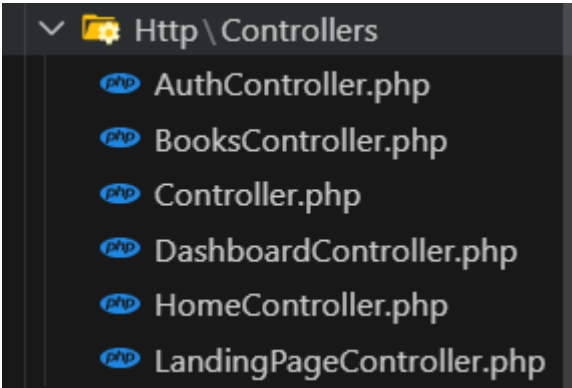
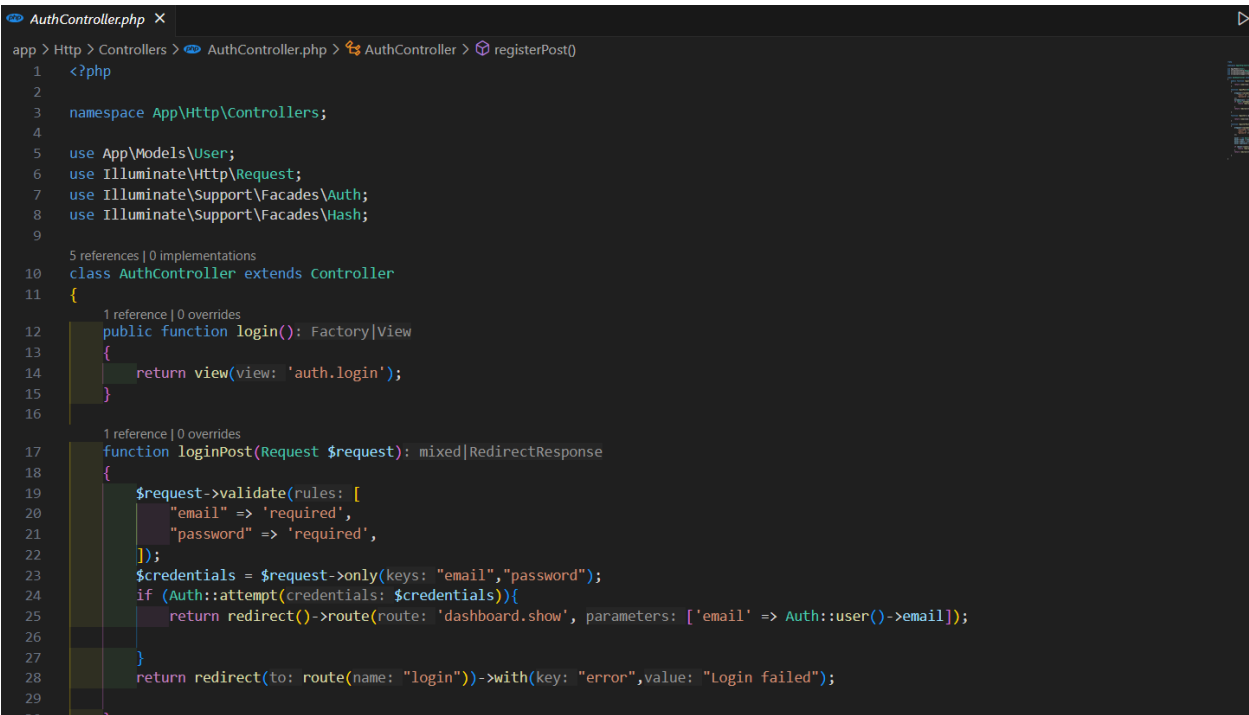


I. Documentation

- 5 controllers ( the Controller.php is given when project is created)



- AuthController handles logic in creating an account and logging in, I already discussed this controller in the last laboratory and there’s not yet any modification (still the same) was made to this one



- The BooksController handles the retrieval and display of a list of books (in our landing-page, home). It has a method called loadBooks(), which returns a predefined array of books, each with detailed attributes like book ID, title, author, rating, cover image path, description, number of ratings, number of reviews, genres, and comments. Each book is represented as an associative array, with nested arrays for comments, allowing for multiple comments on each book, each containing the reader's name, comment text, and time of posting (comments was not yet utilized).
- Purpose of this controller method is to prepare a set of book data that could be passed to a view in a Laravel application. The loadBooks() method could be called when a user visits a page that lists books.

```
BooksController.php X
app > Http > Controllers > BooksController.php > BooksController > show()
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class BooksController extends Controller
8 {
9     public function loadBooks(): array
10     {
11         $posts = [
12             [
13                 'book_id' => 1,
14                 'title' => "Omniscient Reader's Viewpoint",
15                 'author' => 'Sing Shong',
16                 'rating' => 4.5,
17                 'cover' => 'asset/images/orv.jpg',
18                 'description' => "Only I know the end of this world.One day our MC finds himself stuck in the world of his favorite webnovel",
19                 'num_ratings' => 121212,
20                 'num_reviews' => 20212,
21                 'genres' => ['Fantasy', 'Manga', 'Light Novel', 'Science Fiction', 'Manhwa', 'Fiction'],
22                 'comments' => [
23                     [
24                         'reader' => 'Renato Jr.',
25                         'text' => 'An incredible read! So many twists and unforgettable moments!',
26                         'time' => '1 day ago'
27                     ],
28                 ],
29                 'book_id' => 2,
30                 'title' => 'Lord of the Mysteries',
```

- The show method is designed to display the details of a specific book based on the book's unique identifier (book\_id). When a request is made to view a particular book, this method retrieves the relevant book's information from an array of books and returns it in a structured format (book-details view). It takes a book\_id parameter, searches for the matching book in the posts array (the above method), and returns the detailed information, including the title, author, rating, description, genres, comments, and other related attributes. If no book is found with the provided book\_id, it would return a 404 page.

```
BooksController.php X
app > Http > Controllers > BooksController.php > BooksController > show()
7 class BooksController extends Controller
9     public function loadBooks(): array
200
201     ];
202 ];
203 return $posts;
204 }
205
206
207
208
209
210     public function show($id): Factory|View
211     {
212
213         $posts = $this->loadBooks();
214
215         $book = collect(value: $posts)->firstWhere(key: 'book_id', operator: $id);
216
217         if (!$book) {
218             abort(code: 404);
219         }
220
221         return view(view: 'book-details', data: compact(var_name: 'book'));
222
223
224     }
225 }
226
227
```

- The DashboardController index method creates a new instance of the BooksController. This instance is then used to call the loadBooks() method, which retrieves the array of book data (including some comments/ recent reviews that is seen in the dashboard only for now). After fetching the data, the index method passes this data to the dashboard view using the compact() function which allows the book data, stored in the \$posts variable, to be made available to the view (which the view uses foreach)

```
DashboardController.php X
app > Http > Controllers > DashboardController.php > DashboardController > index()
1  <?php
2
3  namespace App\Http\Controllers;
4  use App\Http\Controllers\BooksController;
5
6  use Illuminate\Http\Request;
7
8  2 references | 0 implementations
9  class DashboardController extends Controller
10 {
11     1 reference | 0 overrides
12     public function index(): Factory|View
13     {
14         $contentController = new BooksController();
15         $posts = $contentController->loadBooks();
16         return view('dashboard', data: compact('posts'));
17     }
18 }
```

- logic of the HomeController and LandingPageController is just the same in the dashboard (above) as it all must display books

```
HomeController.php X
app > Http > Controllers > HomeController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4  use App\Http\Controllers\BooksController;
5  use Illuminate\Http\Request;
6
7  2 references | 0 implementations
8  class HomeController extends Controller
9  {
10     1 reference | 0 overrides
11     public function index(): Factory|View
12     {
13         $contentController = new BooksController();
14         $posts = $contentController->loadBooks();
15         return view('home', data: compact('posts'));
16     }
17 }
```

```
LandingPageController.php X
app > Http > Controllers > LandingPageController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4 use App\Http\Controllers\BooksController;
5 use Illuminate\Http\Request;
6
7 class LandingPageController extends Controller
8 {
9     1 reference | 0 overrides
10     public function index(): Factory|View
11     {
12         $contentController = new BooksController();
13         $posts = $contentController->loadBooks();
14         return view('landing-page', data: compact('posts'));
15     }
16 }
17
```

- @foreach directive is used to loop through the \$posts array passed from the LandingPageController. The @foreach directive iterate over the array and display each item within the loop.
- In the first @foreach block, the array\_slice(\$posts, 0, 4) function is used to select the first 4 items from the \$posts array. This slice is displayed in the "Popular Now" section. Each \$post represents a book, and for each iteration, it generates a card that displays the book's cover image, title, author, rating, and information about the last reader.
- Similarly, in the second @foreach block, array\_slice(\$posts, 4, 4) slices the \$posts array starting from index 4 to get the next 4 books, which are displayed in the "Latest Books" section.
- Image below is not the full snippet, I didn't include the latest books section screenshot as it's just the same logic for the popular now

```
landing-page.blade.php X
resources > views > landing-page.blade.php
10 <section class="popular-now-section">
11     <div class="container">
12         <div class="d-flex justify-content-between align-items-center mb-4">
13
14         </div>
15
16         <div class="row">
17             @foreach (array_slice($posts, 0, 4) as $post)
18                 <div class="col-md-3 mb-4">
19                     <div class="card">
20                         <a href="{ route('books.show', parameters: $post['book_id']) }">
21                             
22                             <div class="card-body">
23                                 <h5 class="book-title">{{ $post['title'] }}</h5>
24                                 <h6 class="book-author">{{ $post['author'] }}</h6>
25                                 <div class="star-rating">
26                                     @for ($i = 1; $i <= 5; $i++)
27                                         <i class="bi bi-star" {{ $i <= $post['rating'] ? '-fill' : '' }}></i>
28                                     @endfor
29                                 </div>
30                                 @if (isset($post['last_reader']))
31                                     <div class="last-reader mt-3">
32                                         
37                                         <span class="reader">{{ $post['last_reader']['name'] }}</span>
38                                         <span class="time">{{ $post['last_reader']['time_ago'] }}</span>
39                                     </div>
40                                 @endif
41                             </div>
42                         </a>
43                     </div>
44                 </div>
45             @endforeach
46         </div>
47     </div>
48 </section>
49
50
```

- Same thing happens to the home and dashboard view, since the logic of its corresponding controller is the same. But in the dashboard the comments are iterated to the array passed

```

home.blade.php X
resources > views > home.blade.php
9      <x-carousel />
10
11     <section class="popular-now-section">
12         <div class="container">
13             <div class="d-flex justify-content-between align-items-center mb-4">
14                 <h3 class="text-left">Popular Now</h3>
15                 <a href="" class="view-all-link">View All</a>
16             </div>
17
18             <div class="row">
19                 @foreach (array_slice(array: $posts, offset: 0, length: 4) as $post)
20                     <div class="col-md-3 mb-4">
21                         <div class="card">
22                             <a href="{{ route(name: 'books.show', parameters: $post['book_id']) }}">
23                                 
24                                 <div class="card-body">
25                                     <h5 class="book-title">{{ $post['title'] }}</h5>
26                                     <h6 class="book-author">{{ $post['author'] }}</h6>
27                                     <div class="star-rating">
28                                         @for ($i = 1; $i <= 5; $i++)
29                                             <i class="bi bi-star{{ $i <= $post['rating'] ? '-fill' : '' }}"></i>
30                                         @endfor
31                                     </div>
32                                     @foreach ($post['comments'] as $comment)
33                                         <div class="last-reader mt-3">
34                                             
39                                             <span class="reader">{{ $comment['reader'] }}</span>
40                                             <span class="time">{{ $comment['time'] }}</span>
41                                         </div>
42                                     @endforeach
43                                 </div>
44                             </a>
45                         </div>
46                     </div>
47                 @endforeach
48             </div>

```

```

dashboard.blade.php X
resources > views > dashboard.blade.php
11     <main>
12
13     </section>
14
15     <section class="recent_reviews">
16         <div class="container">
17             <div class="d-flex justify-content-between align-items-center mb-4">
18                 <h3 class="text-left">Recent Reviews</h3>
19                 <button id="view-all-reviews" class="view-all-link">View All</button>
20             </div>
21             <div class="row" id="review-list">
22                 @foreach (array_slice(array: $posts, offset: 0, length: 4) as $post)
23                     <div class="col-md-3 mb-4 review-item">
24                         <div class="card">
25                             <div class="reviews">
26                                 <h5 class="book_title">{{ $post['title'] }}</h5>
27                                 @foreach ($post['comments'] as $comment)
28                                     <span class="text">{{ $comment['text'] }}</span>
29                                     <span class="time">{{ $comment['time'] }}</span>
30                                 @endforeach
31                             </div>
32                         </div>
33                     </div>
34                 @endforeach
35             </div>
36
37             <!-- Hidden reviews that will be shown when the "View All" button is clicked -->
38             @foreach (array_slice(array: $posts, offset: 4) as $post)
39                 <div class="col-md-3 mb-4 review-item" style="display: none;">
40                     <div class="card">
41                         <div class="reviews">
42                             <h5 class="book_title">{{ $post['title'] }}</h5>
43                             @foreach ($post['comments'] as $comment)
44                                 <span class="text">{{ $comment['text'] }}</span>
45                                 <span class="time">{{ $comment['time'] }}</span>
46                             @endforeach
47                         </div>
48                     </div>
49                 </div>
50             @endforeach
51         </div>
52     </section>

```

- When a user clicks on a book from the home, it triggers the show method in the BookController. This method calls the loadBooks function to retrieve all available book data (array), and uses the firstWhere method from the collect helper to find the book that matches the book\_id passed from the clicked link. If no matching book is found it will return a 404 error.
- Once correct book is located, its details are passed to the book-details view using compact('book').
- In the book-details view, the book's details are accessed through the book array. The book's cover image is displayed using {{ asset(\$book['cover']) }}, while the title and author are shown with {{ \$book['title'] }} and {{ \$book['author'] }}. The star rating is generated based on the book's rating, with a loop used to display the appropriate number of filled or empty stars. Also, the book's description and genres are displayed, with each genre being rendered as a clickable link.

```
book-details.blade.php X
resources > views > book-details.blade.php
10 <section class="book-details-section">
11 <div class="container">
12 <div class="row" id="card-style">
13 <div class="col-md-1"></div>
14 <div class="col-md-8">
15 <h2 class="book-title">{{ $book['title'] }}</h2>
16 <h4 class="book-author">by {{ $book['author'] }}</h4>
17 <div class="star-rating mb-3">
18 <span class="rating-number me-2">{{ number_format(num: $book['rating'], decimals: 1) }}</span>
19 @for ($i = 1; $i <= 5; $i++)
20 @if ($i <= floor(num: $book['rating']))
21 <i class="bi bi-star-fill star-icon"></i>
22 @elseif ($i == ceil(num: $book['rating']) && $book['rating'] - floor(num: $book['rating']) >= 0.5)
23 <i class="bi bi-star-half star-icon"></i>
24 @else
25 <i class="bi bi-star star-icon"></i>
26 @endif
27 @endfor
28 <span class="num-ratings">{{ number_format(num: $book['num_ratings']) }} ratings</span>
29 <span class="separator">•</span>
30 <span class="num-reviews">{{ number_format(num: $book['num_reviews']) }} reviews</span>
31 </div>
32 <p class="book-description"><span id="syn">Synopsis:</span><br>{!! nl2br(string: e(value: $book['description'])) !!}</p>
33 <p class="book-genres">
34 <span id="genres-title">Genres:</span>
35 @foreach ($book['genres'] as $genre)
36 <a href="#" class="genre-item">{{ $genre }}</a>
37 @if (!loop->last)
38 &nbsp;
39 @endif
40 @endforeach
41 </p>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
```

- Routes defined in the code below are for loading and displaying book content. The first two routes, /home/content and /dashboard/content, are associated with the loadBooks method in the BooksController, which retrieves and returns the list of books (array) to be displayed on the home and dashboard view. Routes are named home.books and dashboard.books, for easy reference within the application.
- Third route, /books/{id}, uses the show method of the BooksController, which is responsible for displaying detailed information about a specific book. {id} parameter represents the unique identifier of the book, and this route is named books.show. When user visits a specific book's URL (e.g /books/4), the show method will be triggered, and the corresponding book details will be retrieved and displayed.

```
// Authenticated Routes (for logged-in users)
Route::middleware(middleware: 'auth')->group(callback: function (): void {
    Route::get(uri: '/contact', action: function (): Factory|View {
        return view(view: 'contact');
    });

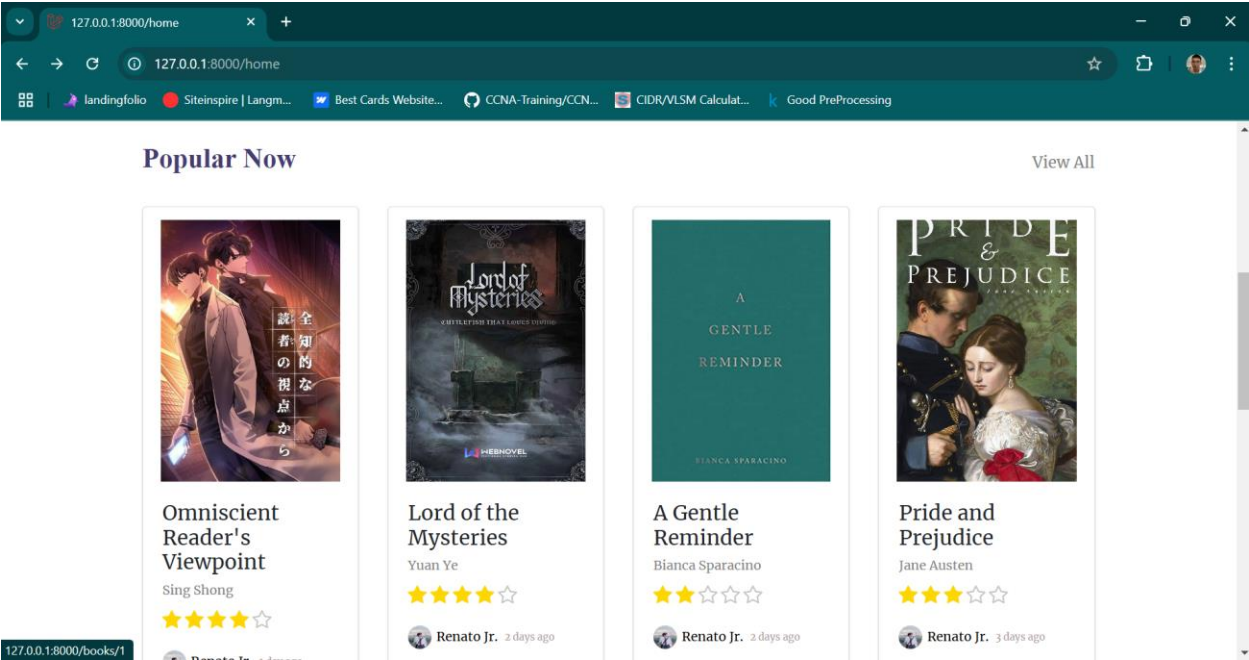
    Route::get(uri: '/home', action: [HomeController::class, 'index']->name(name: 'home.index');
    Route::get(uri: '/dashboard', action: [DashboardController::class, 'index']->name(name: 'dashboard.show');

    Route::post(uri: '/logout', action: function (): Redirector|RedirectResponse {
        auth()->logout(); // Logs out the user
        return redirect(to: route(name: 'landing-page'));
    })->name(name: 'logout');

    // Content loading routes
    Route::get(uri: '/home/content', action: [BooksController::class, 'loadBooks']->name(name: 'home.books');
    Route::get(uri: '/dashboard/content', action: [BooksController::class, 'loadBooks']->name(name: 'dashboard.books');

    Route::get(uri: '/books/{id}', action: [BooksController::class, 'show']->name(name: 'books.show');
});
```

Rendered Page: (included only some as the rest of the page are still the same as before)






127.0.0.1:8000/books/1

127.0.0.1:8000/books/1

landingfolioSiteinspire | Langm...Best Cards Website...CCNA-Training/CCN...CIDR/VLSM Calculat...Good PreProcessing



讀者  
の  
視  
点  
か  
ら

Omniscient Reader's Viewpoint

by Sing Shong

4.5★★★★☆121,212 ratings • 20,212 reviews

Synopsis:  
Only I know the end of this world. One day our MC finds himself stuck in the world of his favorite webnovel. What does he do to survive? It is a world struck by catastrophe and danger all around.  
  
His edge? He knows the plot of the story to the end. Because he was the sole reader that stuck with it. Read his story to see how he survives!

Genres: Fantasy Manga Light Novel Science Fiction Manhwa Fiction

Add to Favorites

Rate this Book

127.0.0.1:8000/books/2

127.0.0.1:8000/books/2

landingfolioSiteinspire | Langm...Best Cards Website...CCNA-Training/CCN...CIDR/VLSM Calculat...Good PreProcessing



Lord of Mysteries  
CHILLERISH THAT LEAVES TREMBLING

Lord of the Mysteries

by Yuan Ye

4.0★★★★☆121,212 ratings • 20,212 reviews

Synopsis:  
Waking up to be faced with a string of mysteries, Zhou Mingrui finds himself reincarnated as Klein Moretti in an alternate Victorian era world where he sees a world filled with machinery, cannons, dreadnoughts, airships, difference machines, as well as Potions, Divination, Hexes, Tarot Cards, Sealed Artifacts... The Light continues to shine but the mystery has never gone far. Follow Klein as he finds himself entangled with the Churches of the world—both orthodox and unorthodox—while he slowly develops newfound powers thanks to the Beyonder potions.

Genres: Mystery Fantasy Light Novel Steampunk Fiction

Add to Favorites

Rate this Book

127.0.0.1:8000/dashboard

127.0.0.1:8000/dashboard

landingfolioSiteinspire | Langm...Best Cards Website...CCNA-Training/CCN...CIDR/VLSM Calculat...Good PreProcessing

Recent Reviews

View All

Omniscient Reader's Viewpoint

"An incredible read! So many twists and unforgettable moments!"

1 day ago

Lord of the Mysteries

"A thrilling mystery that keeps you on the edge of your seat. Highly recommend!"

2 days ago

A Gentle Reminder

"It's a beautiful book with gentle life lessons. Very thought-provoking."

2 days ago

Pride and Prejudice

"A classic with timeless romance and witty characters. A must-read!"

3 days ago

Reviewed Books

View All

