Documentation (Controllers)

Controllers

DashboardController

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Support\Facades\Auth;
use App\Models\User;

3 references | 0 implementations
class DashboardController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory|View
    {
        return view(view: 'dashboard');
    }

    1 reference | 0 overrides
    public function showBook($book_id): Factory|View
    {
        // Dummy book data
        $books = [
            1 => [
                'title' => 'The Great Gatsby',
                'author' => 'F. Scott Fitzgerald',
                'review' => 'A fascinating story of the Jazz Age.',
                'rating' => 5
            ],
            2 => [
                'title' => '1984',
                'author' => 'George Orwell',
                'review' => 'A dystopian novel about totalitarianism.',
                'rating' => 4
            ]
        ];

        // Check if the book exists in the dummy data
        if (!isset($books[$book_id])) {
            abort(code: 404, message: 'Book not found');
        }

        // Return the view with the book details
        return view(view: 'book', data: ['book' => $books[$book_id]]);
    }
}
```

HomeController

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

2 references | 0 implementations
class HomeController extends Controller
{
    1 reference | 0 overrides
    public function index (): Factory|View
    {
        return view (view: 'landing-page');
    }
}
```

HomeController – returns the landing-page.

DashboardController
- The index() method returns the dashboard view when users visit the /dashboard route.
- The showBook($book_id) method retrieves and displays details for a specific book (from a hardcoded array) based on the book_id passed in the URL. If the book doesn't exist, it triggers a 404 error.

AuthController

```php
1    <?php
2
3    namespace App\Http\Controllers;
4
5    use App\Models\User;
6    use Illuminate\Http\Request;
7    use Illuminate\Support\Facades\Auth;
8    use Illuminate\Support\Facades\Hash;
9
     5 references | 0 implementations
10   class AuthController extends Controller
11   {
         1 reference | 0 overrides
12       public function login(): Factory|View
13       {
14           return view(view: 'auth.login');
15       }
16
         1 reference | 0 overrides
17       function loginPost(Request $request): mixed|RedirectResponse
18       {
19           $request->validate(rules: [
20               "email" => 'required',
21               "password" => 'required',
22           ]);
23           $credentials = $request->only(keys: "email","password");
24           if (Auth::attempt(credentials: $credentials)){
25               return redirect()->route(route: 'dashboard.show', parameters: ['email' => Auth::user()->email]);
26
27           }
28           return redirect(to: route(name: "login"))->with(key: "error",value: "Login failed");
29
30       }
31
         1 reference | 0 overrides
32       function register(): Factory|View{
33
34           return view(view: 'auth.register');
35       }
--
         1 reference | 0 overrides
37       function registerPost(Request $request): mixed|RedirectResponse
38       {
39           $request->validate(rules: [
40               "fullname" => 'required',
41               "email" => 'required',
42               "password" => 'required',
43           ]);
44
45           $user = new User();
46           $user->name = $request->fullname;
47           $user->email = $request->email;
48           $user->password = Hash::make(value: $request->password);
49
50           if ($user->save()){
51               return redirect(to: route(name: "login"))->with(key: "success", value: "User created successfully");
52           }
53           return redirect(to: route(name: "register"))->with(key: "error",value: "Failed to create account");
54
55       }
56
57   }
58
```

➔ Displays the login form.

➔ Validates the email and password, attempts to log the user in. If successful, redirects to the dashboard; otherwise, redirects back to login with an error.

➔ Displays the registration form.

➔ Validates the input, creates a new user, and saves it. Redirects to the login page on success, or back to register with an error.

Routes

```php
web.php  ×

routes >  web.php
   1   <?php
   2
   3   use Illuminate\Support\Facades\Route;
   4   use App\Http\Controllers\HomeController;
   5   use App\Http\Controllers\DashboardController;
   6   use App\Http\Controllers\AuthController;
```

```php
   8   // Landing Page
   9   Route::get(uri: '/', action: [HomeController::class, 'index'])->name(name: 'landing-page');
```

- When someone visits the homepage (/), it will call the index method in the HomeController. It also gives this route the name landing-page for easy reference.

```php
  12   // Authentication Routes (only for guests)
  13   Route::middleware(middleware: 'guest')->group(callback: function(): void {
  14       Route::get(uri: "/login", action: [AuthController::class, 'login'])->name(name: 'login');
  15       Route::post(uri: "/login", action: [AuthController::class, "loginPost"])->name(name: "login.post");
  16       Route::get(uri: "/register", action: [AuthController::class, "register"])->name(name: "register");
  17       Route::post(uri: "/register", action: [AuthController::class, "registerPost"])->name(name: "register.post");
  18   });
```

 - The routes are grouped under the guest middleware.
- They are only accessible to users who are not logged in. If a user is already authenticated, they will be redirected to dashboard.
- This route listens for a GET request to the /login URL, and when it's accessed, the login method inside the AuthController is executed.
- This route listens for a POST request to the /login URL. When a user submits a login form, the loginPost method inside the AuthController is executed.
*same goes for the other 2

```php
  20   // Authenticated Routes (for logged-in users)
  21   Route::middleware(middleware: 'auth')->group(callback: function(): void {
  22       Route::get(uri: '/contact', action: function (): Factory|View {
  23           return view(view: 'contact');
  24       });
  25
  26       Route::get(uri: '/home', action: function (): Factory|View {
  27           return view(view: 'home');
  28       });
  29       Route::get(uri: '/dashboard', action: [DashboardController::class, 'index'])->name(name: 'dashboard.show');
  30
  31       Route::get(uri: '/dashboard/book/{book_id}', action: [DashboardController::class, 'showBook']);
  32
  33       Route::post(uri: '/logout', action: function (): Redirector|RedirectResponse { // Logout Route
  34           Auth::logout();  // Logs out the user
  35           return redirect(to: route(name: "landing-page"));
  36       })->name(name: 'logout');
  37
  38   });
```

- The routes are grouped under the auth middleware.
- They are only accessible to authenticated users.

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

- This route listens for a GET request to the /dashboard URL. It triggers the index method of the DashboardController and returns the dashboard page. The route is named dashboard.show, making it easier to refer to in the code.

```
Route::get(uri: '/dashboard/book/{book_id}', action: [DashboardController::class, 'showBook']);
```

- This route listens for a GET request to the /dashboard/book/{book_id} URL, where {book_id} is a dynamic parameter (the ID of a specific book).
- It triggers the showBook method of the DashboardController to display the details of the book with the specified ID.

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

- This route listens for a POST request to the /logout URL.
- When accessed by an authenticated user, it logs the user out using Auth::logout() and then redirects them to the route named landing-page.
- The route is named logout for easy reference.

Rendered Pages

Landing Page

Signup Page

Welcome to BookBayarn!

Sign Up to Continue

laiza mae barias

lksdf@gmail.com

......

Register

Login with

G f X

Already have an account? Login

Login Page

Welcome Back

Login to Continue

lksdf@gmail.com

......

Login

Forgot Password

Login with

G f X

New User? Sign Up

Dashboard

BookBayarn!

Find the book you are looking for...

Logout

Home   **Dashboard**   Contact Us   Blog

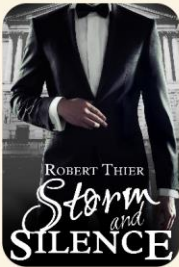| 100 Books | 1, 245 Friends | 8 Following |

Joined in November 1, 2024

Favorite Genres
Romance, Mystery/Thriller, Fantasy, Science Fiction, +5 More

My Bookshelves

Reviewed (01)   Favorites (01)

Hello, laiza mae barias

Recent Reviews

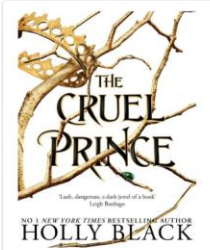| The Shank Love this book! | Lucid Dream A masterpiece! | Death Delayed Plot twist is 0.0 | The Wallflower's Revenge 10/10 |

Reviewed Books   View All

The Cruel Prince
Holly Black
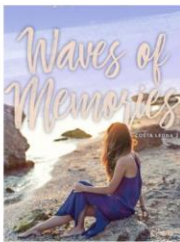★★★★☆
Renato Jr.   5 days ago

Storm and Silence
Robert Thier
★★★★☆
Renato Jr.   5 days ago

Waves of Memories
Jonaxx Stories
★★★★☆
Renato Jr.   5 days ago

The Invisible Girl
Alexisse Rose
★★★★☆
Renato Jr.   5 days ago

BookBayarn!
Find your next favorite read!

Quick Links
Home
Categories
Community
Blog

About
Developed by 3rd Year IT students of Bicol University

bookbayarn!@gmail.com

© 2024 BookBayarn! All rights reserved.

/dashboard/book/1

Book Details

The Great Gatsby

Author: F. Scott Fitzgerald

Review: A fascinating story of the Jazz Age.

Rating: 5 / 5

/dashboard/book/2

Book Details

1984

Author: George Orwell

Review: A dystopian novel about totalitarianism.

Rating: 4 / 5

/dashboard/book/3

404 | NOT FOUND

CONTACT US

We'd Love To Hear From You!
Let's Get in Touch

Full Name

Enter name

Company

Company name

Email

example@gmail.com

Phone Number

+1 (555) 000 - 0000

Address

Brgy, City, Province

Your Message

Type your message here

Send Message