

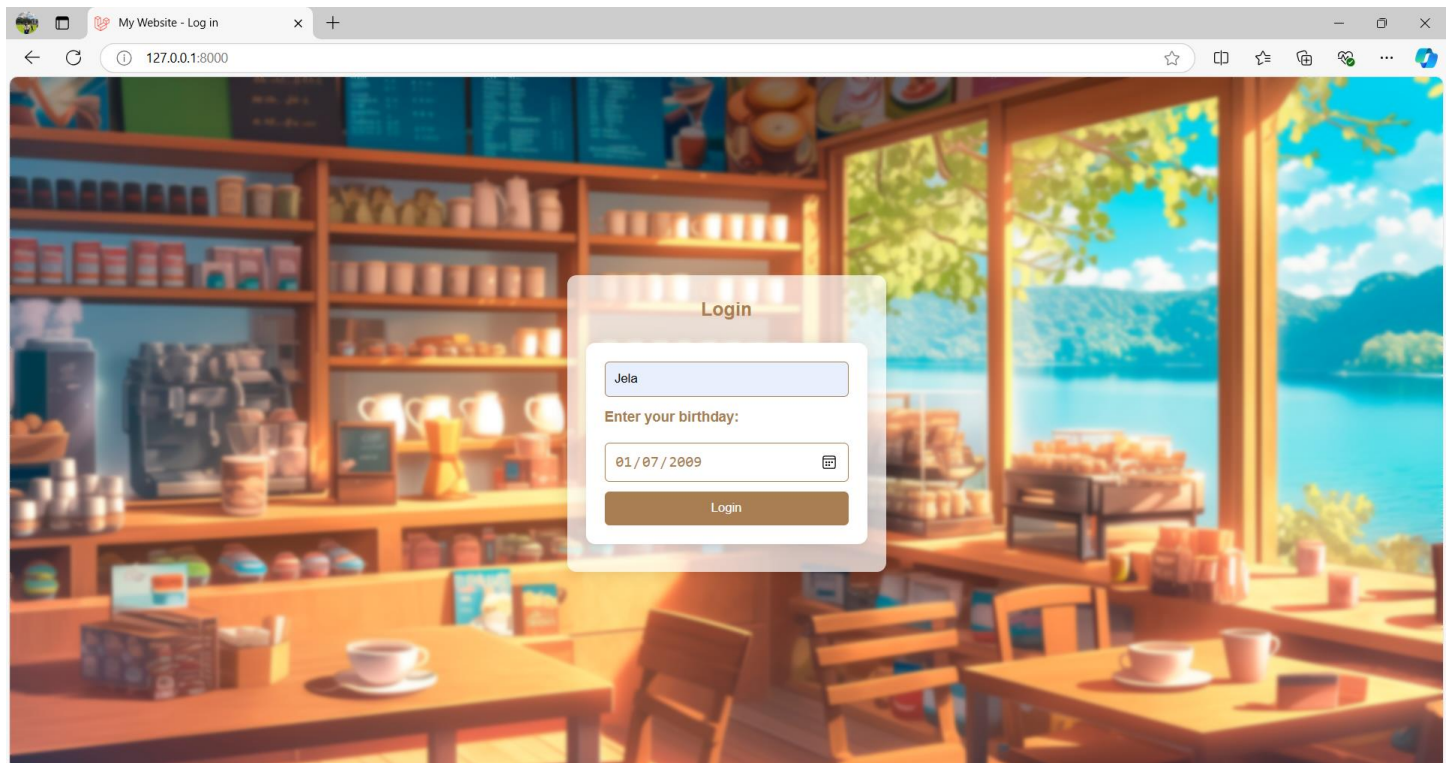
- The code below is for the **CheckAge** middleware, which checks if the user's age meets the required condition. Instead of directly inputting the age, we decided to use their birthdate. If the user meets the condition, they will be redirected to the appropriate page. Users under 18 will be redirected to the “**Access Denied**” page, while users 18 and above will be redirected to the homepage.

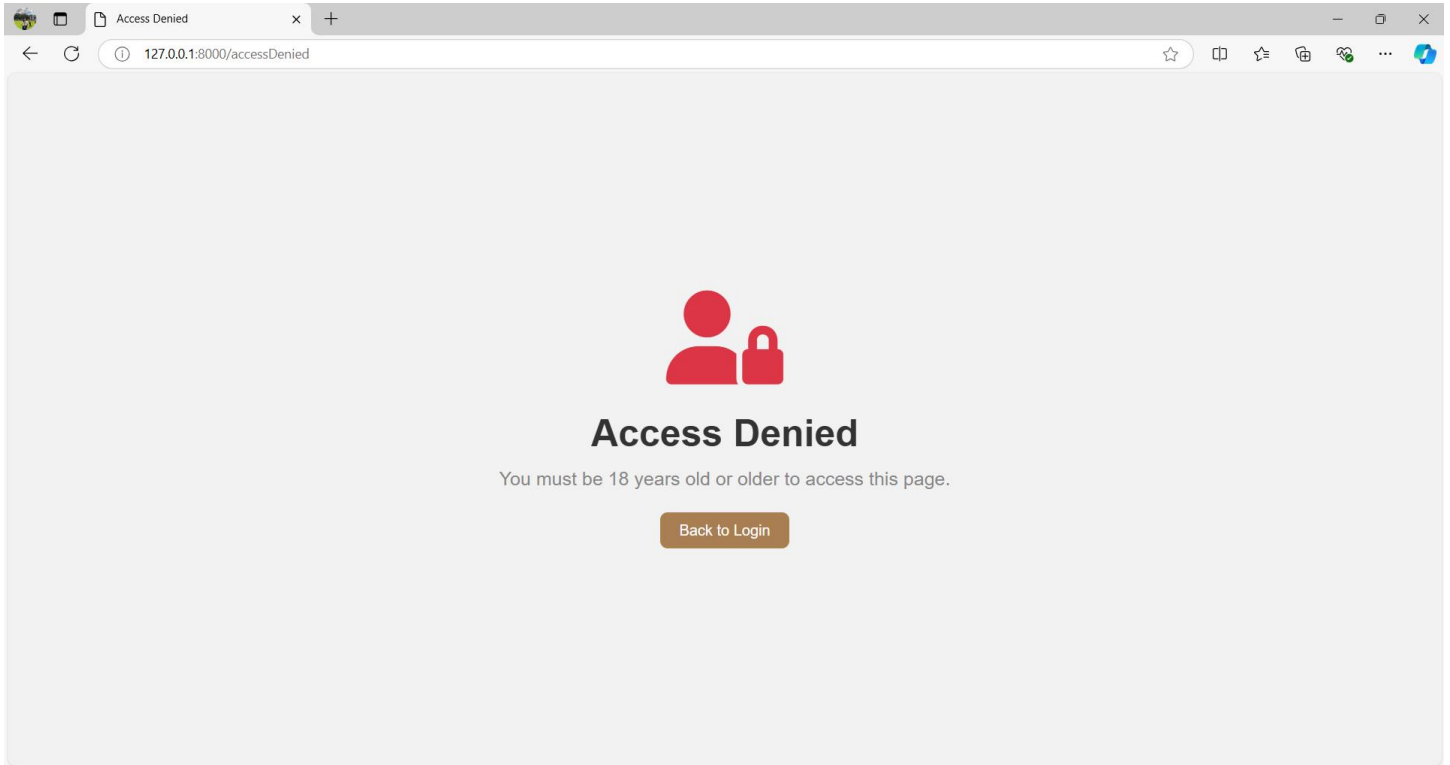
```
File Edit Selection View Go Run Terminal Help
app > Http > Middleware > CheckAge.php
10 {
19 {
25     if (!$birthdate) {
26         // Redirect to Access Denied if the birthdate is missing
27         return redirect('/accessDenied')->with('error', 'Birthdate is required.');
```

```
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.');
```

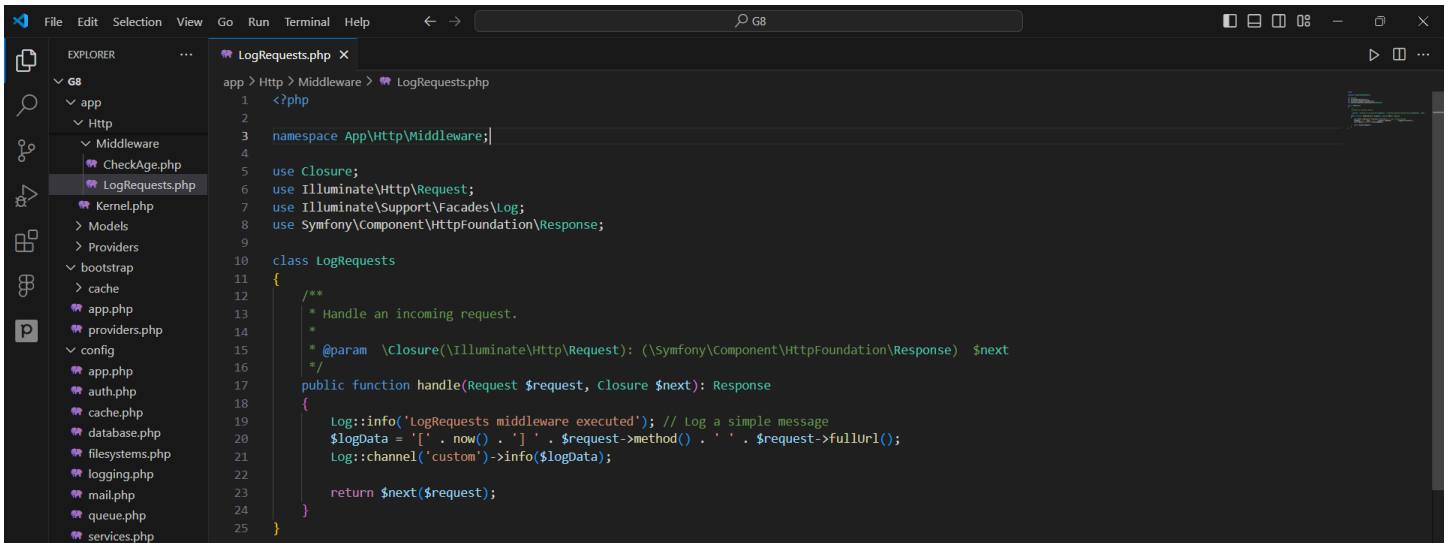
```
36     }
37
38     \Log::info("User Age: $age, Username: {$request->input('username')}");
39
40     // Check if the user is under 18
41     if ($age < 18) {
42         // Redirect to Access Denied page if under 18
43         return redirect('/accessDenied')->with('error', 'You must be 18 or older to access this page.');
```

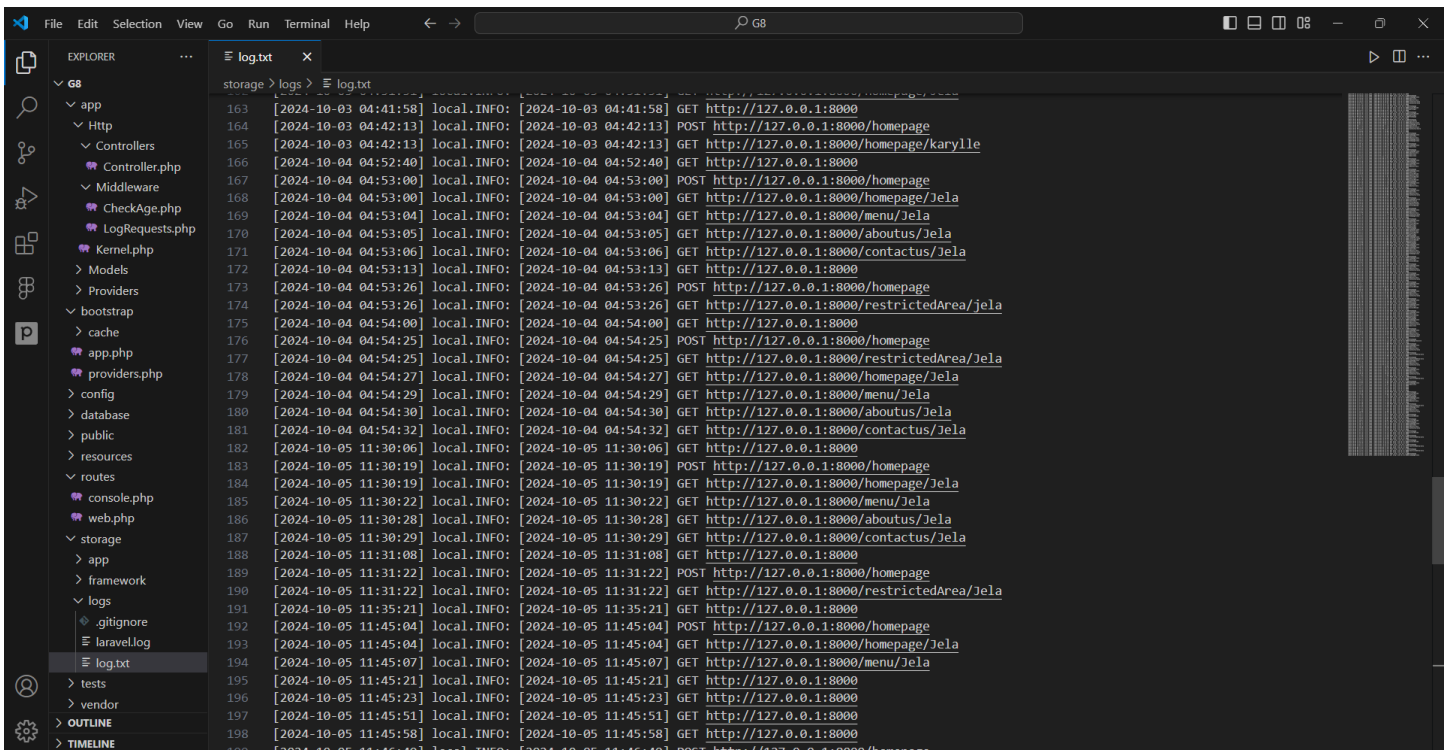
```
44     }
45     elseif ($age >= 18 && $age <= 20) {
46         // Redirect to homepage if age is between 18 and 20
47         return redirect('/homepage/' . $username);
48     }
49     elseif ($age >= 21) {
50         \Log::info("Redirecting to restricted area for user: ' . $username);
51         // Redirect to restricted area if 21 or older
52         return redirect("/restrictedArea/" . $username);
53     }
54
55     // Proceed if 18 or older
56     return $next($request);
57 }
58 }
```





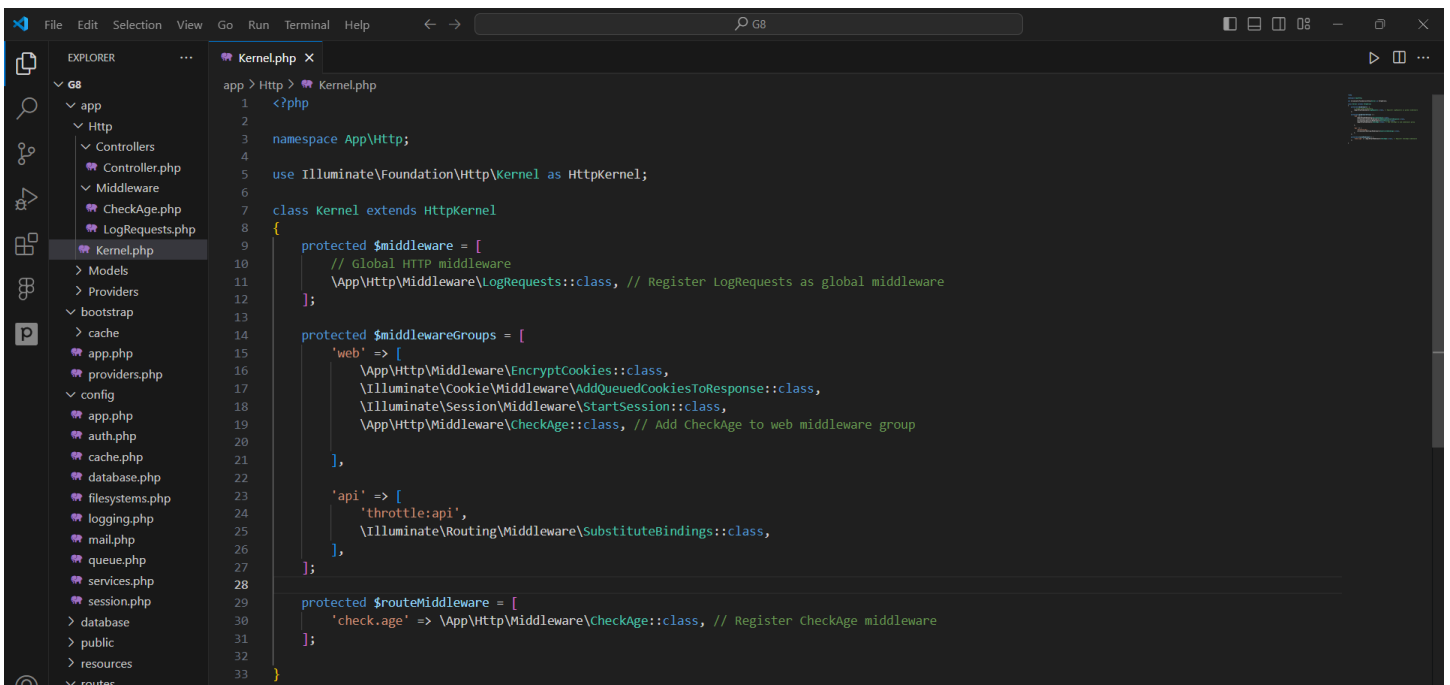
- The **LogRequests** file logs the details of all HTTP requests, including the URL, method and timestamp which are saved and can be viewed in the **log.txt** file. This middleware file **LogRequests.php** logs all the HTTP requests, capturing the method (e.g., GET, POST) and the full URL of the request.





```
storage > logs > log.txt
163 [2024-10-03 04:41:58] local.INFO: [2024-10-03 04:41:58] GET http://127.0.0.1:8000
164 [2024-10-03 04:42:13] local.INFO: [2024-10-03 04:42:13] POST http://127.0.0.1:8000/homepage
165 [2024-10-03 04:42:13] local.INFO: [2024-10-03 04:42:13] GET http://127.0.0.1:8000/homepage/karylle
166 [2024-10-04 04:52:40] local.INFO: [2024-10-04 04:52:40] GET http://127.0.0.1:8000
167 [2024-10-04 04:53:00] local.INFO: [2024-10-04 04:53:00] POST http://127.0.0.1:8000/homepage
168 [2024-10-04 04:53:00] local.INFO: [2024-10-04 04:53:00] GET http://127.0.0.1:8000/homepage/Jela
169 [2024-10-04 04:53:04] local.INFO: [2024-10-04 04:53:04] GET http://127.0.0.1:8000/menu/Jela
170 [2024-10-04 04:53:05] local.INFO: [2024-10-04 04:53:05] GET http://127.0.0.1:8000/aboutus/Jela
171 [2024-10-04 04:53:06] local.INFO: [2024-10-04 04:53:06] GET http://127.0.0.1:8000/contactus/Jela
172 [2024-10-04 04:53:13] local.INFO: [2024-10-04 04:53:13] GET http://127.0.0.1:8000
173 [2024-10-04 04:53:26] local.INFO: [2024-10-04 04:53:26] POST http://127.0.0.1:8000/homepage
174 [2024-10-04 04:53:26] local.INFO: [2024-10-04 04:53:26] GET http://127.0.0.1:8000/restrictedArea/Jela
175 [2024-10-04 04:54:00] local.INFO: [2024-10-04 04:54:00] GET http://127.0.0.1:8000
176 [2024-10-04 04:54:25] local.INFO: [2024-10-04 04:54:25] POST http://127.0.0.1:8000/homepage
177 [2024-10-04 04:54:25] local.INFO: [2024-10-04 04:54:25] GET http://127.0.0.1:8000/restrictedArea/Jela
178 [2024-10-04 04:54:27] local.INFO: [2024-10-04 04:54:27] GET http://127.0.0.1:8000/homepage/Jela
179 [2024-10-04 04:54:29] local.INFO: [2024-10-04 04:54:29] GET http://127.0.0.1:8000/menu/Jela
180 [2024-10-04 04:54:30] local.INFO: [2024-10-04 04:54:30] GET http://127.0.0.1:8000/aboutus/Jela
181 [2024-10-04 04:54:32] local.INFO: [2024-10-04 04:54:32] GET http://127.0.0.1:8000/contactus/Jela
182 [2024-10-05 11:30:06] local.INFO: [2024-10-05 11:30:06] GET http://127.0.0.1:8000
183 [2024-10-05 11:30:19] local.INFO: [2024-10-05 11:30:19] POST http://127.0.0.1:8000/homepage
184 [2024-10-05 11:30:19] local.INFO: [2024-10-05 11:30:19] GET http://127.0.0.1:8000/homepage/Jela
185 [2024-10-05 11:30:22] local.INFO: [2024-10-05 11:30:22] GET http://127.0.0.1:8000/menu/Jela
186 [2024-10-05 11:30:28] local.INFO: [2024-10-05 11:30:28] GET http://127.0.0.1:8000/aboutus/Jela
187 [2024-10-05 11:30:29] local.INFO: [2024-10-05 11:30:29] GET http://127.0.0.1:8000/contactus/Jela
188 [2024-10-05 11:31:08] local.INFO: [2024-10-05 11:31:08] GET http://127.0.0.1:8000
189 [2024-10-05 11:31:22] local.INFO: [2024-10-05 11:31:22] POST http://127.0.0.1:8000/homepage
190 [2024-10-05 11:31:22] local.INFO: [2024-10-05 11:31:22] GET http://127.0.0.1:8000/restrictedArea/Jela
191 [2024-10-05 11:35:21] local.INFO: [2024-10-05 11:35:21] GET http://127.0.0.1:8000
192 [2024-10-05 11:45:04] local.INFO: [2024-10-05 11:45:04] POST http://127.0.0.1:8000/homepage
193 [2024-10-05 11:45:04] local.INFO: [2024-10-05 11:45:04] GET http://127.0.0.1:8000/homepage/Jela
194 [2024-10-05 11:45:07] local.INFO: [2024-10-05 11:45:07] GET http://127.0.0.1:8000/menu/Jela
195 [2024-10-05 11:45:21] local.INFO: [2024-10-05 11:45:21] GET http://127.0.0.1:8000
196 [2024-10-05 11:45:23] local.INFO: [2024-10-05 11:45:23] GET http://127.0.0.1:8000
197 [2024-10-05 11:45:51] local.INFO: [2024-10-05 11:45:51] GET http://127.0.0.1:8000
198 [2024-10-05 11:45:58] local.INFO: [2024-10-05 11:45:58] GET http://127.0.0.1:8000
199 [2024-10-05 11:45:59] local.INFO: [2024-10-05 11:45:59] POST http://127.0.0.1:8000/homepage
```

- **Global middleware** is registered under the **\$middleware** property in the Kernel class, ensuring that it runs on every single request. For example, in the provided code below, **LogRequests** is registered globally, meaning it will log all HTTP requests made to the application, whether they come from web routes or API routes. In contrast, **route-specific middleware** is registered in the **\$routeMiddleware** array and can be applied only to certain routes. Each route-specific middleware is assigned an alias, like **'check.age'** for the **CheckAge** middleware, allowing it to apply to specific routes that require an age check. This setup provides precise control over which middleware runs for which routes. For example, the **CheckAge** middleware can be applied to routes that should only be accessible to users of certain age, without affecting other routes in the application.



```
app > Http > Kernel.php
1 <?php
2
3 namespace App\Http;
4
5 use Illuminate\Foundation\Http\Kernel as HttpKernel;
6
7 class Kernel extends HttpKernel
8 {
9     protected $middleware = [
10         // global HTTP middleware
11         \App\Http\Middleware\LogRequests::class, // Register LogRequests as global middleware
12     ];
13
14     protected $middlewareGroups = [
15         'web' => [
16             \App\Http\Middleware\EncryptCookies::class,
17             \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
18             \Illuminate\Session\Middleware\StartSession::class,
19             \App\Http\Middleware\CheckAge::class, // Add CheckAge to web middleware group
20         ],
21
22         'api' => [
23             'throttle:api',
24             \Illuminate\Routing\Middleware\SubstituteBindings::class,
25         ],
26     ];
27
28     protected $routeMiddleware = [
29         'check.age' => \App\Http\Middleware\CheckAge::class, // Register CheckAge middleware
30     ];
31
32
33 }
```

- Group routes in Laravel allow you to bundle multiple routes together and apply shared configurations like middleware to all of them at once, making the code cleaner and more efficient. In this code, the **web** and **LogRequests** middleware are applied to every route inside the group, meaning all these routes benefit from session handling, CSRF protection, and logging without needing to add these middlewares individually. Grouping routes ensure that the shared logic only needs to be written once, helping to avoid repetition and making the code easier to maintain.

```
// Group routes that use the 'web' middleware and log requests
Route::middleware(['web', LogRequests::class])->group(function () {
```

- This code snippet is part of the **CheckAge** middleware, which verifies the user's age and redirects them to the appropriate page based on the age provided. First, it checks if the user's age is below 18. If so, the user is redirected to the **Access Denied** page with an error message stating that they must be 18 or older to access the content. If the user's age is between 18 and 20, they are redirected to the **Homepage** using their username. In this case, they can also access other pages like the **Menu, About Us and Contact Us** pages. For users who are 21 or older, a log entry is created to note that the user is being redirected, and they are then sent to the **Restricted Area** page. Additionally, they can also access the other pages like the **Menu, About Us and Contact Us Pages**. Each redirection corresponds to specific age ranges, ensuring that only users who meet the age criteria can access the restricted areas, with the logic handling various age groups appropriately. If none of the conditions apply, the request is passed on to proceed further.

```
// 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);
```

- This set of routes handles the **Restricted Page, Homepage and Access Denied Page** each of which is shown based on the user's age.
 - The **Restricted Area Page** is for users who are age 21 years old and above. If a user meets this age requirement, they will be shown this page at the `/restrictedArea/{username}` route.
 - The **Homepage** is for users aged between 18 and 20. If the user falls within this age range, they will be redirected to the homepage after logging in, which is handled by the `/homepage` route. The **CheckAge** middleware is applied here to verify the user's age.
 - Lastly, the **Access Denied Page** is for users below 18. If a user is younger than 18, the middleware will redirect them to the `/accessDenied` route, which displays a message indicating they are not allowed access.

```
// Route for restricted area with dynamic username
Route::get('/restrictedArea/{username}', function ($username) {
    return view('restrictedArea', ['username' => $username]);
})->name('restrictedArea');

// Middleware for age verification, restricted access, or homepage access post-login
Route::post('/homepage', function (Request $request) {
    return view('homepage', ['username' => $request->input('username')]);
})->middleware(CheckAge::class);

// Access Denied route
Route::get('/accessDenied', function () {
    return view('accessDenied');
})->name('accessDenied');
```

