ICS199 final project

**Videogames & more**

Creators:

Zack Bowman

Jasmine Kivari

Nathan Mackenzie

Table of contents

Project Specifications 3 - 4

Visual Usability Guide 5 - 12

Team Challenges 13

Team Lessons 13

- soft skills

- technical skills

Revision for future team projects 13

Sample code we were proud to produce 15 - 16

- (1)

- (2)

**Project specifications / Requirements:**

**Videogames and More Online Store**

**Requirements Specification**

**Version 1.1**

**Revised**

**May 24th, 2017**

**Project Overview:**

The goal of this project is to create an e-commerce web application that is user friendly and easy to maintain for company administrators.

**Purpose and Scope of this Specification:**

The Videogames & More store will be user friendly to navigate with direct guidance

And users will be able to:

- view the product image, name, description, and the price.

- select different products from the website.

- add / remove / empty items in their shopping cart.

- change item quantities and display the contents of their shopping cart.

- login and purchase through secured connection and verification.

Languages that will be used include but not limited to:

- PHP -> communication with the database

- MySQL -> for database

- HTML -> create the webpage / user interface

- CSS -> styling the user interface

In addition, tools used in order for the coordination of the team include:

- GitLab -> local repository for code and documents (Version control software)

- Git -> command line to coordinate code and send to repository avoiding collisions and overwriting others work

**Functional Requirements:**

Bare minimum for the system to be functional:

- database / table creation -> to store customers, products, and cart

- view the product image, name, description, and the price.

- select different products from the website.

- cart functions

- add / remove -> items in their shopping cart.

- display contents of cart -> for purchase review

- display total amount due -> for customer to confirm price

- stripe -> for secure payment

**Non-Functional Requirements:**

Additional requirements: (not required for system to be functional)

- update quantity -> option to purchase more than one item without leaving the cart page

  - empty cart -> discard all items in the shopping cart

- styling the user interface

- cross platform -> availability to mobile devices

- simple UI for ease of use

- 100% uptime -> less interuptions

- quick response -> reduced user frustration

**Precaution:**

This list of requirements is by no means final. The final list will be dictated by implementation constraints and by the end user demands.

Visual Usability Guide



This is just a landing page to our store:

- Click experience greatness to start shopping

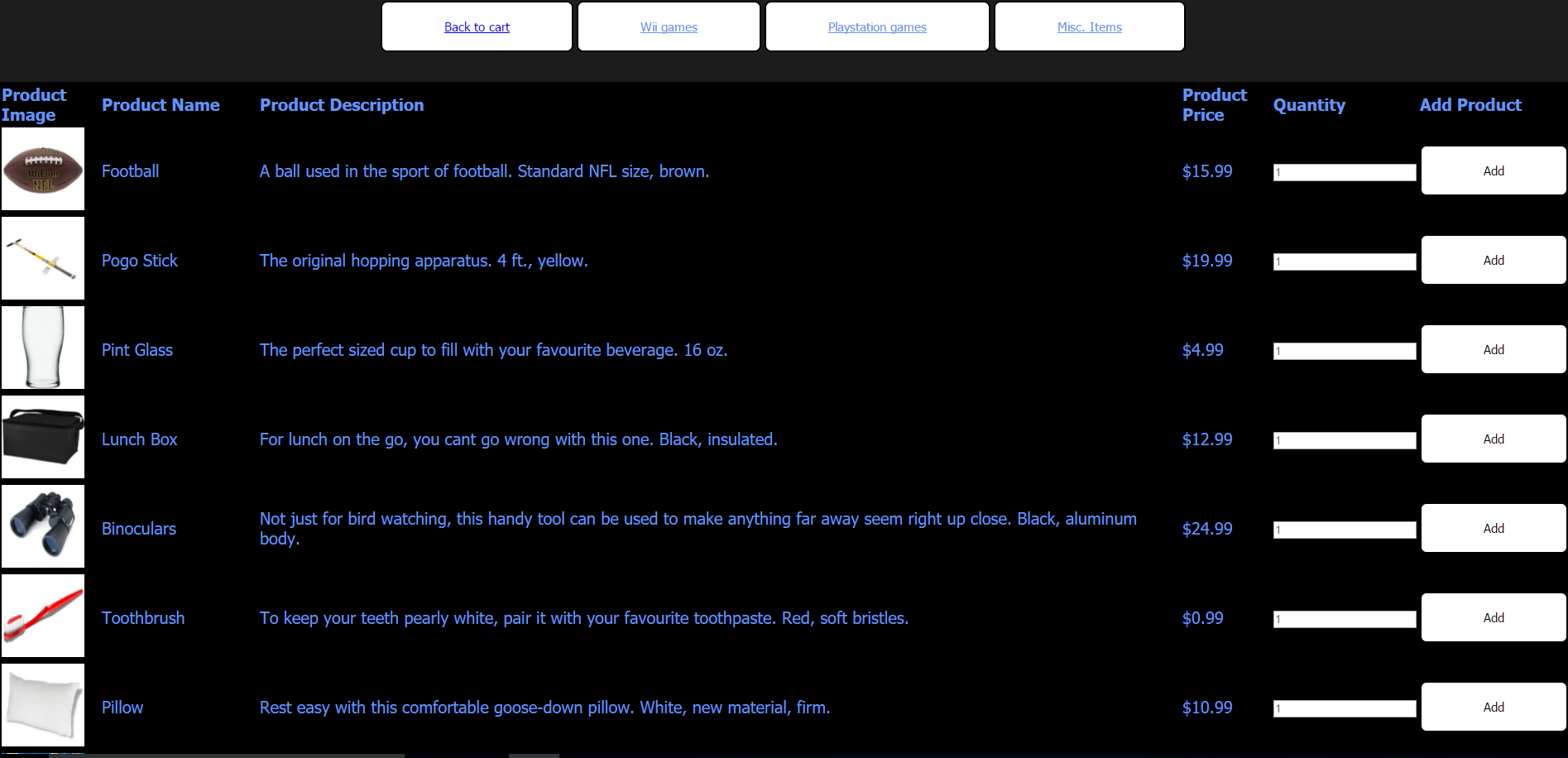


Select products to browse:

- Three option Wii Games, Playstation Games, Misc Items, or all products

to browse

- also contains a link to contact the store



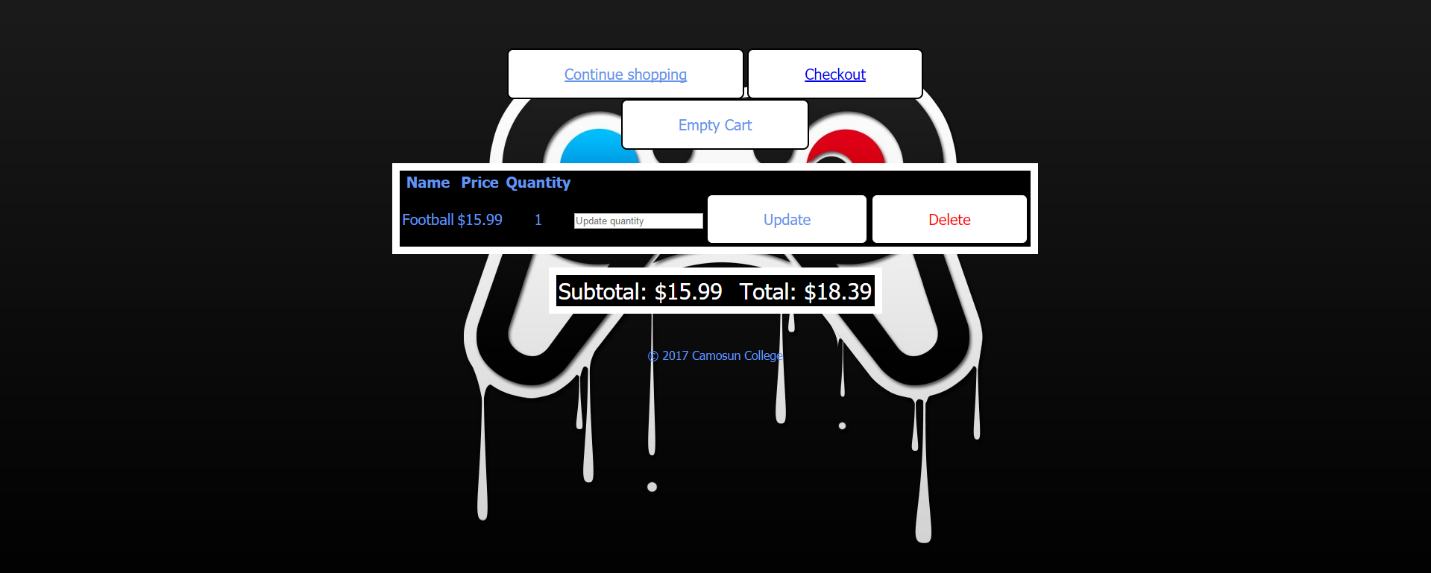
Displays all or partial products based on selected category

- picture of item, name, description, price

- quantity box to allow custom amounts

- add button to add desired product to cart

- links to the cart and other categories not currently viewing



Summary of items in cart with the cost before and after taxes

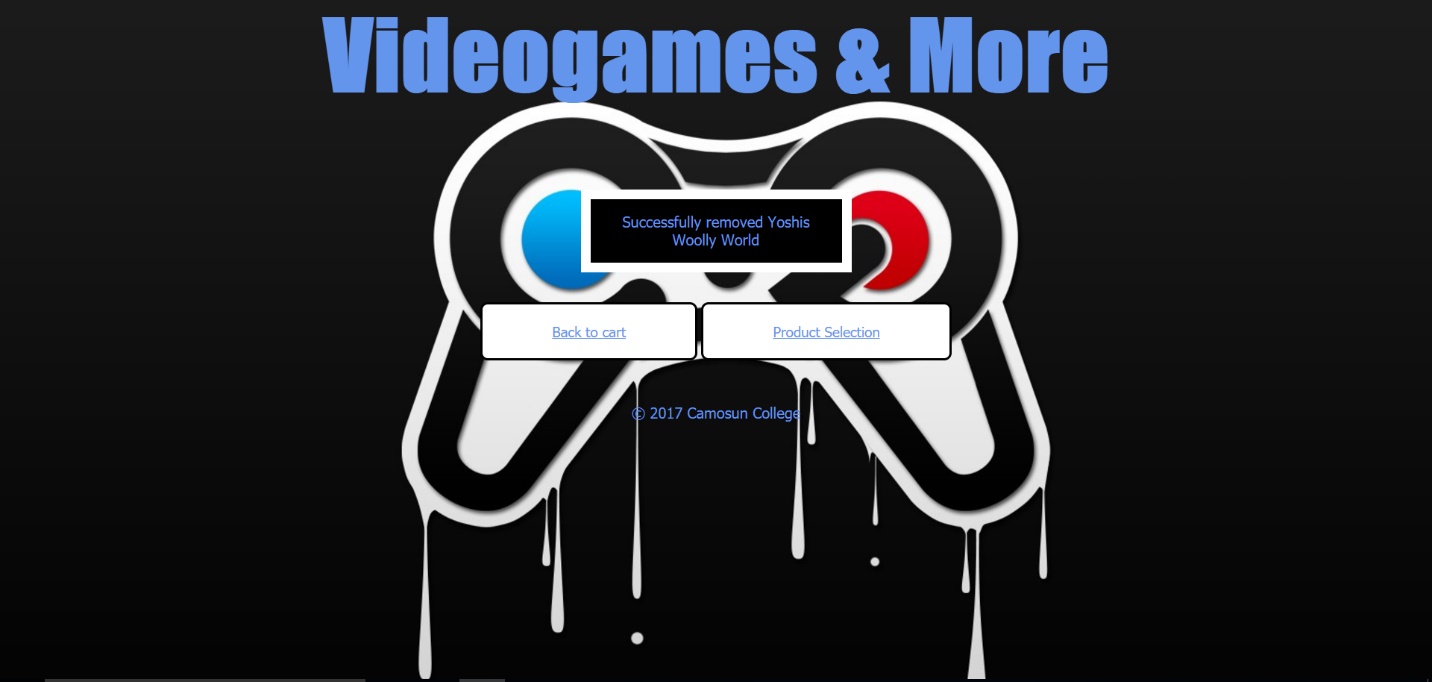
- ability to change quantities / remove products and empty the cart completely

- links to either proceed to checkout or continue shopping



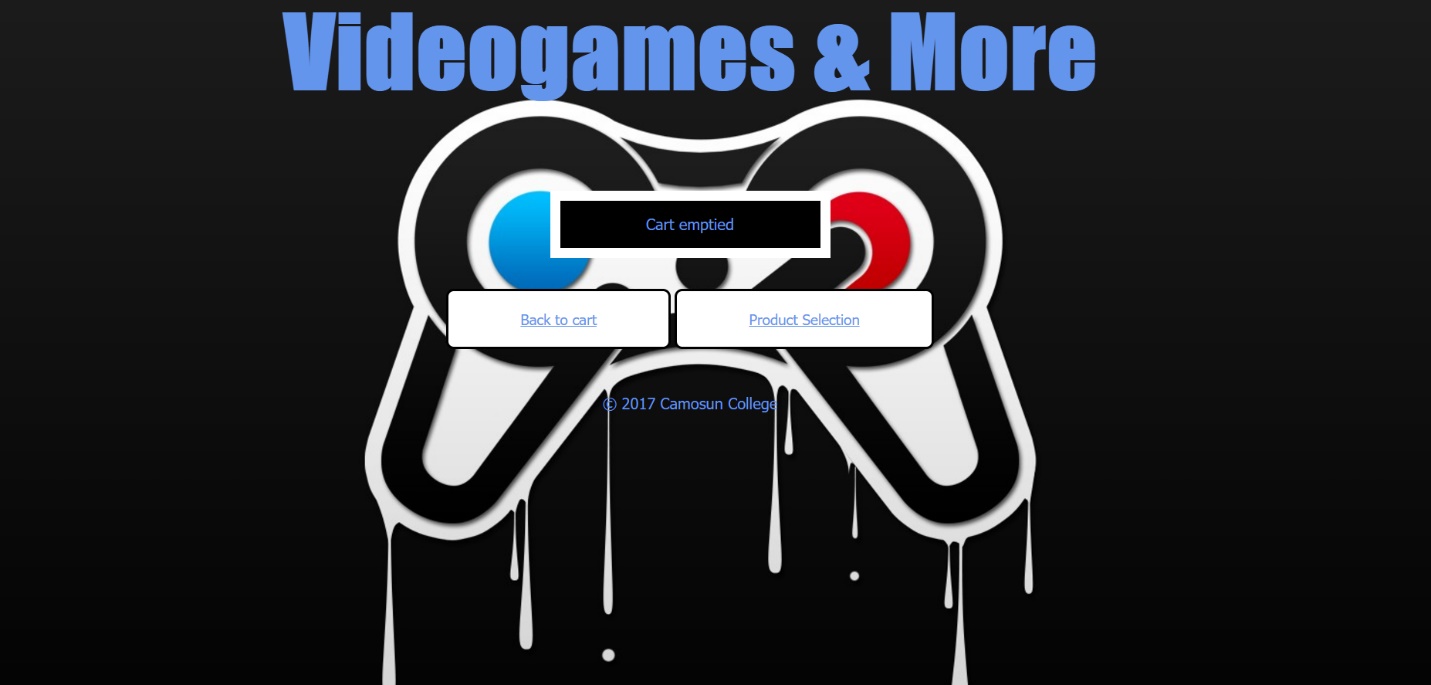
Displays a message to user that the selected item was updated

- links to go back to the cart or product selection screen



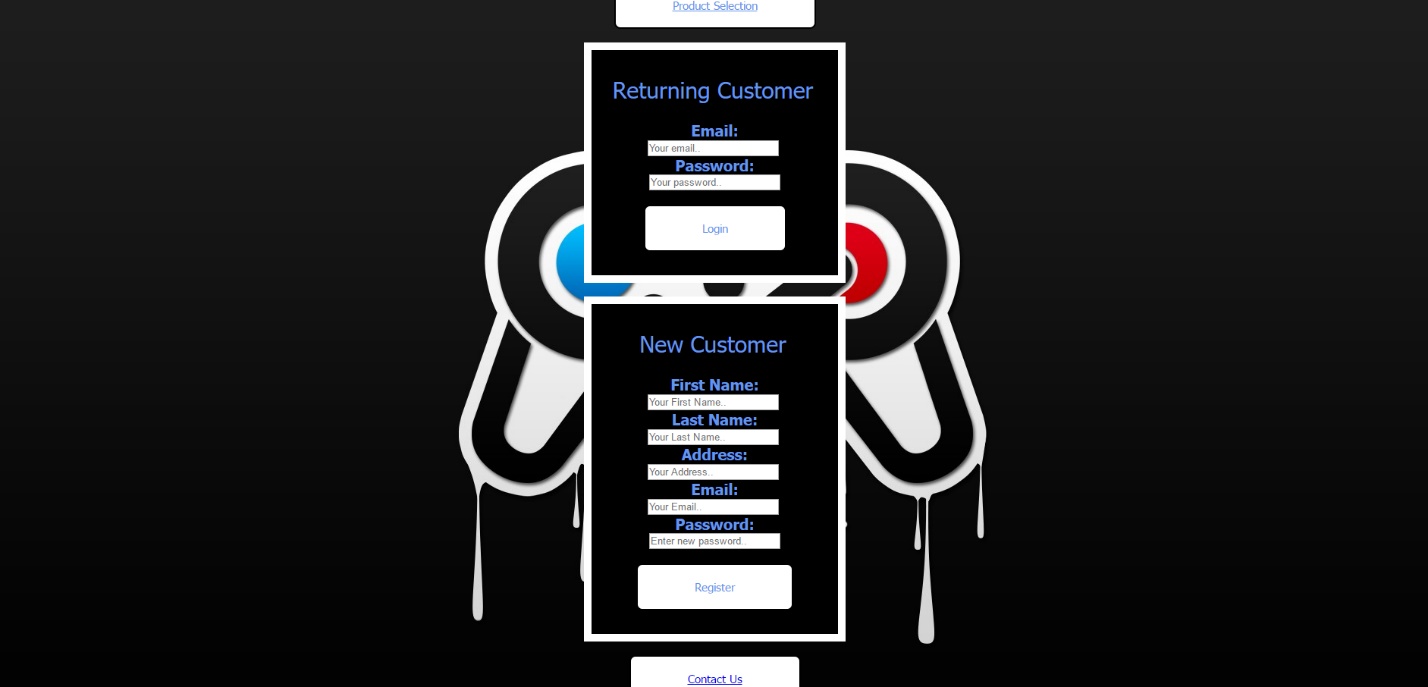
Displays a message to user that the selected item was removed

- links to go back to the cart or product selection screen



Displays a message to user that the cart was emptied

- links to go back to the cart or product selection screen



Form for existing users to login or new users to register

- this permanently adds your cart to a database in the event it needs to be retrieved



If login failed user will be instructed to login again



After successful login users can proceed to payment to finalize purchase



Last time to review selected items being purchase before payment

- button to go back to the cart or continue to payment



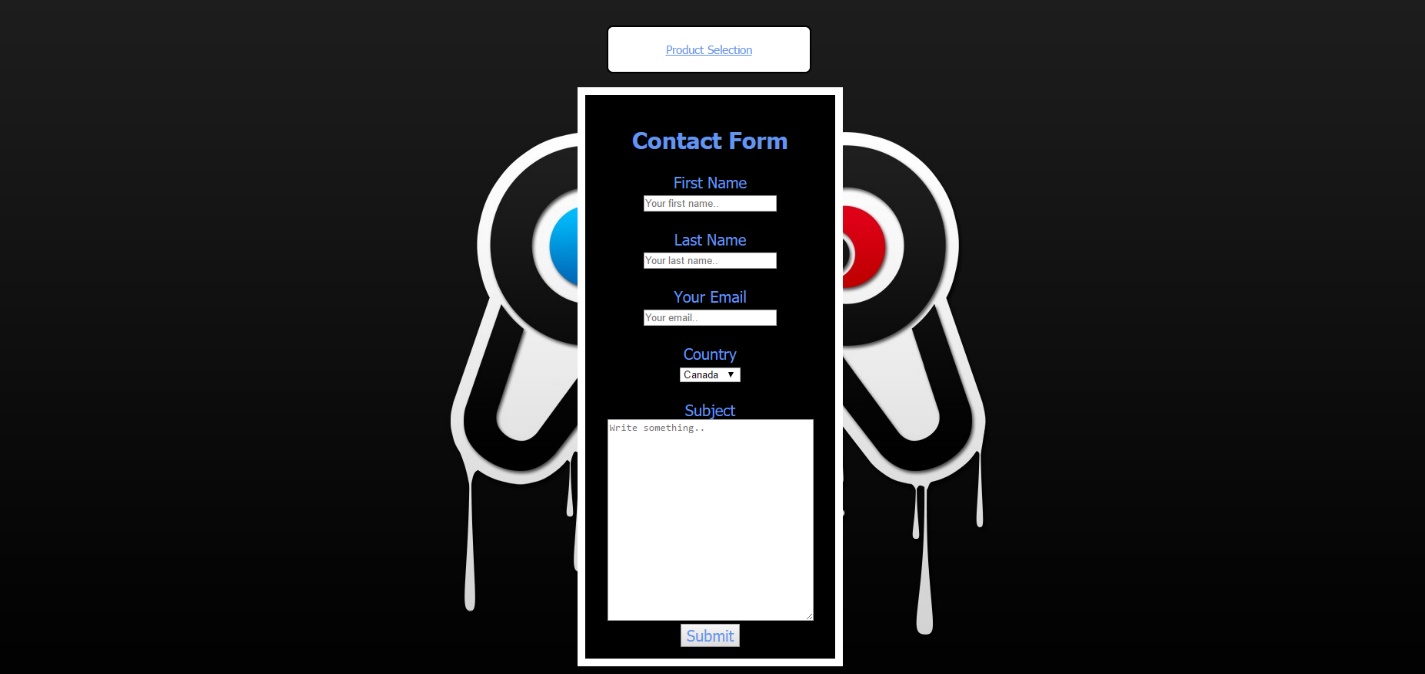
Popup window for user to enter payment details

- final amount due displayed again on the pay button



Notifies user a confirmation email will be sent

- displays the amount charged and a summary of items purchased



For all questions or concerns

- fields for name, country, and space for users to express issues or concerns



Confirms message was received

- confirmation email sent to user

- message sent to store

**Team challenges:**

The largest challenge that our group encountered was with the use of new tools which include Docker, Git and GitLab. This could have been reduced greatly if we had some previous experience. The only other aspect that proved to be a slight challenge was integrating PHP which was a completely new language with very little time to enforce core concepts. These challenges were solved with step by step instruction and documentation provided by the instructor.

**Team lessons:**

**Soft skills:**

The ability to communicate and ensure all team members were allowed to voice opinions without being shut down was a core skill that was put into practice frequently. This in turn allowed all members to collaborate and contribute as much as possible reducing conflict and enable cooperation.

**Technical skills:**

A core technical skill would be the use of version control software Git and GitLab. The use of PHP was a new language that was remarkabley easy to integrate into html was a great learning experience which also lead to reinforcing the importance of a 3 tier architecture.

**Revisions for future projects**

No major roadblocks were hit within our group for this project we all played to our strengths and touched on weaknesses which provided us with a remarkable project in the end. If we were to make revision for future projects it would be to have a more precise gantt chart in order to follow it with greater precision.

**Sample code we are proud of:**

Formatting the mail message to display contents of the cart and formatting based on the theme of our site.

**$mail->Body = '<body style="background-color: black;"><h1 style="color: cornflowerblue;">Videogames & More</h1>';**

**$mail->Body .= '<p style="color: cornflowerblue;">Thank you for your purchase!</p>';**

**$mail->Body .= '<table style=" color: cornflowerblue;"><thead align="left">**

**<tr>**

**<th scope="col">Name</th>**

**<th scope="col">Price</th>**

**<th scope="col">Quantity</th>**

**</tr>**

**</thead><tbody>';**

**foreach ($cart as $item) {**

**$mail->Body .= '<tr style="color: cornflowerblue;"><td>' . $item['product\_name'].'<br></td> <td>' . $item['product\_price']. '</td> <td>' . $item['quantity']. '</td></tr>';**

**}**

**$mail->Body .= '<br><p style="color: cornflowerblue;">You total was: </p><tr style="color: cornflowerblue;">';**

**foreach ($total as $charge) {**

**$mail->Body .= '<td>$' . (round(($charge['total'] \* 0.15 + $charge['total']), 2)). '</td></tr></tbody></table></body>';**

**}**

Our database functions are organized and commented for a clear indication of what they do

**/\*\***

**\* This function populates the customer database and encrypts their password**

**\*/**

**function customer($sessionId, $firstname, $lastname, $address, $email, $password) {**

**global $dbc;**

**$password = password\_hash($password, PASSWORD\_DEFAULT);**

**$query = 'INSERT INTO customer**

**(session\_id, f\_name, l\_name, address, email, password)**

**VALUES (:sessionId, :firstname, :lastname, :address, :email, :password)';**

**$statement = $dbc->prepare($query);**

**$statement->bindValue(':firstname', $firstname);**

**$statement->bindValue(':lastname', $lastname);**

**$statement->bindValue(':address', $address);**

**$statement->bindValue(':email', $email);**

**$statement->bindValue(':password', $password);**

**$statement->bindValue(':sessionId', $sessionId);**

**$statement->execute();**

**$statement->closeCursor();**

**}**