

CheckAge Middleware

- This checks the user's age on the website. After inputting their age, if they are below 18, they are redirected to the access-denied page. If they are 21 or older, it labels them as 'Verified' if not they are 'Unverified'.

```
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Support\Facades\Session;
7
8 3 references | 0 implementations
9 class CheckAge
10 {
11     0 references | 0 overrides
12     public function handle($request, Closure $next): mixed
13     {
14         // Retrieve the age from the session
15         $age = Session::get(key: 'age');
16         $username = Session::get(key: 'username', default: 'Guest'); // Retrieve username from the session, default to 'Guest'
17
18         // Check the age and set the verification status
19         if ($age < 18) {
20             // Log access denied
21             $logData = sprintf(
22                 format: "[%s] Access Denied - Username: %s, Age: %d\n",
23                 values: now()->toDateTimeString(),
24                 $username,
25                 $age
26             );
27             file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND);
28
29             return redirect(to: '/access-denied');
30         } elseif ($age >= 21) {
31             Session::put(key: 'verificationStatus', value: 'Verified');
32
33             // Log access granted for 21 and above
34             $logData = sprintf(
35                 format: "[%s] Access Granted - Username: %s, Age: %d, Status: Verified\n",
36                 values: now()->toDateTimeString(),
37                 $username,
38                 $age
39             );
40             file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND);
41         }
42     }
43 }
```

LogRequests Middleware

- This logs the information details for each request. These information include, username, IP address, user's age, and the full URL. The access verified or denied is also logged here. The information is stored in a log.txt file which presents an error if the logs are not written.

```

1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Symfony\Component\HttpFoundation\Response;
8
9  class LogRequests
10 {
11     /**
12      * Handle an incoming request.
13      *
14      * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
15      */
16     public function handle(Request $request, Closure $next): Response
17     {
18         // Prepare log data with timestamp, HTTP method, and full URL
19         $logData = sprintf(
20             format: "[%s] %s %s\n",
21             values: now()->toDateTimeString(), // Time
22             $request->method(), // HTTP method (GET, POST, etc.)
23             $request->fullUrl() // Full URL of the request
24         );
25
26         // Attempt to log the request details in log.txt file
27         if (file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND) === false) {
28             // Log an error message if writing to the log file fails
29             \Log::error(message: 'Failed to write to log.txt');
30         }
31
32         // Proceed with the next middleware/handler in the stack
33         return $next($request);
34     }
35 }

```

Kernel.php

- This oversees managing the middleware in Laravel. Included here is the global middleware which runs at every request like maintenance or logging requests.

```

59     0 references
60     protected $routeMiddleware = [
61         'auth' => \App\Http\Middleware\Authenticate::class,
62         'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
63         'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
64         'can' => \Illuminate\Auth\Middleware\Authorize::class,
65         'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
66         'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
67         'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
68         'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
69         'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
70         'check.age' => \App\Http\Middleware\CheckAge::class,
71         'log.requests' => \App\Http\Middleware\LogRequests::class,
72     ];
73 }

```

```

1  <?php
2
3  namespace App\Http;
4
5  use Illuminate\Foundation\Http\Kernel as HttpKernel;
6
7  0 references | 0 implementations
8  class Kernel extends HttpKernel
9  {
10     /**
11      * The application's global HTTP middleware stack.
12      * These middleware are run during every request to your application.
13      *
14      * @var array
15      */
16     0 references
17     protected $middleware = [
18         // Handles maintenance mode
19         \App\Http\Middleware\CheckForMaintenanceMode::class,
20         // Validates the request's size limit
21         \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
22         // Converts input data to proper types
23         \App\Http\Middleware\TrimStrings::class,
24         \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
25         \App\Http\Middleware\TrustProxies::class,
26
27         \App\Http\Middleware\LogRequests::class,
28     ];
29
30     /**
31      * The application's route middleware groups.
32      *
33      * @var array
34      */

```

Web.php

- Import the CheckAge and LogRequests middleware to the routes.

```

1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use Illuminate\Support\Facades\Session;
5  use App\Http\Middleware\CheckAge; // Import CheckAge middleware

```

- The route logout, when clicked, will remove the username, age, and verification status from the session. This logs out the user and clearing their information.
- After removing the information, the user is redirected to the register page or (/). They will no longer have access to their past information.

```
// Route to handle user logout
Route::get(uri: '/logout', action: function (): Redirector|RedirectResponse {
    // Clear the username from session
    Session::forget(keys: 'username');

    // Redirect to the homepage or login page
    return redirect(to: '/');
});
```

- This access denied route logs the username when a denied access is made. It then logs the information to the log.txt file. Which then returns the view of access denied.

```
99 // Route to handle when access is denied
100 Route::get(uri: '/access-denied', action: function (): Factory|View {
101     // Get the username from the session
102     $username = Session::get(key: 'username', default: 'Guest');
103
104     // Log the access denied request with the username only once
105     $logData = sprintf(
106         format: "[%s] Access Denied - Username: %s\n",
107         values: now()->toDateTimeString(),
108         $username // Include the username in the log
109     );
110
111     // Attempt to log the access denied details in log.txt file
112     file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND);
113
114     return view(view: 'access-denied'); // Ensure this view exists
115 });
116
```

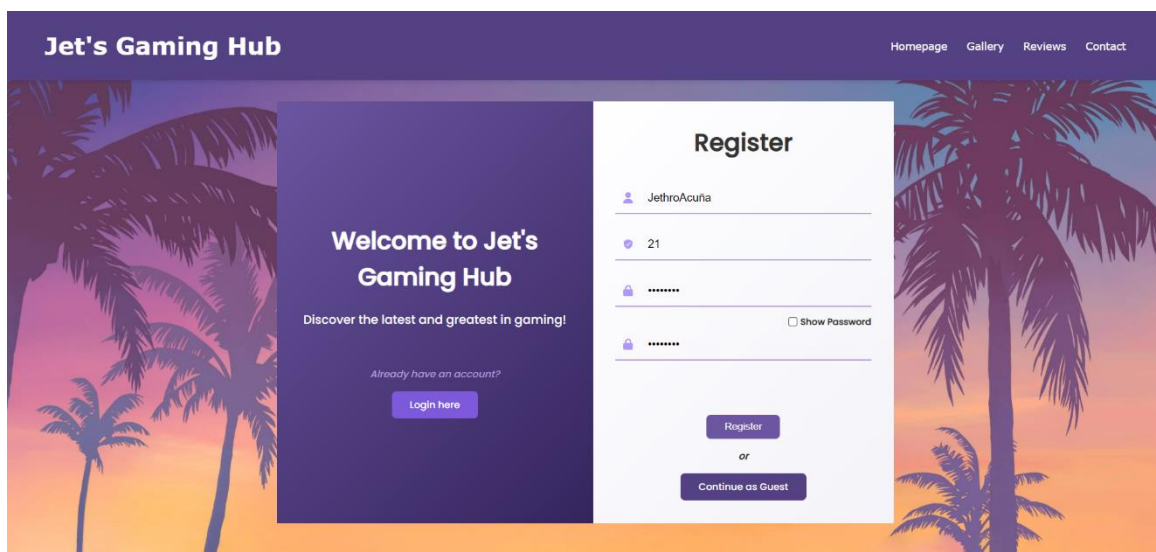
access-denied.blade.php

- A template for when the user age input is 18 below. This is the page they are directed to when accessing is denied.

```
1  @extends(view: 'Components.layout')
2
3  @php
4      $title = 'Access Denied';
5  @endphp
6
7  @push('styles')
8      <link href="{{ asset(path: 'css/access-denied.css') }}" rel="stylesheet">
9  @endpush
10
11 @section(section: 'body-class', content: 'font-sans antialiased dark:bg-black dark:text-white/50')
12
13 @section(section: 'main-class', content: 'main-content')
14
15 @section(section: 'content')
16     <div class="blur-background"></div>
17
18     <div class="center-container">
19         <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100" width="100" height="100" class="circle-x-icon">
20             <!-- Outer Circle -->
21             <circle cx="50" cy="50" r="48" class="outer-circle" />
22
23             <!-- X shape -->
24             <line x1="30" y1="30" x2="70" y2="70" class="cross-line" />
25             <line x1="70" y1="30" x2="30" y2="70" class="cross-line" />
26         </svg>
27
28         <h1>Access Denied</h1>
29         <p>You must be at least 18 years old to access this content.</p>
30     </div>
31 @endsection
32
33
```

VIEWS OF OUR WEBSITE

- This is where the user will input their information.



- This is where the user will be redirected to if they are below the 18 age limit.

