Best Buy Project Final Report[*]

Kyle Hash
University of Idaho
Moscow, Idaho, USA
hash1886@vandals.uidaho.edu

Conner Mullins
University of Idaho
Moscow, Idaho, USA
mull7488@vandals.uidaho.edu

Triston Hardcastle Peck
University of Idaho
Moscow, Idaho, USA
hard7263@vandals.uidaho.edu

ABSTRACT

This document's focus is to inform you of the project we have been working on in CS 360. Our goal was to make a web service that would match products and services to the needs of a customer. Vendors will be able to post their products and services and sell them to the customers. This was accomplished using MySQL, PHP, and a few other web development tools.

Brief Disclaimers: The words "user" and "customer" are used interchangeably below.

CCS CONCEPTS

• **Information systems → Database query processing**; **Query languages for non-relational engines**; *Data model extensions*; Relational database query languages; • **Computing methodologies → Natural language processing**; **Logic programming and answer set programming**; • **Applied computing → Bioinformatics**.

KEYWORDS

MySQL, W3, PHPMyAdmin, HTML, JavaScript, CSS, Bootstrap, products, users, customers, vendors

ACM Reference Format:
Kyle Hash, Conner Mullins, and Triston Hardcastle Peck. 2022. Best Buy Project Final Report. In *Proceedings of December 16, 2022.* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

This document's focus is to inform you of the project we have been working on in CS 360. Our plan was to make a matching database in which users will let us know what their product and service needs are, and our website will attempt to find the closest matches to said need. On our website, customers will be required to create an account with a username and password. After they create an account, they will be taken to our home page where they can see all of the different products and services offered. Selecting a product or service category will bring them to a new page where they can see all products within said category. The user can then specify exactly what they want using a system of checkboxes. After the customer puts in all of the information that they want to search for, they will be shown products that match their search. The user should see two categories: Exact matches, and similar matches. If they decide to purchase an item, they will be redirected to the checkout page where they can enter their payment information and shipping address.

Vendors will be able to list their products and services to our website. When registering a product or service, they will be required to enter all of the details that appear on each type of product or service. When a user is looking within a product category, they will see all of the details that were entered by the vendor.

There is a binding contract for all users in regard to products and services listed on the website. Upon registering an account, the user must select a checkbox agreeing to the contract within the terms and conditions of service. They are able to see this contract at the bottom of the customer and vendor dashboard.

1.1 Design

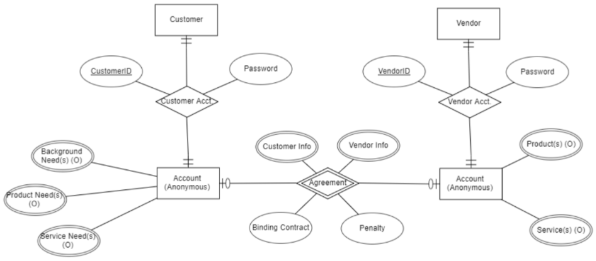


Figure 1: Original Project Diagram Design

The original design for the project diagram was created before we had much direction in terms of how we planned to create the project. We had little experience in both creating diagrams and . There is no key or legend for the symbols. The information that is stored in the accounts is the bare minimum that we believed every account would have. Our initial interpretation of the project was a website where you could create an account, store your background needs, products needs, and service needs within your account information, then that would be used to find products to fulfill said needs. In order to purchase an item from the vendor, the customer would be required to agree to a binding contract. If said binding contract was unfulfilled, the party that broke the contract would receive the penalty listed within the contract. The plan for our project has significantly changed since then.

[*] Any acknowledgement you may want to make.

December 16, 2022, Moscow, Idaho,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

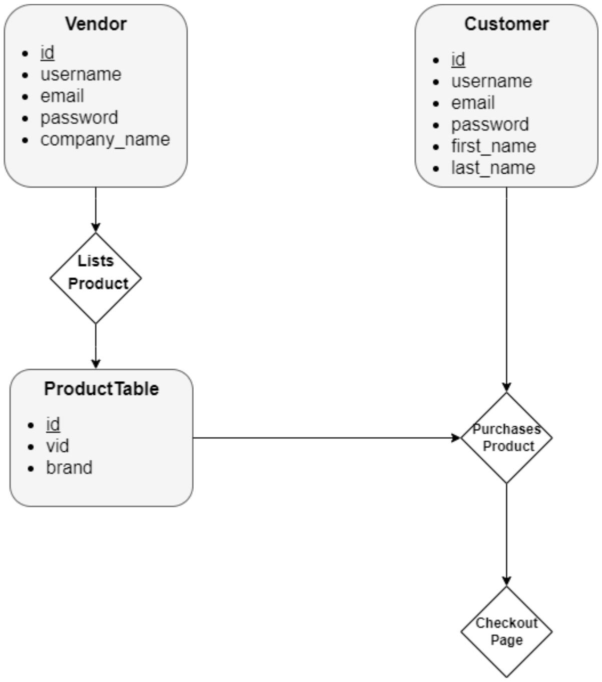


Figure 2: Revised Project Diagram Design

After we had fleshed out much of the project, we decided to redo the diagram. It was inaccurate to the actual working product that we had produced. To start, we simplified all of the symbols that are used. There are three main sections within the diagram: The Vendor, Customer, and ProductTable. The Vendor refers to a vendor account. Each vendor account will have a vendor ID (VID), username, email, password, and company name. Next there is a Customer, which refers to the customer account. Their account is comprised of an ID, username, email, password, first name, and last name. Finally, there is the ProductTable. The ProductTable is a placeholder for a table that holds products, although we do not have a table named ProductTable. This was done to improve readability of the graph, rather than listing all of the tables that are actually used to hold products. The ProductTable is comprised of an ID, a VID, a price, and any other information that would be included for that type of product.

The vendors have the capability of listing a product to a table, which can then be viewed by the customer. When the customer decides to purchase a product, they are taken to a checkout page where they can fill out their purchasing information. Once that is complete, the product will be "purchased" and removed from the corresponding table.

1.2 Tools used for development

There were many software tools used in order to develop our Best Buy Project website. XAMPP was used to locally host our server and our MySQL database. PHPMyAdmin was used to manage our database and tables throughout the course of the project. It provided us with an overview of the database, and the ability to test certain features within the website. PHPMyAdmin was crucial for verifying that the website was correctly storing and retrieving the correct data from each column or table. The coding environments used in developing the project were Microsoft Visual Studio and Visual Studio Code. All of the changes to our project, aside from the database, were stored on Github. This allowed us to have version control for the project and ensure that all developers were working on the most recent version.

The front-end tool that we used to create our user UI was W3 CSS framework. HTML, CSS, and JavaScript were all used in conjunction with the framework. HTML allowed us to format and separate the sections of our pages as needed. CSS was used primarily to implement consistent formatting for forms, headers, and buttons throughout the entire website. JavaScript was used to create special scripts for certain features of our website. For instance, we had to create a function for every accordion button to allow it to collapse and expand properly.

1.3 Languages used for development

The Primary languages used to develop our site are PHP, HTML, CSS and JavaScript. PHP was chosen as the primary language due to its built in tools for managing SQL queries and connecting to our database, as well as it being the primary language taught in the lab section of this course. For HTML, we used W3 for formatting and style as it was much simpler and more easy to manage that the traditionally used Bootstrap. This tool enabled our team of inexperienced web developers to create a competent and visually pleasing website with much less difficulty.

2 BEST BUY DATABASE

For our Best Buy Database, we needed several tables to keep track of the data required for our website to function. Each product page has its own table, which allows for better product descriptions. This project is intended to allow a customer to sign up, login, then select a type of product or service that they would like. Once they have selected a product or service, they are able to purchase it from a vendor. Vendors, as well will have the ability to create and account and login. They will also be able to post a product or service that they have available, which will be anonymously listed. To manage all of the necessary data, users, vendors, products, and services, the following tables were created.

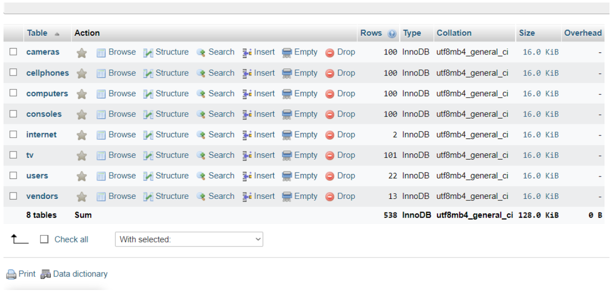


	Table ▲	Action	Rows	Type	Collation	Size	Overhead
□	cameras	★ Browse Structure Search Insert Empty Drop	100	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	cellphones	★ Browse Structure Search Insert Empty Drop	100	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	computers	★ Browse Structure Search Insert Empty Drop	100	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	consoles	★ Browse Structure Search Insert Empty Drop	100	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	internet	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	tv	★ Browse Structure Search Insert Empty Drop	101	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	users	★ Browse Structure Search Insert Empty Drop	22	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	vendors	★ Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	8 tables	Sum	538	InnoDB	utf8mb4_general_ci	128.0 KiB	0 B

Figure 3: Best Buy Project Tables

2.1 User Table

The user table contains all of the necessary information needed by our website to create and manage a profile. The user table consists of the user's ID, username, email, password, first name, last name, and the date the account was created. The user's ID and username serve as the primary key for this table. The username and password are used during the login process. The user ID is checked on every page to verify that a user is logged in. This is to prevent a user from purchasing an item without an account.

2.2 Vendor Table

The vendor user table contains all of the necessary information needed by our website to create and manage a profile. The user table consists of the vendor's ID, username, email, password, company name, and the date the account was created. The vendor's ID and username serve as the primary key for this table. The vendor username and password are used during the login process. The vendor's ID is used during the checkout process. It is also used to prevent a product from being registered without first logging into a vendor account. The vendor ID is also attached to each of their products that are listed, which allows them to view the items or services that they are selling on the vendor dashboard.

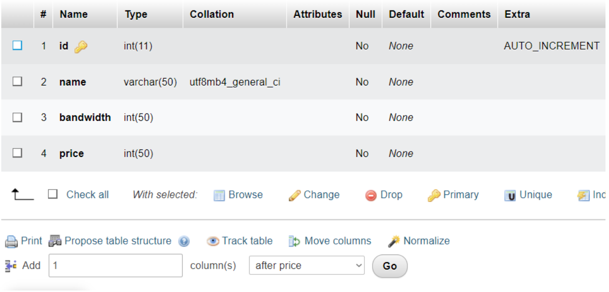
2.3 Product Tables

The product tables by far hold the majority of the information inside our database. They contain all of the different products listed by the different vendors that we have registered. These tables are necessary because they allow us to store what the vendor is selling, along with all of the information that is connected to that product. That information is then shown to our customers, who can purchase the item, causing it to be removed from the database. It is also necessary to separate all of our products by type to allow more options for detail in our product descriptions.

The SQL queries also become faster due to this, since there is less data that it has to search through overall. We currently have a table for the following products: Cameras, Cellphones, Laptop Computers, Consoles, and TVs. Each table holds at least one hundred different products that our vendors have for sale. Each table is also unique to the type of product that it holds. For instance, the TV table holds a particular TV's screen size, resolution, the year is was manufactured, the brand, and the price the vendor is listing it for. There are two IDs on each product. One ID is used as a key within the table, and the other holds the ID number of the vendor that listed it. This allows the vendor to track which products they have listed, across multiple tables.

2.4 Services Table

The service table is the smallest table within our database. It has the fewest rows and columns of all the tables. The internet service category is where vendors can provide internet to users at the listed monthly price. Before any services are shown, the customer is prompted to use our custom bandwidth calculator. The user enters a list of how many devices they own that will be using the internet. Once the user enters this information, the website will match an internet service that fits the calculated bandwidth criteria, as well as similar options. The table contains an ID number, the companies name, the maximum bandwidth for a particular package, and the price for that package.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	id	int(11)			No	*None*		AUTO_INCREMENT
□	2	name	varchar(50)	utf8mb4_general_ci		No	*None*		
□	3	bandwidth	int(50)			No	*None*		
□	4	price	int(50)			No	*None*		

Figure 4: Services Table in PHPMyAdmin

3 BEST BUY SEARCHING INTERFACE

3.1 Customer Login Interface

The login page for our customers is fairly simple. This page is connected to our database so that it has access to the users table. It will check to see if the username exists and if the password matches to the username entered. If both of these hold, then the customer will be able to login and go to the customer dashboard page.

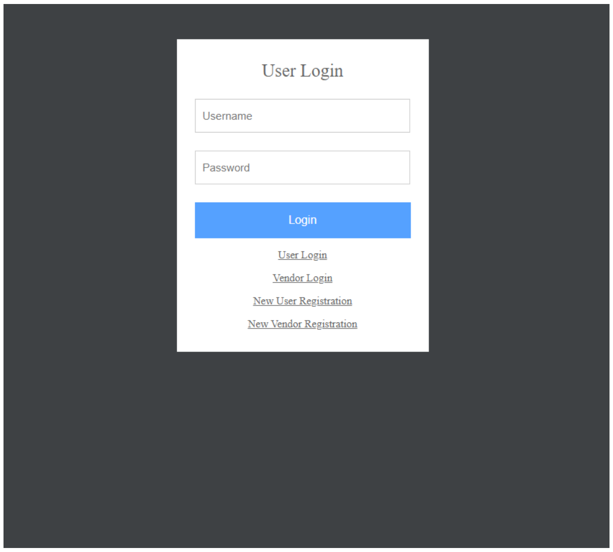
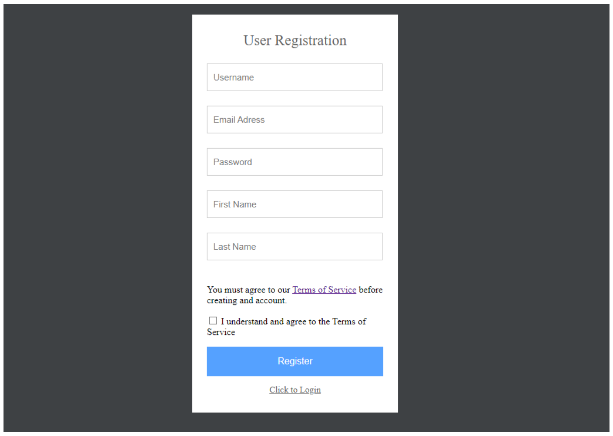


Figure 5: Customer Login Page

3.2 Customer Account Registration Interface

The customer registration page is a bit more complex than the login page. Here, we ask the user to enter their personal information in order to create an account. This information will be stored in the user table in our database. All fields here are required, so they must enter in a username, email, password, first name, and last name. There is also a checkbox that must be checked that states that the user agrees with our terms of service. These terms include the binding contract when purchasing products from the site. After they register, they will be brought back to the user login page.



You must agree to our Terms of Service before creating and account.

□ I understand and agree to the Terms of Service

Figure 6: User Registration Page

3.3 Vendor Login Interface

The login page for our vendors is fairly simple and looks almost identically to the customer login page. This page is connected to our database so that it has access to the vendors table. It will check to see if the username exists and if the password matches to the username entered. If both of these hold, then the vendor will log in and be redirected to the vendor dashboard page.

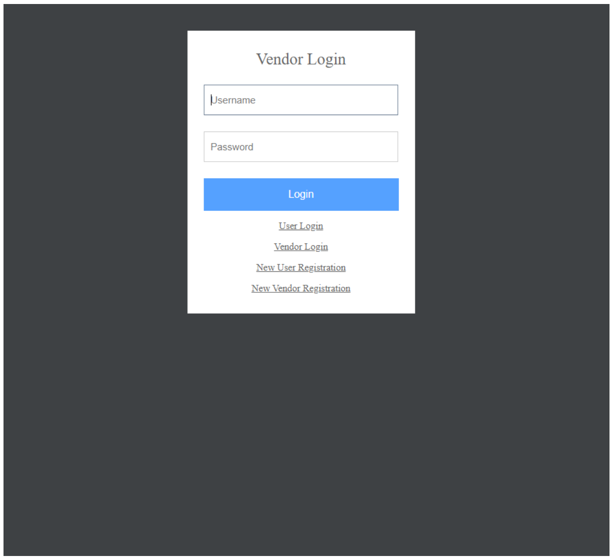
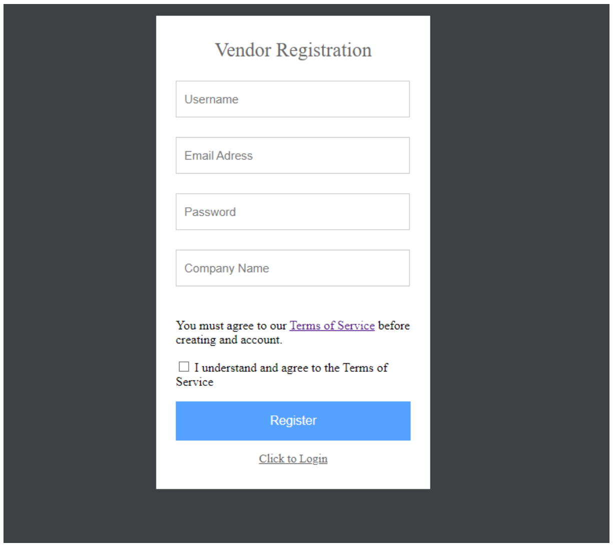


Figure 7: Vendor Login Interface

3.4 Vendor Registration Interface

The vendor registration page is a bit more complex than just the login page and very similar to the customer registration page. Here, we ask the user to enter in all of their information so we can create an account and add them to the vendor table in our database. All fields here are required, so they must enter in a username, email, password, and a company name. There is also a checkbox that must be checked that states that the vendor agrees with our terms of service. These terms include the binding contract when purchasing and listing products on the site. After they register, the will be brought to the vendor login page.



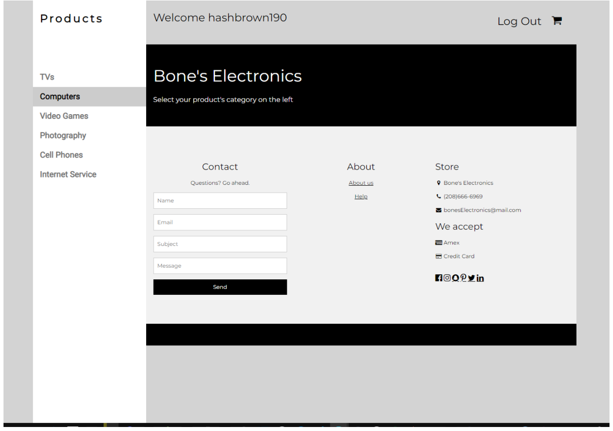
You must agree to our Terms of Service before creating and account.

□ I understand and agree to the Terms of Service

Figure 8: Vendor Registration Interface

3.5 Customer Dashboard Page

Our customer dashboard page is relatively simple to ensure easy navigation through our website. Here you can see the name of our website, your personal username to show that you are logged in, and the products section, and more. The products section is a list of categories that the user can select from. By selecting a category, the user will be redirected to their selected page where those products will be shown. There is a logout button used to end your current session as well as some basic information about payment and other things.



Welcome hashbrown190

Bone's Electronics

Select your product's category on the left

Questions? Go ahead.

Name

Email

Subject

Message

Bone's Electronics

(208)666-6969

bonesElectronics@mail.com

We accept

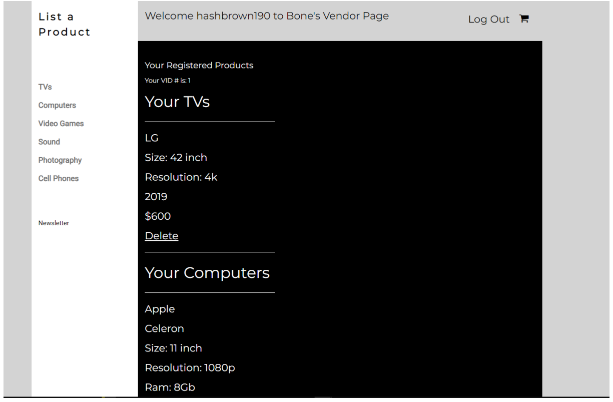
Amex

Credit Card

Figure 9: Customer Dashboard Page

3.6 Vendor Dashboard Page

The Vendor dashboard page is the first page that vendors will see when they login to our site. This page will list all of the products and their relevant information that the specific vendor has listed on the site. For each listing there will also be a delete button that will enable the vendor to quickly and easily remove product listings. On the left hand side of the page is a list of all the categories of items that are able to be listed on our site. The vendor may select one of these options and will be brought to the product registration page for the corresponding product type. Here the vendor will fill out the details of their product and create the listing.



Welcome hashbrown190 to Bone's Vendor Page

Your Registered Products

Your VID # is: 1

Your TVs

LG

Size: 42 inch

Resolution: 4k

2019

$600

Your Computers

Apple

Celeron

Size: 11 inch

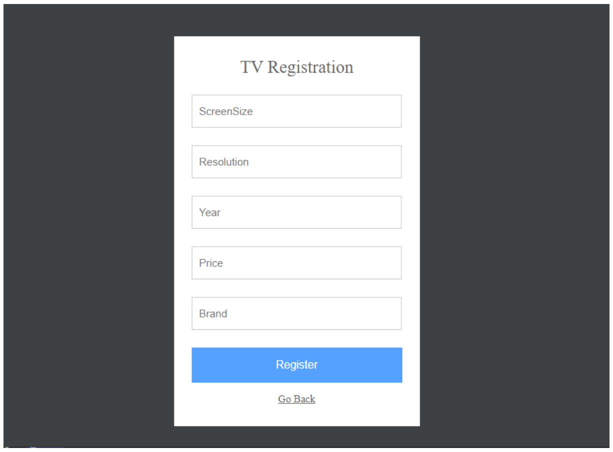
Resolution: 1080p

Ram: 8Gb

Figure 10: Vendor Dashboard Page

3.7 Product Registration Page

The individual product registration pages each include a form that is tailored to that particular category of product. Here the relevant information will be filled in by the vendor and the listing will be created. Each listing will also have and associated ID number that will match the vendor ID number of the vendor that listed it. This is what allows us to retrieve the listings associated with each vendor on the vendor dashboard.



TV Registration

ScreenSize

Resolution

Year

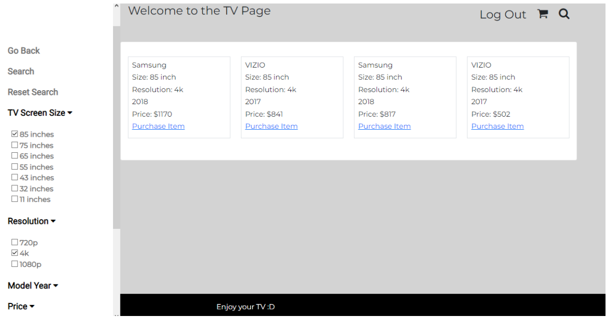
Price

Brand

Figure 11: TV Registration Page

3.8 Product Searching

When selecting a category of product, several drop-downs will be able to be selected. These drop-downs correspond to relevant search criteria for the product. For example, for the TV search you will be able to specify the screen size, resolution, Model Year, Price and Brand of the TV you wish to purchase. These criteria are selected by checking a number of check boxes that corresponds to certain specifications. In the example Image below you can see that we have searched for TV's that are 85 inches in screen size and have a resolution of 4K. The site returned all TV results that match these criteria. All criteria are exact-match based except for price. Price is specified in increments of $100 up to $1000. Selecting $600 will retrieve all results with a price of $600 or less.



Welcome to the TV Page

TV Screen Size

☑ 85 inches
☐ 75 inches
☐ 65 inches
☐ 55 inches
☐ 43 inches
☐ 32 inches
☐ 11 inches

Resolution

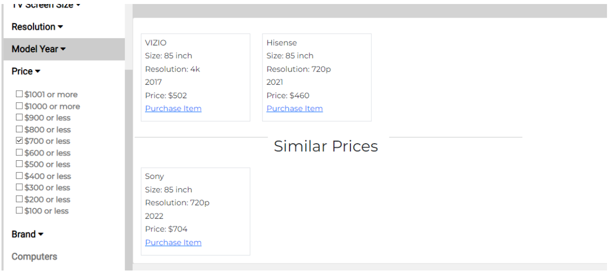
☐ 720p
☑ 4k
☐ 1080p

Samsung	VIZIO	Samsung	VIZIO
Size: 85 inch	Size: 85 inch	Size: 85 inch	Size: 85 inch
Resolution: 4k	Resolution: 4k	Resolution: 4k	Resolution: 4k
2018	2017	2018	2017
Price: $1170	Price: $841	Price: $817	Price: $502

Enjoy your TV :D

Figure 12: Search Criteria on TV Products Page

When specifying a price, if there are items just outside of your specification our site will list those in a new category below labelled "Similar Prices". This allows the User to see listings that may be only slightly more expensive that what they are looking for. An example of A query that produces such a case is shown below.



VIZIO
Size: 85 inch
Resolution: 4k
2017
Price: $502

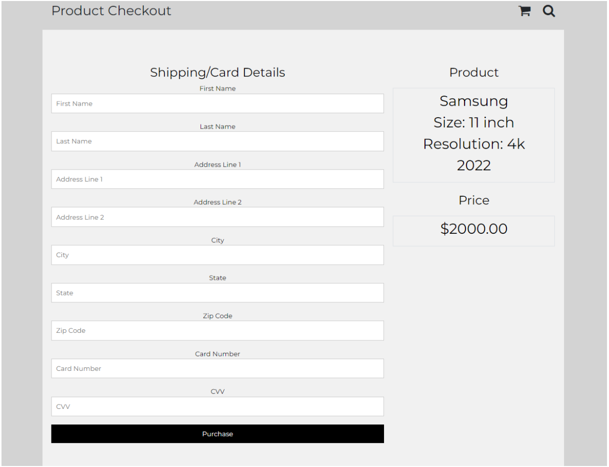
Hisense
Size: 85 inch
Resolution: 720p
2021
Price: $460

Sony
Size: 85 inch
Resolution: 720p
2022
Price: $704

Figure 13: Similar Prices on TV Products Page

3.9 Checkout Page

Once a user clicks on the purchase link on a product they wish to buy, the user will be brought to the checkout page where they will be able to purchase their item. The checkout page contains the form where the user will enter their card information as well as shipping information for the product to be delivered. The product information is listed on the right side of the page for the user to see.

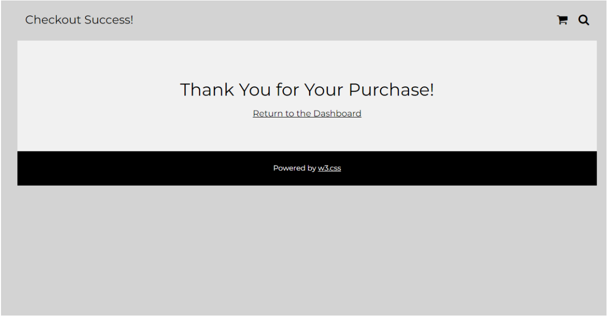


Samsung
Size: 11 inch
Resolution: 4k
2022

$2000.00

Figure 14: Checkout Page

When all of the fields in the form are filled out with relevant information and the user clicks purchase, the user will be redirected to a confirmation page as shown below. This will also delete that listing from the database.



Thank You for Your Purchase!

Figure 15: Purchase Confirmation

3.10 Internet Service Page

When a user clicks on the internet service button they will be directed to the internet services page. On this page is a form that lists several of the most common devices that would use internet in the average home. Each of these fields are designed to only accept values between 0-100 and the form will not submit unless all fields are filled in.

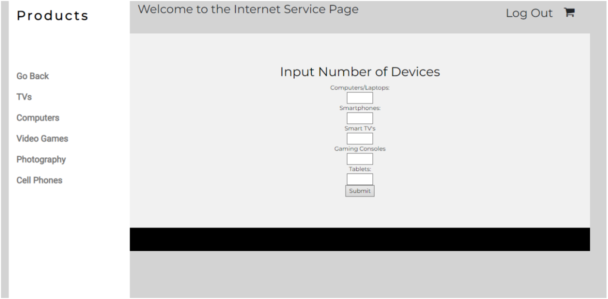
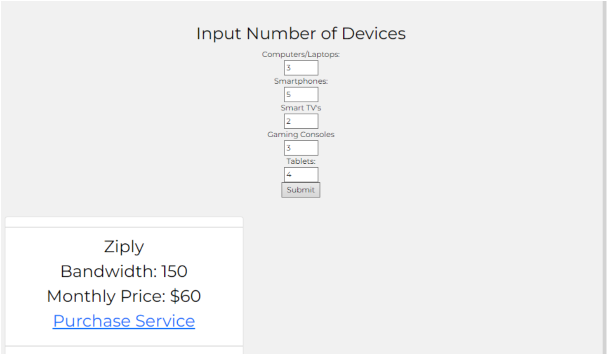


Figure 16: Internet Services Page

When the user inputs the number devices they have in their home or office and hit submit, the page will calculate the total bandwidth that the user would need based on average bandwidth usage per device. It will then use this calculation and bring up all internet service offers that include that amount of bandwidth or greater as show below.

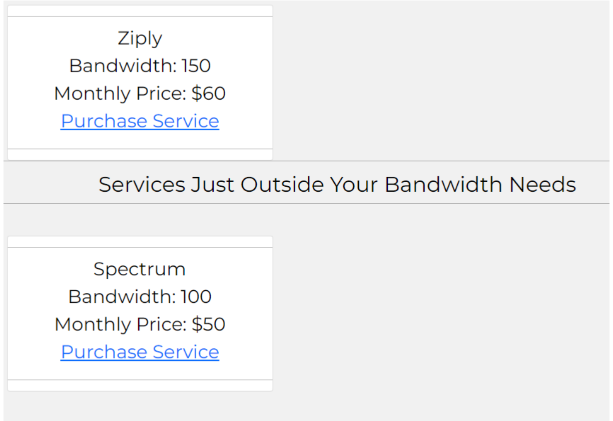


Computers/Laptops: 3
Smartphones: 5
Smart TV's 2
Gaming Consoles 3
Tablets: 4

Ziply
Bandwidth: 150
Monthly Price: $60

Figure 17: Internet Search Page

The site will also include internet offers that have slightly less bandwidth in order to give the customer more options and potentially saving them money. An example of this is shown in the figure below.



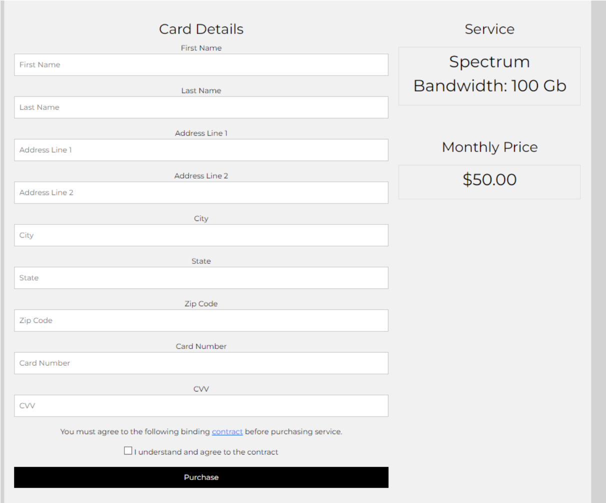
Ziply
Bandwidth: 150
Monthly Price: $60
Purchase Service

Services Just Outside Your Bandwidth Needs

Spectrum
Bandwidth: 100
Monthly Price: $50
Purchase Service

Figure 18: Internet Search Page

The checkout page for Internet service is very similar to the product checkout page but includes the information on the service that the customer is buying. At the bottom of the page is a checkbox that the customer must check to acknowledge that they agree to the binding contract between them and the service provider. This contract is different than the general terms of service contract that users and vendors agree to when creating an account.



Card Details

First Name: First Name
Last Name: Last Name
Address Line 1: Address Line 1
Address Line 2: Address Line 2
City: City
State: State
Zip Code: Zip Code
Card Number: Card Number
CVV: CVV

You must agree to the following binding contract before purchasing service.

□ I understand and agree to the contract

Purchase

Service

Spectrum
Bandwidth: 100 Gb

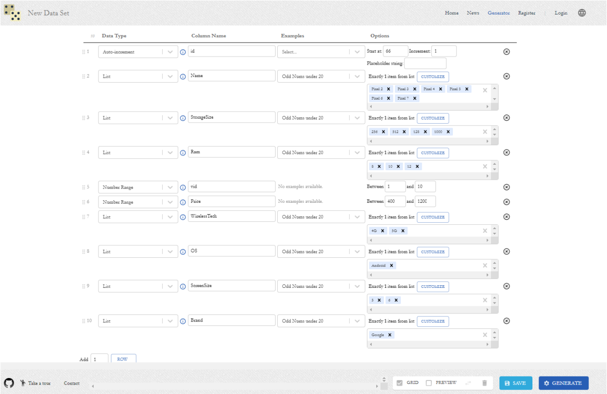
Monthly Price

$50.00

Figure 19: Internet Checkout Page

4 DATABASE POPULATION WITH GENERATE DATA TOOL

A website named generatedata.com was used to fill all of our databases with products. There were a few different ways that we could have gone about generating mass amounts of data for our project. The two main concepts that were suggested to us were to use Faker or Mockaroo. Faker was a bit more complex then what we were looking for. Faker is a Python package that generates fake data for you. Mockaroo is a simpler way to generate data, and would have been a good alternative to generatedata.com, however it was still a bit complex in comparison to what we had found. The website generatedata.com was quite convenient because it generates the SQL query needed to import the data into a PHPMyAdmin table.



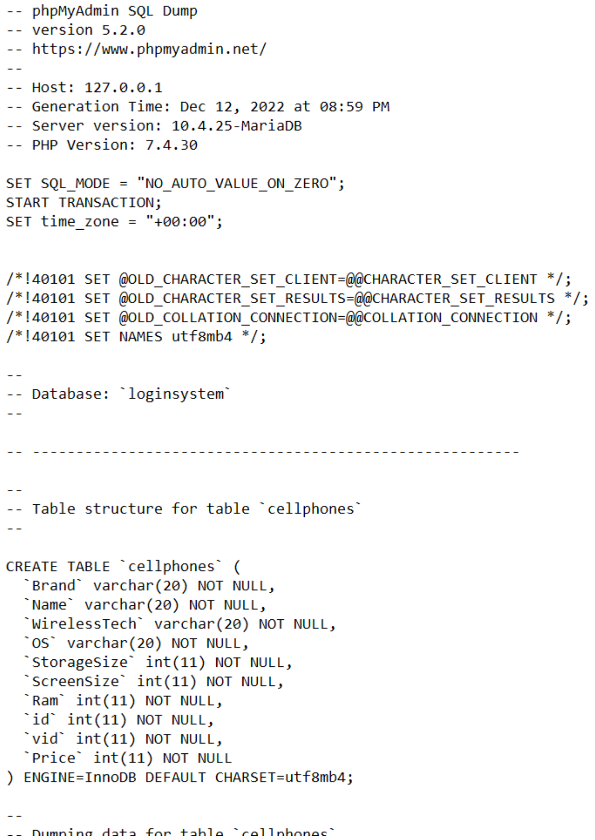
	Data Type	Column Name	Examples	Options
1	Auto-increment	id	Select...	Start at: 66 Increment: 1 Placeholder string:
2	List	Name	Odd Nums under 20	Exactly 1 item from list CUSTOMIZE Pixel 2, Pixel 3, Pixel 4, Pixel 5, Pixel 6, Pixel 7
3	List	StorageSize	Odd Nums under 20	Exactly 1 item from list CUSTOMIZE 256, 512, 128, 1000
4	List	Ram	Odd Nums under 20	Exactly 1 item from list CUSTOMIZE 8, 10, 12
5	Number Range	vid	No examples available.	Between 1 and 10
6	Number Range	Price	No examples available.	Between 400 and 1200
7	List	WirelessTech	Odd Nums under 20	Exactly 1 item from list CUSTOMIZE 4G, 5G
8	List	OS	Odd Nums under 20	Exactly 1 item from list CUSTOMIZE Android
9	List	ScreenSize	Odd Nums under 20	Exactly 1 item from list CUSTOMIZE 5, 6
10	List	Brand	Odd Nums under 20	Exactly 1 item from list CUSTOMIZE Google

Add 1 ROW

Figure 20: Data Generation Tool

4.1 Data Generation

To start creating the data, you must copy the row names of the tables into the website. This was accomplished by selecting various data types depending on what the data was going to be. If we needed a random integer, a number range could be used. For product names or brands, a list that contains everything that we would like to be an option can be created. After all of the options that we want are listed, we can randomly generate up to one-hundred random table entries. Clicking the generate button will download a file which contains the SQL statements that are used to insert the data into a database. From there, all that is left to do is to import the file into your table in PHPMyAdmin.



```
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Dec 12, 2022 at 08:59 PM
-- Server version: 10.4.25-MariaDB
-- PHP Version: 7.4.30

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `loginsystem`
--

-- --------------------------------------------------------

--
-- Table structure for table `cellphones`
--

CREATE TABLE `cellphones` (
  `Brand` varchar(20) NOT NULL,
  `Name` varchar(20) NOT NULL,
  `WirelessTech` varchar(20) NOT NULL,
  `OS` varchar(20) NOT NULL,
  `StorageSize` int(11) NOT NULL,
  `ScreenSize` int(11) NOT NULL,
  `Ram` int(11) NOT NULL,
  `id` int(11) NOT NULL,
  `vid` int(11) NOT NULL,
  `Price` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `cellphones`
```

Figure 21: Sample Generated Data

5 MYSQL QUERIES

Searching through the database for products was accomplished by using SQL queries. Checkboxes were used rather than a search bar that the user could type into for simplicity. This also acted as a safety feature and improved error checking since it limited what the user could enter. The checkboxes are generated by an SQL statement that looks through a database and selects distinct values for each column. These columns are then ordered and displayed for the user. Selecting these checkboxes and submitting will trigger another SQL statement to run, which will find all of the values related to the checked boxes. All of the pages that display products use SQL to get said information from the various databases.

6 PROJECT DIFFICULTIES

There were many small issues throughout the creation of this website, however the most difficult to deal with was initially setting up the search. We did not plan to use AJAX, but wanted to have multiple checkbox forms. This posed a problem since only one form can be submitted at a time. If multiple forms are submitted, only the last form will work. After much time and experimentation, we decided to use a single form for all checkboxes, which meant that we had to process everything at once. To accomplish this, each section of checkboxes would store the selected values in an array, which would then be used in an SQL statement to find matching products in our database.

This also became a problem, as the user may not check an option from each checkbox section. Initially, we had one large SQL statement that processed all of checked values for each checkbox section. The first design was to use many nested for-each statements to cycle through all of the selected checkbox values, run the SQL statement, store the result in an array, remove duplicates from the array, and display the results. As you might imagine this became too messy to deal over time. We eventually found a way to exclude part of an SQL statement if none of the options are selected in a checkbox section.

Another issue that we dealt with was planning and accomplishing progress within the time frame. Our overall planning was mediocre, which was in part due to the fact that all of the members of the project were quite busy with their other classes, and none of us had experience in web development before starting this class. It was hard to estimate the kind of work and time required to accomplish each goal. We did not have a timeline as we were uncertain of what would be required to accomplish the tasks at hand. For most of the project, we worked on issues as we encountered them while blindly making something that we expected would work to meet all of the requirements.

7 IMPROVEMENTS

7.1 Site Improvements

There are several things that could be done to improve our site functionality and give it more polish. One major thing that could be done to improve the UI would be to add the ability to add and display images for the product listings. These images are standard for essentially every marketplace on the internet and would definitely make our website seem much more polished and professional. We chose not to include this in our project since the majority of our data would be randomly generated. The images cannot be randomly generated, so we would have had to add each individual image for each product, which would have required much more time and effort than necessary. Another way that the site could be improved would be to use AJAX to display and search for the products. This would allow the site to retrieve search results without refreshing the current page. The site could also be improved by added the ability to specify a custom price range when searching for products. Product categories should have multiple pages to help improve the speed that it refreshes, especially if we had thousands of products listed in one category.

7.2 Course Improvements

While there were many good aspects of the course, there are some things that we would like to see improved for future students. One issue that we had was the lack of a timeline for project development. All one-on-one check-ins were optional. We believe that some should have been required to provide more incentive for progress. If these check-ins were graded, then more people would make progress on their projects, which would improve the results by the Project Report: Phase II slideshow. Another potential idea is holding students accountable for their contribution. While it was not much of an issue in this project, individuals should list their contributions on each Project Report and check-in. Those who fall significantly short should not be credited for the complete product at the end of the project.

8 CONCLUSION

In conclusion, our site includes many pages and demonstrates near full functionality for the sites purpose. Each user or customer can search for their needs. Vendors can list products and services under a binding contract. Product searches result in an exact match, and will also display similar prices if a price limit is selected. Services are sorted by bandwidth, and the customer can find exactly what they need by using our built-in bandwidth calculator as a way to prevent customers from purchasing a service that they do not need. Logging in is required for site access, and all users can create an account as a vendor or customer under our binding contract. Our Best Buy website is a product and service matching site that meets the requirements laid out in the CS 360 syllabus.

9 SOURCE CODE

The source code for this project is available at
https://github.com/cmullins00/CS-360-Group-Project.git

10 REFERENCES

[1] Hasan Jamil. CS 360: Database Systems. Retrieved December 13, 2022 from https://canvas.uidaho.edu/courses/12913
[]