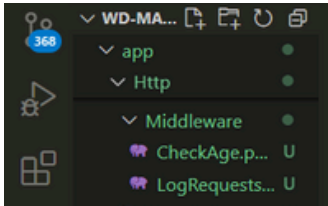


DOCUMENTATION

Part 1: Creating and Registering New Middleware



- Using the command line, create new middleware named CheckAge and LogRequests.

CheckAge.php

```
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\View;
8
9 class CheckAge
10 {
11     public function handle($request, Closure $next, $minAge = 18)
12     {
13         $age = $request->input('age', 0);
14         if ($age >= 18 && $age <= 20) {
15             return response()->view('home');
16         }
17         if ($age >= 21) {
18             return response()->view('adults');
19         }
20         return response()->view('denied');
21     }
22 }
```

LogRequest.php

```
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Storage;
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         $logData = 'URL: ' . $request->url() . ' | Method: ' . $request->method() . ' | Timestamp: ' . now()->format('Y-m-d H:i:s') . PHP_EOL;
20         Storage::append('LogReqLab3/log.txt', $logData);
21         return $next($request);
22     }
23 }
```

- The CheckAge middleware should check if a user's age is greater than or equal to 18. If the age does not meet the condition, redirect the user to an "Access Denied" page.
- LogRequests should log the details of all HTTP requests to a file called log.txt, include the URL method, and timestamp

CheckAge.php

```
1 class CheckAge
2 {
3     public function handle($request, Closure $next, $minAge = 18)
4     {
5         $age = $request->input('age', 0);
6         if ($age >= 18 && $age <= 20) {
7             return response()->view('home');
8         }
9         if ($age >= 21) {
10             return response()->view('adults');
11         }
12         return response()->view('denied');
13     }
14 }
```

LogRequest.php

```
17 public function handle(Request $request, Closure $next): Response
18 {
19     $logData = 'URL: ' . $request->url() . ' | Method: ' . $request->method() . ' | Timestamp: ' . now()->format('Y-m-d H:i:s') . PHP_EOL;
20     Storage::append('LogReqLab3/log.txt', $logData);
21     return $next($request);
22 }
23 }
```

```
storage > app > LogReqLab3 > log.txt
1 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-09-27 21:45:44
2
3 URL: http://127.0.0.1:8000/homepage/Guest | Method: GET | Timestamp: 2024-09-27 21:45:51
4
5 URL: http://127.0.0.1:8000/about/Guest | Method: GET | Timestamp: 2024-09-27 21:45:56
6
7 URL: http://127.0.0.1:8000/content/Guest | Method: GET | Timestamp: 2024-09-27 21:45:58
8
9 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-10-02 07:39:21
10
11 URL: http://127.0.0.1:8000/home | Method: GET | Timestamp: 2024-10-02 07:39:30
12
13 URL: http://127.0.0.1:8000/homepage/Guest | Method: GET | Timestamp: 2024-10-02 07:39:37
14
15 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-10-02 07:39:55
16
17 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-10-02 07:40:08
18
19 URL: http://127.0.0.1:8000/homepage/Clarence | Method: GET | Timestamp: 2024-10-02 07:40:16
20
21 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-10-02 07:40:50
22
23 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-10-02 07:40:58
24
25 URL: http://127.0.0.1:8000/home | Method: GET | Timestamp: 2024-10-02 07:41:06
26
27 URL: http://127.0.0.1:8000/homepage/claaa | Method: GET | Timestamp: 2024-10-02 07:41:14
28
29 URL: http://127.0.0.1:8000/about/claaa | Method: GET | Timestamp: 2024-10-02 07:41:28
30
31 URL: http://127.0.0.1:8000/content/claaa | Method: GET | Timestamp: 2024-10-02 07:41:34
32
33 URL: http://127.0.0.1:8000/contact/claaa | Method: GET | Timestamp: 2024-10-02 07:41:39
34
35 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-10-02 07:44:13
36
37 URL: http://127.0.0.1:8000 | Method: POST | Timestamp: 2024-10-02 07:44:24
```

- Register the middleware in the app/Http/Kernel.php file under the appropriate section.

```
protected $middleware = [
    \App\Http\Middleware\TrustProxies::class,
    \Illuminate\Http\Middleware\HandleCors::class,
    \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
    \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
    \App\Http\Middleware\TrimStrings::class,
    \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
    //global middleware part added
    \App\Http\Middleware\CheckAge::class,
    \App\Http\Middleware\LogRequests::class,
];
```

- Register middleware both as global middleware and route-specific middleware.

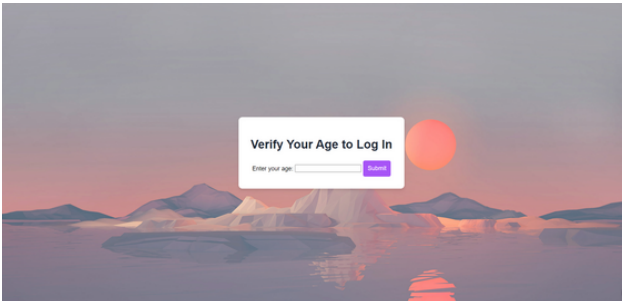
```
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    //routeMiddleware part added
    'check.age' => \App\Http\Middleware\CheckAge::class,
    'log.requests' => \App\Http\Middleware\LogRequests::class,
];
```

Part 2: Assign Middleware to Routes:

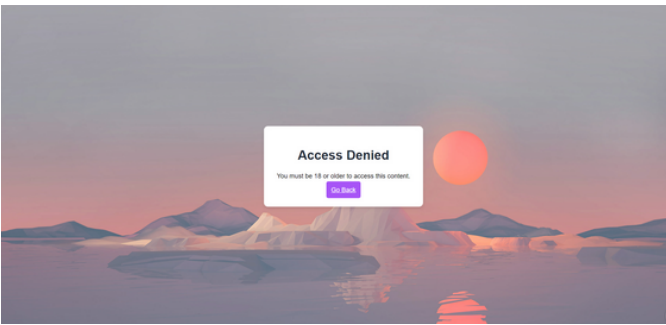
- Create a route group that assigns the CheckAge middleware to a specific route.

```
// CheckAge middleware to restricted contents
Route::get('/adults', function () {
    return view('adults');
})->name('adults')->middleware(CheckAge::class.':21');
```

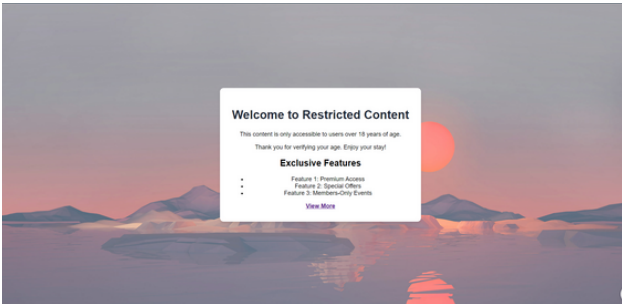
The root will ask the user's age.



If the user is below 18, it will be directed to access denied



But if the user is 21 or older, it will redirect to this restricted content page.



Part 3: Create Middleware with Parameters:

- Modify the CheckAge middleware to accept a parameter (e.g., the minimum age requirement).

Namespace Declaration: It organizes the CheckAge middleware class under App\Http\Middleware, helping Laravel know where to find the class.

Imports:

Closure: A function used to move to the next middleware or request handler.

Request: Handles incoming data from the HTTP request (e.g., form inputs).

View: Allows the middleware to display different web pages (views) based on the user's age.

Middleware Class:

CheckAge contains the age-checking logic.

handle Method:

- **It takes three arguments:**

\$request: Contains the data from the user, including their age.

\$next: Passes the request to the next middleware or controller.

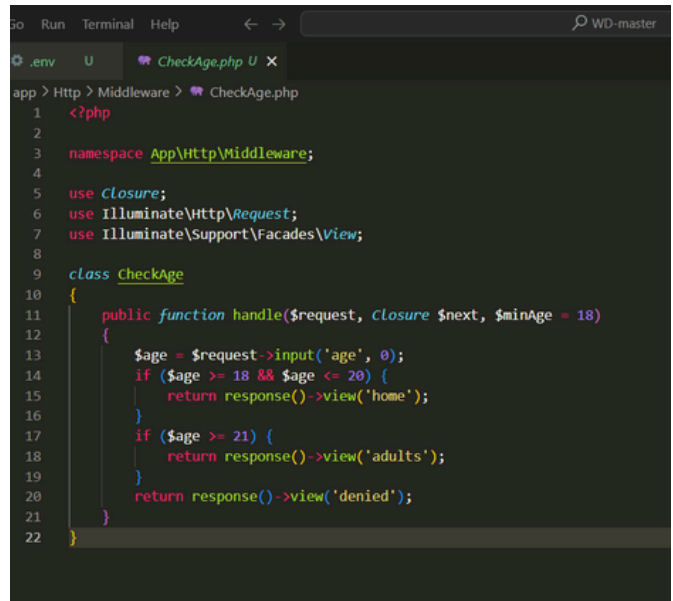
\$minAge: Sets the minimum age, with 18 as the default.

- **It checks the user's age:**

If the age is 18 to 20, it shows the home page.

If the age is 21 or older, it shows the adults page.

If the age is less than 18, it shows the denied page.



```
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\View;
8
9 class CheckAge
10 {
11     public function handle($request, Closure $next, $minAge = 18)
12     {
13         $age = $request->input('age', 0);
14         if ($age >= 18 && $age <= 20) {
15             return response()->view('home');
16         }
17         if ($age >= 21) {
18             return response()->view('adults');
19         }
20         return response()->view('denied');
21     }
22 }
```

Basically, this middleware controls which page the user sees based on their age.

- Create a new route that assigns the middleware with a parameter to enforce a different age restriction (e.g., 21 years old).

```
// CheckAge middleware to restricted contents
Route::get('/adults', function () {
    return view('adults');
})->name('adults')->middleware(CheckAge::class.':21');
```

This route defines a URL /adults that returns the adults view. It is protected by the CheckAge middleware, which checks if the user's age is at least 21. If the user's age meets this requirement, they can access the page; otherwise, they will be denied. The 21 is passed as the minimum age parameter to the CheckAge middleware.