# 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**

# 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

**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 mongoose = require("mongoose");  var bodyparser = require("body-parser");  var express = require("express");  var app = express();  var session = require('express-session');  const multer = require('multer');  const storage = multer.diskStorage({      destination: function (req, file, cb) {        cb(null, \_\_dirname + '/public/images')      },      filename: function (req, file, cb) {        cb(null, file.originalname)      }    })  const upload = multer({storage: storage});  app.use(express.static( \_\_dirname + '/public'));  app.use(bodyparser.urlencoded({ extended: true }));  app.use(session({secret: 'ssshhhhh'}));  var sess;  app.set("view engine", "pug");  app.set("views", "./views");    mongoose.connect("mongodb://localhost/PMS");  var personschema = mongoose.Schema({ name: String, email: String, password: String, role: String });  var Person = mongoose.model("Person", personschema);  var productschema = mongoose.Schema({ name: String, price: Number, category: String, image:String, seller\_id: String });  var Product = mongoose.model("Product", productschema);  app.get("/", function (req, res) {      res.render("login");  });  app.post("/", function (req, res) {      sess = null;      var personInfo = req.body;      if (personInfo.password != personInfo.password\_check) {          res.render("signup", { message: "Invalid Password!", type: "error" });      }      else {          var person\_role;          if (personInfo.seller == "Sign Up as a Seller")              person\_role = "Seller"          else              person\_role = "Buyer"          var newPerson = new Person({ name: personInfo.name, email: personInfo.email, password: personInfo.password, role: person\_role });          newPerson.save(function (err, Person) {              if (err) {                  res.render("signup", { message: "Error adding record!", type: "error" })              }              else {                  res.render("login", { message: "Signed up!", type: "success" });                  console.log(req.body);              }          })      }  });  app.all("/signup", function (req, res) {      res.render("signup");  });  app.post("/home", function (req, res) {        var personInfo = req.body;      Person.findOne({ email: personInfo.email, password: personInfo.password }, function (err, response) {          if(err || response == null) {              res.render("login", { message: "Invalid email or password", type: "error" });          }             else{                 sess= req.session;               sess.user = response;               Product.find(function(err, prod\_response){                  res.render("home", { user: sess.user, products: prod\_response });               })               console.log(response);           }      });  });  app.get("/home", function(req, res){      console.log(sess.user.\_id);      Product.find(function(err, prod\_response){          res.render("home", { user: sess.user, products: prod\_response });       })    })  app.post("/save", upload.single('photo'), function(req,res){      var productInfo = req.body;          console.log(req.file.filename);          var newProduct = new Product({ name: productInfo.name, price: productInfo.price, category: productInfo.category, image: "/images"+"/" + req.file.filename, seller\_id: sess.user.\_id });          newProduct.save(function (err, Person) {              if (err) {                  res.render("home", { message: "Error adding product!", type: "error" })              }              else {                  res.redirect("home");                  console.log(req.body);              }          })    })    app.post("/update", function(req,res){      var productInfo = req.body;          console.log(sess.user.\_id);          Product.findOneAndUpdate({ \_id : productInfo.id  },{name: productInfo.name, price: productInfo.price, category: productInfo.category},function(err, prod\_response){                res.redirect("home");             })    })  app.post("/delete", function(req,res){      var productInfo = req.body;          console.log(sess.user.\_id);          Product.findByIdAndRemove(productInfo.id,function(err, prod\_response){                res.redirect("home");             })    })    app.listen(3000); |

### 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).