Final Paper

Austin Gutierrez (1214766501)

Arizona State University

IFT 200

Table of Contents

Summary

Thank you for choosing Brew Brothers for your database needs. We are a Technology Company specializing in creating, maintaining, and improving relational database for everyone. Databases are essential and play an integral part of any business. We will not only establish the backbone to your database, but we will also develop a graphical user interface (GUI), in the form of a website, that will allow your employees to manage the data accordingly. This document outlines the steps taken, from start to finish, to create an example database using beer, spirits, and wine.

Main Page

First things first, we must install the appropriate applications needed to lay the ground work of the database. For this project, we will be using a "LAMP" stack consisting of a Linux OS (Manjaro), Apache web server, PHP scripting language, PhpMyAdmin for data management, and MariaDB for the database. Once these programs are installed and configured for our needs, our next step is to outline the planned database.

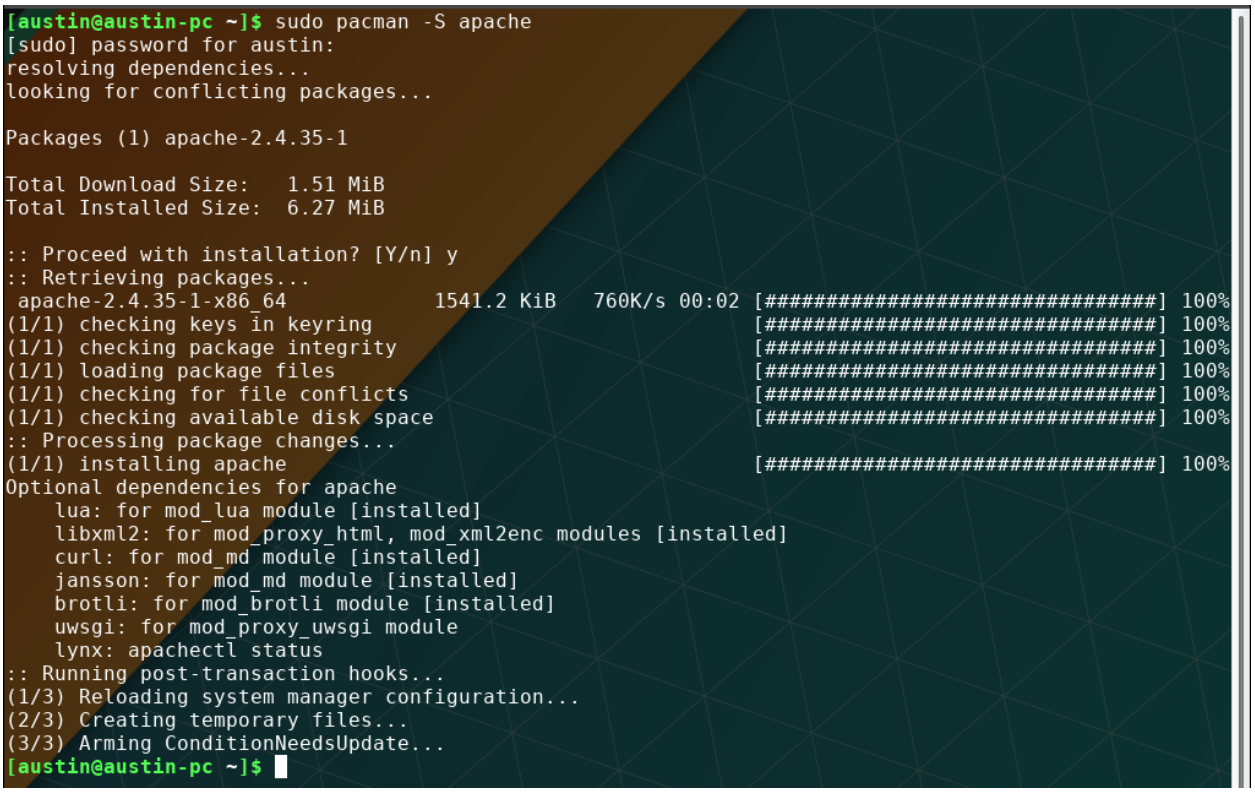
To outline the database, we need to create an Entity Relationship Diagram (ERD) to establish the data, table, attributes, and relationship for our database. Once this diagram is created, it will allow us to organize the tables accordingly and ensure the proper primary and foreign keys are created. This will be used to help maintain the database and increase the efficiently as the database grows. After we have the outline, we will connect to MariaDB services and create the database and each table accordingly. After the tables are created, we can fill in the information from an already completed flat file.

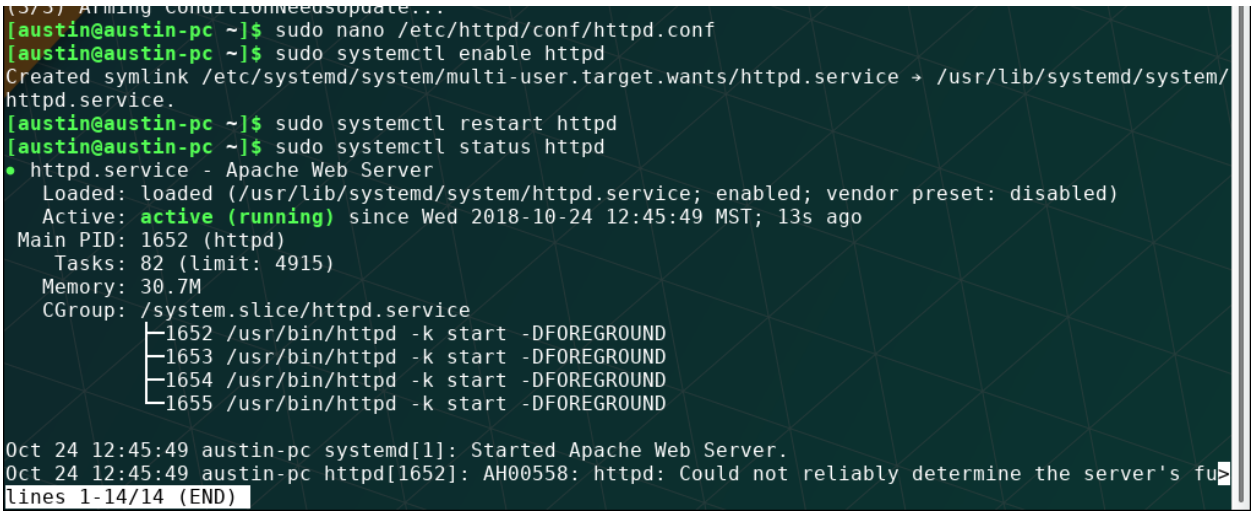
Our next step in the process will be to create basic forms that allow user to input data into the appropriate table. This will be completed using PHP and HTML to provide an GUI. Each table will have its own form dedicated to inputting information into the database. For administration purposes, we will only allow information to be deleted through the admin account using phpMyAdmin. We will also use PHP and HTML to create the "Query" page. This page will allow user to query each one of the tables and display the results on the same page. More detailed queries can be developed later.

Once all these products are completed, they will be compiled into a website with the options to choose what they want to do. This GUI will allow for functionality of the database for the user with limited access based on their needs.

Install Apache Web Server

Utilizing the pacman package manager, I typed the command into the terminal to install apache onto my computer. After Apache was installed, I had to enable httpd and verify that the server was running using the commands “sudo systemctl enable httpd”; “sudo systemctl restart httpd”, and “systemctl status httpd”. Below are the screenshots of the installation and verification that the server is running.



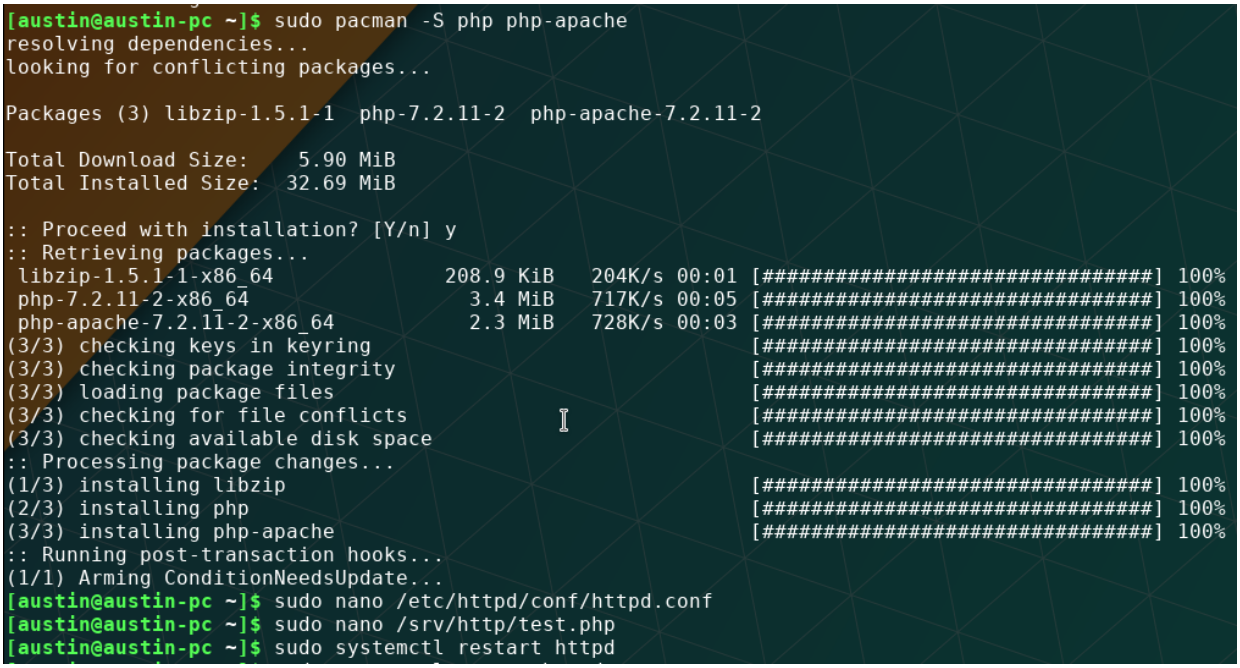


Installing MySQL/MariaDB

To install MySQL, I used the command “sudo pacman -S mysql”. During the installation, I was asked if I wanted to use mariadb or percona-server. I choose to use mariadb. Once mysql was installed, I had to run the mysql_secure_installation to set up the database and give me a admin account to manipulate data. After these steps were complete, I was able to log into my database to manipulate data.

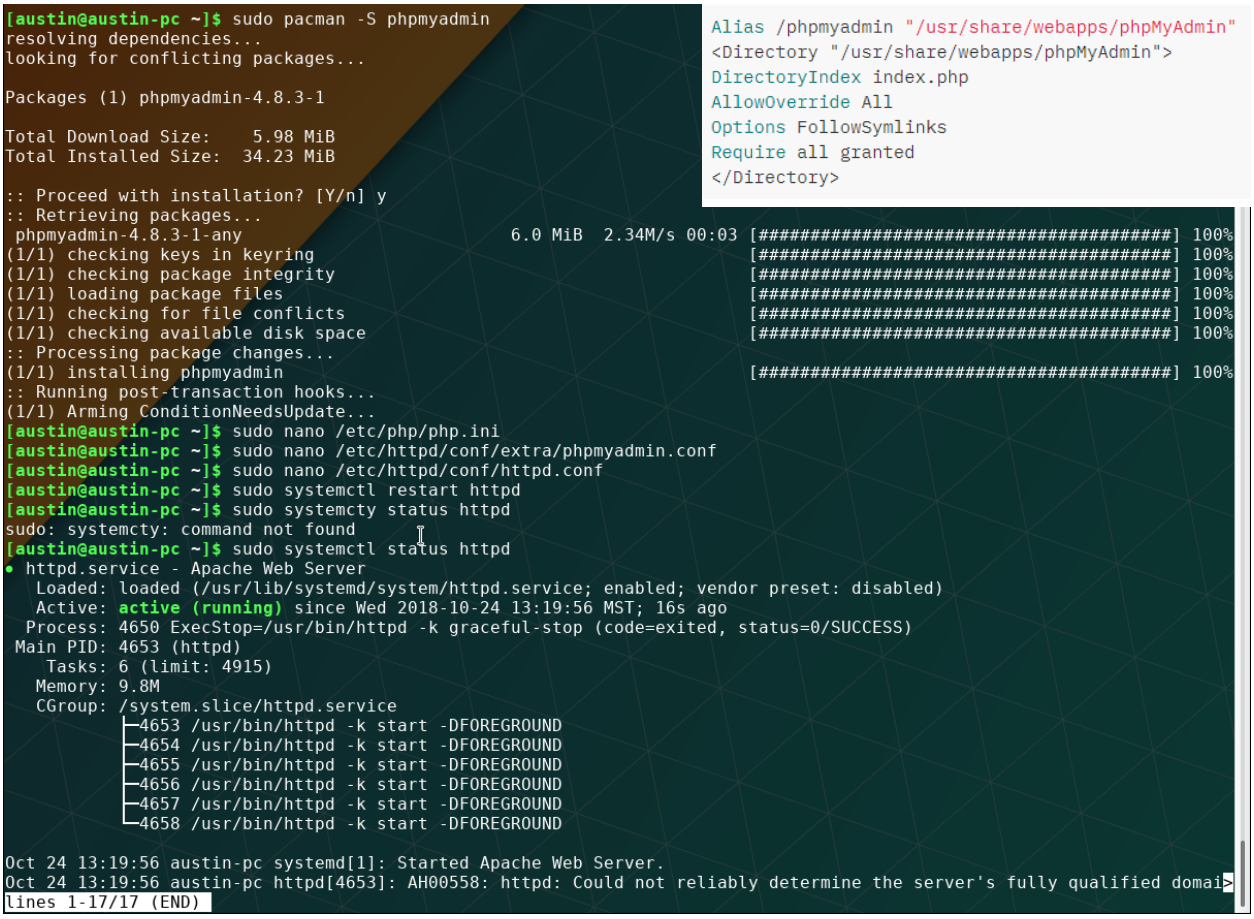
Install PHP

To install PHP onto my system, I ran the command “sudo pacman -S php php-apache”. After the application was installed onto the system, I had to configure the Apache PHP module by editing the /etc/httpd/conf/httpd.conf file to include php7 modules, load the mom_prefork_module, and include the php7_module.conf file. Once this was complete I had to restart the httpd service with the “sudo systemctl restart httpd” command (since I edited the configuration file).



Install phpMyAdmin

To install phpMyAdmin, I ran the command " sudo pacman -S phpmyadmin". Once the application was installed, I had to edit the /etc/php/php.ini file to uncomment the extensions for bz2.so and mysqli.so (the mysqli.so will be needed when creating the forms later on with php and html). After this is complete, the next step is to edit the configuration file for phpMyAdmin located in /etc/httpd/conf/extra/phpmyadmin.conf to include the following lines (white text box in terminal image below). After this is added, we have to edit the Apache config file and add "Include conf/extra/phpmyadmin.conf" to it. Last step is to restart the httpd service and verify that it works.



Entity Relationship Diagram

Once the applications for the LAMP stack were installed, it is time to build the Entity Relationship Diagram for the database. This diagram will identify each table that was created and their attributes. We will also identify the primary and foreign keys used to create the relationships in our database. This will also give us a visual representation of the database and workflow. For this example, all the items in the database will be linked together based off a category called Manufacture (brewing company).

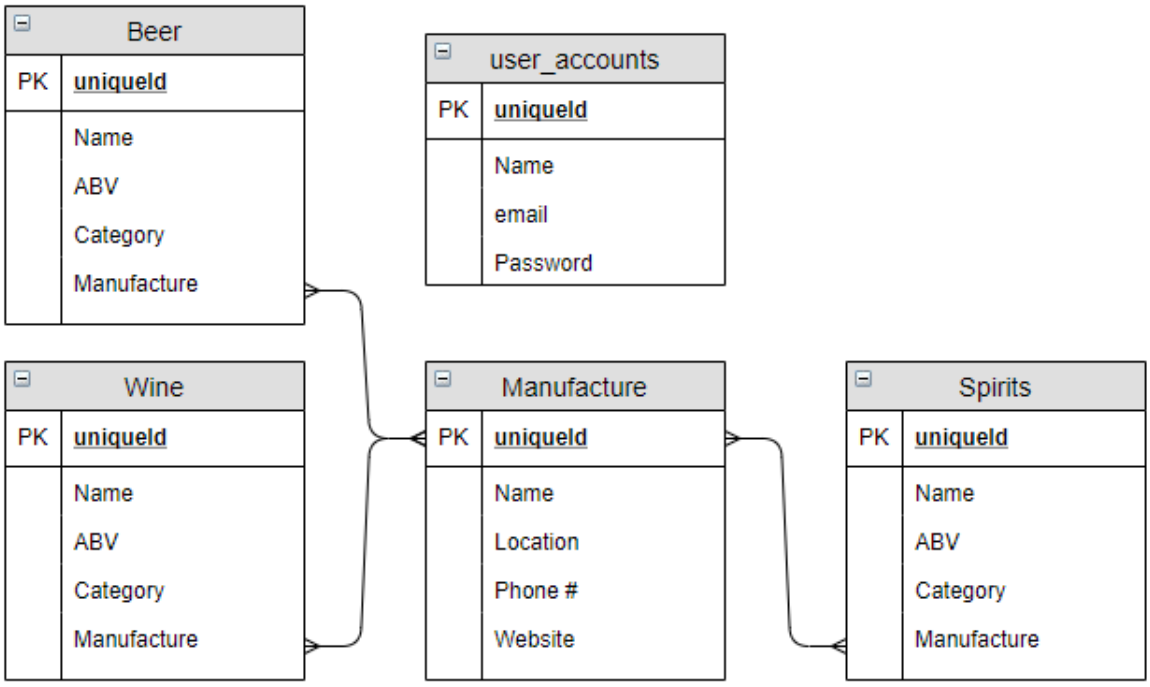


Table Creation in MariaDB

After the ERD was created and the outline of the database was complete, there was enough information available to create the tables into the database. To do this, I utilized the command line to connect to the database and create the table information. In order to do this, I had to connect to the database through the command line with the following command "mysql -u root -p -h localhost AdultBeverage". This connected me to the database and allowed me to create the tables from the command line instead of using phpMyAdmin (alternate way)

PHP/HTML/MYSQL Forms

Below is an example script used to create each form and the result. This form is styled with HTML and utilizes PHP scripts to run MYSQL commands directly into the database. This allows for data to be inputted into the database with ease. The user doesn't need any knowledge of MySQL or programming to use these forms. These forms also display a query at the bottom of the form to allow the user to see if certain information is/is not in the database yet and provides the option to add the appropriate data that is missing.

```
<h1>Input a new brew <a href="../Home.php" style="float:right">Homepage</a>

    <div id="contact-area">

        <form action="Beer_Process.php" method="post" name="Beer">

            <label for="Name">Name:</label>
            <input type="text" name="Name" placeholer="Amazing Drink" value="
            <br>

            <label for="ABV">ABV:</label>
            <input type="text" name="ABV" placeholder="Alcohol content" value
            <br>

            <label for="Category">Category:</label>
            <input type="text" name="Category" placeholder="Stout, Blonde, La
            <br>

            <label for="Manufacture">Manufacture:</label>
            <input type="text" name="Manufacture" placeholder="Manufacture ID
            <br>

            <input type="submit" name="submit" value="Submit" class="submit-b

        </form>
        <br><br><br><br>

    </div>

    <div>
        <table>
            <h2>Manufacture ID</h2>
            <caption>
                *If manufacture is not listed, please add information to data
                <a href="../AddMan-form.php" style="float:right">Add Manufact
            </caption>
            <?php
            $sqlMan = "SELECT DISTINCT Man_ID, Man_Name FROM Manufacture JOIN Beer ON Manufactur
            $resultMan=mysqli_query($db_connect, $sqlMan) or die("bad query");

            echo "<tr><th>ID</th><th>Name</th></tr>";
            while ($row = mysqli_fetch_assoc($resultMan)) {
                echo"<tr><td>{$row['Man_ID']}</td><td>{$row['Man_Name']}</td></tr>";
                }
            ?>
        </table>
```

Input a new brew Homepag

Name:

ABV: Alcohol content

Category: Stout, Blonde, Lager...

Manufacture: Manufacture ID, see below

Submit

Manufacture ID

*If manufacture is not listed, please add information to database Add Manufacture

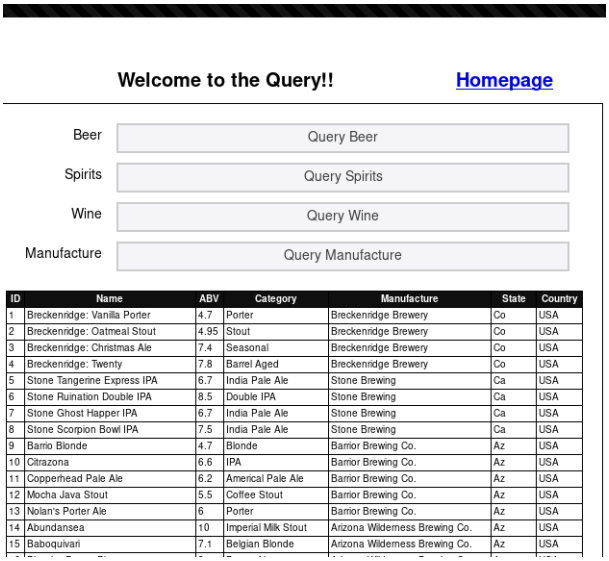
ID	Name
1	Breckenridge Brewery
2	Stone Brewing
3	Barrior Brewing Co.
4	Arizona Wilderness Brewing Co.
5	SanTan Brewing Company
6	Guinness Storehouse
7	Samuel Adams Brewery

```
//inserting data into table

$query=mysqli_query($db_connect, "INSERT INTO Beer
    (Name, ABV, Category, Manufacture)
    VALUES ('$name', '$ABV', '$Category', '$Manufacture');")
    or die(mysqli_error($db_connect));

mysqli_close($db_connect);

//echo "Successfully Updated, Please return to forms";
header("Location: ../AddBeer-form.php?note=success");
```

Queries / Reports

Below is the example script used to create the query/report page that allows user to query each table. The query searches two tables (the item and the manufacture) and combines the results to display the relevant information combined. Every time the user clicks on the respective query button, the information is displayed below. If new information is added to the database, it will display once the user clicks on the button again. These queries are completed using PHP to inject MySQL statements into the database and fetch the results. You can see the queries that were used below to get the appropriate information from each table

```
rm action="QueryForm.php" method="post">

<label>Beer</label>
<input type="submit" name="Beer" value="Query Beer"> <br>

<label>Spirits</label>
<input type="submit" name="Spirits" value="Query Spirits"> <br>

<label>Wine</label>
<input type="submit" name="Wine" value="Query Wine"> <br>

<label>Manufacture</label>
<input type="submit" name="Manufacture" value="Query Manufacture"><br>

<?php
    if (isset($_POST['Beer'])) {
        $sql = "SELECT Beer.*, Manufacture.Man_Name, Manufacture.State, Manufacture.Country
            FROM Beer INNER JOIN Manufacture ON Beer.Manufacture=Manufacture.Man_ID;";
        $result = mysqli_query($db_connect, $sql) or die("bad query");

        echo"<table>";
        echo"<tr><th>ID</th><th>Name</th><th>ABV</th><th>Category</th>
            <th>Manufacture</th><th>State</th><th>Country</th></tr>";

        while($row=mysqli_fetch_assoc($result)) {
            echo"<tr><td>{$row['Beer_ID']}</td><td>{$row['Name']}</td><td>{$row['ABV']}</td>
                <td>{$row['Category']}</td><td>{$row['Man_Name']}</td><td>{$row['State']}</td>
                <td>{$row['Country']}</td></tr>";
        }
    }

    if (isset($_POST['Spirits'])) {
        $sql = "SELECT Spirits.*, Manufacture.Man_Name, Manufacture.State, Manufacture.Country
            FROM Spirits INNER JOIN Manufacture ON Spirits.Manufacture=Manufacture.Man_ID;";
        $result = mysqli_query($db_connect, $sql) or die("bad query");

        echo"<table>";
        echo"<tr><th>ID</th><th>Name</th><th>ABV</th><th>Type</th><th>Category</th><th>Age</th>
            <th>Manufacture</th><th>State</th><th>Country</th></tr>";

        while($row=mysqli_fetch_assoc($result)) {
            echo"<tr><td>{$row['Spirit_ID']}</td><td>{$row['Spirit_Name']}</td><td>{$row['ABV']}<,
                <td>{$row['Type']}</td><td>{$row['Category']}</td><td>{$row['Age']}</td>
                <td>{$row['Man_Name']}</td><td>{$row['State']}</td><td>{$row['Country']}</td></tr>
        }
    }

    if (isset($_POST['Wine'])) {
        $sql = "SELECT Wine.*, Manufacture.Man_Name, Manufacture.State, Manufacture.Country \
            FROM Wine INNER JOIN Manufacture ON Wine.Manufacture=Manufacture.Man_ID;";
        $result = mysqli_query($db_connect, $sql) or die("bad query");

        echo"<table>";
        echo"<tr><th>ID</th><th>Name</th><th>ABV</th><th>Category</th><th>Year</th><th>Manufactur

        while($row=mysqli_fetch_assoc($result)) {
            echo"<tr><td>{$row['Wine_ID']}</td><td>{$row['Wine_Name']}</td><td>{$row['ABV']}</td><
```



Welcome to the Query!! Homepage

Beer — Query Beer

Spirits — Query Spirits

Wine — Query Wine

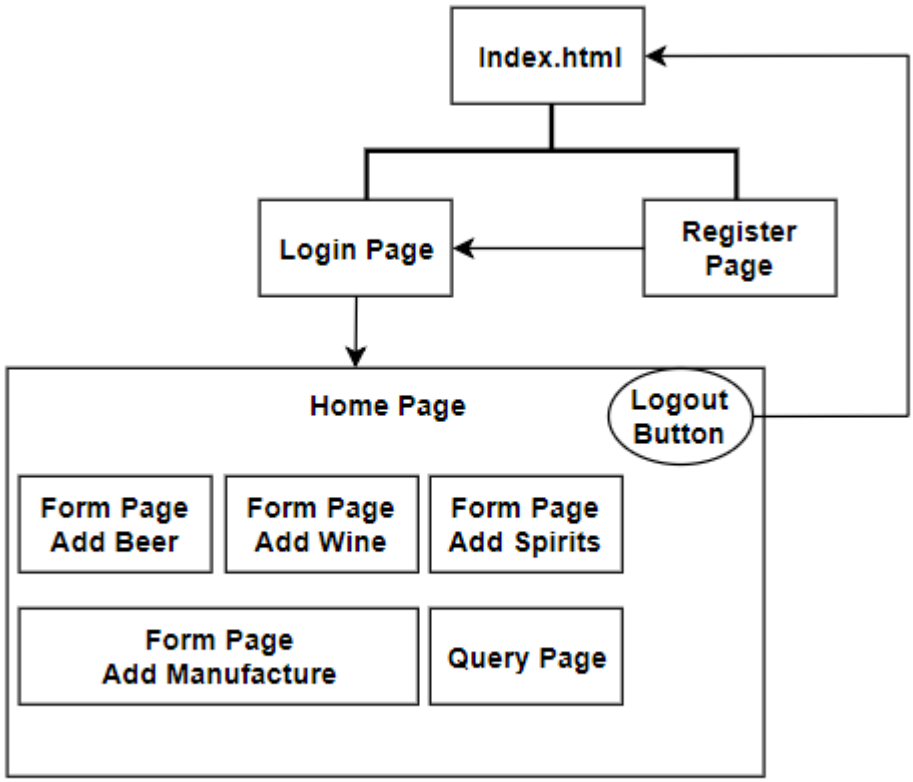
Manufacture — Query Manufacture

ID	Name	ABV	Category	Manufacture	State	Country
1	Breckenridge: Vanilla Porter	4.7	Porter	Breckenridge Brewery	Co	USA
2	Breckenridge: Oatmeal Stout	4.95	Stout	Breckenridge Brewery	Co	USA
3	Breckenridge: Christmas Ale	7.4	Seasonal	Breckenridge Brewery	Co	USA
4	Breckenridge: Twenty	7.8	Barrel Aged	Breckenridge Brewery	Co	USA
5	Stone Tangerine Express IPA	6.7	India Pale Ale	Stone Brewing	Ca	USA
6	Stone Ruination Double IPA	8.5	Double IPA	Stone Brewing	Ca	USA
7	Stone Ghost Happer IPA	6.7	India Pale Ale	Stone Brewing	Ca	USA
8	Stone Scorpion Bowl IPA	7.5	India Pale Ale	Stone Brewing	Ca	USA
9	Barrio Blonde	4.7	Blonde	Barrior Brewing Co.	Az	USA
10	Citrazona	6.6	IPA	Barrior Brewing Co.	Az	USA
11	Copperhead Pale Ale	6.2	Americal Pale Ale	Barrior Brewing Co.	Az	USA
12	Mocha Java Stout	5.5	Coffee Stout	Barrior Brewing Co.	Az	USA
13	Nolan's Porter Ale	6	Porter	Barrior Brewing Co.	Az	USA
14	Abundansea	10	Imperial Milk Stout	Arizona Wilderness Brewing Co.	Az	USA
15	Baboquivari	7.1	Belgian Blonde	Arizona Wilderness Brewing Co.	Az	USA

Website Outline

Once the pages are complete, we need a way to access the information and navigate between each form. To do this we will design a basic website layout that will allow individuals to access the appropriate forms. This will have an index page, login page (to verify users) register page for new users (this can be taken away and managed by the Admin if needed), and a Home page. The home page will be link, after the user has logged in, have all the available options for the user. These pages can be made available or unavailable based off the user's privileges. Right now, for testing purposes, the user will have access to all the pages listed below.

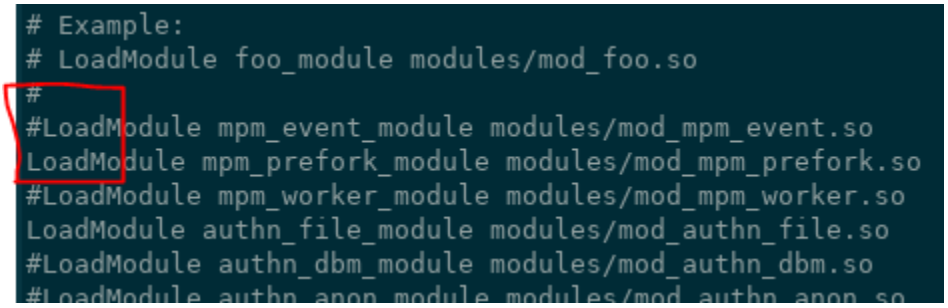
.



Failure Report

I did face some issues when first working this project. Initially when installing Linux onto my system, everything was good until I booted back into windows. Once I did that the partitions with Linux installed onto it became corrupt and I had to redo the installation process. I found out that this was due to the UEFI boot loader and the grub boot loader needing to be updated after installing Linux. Once I completed the reinstallation and updating the bootloader everything worked fine.

Another issue faced during the installation of the LAMP stack was that I had moved through it too fast and didn't complete every step. This caused my Apache server to crash after installing php because I didn't comment out a specific line in the configuration files. This caused the wrong modules to be loaded and crashed the server. It was an easy fix that could have been avoided with attention to detail upon installation.



I ran into issues with my programming. I had syntax and grammatical errors that were hard for me to find. This was largely due to me typing too fast which resulted in misspelled words/commands. These errors caused the entire program to malfunction, or not display any information.

On the bottom of each form, I used a query command to display the information only relevant to the current table based off the manufacture. I ran into the issues of it displaying repeated information over and over. Upon investigation, I found out that I need to use the SELECT DISTINCT command instead of just SELECT for the query to keep this from happening.

```
            $sqlMan = "SELECT DISTINCT Man_ID, Man_Name FROM Manufacture
JOIN Beer ON Manufacture.Man_ID=Beer.Manufacture WHERE
Man_ID=Beer.Manufacture";
            $resultMan=mysqli_query($db_connect, $sqlMan) or die("bad
query");

            echo "<tr><th>ID</th><th>Name</th></tr>";
            while ($row = mysqli_fetch_assoc($resultMan)) {
                echo"<tr><td>{$row['Man_ID']}</td><td>{$row['Man_Name']}</
td></tr>";
```

When conducting the query for the query page that joins multiple tables together in the results, I had an issue with the join statement that I was using. My JOIN statement was working but the while loop that I created to show the results were not working. I realized that this was because the table names were not unique. I used Name for both tables and it didn't know which it was supposed to display. When I changed the tables headers from Name to "Spirit_Name" and "Man_Name", the results displayed like they were supposed to.

```
if (isset($_POST['Spirits'])) {
    $sql = "SELECT Spirits.*, Manufacture.Man_Name, Manufacture.State, Manufacture.Country
     FROM Spirits INNER JOIN Manufacture ON Spirits.Manufacture=Manufacture.Man_ID;";
    $result = mysqli_query($db_connect, $sql) or die("bad query");

    echo"<table>";
    echo"<tr><th>ID</th><th>Name</th><th>ABV</th><th>Type</th><th>Category</th>
        <th>Age</th><th>Manufacture</th><th>State</th><th>Country</th></tr>";

    while($row=mysqli_fetch_assoc($result)) {
        echo"<tr><td>{$row['Spirit_ID']}</td><td>{$row['Spirit_Name']}</td>
        <td>{$row['ABV']}</td><td>{$row['Type']}</td><td>{$row['Category']}</td>
        <td>{$row['Age']}</td><td>{$row['Man_Name']}</td><td>{$row['State']}</td>
        <td>{$row['Country']}</td></tr>";
    }
}
```

Conclusion

I learned a lot during this process about mariadb, php, and apache. A large part of this was due to the fact that each application had to be installed and configured manually in the Arch Linux distro I was using. I am happy with my choice to use a LAMP stack because I have been able to find a lot of resources on the internet to help me resolve any issues I have had so far. I have learned through this process that it takes time to develop anything and sometimes you need to walk away and come back with a fresh set of eyes. Overall, I am happy with the outcome of my database and products for this presentation.

References

Manjaro Linux installation steps retrieved from: https://forum.manjaro.org/t/install-apache-mariadb-php-lamp-2016/1243/1

MariaDB guide retrieved from: https://mariadb.com/kb/en/library/a-mariadb-primer/

PHP7 tutorial from tutorialspoint.com retrieved from:

https://www.tutorialspoint.com/php7/php7_scalartype_declarations.htm

HTML/PHP.MYSQL form tutorial retrieved from:

https://www.youtube.com/watch?v=yhsVNdNF9sU

HTML/PHP/MYSQL tutorials retrieved from: https://www.w3schools.com/default.asp

How to create login system YoutTube.com retrieved from:

https://www.youtube.com/watch?v=LC9GaXkdxF8&index=4&t=0s&list=PLgJYeu29xjqMAyRRw--jmYC6wBKHb6vpb

Display query results in html table Youtube.com retrieved from:

https://www.youtube.com/watch?v=pc0otVM80Sk&index=3&t=0s&list=PLgJYeu29xjqMAyRRw--jmYC6wBKHb6vpb