|  |  |
| --- | --- |
|  |  |
| NIT1204Web Application and Server Management |  |
|  |  |
|  | Date:15/11/2024 |
|  | Project: Developing an application for Cold Drink Café  | **Student Name** | **Student ID** | **Signature** | | --- | --- | --- | | Swojan Singh Maharjan | s4676993 | SWOJAN | | Bijay Tamang | s8169144 |  | |

Contents

[NIT1204 1](#_Toc182612948)

[Web Application and Server Management 1](#_Toc182612949)

[**Date:15/11/2024** 1](#_Toc182612950)

[Project: Developing an application for Cold Drink Café 1](#_Toc182612951)

[Figures : 3](#_Toc182612952)

[1. INTRODUCTION 4](#_Toc182612953)

[2. Project Objectives: 4](#_Toc182612954)

[3. Usage Instructions: 4](#_Toc182612955)

[4. Installation/Usage Instructions 8](#_Toc182612956)

[5. Source Code 8](#_Toc182612957)

[6. Screenshots of a Database 18](#_Toc182612958)

[7. Test Report with Test Cases 19](#_Toc182612959)

[8. Group Meeting 20](#_Toc182612960)

[9. Determine the actions 20](#_Toc182612961)

[10. Conclusion 21](#_Toc182612962)

## Figures :

[Figure 1 Add Drink 4](#_Toc182612923)

[Figure 2 Add Drink with Text 5](#_Toc182612924)

[Figure 3 View Drink List 5](#_Toc182612925)

[Figure 4 Modify Drink 6](#_Toc182612926)

[Figure 5 Sort Ascending 7](#_Toc182612927)

[Figure 6 Sort Descending 7](#_Toc182612928)

[Figure 7 Database Tables 18](#_Toc182612929)

[Figure 8 Database Categories 18](#_Toc182612930)

[Figure 9 Database Products 19](#_Toc182612931)

## INTRODUCTION

This particular web application that we created is called Cold Drink Café that allows customer of this cafe to sort the list of drinks of their choice but also allow them to be able to add, remove, and update drink items in the system. It is a PHP-based tool which data is stored in a MySQL database.

This web application has few features that are listed below.

• Add New Drink

• Delete Drink

• Update Drink Name:

• View Products:

• Category Filtering

1. Project Objectives: To create a database and an application to manage the inventory of a Cold Drink Café.
2. Usage Instructions: Once you open a website it displays a list of drinks with different categories, name, and price which can be modify or delete with the help of click button that says ‘modify’ and ‘delete’. And at the very bottom of the application there are options for users to add drinks, sort drinks in ascending or descending order.
3. Add Drink

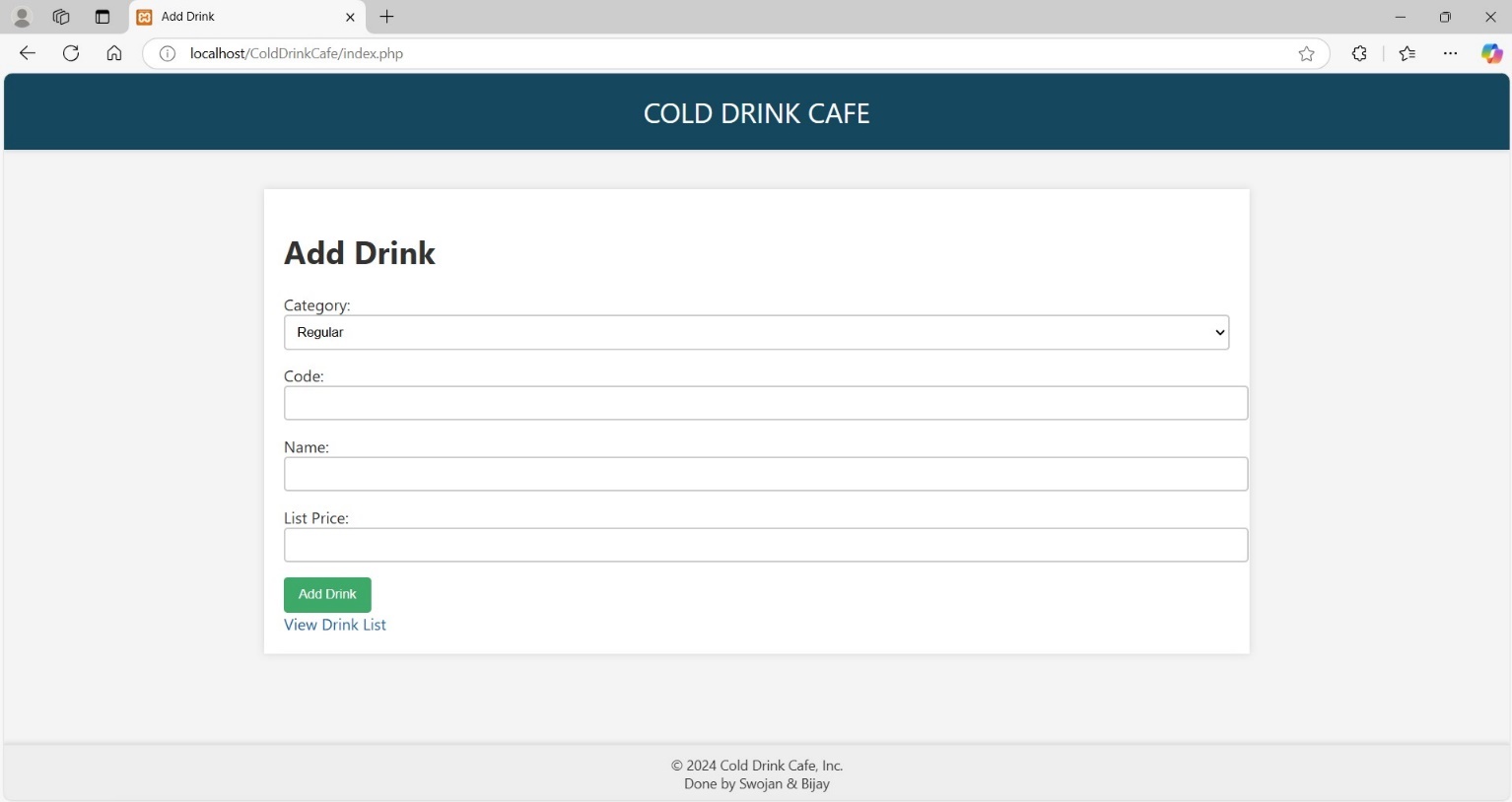


Figure 1 Add Drink

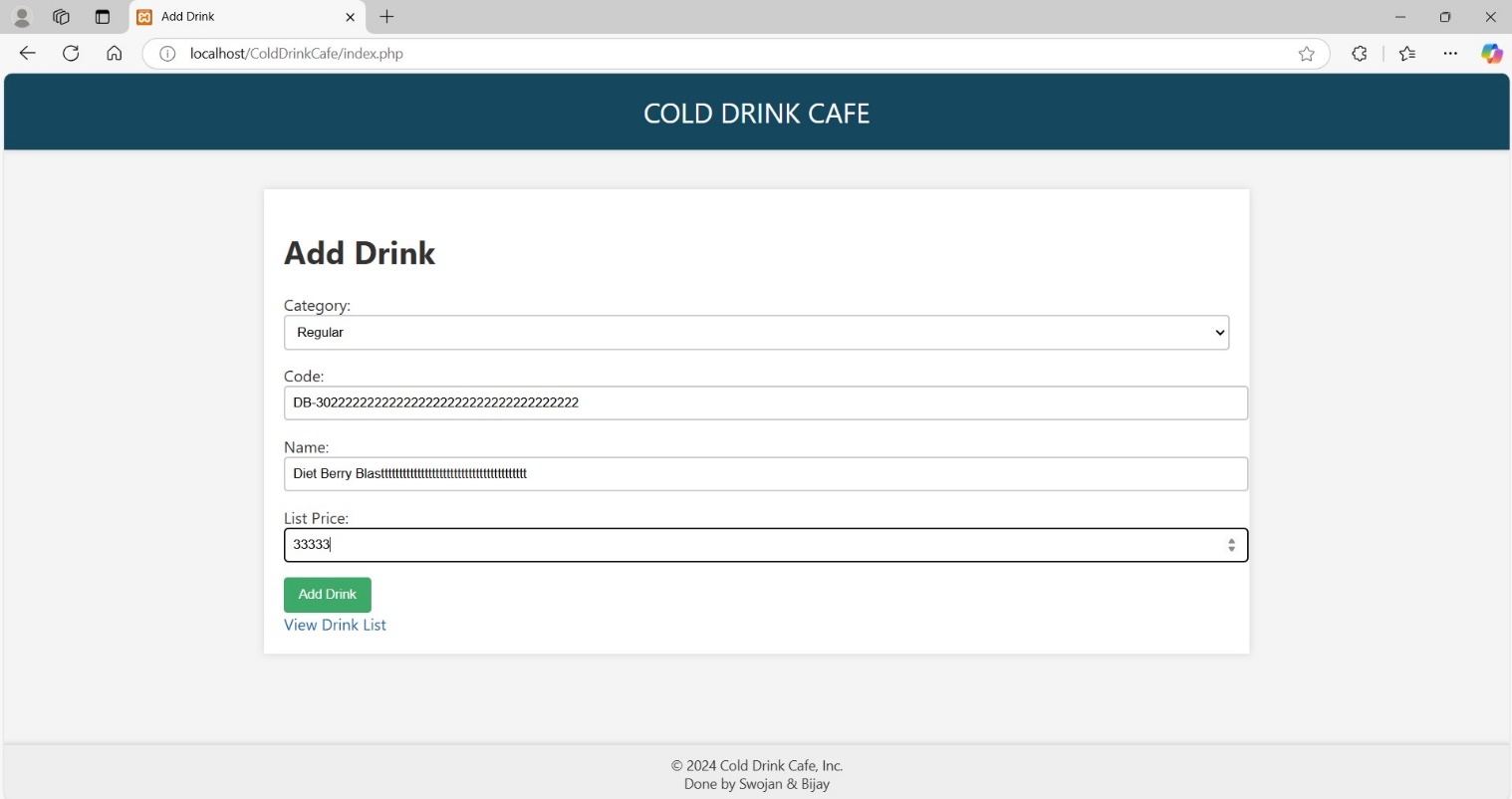


Figure 2 Add Drink with Text

Description:

This page allows users to add a new drink to the database.

Users can select the category (e.g., Regular, Zero Sugar, or Diet), provide the drink code, name, and price.

After submitting, the drink is saved in the database and appears on the "View Drink List" page.

1. View Drink List:

A screenshot of a computer

Description automatically generated

Figure 3 View Drink List

Description:

This page lists all available drinks in the selected category.

Drinks are displayed in a table format with their name, price, and action buttons.

Users can sort the drinks by price in ascending or descending order using the sorting buttons.

Navigation to the "Add Drink" page is provided via a link.

1. Modify Drink:

A screenshot of a computer

Description automatically generated

Figure 4 Modify Drink

Description:

This page allows users to edit the details of an existing drink.

Users can modify the name or price of the drink.

After saving changes, the updated drink appears in the "View Drink List."

1. Delete Drink

Description:

Users can delete a specific drink from the database by clicking the "Delete" button.

Once deleted, the drink is removed from the list and no longer appears on the page.

1. Category Navigation

Description:

Categories are displayed as clickable links on the "View Drinks" page.

Selecting a category filters the drink list to show only the drinks belonging to that category.

1. Sorting Functionality:

A screenshot of a computer

Description automatically generated

Figure 5 Sort Ascending

A screenshot of a menu

Description automatically generated

Figure 6 Sort Descending

Description:

Users can sort drinks by price in ascending or descending order.

Sorting updates the table dynamically to reflect the chosen order.

## Installation/Usage Instructions

* Install XAMPP or any similar local server environment.
* Clone the project or download it as a ZIP file.
* Extract the project files to the htdocs folder in your XAMPP directory.
* Open phpMyAdmin and import the provided cold\_drink\_cafe.sql file to set up the database.
* Start Apache and MySQL in XAMPP.
* Open your browser and navigate to <http://localhost/ColdDrinkCafe>.

## Source Code

1. index.php

<?php

// Include the database connection script

include 'includes/db.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Add Drink</title>

    <!-- Link to the external CSS for styling -->

    <link rel="stylesheet" href="style.css">

</head>

<body>

<header>Cold Drink Cafe</header>

<div class="container">

    <h1>Add Drink</h1>

    <!-- Form for adding a new drink -->

    <form action="addDrink.php" method="post">

        <!-- Dropdown for selecting the drink category -->

        <label for="category">Category:</label>

        <select id="category" name="category">

            <?php

            // Fetch all categories from the database and display them as options

            $stmt = $pdo->query('SELECT \* FROM Categories');

            while ($row = $stmt->fetch()) {

                echo "<option value='{$row['categoryID']}'>{$row['categoryName']}</option>";

            }

            ?>

        </select>

        <!-- Input field for the drink code -->

        <label for="code">Code:</label>

        <input type="text" id="code" name="code">

        <!-- Input field for the drink name -->

        <label for="name">Name:</label>

        <input type="text" id="name" name="name">

        <!-- Input field for the drink price -->

        <label for="price">List Price:</label>

        <input type="number" id="price" name="price" step="0.01">

        <!-- Submit button for the form -->

        <input type="submit" value="Add Drink">

    </form>

    <!-- Link to view the list of drinks -->

    <a href="viewDrinks.php">View Drink List</a>

</div>

<!-- Footer section with copyright and custom text -->

<footer>© 2024 Cold Drink Cafe, Inc. <br> Done by Swojan & Bijay</footer>

</body>

</html>

1. modifyDrink.php

<?php

// Include the database connection setup

include 'includes/db.php';

// Check if the form has been submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    // Get the posted form data

    $productID = $\_POST['id'];

    $productName = $\_POST['name'];

    $listPrice = $\_POST['price'];

    // Prepare an SQL statement to update drink details

    $sql = "UPDATE Products SET productName = ?, listPrice = ? WHERE productID = ?";

    $stmt = $pdo->prepare($sql);

    // Execute the statement with provided data

    $stmt->execute([$productName, $listPrice, $productID]);

    // Redirect to the drink list page after updating

    header("Location: viewDrinks.php");

    exit();

} elseif (isset($\_GET['id'])) { // Check if a drink ID was passed in the URL

    // Fetch the drink data to be modified

    $productID = $\_GET['id'];

    $sql = "SELECT \* FROM Products WHERE productID = ?";

    $stmt = $pdo->prepare($sql);

    $stmt->execute([$productID]);

    $product = $stmt->fetch();

    // If no drink is found with the ID, redirect to the drink list

    if (!$product) {

        header("Location: viewDrinks.php");

        exit();

    }

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Modify Drink</title>

    <link rel="stylesheet" href="style.css"> <!-- Link to the external CSS -->

</head>

<body>

<header>Cold Drink Cafe</header>

<div class="container">

    <h1>Modify Drink</h1>

    <!-- Form for submitting drink modifications -->

    <form action="modifyDrink.php" method="post">

        <!-- Hidden input to hold the drink ID -->

        <input type="hidden" name="id" value="<?= htmlspecialchars($product['productID']) ?>">

        <label for="name">Name:</label>

        <!-- Input for editing the drink name -->

        <input type="text" name="name" value="<?= htmlspecialchars($product['productName']) ?>"><br>

        <label for="price">Price:</label>

        <!-- Input for editing the drink price -->

        <input type="number" name="price" value="<?= htmlspecialchars($product['listPrice']) ?>" step="0.01"><br>

        <input type="submit" value="Update Drink">

    </form>

    <!-- Link to view the list of all drinks -->

    <a href="viewDrinks.php">View Drink List</a>

</div>

<footer>© 2024 Cold Drink Cafe, Inc. <br> Done by Swojan & Bijay</footer>

</body>

</html>

<?php

} else {

    // Redirect to the drink list if no drink ID is specified

    header("Location: viewDrinks.php");

    exit();

}

?>

1. deleteDrink.php

<?php

include 'includes/db.php'; // Include the database connection

if (isset($\_GET['id'])) {

    $productID = $\_GET['id'];

    // SQL query to delete the drink from the Products table

    $sql = "DELETE FROM Products WHERE productID = ?";

    $stmt = $pdo->prepare($sql);

    $stmt->execute([$productID]);

    // Redirect to the drink list page

    header("Location: viewDrinks.php");

    exit();

} else {

    // Redirect to the drink list if no ID is specified

    header("Location: viewDrinks.php");

    exit();

}

?>

1. viewDrinks.php

<?php

include 'includes/db.php'; // Include database connection

// Initialize categories and drinks arrays

$categories = [];

$drinks = [];

// Fetch all categories from the database

try {

    $categoriesStmt = $pdo->query('SELECT \* FROM Categories');

    $categories = $categoriesStmt->fetchAll();

} catch (Exception $e) {

    error\_log("Error fetching categories: " . $e->getMessage()); // Log error message

    // Optionally handle user-facing error feedback here

}

// Set the current category to the first available or from the user selection

$currentCategory = $categories[0]['categoryID'] ?? null;

if (isset($\_GET['category']) && in\_array($\_GET['category'], array\_column($categories, 'categoryID'))) {

    $currentCategory = $\_GET['category'];

}

// Fetch drinks for the selected category

if ($currentCategory) {

    try {

        $sql = "SELECT \* FROM Products WHERE categoryID = ? ORDER BY productName";

        $drinksStmt = $pdo->prepare($sql);

        $drinksStmt->execute([$currentCategory]);

        $drinks = $drinksStmt->fetchAll();

    } catch (Exception $e) {

        error\_log("Error fetching drinks: " . $e->getMessage()); // Log error message

        // Optionally handle user-facing error feedback here

    }

}

// Implement sorting functionality if requested

if (isset($\_GET['sort']) && in\_array($\_GET['sort'], ['asc', 'desc'])) {

    $sortOrder = $\_GET['sort'] === 'asc' ? 'ASC' : 'DESC';

    $sql = "SELECT \* FROM Products WHERE categoryID = ? ORDER BY listPrice $sortOrder";

    $drinksStmt = $pdo->prepare($sql);

    $drinksStmt->execute([$currentCategory]);

    $drinks = $drinksStmt->fetchAll();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>View Drinks</title>

    <link rel="stylesheet" href="style.css"> <!-- Include CSS for styling -->

</head>

<body>

<header>Cold Drink Cafe</header>

<div class="container">

    <h1>Drink List</h1>

    <!-- Categories Navigation -->

    <div class="categories">

        <?php foreach ($categories as $category): ?>

            <a href="?category=<?= $category['categoryID'] ?>" class="<?= $currentCategory == $category['categoryID'] ? 'active' : '' ?>">

                <?= htmlspecialchars($category['categoryName']) ?>

            </a>

        <?php endforeach; ?>

    </div>

    <!-- Drinks Table -->

    <table>

        <thead>

            <tr>

                <th>Name</th>

                <th>Price</th>

                <th>Actions</th>

            </tr>

        </thead>

        <tbody>

            <?php foreach ($drinks as $drink): ?>

                <tr>

                    <td><?= htmlspecialchars($drink['productName']) ?></td>

                    <td><?= htmlspecialchars($drink['listPrice']) ?></td>

                    <td>

                        <a href="modifyDrink.php?id=<?= $drink['productID'] ?>">Modify</a>

                        <a href="deleteDrink.php?id=<?= $drink['productID'] ?>">Delete</a>

                    </td>

                </tr>

            <?php endforeach; ?>

        </tbody>

    </table>

    <!-- Sorting Buttons -->

    <div class="sorting">

        <a href="?category=<?= $currentCategory ?>&sort=asc">Sort Drink in Ascending Order</a>

        <a href="?category=<?= $currentCategory ?>&sort=desc">Sort Drink in Descending Order</a>

    </div>

    <!-- Link to add a new drink -->

    <a href="index.php">Add Drink</a>

</div>

<footer>© 2024 Cold Drink Cafe, Inc. <br> Done by Swojan & Bijay</footer>

</body>

</html>

1. addDrink.php

<?php

include 'includes/db.php'; // Include the database connection

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    // Retrieve form data

    $categoryID = $\_POST['category'];

    $productCode = $\_POST['code'];

    $productName = $\_POST['name'];

    $listPrice = $\_POST['price'];

    // SQL query to insert data into Products table

    $sql = "INSERT INTO Products (categoryID, productCode, productName, listPrice) VALUES (?, ?, ?, ?)";

    $stmt = $pdo->prepare($sql);

    $stmt->execute([$categoryID, $productCode, $productName, $listPrice]);

    // Redirect to the main page or wherever you wish

    header("Location: index.php");

    exit();

}

?>

1. style.css

/\* General Reset and Font Settings \*/

body, html {

    margin: 0;

    padding: 0;

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

    background-color: #f4f4f4;

    color: #333;

}

/\* Header styles with more professional appearance \*/

header {

    background-color: #02475e; /\* Deep blue for better contrast \*/

    color: white;

    text-align: center;

    font-size: 28px;

    padding: 20px 0;

    text-transform: uppercase;

    box-shadow: 0 2px 4px rgba(0,0,0,0.1);

}

/\* Footer with fixed positioning and more subtle styling \*/

footer {

    background-color: #eeeeee; /\* Soft gray background \*/

    color: #333;

    text-align: center;

    padding: 10px 0;

    font-size: 14px;

    border-top: 1px solid #ddd;

    position: fixed;

    bottom: 0;

    width: 100%;

    box-shadow: 0 -2px 4px rgba(0,0,0,0.1);

}

/\* Container for better spacing and alignment \*/

.container {

    width: 80%;

    max-width: 960px;

    margin: 40px auto;

    background-color: #fff;

    padding: 20px;

    box-shadow: 0 0 8px rgba(0,0,0,0.12);

}

/\* Improved table styling for readability \*/

table {

    width: 100%;

    border-collapse: collapse;

    margin-top: 20px;

    border-radius: 8px;

    overflow: hidden;

}

th, td {

    border-bottom: 1px solid #ddd;

    padding: 12px 15px;

    text-align: left;

}

th {

    background-color: #0284c7; /\* Slightly lighter blue for headers \*/

    color: #ffffff;

}

/\* Input and Select box styling for cleaner interface \*/

input[type="text"], input[type="number"], select {

    width: 100%;

    padding: 8px;

    margin-bottom: 15px;

    border: 2px solid #ccc;

    border-radius: 4px;

}

/\* Button styles with hover effects for interactivity \*/

input[type="submit"], a.button-link {

    background-color: #04aa6d; /\* Green for primary actions \*/

    color: white;

    padding: 10px 15px;

    text-align: center;

    display: inline-block;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    text-decoration: none;

}

input[type="submit"]:hover, a.button-link:hover {

    background-color: #037d56; /\* Darker green on hover \*/

    box-shadow: 0 2px 4px rgba(0,0,0,0.2);

}

/\* Link styles for consistency and better usability \*/

a {

    color: #0165a8; /\* Consistent blue for links \*/

    text-decoration: none;

}

a:hover {

    text-decoration: underline;

}

/\* Responsive adjustments for better mobile viewing \*/

@media (max-width: 768px) {

    .container {

        width: 95%;

        margin: 20px auto;

        padding: 15px;

    }

    header {

        font-size: 24px;

        padding: 15px 0;

    }

    footer {

        font-size: 12px;

    }

}

/\* Categories Navigation Styling \*/

.categories {

    margin-bottom: 20px;

    padding-bottom: 10px;

    border-bottom: 1px solid #ccc; /\* Visual separation \*/

}

.categories a {

    font-size: 16px;

    padding: 5px 10px;

    background-color: #f0f0f0;

    border: 1px solid #ddd;

    border-radius: 4px;

    margin-right: 5px;

}

.categories a.active {

    background-color: #0284c7;

    color: white;

    border-color: #026ba1;

}

/\* Sorting and Add Drink Button Styling \*/

.sorting, .add-drink {

    margin: 20px 0;

    display: flex;

    justify-content: space-between;

}

.sorting a, .add-drink a {

    padding: 8px 15px;

    background-color: #04aa6d;

    color: white;

    border-radius: 4px;

    text-decoration: none;

}

.sorting a:hover, .add-drink a:hover {

    background-color: #037d56;

}

1. db.php

<?php

$host = 'localhost'; // Server where your MySQL is hosted

$dbname = 'ColdDrinkCafe'; // Database name

$username = 'root'; // Your MySQL username, default is 'root' in XAMPP

$password = ''; // Your MySQL password, default is empty in XAMPP

$dsn = "mysql:host=$host;dbname=$dbname;charset=utf8mb4"; // Data Source Name

$options = [

    PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

    PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,

    PDO::ATTR\_EMULATE\_PREPARES => false,

];

try {

    $pdo = new PDO($dsn, $username, $password, $options);

} catch (PDOException $e) {

    throw new PDOException($e->getMessage(), (int)$e->getCode());

}

?>

## Screenshots of a Database

A screenshot of a computer

Description automatically generated

Figure 7 Database Tables

A screenshot of a computer

Description automatically generated

Figure 8 Database Categories

A screenshot of a computer

Description automatically generated

Figure 9 Database Products

## Test Report with Test Cases

Create a table of test cases to verify functionality.

| **Test ID** | **Test Case** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- |
| TC001 | Add a drink | Drink is added and visible on the list | Works as expected | Pass |
| TC002 | Modify a drink | Updates the drink details in the database | Works as expected | Pass |
| TC003 | Delete a drink | Removes the drink from the list and database | Works as expected | Pass |
| TC004 | Sort drinks in ascending order | Drinks are displayed by ascending price | Works as expected | Pass |
| TC005 | View drinks filtered by category | Displays only drinks in the selected category | Works as expected | Pass |

## Group Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| **Title of Meeting:** | Cold Drink Café App | | |
| **Meeting Objectives:** | To discuss all the information that we got while developing a database and an application. | | |
| **Minutes Of Meeting:** | 120 Minutes | | |
| **Date Of Meeting:** | 15th of November | **Meeting Commenced:** | 12:00 pm |
| **Location:** | Victoria University | **Meeting Ended:** | 2:00pm |

## *Determine the actions*

|  |  |  |
| --- | --- | --- |
| **Actions** | **Done By** | **Date** |
| Research on the project | All | 15th Nov |
| Create Database and test the File | Bijay | 15th Nov |
| Create Css, PHP, Httml, and test the program | Swojan | 15th Nov |
| Finalising the project | all | 15th Nov |

Next Meetings on: Tuesday the 19th Nov 2024

## Conclusion

The Cold Drink Cafe Web Application allowed us to apply web development skills to create a functional system for managing drinks and categories. It enhanced our understanding of PHP, MySQL, and CSS while improving teamwork and problem-solving. This project reflects our ability to build practical and user-friendly web solutions, preparing us for future challenges.