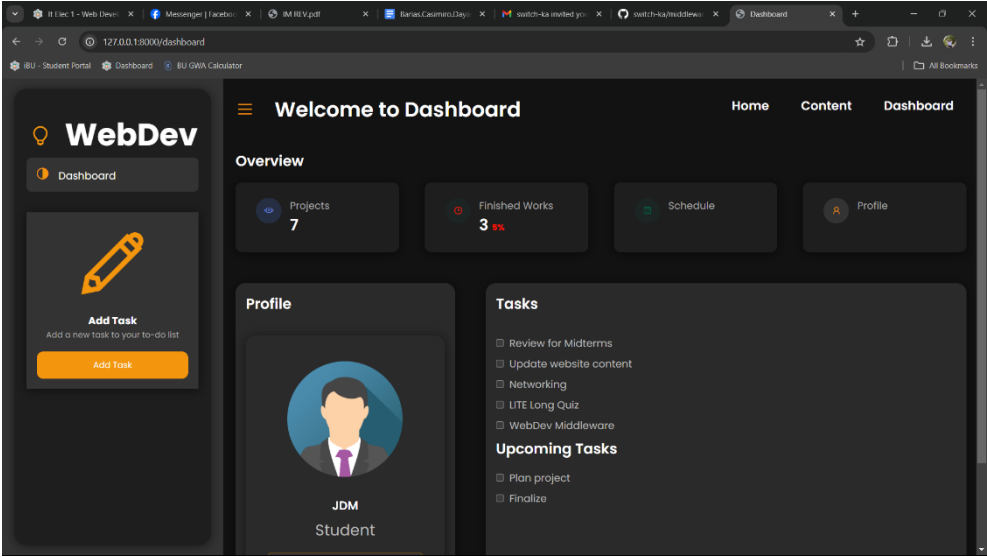


Juliane Rebecca S. Dayandante

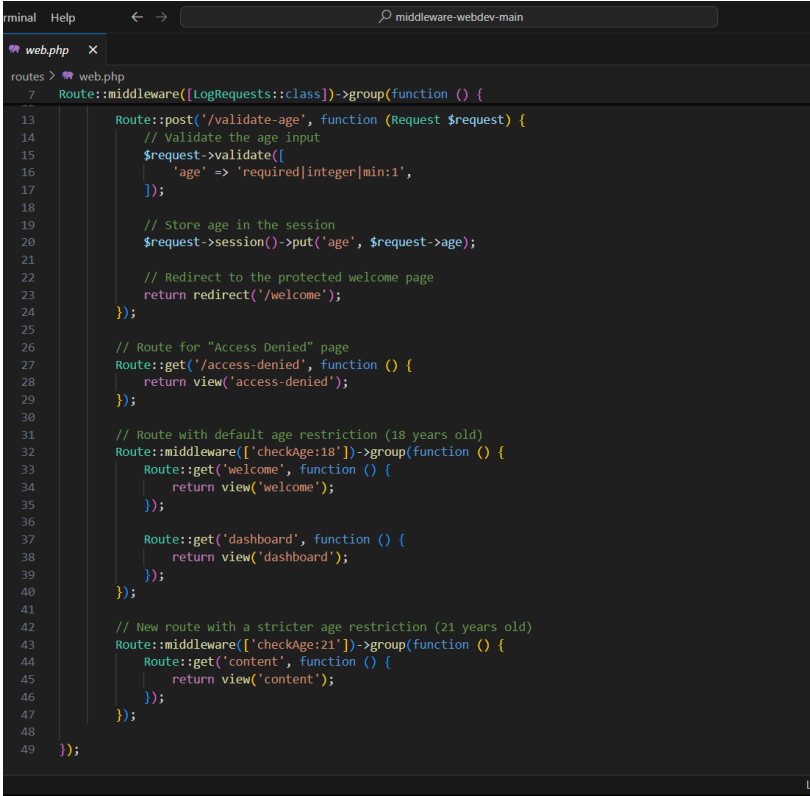
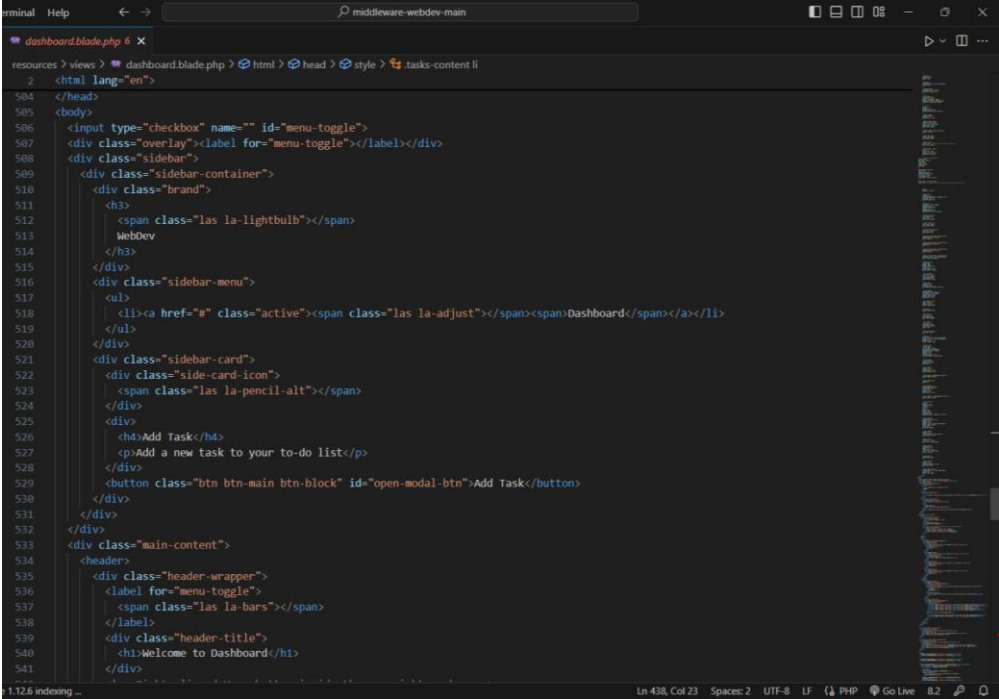
BSIT – 3C

Web Development



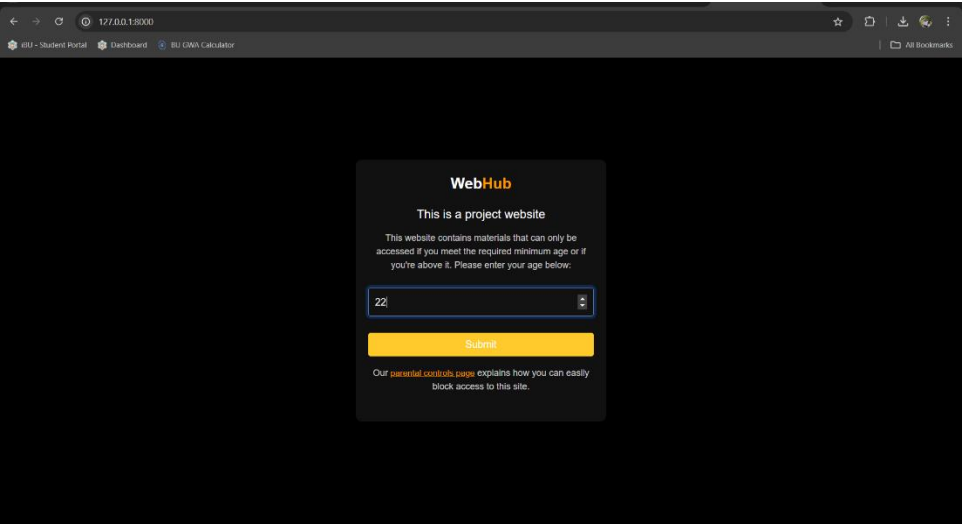
dashboard.php

I was assigned to design and develop the dashboard interface in this task. It features a sidebar for navigation, a section for displaying task and project overviews, and modals for adding new tasks and editing profile details. I implemented dynamic task management using JavaScript.



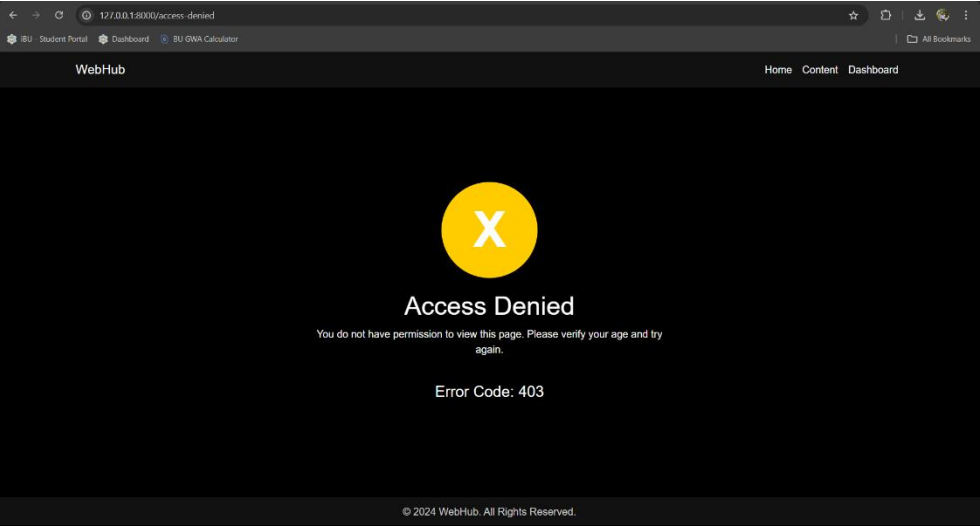
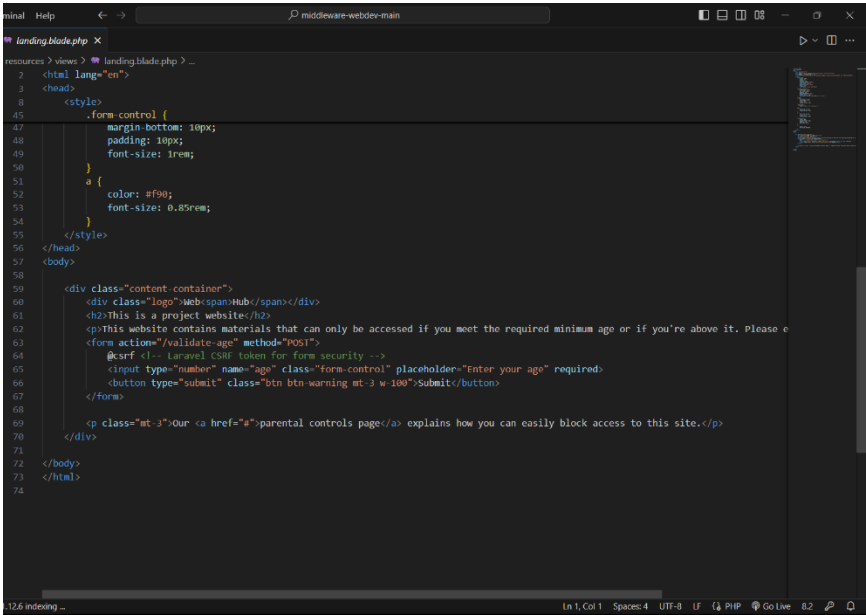
web.php

This is the routes with middleware to validate user age and control access. The system handles age verification on form submission and restricts access to specific pages based on age thresholds (18+ for welcome and dashboard, 21+ for content). The middleware LogRequests is applied to log all requests in this group.



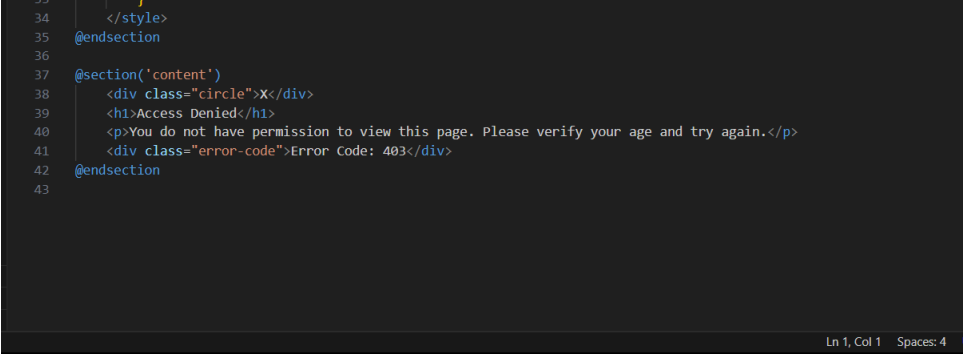
landing.blade.php

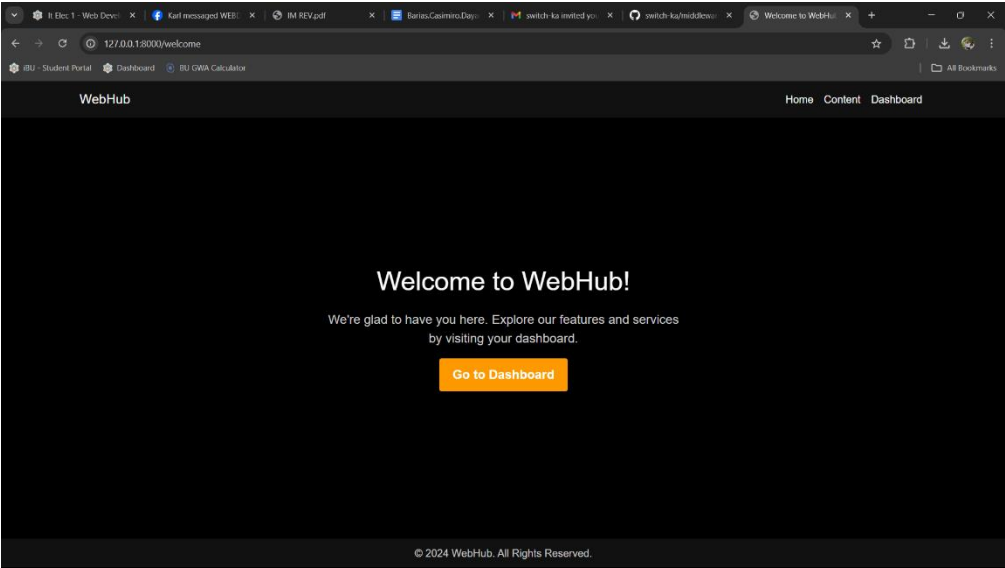
This page provides an age verification form for users to access restricted content. It features a "WebHub" logo, a brief description, and a form where users can input their age. The form uses Laravel's CSRF token for security and submits the data for validation. Additionally, there's a link to a "parental controls" page, allowing users to manage access



access-denied.blade.php

This section displays an "Access Denied" page for users who don't meet the required age. It shows an "X" symbol, the title "Access Denied," and a message explaining that the user doesn't have permission to view the page. It also includes an error code "403" to indicate a forbidden access error.



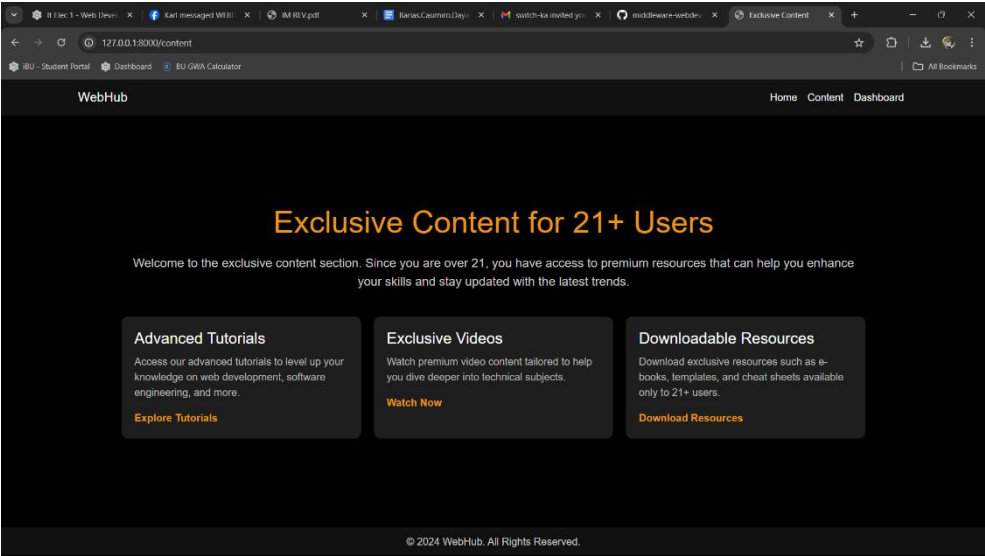


welcome.blade.php

After successfully entering the required age, users are redirected to this "Welcome to WebHub" page. It greets the user with a welcome message and provides a brief introduction to the site's features and services. There's also a button that directs users to their dashboard, allowing them to explore further.

```
Terminal  Help  ← →  middleware-webdev-main

welcome.blade.php x
resources > views > welcome.blade.php > ...
6  <style>
7      h1 {
8          font-size: 2.5rem;
9          margin-bottom: 20px;
10     }
11     p {
12         font-size: 1.2rem;
13         color: #ccc;
14     }
15     .btn-custom {
16         background-color: #f90;
17         border-color: #f90;
18         padding: 10px 20px;
19         font-size: 1.2rem;
20         color: #fff;
21         font-weight: bold;
22         text-decoration: none;
23     }
24     .btn-custom:hover {
25         background-color: #ffb84d;
26         border-color: #ffb84d;
27     }
28 </style>
29 @endsection
30
31 @section(section: 'content')
32     <h1>Welcome to WebHub!</h1>
33     <p>We're glad to have you here. Explore our features and services by visiting your dashboard.</p>
34     <a href="/dashboard" class="btn btn-custom">Go to Dashboard</a>
35 @endsection
36
```



content.blade.php

This section is exclusively for users who are 21 years old and above. It features premium content, including advanced tutorials, exclusive videos, and downloadable resources designed to enhance technical skills. If a user is only 18, they will receive an "Access Denied" message when attempting to view this page but can still access the "Welcome" page.

```
terminal Help ← → middleware-webdev-main

content.blade.php X
resources > views > content.blade.php > ...
6      <style>
43      .premium-item a {
45          font-weight: bold;
46          text-decoration: none;
47      }
48  </style>
49  @endsection
50
51  @section('content')
52      <h1>Exclusive Content for 21+ Users</h1>
53      <p class="content-description">
54          Welcome to the exclusive content section. Since you are over 21, you have access to premium resources that can help you enhance y
55      </p>
56
57      <div class="premium-content">
58          <div class="premium-item">
59              <h3>Advanced Tutorials</h3>
60              <p>Access our advanced tutorials to level up your knowledge on web development, software engineering, and more.</p>
61              <a href="#">Explore Tutorials</a>
62          </div>
63          <div class="premium-item">
64              <h3>Exclusive Videos</h3>
65              <p>Watch premium video content tailored to help you dive deeper into technical subjects.</p>
66              <a href="#">Watch Now</a>
67          </div>
68          <div class="premium-item">
69              <h3>Downloadable Resources</h3>
70              <p>Download exclusive resources such as e-books, templates, and cheat sheets available only to 21+ users.</p>
71              <a href="#">Download Resources</a>
72          </div>
73      </div>
74  @endsection
75
```

TerminalHelp←→middleware-webdev-main

CheckAge.php X

app > Http > Middleware > CheckAge.php > ...

```
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7
8  class CheckAge
9  {
10     public function handle(Request $request, Closure $next, $minAge): mixed
11     {
12         // Retrieve the age from the session
13         $age = $request->session()->get(key: 'age');
14
15         // If no age is set or the age is less than the minimum required, deny access
16         if (is_null(value: $age) || $age < $minAge) {
17             return redirect(to: 'access-denied');
18         }
19
20         // If the age meets the requirement, proceed to the next middleware/request
21         return $next($request);
22     }
23 }
24
```

CheckAge.php

This code defines the CheckAge middleware in a Laravel application, which uses a parameter to enforce age restrictions. The *handle* method takes a \$minAge parameter, allowing routes to specify the minimum required age for access. It checks the user's age from the session; if the age is not set or is below the specified minimum, the user is redirected to an "Access Denied" page. If the age requirement is met, the request continues to the welcome page.

terminalHelp←→middleware-webdev-main

LogRequests.php X

app > Http > Middleware > LogRequests.php > PHP > LogRequests > handle()

```
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Log;
8
9  class LogRequests
10 {
11     public function handle(Request $request, Closure $next): mixed
12     {
13         // Prepare the log data with the timestamp, HTTP method, and full URL
14         $logData = sprintf(
15             format: "[%s] Method: %s | URL: %s\n",
16             values: now()->toDateTimeString(),
17             $request->method(),
18             $request->fullUrl()
19         );
20
21         // Append the log entry to log.txt
22         file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND);
23
24         // Proceed to the next middleware/request handler
25         return $next($request);
26     }
27 }
28
```

LogRequests.php

The LogRequests middleware logs details of incoming HTTP requests, including the timestamp, HTTP method, and URL. It saves this information in a log.txt file in the storage/logs directory. After logging the request, it allows the application to continue processing normally.