# CS 405 – Final Project Report

**Team Members** – Arthur Davis, Tam Nguyen, Blake Sweet, Tej Patel


**Project Chosen** – Ecommerce / ToyzRus

Development Website:

http://172.31.148.24/Ecommerce-Project/index.html

Staging Website:

http://172.31.148.24/production/index.html

Github Link (for our code):

https://github.com/TejPatel98/Ecommerce-Project

# Table of Contents


**Work Division** **3**
- Tej Patel 3
- Tam Nguyen 4
- Blake Sweet 4
- Arthur Davis 5

**Reason behind choosing this database application** **6**

**Finalized Database Design** **7**
- Final ER Diagram 7
- Database Schema Design 8
- Functional Dependency & Highest Degree of Form 8
- Initial v/s Final Database Design 9

**Finalized Functionalities supported by application** **10**
- Different types of users and their Functionalities 10
  - Customer 10
  - Employee 10
  - Manager 10

**SQL Showoff** **11**

**Things to improve upon** **12**

**Experience with the Project** **12**
- Tej Patel 12
- Tam Nguyen 13
- Blake Sweet 13
- Arthur Davis 13

**Lessons Learned** **14**

**Conclusion** **14**

# Work Division

We tried on making sure that the work is divided fairly amongst the team members. We basically divided the work among each member based on the features required by the project. We did have certain hindrances such as lack of database accessibility by teammates due to permission-issues, the lack of pilot data and the lack of web-development experience; but we worked our way around it and put in our best work.

## Tej Patel

I worked on setting up the initial databases and the relationships. I also worked on the signIn and the signUp pages, and then mainly worked on implementing the features and functionalities concerning the Customer along with the SQL queries relating to it.

I worked on the homepage.php. Given that I had almost zero web development experience before this project, I had to look up the internet and implement the code for styling purposes on our website. The homepage.php takes the customer to the main page where he/she can see all the products the website has to offer. They can search a product based on the categories. They can then add the products that they like to the cart. The code is smart enough to understand that if the product is already in the cart, then it would flash a message saying something like “the product is already in the cart”.

From there once the customer clicks on the cart, the Cart.php. Which is also something I worked on. This php file is responsible for displaying the products that exist in the cart for that customer. The customer can go ahead and change the quantity of the items they want to place an order for, and also remove an item from the cart if necessary.

Once the customer clicks “submit”, they are redirected to the orderPlaced.php. Which is also something that I worked on. From here the customer can either check their Order History, cancel the order they placed in the last 24 hours or logout.

Furthermore I also worked a little on the Manager’s functionalities. The functionalities I worked with were the Sales Statistics/Past Sales and the extra features that were outside the scope of our project requirements. We wanted our project to have added features so I worked on adding them. Hence, my job was to write the code for them and develop database queries for them. The Sales Statistics/Past Sales feature is designed in such a way that the manager gets all the data that can enable him/her to see the sales records in the chosen time frame (either a week, month or year).

The added features for the project that I worked on are *Most Popular Product/s*, *Individual/s with Most Transactions*, *Graphical representations of sales statistics* for the past week, month, year and the *download-data functionality*.

The *Most Popular Product/s* shows the most sold product in a given time frame. The *Individual/s with Most Transactions* displays the individual who has been transacting with the website most frequently in a given time frame. The *Graphical representations of sales statistics* shows a pie chart of the share of each product sold out of the total number of products sold, for each time frame. The *download-data functionality* enables the manager to download data in terms of ids, for the chosen time frame as a csv file. This helps the manager to better analyze and better look into the data.

## Tam Nguyen

I worked on the Manager requirements, located in the Manager.php file. I was helped by Tej, who helped write the SQL queries for several Manager functionalities. In making the Manager page, I used several Bootstrap elements, and arranged them in an Inventory table to make it easier for the user to view.

This Inventory table consists of all products, their names, the quantity of the product, and a place for promotions to be selected.

One of the more important Manager requirements I implemented focused on that addition of promotions to the Manager page, allowing a manager to discount the prices of specific products. I did this by implementing an HTML dropdown with decimal values that I could reference in the SQL query to update the Product table. The chosen value would be multiplied with the existing cost in the Product table, thus discounting the price by 5%, 10%, or 50%.

Once a promotion is chosen, the ‘Update Promotions’ button should be clicked, which sends the SQL query to update the table. A new button will appear allowing users to see the new prices in a pop-up modal when clicked, or the page should be refreshed to see the new prices within the table.

In using Bootstrap, I wanted to make the Manager page as streamlined as possible. I included a helpful message upon visiting the page that references the Manager’s username as well.

## Blake Sweet

Due to issues with MySQL not working on my local or virtual machine, I did not get a chance to program as much as I had hoped. This being said however, I attempted to help my group in their progress of developing all other parts of the assignment. In addition, I worked on the presentation as well as the normalization of the overall project.

# Arthur Davis

A majority of my work was on the Employee requirements specified in the documentation. I also worked on the SQL queries associated with these functionalities. In addition to that, I coded the necessary CSS to improve the front-end view of the site, specifically the Employee and Manager pages. I also helped with the collection of assets like .png files for each of our products and composing the sql queries that would add these products to the site.

After adding the necessary data to the Product table, I started with the development of showing all products with their ProductId, Name, Cost and Quantity. The sql query selects this data from Product table and prints it to a table. The table automatically adjusts its column width based off of the max size of any of its values. Once I got this table completed, I made it so that the quantity of each product was in a textbox and added an update button next to each, allowing the employee to update the quantity once it is clicked. The sql query with this function updates the quantity of the product based off the ProductId.

Next I worked on reviewing all pending orders. When the page loads, pending orders will not show until the user clicks the button labelled ‘View Pending Orders’ to show all orders from the Cart database with the pending status. The sql command natural joins Transaction and Cart and displays the following values: Order ID, Transaction Date, Customer ID, Product Name, Quantity, and Cost.

Naturally, the next step would be to ship these pending orders. I added a text input in which to put the Order ID and a button to submit the order.

Finally for Employee, I added a form at the bottom of the page prompting the user for Product ID, Product Name, Keyword, Cost, and Image Name. Upon submission a sql query adds the values to the Product database and reloads the page.

I also did some work on Manager.php. Having marginally more experience using HTML and CSS, I made changes to the positioning of elements and the aesthetics of the page. Initially the Manager did not have the ability to update the quantity of the product. Initially I tried to integrate the requirement the same way as I had with the Employee. However, I ran into problems because of how the manager product table is coded. It is one large form which allows the manager to update the promotions. So I chose to implement the requirement in the same way. Now when the manager updates the promotions, they can update the quantity at the same time with the same form submission.

# Reason behind choosing this database application

As a team, after looking over both the topics we could choose from, we unanimously agreed working on this. The main reason this application sparked interest in us is because of the abundance of E-Commerce websites surrounding us.

E-Commerce websites are something that have become an integral part of our lives. They act as a one-stop destination for all and any goods we wish to purchase and also helps people save some time. Moreover all team members use such websites to buy goods and have their accounts on them, hence it was more interesting to learn and build the bare bones of something that we use in our day-to-day lives.

From Milestone 1:

Arthur Davis – *"The reason why I, Arthur Davis, am passionate about eCommerce is because I have a prospective job in the field of data storage and visualization after I graduate. Getting the hang of dealing with a customer/business relation will be useful. The information that I learn from doing this will be more applicable than the equine prompt. As someone who has worked in retail before, I also have more knowledge about that than horses."*

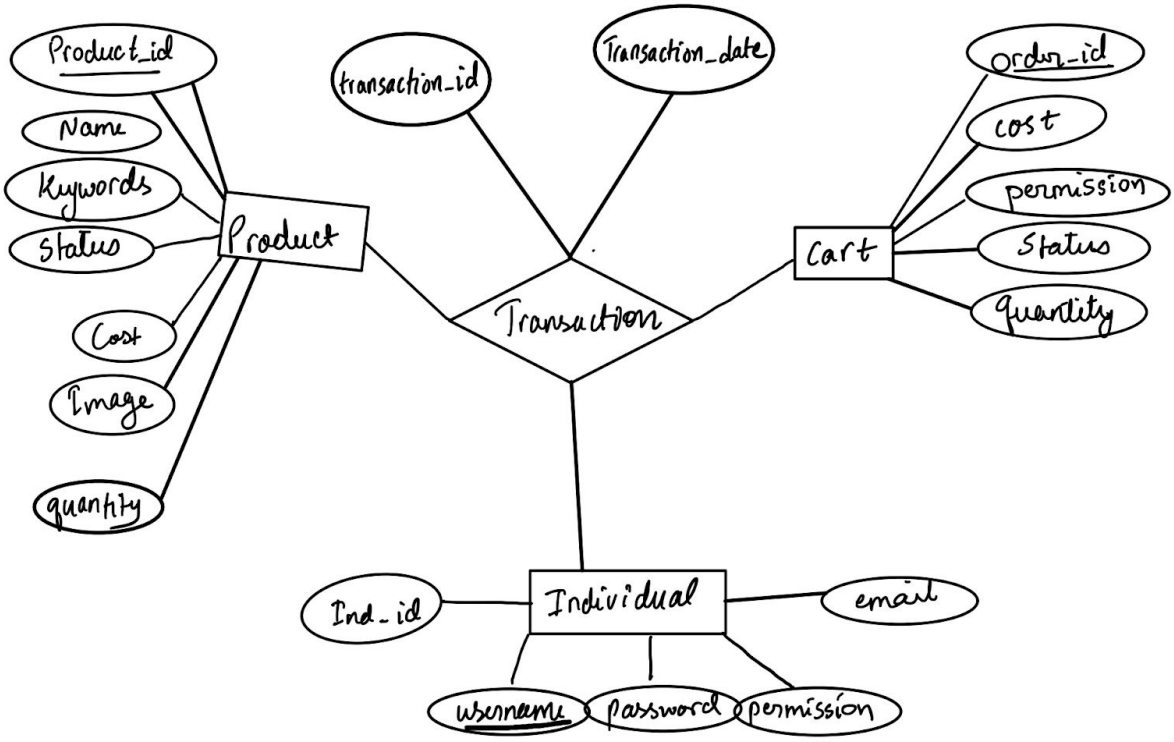
Tam Nguyen – *"I'm passionate about e-commerce because I believe in the value of human-computer interaction. Developing consumer relationships through this e-commerce project will help me better understand the design and construction of databases. In addition, developing a project around how e-commerce is structured will provide insight into the business and consumer relationship."*

Tej Patel – "*I have a strong interest in ecommerce as a field. It is something that exists all around us and I am interested in learning more about it. I wish to gain better knowledge about the processes that go into backend, databases-related parts of the field; and this project would be a good stepping stone for that. Also, I worked on a project that integrated a WebApp and a database, during my summer internship; which is why this project interests me even more. Hence, this would add to the understanding I already have about it."*

Blake Sweet - *"I believe in modern day society that an e-commerce project is universally applicable. As I continue on my path in my computer science expertise, an e-commerce project not only shows me the implications of PHP and HTML; But also real world business strategies and the powerful effects of having a database using SQL. I have enjoyed learning the effects of different SQL data structures."*

# Finalized Database Design

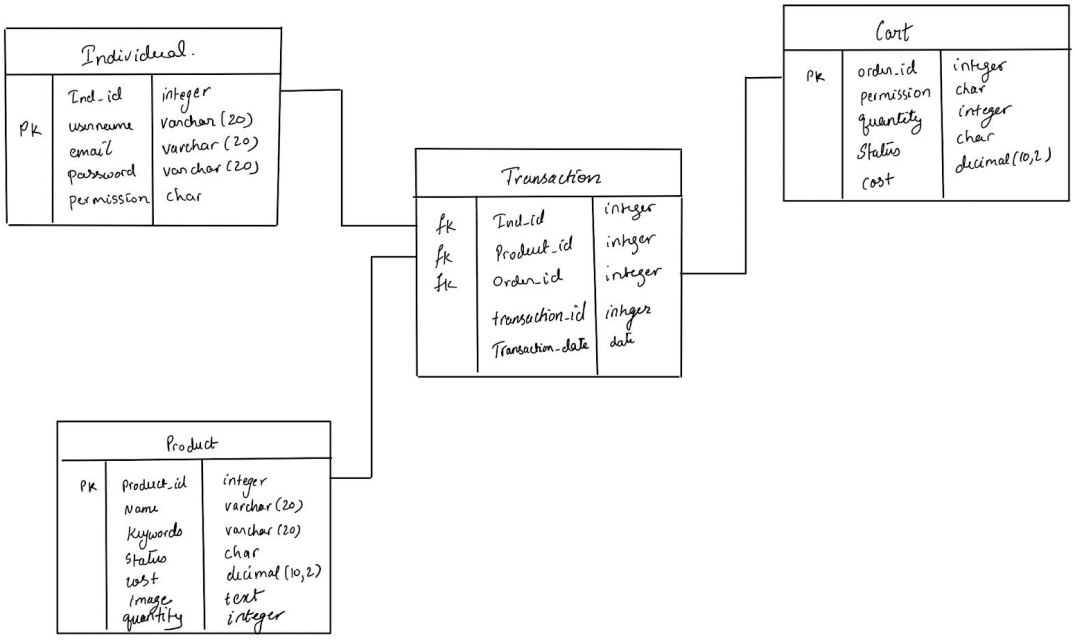
## Final ER Diagram




The above displayed is the final ER diagram that we have set up for our website. There are 3 entities - Product, Individual & Cart. There is 1 relationship - Transactions.

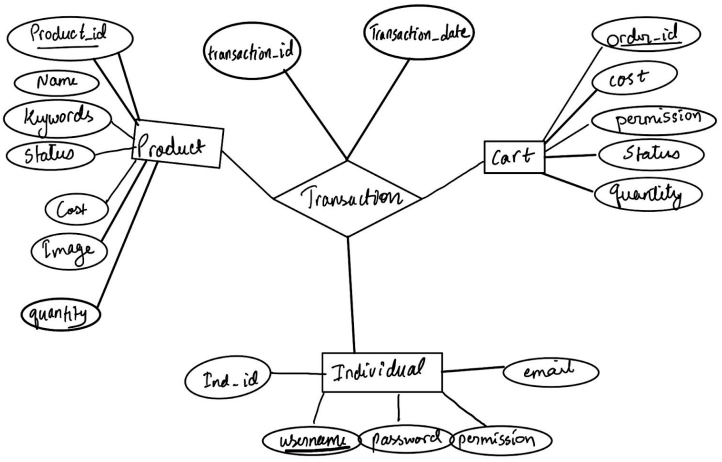
- Product (product_id: int, Name:string, Keywords:string, status:char, cost:float, Image: string, quantity:int)
- Cart (order_id: int, cost: float, permission: char, status: char, quantity: int)
- Individual (username: string, individual_id: int, password: string, email: string)
- Transition (Product_id: int, Username: string, Order_id:int, transaction_id: int, transaction_date: datetime)

# Database Schema Design




The above picture shows our database schema design. The transaction relation holds all the data for when orders are made (transaction Date). Where it relates which Individual with an Username, placed an Order through orderId and for what product through productId. The Individual contains user related data, the Product contains product related data and the Cart table holds order related data.

# Functional Dependency & Highest Degree of Form

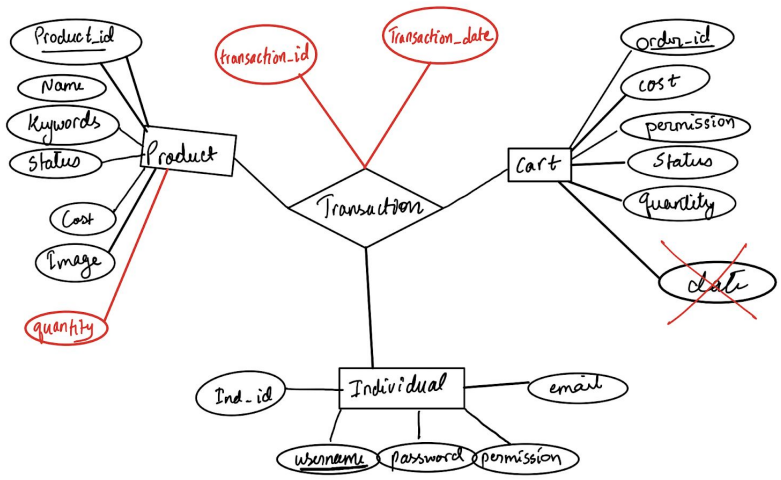


With our design, we were able to eliminate all repetitive data. We also have separate tables for sets of related data. For instance the Product table only deals with the products that the website has to offer, the Cart only deals with the details of the cart of products and the Individual table only holds the data pertaining to each account holder. Each set of related data has a primary key as well. That is why it is in 1NF.

The design sees no need to have attribute values that apply to multiple records. Which eliminates any presence of partial dependency in the design. All the tables are related to each other through the Transaction relation where the primary keys from each table are set as foreign keys. Hence it is in 2NF.

We have each field that depends on a key, hence eliminating any transitive dependency. Therefore it is in 3NF.

## Initial v/s Final Database Design




The ER diagram in black was the original ER diagram design that we initially had. As the time went by, we realized that we needed more attributes for the entities and the relationship. We realized that it will be easier and more convenient to have username as a primary key rather than the individual_id. We realised we need the product quantity. We needed to add a transaction id and transaction date for the sales statistics. We couldn't have order as a table name, so we changed it to Cart.

# Finalized Functionalities supported by application

## Different types of users and their Functionalities

### Customer

The customer is the end user for the E-Commerce website. He/she, in our website, can register, log in, look at all the products, search products based on categories, add them to the cart, place an order of one or more products. The customer can check the status of their orders, keep a track of the history of orders placed and cancel an order within the first 24 hours of placing it.

Functionalities implemented (based off the project specifications from Canvas):

- Register
- Shopping
- Purchase
- Orders

### Employee

The Employee/Staff member helps to facilitate the customers. He/she, in our website, can monitor inventory, add new products, re-stock the products & ship pending orders. Also has a login to perform the above mentioned tasks.

Functionalities implemented (based off the project specifications from Canvas):

- Login
- Add Inventory
- View Inventory
- Update Inventory
- Ship Pending Orders

### Manager

Manager oversees the employees and therefore has certain privileges and has access to certain added functionalities. The Manager, in our website, can do all the tasks that an employee can do. Along with that he/she can can view sales statistics for the past week or month or year. The manager can role our sales promotions, hence reducing the prices of products. We also added out of scope functionalities for the manager enabling him/her to see the top selling product, the most engaged customer, the pie charts of product

sales from the past week or month or year and can download data regarding sales from the past week or month or year or the overall data.

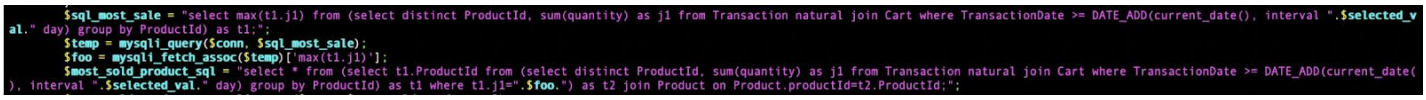
Functionalities implemented (based off the project specifications from Canvas):

- Login
- Add Inventory
- View Inventory
- Update Inventory
- Ship pending Orders
- Sales Statistics
- Sales Promotion

Added Functionalities we implemented:

- Top selling product in the past (week, month, year)
- Most actively purchasing Individual in the past (week, month, year)
- Pie chart for product sales in the past (week, month, year)
- Download sales Data  from the past (week, month, year)

# SQL Showoff



```
$sql_most_sale = "select max(t1.j1) from (select distinct ProductId, sum(quantity) as j1 from Transaction natural join Cart where TransactionDate >= DATE_ADD(current_date(), interval ".$selected_val." day) group by ProductId) as t1;";
$temp = mysqli_query($conn, $sql_most_sale);
$foo = mysqli_fetch_assoc($temp)['max(t1.j1)'];
$most_sold_product_sql = "select * from (select t1.ProductId from (select distinct ProductId, sum(quantity) as j1 from Transaction natural join Cart where TransactionDate >= DATE_ADD(current_date(), interval ".$selected_val." day) group by ProductId) as t1 where t1.j1=".$foo.") as t2 join Product on Product.productId=t2.ProductId;";
```

```
if(isset($_POST['History'])){
    $history = "select * from Transaction T natural join Product P join Cart on T.OrderId=Cart.OrderId where T.Username='".$identification."';"
    $result = mysqli_query($conn, $history);
```

They are two of our most complicated SQL queries. The first one is to find the most sold product over a given time frame (week, month, Year). The second one is not necessarily the most convoluted one, but it is very important since it helps get a specific user's order history.

For the first one we select the number of maximum sales, then based on that number we look up the table which consists of all the productId and number of sales. After retrieving the productId, we use that to look up the product information and display that.

For the second query, we basically natural join Transactions with Product. Then join that derived table to the Cart Table based on the orderId and the customer's username. All of this gives us all the historical data for that specific username.

# Things to improve upon

We would like to improve the formatting and design of our webpages. Since we wanted to ensure the functionality, several elements of our website got misaligned, and while we meet the functional requirements, if we had more time, formatting would certainly be something we would look at.

We would also like to add certain security features. Such as protection against SQL injection. This would make the database and the website more secure from the breach. Currently, on the database side, all the user passwords are being stored as plaintext, and it would be more secure to store an encrypted version of them in a real world scenario.

There are few functionalities that work perfectly but require 2 page reloads instead of 1. The first page reload for writing to the database and the second one to display the correct data from the newly written values to the database. We tried fixing this issue by having the php code removed from the file and having an action set up for thosem, but it did not solve the problem.

# Experience with the Project

## Tej Patel

This project served as a great learning curve for me. I had little experience with Web development. And whatever experience I had did not include such a wide range of functionalities being implemented.

I learnt to program with PHP, HTML and CSS. I also learnt how databases are hooked up to a website and how the workflow is for such real life projects. This project definitely helped me understand how to code for front end of a website. Providing some valuable frontend programming experience.

The most interesting part for me was the database hookup and the database queries. It was very intriguing to learn about how the website and the database worked together and how we could work with the data that we received. Hence providing with a comprehensive backend programming experience.

All in all, it was a great learning experience whether it be on the programming side or the interpersonal relations and the team side. It also provided me with some valuable full stack development experience.

## Tam Nguyen

In this project, I enjoyed creating a webpage and ensuring it both delivered functionality and usability. Working with Bootstrap to create a clean, modern interface was exciting, and I learned much about databases in working with SQL.

What turned out to be especially challenging was setting up a good working environment through the installation of the SQL server, setting up servers for testing, and utilizing version control between team members.

## Blake Sweet

In this project I enjoyed experiencing the implications of an SQL database into a PHP/HTML webpage. Particularly I enjoyed seeing how practical and useful such features can be.

In this project I got the privilege of exploring the benefits of different kinds of databases. For example MySQL vs SQLite. Overall I would recommend a smaller scoped project for the sake of time and clarity, especially since I spent the duration of the project resolving an issue that prevented me from experiencing the lesson intended.

## Arthur Davis

I went into this project with a minimal knowledge of HTML and CSS. I had never heard of PHP. It was a challenge to learn enough to start getting the backend of the site working. The easiest part of the assignment were the mysql queries and connecting to the database.

Having more practice with front end development was useful and I got the chance to work with an external CSS API, Bootstrap. I also got a lot more practice using github and working with a team.

Finally, this project provided an opportunity to put into practice what I have been learning as a computer scientist. Continuous development and and that entails as well as also meeting development milestones and requirements, working towards the demo and presentation of our work.

# Lessons Learned

The team learned a lot during the course of completion of this project, but the major aspects we grew our knowledge and understanding about are the following:

- Programming in PHP, HTML, CSS
- Integrating Databases and Front end
- Team Work
- Team Communication
- Source Control
- Full stack Development

# Conclusion

The final version of the product is something we as a group are proud to present. The only immediate area to improve our project is the styling aspects of the assignment. We could also have implemented more features that were not asked if given more time. The structure was revised to improve as we progressed, resulting in both clarity and efficiency. Finally adding certain security features to avoid SQL injections and data breach should make it pretty close to a real life E-Commerce website.

All considered, it was a learning experience where we took the theoretical side of databases learnt in the class, and used it in the implementation of a real life project. Which gave us more understanding into the practical aspects as well.