

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

models/user.js

```
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

```
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

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

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

views/add-user.ejs

```
<%- 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

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

2. 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 } } }
);
}

```

3. 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>
```