

\*\*\*Please test this web app on Chrome.\*\*\*

## **Summary of feedback and changes made to this Project:**

We have made many changes to our project based on the feedback from our peer reviews as well as from our own design decisions. Our biggest decision was to convert our React project to HTML, prior to submitting Project Step 3 Draft Version, after failing to understand how to connect our project to the flip server.

We made a few changes on our entity relationships along the way. We corrected the ‘Orders’ relationships with ‘Customers’ and ‘Employees’ from 1:1 to 1:M, and the ‘OrderDetails’ relationships with ‘Orders’ and ‘Products’ from 1:1 to 1:M after a few of our peers brought the issue to our attention. We also updated the ‘Orders’ relationship with ‘OrderDetails’ to orders being related to 0 or more ‘OrderDetails’ to accommodate for cancelled orders. In addition, we updated the ‘Orders’ relationship with ‘Employees’ to allow an order to be related to 0 or 1 employee to fulfill the requirement of allowing a 1:M relationship Foreign Key to be set as NULL.

We made several updates to various attributes as well. We chose to remove the attribute ‘orderDetailID’ from ‘OrderDetails’ since we created a primary key by concatenating the foreign keys ‘OrderNumber’ and ‘ProductCode’. We also removed ‘orderLineNumber’ from ‘OrderDetails’ after feedback from a peer because it did not serve a purpose. We chose to remove “pending” as an option for the ‘Orders’ orderStatus attribute seeing as a shippedDate would not exist if an order was pending. Furthermore, we made minor changes to attribute types and formats to allow for additional characters and consistency between UI and database respectively.

We made very little, but vital, changes to our DML. The first change we made was updating our SQL queries for ‘Orders’. Initially we used an INNER JOIN to join the ‘Employees’ table to our ‘Orders’ table using Orders.employee\_id = Employees.employee\_id. However, we discovered that this was excluding the Orders.employee\_id = NULL. To solve this issue, we changed INNER JOIN to LEFT JOIN so that all the rows from ‘Orders’ would be included in our table. In addition, we updated the ‘Run Product Report’ query by changing INNER JOIN to RIGHT JOIN; as well as, adding IFNULL and WHERE clause to display all active products and their total amount of purchases. These database query changes were necessary since our client requests the total purchase amount of each active product before deciding which products to keep available for the following year.

We also made some important updates to our DDL. We incorporated UNIQUE constraints in our Customers, Employees, and Products tables to prevent identifying attributes to share the same value. Furthermore, we removed ON UPDATE CASCADE in our foreign keys since our primary keys are all auto\_incrementing and not allowed to be updated or deleted.

Lastly, we made a couple of key changes to our project code. We were able to troubleshoot our HTML form validation issue by using “if(this.form.checkValidity())” to confirm whether a form has been filled out correctly prior to submission. Finally, we managed to create drop-down menus for our foreign key inputs during Week 10. We decided to include this change to our project because of the notes in Project Step 5 explaining why allowing the user to enter IDs is unacceptable.

## **Project Overview, Database Outline, ERD, and Schema:**

### **Project Overview:**

Canine Munchies is a newly opened, online canine treat online company operating out of Corvallis, Oregon and serving ~10-20 customers a day in the Pacific Northwest (US only). The company currently offers 10 different canine treats for purchase. To ensure maximum quality and freshness, all treats are made to order in the shop's Department of Health – Certified kitchen using only premium ingredients.

Canine Munchies would like to determine which 5 treats, of the 10 currently available, have the highest total order count within a calendar year. The top 5 treats will remain on the menu for the following year, while the bottom 5 will be replaced with new treats. Since all products are expected to have the same production cost, the decision of which treats to replace will be based solely on the total quantity sold for each product. The company is not interested in tracking product costs or sales figures.

To determine the production schedule the owners have commissioned the development of a backend database, Munchies Orders, that will record customer purchase orders. The database has complete CRUD (create, read, update, and delete) functionality and consists of five entity tables (Customers, Products, Orders, OrderDetails, and Employees) that record all relevant information regarding a customer's purchase. In addition to the CRUD functionality, the database will be capable of creating a Customers entity without purchasing a product. However, if a product is purchased, a Customers entity must be tied to that purchase (ie. no "Guest" checkout).

To illustrate the desired CRUD functionality, imagine a customer orders a product (or multiple products) from the online store. First, the database will create Customers, Orders, and OrderDetails entities based on the purchase(s). The created Customers entity will capture all relevant information about the customer and the Orders entity will capture the order summary, mostly shipping status information. Additionally, an OrderDetails entity will be created, with the number of entities created matching the number of different products purchased. If the customer decides to modify their profile or an order after it has been submitted, the database will need to be updated with the new information. The "read" and "delete" functionality will be primarily used by database administrators to retrieve and analyze data or modify the database manually in the event some data needs to be deleted.

Using Munchies Orders will allow administrators to efficiently gather relevant information regarding customer purchase orders. This will be vital information that will allow analysts to easily identify order trends, which will then impact future product decisions.

## **Database Outline:**

### **Products:**

**Description of Entity** – Records product details that a customer can order and legacy products that can no longer be ordered, but that still exist in product records.

**Relationships** – ‘Products’ will have a 1:M relationship with ‘OrderDetails’, with a minimum number of 0 order detail per product and productCode implemented as the FK inside of ‘OrderDetails’.

#### **Attributes –**

- **productCode** – int(3); auto\_incremented; starting value of 1; unique; not *NULL*; primary key
- **productName** – varchar(50); not *NULL*
- **productDescription** – varchar(250); not *NULL*
- **legacyProduct** – boolean; not *NULL*
  - True means the product is **NOT** available.

### **Customers:**

**Description of Entity** – Records relevant customer shipping information and delivery details.

**Relationships** – ‘Customers’ will have a 1:M relationship with ‘Orders’, with a minimum number of 0 orders per customer and customerID implemented as the FK inside of ‘Orders’.

#### **Attributes –**

- **customerID** – int(5); auto\_incremented; starting value of 1; unique; not *NULL*; primary key
- **companyName** – varchar(50); *NULL* OK
  - May or may not be delivering to another business
- **contactLastName** – varchar(50); not *NULL*
- **contactFirstName** – varchar(50); not *NULL*
- **phoneNumber** – varchar(12); *NULL* OK
  - phoneNumber should be formatted in database as ‘XXX-XXX-XXXX’.
- **addressLine1** – varchar(50); not *NULL*
- **addressLine2** – varchar(50); *NULL* OK
- **city** – varchar(50); not *NULL*
- **contactState** – varchar(2); not *NULL*
- **postalCode** – varchar(5); not *NULL*
- **email** – varchar(50); not *NULL*
- **specialDeliveryInst** – varchar(200); *NULL* OK
  - Captures any special delivery instructions (ie. Use callbox for access to mailroom, etc.)

### **Orders:**

**Description of Entity** – Summarizes order receiving/shipping dates and status of shipments.

**Relationships** – ‘Orders’ will have a 1:M relationship with ‘Customers’, an order must be related to exactly 1 customer. ‘Orders’ will have a 1:M relationship with ‘Employees’, an order can be related to

## CS340 Project Step 7 Final Version

Group 129 – Staci Ihori, Nathan McAlister

one employee. ‘Orders’ will have a 1:M relationship with ‘OrderDetails’, an order can be related to one or many ‘OrderDetails’ and orderNumber implemented as the FK inside of ‘OrderDetails’.

### Attributes –

- **orderNumber** – int(5); auto\_incremented; starting value of 1; unique; not *NULL*; primary key
  - orderDate should be formatted in database as ‘YYYY-MM-DD’.
- **orderDate** – Date(); not *NULL*
  - requiredDate should be formatted in database as ‘YYYY-MM-DD’.
- **requiredDate** – Date(); not *NULL*
  - requiredDate should be formatted in database as ‘YYYY-MM-DD’.
- **shippedDate** – Date(); not *NULL*
  - shippedDate should be formatted in database as ‘YYYY-MM-DD’.
- **orderStatus** – varchar(9); not *NULL*
  - Should be one of two choices (‘Shipped’, ‘Cancelled’)
- **comments** – varchar(200); *NULL* OK
  - Capturing comments, especially about cancelled orders.
- **customerID** – int(5); not *NULL*
  - This field should be linked to a ‘Customer’ entity.
- **employeeID** – int(3); *NULL* OK
  - This field can be linked to a ‘Employee’ entity.

### OrderDetails:

**Description of Entity** – Composite entity that captures which products were purchased on each order.

**Relationships** – ‘OrderDetails’ will have a 1:M relationship with ‘Orders’, an ‘OrderDetails’ must be related to exactly 1 order. ‘OrderDetails’ will have a 1:M relationship with ‘Products’, an ‘OrderDetails’ must be related to exactly 1 product. ‘OrderDetails’ will create a M:M relationship between ‘Orders’ and ‘Products’.

### Attributes –

- **orderNumber** – int(5); not *NULL*;
  - Primary key (FK ‘orderNumber’ from ‘Orders’, FK ‘productCode’ from ‘Products’); unique
- **productCode** – int(3); not *NULL*;
  - Primary key (FK ‘orderNumber’ from ‘Orders’, FK ‘productCode’ from ‘Products’); unique
- **quantityOrdered** – int(2); not *NULL*;
  - Limits the maximum possible quantity to order to ‘99’.

### Employees:

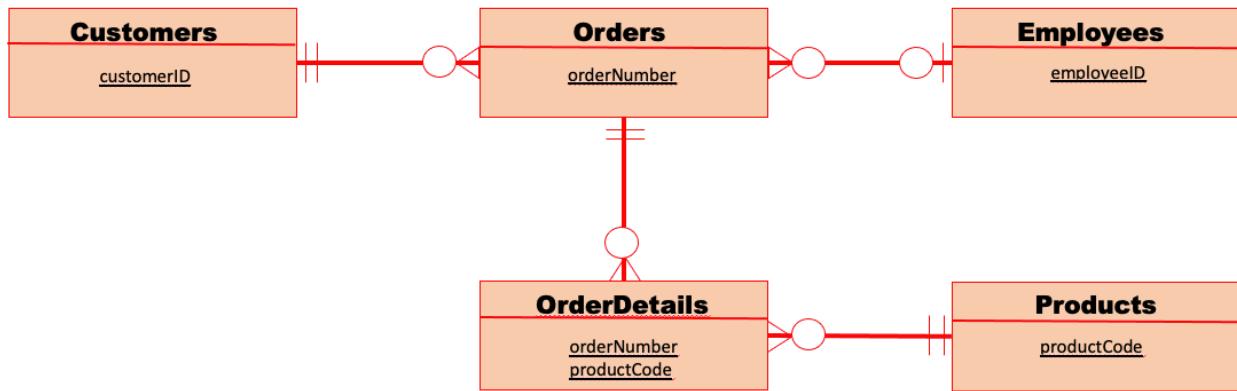
**Description of Entity** – Records employee personal information.

**Relationships** – ‘Employees’ will have a 1:M relationship with ‘Orders’, with a minimum number of 0 orders per employee and employeeID implemented as the FK inside of ‘Orders’.

**Attributes –**

- **employeeID** – int(3); unique; auto\_incremented; starting value of 1; not *NULL*; primary key
- **employeeLastName** – varchar(50); not *NULL*
- **employeeFirstName** – varchar(50); not *NULL*
- **phoneNumber** – varchar(12); not *NULL*
  - phoneNumber should be formatted in database as 'XXX-XXX-XXXX'.
- **addressLine1** – varchar(50); not *NULL*
- **addressLine2** – varchar(50); *NULL* OK
- **city** – varchar(50); not *NULL*
- **employeeState** – varchar(2); not *NULL*
- **postalCode** – varchar(5); not *NULL*
- **email** – varchar(50); not *NULL*
- **employeeStatus** – varchar(6); not *NULL*
  - Should be one of two choices ('active', 'termed')

**Entity-Relationship Diagram:**



## Schema:

```
Customers(  
    customerID,  
    companyName,  
    contactFirstName,  
    contactLastName,  
    phoneNumber,  
    addressLine1,  
    addressLine2,  
    city,  
    contactState,  
    postalCode,  
    email,  
    specialDeliveryInst)
```

```
Orders(  
    orderNumber,  
    orderDate,  
    requiredDate,  
    shippedDate,  
    orderStatus,  
    comments,  
    customerID,  
    employeeID)
```

```
Employees(  
    employeeID,  
    employeeFirstName,  
    employeeLastName,  
    phoneNumber,  
    addressLine1,  
    addressLine2,  
    city,  
    employeeState,  
    postalCode,  
    email,  
    employeeStatus)
```

```
OrderDetails(  
    orderNumber,  
    productCode,  
    quantityOrdered)
```

```
Products(  
    productCode,  
    productName,  
    productDescription,  
    legacyProduct)
```

## Screen Captures:

### Home Page:

Home Page with navigation side bar

# Canine Munchies

**Canine Munchies' Backend Database Portal**

Welcome to Canine Munchies' Backend Database Portal! This web portal gives the client access and CRUD capabilities to modify any of the entities within Canine Munchies Database! Use the menu on the left to navigate to your desired entity table!

### Customers Page:

CREATE/READ/UPDATE Customers Page

# Canine Munchies

## Customers

Please use one of the provided buttons to:

Add New Customer   Customer Look-up   List All Customers   Clear All

CREATE/INSERT/ADD a new customer

# Canine Munchies



Home Page

**Customers**

Employees

Order Details

Orders

Products

## Customers

Please use one of the provided buttons to:

[Add New Customer](#) [Customer Look-up](#) [List All Customers](#) [Clear All](#)

### Add New Customer

Please enter the following information:

Company Name:

Last Name:

First Name:

Phone Number:  
 XXX-XXX-XXXX

Address Line 1:

The screenshot shows a web application for managing canine customers. At the top, there is a search bar labeled "SEARCH/FILTER an existing customer". Below the search bar is the title "Canine Munchies" flanked by two small dog icons. A navigation menu on the left includes links for Home Page, Customers (which is highlighted in purple), Employees, Order Details, Orders, and Products. The main content area is titled "Customers" and contains a section for "Customer Look-up". This section includes fields for Customer ID Number, Company Name, Last Name, First Name, and Phone Number, each with a corresponding input box. There is also a note instructing users to use one of four provided buttons: Add New Customer, Customer Look-up, List All Customers, or Clear All.

SEARCH/FILTER an existing customer

# Canine Munchies

Home Page

Customers

Employees

Order Details

Orders

Products

## Customers

Please use one of the provided buttons to:

Add New Customer   Customer Look-up   List All Customers   Clear All

### Customer Look-up

If available, please enter the customer ID number:

Customer ID Number:

Otherwise, please enter the following information:

Company Name:

Last Name:

First Name:

Phone Number:  XXX-XXX-XXXX

CS340 Project Step 7 Final Version  
 Group 129 – Staci Ihori, Nathan McAlister  
**Customers Page (continued):**



**DISPLAY/UPDATE an existing customer**

# Canine Munchies

**Customers**

Please use one of the provided buttons to:

[Add New Customer](#) [Customer Look-up](#) [List All Customers](#) [Clear All](#)

customerID	companyName	contactFirstName	contactLastName	phoneNumber	addressLine1	addressLine2	city	contactState	postalCode	email	specialDeliveryInst	Update
1	Furry Fiestas	Nathan	McAlister	555-555-0000	744 Belmont Pl E	Studio 302A	Seattle	WA	98102	mcalistn@gmail.com	Please use the keypad code 653 to contact store clerk	<a href="#">Update</a>
2	Fuzzy Wuzzy Pet Store	Emiko	Nagashima	555-555-0001	2601 Bettie Dr		Seattle	WA	98102	em5764385@gmail.com		<a href="#">Update</a>
3		Staci	Ihori	555-555-0002	9954 Stone St.	APT #302	Dallas	OR	97338	ihoris@gmail.com	Leave outside apartment door	<a href="#">Update</a>
4	Pounce n Play Pet Store	Natalsha	Woodruff	555-555-0003	4454 Jinx Cr		Portland	OR	97035	iamwoodruff@yahoo.com	Home business - leave on front porch	<a href="#">Update</a>
5	Pounce n Play Pet Store	Chris	Vilhauer	555-555-0003	4454 Jinx Cr		Portland	OR	97035	iamvilhauer@yahoo.com	Home business - leave on main house front porch	<a href="#">Update</a>
6		Richard	Alanis	555-555-0005	543 Blackrock Ln		Salem	OR	97301	alanis@gmail.com		<a href="#">Update</a>
7		Ivan	Gomez	555-555-0006	3498 Center St.	APT #120	Tacoma	WA	98402	gomeyluvets@hotmail.com		<a href="#">Update</a>
8	Puppy Paws Pets	Megan	Johnson	555-555-0007	6234 Lasas Dr.		Olympia	WA	98501	iheartchris@yahoo.com		<a href="#">Update</a>
9		Pet	Mitchel	555-555-0008	1532 Hawaii St.	APT #451	Seattle	WA	98102	maverick@gmail.com	Just leave on back porch	<a href="#">Update</a>
10		Nick	Bradshaw	555-555-0009	1532 Hawaii St.	APT #452	Seattle	WA	98102	goose@gmail.com	Just leave on back porch	<a href="#">Update</a>
11		Tom	Kazansky	555-555-0010	1532 Hawaii St	APT #453	Seattle	WA	98102	iceman@gmail.com		<a href="#">Update</a>
12	We Love Pets	John	Keating	555-555-0011	4123 Welton Cr		Portland	OR	97035	ocaptain@gmail.com	Please deliver bewteen 11-2	<a href="#">Update</a>
13	Aww Pet	Hollie	Parker	4444444444444444	7942 Fairview Rd		Tacoma	WA	98407	narintran1@gmail.com		<a href="#">Update</a>

**Employees Page:**

CREATE/READ/UPDATE Employees Page

# Canine Munchies



Home Page

Customers

**Employees**

Order Details

Orders

Products

## Employees

Please use one of the provided buttons to:

[Add New Employee](#)

[Employee Look-up](#)

[List All Employees](#)

[Clear All](#)

CREATE/INSERT/ADD a new employee

# Canine Munchies



Home Page

Customers

**Employees**

Order Details

Orders

Products

## Employees

Please use one of the provided buttons to:

[Add New Employee](#)

[Employee Look-up](#)

[List All Employees](#)

[Clear All](#)

### Add New Employee

Please enter the following information:

Last Name:

First Name:

Phone Number:

 XXX-XXX-XXXX

Address Line 1:

Address Line 2:

City:

CS340 Project Step 7 Final Version  
Group 129 – Staci Ihori, Nathan McAlister  
**Employees Page (continued):**



SEARCH/FILTER an existing employee

# Canine Munchies



Home Page  
Customers  
**Employees**  
Order Details  
Orders  
Products

## Employees

Please use one of the provided buttons to:

[Add New Employee](#) [Employee Look-up](#) [List All Employees](#) [Clear All](#)

### Employee Look-up

If available, please enter the employee ID number:

Employee ID Number:

Otherwise, please enter the following information:

Last Name:

First Name:

Phone Number:  XXX-XXXX-XXXX

Address Line 1:



DISPLAY/UPDATE an existing employee

# Canine Munchies



Home Page  
Customers  
**Employees**  
Order Details  
Orders  
Products

## Employees

Please use one of the provided buttons to:

[Add New Employee](#) [Employee Look-up](#) [List All Employees](#) [Clear All](#)

employeeID	employeeFirstName	employeeLastName	phoneNumber	addressLine1	addressLine2	city	employeeState	postalCode	email	employeeStatus	Update
1	Vincent	Vega	555-555-1000	6324 Pulp St		Corvallis	OR	97330	snuggles@caninemunchies.com	active	<a href="#">Update</a>
2	Mia	Wallace	555-555-2000	9954 Fiction St.	APT #302	Dallas	OR	97338	cuddles@caninemunchies.com	active	<a href="#">Update</a>
3	Julia	Winnfield	555-555-3000	44 Taberock Cr		Corvallis	OR	97330	smooches@caninemunchies.com	active	<a href="#">Update</a>
4	Jamie	Lannister	555-555-4000	322 Hollow Ln	APT #132	Corvallis	OR	97330	nuzzles@caninemunchies.com	terminated	<a href="#">Update</a>
5	Catelyn	Stark	555-555-5000	2222 Huckleberry Ln		Corvallis	OR	97330	nestles@caninemunchies.com	active	<a href="#">Update</a>
6	Paul	Walker	555-555-6000	2002 Skyline Drive		Corvallis	OR	97330	skylinegr@caninemunchies.com	terminated	<a href="#">Update</a>
7	Mark	Maeda	808-456-1654	305 NW 10th St		Corvallis	OR	97330	maedam@caninemunchies.com	active	<a href="#">Update</a>

**Order Details Page:**

CREATE/READ/UPDATE/DELETE Order Details Page

# Canine Munchies



Home Page  
Customers  
Employees  
**Order Details**  
Orders  
Products

## Order Details

Please use one of the provided buttons to:

[Add Order Details](#) [Order Details Look-up](#) [List All Order Details](#) [Clear All](#)

CREATE/INSERT/ADD a new order details

# Canine Munchies



Home Page  
Customers  
Employees  
**Order Details**  
Orders  
Products

## Order Details

Please use one of the provided buttons to:

[Add Order Details](#) [Order Details Look-up](#) [List All Order Details](#) [Clear All](#)

### Add New Order Details

Please enter the following information:

Order Number:

Product Code:

Quantity Ordered:

CS340 Project Step 7 Final Version  
Group 129 – Staci Ihori, Nathan McAlister  
**Order Details Page (continued):**

 **SEARCH/FILTER** an existing order details 

# Canine Munchies

**Order Details**

Please use one of the provided buttons to:

[Add Order Details](#) [Order Details Look-up](#) [List All Order Details](#) [Clear All](#)

**Order Details Look-up**

Please enter the following information:

Order Number:

Product Code:

Quantity Ordered:

 **DISPLAY/UPDATE/DELETE** an existing order details 

# Canine Munchies

**Order Details**

Please use one of the provided buttons to:

[Add Order Details](#) [Order Details Look-up](#) [List All Order Details](#) [Clear All](#)

orderNumber	productName	quantityOrdered	Update	Delete
1	Beef Bites	2	<a href="#">Update</a>	<a href="#">Delete</a>
1	Bibbi-Bobbi-Boo	5	<a href="#">Update</a>	<a href="#">Delete</a>
1	Cheese Please	7	<a href="#">Update</a>	<a href="#">Delete</a>
2	BaconYoMouthWater	30	<a href="#">Update</a>	<a href="#">Delete</a>
3	Cheese Please	2	<a href="#">Update</a>	<a href="#">Delete</a>
3	Pooch Quackers	1	<a href="#">Update</a>	<a href="#">Delete</a>
6	BaconYoMouthWater	33	<a href="#">Update</a>	<a href="#">Delete</a>
6	Beef Bites	1	<a href="#">Update</a>	<a href="#">Delete</a>
6	Bibbi-Bobbi-Boo	2	<a href="#">Update</a>	<a href="#">Delete</a>
6	Danner's Favorite	6	<a href="#">Update</a>	<a href="#">Delete</a>
6	Nutz for Nuts!	4	<a href="#">Update</a>	<a href="#">Delete</a>
7	Pooch Quackers	10	<a href="#">Update</a>	<a href="#">Delete</a>

**Orders Page:**



Home Page Customers Employees Order Details <b>Orders</b> Products	<h2>Orders</h2> <p>Please use one of the provided buttons to:</p> <p><a href="#">Add New Order</a> <a href="#">Order Look-up</a> <a href="#">List All Orders</a> <a href="#">Clear All</a></p>
---	--



Home Page Customers Employees Order Details <b>Orders</b> Products	<h2>Orders</h2> <p>Please use one of the provided buttons to:</p> <p><a href="#">Add New Order</a> <a href="#">Order Look-up</a> <a href="#">List All Orders</a> <a href="#">Clear All</a></p> <h3>Add New Order</h3> <p>Please enter the following information:</p> <p>Order Date: <input type="text" value="mm/dd/yyyy"/> <input type="button" value="▼"/></p> <p>Required Date: <input type="text" value="mm/dd/yyyy"/> <input type="button" value="▼"/></p> <p>Shipped Date: <input type="text" value="mm/dd/yyyy"/> <input type="button" value="▼"/></p> <p>Order Status: <input type="text" value="Select an order status ▾"/></p> <p>Comments: <input type="text" value="e.g. Order cancelled per customer request on 02/02/2022."/></p>
---	---

 SEARCH/FILTER an existing order

# Canine Munchies



[Home Page](#)

[Customers](#)

[Employees](#)

[Order Details](#)

[Orders](#)

[Products](#)

## Orders

**Please use one of the provided buttons to:**

[Add New Order](#) [Order Look-up](#) [List All Orders](#) [Clear All](#)

### Order Look-up

If available, please enter the Order Number:

Order Number:

Otherwise, please enter the following information:

Order Date:  mm/dd/yyyy

Required Date:  mm/dd/yyyy

Shipped Date:  mm/dd/yyyy

Order Status:



DISPLAY/UPDATE an existing order

# Canine Munchies



Home Page

Customers

Employees

Order Details

Orders

Products

## Orders

Please use one of the provided buttons to:

Add New Order | Order Look-up | List All Orders | Clear All

orderNumber	orderDate	requiredDate	shippedDate	orderStatus	comments	contactFirstName	contactLastName	employeeFirstName	employeeLastName	Update
1	2022-02-12	2022-02-22	2022-02-13	shipped		Nathan	Makister			<button>Update</button>
2	2022-02-12	2022-02-22	2022-02-13	shipped		Emiko	Nagashima	Vincent	Vega	<button>Update</button>
3	2022-02-12	2022-02-22	2022-02-01	shipped		Emiko	Nagashima	Vincent	Vega	<button>Update</button>
4	2022-02-12	2022-02-22	2022-02-13	shipped		Natasha	Woodruff	Mia	Wallace	<button>Update</button>
5	2022-02-12	2022-02-22	2022-02-13	shipped		Chris	Vilhauer			<button>Update</button>
6	2022-02-13	2022-02-23	2022-02-14	shipped		Richard	Alaris	Julia	Winfield	<button>Update</button>
7	2022-02-13	2022-02-23	2022-02-14	shipped		Ivan	Gomez	Jamie	Lannister	<button>Update</button>
8	2022-02-13	2022-02-23	2022-02-14	shipped		Megan	Johnson	Catelyn	Stark	<button>Update</button>
9	2022-02-13	2022-02-23	2022-02-14	shipped		Pet	Mitchel	Catelyn	Stark	<button>Update</button>
10	2022-02-13	2022-02-23	2022-02-14	cancelled	Cancelled per customer request on 2/19/2022.	Ivan	Gomez	Mia	Wallace	<button>Update</button>

**Products Page:**

CREATE/READ/UPDATE Products Page

# Canine Munchies



Home Page

Customers

Employees

Order Details

Orders

Products

## Products

Please use one of the provided buttons to:

[Add New Product](#)

[Product Look-up](#)

[List All Products](#)

[Run Product Report](#)

[Clear All](#)

CREATE/INSERT/ADD a new product

# Canine Munchies



Home Page

Customers

Employees

Order Details

Orders

Products

## Products

Please use one of the provided buttons to:

[Add New Product](#)

[Product Look-up](#)

[List All Products](#)

[Run Product Report](#)

[Clear All](#)

### Add New Product

Please enter the following information:

Product Name:

Product Description:

Legacy Product:

Yes

CS340 Project Step 7 Final Version  
 Group 129 – Staci Ihori, Nathan McAlister  
**Products Page (continued):**

productCode	productName	productDescription	legacyProduct	Update
1	BaconYoMouthWater	A perfect blend of salty and sweet, this deliciously soft treat is made with hickory-smoked bacon from pasture raised pigs and syrup from the big-leat maple in the Pacific Northwest. Your dog's mouth will water!	0	<button>Update</button>
2	Beef Bites	Savory, tender dog treats made with real beef and flet mignon infused with a natural smoke flavor.	1	<button>Update</button>
3	Bibbidy-Bobbidy-Boo	These pumpkin biscuits are concocted with fresh organic pumpkins, whole grain flour, and a dash of cinnamon and a dash of fresh ground ginger. Bibbidy-Bobbidy-Boo's magical biscuits will leave your dog wanting more!	0	<button>Update</button>
4	Canine Jerky Chips	Our signature jerky chips are produced using USDA beef raised in the Pacific Northwest. The meat is sliced thin to produce the crispy goodness dogs love. These chips are a great option for dogs with food allergies.	0	<button>Update</button>
5	Cheese Please	We mix barley flour, maple syrup, roasted chicken, and Tillamook cheddar cheese to create this soft chewy treat. Your dog will love it!	0	<button>Update</button>
6	Danner's Favorite	Made with old-fashion rolled oats, wheat bran, blueberries, local honey freshly harvested from the hive. These cookies are a favorite of one of our beloved pups!	0	<button>Update</button>
7	Munchin' Minis	These soft, mouthwatering meatballs are made with high-quality USA sourced lamb, pea flour, maple syrup, sweet potato, and carrots. Munchin' Minis will get your dogs tail moving.	1	<button>Update</button>
8	Nutz for Nutz!	Calling all peanut butter lover! Our fresh ground peanut butter is mixed with sweet ripe bananas and old-fashioned rolled oats to create a blast of nutty goodness that your dogs will love.	0	<button>Update</button>
9	Ocean Blue	Wild Pacific King Salmon makes a splash in this treat. Using salmon from the Pacific Northwest, these fish are combined with premium sweet potatoes grown by Oregon State University Master Gardeners.	0	<button>Update</button>

CS340 Project Step 7 Final Version  
Group 129 – Staci Ihori, Nathan McAlister  
**Products Page (continued):**

DISPLAY all active products with total quantity purchased

# Canine Munchies



**Products**

Please use one of the provided buttons to:

productName	qty
BaconYoMouthWater	73
Carine Jerky Chips	42
milk bones	21
Nutz for Nutz!	14
Pooch Quackers	12
Cheese Please	9
Bibidi-Bobidi-Boo	8
Danner's Favorite	6

CS340 Project Step 7 Final Version  
Group 129 – Staci Ihori, Nathan McAlister  
**Citations:**

Rooney (2022) How to Fetch and Display JSON Data in HTML Using JavaScript [Source code].  
<https://howtoreateapps.com/fetch-and-display-json-html-javascript/>

Pranchal Katiyar (2021) How to convert JSON data to a html table using JavaScript/jQuery ? [Source code]. <https://www.geeksforgeeks.org/how-to-convert-json-data-to-a-html-table-using-javascript-jquery/>

Zeeshan Hassan Memon (2018) How to insert data into MySQL table using Node and express? [Source Code]. <https://stackoverflow.com/questions/49821318/how-to-insert-data-into-mysql-table-using-node-and-express>

Serkan Sendur (2009) Delete all rows from table in javascript [Source Code].  
<https://www.daniweb.com/programming/web-development/threads/113340/delete-all-rows-from-table-in-javascript>