

Initial Setup and Running the Application

Navigate to the project directory and run **npm install** to install the node packages.

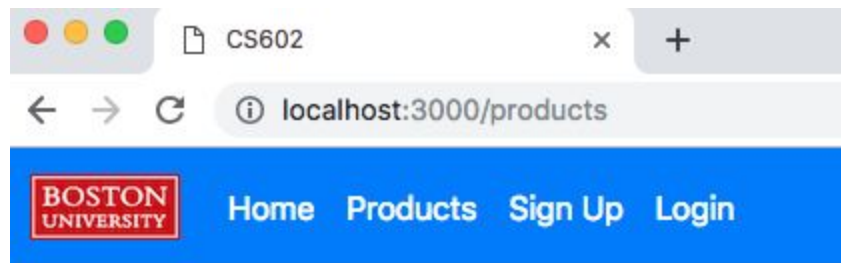
Populate the database by running **node populate_database.js**

Start the application by running **node server.js**

In a web browser navigate to **http://localhost:3000**

Creating an Account

In the navigation bar click on **Sign Up**

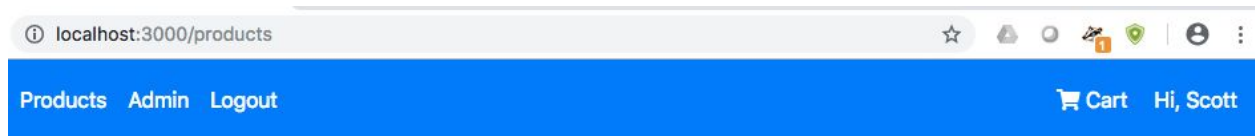


Fill out the new user form to create an account. The web application is using **passport.js** for authentication as well as **bcrypt** for securely storing the passwords.

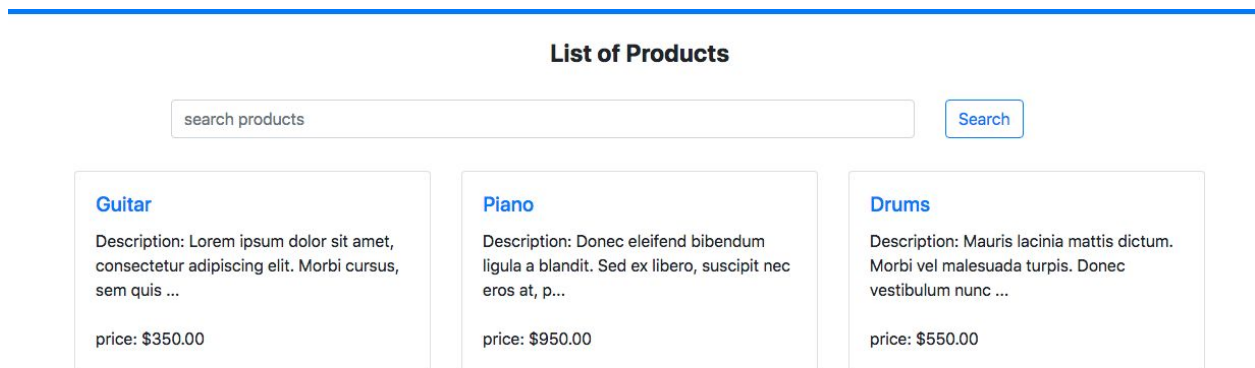
Sign Up

First Name:	<input type="text" value="Scott"/>
Last Name:	<input type="text" value="Kaeneman"/>
Email:	<input type="text" value="skaeneman@gmail.com"/>
Password:	<input type="password" value="....."/>
Confirm Password:	<input type="password" value="....."/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	
<hr/>	
Already have an account? login	

After creating an account the user is automatically signed into the application which is evidenced by their first name appearing in the top right of the navigation bar. The other links have now changed on the navigation bar after signing in to display the cart, admin interface, user profile, and logout links.

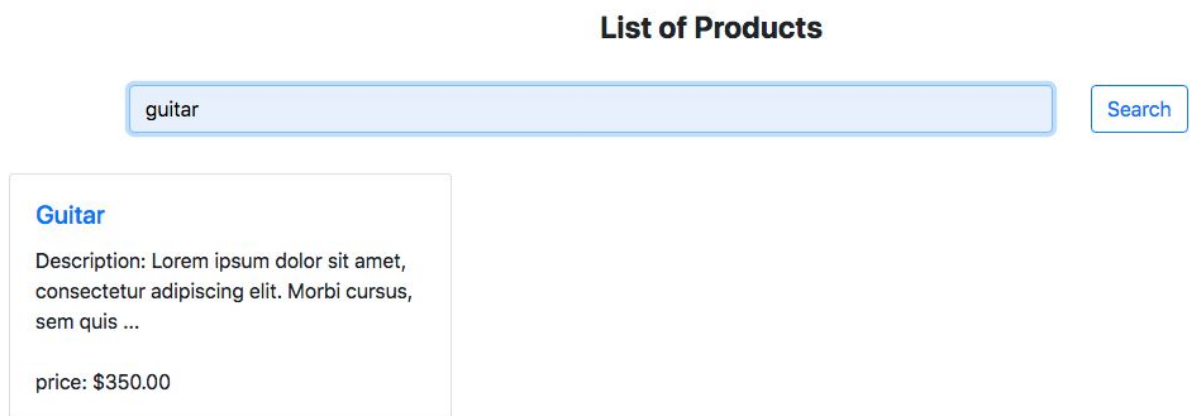


The application shows a list of all available products on the main products page.



Searching for Products

A user can **search for a product by name or description**. **Fuzzy search** is utilized so that the entire name or description does not have to be entered as input, partial words or phrases can be used.



Clicking the search button with nothing in the search box will render all the products again.

Admin Interface - Adding and Updating a Product

Click on the **Admin** link on the top left of the navigation bar. Then click on **Manage products**

Admin

Manage products	Manage users
-----------------	--------------

To **Add a new product** click on the Add button located at the bottom of the product list.

List of Products

Product Name	Description	Price	Quantity	
Guitar	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi cursus, sem quis ...	350	10	Edit Delete
Piano	Donec eleifend bibendum ligula a blandit. Sed ex libero, suscipit nec eros at, p...	950	20	Edit Delete
Drums	Mauris lacinia mattis dictum. Morbi vel malesuada turpis. Donec vestibulum nunc ...	550	15	Edit Delete
Add				

Fill out the product form and click **Save**.

Add a product

Product Name:	<input type="text" value="Test Product"/>
Description:	<input type="text" value="Test Product Description"/>
Price:	<input type="text" value="100"/>
Quantity:	<input type="text" value="10"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

The new product has now been added to the product list.

Product Name	Description	Price	Quantity	
Guitar	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi cursus, sem quis ...	350	10	Edit Delete
Piano	Donec eleifend bibendum ligula a blandit. Sed ex libero, suscipit nec eros at, p...	950	20	Edit Delete
Drums	Mauris lacinia mattis dictum. Morbi vel malesuada turpis. Donec vestibulum nunc ...	550	15	Edit Delete
Test Product	Test Product Description	100	10	Edit Delete
Add				

To Edit the product click on the **edit link**. Both the add and edit product forms **protect against strings and negative numbers entered into the price and quantity fields**.

Invalid input - please enter a postive integer ×

Name:

Description:

Price:

Quantity:

[Save](#) [Cancel](#)

The new product has been added to the database collection.

key	value
▶ (1) ObjectId("5c81a4b1ad7e0828be457a74")	{ 7 fields }
▶ (2) ObjectId("5c81a4b1ad7e0828be457a76")	{ 7 fields }
▶ (3) ObjectId("5c81a4b1ad7e0828be457a75")	{ 7 fields }
▼ (4) ObjectId("5c81ae60905aea28c1ae0f81")	{ 7 fields }
_id	ObjectId("5c81ae60905aea28c1ae0f81")
qtyCount	[11 elements]
name	Test Product
description	Test Product Description
price	100
quantity	10
__v	0

After saving the edit you are returned to the main admin products page. Click on **delete** to remove the product from the database.

Ordering Products

A user can order a product by navigating to the main products page **localhost:3000/products** and then clicking on a product name.

Show product

Guitar

\$350.00

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi cursus, sem quis congue vulputate, neque erat faucibus nibh, ut pharetra ex purus non quam. Cras lobortis, augue a malesuada faucibus, quam neque imperdiet dui, non vulputate mi sem vel purus. Phasellus non facilisis massa. Ut sit amet mattis mi. Cras eu nulla nisl. Duis fringilla scelerisque condimentum. Praesent placerat commodo lacinia. Suspendisse finibus lectus sollicitudin, rutrum felis id, elementum libero. Ut magna lorem, varius a sagittis nec, vehicula ac ligula. Phasellus vulputate aliquet lectus, in rhoncus mauris convallis a. Praesent imperdiet nunc ut consequat condimentum.

Only 10 left in stock - order soon.

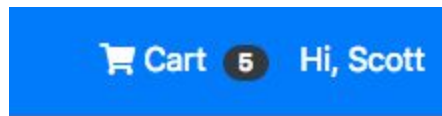
Quantity

The dropdown will render how many products are available in the database.

#	price	350
#	quantity	10
#	_v	0
▶	(2) ObjectId("5c81a4b1ad7e0828be457a76")	{ 7 fields }
▶	(3) ObjectId("5c81a4b1ad7e0828be457a75")	{ 7 fields }

To order a product click on the quantity dropdown to make a selection and then click **Add to cart** button. In the **top right of the navigation bar** the cart quantity will be displayed.

Click on the **cart icon** to visit the cart (5 guitars out of the 10 available were added to the cart).



The **cart can be emptied**. Or an **order can be placed** by clicking on the **Checkout button**.

Product Name	Price	Quantity
Guitar	1750	5

cart total: 1750

[Cancel](#) [Empty](#) [Checkout](#)

The **order total page** is now displayed. Click on **Place your order**.

Order Total

1 Shipping Address:

742 Evergreen Terrace, Springfield

2 Payment method

credit card

Visa 1234 5678 9101 1234

exp

09/20

cvc

123

3 Review items and shipping

Product Name	Price	Quantity
Guitar	\$1,750.00	5

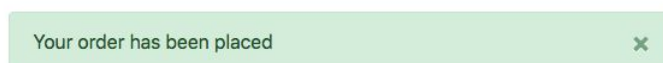
Quantity: 5

Order total: \$1,750.00

[Cancel](#) [Place your order](#)

The user is redirected back to the home page with a message that the order was completed

List of Products



To **verify an order click on the user name** in top right of the navbar to go to the users profile.

User Profile

order id: 5c81b552905aea28c1ae0f82

created at: Thu Mar 07 2019 19:20:34 GMT-0500 (Eastern Standard Time)

Guitar	\$1,750.00	5
--------	------------	---

order quantity: 5

order total: \$1,750.00

Query the **product collection** and observe that the count has decreased by 5 items.

cs602_termproject localhost:27017 cs602_termproject_db

```
db.getCollection('productmodels').find({})
```

productmodels 0.001 sec.

Key	Value
(1) ObjectId("5c81a4b1ad7e0828be457a74")	{ 7 fields }
_id	ObjectId("5c81a4b1ad7e0828be457a74")
qtyCount	[6 elements]
name	Guitar
description	Lorem ipsum
price	350
quantity	5
__v	1

The **order collection** shows a purchase completed with a quantity of 5 items.

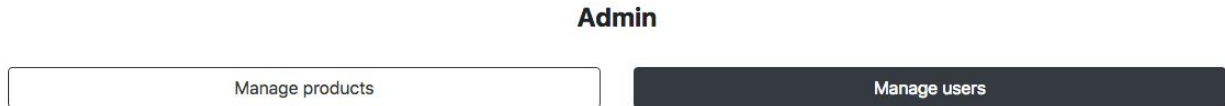
```
db.getCollection('ordermodels').find({})
```

ordermodels 0.002 sec.

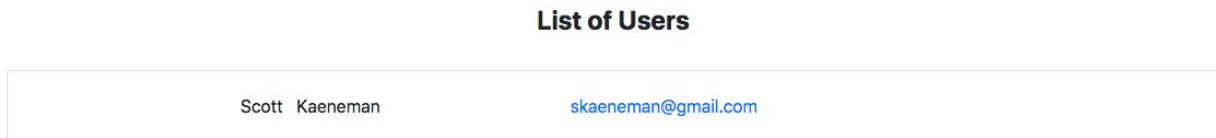
Key	Value
(1) ObjectId("5c81b552905aea28c1ae0f82")	{ 7 fields }
_id	ObjectId("5c81b552905aea28c1ae0f82")
shoppingCart	{ 3 fields }
products	{ 1 field }
5c81a4b1ad7e0828be457a74	{ 3 fields }
prod	{ 7 fields }
price	1750
quantity	5
cartQuantity	5
cartTotal	1750
userId	ObjectId("5c81a617905aea28c1ae0f82")
orderTotal	1750
orderQuantity	5
createdAt	2019-03-08 00:20:34.198Z
__v	0

Admin Interface - Show list of Customers

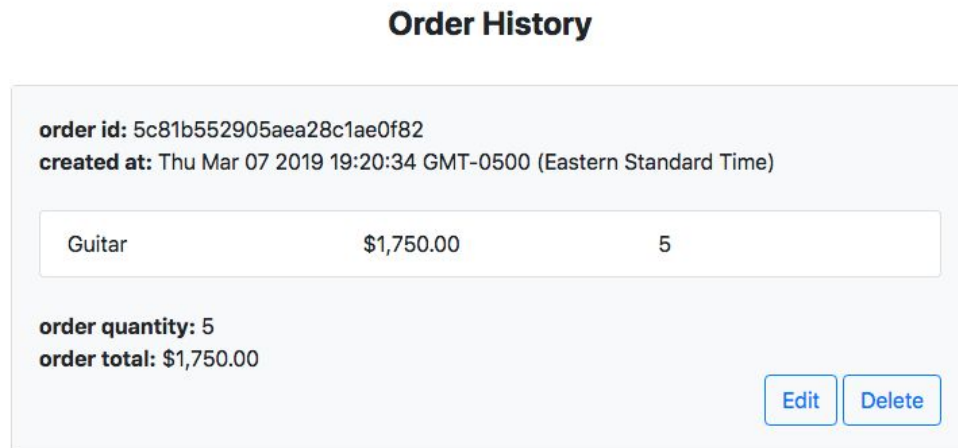
Click on the **Admin** link on the top left of the navigation bar. Then on **Manage users**.



The **List of Users** will be displayed. Click on the users email address to see their orders.



The **users order history** page is now displayed.



Admin Interface - Add Products to a Users Order

Change the **quantity from 5 to 6** to **add** a product and **click Save**.

Edit Order

Guitar	6
--------	---

Cancel

Save

The admin will be redirected back to the order history for that particular user. The updated order quantity will be displayed.

Order History

Order id 5c81b552905aea28c1ae0f82 successfully updated



order id: 5c81b552905aea28c1ae0f82

created at: Thu Mar 07 2019 19:20:34 GMT-0500 (Eastern Standard Time)

Guitar

\$2,100.00

6

order quantity: 6

order total: \$2,100.00

Edit

Delete

Check the database to verify that the quantity in the **product collection** has **decreased by a count of 1** to a **total of 4** items remaining.

```
db.getCollection('productmodels').find({})
```

productmodels 0.001 sec.	
Key	Value
(1) ObjectId("5c81a4b1ad7e0828be457a74")	{ 7 fields
_id	ObjectId
qtyCount	[6 elem
name	Guitar
description	Lorem ip
price	350
quantity	4

The **order collection** reflects the new product being added to it's quantity for a **total of 6 items**.

```
db.getCollection('ordermodels').find({})
```

ordermodels 0.002 sec.	
Key	Value
(1) ObjectId("5c81b552905aea28c1ae0f82")	{ 7 fields }
_id	ObjectId("5c
shoppingCart	{ 3 fields }
products	{ 1 field }
5c81a4b1ad7e0828be457a74	{ 3 fields }
prod	{ 7 fields }
price	2100
quantity	6
cartQuantity	6
cartTotal	2100

Admin Interface - Delete Products from a Users Order

Back on the **users order history** page click on **edit** again.

order id: 5c81b552905aea28c1ae0f82
created at: Thu Mar 07 2019 19:20:34 GMT-0500 (Eastern Standard Time)

Guitar	\$2,100.00	6
--------	------------	---

order quantity: 6
order total: \$2,100.00

Edit
Delete

This time **decrease the order to 0** and click on the **Save button**.

Edit Order

Guitar	0
--------	---

CancelSave

The order quantity and total have been updated to reflect the new values.

Order History

Order id 5c81b552905aea28c1ae0f82 successfully updated

order id: 5c81b552905aea28c1ae0f82
created at: Thu Mar 07 2019 19:20:34 GMT-0500 (Eastern Standard Time)

Guitar	\$0.00	0
--------	--------	---

order quantity: 0
order total: \$0.00

Edit Delete

Check the quantities in the database to verify. The product collection shows all 6 items have been added back.

```
db.getCollection('productmodels').find({})
```

Key	Value
(1) ObjectId("5c81a4b1ad7e0828be457a74")	{ 7 fields }
_id	ObjectId("5c81a4b1ad7e0828be457a74")
qtyCount	[6 elements]
name	Guitar
description	Lorem ipsum d
price	350
quantity	10

The order collection correctly shows that it now has 0 items.

```
db.getCollection('ordermodels').find({})
```

ordermodels 0.002 sec.	
Key	Value
▼ (1) ObjectId("5c81b552905aea28c1ae0f8...")	{ 7 fields }
_id	ObjectId("5c81b552905aea28c1ae0f8...")
shoppingCart	{ 3 fields }
products	{ 1 field }
5c81a4b1ad7e0828be457a74	{ 3 fields }
prod	{ 7 fields }
price	0
quantity	0
cartQuantity	0
cartTotal	0

Admin Interface - Delete a Users Order

Back on the users order page click on the **delete button** to remove the order.

Order History

order id: 5c81b552905aea28c1ae0f82
created at: Thu Mar 07 2019 19:20:34 GMT-0500 (Eastern Standard Time)

Guitar	\$0.00	0
--------	--------	---

order quantity: 0
order total: \$0.00

EditDelete

A notice that the order has been deleted will be shown.

Order History

Order id 5c81b552905aea28c1ae0f82 successfully deleted ✕

Querying the database for the order collection shows that the order has been removed.

```
cs602_termproject localhost:27017 cs
db.getCollection('ordermodels').find({})
0.001 sec.
Fetched 0 record(s) in 1ms
```

REST API's (XML and JSON)

There are 3 different queries that can be performed to retrieve data from the application.

Retrieve all products in XML format by going to <http://localhost:3000/rest/xml/products> or in JSON by going to <http://localhost:3000/rest/json/products>

```
← → ↻ ⓘ localhost:3000/rest/xml/products
This XML file does not appear to have any style information associa

<products>
  <product>
    <id>5c81a4blad7e0828be457a74</id>
    <name>Guitar</name>
    <description>
      Lorem ipsum dolor sit amet, consectetur adipiscing
      mi sem vel purus. Phasellus non facilisis massa. U
      libero. Ut magna lorem, varius a sagittis nec, veh
    </description>
    <price>350</price>
    <quantity>10</quantity>
    <qtyCount>0,1,2,3,4,5</qtyCount>
  </product>
  <product>
    <id>5c81a4blad7e0828be457a76</id>
    <name>Piano</name>
    <description>
      Donec eleifend bibendum ligula a blandit. Sed ex l
      nisi. Donec sagittis, libero nec imperdiet porta,
      Pellentesque faucibus condimentum dolor, a volutpa
    </description>
    <price>950</price>
    <quantity>20</quantity>
    <qtyCount/>
  </product>
  <product>
    <id>5c81a4blad7e0828be457a75</id>
    <name>Drums</name>
    <description>
      Mauris lacinia mattis dictum. Morbi vel malesuada
      nisi justo, vitae lobortis augue facilisis quis. N
    </description>
    <price>550</price>
    <quantity>15</quantity>
    <qtyCount/>
  </product>
</products>
```

Retrieve a product by name by appending the product name to the end of the url

XML = `http://localhost:3000/rest/xml/products/product_name`

JSON = `http://localhost:3000/rest/json/products/product_name`

← → ↻ ⓘ localhost:3000/rest/xml/products/guitar

This XML file does not appear to have any style information associated with it.

```
<products>
  <product>
    <id>5c81cb5feb9b3e2f8b65fd84</id>
    <name>guitar</name>
    <description>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit.
      mi sem vel purus. Phasellus non facilisis massa. Ut sit
      libero. Ut magna lorem, varius a sagittis nec, vehicula
    </description>
    <price>350</price>
    <quantity>10</quantity>
    <qtyCount/>
  </product>
</products>
```

← → ↻ ⓘ localhost:3000/rest/json/products/guitar

```
{
  "products": [
    {
      "qtyCount": [],
      "_id": "5c81cb5feb9b3e2f8b65fd84",
      "name": "guitar",
      "description": "Lorem ipsum dolor
neque imperdiet dui, non vulputate mi sem vel purus. Phas
rutrum felis id, elementum libero. Ut magna lorem, varius
      "price": 350,
      "quantity": 10,
      "__v": 0
    }
  ]
}
```

Retrieve a product by price range by appending the product price range to the url

XML = `http://localhost:3000/rest/xml/products/min_price/max_price`

JSON = `http://localhost:3000/rest/json/products/min_price/max_price`

Here we are getting products in the price range of \$300 to \$600 and 2 out of the 3 products are returned since the third product is outside of the price range.

```
← → ↻ ⓘ localhost:3000/rest/xml/products/300/600

This XML file does not appear to have any style information associated with it.

<?xml version="1.0"?>
<products>
  <product>
    <id>5c81cb5feb9b3e2f8b65fd84</id>
    <name>guitar</name>
    <description>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit.
      mi sem vel purus. Phasellus non facilisis massa. Ut
      libero. Ut magna lorem, varius a sagittis nec, vehicula
    </description>
    <price>350</price>
    <quantity>10</quantity>
    <qtyCount/>
  </product>
  <product>
    <id>5c81cb5feb9b3e2f8b65fd85</id>
    <name>drums</name>
    <description>
      Mauris lacinia mattis dictum. Morbi vel malesuada turpis.
      nisi justo, vitae lobortis augue facilisis quis. Nullam
    </description>
    <price>550</price>
    <quantity>15</quantity>
    <qtyCount/>
  </product>
</products>
```

```
← → ↻ ⓘ localhost:3000/rest/json/products/300/600

{
  "products": [
    {
      "qtyCount": [],
      "_id": "5c81cb5feb9b3e2f8b65fd84",
      "name": "guitar",
      "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. mi sem vel purus. Phasellus non facilisis massa. Ut libero. Ut magna lorem, varius a sagittis nec, vehicula",
      "price": 350,
      "quantity": 10,
      "__v": 0
    },
    {
      "qtyCount": [],
      "_id": "5c81cb5feb9b3e2f8b65fd85",
      "name": "drums",
      "description": "Mauris lacinia mattis dictum. Morbi vel malesuada turpis. nisi justo, vitae lobortis augue facilisis quis. Nullam",
      "price": 550,
      "quantity": 15,
      "__v": 0
    }
  ]
}
```