Jethro J. Manzanillo
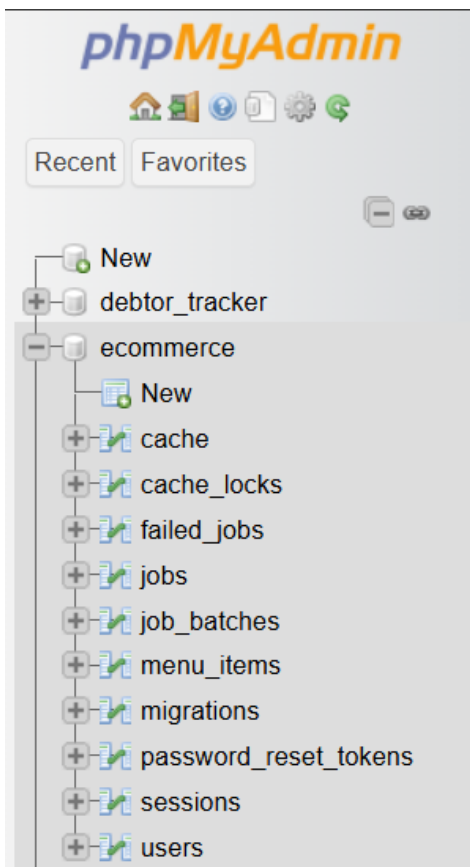**BSIT - 3C**
Web Development 2

Setting Up the Database:

- We established a connection to the database using MyPHPAdmin. A database named "ecommerce" was created, and the name was added to the `.env` file under `DB_DATABASE`.

Defining the MenuItem Model:

- The `MenuItem` model was generated with the command `php artisan make:model MenuItem`. It facilitates interaction with the database to display menu data on the website.

```php
FinalProject > app > Models > 🐘 MenuItem.php > ...
1    <?php
2
3    namespace App\Models;
4
5    use Illuminate\Database\Eloquent\Factories\HasFactory;
6    use Illuminate\Database\Eloquent\Model;
7
     9 references | 0 implementations
8    class MenuItem extends Model
9    {
10       use HasFactory;
11   }
12
```

Implementing the MenuController:

- The `MenuController` was created to handle logic for retrieving menu items from the database and passing them to the view.

```php
FinalProject > app > Http > Controllers > 🐘 MenuController.php > ...
1    <?php
2
3    namespace App\Http\Controllers;
4
5    use Illuminate\Http\Request;
6    use App\Models\MenuItem;
7
     2 references | 0 implementations
8    class MenuController extends Controller
9    {
         1 reference | 0 overrides
10       public function showMenu($username): View
11       {
12           // Fetch the first 6 menu items from the database
13           $menuItems = MenuItem::take(value: 6)->get();
14
15           return view(view: 'menu', data: [
16               'menuItems' => $menuItems,
17               'username' => $username
18           ]);
19       }
20   }
21
```

Creating the MenuItemSeeder:

- The `MenuItemSeeder` inserts sample data into the `menu_items` table, including attributes like `image_path`, `name`, `price`, and `rating`.
- After execution, the `ecommerce` database is populated with this sample data, which can be dynamically retrieved for display.

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\MenuItem;

class MenuItemSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        MenuItem::create(attributes: [
            'image' => 'css/bfast.jpg',
            'name' => 'Classic Droissants',
            'price' => '₱150',
            'rating' => 4,
        ]);

        MenuItem::create(attributes: [
            'image' => 'css/img2.jpg',
            'name' => 'Lavander Latte',
            'price' => '₱120',
            'rating' => 4,
        ]);

        MenuItem::create(attributes: [
            'image' => 'css/img3.jpg',
            'name' => 'Banana Walnut Muffin',
            'price' => '₱100',
            'rating' => 4,
        ]);

        MenuItem::create(attributes: [
```
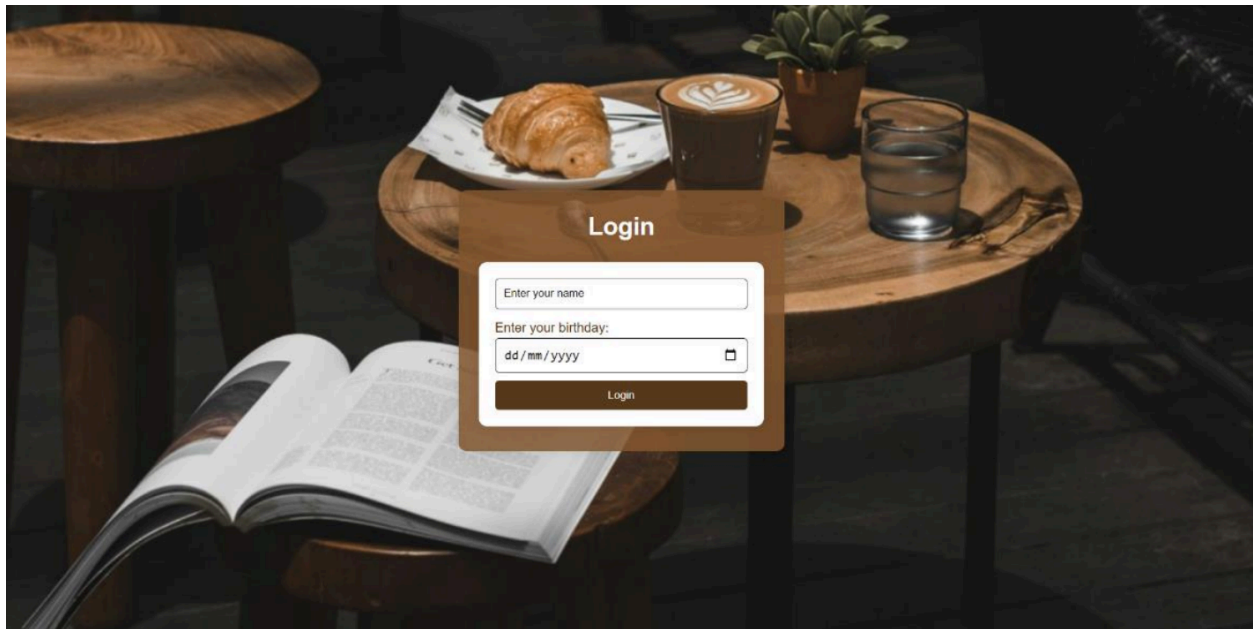
	Table ▲	Action						Rows ❓	Type	Collation	Size	Overhead
☐	cache	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	cache_locks	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	failed_jobs	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
☐	jobs	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
☐	job_batches	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	menu_items	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	migrations	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	password_reset_tokens	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	sessions	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	1	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
☐	users	⭐	📄 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	⛔ Drop	3	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
	10 tables	Sum							14	InnoDB	utf8mb4_general_ci	224.0 KiB	0 B

Website Interfaces:

Login Page:

Menu Page: