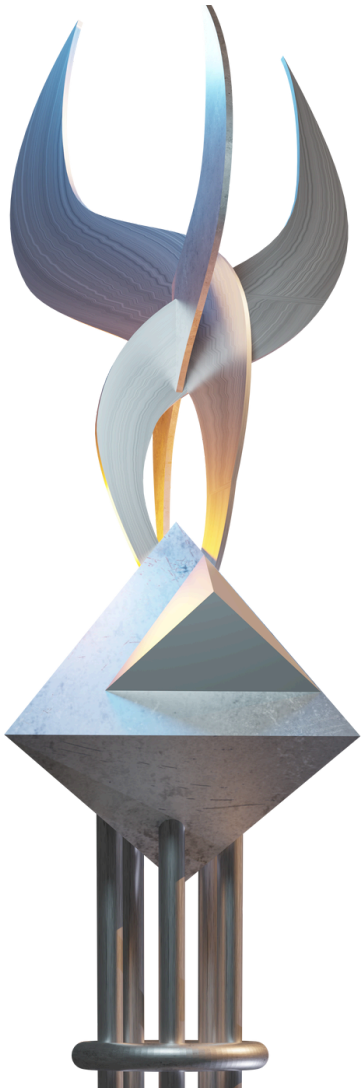
Bicol University
College of Science
Legazpi City, Albay

It Elect. 1 - Web Depelopment 1

LAB 5



Soreda, John Russel N.
BSIT - 3C

PART 1: CREATE AND REGISTER CONTROLLERS

HomePageController.php

Group_2_Lab5_WD-master > app > Http > Controllers > HomePageController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomePageController extends Controller
{
    public function index($username = 'Guest')
    {
        return view('homepage', ['username' => $username]);
    }

    public function accessDenied()
    {
        return view('accessDenied');
    }
}
```

- **index($username** = 'Guest') method: This method loads the homepage. It accepts a parameter called $username, which is passed to the view as an array. If no username is provided, it defaults to 'Guest'.
- **accessDenied() method:** This method displays an accessDenied view when a user doesn't meet certain requirements, such as age limits or missing necessary information.

MenuController.php

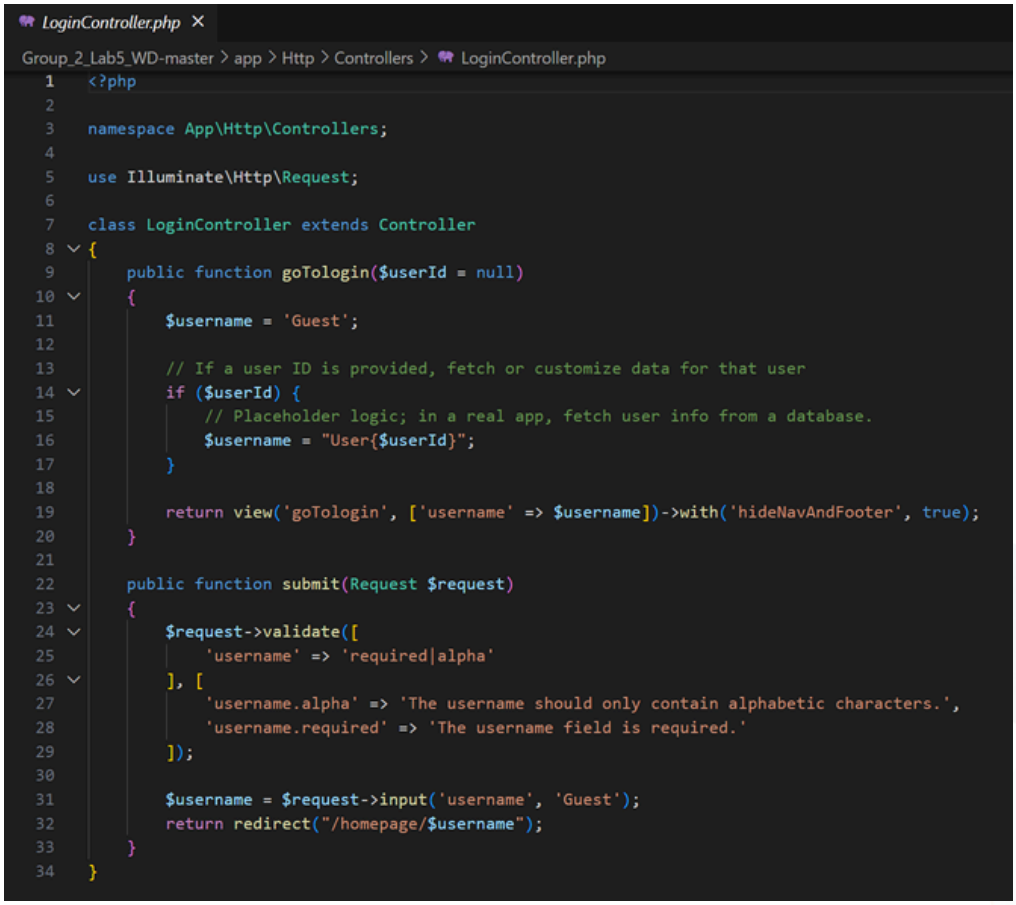
Group_2_Lab5_WD-master > app > Http > Controllers > MenuController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class MenuController extends Controller
{
    public function index($username)
    {
        return view('menu', ['username' => $username]);
    }
}
```

- **index($username) method:** This controller method accepts a dynamic parameter called $username and passes it to the corresponding view (like menu, about us, or contact us). This allows each view to show personalized content based on the username provided in the URL.

PART 1: CREATE AND REGISTER CONTROLLERS

LoginController.php



```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class LoginController extends Controller
{
    public function goTologin($userId = null)
    {
        $username = 'Guest';

        // If a user ID is provided, fetch or customize data for that user
        if ($userId) {
            // Placeholder logic; in a real app, fetch user info from a database.
            $username = "User{$userId}";
        }

        return view('goTologin', ['username' => $username])->with('hideNavAndFooter', true);
    }

    public function submit(Request $request)
    {
        $request->validate([
            'username' => 'required|alpha'
        ], [
            'username.alpha' => 'The username should only contain alphabetic characters.',
            'username.required' => 'The username field is required.'
        ]);

        $username = $request->input('username', 'Guest');
        return redirect("/homepage/$username");
    }
}
```

- The return view('goTologin', ['username' => $username])->with('hideNavAndFooter', true); method displays the login page. It allows the user to log in as either a guest or a specific user. If a $userId is provided, it sets a personalized username; otherwise, it defaults to 'Guest'.
- The submit(Request $request) method processes the form from the login page. It checks if the username contains only letters. If the username is valid, it redirects the user to the homepage with that username.

RestrictedAreaController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class RestrictedAreaController extends Controller
{
    public function index($username)
    {
        return view('restrictedArea', ['username' => $username]);
    }
}
```

- The index($username) controller loads the restricted area page. It takes a dynamic $username parameter to show personalized content in the restrictedArea view.

PART 1: CREATE AND REGISTER CONTROLLERS

AboutUsController.php

AboutUsController.php

Group_2_Lab5_WD-master > app > Http > Controllers > AboutUsController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AboutUsController extends Controller
{
    public function index($username)
    {
        return view('aboutus', ['username' => $username]);
    }
}
```

- This Controller uses the index($username) method to load the **About Us** page, passing the provided username to the aboutus view for personalized content.

ContactUsController.php

ContactUsController.php

Group_2_Lab5_WD-master > app > Http > Controllers > ContactUsController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ContactUsController extends Controller
{
    public function index($username)
    {
        return view('contactus', ['username' => $username]);
    }
}
```

- The ContactUsController class has an index method that accepts a $username parameter and returns the contactus view. This view uses the $username data to display personalized content, like a welcome message or user-specific details, on the Contact Us page.

PART 1: CREATE AND REGISTER CONTROLLERS

HomePageController.php

Group_2_Lab5_WD-master > app > Http > Controllers > HomePageController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomePageController extends Controller
{
    public function index($username = 'Guest')
    {
        return view('homepage', ['username' => $username]);
    }

    public function accessDenied()
    {
        return view('accessDenied');
    }
}
```

- **index($username** = 'Guest') method: This method loads the homepage. It accepts a parameter called $username, which is passed to the view as an array. If no username is provided, it defaults to 'Guest'.
- **accessDenied() method:** This method displays an accessDenied view when a user doesn't meet certain requirements, such as age limits or missing necessary information.

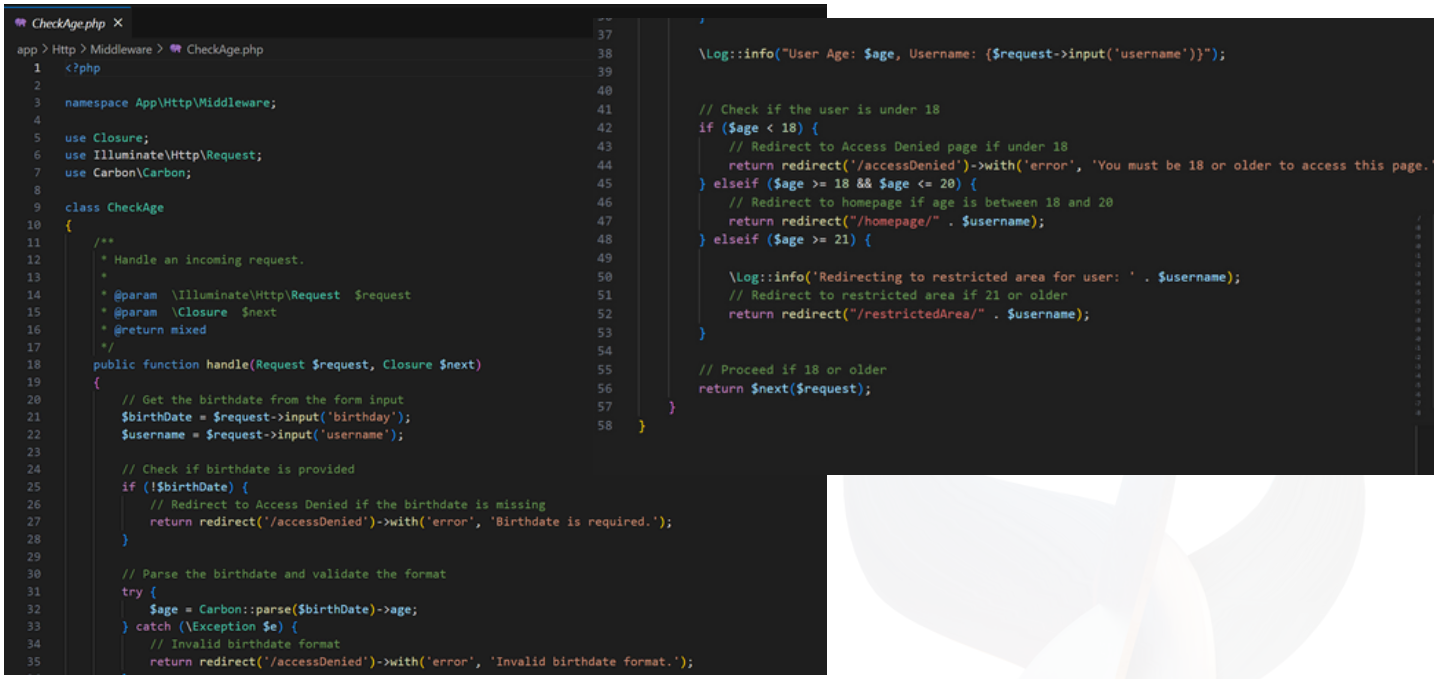
Group_2_Lab5_WD-master > app > Http > Controllers > MenuController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class MenuController extends Controller
{
    public function index($username)
    {
        return view('menu', ['username' => $username]);
    }
}
```

- **index($username) method:** This controller method accepts a dynamic parameter called $username and passes it to the corresponding view (like menu, about us, or contact us). This allows each view to show personalized content based on the username provided in the URL.

PART 2: ASSIGNING ROUTE TO CONTROLLER

Middleware | CheckAge.php

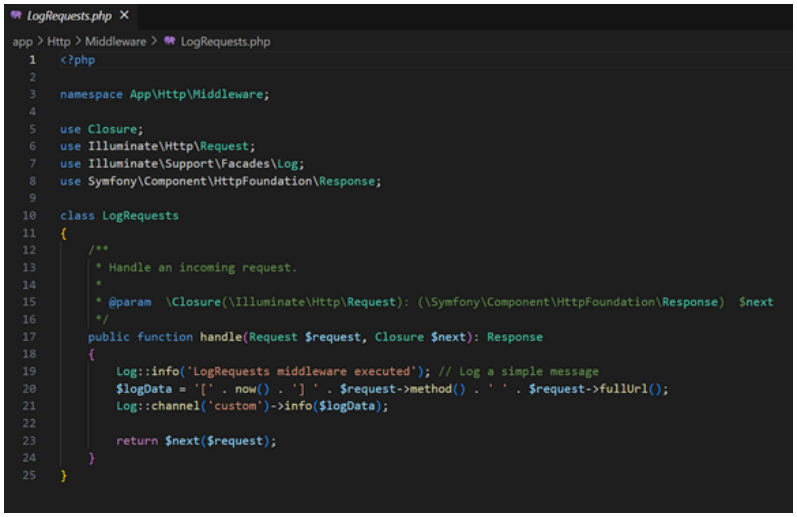


```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Carbon\Carbon;

class CheckAge
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle(Request $request, Closure $next)
    {
        // Get the birthdate from the form input
        $birthDate = $request->input('birthday');
        $username = $request->input('username');

        // Check if birthdate is provided
        if (!$birthDate) {
            // Redirect to Access Denied if the birthdate is missing
            return redirect('/accessDenied')->with('error', 'Birthdate is required.');
        }

        // Parse the birthdate and validate the format
        try {
            $age = Carbon::parse($birthDate)->age;
        } catch (\Exception $e) {
            // Invalid birthdate format
            return redirect('/accessDenied')->with('error', 'Invalid birthdate format.');
        }

        \Log::info("User Age: $age, Username: {$request->input('username')}");


        // Check if the user is under 18
        if ($age < 18) {
            // Redirect to Access Denied page if under 18
            return redirect('/accessDenied')->with('error', 'You must be 18 or older to access this page.');
        } elseif ($age >= 18 && $age <= 20) {
            // Redirect to homepage if age is between 18 and 20
            return redirect("/homepage/" . $username);
        } elseif ($age >= 21) {

            \Log::info('Redirecting to restricted area for user: ' . $username);
            // Redirect to restricted area if 21 or older
            return redirect("/restrictedArea/" . $username);
        }

        // Proceed if 18 or older
        return $next($request);
    }
}
```

The CheckAge middleware's handle(Request $request, Closure $next) function verifies if a user meets the age requirement to access certain pages. It determines the user's age based on their birthdate from the request and then redirects them accordingly:

- **Users under 18 are redirected to an access denied page.**
- **Users aged 18 to 20 are redirected to the homepage.**
- **Users aged 21 and above are redirected to a restricted area.**
- **If none of these conditions apply, the request proceeds as usual.**

Middleware | LogRequests.php



```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Symfony\Component\HttpFoundation\Response;

class LogRequests
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        Log::info('LogRequests middleware executed'); // Log a simple message
        $logData = '[' . now() . '] ' . $request->method() . ' ' . $request->fullUrl();
        Log::channel('custom')->info($logData);

        return $next($request);
    }
}
```

The LogRequest middleware's handle(Request $request, Closure $next) function logs every incoming HTTP request, capturing its method and full URL. This helps with debugging and monitoring user activity in the application.

PART 2: ASSIGNING ROUTE TO CONTROLLER

HTTP | Kernel.php

Kernel.php

app > Http > Kernel.php

```php
<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    protected $middleware = [
        // Global HTTP middleware
        \App\Http\Middleware\LogRequests::class, // Register LogRequests as global middleware
    ];

    protected $middlewareGroups = [
        'web' => [
            \App\Http\Middleware\EncryptCookies::class,
            \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            \App\Http\Middleware\CheckAge::class, // Add CheckAge to web middleware group


        ],

        'api' => [
            'throttle:api',
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],
    ];

    protected $routeMiddleware = [
        'check.age' => \App\Http\Middleware\CheckAge::class, // Register CheckAge middleware
    ];

```

The LogRequests middleware is globally registered, meaning it runs on every request. Meanwhile, the CheckAge middleware is part of the web middleware group, ensuring age checks are only enforced on routes within that group.

Middleware | LogRequests.php

web.php

routes > web.php

```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Middleware\LogRequests;
use App\Http\Middleware\CheckAge;
use App\Http\Controllers\HomePageController;
use App\Http\Controllers\AboutUsController;
use App\Http\Controllers\MenuController;
use App\Http\Controllers\ContactUsController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\RestrictedAreaController;

Route::middleware(['web', LogRequests::class])->group(function () {

    // Login route at the root URL "/"
    Route::get('/', [LoginController::class, 'goTologin'])->name('goTologin');

    // Form submission route for homepage with username validation
    Route::post('/homepage', [LoginController::class, 'submit']);

    // Homepage, About Us, Menu, Contact Us routes with dynamic username
    Route::get('/homepage/{username}', [HomePageController::class, 'index'])->name('homepage');
    Route::get('/aboutus/{username}', [AboutUsController::class, 'index'])->name('aboutus');
    Route::get('/menu/{username}', [MenuController::class, 'index'])->name('menu');
    Route::get('/contactus/{username}', [ContactUsController::class, 'index'])->name('contactus');

    // Route for restricted area with dynamic username
    Route::get('/restrictedArea/{username}', [RestrictedAreaController::class, 'index'])->name('restrictedArea');

    // Middleware for age verification, restricted access, or homepage access post-login
    Route::post('/homepage', [HomePageController::class, 'index'])->middleware(CheckAge::class);

    // Access Denied route
    Route::get('/accessDenied', [HomePageController::class, 'accessDenied'])->name('accessDenied');
});
```

All routes within the Route group are secured by both the web and LogRequests middleware, ensuring consistent logging and session management for web-based routes.

Route Definitions:

- **The root URL (/) triggers the goTologin() method of LoginController for user login.**
- **POST routes (e.g., /homepage) handle form submissions and redirect users based on their input.**
- **GET routes (e.g., /homepage/{username}, /aboutus/{username}, /menu/{username}) load various pages like the homepage, about us, and menu, using the username as a dynamic parameter.**
- **The /restrictedArea/{username} route opens a restricted page, while /accessDenied displays an access denied message for users facing restrictions.**

PART 3: CONTROLLERS WITH PARAMETERS

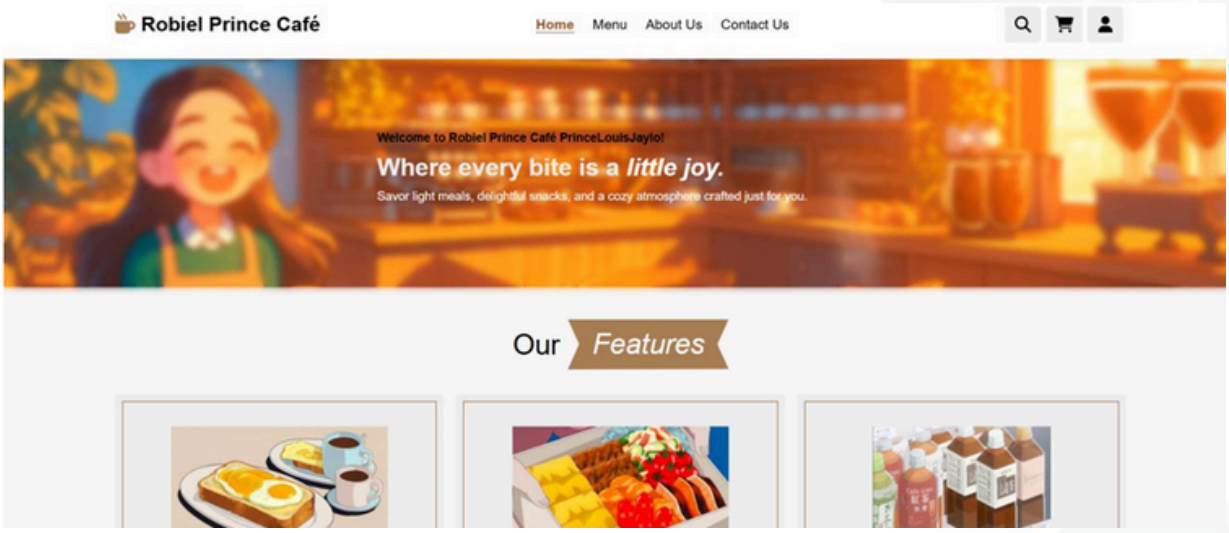
LOGIN CONTROLER

```php
public function goTologin($userId = null)
{
    $username = 'Guest';

    // If a user ID is provided, fetch or customize data for that user
    if ($userId) {
        // Placeholder logic; in a real app, fetch user info from a database.
        $username = "User{$userId}";
    }

    return view('goTologin', ['username' => $username])->with('hideNavAndFooter', true);
}
```

The LoginController handles the $userId parameter as follows:

- **If $userId is given, it generates a dynamic username (e.g., User{$userId}) to simulate a user-specific response.**
- **If not, it defaults to using "Guest" as the username.**

Web.php with the homepage route controller:




```php
// Homepage, About Us, Menu, Contact Us routes with dynamic username
Route::get('/homepage/{username}', [HomePageController::class, 'index'])->name('homepage');
```

This route maps to the index method of HomePageController, using the {username} parameter to provide dynamic access. This allows users to visit URLs like /homepage/PrinceLouisJaylo or /homepage/Guest to load personalized pages.

CONTROLLER LOGIC

1. **HomePageController: Displays the homepage and restricts access if users don't meet the age requirements.**
2. **MenuController: Loads the menu page for a specific username.**

PART 3: CONTROLLERS WITH PARAMETERS

3. **AboutUsController: Displays the About Us page for a specified username.**
4. **ContactUsController: Loads the Contact Us page for a given username.**
5. **LoginController: Manages user login, supporting login with a username or as a guest.**
6. **RestrictedAreaController: Provides access to a restricted area for authorized users.**

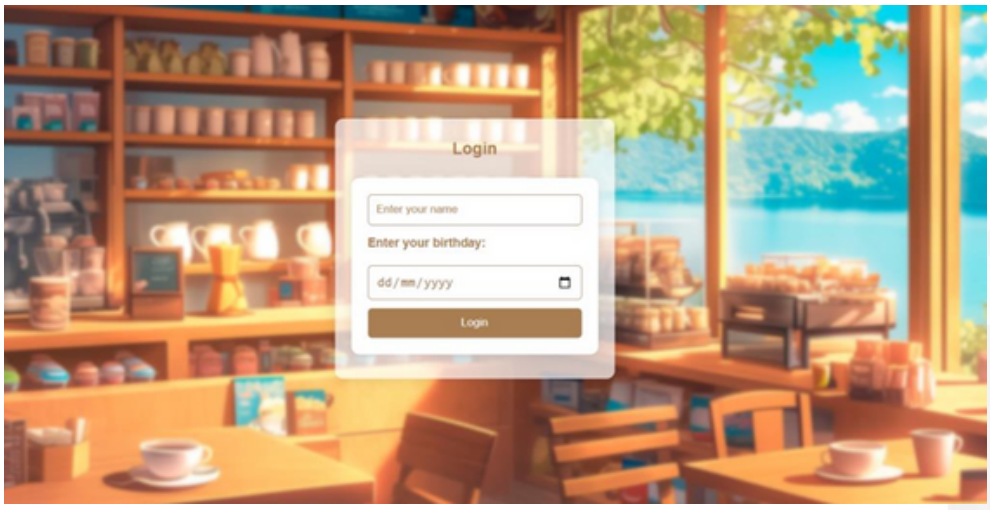
Parameter Handling:

- **Username: Personalizes the app by displaying greetings (e.g., "Welcome, John") on each page.**
- **User ID: In the login process, it generates a default username (e.g., "User123") if no username is provided.**
- **Age Check: The CheckAge middleware validates the user's age before granting access to specific pages.**

ROUTE ASSIGNMENT:

- **Basic Routes: Each page (e.g., /homepage, /menu) is linked to a controller method that loads the correct view.**
- **Middleware: Middleware like CheckAge is applied to control access based on user details (e.g., age).**
- **Dynamic Routes: Routes such as /homepage/{username} allow dynamic loading of user-specific pages without creating separate routes for each user.**

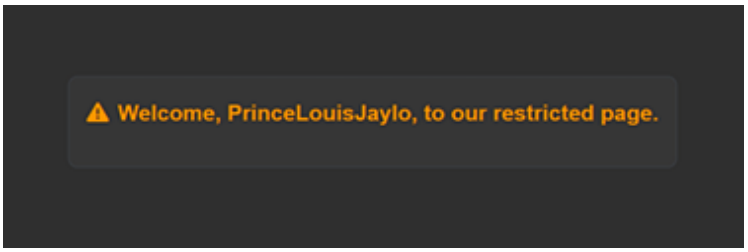
Rendered Web Pages: goTologin.blade.php



The user can input their name and birthdate. If the birthdate is 2003 or earlier, they'll be shown a restricted page but will still have access to other sections like the menu. If the birthdate is 2007 or later, they'll be redirected to an Access Denied page, accompanied by a message that says, "You must be 18 or older," along with a "Back to login" option. For birthdates between 2004 and 2006, the user will be taken to the homepage to explore without restrictions.

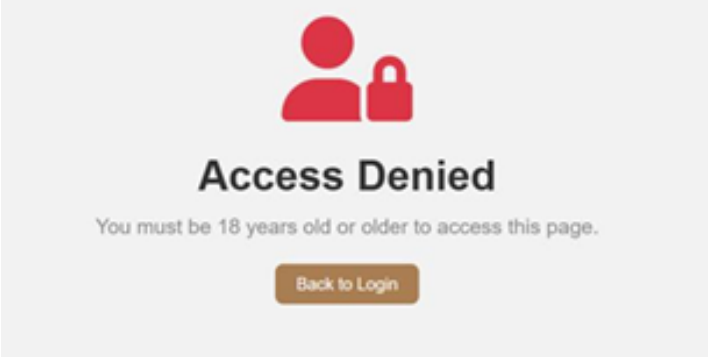
PART 3: CONTROLLERS WITH PARAMETERS

restrictedArea.blade.php



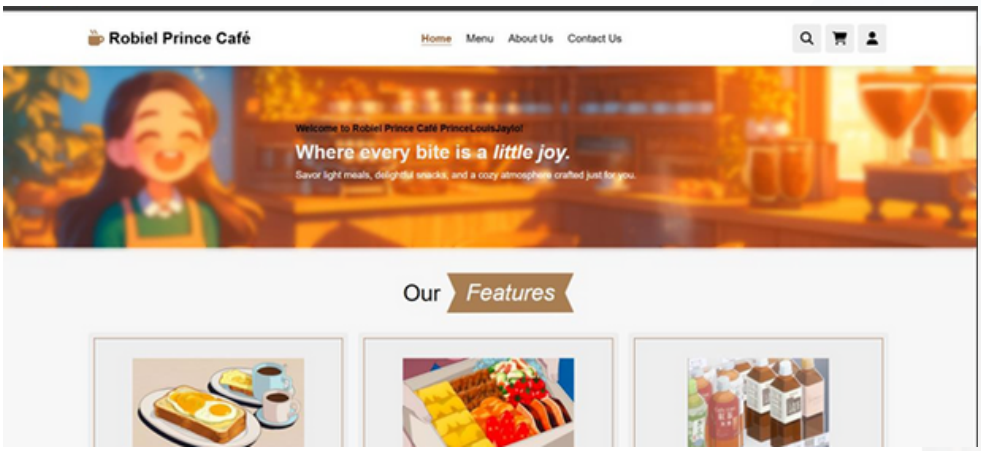
In the restrictedArea.blade.php file, when I entered 2003 as the birthdate, I was redirected to the "Welcome to our restricted page" message, but I was still able to browse other pages, such as the menu.

accessDenied.blade.php



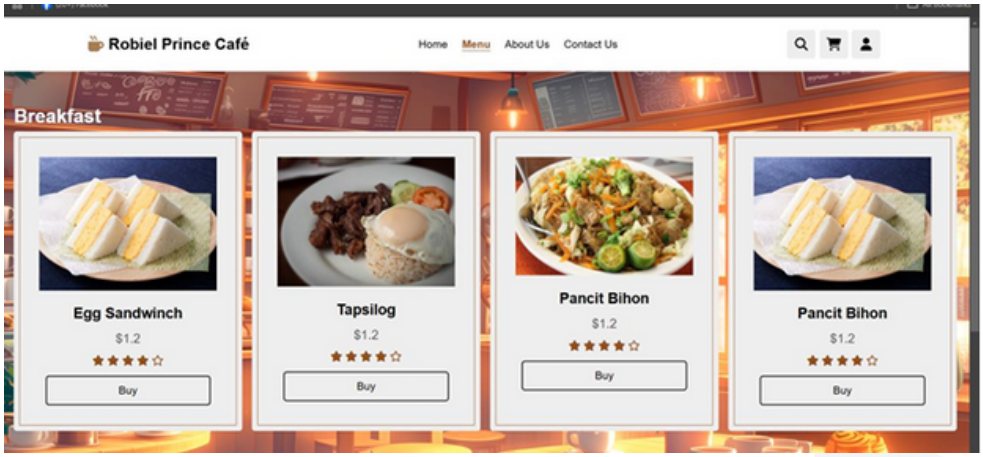
In the accessDenied.blade.php file, with a 2007 birthdate input, I was redirected to the Access Denied page, which displayed a message saying, "You must be 18 or older to access this page," along with a "Back to login" button to return to the login page. Note: The "Back to login" button is still being developed.

homepage.blade.php

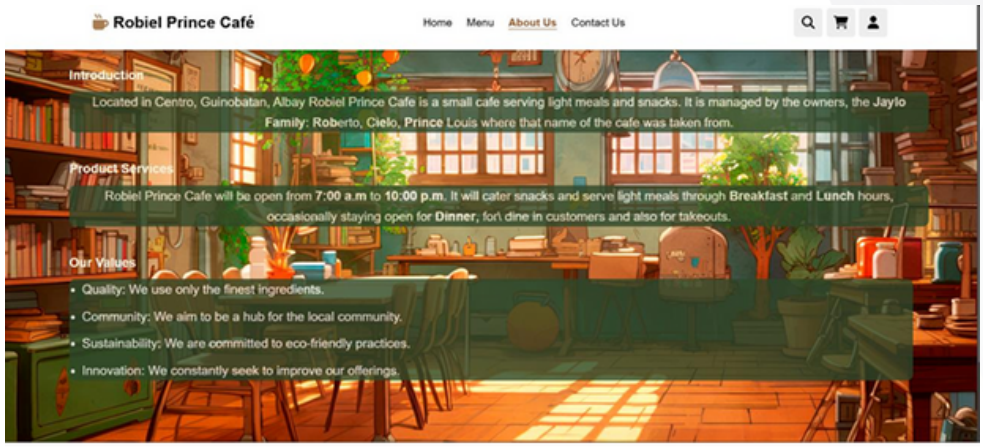


PART 3: CONTROLLERS WITH PARAMETERS

menu.blade.php



aboutus.blade.php



contactus.blade.php