# Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7AB**

# Lab 09: React JS

**Date: 14 November, 2019**

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

**Aamna Sarosh (218953)**

# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

# 

# Lab 09: React JS

**Introduction**

**React.js** allows us to express how our app should look at any given point in time. React will automatically manage all UI updates when you’re underlying data changes.

**Objectives**

This lab will get students familiar with the ReactJS by creating a web application.

**Tools/Software Requirement**

ReactJs, ExpressJs

**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

**Hint:**

You can use React.js as the front-end framework, and a backend server built using Node.js and Express.js.

|  |
| --- |
| **Solution** |
| **Task Code:**  **var express = require('express')**  var app = express()  var bodyParser = require('body-parser')  var mongoose = require('mongoose');  mongoose.connect('mongodb://localhost/product-manager');  var userSchema = mongoose.Schema({ name: String, email: String, password: String, role: String });  var Users = mongoose.model("Users", userSchema)  var productSchema = mongoose.Schema({ name: String, category: String, price: Number, sellerId: String });  var products = mongoose.model("products", productSchema)  app.use(bodyParser.json())  app.use(bodyParser.urlencoded({ extended: true }))  app.set('view engine', 'pug')  app.set('views', './views')  app.get('/', function (req, res) {  res.render('Log in')  })  app.post('/', function (req, res) {  var userinfo = req.body  console.log(userinfo)  if (userinfo.password != userinfo.cpassword)  res.render('Sign up', { message: "passwords do not match", type: "error" })  else {  if (userinfo.seller == 'Sign Up as a Seller')  var userrole = 'seller'  else  var userrole = 'buyer'  var newUser = new Users({ name: userinfo.name, email: userinfo.email, password: userinfo.password, role: userrole });  newUser.save(function (err, Users) {  if (err) {  res.render('Sign up', { message: "Database error", type: "error" });  } else {  res.render('Log in', { message: "Sign up Successful!", type: "success" });  console.log(req.body)  }  })  }  })  app.post('/home', function (req, res) {  var userinfo = req.body  Users.findOne({ email: userinfo.email, password: userinfo.password }, function (err, response) {  if (err || response == null) {  res.render("Log in", { message: "Invalid Id or password!", type: "error" })  } else {  products.find({}, function (err, result) {  if (err)  throw err;  res.render("home", { products: result, user: response, type: "success" })  })  }  })  })  app.post('/save', function (req, res) {  var info = req.body  var newProduct = new products({ name: info.product\_name, category: info.product\_category, price: info.product\_price, sellerId: info.id });  newProduct.save(function (err, product) {  if (err) {  res.render('home', { message: "Database error", type: "error" });  } else {  Users.findOne({ \_id: info.id }, function (err, user) {  products.find({}, function (err, prod) {  res.render('home', { user: user, products: prod })  })  })  }  })  })  app.all('/signup', function (req, res) {  res.render('Sign up')  console.log(req.body)  })  app.post('/update', function (req, res) {  var details = req.body  var edit = { name: details.product\_name, price: details.product\_price, category: details.product\_category }  var query = { \_id: details.id }  products.findOneAndUpdate(query, edit, function (err, result) {  if (err) {  res.render('home', { message: "Database error", type: "error" });  } else {  Users.findOne({ \_id: result.sellerId }, function (err, user) {  products.find({}, function (err, prod) {  res.render('home', { user: user, products: prod })  })  })  }  })  })  app.post('/delete', function (req, res) {  var details = req.body  var query = { \_id: details.product\_id }  console.log(details)  products.findOneAndDelete(query, function (err, result) {  if (err) {  res.render('home', { message: "Database error", type: "error" });  } else {  Users.findOne({ \_id: details.sellerId }, function (err, user) {  products.find({}, function (err, prod) {  res.render('home', { user: user, products: prod })  })  })  }  })  })  app.listen(8080)  **Task Output Screenshot:** |

### Deliverable

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva/quiz related to the tasks. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to Ms. Ayesha Asif: [ayesha.asif@seecs.edu.pk](mailto:ayesha.asif@seecs.edu.pk).