1. **Be able to add a new User using a form.**

[**models/user.js**](https://github.com/jpalmat/MWA/commit/a4e7dab6080cfae1746f00d7ee76852806a24c63#diff-d4bb506efaa68907a7cd3e46833a934c)

const { ObjectID } = require('mongodb');

const getDb = require('../util/database').getDb;

class User {

constructor(username, email, cart, id) {

this.username = username;

this.email = email;

this.cart = cart; // {items: []}

this.\_id = id;

}

save() {

const db = getDb();

return db.collection('users')

.insertOne(this);

}

static findById(userId) {

const db = getDb();

console.log('the user id is '+userId);

return db.collection('users').findOne({ \_id: userId });

}

module.exports = User;

[**controllers/admin.js**](https://github.com/jpalmat/MWA/commit/a4e7dab6080cfae1746f00d7ee76852806a24c63#diff-ece366a40c91b6f7beae0db22760d1ce)

const User = require('../models/user');

exports.getUserForm = (req, res, next) => {

res.render('add-user', { name: 'Jimmy', path: '/admin/add-user', pageTitle: 'Add User' });

}

exports.postUser = (req, res, next) => {

const user = new User(req.body.name, req.body.email, [], '5e49bdefce95301ad0fa9870');

user.save();

res.redirect('/');

}

[**/routes/admin.js**](https://github.com/jpalmat/MWA/commit/a4e7dab6080cfae1746f00d7ee76852806a24c63#diff-972fbb4c0678b99b68c18b1edbc89f2c)

router.get('/add-new-user', adminController.getUserForm);

router.post('/add-new-user', adminController.postUser);

[**views/add-user.ejs**](https://github.com/jpalmat/MWA/commit/a4e7dab6080cfae1746f00d7ee76852806a24c63#diff-4c575a0128bd59368161da742c8a3ea7)

<%- include('layouts/head.ejs')%>

<body>

<%- include('layouts/navigation.ejs')%>

<!-- Page Content -->

<div class="container">

<div class="row">

<%- include('layouts/sidebar.ejs')%>

<div class="col-lg-9">

<%- include('layouts/carousel.ejs')%>

<div class="row">

<div class="col-sm-12">

<h2>Add Product</h2>

<form action="/admin/add-new-user" method="POST">

<div class="form-group">

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

<input class="form-control" id="name" name="name">

</div>

<div class="form-group">

<label for="email">Email</label>

<input class="form-control" id="email" name="email">

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

<br>

</div>

</div>

<!-- /.row -->

</div>

<!-- /.col-lg-9 -->

</div>

<!-- /.row -->

</div>

<!-- /.container -->

<%- include('layouts/end.ejs')%>

[**/layouts/navigation.ejs**](https://github.com/jpalmat/MWA/commit/a4e7dab6080cfae1746f00d7ee76852806a24c63#diff-6b88c1fae2b55e945fca46063449f400)

<li class="nav-item">

<a class="nav-link" href="/admin/add-new-user">Add New User</a>

</li>

1. **Add/remove products to shopping cart**

**User.js**

addToCart(product) {

if (this.cart === undefined) {

this.cart = { items: [] };

}

this.cart = { items: [] };

const cartProductIndex = this.cart.items.findIndex(cp => {

return cp.productId.toString() === product.\_id.toString();

});

let newQuantity = 1;

const updatedCartItems = [...this.cart.items];

if (cartProductIndex >= 0) {

newQuantity = this.cart.items[cartProductIndex].quantity + 1;

updatedCartItems[cartProductIndex].quantity = newQuantity;

} else {

updatedCartItems.push({

productId: new ObjectID(product.\_id),

quantity: newQuantity

});

}

}

deleteItemFromCart(productId) {

const updatedCartItems = this.cart.items.filter(item => {

return item.productId.toString() !== productId.toString();

});

const db = getDb();

return db

.collection('users')

.updateOne(

{ \_id: new ObjectID(this.\_id) },

{ $set: { cart: { items: updatedCartItems } } }

);

}

1. **Place an order**

**User.js**

addOrder() {

const db = getDb();

return this.getCart()

.then(products => {

const order = {

items: products,

user: {

\_id: new ObjectID(this.\_id),

name: this.name

}

};

return db.collection('orders').insertOne(order);

})

.then(result => {

this.cart = { items: [] };

return db

.collection('users')

.updateOne(

{ \_id: new ObjectID(this.\_id) },

{ $set: { cart: { items: [] } } }

);

});

}

**Controller/shop.js**

exports.postOrder = (req, res, next) => {

let fetchedCart;

req.user

.addOrder()

.then(result => {

res.redirect('/orders');

})

.catch(err => console.log(err));

};

**Routes/shop.js**

router.post('/create-order', shopController.postOrder);

**Cart.ejs**

<div class="centered">

<form action="/create-order" method="POST">

<button type="submit" class="btn">Order Now!</button>

</form>

</div>