Prince Louis M. Jaylo BSIT 3-C IT Elec 1 Web Development

Laravel Laboratory Activity 5: Controllers Documentation

Part 1: Controller Creation and Registration

HomePageController:

```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 handles loading the homepage. It accepts a dynamic parameter $username, which is passed to the view as an array. If no username is provided, it defaults to 'Guest'.
- accessDenied() method: This method returns a view named accessDenied, which is displayed when the user does not meet certain criteria (such as age restrictions or missing input data).

Menu Controller:

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($username) method: The controller's index($username) accepts a dynamic parameter $username, which is passed to the respective view (menu, about us, or contact us). This allows the views to display personalized information based on the username passed in the URL.

AboutUsController:

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 loads the "About Us" page using index($username) that loads the aboutus view with the provided username.

ContactUsController:

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 defines the class, specifically its index method, which takes a $username parameter and returns the contactus view. The contactus view is then provided with the $username data, allowing personalized content (such as a welcome message or user-specific details) to be displayed on the contact us page.

LoginController:

app > Http > 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 handles the login page, allowing a user to be logged in either as a guest or a specific user (based on the $userId). If $userId is provided, a personalized username is created. Otherwise, it defaults to 'Guest'.
- The submit(Request $request) method handles the form submission from the login page. It validates that the username contains only alphabetic characters. After successful validation, it redirects the user to the homepage with the given username.

RestrictedAreaController:

app > Http > Controllers > 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 is responsible for loading the restricted area page. It receives a dynamic $username parameter to display personalized content in the view restrictedArea.

Part 2: Assigning Controllers to Routes

Middleware: CheckAge:

```
app > Http > Middleware > CheckAge.php
10  {
18      public function handle(Request $request, Closure $next)
19      {
20          // Get the birthdate from the form input
21          $birthDate = $request->input('birthday');
22          $username = $request->input('username');
23
24          // Check if birthdate is provided
25          if (!$birthDate) {
26              // Redirect to Access Denied if the birthdate is missing
27              return redirect('/accessDenied')->with('error', 'Birthdate is required.');
28          }
29
30          // Parse the birthdate and validate the format
31          try {
32              $age = Carbon::parse($birthDate)->age;
33          } catch (\Exception $e) {
34              // Invalid birthdate format
35              return redirect('/accessDenied')->with('error', 'Invalid birthdate format.');
36          }
37
38          \Log::info("User Age: $age, Username: {$request->input('username')}");
39
40
41          // Check if the user is under 18
42          if ($age < 18) {
43              // Redirect to Access Denied page if under 18
44              return redirect('/accessDenied')->with('error', 'You must be 18 or older to access this page.');
45          } elseif ($age >= 18 && $age <= 20) {
46              // Redirect to homepage if age is between 18 and 20
47              return redirect("/homepage/" . $username);
48          } elseif ($age >= 21) {
49
50              \Log::info('Redirecting to restricted area for user: ' . $username);
51              // Redirect to restricted area if 21 or older
52              return redirect("/restrictedArea/" . $username);
```

The handles(Request $request, Closure $next) of CheckAge middleware checks if the user is old enough to access specific pages. It reads the user's birthdate from the request, calculates their age, and redirects them accordingly:

- If the user is under 18, they are redirected to the access denied page.
- If the user is between 18 and 20, they are redirected to the homepage.
- If the user is 21 or older, they are redirected to a restricted area.
- If none of the above conditions are met, the middleware allows the request to continue.

Middleware LogRequest:

```
app > Http > Middleware > LogRequests.php
1   <?php
2
3   namespace App\Http\Middleware;
4
5   use Closure;
6   use Illuminate\Http\Request;
7   use Illuminate\Support\Facades\Log;
8   use Symfony\Component\HttpFoundation\Response;
9
10  class LogRequests
11  {
12      /**
13       * Handle an incoming request.
14       *
15       * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
16       */
17      public function handle(Request $request, Closure $next): Response
18      {
19          Log::info('LogRequests middleware executed'); // Log a simple message
20          $logData = '[' . now() . '] ' . $request->method() . ' ' . $request->fullUrl();
21          Log::channel('custom')->info($logData);
22
23          return $next($request);
24      }
25  }
```

- The handle(Request $request, Closure $next) of LogRequest middleware logs every incoming HTTP request with its method (GET, POST, etc.) and the full URL. This is useful for debugging or tracking all user activity within the application.

Middleware Kernel.php (For Registration):

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 middleware LogRequests is globally registered, meaning it will be executed for every request.
- The CheckAge middleware is registered in the web middleware group, so it will only run on routes that are part of the web group, ensuring age validation is only applied to certain routes.

Routes Definition with the controllers

```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 the routes inside the Route group are protected by the web middleware and LogRequests middleware. This ensures consistent logging of all user interactions and session management for web-based routes.
- **Route Definitions:**
  - The root URL (/) maps to the goTologin() method of LoginController, allowing users to log in.
  - POST routes like /homepage submit the username, redirecting the user accordingly.
  - GET routes such as /homepage/{username}, /aboutus/{username}, /menu/{username}, etc., load different pages like the homepage, about us, and menu, passing the username as a dynamic parameter.
  - The /restrictedArea/{username} route loads the restricted area, and the /accessDenied route displays the access denied page if the user is redirected due to restrictions.

Part 3: Controllers with Parameters

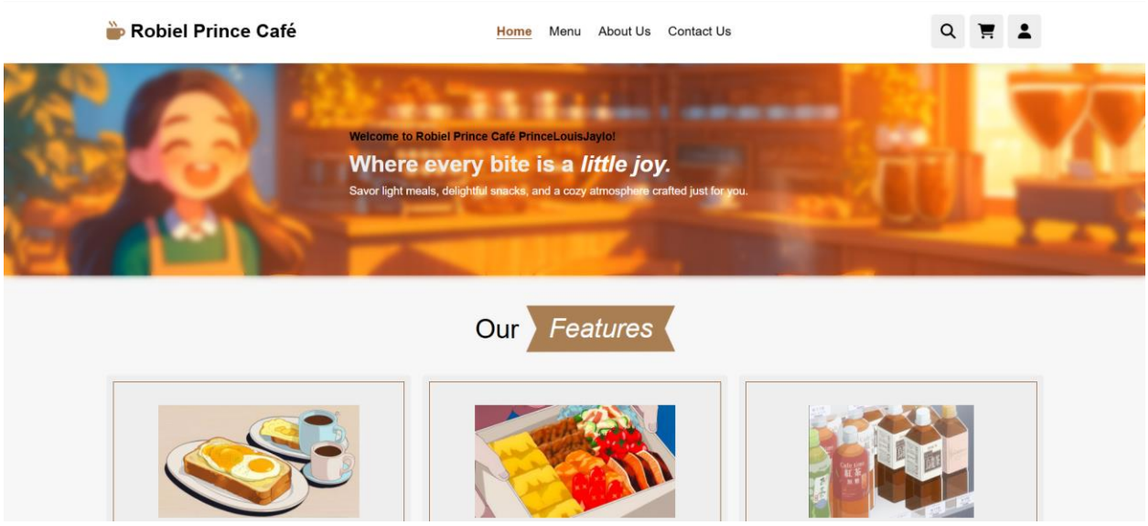
LoginController:

```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);
}
```

- This part of the code of the LoginController demonstrates how the LoginController handles the $userId parameter.
- If $userId is provided, a dynamic username (User{$userId}) is assigned, simulating a user-specific response.
- Otherwise, a default value of "Guest" is used.

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 the HomePageController, allowing dynamic access based on the {username} parameter.
- This enables users to access URLs like /homepage/PrinceLouisJaylo or /homepage/Guest, loading personalized pages.

Controller Logic

1. **HomePageController**: Loads the homepage and handles "Access Denied" if users don't meet age criteria.
2. **MenuController**: Loads the menu page with the given username.
3. **AboutUsController**: Loads the About Us page with the given username.
4. **ContactUsController**: Loads the Contact Us page with the given username.
5. **LoginController**: Handles login, allowing users to log in with a name or as a guest.
6. **RestrictedAreaController**: Loads a restricted area page for authorized users.

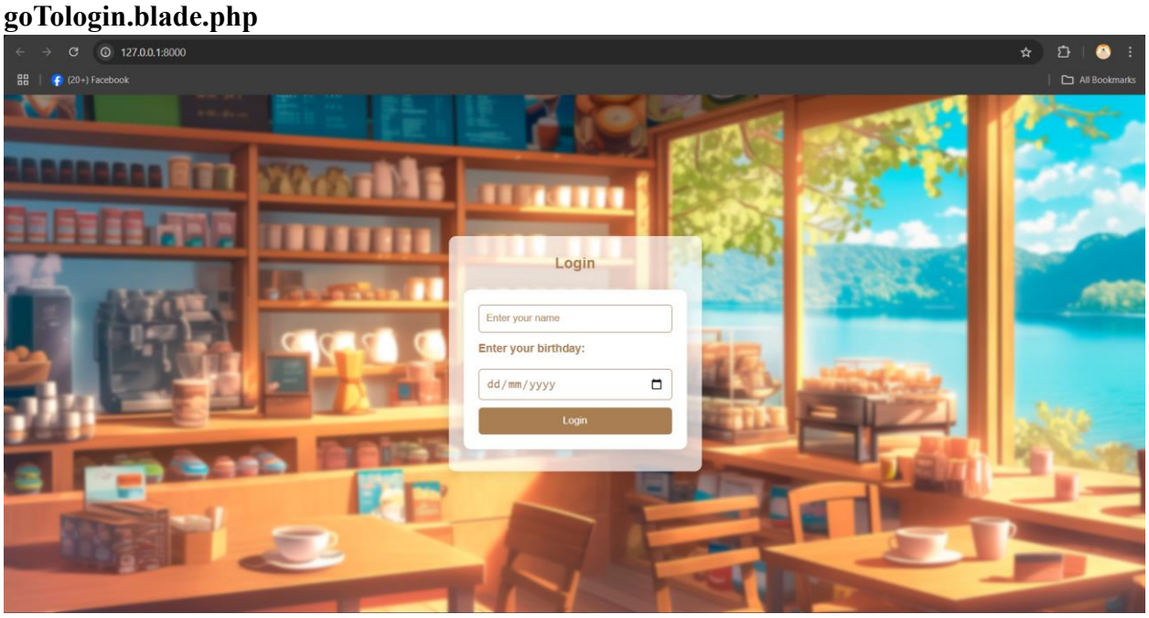
Parameter Handling

1. **Username**: Allows the app to display personalized greetings (e.g., "Welcome, John") on each page.
2. **User ID**: In login, it can create a default username (e.g., "User123") if no name is provided.
3. **Age Check**: The CheckAge middleware verifies the user's age and either allows access or redirects based on age.

Route Assignments

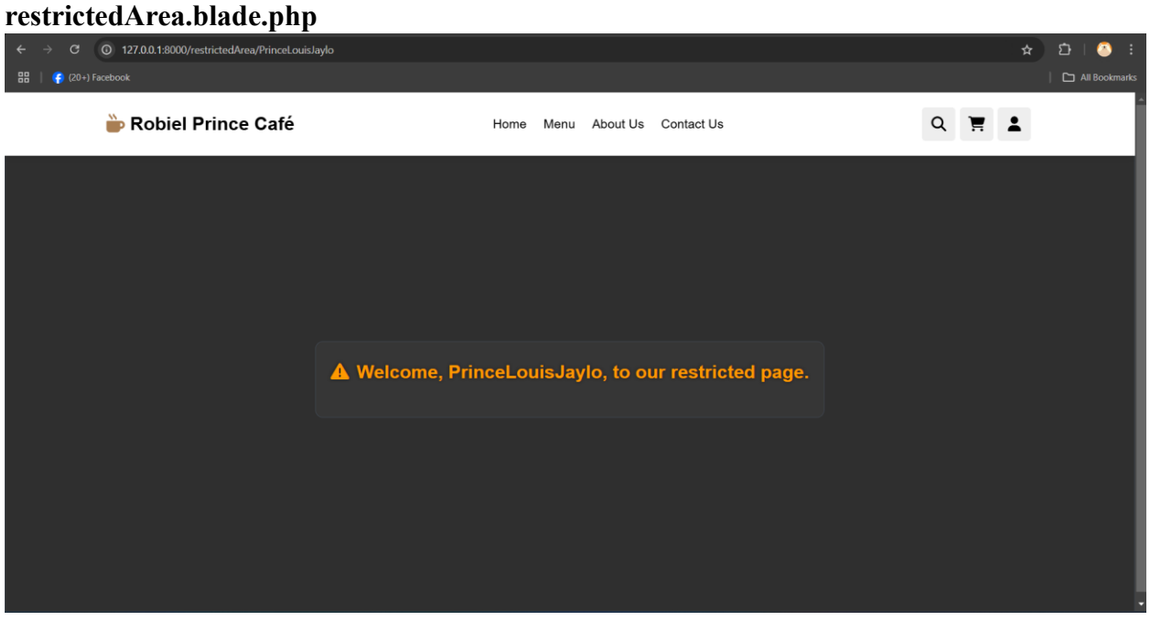
1. **Basic Routes**: Each page (e.g., /homepage, /menu) is linked to a specific controller method for loading the correct view.
2. **Middleware**: Routes have middleware (like CheckAge) that controls access based on user information (e.g., age).
3. **Dynamic Routes**: Routes like /homepage/{username} allow user-specific pages to be loaded without needing separate routes for each user.

Rendered Web Pages:
goTologin.blade.php



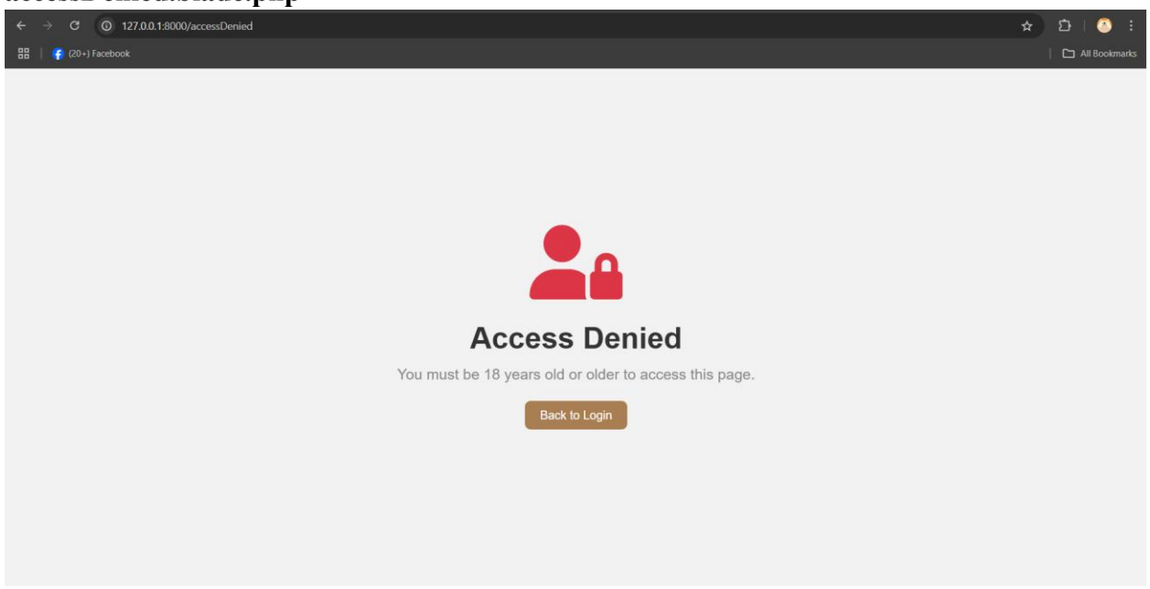
The user may input their name and birthdate. If the user inputs 2003 below the page will display Welcome to our restricted page but the user can still browse through web pages like the menu. If the user inputs 2007 above the user will be redirected to the Access Denied Page and will give a message saying "You must be 18 years old or older to access this page with a Back to login button so the user can go back to the login page. Lastly if the user inputs 2004 – 2006 the user will be redirected to the directly to the homepage where the user can browse freely.

restrictedArea.blade.php



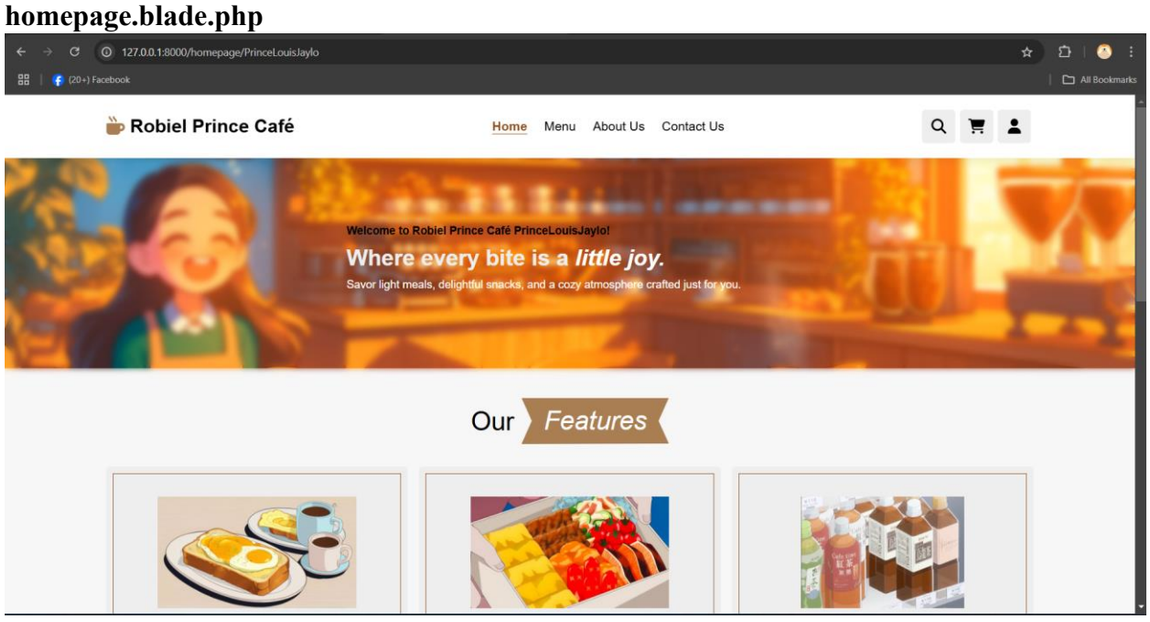
Here in the restrictedArea.blade.php, my input was 2003 so I was redirected to Welcome to our restricted page, but I can still browse through web pages like the menu.

accessDenied.blade.php

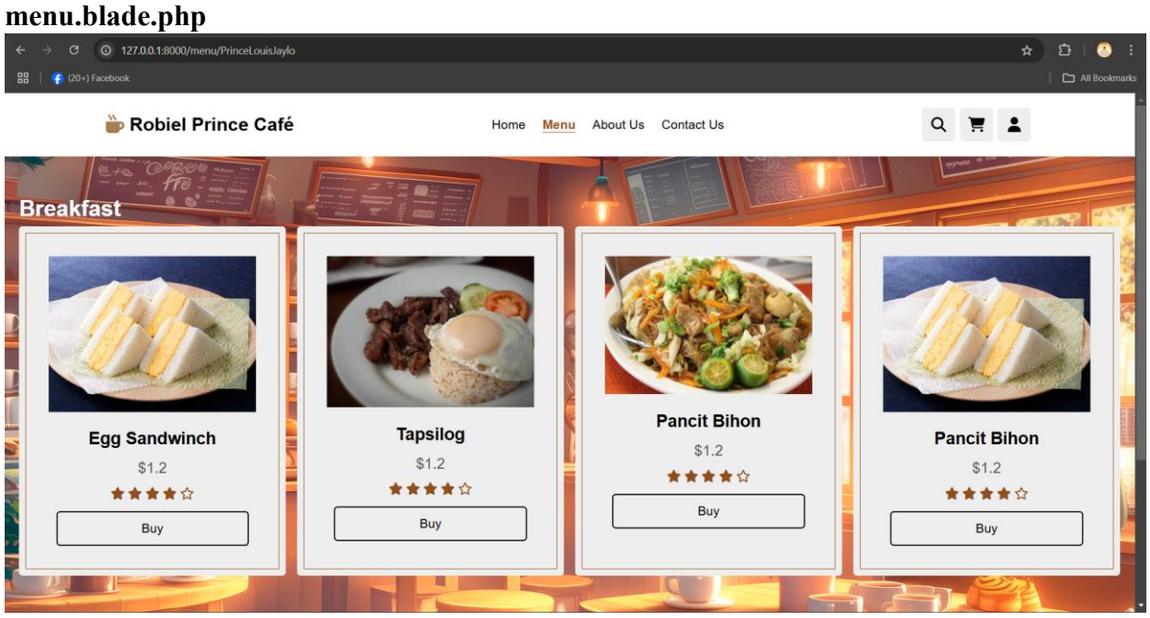


Here on the accessDenied.blade.php, My input was 2007 therefor I am redirected to the Access Denied Page and I gave me a message saying “You must be 18 years old or older to access this page with a Back to login button so I can return to the login page.
Note: The back to log in button is still in progress.

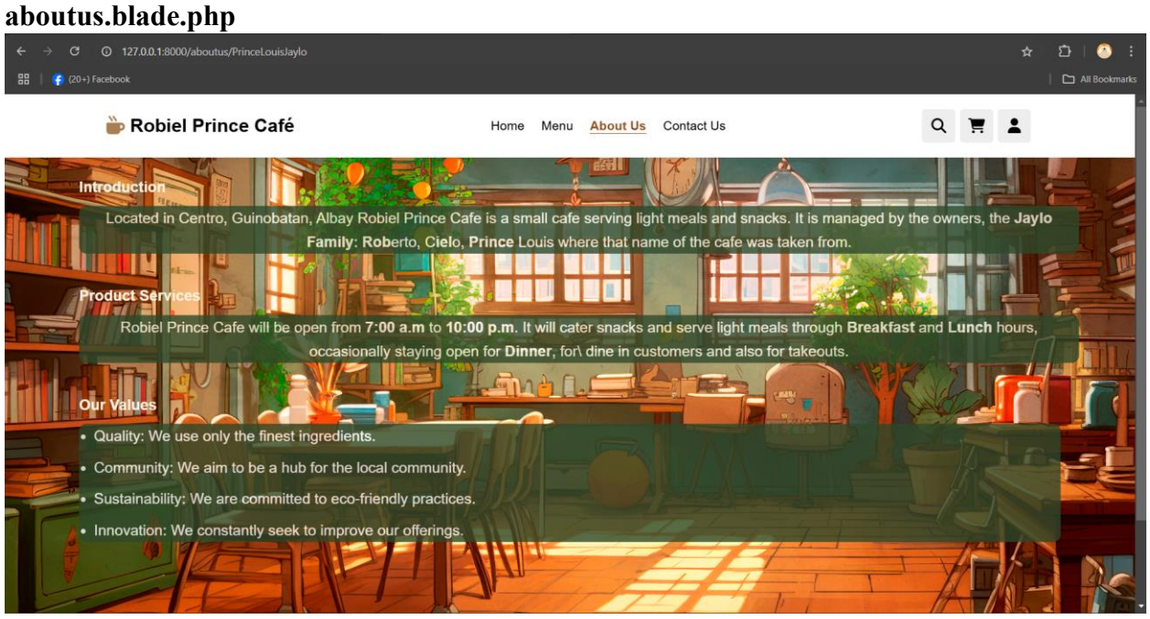
homepage.blade.php



menu.blade.php



aboutus.blade.php



contactus.blade.php