

Menu Controller Logic

- **Purpose:** The `MenuController` manages data for the menu page. It simulates fetching menu items from a database and limits the display to six items.
- **Key Points:**
 - **Simulated Data:** Menu items are stored in an array, each with `image`, `name`, `price`, and `rating`.

```
<?php

namespace App\Http\Controllers;

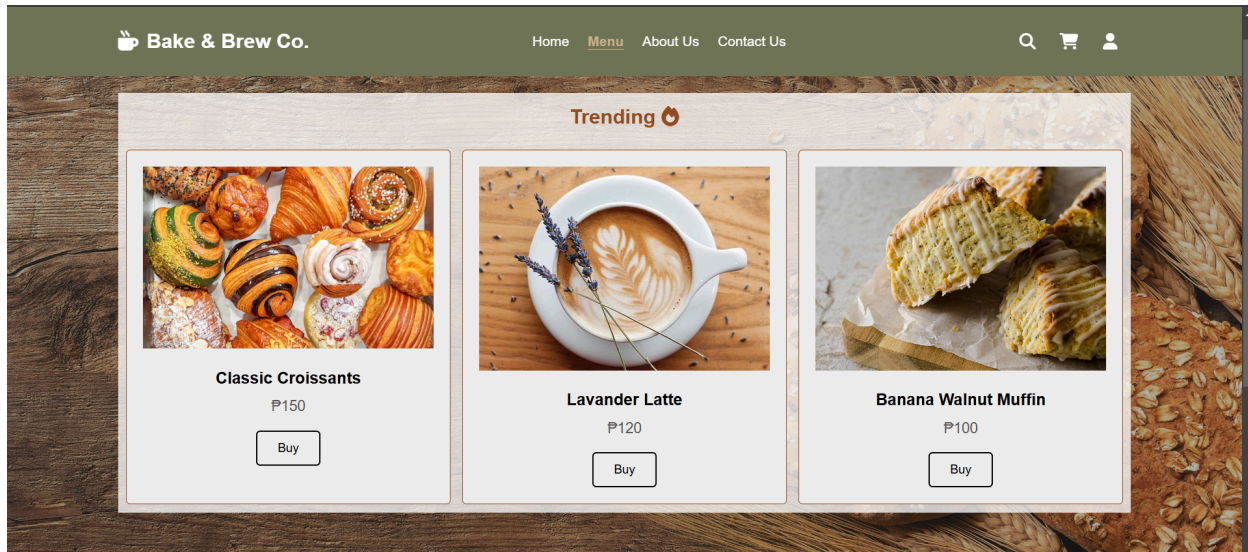
use Illuminate\Http\Request;

2 references | 0 implementations
class MenuController extends Controller
{
    1 reference | 0 overrides
    public function showMenu($username): mixed
    {
        // Simulate data from a database
        $menuItems = [
            [
                'image' => 'css/bfast.jpg',
                'name' => 'Classic Croissants',
                'price' => 'P150',
                'rating' => 4,
            ],
        ],
```

- **Limiting Items:** A `for` loop iterates over the array to limit the items to three. This approach uses the `min()` function to ensure no out-of-bounds errors occur if the total number of items is fewer than three.

```
// Limit to 6 items using a for loop
$limitedItems = [];
for ($i = 0; $i < min(value: count(value: $menuItems), values: 3); $i++) {
    $limitedItems[] = $menuItems[$i];
}

return view('menu', data: [
    'menuItems' => $limitedItems,
    'username' => $username
]);
```



- **Data Passing:** Data is passed to the menu Blade template using the `view()` function, along with a `username` parameter.

```

menu.blade.php X
resources > views > menu.blade.php
1  @extends(view: 'Components.Layout', data: ['username' => $username])
2
3  @section(section: 'title', content: 'Menu')
4
5  @section(section: 'content')
6  <main class="content-main">
7  <section class="Trending">
8    <div class="heading"><span></span></div>
9    <div class="Trending-box">
10     <div class="bfast">Trending <span><i class="fa-solid fa-fire"></i></span></div>
11     <div class="first-section">
12       @foreach ($menuItems as $item)
13       <div class="firstsection-box">
14         
15         <h3>{{ $item['name'] }}</h3>
16         <div class="price">{{ $item['price'] }}</div>
17         <a href="#" class="check-btn">Buy</a>
18       </div>
19     @endforeach
20   </div>
21 </section>
22 </main>
23 @endsection
24
25

```

- aboutus.blade.php
- accessDenied.blade.php
- contactus.blade.php
- goTologin.blade.php
- homepage.blade.php
- menu.blade.php
- restrictedArea.blade.php

Parameter Handling

- The `username` is captured from the URL (e.g., `/menu/Jethro`) and passed as a variable to both the controller and view.
- This allows dynamic personalization of the page, e.g., "Welcome, Jethro!"

Routes

- The `/menu/{username}` route is registered using the `MenuController` and linked to the `showMenu` method.

```
Route::get(uri: '/menu/{username}', action: [MenuController::class, 'showMenu'])->name(name: 'menu');
```

Things I Learned While Working on This Laravel Task:

Data Passing to Views: Passing variables to Blade templates using the `view()` function was a key skill I developed. I also learned how to loop through arrays in a Blade view using directives like `@foreach`.