**Muhammad Tayyab**

**217038**

**BSCS 7B**

**Assignment #2**

# Code:

***App.js:***

var express = require('express');

var path = require('path');

var favicon = require('serve-favicon');

var logger = require('morgan');

var cookieParser = require('cookie-parser');

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

var hbs = require('hbs');

var session = require('express-session');

var index = require('./routes/index');

var app = express();

// view engine setup

app.set('views', path.join(\_\_dirname, 'views'));

hbs.registerPartials(\_\_dirname + '/views/partials');

app.set('view engine', 'hbs');

// uncomment after placing your favicon in /public

//app.use(favicon(path.join(\_\_dirname, 'public', 'favicon.ico')));

app.use(logger('dev'));

app.use(bodyParser.json());

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

app.use(cookieParser());

app.use(session({

secret: 'secret',

resave: false,

saveUninitialized: true

}))

app.use(express.static(path.join(\_\_dirname, 'public')));

// res.locals is an object passed to hbs engine

app.use(function(req, res, next) {

res.locals.session = req.session;

next();

});

app.use('/', index);

// catch 404 and forward to error handler

app.use(function(req, res, next) {

var err = new Error('Not Found');

err.status = 404;

next(err);

});

// error handler

app.use(function(err, req, res, next) {

// set locals, only providing error in development

res.locals.message = err.message;

res.locals.error = req.app.get('env') === 'development' ? err : {};

// render the error page

res.status(err.status || 500);

res.render('error');

});

***Cart.js:***

module.exports = app;

module.exports = function Cart(cart) {

this.items = cart.items || {};

this.totalItems = cart.totalItems || 0;

this.totalPrice = cart.totalPrice || 0;

this.add = function(item, id) {

var cartItem = this.items[id];

if (!cartItem) {

cartItem = this.items[id] = {item: item, quantity: 0, price: 0};

}

cartItem.quantity++;

cartItem.price = cartItem.item.price \* cartItem.quantity;

this.totalItems++;

this.totalPrice += cartItem.item.price;

};

this.remove = function(id) {

this.totalItems -= this.items[id].quantity;

this.totalPrice -= this.items[id].price;

delete this.items[id];

};

this.getItems = function() {

var arr = [];

for (var id in this.items) {

arr.push(this.items[id]);

}

return arr;

};

};

***Index.js:***

var express = require('express');

var router = express.Router();

var fs = require('fs');

var Cart = require('../models/cart');

var products = JSON.parse(fs.readFileSync('./data/products.json', 'utf8'));

router.get('/', function (req, res, next) {

var productId = products && products[0].id;

res.render('index',

{

title: 'NodeJS Shopping Cart',

products: products

}

);

});

router.get('/add/:id', function(req, res, next) {

var productId = req.params.id;

var cart = new Cart(req.session.cart ? req.session.cart : {});

var product = products.filter(function(item) {

return item.id == productId;

});

cart.add(product[0], productId);

req.session.cart = cart;

res.redirect('/');

inline();

});

router.get('/cart', function(req, res, next) {

if (!req.session.cart) {

return res.render('cart', {

products: null

});

}

var cart = new Cart(req.session.cart);

res.render('cart', {

title: 'NodeJS Shopping Cart',

products: cart.getItems(),

totalPrice: cart.totalPrice

});

});

router.get('/remove/:id', function(req, res, next) {

var productId = req.params.id;

var cart = new Cart(req.session.cart ? req.session.cart : {});

cart.remove(productId);

req.session.cart = cart;

res.redirect('/cart');

});

module.exports = router;

# Output:



