.cartHeading}>Cart</h1>

<Container>

<Row>

<Col><h3 className={styles.cartHeading}>Product Name</h3></Col>

<Col><h3 className={styles.cartHeading}>Description</h3></Col>

<Col><h3 className={styles.cartHeading}>Price</h3></Col>

<Col><h3 className={styles.cartHeading}>Remove</h3></Col>

</Row>

{this.props.products.map(item =>(

<Row key={item.\_id}>

<Col><h5 className={styles.cartHeading}>{item.name}</h5></Col>

<Col><h5 className={styles.cartHeading}>{item.description}</h5></Col>

<Col><h5 className={styles.cartHeading}>{item.price}</h5></Col>

<Col><FontAwesome

name="trash"

onClick={this.props.delete(item.name)}

style={{color:'#000', paddingLeft:'100px',cursor:'pointer'}}

/></Col>

</Row>

))}

</Container>

</div>

)

}

}

export default Cart;

**Express Code:**

**Index.js:**

const express = require('express');

const mongoose = require('mongoose');

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

require('./models/Product');

const app = express();

mongoose.Promise = global.Promise;

mongoose.connect(process.env.MONGODB\_URI || `mongodb://localhost:27017/node-react-app`);

app.use(bodyParser.json());

require('./routes/productRoute')(app);

if (process.env.NODE\_ENV === 'production') {

app.use(express.static('client/build'));

const path = require('path');

app.get('\*', (req,res) => {

res.sendFile(path.resolve(\_\_dirname, 'client', 'build', 'index.html'))

})

}

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

console.log('app running on port ${PORT}');

});

**Product.js:**

const mongoose = require('mongoose');

const {Schema} = mongoose;

const productSchema = new Schema({

name: String,

description: String,

})

mongoose.model('products', productSchema);

`

**Productservice.js:**

const mongoose = require('mongoose');

const Product = mongoose.model('products');

module.exports = (app) => {

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

let products = await Product.find();

return res.status(200).send(products);

});

app.post('/api/product', async (req, res) => {

let product = await Product.create(req.body);

return res.status(201).send({

error: false,

product

})

})

app.put('/api/product/:id', async (req, res) => {

const {id} = req.params;

let product = await Product.findByIdAndUpdate(id, req.body);

return res.status(202).send({

error: false,

product

})

});

app.delete('/api/product/:id', async (req, res) => {

const {id} = req.params;

let product = await Product.findByIdAndDelete(id);

return res.status(202).send({

error: false,

product

})

})

}

**Explanation:**

We need to add the data in the database manually. Then copy this code in separate files and use npm install to install this code and npm start to run this code and check this site.