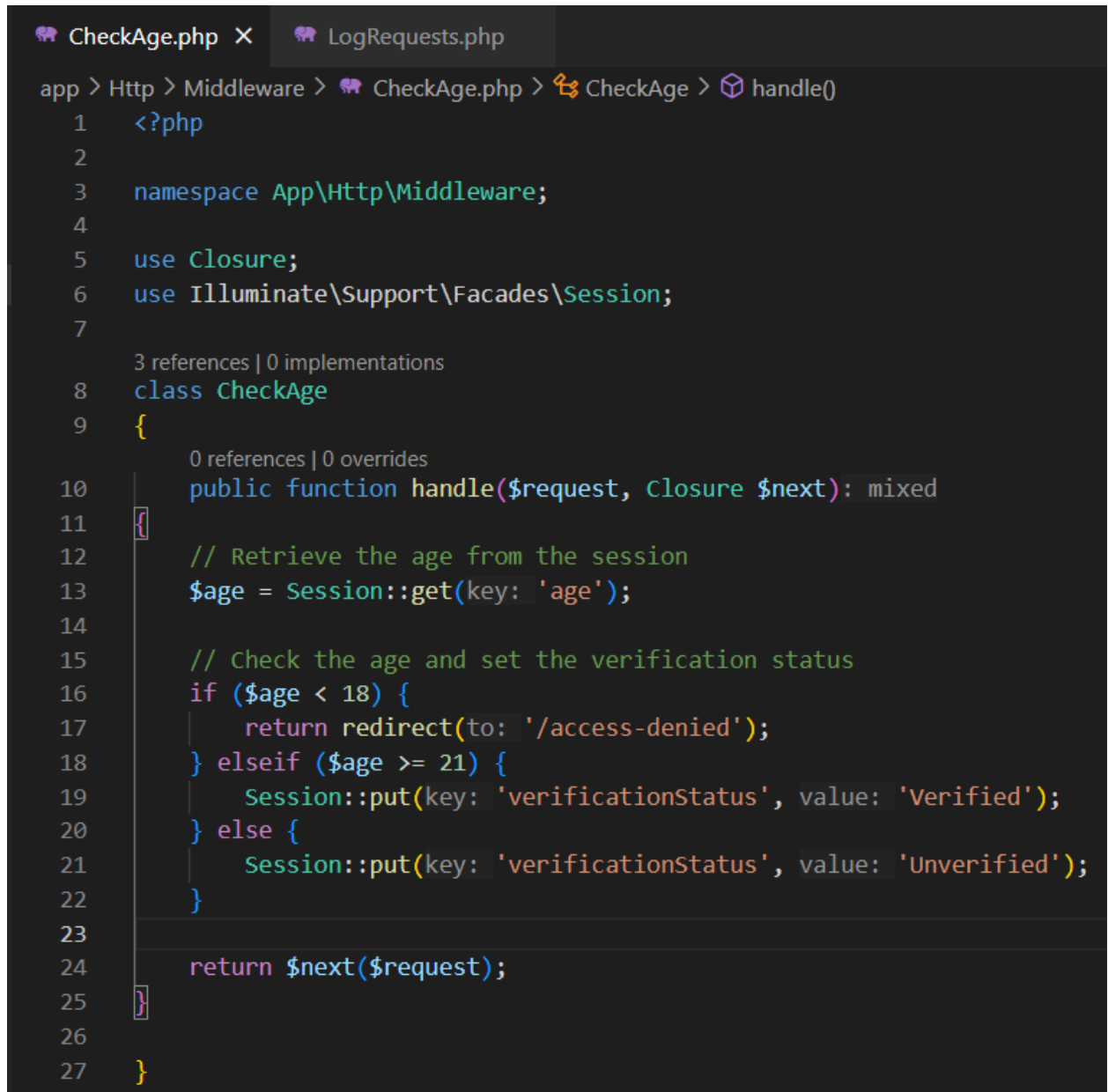


Web Development Lab: Middleware

CheckAge Middleware



```
app > Http > Middleware > CheckAge.php > CheckAge > handle()
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Support\Facades\Session;
7
8  class CheckAge
9  {
10     0 references | 0 overrides
11     public function handle($request, Closure $next): mixed
12     {
13         // Retrieve the age from the session
14         $age = Session::get(key: 'age');
15
16         // Check the age and set the verification status
17         if ($age < 18) {
18             return redirect(to: '/access-denied');
19         } elseif ($age >= 21) {
20             Session::put(key: 'verificationStatus', value: 'Verified');
21         } else {
22             Session::put(key: 'verificationStatus', value: 'Unverified');
23         }
24
25         return $next($request);
26     }
27 }
```

- The **CheckAge** middleware checks the user's age from the session. If they're under **18**, it redirects them to an **access-denied** page; if **21 or older**, it sets them as **'Verified'**; **otherwise**, they're marked as **'Unverified'**. After this, the request continues to the next step.

LogRequests Middleware

```
app > Http > Middleware > LogRequests.php > ...
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  4 references | 0 implementations
10 class LogRequests
11 {
12     /**
13      * Handle an incoming request.
14      *
15      * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
16      */
17     0 references | 0 overrides
18     public function handle(Request $request, Closure $next): Response
19     {
20         $username = $request->session()->get(key: 'username', default: 'Guest');
21         $ipAddress = $request->ip();
22         $age = $request->session()->get(key: 'age', default: 'Unknown');
23         $verificationStatus = $request->session()->get(key: 'verificationStatus', default: 'Not Verified');
24
25         // Prepare log data with timestamp, HTTP method, full URL, and username
26         $logData = sprintf(
27             format: "[%s] %s %s - Username: %s, IP: %s, Age: %s, Verification Status: %s\n",
28             values: now()->toDateTimeString(), // Time
29             $request->method(), // HTTP method (GET, POST, etc.)
30             $request->fullUrl(), // Full URL of the request
31             $username,
32             $ipAddress,
33             $age,
34             $verificationStatus
35         );
36
37         // Check if the user is accessing the login route
38         if ($request->is(patterns: '/user') && $request->method() === 'POST') {
39             // Set verification status based on age
40             $verificationStatus = ($age >= 21) ? 'Verified' : 'Not Verified';
41
42             // Log entry for successful login
43             $logData = sprintf(
44                 format: "[%s] Access Granted - IP: %s, Username: %s, Age: %s, Verification Status: %s\n",
45                 values: now()->toDateTimeString(), // Time
46                 $ipAddress, // User's IP address
47                 $username, // Username from the session
48                 $age, // Age from the session
49                 $verificationStatus // Correct verification status
50             );
51         }
52
53         // Check if access is denied (you can adjust the condition based on your actual access denial logic)
54         if ($request->is(patterns: '/access-denied')) {
55             // Log entry for access denied
56             $logData = sprintf(
57                 format: "[%s] Access Denied - IP: %s, Username: %s, Age: %s\n",
58                 values: now()->toDateTimeString(), // Time
59                 $ipAddress, // User's IP address
60                 $username, // Username from the session
61                 $age // Age from the session
62             );
63         }
64
65         // Attempt to log the request details in log.txt file
66         if (file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND) === false) {
67             // Log an error message if writing to the log file fails
68             \Log::error(message: 'Failed to write to log.txt');
69         }
70     }
71 }
```

```

69     // Proceed with the next middleware/handler in the stack
70     return $next($request);
71 }
72 }

```

- The **LogRequests** middleware logs important details about each request, like the user's **username, IP address, age**, and the full URL. It also logs when a user **successfully logs in** or when **access is denied**. All the information is saved in a **log.txt** file, and if it can't write to the file, an error is reported.

Kernel.php

```

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

```

- Register middleware as global middleware.

```

34 protected $middlewareGroups = [
35     'web' => [
36         \App\Http\Middleware\EncryptCookies::class,
37         \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
38         \Illuminate\Session\Middleware\StartSession::class,
39         // For CSRF protection
40         \Illuminate\Session\Middleware\AuthenticateSession::class,
41         \Illuminate\View\Middleware\ShareErrorsFromSession::class,
42         \App\Http\Middleware\VerifyCsrfToken::class,
43         \Illuminate\Routing\Middleware\SubstituteBindings::class,
44     ],
45
46     'api' => [
47         'throttle:api',
48         \Illuminate\Routing\Middleware\SubstituteBindings::class,
49     ],
50 ];
51
52 /**
53  * The application's route middleware.
54  *
55  * These middleware may be assigned to groups or used individually.
56  *
57  * @var array
58  */
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 ];

```

➤ Register middleware as route-specific middleware.

- The **Kernel** class manages middleware in the Laravel app. It includes **global middleware** that runs on every request, like checking for maintenance mode or logging requests, and organizes other middleware into groups for web and API routes. It also lists specific middleware that can be applied to individual routes, like authentication or age checking.

web.php

```
web.php X
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use Illuminate\Support\Facades\Session;
5  use App\Http\Middleware\CheckAge; // Import CheckAge middleware
6  use App\Http\Middleware\LogRequests; // Import LogRequests middleware
7
```

- Import the **CheckAge** and **LogRequests** middleware classes from the **App\Http\Middleware** namespace into the current file.

```
36 // Route to handle user login or continue as guest
37 Route::get(uri: '/user', action: function (): Factory|View {
38     // Get the username from the request or default to 'Guest'
39     $username = request()->input(key: 'username', default: 'Guest');
40
41     // Validate username to ensure it only contains alphabetic characters
42     if (!preg_match(pattern: '/^[A-Za-z]+$/', subject: $username)) {
43         $username = 'Guest';
44     }
45
46     // Store the username in session
47     Session::put(key: 'username', value: $username);
48
49     // Retrieve age and verification status
50     $age = Session::get(key: 'age');
51     $verificationStatus = Session::get(key: 'verificationStatus', default: 'Not Verified');
52
53     // Return the user view with the username and verification status
54     return view(view: 'user', data: ['username' => $username, 'age' => $age, 'verificationStatus' => $verificationStatus]);
55 });
```

- Since we added age in the form, and there is a need to verify that age, we added code in the logic to **retrieve user's age** and **verification status** from the session and then display them along with the username on the user view page.

```
57 // Route to handle user login or registration submission
58 Route::post(uri: '/user', action: function (): Redirector|RedirectResponse {
59     // Get the username and age from the request
60     $username = request()->input(key: 'username', default: 'Guest');
61     $age = request()->input(key: 'age'); // Retrieve the age input from the form
62
63     // Validate username to ensure it only contains alphabetic characters
64     if (!preg_match(pattern: '/^[A-Za-z]+$/', subject: $username)) {
65         $username = 'Guest';
66     }
67
68     // Store the username and age in the session
69     Session::put(key: 'username', value: $username);
70     Session::put(key: 'age', value: $age); // Store age in session
71
72     // Set verification status based on age
73     $verificationStatus = ($age >= 21) ? 'Verified' : 'Not Verified';
74     Session::put(key: 'verificationStatus', value: $verificationStatus);
75
76     // Redirect after login or registration
77     return redirect(to: '/user');
78 });
```

- Added code to store user's age in the session

- If the age is 21 or older, it sets the verification status to "Verified"; otherwise, it sets it to "Not Verified." This status is also saved in the session.

```

80 // Route to handle user logout
81 Route::get(uri: '/logout', action: function (): Redirector|RedirectResponse {
82     // Clear the username from session
83     Session::forget(keys: 'username');
84     Session::forget(keys: 'age'); // Clear age from session
85     Session::forget(keys: 'verificationStatus'); // Clear verification status
86
87     // Redirect to the homepage or login page
88     return redirect(to: '/');
89 });

```

- When the user visits the **/logout** route, the code **removes the username, age, and verificationStatus from the session**. This effectively logs the user out by clearing their stored information.
- After clearing the session data, the user is **redirected** to the **register** page (/). This means they will no longer have access to their previous session data and will see the main or login screen.

```

91 // Group routes that require age validation and logging
92 Route::middleware(middleware: [LogRequests::class, CheckAge::class])>group(callback: function (): void {
93     Route::get(uri: '/welcome', action: function (): Factory|View {
94         return view(view: 'welcome');
95     });
96
97     Route::get(uri: '/dashboard', action: function (): Factory|View {
98         return view(view: 'dashboard');
99     });
100
101     Route::get(uri: '/user', action: function (): Factory|View {
102         // Get the username from the session
103         $username = Session::get(key: 'username', default: 'Guest');
104
105         // Retrieve age and verification status
106         $age = Session::get(key: 'age');
107         $verificationStatus = Session::get(key: 'verificationStatus', default: 'Not Verified'); // Default value if not set
108
109         // Return the user view with the username and verification status
110         return view(view: 'user', data: ['username' => $username, 'age' => $age, 'verificationStatus' => $verificationStatus]);
111     });
112 });

```

- Sets up a **group of routes** that require **both logging of requests and age verification**, ensuring that users must meet specific criteria before accessing the associated views.

```

114 // Route to handle when access is denied
115 Route::get(uri: '/access-denied', action: function (): Factory|View {
116     // Get the username from the session
117     $username = Session::get(key: 'username', default: 'Guest');
118
119     // Clear the username and age from the session
120     Session::forget(keys: 'username');
121     Session::forget(keys: 'age');
122
123     return view(view: 'access-denied');
124 });

```

- **Removes the username and age** from the session so that values are reset if they want to go back to register again. Then, it displays the **access-denied view**, informing the user that they **do not have permission** to access the requested content.

access-denied.blade.php

```
access-denied.blade.php X
resources > views > access-denied.blade.php
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
```

- Blade template for the "Access Denied" page. This is where users are redirected if they are below 18.

register.blade.php

```
register.blade.php X
resources > views > register.blade.php
14 <div class="content-wrapper">
25     <div class="form-container">
26         <form action="{{ url(path: '/user') }}" method="POST">
30             <div class="input-container">
31                 <div class="input-group">
33                     <path fill="#B197FC" d="M224 256A128 128 0 1 0 224 0a128 128 0 1 0 0 256zm-45.7 48C79.8 304 0
34                     </svg>
35                     <input type="text" name="username" id="username" pattern="[A-Za-z]+" title="Only alphabetic charac
36                 </div>
37                 <div class="input-group">
38                     <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 512 512" fill="#B197FC">
39                         <path d="M466.5 83.711-192-80c-8.203-3.422-17.81-3.422-26.02 0l-192 80C45.1 89.28 32 108.1 32
40                     </svg>
41                     <input type="number" name="age" id="age" placeholder="Enter your age" required min="1" max="100">
42                 </div>
43             </div>
```

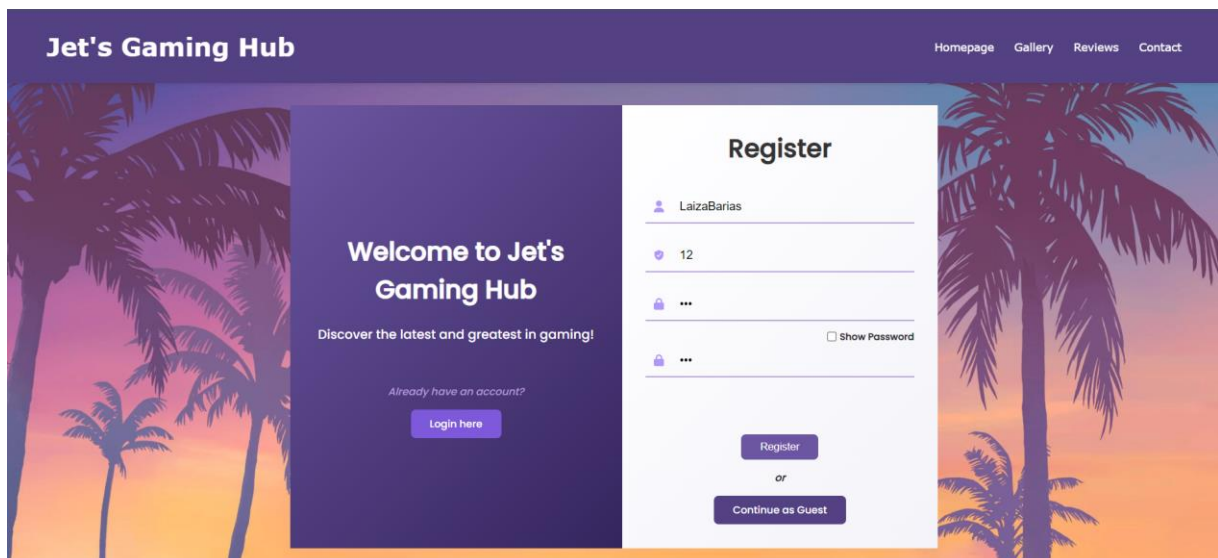
- In the register page, we added an input field for the age.

user.blade.php

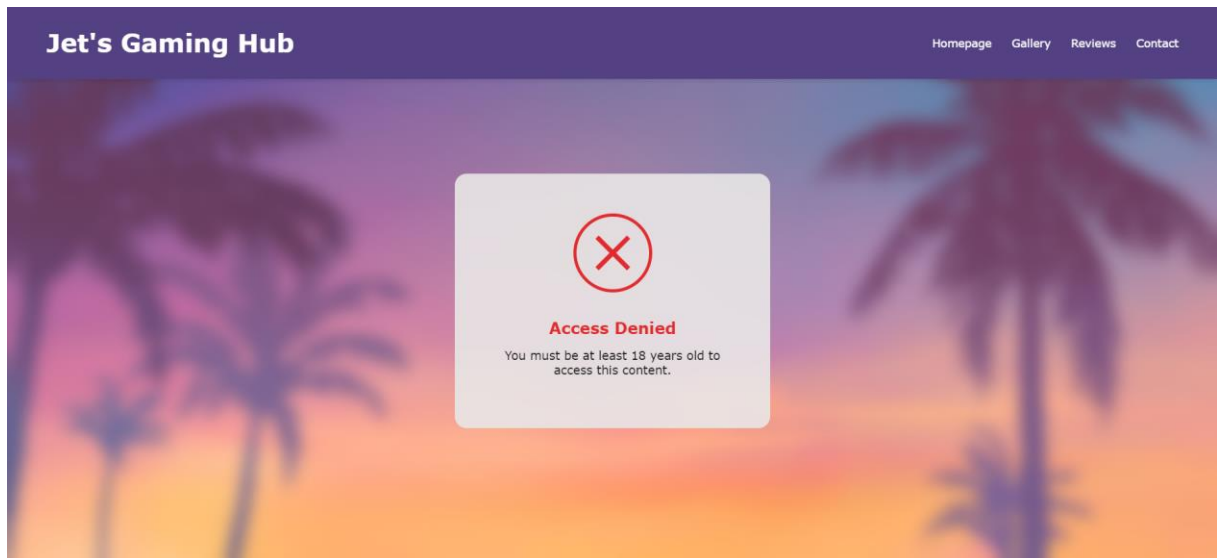
```
user.blade.php X
resources > views > user.blade.php
1 @extends(view: 'Components.Layout')
2
3 @php
4     $title = 'Welcome User';
5 @endphp
6
7 @section(section: 'body-class', content: 'homepage')
8
9 @section(section: 'content')
10     <h1>Welcome, {{ $username }}</h1>
11     <p>Enjoy exploring the gaming hub!</p>
12     <p>Status: {{ $verificationStatus }}</p> <!-- Display the verification status -->
13     <button class="button" onclick="location.href='{{ url(path: '/gallery') }}'">Explore Gallery</button>
14 @endsection
```

- Added **verification status**

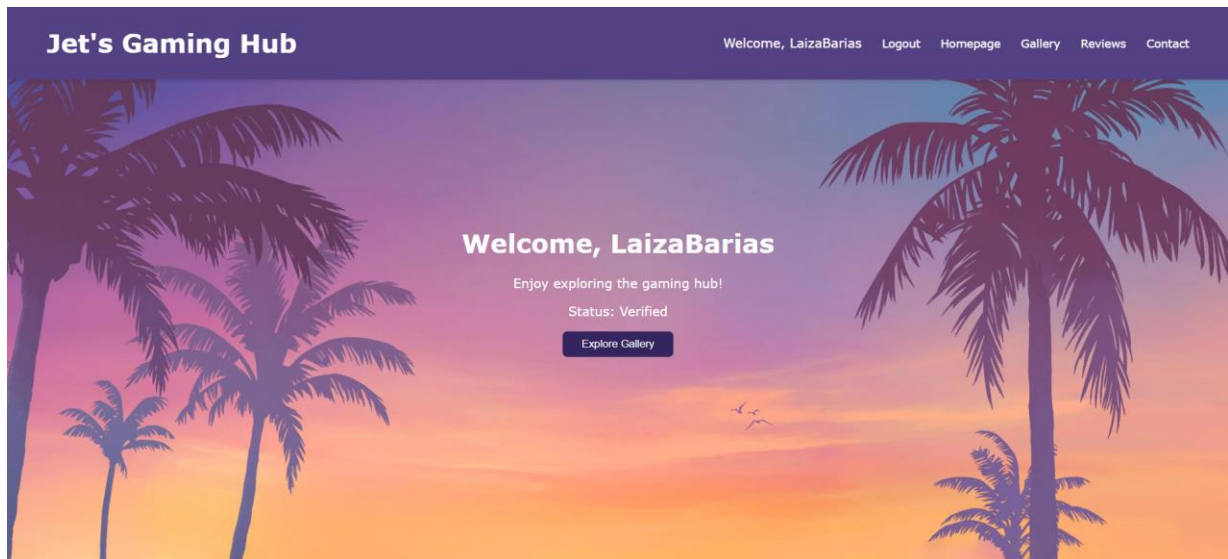
VIEWS



- **Register page** where users will input their information.



- **access-denied page** where users are redirected if they are under 18 years old



- If user's age passes the verification, they will be redirected to user page. Here, it will show the **username** along with the **verification status**.

****Other views (Gallery of Games, Game Reviews, Contact Us page) were not changed/modified but will still be included for documentation purposes.***

Gallery of Games

Explore some very great set of games.



Grand Theft Auto V

Grand Theft Auto V is a 2013 action-adventure game developed by Rockstar North and published by Rockstar Games. It is the seventh main entry in the Grand Theft Auto series, following 2008's Grand Theft Auto IV, and the fifteenth instalment overall.

Set within the fictional state of San Andreas, based on Southern California, the single-player story follows three protagonists—retired bank robber Michael De Santa (Ned Luke), street gangster Franklin Clinton (Shawn Fonteno), and drug dealer and gunrunner Trevor Philips (Steven Ogg), and their attempts to commit heists while under pressure from a corrupt government agency and powerful criminals. Players freely roam San Andreas's open world countryside and fictional city of Los Santos, based on Los Angeles.

See review

Valorant

Game Reviews

See my reviews on some great games.



Grand Theft Auto V

As a fan of gaming, I think that it would be difficult to have not heard of the Grand Theft Auto Series. The franchise has many games under it's belt, and while controversial, it still remains the modern go to game for the majority of the gaming community. The latest in the franchise, Grand Theft Auto V, is Rockstar Games love letter to all the fans of the franchise. The game is groundbreaking in its innovation and was ahead of its time. Due to this, the game had an immense popularity leading up to and on its release day. Does it live up to the hype? Lets find out.

Personally, I am a huge fan of open-world games and GTA V did not disappoint. The game is very amazing graphically and gameplay wise. The game has many things for the player to do which makes its replay value higher. Add to that the amazing story and cast of characters just makes the game rich in development.

Contact Us

Name:

Email Address:

We'll never share your email with anyone else.

Message:

Submit