Interactive Web App Development

Semester Project Final Report

Noah Mosel and Gage Druen

*Abstract*—A report on the motivation and structure of TechQuest: modern technology supplier website. There are many reasons for creating a site like this. There is the intent to combat bad actors of a profitable industry, as well as the thought of becoming part of the industry ourselves. The structure of our website is focused on the thought that the customer comes first. We need to ensure that their experience is enhanced to ensure that their participation within our company lasts. The structure is focused on the fact that only one piece of the site continues to change. The user also must use an account, this ensures that the session can be continued. There are also many components that are focused on ensuring that data is secure and non-confusing. There are also features for the administration to use to create many products with ease, rather than creating every aspect from scratch.

*Index Terms*—HTML, e-commerce, PHP, SQL, WAMP Server, website, CVV, payment, bounce rate

# Nomenclature

HTML – HyperText Markup Language

PHP – PHP: HyperText Preprocessor

SQL – Structured Query Language

# Introduction

A report about the system of TechQuest; A e-commerce site focused on the idea of reasonable prices on technological goods. A complete look at the software architecture and technologies used. How to keep the customer informed, active, and loyal. The methods to ensure that the website is maintainable and set for the future. What technologies allow for the success of the company and its future endeavors? These are all problems that need to be solved in the creation of our system for selling goods online.

# Motivation

There are so many options to select from when creating a website. We looked at the many online marketplaces to determine that e-commerce is a large part of our lives. This was cemented by the fact that COVID-19 moved many jobs and markets online. This was an option because it allows businesses to continue operating and supply customers with the possible exposure. This motivated us to decide to create an online store for a company, but we still needed to determine our audience.

Our audience was a large part of the decision-making process. Without knowing your customers, you cannot ensure that you create a system that would function for them. We, eventually, decided on the creation of a website and marketplace focused on the current computer technologies. When selecting this category, we realized the potential that the industry had for our theoretical profit. This is due in part to the many soaring prices of the current market. These prices are results of a global chip-storage, scalping, shipping issues, etc. This means that we would have a large demand for the items we would be selling, and we could offer our products at a price that would compete with scalpers, hopefully placing more of those elusive items into the hands of individuals who will use them.

# Website Components

As this website is planned to create a marketplace for the many modern computer technologies, it is our job to ensure that our customers can utilize our site to its fullest. To do this we need to ensure that some basic features worked. The features that are necessary for any online marketplace are landing pages, categorized viewing, product viewing, cart creation, and checkout. However, we needed to ensure that the customer experience was also enhanced by other features, so we implemented accounts and search features.

## Landing Page

The landing page of any website is the most important. It provides the first look at your company for all who encounter it. On average, every website has 15 seconds to catch their audience [1]. If the landing page fails to draw the attention of the user, you will lose that customer and their business. Yet, we need to also ensure that we do not make the landing seem suspicious because that would also light some red flags of the potential users. If they are unsure of the trustworthiness of the website, there is little to be done to ensure that they stay active with our business. Yet, even if we do this well, we need to ensure that our site stays responsive.

An unresponsive website will drive an already impatient customer base away. We are so privileged with the current technologies of our time that we do not need to wait for most applications to fully load before we utilize them. Most consumers will not wait much longer than three seconds before abandonment of a website occurs [2]. This means that we need to ensure that our system can function to load within three seconds or else we will face high bounce rate issues. To ensure that a page loads quickly we can try to utilize fewer large resources and try compression of resources that are necessary to the experience. On top of this the use of asynchronous calls ensures that the website can still load as the user waits for the data that they may soon require.

These requirements were looked upon when considering what our landing page and website would consist of. The final solution came to the use of framesets to create a header, and body of our website. In using this the landing page is the frameset page, which then loads the header and main body files.

The header that gets loaded here is then useable throughout the entire session of the customer. This is then loading with the buttons that would allow our consumers to get to where they need to be. This includes buttons to access the home page, cart, and account information. On top of this, the header contains the logo for the website and the product search bar.

The main page utilized in the frameset will be updated as the user progresses through their session. As they open each new webpage, they will be greeted with the same layout throughout the site, but when initially starting their session, they will be greeted by the featured products of the website. This is a list of products that might be on sale, great deals compared to competitors, or just products that we may want to show off to potential buyers. Yet, this is not the only aspect of this main page.

The home page here also gives a side navigation window that allows the customer to search through a category. This is done by calling a PHP script that first looks to find all available tables within our database. This then outputs the necessary HTML code to be displayed as a clickable link to the user and places this within the navigation bar.

These links in the navigation bar will then create calls to other scripts when clicked on. These scripts will pull the table to be searched from the button and display all products within a divider. This divider then contains all the information that would pertain to that category. Each item that gets displayed will contain the brand name, item name, price, and a brief description of each item. Included with this all items, when clicked, will redirect the user to the product that they wish to visit.

## Products

When redirected to the view of a product on our website, the user will be shown an array of information. This includes the brand names, item name, price, and full description of the item in question. This is done by calling a set of code that queries a database with the entry of the product name, to select the necessary set of data. This then selects from the table of where this product exists to receive the data values located at the index of the item. These are then sent to the user and displayed in the correct locations on the webpage. However, the user will also be shown a preview of the product.

The area that provides the preview images allows the user to click a button to change which image is being viewed. This is done by utilizing a call to a script that takes an input of the item. From there, the script searches for the folder that contains the image files. These will then be sent to the user’s system. These images will be placed into an array, which then can be referenced at an index to grab the required image. This index can then be updated using modulus to ensure that the user can continue clicking the same button and continue receiving an image until the end of the array, at which point the selection moves back to the beginning of the array and vice versa. This ensures that the customer knows what they may be purchasing before they add the object to the cart.

The features to allow for cart additions are also shown here. This being the quantity input, as well as the button to make a call to the “addcart.php” script. When the user utilizes the feature to add to cart, there is a check to determine whether the user is logged in.

When logged in, the call to the script allowing for the creation of retrieval data will take inputs of account number, item name, and quantity. These will then be stored within a table of the database alongside the addition of category for the item.

If there is no account logged in, the system will prompt for the login information in a pop-up. If the user has an account, they can log in and continue to add items to their cart. This function will utilize the same checks that are seen on the login page. Otherwise, there will be no call to create a row containing necessary information for retrieval. There is also an option to create an account, if necessary, which will redirect the user to the sign-up page.

## Accounts/Login

On the sign-up page, the user will be asked for basic information such as username, email, and password. They will also have to enter the password a second time to confirm that the user knows what they have entered as a password. If all inputs are valid and not null, and no other account with the same username exists their account will be made. They will then be redirected to the login page.

At the login page, the user is prompted with two inputs: username and password. When valid inputs are created, the user can click the login button for a call to be made which will be requested from the database for the corresponding location within the accounts table. If the corresponding information does not exist or does not match, the server responds with an error to show that the login failed. It will tell the user what might be wrong in the situation that just occurred. So, in the case of incorrect information, “the username/password is incorrect!” would be shown.

When logged in the login button will redirect to page that contains basic information about the current user. This will include username and email of the current account. On top of this, the page allows for the ability to log out of the system. This will clear the session data and redirect the user to the homepage.

The creation and login of an account is a necessary aspect of the utilization of our system. This was done to ensure that we can keep the data that passes through our system to stay secure and allows us to ensure that our customers are able to leave our site and start off where they began without the use of storing information on their system. This means that we can ensure that each customer has their own cart, and that even a large cart can be stored, as we would have the location and resources to keep this data on hand.

## Cart

The cart is a primary part of any system that allows for the purchase of products. The cart should show the customer the total price of all items within. This should also show the customer a final preview of their potential order and allow for changes to occur. These were all taken into consideration when creating our cart viewing system.

When the user clicks on the link to view their cart, they will be prompted with the same login pop-up as seen when adding to their cart. If logged in a query will be created that will request for all items located under the users account number.

These will then be displayed in a similar fashion to the items within the product page. However, each item will only be viewable by clicking the image, and there will be a button to remove all the items of a type to be removed from the cart. There will also be a display of the total cost of each set of items. So, ten items at $30.00 each would total to $300.00. These totals would then be totaled and displayed in a side-pane which would include the total price for the cart and total number of items within the cart. There would then be a button that would allow the customer to head to checkout.

## Checkout

The checkout is a feature that allows the customer to enter all information pertaining to payment method and shipping address. This being considered, was created as the main part of the checkout process. There are tables set up to take in the input of payment details and shipping information. These are visibly separated with borders to avoid confusion with the customer.

The payment details section allows for the input of card number, CVV, expiration date, and name on card. These are checked to ensure that no input is left blank on the attempt of order, however, if we did have a system to process real payments, there would also be a check to ensure that the payment can occur. This would be done by checking for a valid card number, as well as ensuring that the rest of the inputs match the card’s information.

The shipping address section allows the user to be able to receive the products that they order to the correct facility or location. This is validated to ensure that the necessary inputs are not null. If the site did exist, it would connect to a dictionary to ensure that the location is a valid shipping address. This would match the zip code to each city/state group.

When all the inputs are validated, the order will be allowed to continue. When this does occur, a prompt will display whether the order was successful. If successful, the charge would occur, and cart emptied. When the prompt is then left, the customer is redirected to the home page.

## Search

Search was one of the features that was implemented to be an advanced feature. This occurred because it is not necessary to the system, however, it would enhance the user experience by allowing the user to search for items that pertain to exactly what they are looking for.

This is done by searching through every table of products to determine the items that may be related to the search. All items’ names, brands, and descriptions will be read to determine relation. If the search term is found the item will be sent to the user’s session to be displayed. Any search term that is like that of a table name will call for all items within the table. Each item utilizes the same display system as seen in the category searches. Also as seen in the same section, when the divider for each item is clicked on, the product page will be displayed so that the user can select the item for their cart. The search is also not cleared to allow for the same search to occur, if needed.

## Admin System

The administration system can be accessed through the browser by utilizing the admin account. When logged in, the admin can access a page that allows for the addition of products to the system. This is available within the “Profile Details” page and makes it much easier to add the distinct items to our list of website products.

The “Add Product” page will give the administrator a page that contains inputs for category, brand, item, price, description, and images.

The category input is updated with all available categories to the system, as well as an option for adding extra categories. When this option is selected, another input appears, allowing for the input of the category name.

The images section includes an input for an image that we be set as the primary image. The remaining input includes the ability to set as many images as the admin likes. This allows for many images to be saved to the folder at once. When these inputs are changed, a function is called that places the images into an array and updates the image preview section with the images. This allows the admin to view the images that they are uploading and ensure that nothing is out of order.

# Backend System

The system that allows our website to work includes the use of WAMP Server. This is a French developed software that makes the process of running a webserver much easier and friendlier to the user, and business. This includes much of the work to create the database, run php code, and ensure that all files can run successfully.

## Database

The database can be instantiated utilizing MariaDB or MySQL. We chose MySQL due to convenience. This database service, as said before, was setup by WAMP Server automatically, but we did need to customize this. Having little experience with databases, it took time to figure out what was to occur to create the features we needed.

The sets of databases that were created were products and users. These two, hold product information and user information, respectively.

The product database is then split into tables that correspond to the categories of the products. Each table has columns for brand, item name, price, description, discount, and discount toggle. Each item stored within these must include a value for brand, item name, price, and description. Otherwise, the discount was a column setup to allow for the utilization of discounts if that feature had come along.

The user database is split into a table of accounts and a table to hold all cart information. The account table holds information such as username, email, password, and account number. The cart table holds all information for the carts of the users. This information is stored utilizing the item name, category name, and user’s account number. This allows the cart to search through only the cart inputs which have the corresponding search account number.

## PHP

All PHP installation is covered using WAMP Server. This allows us as a company to ensure that the PHP scripts can be utilized to their fullest potential. If these did not run it would be difficult to supply our users with the information, as well as services required to complete their orders. This would require other methods to retrieve and save data efficiently.

## File Structure

The file structure of the webserver involves the use of a main folder with all general files. Within this folder there are subfolders “Scripts,” “Products,” and “Images.” Each of these stores the corresponding information.

The “Scripts” folder contains all PHP scripts that allow for backend functions to occur. This folder also includes a template file to be used to create each page for the products. The “Images” folder contains all images that are used for multiple or general locations throughout the website.

The “Products” folder contains folders for every category. For example, folders named “cpu” or “gpu” sit within here. These category folders then include folders using a “BRANDItem\_Name” naming scheme. Within these folders is a HTML file titled with the item name, and there are also image files created utilizing the ”webp” extension. These are numbered from zero to the total number of images minus one.

# Learned

There has been much learned throughout the creation of the project. It proved very difficult to create the website. There were many features that we had to create. These features are quite reliant on each other.

## Planning

The creation of a website requires a lot of planning. This is something that our group lack a lot of. If we had done more planning, we likely could have implemented more features.

Planning would have told us that we needed to decide how file structure was defined and what services were required to ensure success. We would have realized the need to utilize a database and would have been able to focus on our features in the correct order.

Planning would have allowed for us to creating many features and ensure that they stayed useable throughout the creation of other features later. This would have meant less work to ensure that all features continued to work as they were expected.

## Products Pages

Without creating the feature to add products to your page, you must be consistent on creating each website. This must be done by yourself each time. If we would have focused on creating this aspect of our website first, we would have been able to create many more products with ease. It would have also allowed for us to make the footprint much smaller.

If we would have known more about the needs of a website that does as much as our does, we would have been able to utilize “PHP” scripts more often. This would have meant less text stored on the backend of the server. We could have just had folders containing the images for each product, and that would have been sufficient. This would have allowed for some scripts to be smaller as well.

The creation of the database also helped much. This took a fair amount of time to utilize, but when we did realize how necessary the database was, it was a feature that had to be used. It cut down on a lot of work. This was able to store all the information for every product, so if we had utilized the “PHP” scripts more, we would have only needed a single script. This script would then have contained a variable that would be set by the call to the item, but then would have placed the response within a divider that would have been created on the main page. This would have meant pages would have to change, and that the frameset features would not be required to have similar functionality. This feature is likely to be long gone since it is deprecated.

## Accounts

The account of our website are not secure. It was a feature that would have been nice to include, but the work to due such would have been difficult. There was thought to encrypt all passwords that were stored during signup and decrypt each when we looked to check for valid login information. This would have been challenging, but for our use case not necessary.

We did learn that the use of “POST” allowed for many improvements for the security of the site. “POST” does not operate through the “search bar” of the browser, so the data would not be available to a threat that may be looking at the data that is transmitted through those means.

The account feature was not really expanded on as much as we had wanted, but it was also difficult to decide what was necessary information that needed to be saved. We learned that many websites give users a login token. This token is then used to save session data. Instead, we sent the user the account number and saved that in the session storage of the browser. This meant that if a knowledgeable user looked at what was saved, they could get their account number. This would also mean the same issue for security, as anyone who had known about this could change the account number that they were using to alter data of someone else. If this was known at time of creation this would have changed.

## Images and HTML

Images are an amazing thing to be seen on websites. It is amazing that we can request for the data, and have our eyes exposed to beautiful images, but it was difficult to advance the basic features that are given using HTML.

HTML offers a way to add images to an object, however, we wanted to give the user a way to go through multiple images of each item. To do this, we had tried to call all the images when the page loaded. This functioned if we reloaded the frame at each attempt to view products. We then altered the method and tried storing images within an array. Unfortunately, this had a similar issue.

We then found that we could use a script that would call each image from the folder. This allowed for the webpage to receive the images and finish loading. Then in-turn the images that were received we save to the cache of the browser. These allowed for the array method to function, as each image was able to successfully go to the webpage.

The information that we had learn regarding images was then useable to create the image preview for image of products being added to the page. There was other stuff to learn here, but thankfully we were able to reuse some knowledge. When displaying the images, we needed to determine a way to pull the files from the uploads, these would then need to be placed in an array as before. From there we could display each image we little effort, due to previous experiences.

## Search

Search was implemented much sooner than expected. The creation of this feature occurred since we were unsure how we could pull the necessary information for each product.

Initially, searching was done through the utilization of looking through the file structure. If the search was able to find the related information, it would then display basic information to the user. This had issues because it would not go through the names of files completely. This also did not pull items that had descriptions that matched.

Through the implementation of the database, we were able to get the feature functioning considerably well. We were able to do a search through all columns of the tables to find information that the search term was part of. If this was found, then the information would then be displayed. This had no known issues and allowed for the search to perform very quickly.

# Work Log

## Week 1 (26SEP21-02OCT21)

The main page layout was figured out. This included what colors would be used for the backgrounds of the website. This also brought about the creation of the logo and other images for the buttons.

## Week 2 (03OCT21-09OCT21)

Product pages were designed. These were mainly just getting the placement figured out. This was useful in getting many features added.

## Week 3 (10OCT21-16OCT21)

Accounts were designed at this time. This was a time that we realized we needed to have a database for much of our information. The database was created, and the account table was structured.

## Week 4 (17OCT21-23OCT21)

This time comprised the creation of sign up and login. Without this we would have no way to ensure that customers had the correct carts. This allowed for many features to come about.

## Week 5 (24OCT21-28OCT21)

The admin account was created. This allowed for the ability to get to the page to add products. This time also created a basic method of adding products.

## Week 6 (29OCT21-04NOV21)

Many touch-ups came along during this week. It allowed for many of the features to get much easier to use and fixed most bugs that existed at the time.

## Week 7 (05NOV21-12NOV21)

Search was developed through this week. It took some work, but we knew how we could look for specific items within the database at this point.

## Week 8 (13NOV21-20NOV21)

Very little occurred this week. There were some style updates.

## Week 9 (21NOV21-27NOV21)

A lot of progress came about this week. The cart was created and functioning very well. This had some updates in the user interface that needed to occur, but overall functioned. Another feature that got a big overhaul was the add product. It received much of its backend functionality. This then cemented the file structure of the webserver.

## Week 10 (28NOV21-04DEC21)

Payment and checkout came about during this week. This allowed for the user to pay for items if the system had that functionality. On top of this, the cart received updates to the interface and allowed for the removal of items from the cart before checkout.

This week also brought about the creation of the main page categories. This allowed for the user to look at one table of products at a time. The items within these table were also designed to show in a pleasing way, which in turn got utilized for the search functionality as well. This ensured that the webpages were resemblant to each other.

## Week 11 (05DEC21-09DEC21)

This final week was filled with the touching-up and ensuring that all features work as expected. This also involved the creation of many products on the website as all structure was done at this point. Final styling and other interfaces changes occurred here, as well.

# References

[1] David Zheng.” The 15 Second Rule: 3 Reasons Why Users Leave a Website.” The Daily Egg. [www.crazyegg.com/blog/why-users-leave-a-website/](http://www.crazyegg.com/blog/why-users-leave-a-website/) (accessed 12/8/2021).

[2] Maura Monaghan. “Website Load Time Statistics: Why Speed Matters in 2021.” Website Builder Experts. [www.websitebuilderexpert.com/building-websites/website-load-time-statistics/](http://www.websitebuilderexpert.com/building-websites/website-load-time-statistics/).