**Practicum Report Web Technology**

****

**Chiko Gita Satria 2215101053**

**UNIVERSITAS PENDIDIKAN GANESHA 2023-2024**

# Table of Contents

[Table of Contents 1](#_Toc151729486)

[Tampilan Landing (guest) 3](#_Toc151729487)

[Tampilan Landing (Loged in) 4](#_Toc151729488)

[Tampilan index/mainpage(notlogin) 5](#_Toc151729489)

[Tampilan index/mainpage (login user) 6](#_Toc151729490)

[(login admin) 7](#_Toc151729491)

[Index.blade.php Code 7](#_Toc151729492)

[Layout Admin 13](#_Toc151729493)

[Layout admin (SuperAdmin) 13](#_Toc151729494)

[CRUD Implementation 14](#_Toc151729495)

[Create: 14](#_Toc151729496)

[Code 14](#_Toc151729497)

[Update: 15](#_Toc151729498)

[Code 16](#_Toc151729499)

[Delete 17](#_Toc151729500)

[Routes 17](#_Toc151729501)

[BookController (also code for the delete): 18](#_Toc151729502)

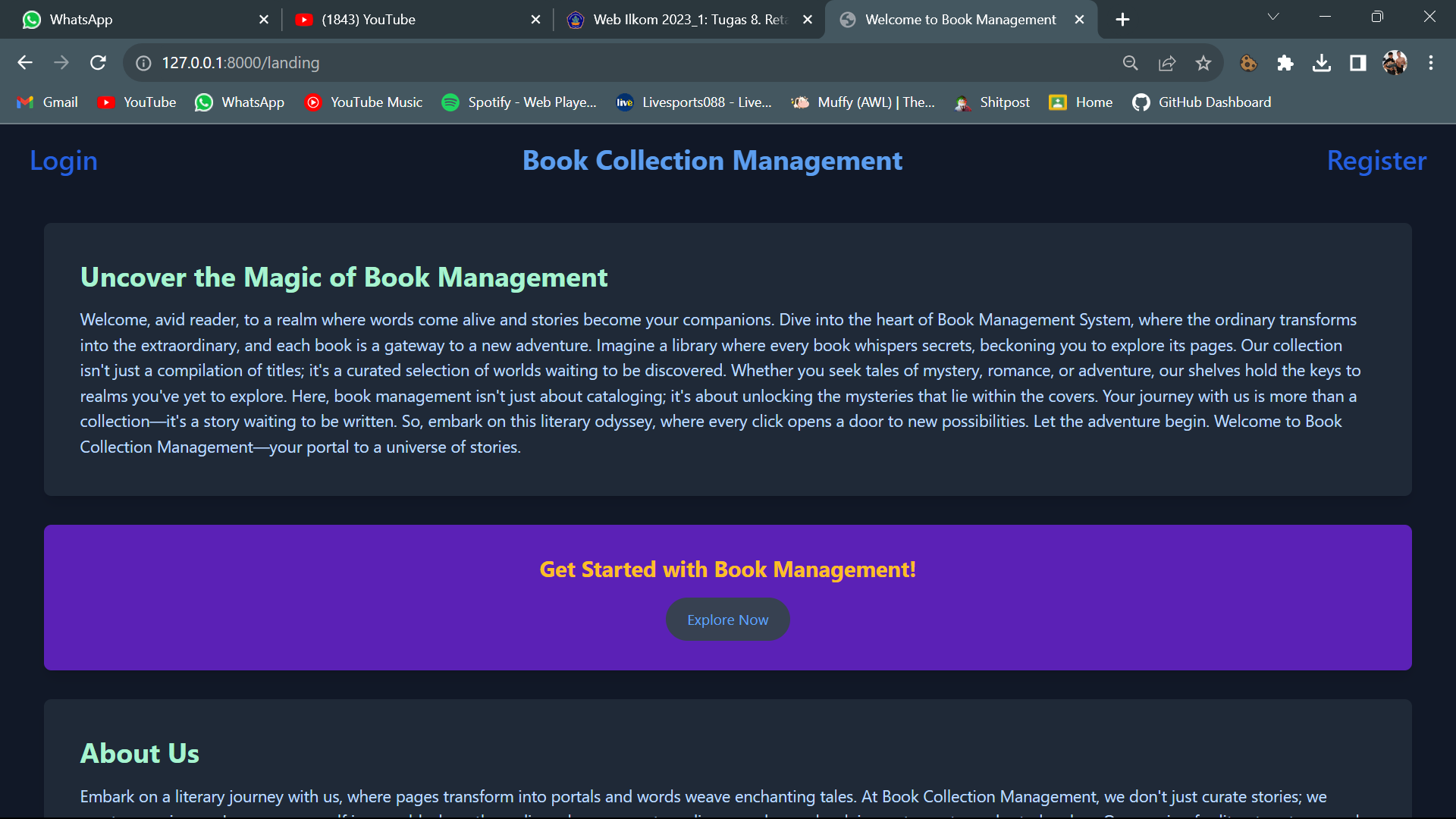
[Pagination and filtering: 22](#_Toc151729503)

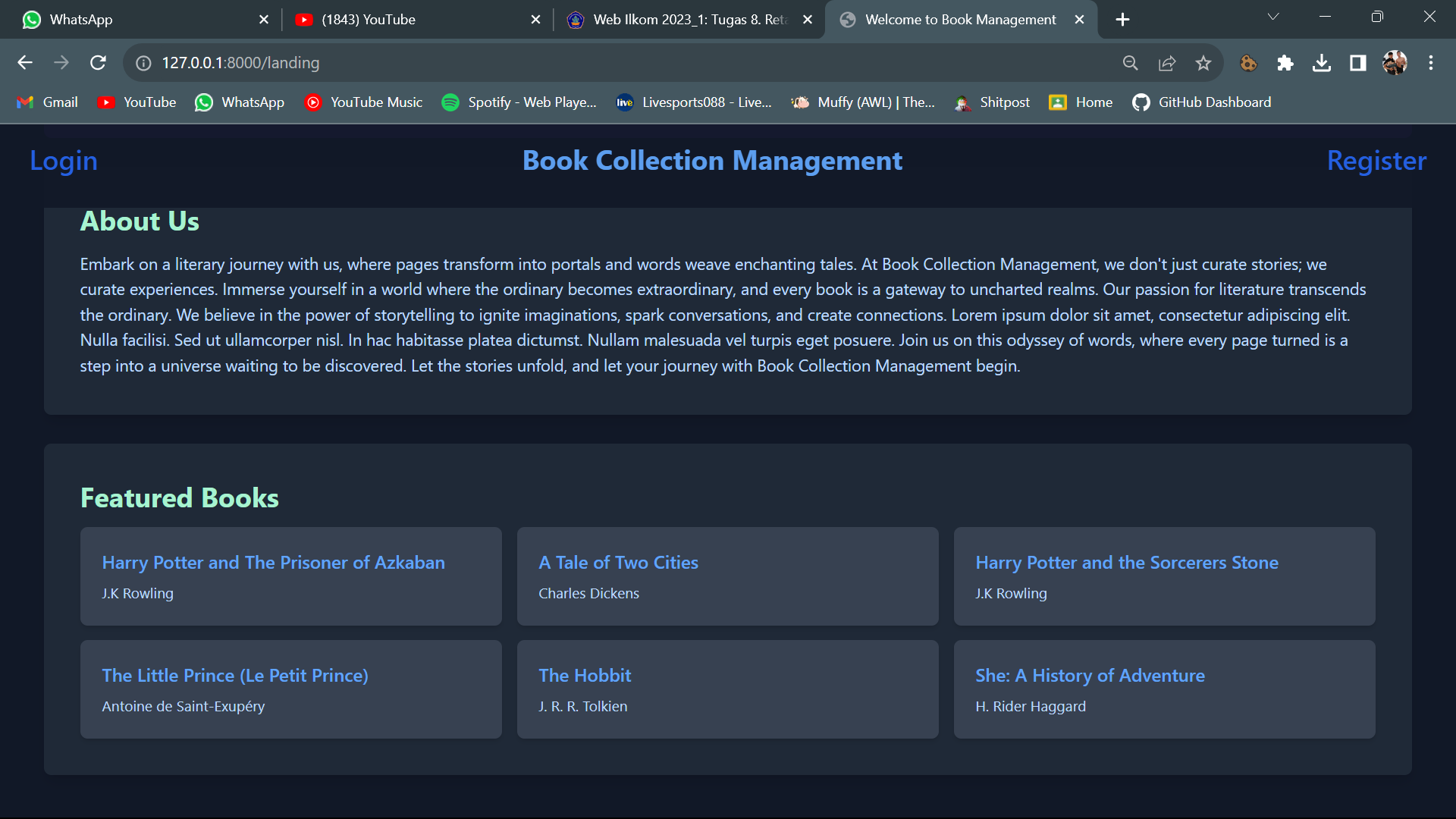
[Relationship database concept 22](#_Toc151729504)

[Model Book.php 22](#_Toc151729505)

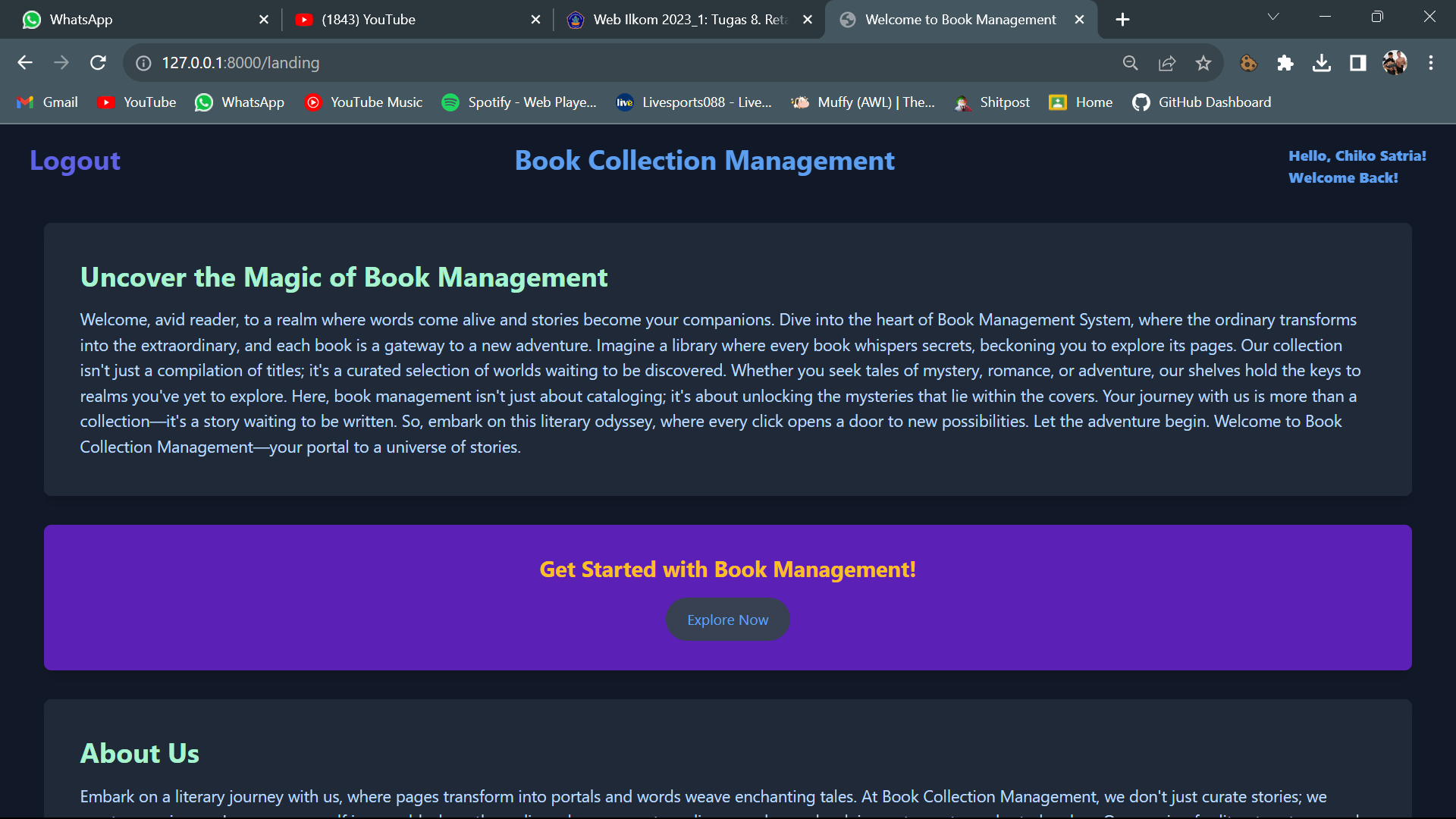
[Migration 24](#_Toc151729506)

# Tampilan Landing (guest)

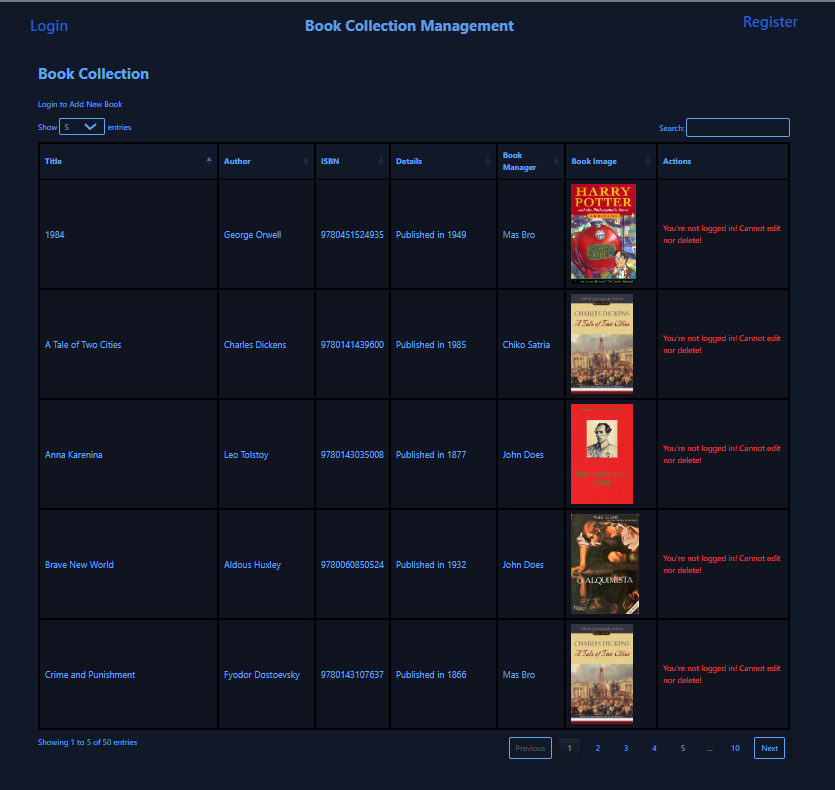




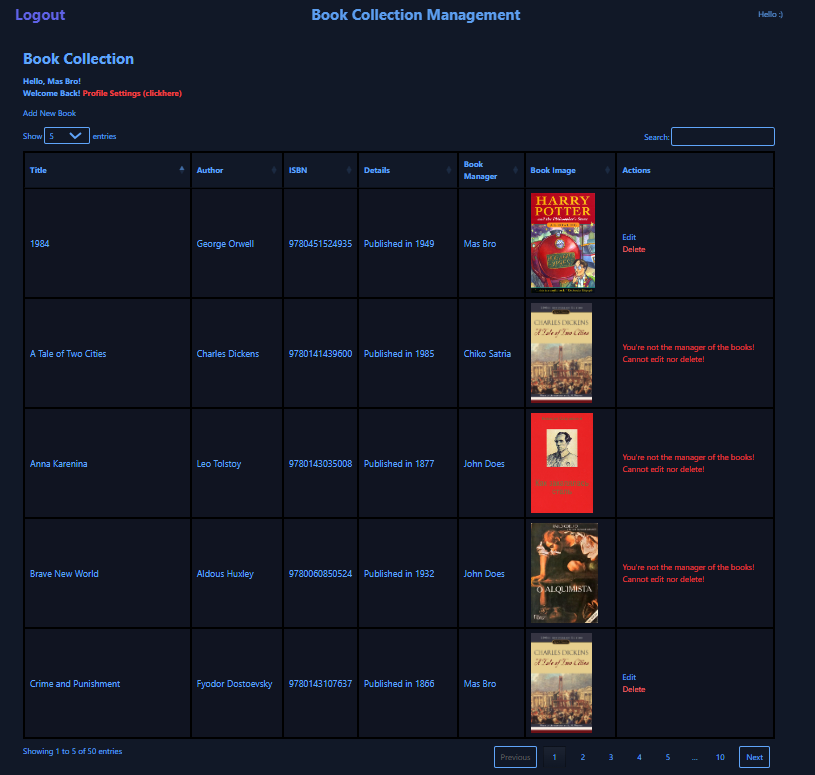
## Tampilan Landing (Loged in)



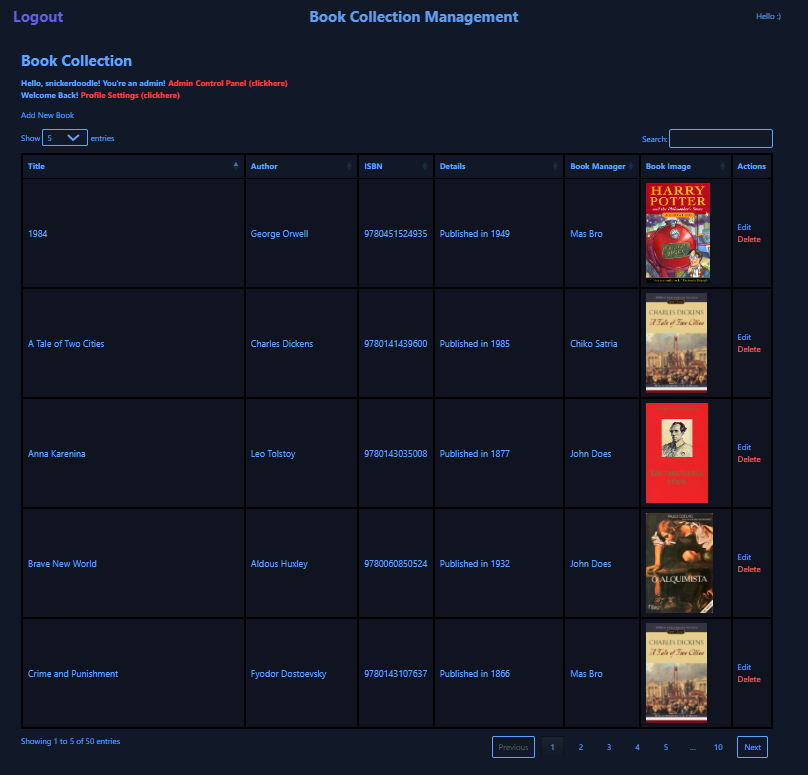
# Tampilan index/mainpage(notlogin)



## Tampilan index/mainpage (login user)



## (login admin)



## Index.blade.php Code

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Book Collection</title>

    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.13.6/css/jquery.dataTables.min.css">

    <style>

        div.dataTables\_wrapper div.dataTables\_filter input {

            background-color: #111827;

            color: #60a5fa;

            border: 1px solid #60a5fa;

            border-radius: 5px;

            padding: 5px;

            margin-bottom: 10px;

        }

        .dataTables\_wrapper .dataTables\_length label {

            color: #60a5fa;

        }

        .dataTables\_wrapper .dataTables\_length select {

            background-color: #111827;

            color: #60a5fa;

            border: 1px solid #60a5fa;

            border-radius: 4px;

            padding: 4px;

            outline: none;

        }

        #myTable\_paginate {

            margin-top: 10px;

        }

        #myTable\_previous,

        #myTable\_next {

            background-color: #111827;

            color: #60a5fa;

            border: 1px solid #60a5fa;

            border-radius: 4px;

            padding: 8px 12px;

            margin-right: 5px;

            cursor: pointer;

        }

        #myTable\_previous:hover,

        #myTable\_next:hover {

            background-color: #333;

        }

        #myTable\_paginate .paginate\_button:not(:last-child),

        #myTable\_paginate .paginate\_button:last-child {

            margin-right: 10px;

        }

        #myTable {

            border: 1px solid black;

        }

        #myTable tbody {

            background-color: rgba(14, 15, 25, 0.433);

        }

        \*::-webkit-scrollbar {

            display: none;

        }

    </style>

</head>

<body class="bg-gray-900 text-blue-400">

    <header class="px-5 sticky top-0 bg-gray-900 py-5 opacity-95 backdrop-blur-sm z-10">

        <nav class="container mx-auto flex items-center justify-between opacity-100">

            @if (Auth::check())

            <div class="text-3xl font-bold mb-4">

                <form id="logout-form" action="{{ route('logout') }}" method="POST">

                    @csrf

                    <button type="submit" class="font-bold text-indigo-500 hover:underline">Logout</button>

                </form>

            @else

            <a href="{{ url('/login') }}" class="text-3xl text-blue-600 font-medium mb-4">Login</a>

            @endif

            </div>

            <div class="text-center text-blue-400 justify-center items-center space-x-4 hidden sm:flex flex-grow">

                <a href="{{ url('/landing') }}"><h1 class="text-3xl font-bold mb-4">Book Collection Management</a>

            </div>

                <div class="flex space-x-4 inline mb-4">

                @if (Auth::check())

                   Hello :)

                @else

                <a href="{{ route('register') }}" class="text-3xl text-blue-600 font-medium mb-4">Register</a>

                @endif

            <div class="sm:hidden relative">

                <button id="menu-toggle" class="text-blue-400 hover:text-white transition-transform transform">

                    <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 transition-transform transform" fill="none" viewBox="0 0 24 24"

                        stroke="currentColor">

                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M4 6h16M4 12h16M4 18h16"></path>

                    </svg>

                </button>

                <div id="menu" class="hidden absolute right-0 mt-2 py-2 bg-gray-700 rounded-lg shadow-lg">

                    <a href="#home" class="block px-4 py-2 text-blue-200 hover:text-white">Home</a>

                    <a href="#about" class="block px-4 py-2 text-blue-200 hover:text-white">About</a>

                    <a href="#portfolio" class="block px-4 py-2 text-blue-200 hover:text-white">Portfolio</a>

                    <a href="#contact" class="block px-4 py-2 text-blue-200 hover:text-white">Contact</a>

                </div>

            </div>

        </nav>

    </header>

    <div class="container mx-auto p-4">

        <div class="inline">

            <h1 class="text-3xl font-bold mb-4 text-blue-400">Book Collection</h1>

            <div class="flex space-x-4 inline mb-4">

            @if (Auth::check())

                <span class="font-bold text-blue-400">Hello, {{ Auth::user()->name }}!

                @if (Auth::user()->admin == 1)

                    You're an admin! <a href="{{ url('/admin') }}" class="inline text-red-500 ">Admin Control Panel (clickhere) </a>

                @elseif (Auth::user()->admin == 2)

                    You're a super admin! <a href="{{ url('/admin') }}" class="inline text-red-500 ">Admin Control Panel (clickhere) </a>

                @endif  <br> Welcome Back!  <a href="{{ url('/profile') }}" class="inline text-red-500 ">Profile Settings (clickhere) </a>

            @endif

        </div>

    </div>

        @if (Auth::check())

        <div class="flex justify-start mb-4">

            <a href="{{ url('/books/create') }}" class="text-blue-400 inline-block">Add New Book</a>

        </div>

        @else

        <div class="flex justify-start mb-4">

            <a href="{{ route('login') }}" class="text-blue-400 inline-block">Login to Add New Book</a>

        </div>

        @endif

        <table id="myTable" class="w-full border-collapse border my-4 text-blue-400">

            <thead>

                <tr class="bg-gray-900">

                    <th class="border border-black py-2 px-3">Title</th>

                    <th class="border border-black py-2 px-3">Author</th>

                    <th class="border border-black py-2 px-3">ISBN</th>

                    <th class="border border-black py-2 px-3">Details</th>

                    <th class="border border-black py-2 px-3">Book Manager</th>

                    <th class="border border-black py-2 px-3">Book Image</th>

                    <th class="border border-black py-2 px-3">Actions</th>

                </tr>

            </thead>

            <tbody>

                @forelse ($books as $book)

                <tr class="bg-gray-800">

                    <td class="border border-black py-3 px-4 text-lg">{{ $book->title }}</td>

                    <td class="border border-black py-3 px-4 text-lg">{{ $book->author }}</td>

                    <td class="border border-black py-3 px-4 text-lg">{{ $book->isbn }}</td>

                    <td class="border border-black py-3 px-4 text-lg">{{ $book->details }}</td>

                    <td class="border border-black py-3 px-4 text-lg">{{ $book->user->name }}</td>

                    <td class="border border-black py-3 px-4 w-10 h-10">

                        <img src="{{ asset('images/' . $book->image) }}" alt="Book Image" style="max-width: 200px; max-height: 200px;">

                    </td>

                    <td class="border border-black py-3 px-4">

                        @if (Auth::check())

                            @if (Auth::user()->admin == 1 || Auth::user()->id == $book->user\_id || Auth::user()->admin == 2)

                            <a href="{{ url('/books/' . $book->id . '/edit') }}" class="text-blue-400">Edit</a>

                            <form method="post" action="{{ url('/books/' . $book->id) }}" onsubmit="return confirm('Are you sure you want to delete this book?');">

                                @csrf

                                @method('delete')

                                <button type="submit" class="text-red-400">Delete</button>

                            </form>

                            @else

                            <span class="text-red-500">You're not the manager of the books! Cannot edit nor delete!</span>

                            @endif

                        @else

                        <span class="text-red-500">You're not logged in! Cannot edit nor delete!</span>

                        @endif

                    </td>

                </tr>

                @empty

                <tr class="bg-gray-800">

                    <td colspan="6" class="border border-black py-3 px-4 text-lg">No books available</td>

                </tr>

                @endforelse

            </tbody>

        </table>

    </div>

    <script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-/JqT3SQfawRcv/BIHPThkBvs0OEvtFFmqPF/lYI/Cxo="

        crossorigin="anonymous"></script>

    <script src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.min.js"></script>

    <script>

        $(document).ready(function () {

            let table = $('#myTable').DataTable({

                "lengthMenu": [5, 10, 15, 20, 25, 50, 75, 100],

                "ordering": true,

                "searching": true,

                "info": true,

                "paging": true,

                "responsive": true,

                "columnDefs": [

                    { "orderable": false, "targets": [6] }

                ],

            });

        });

    </script>

</body>

</html>

# Layout Admin

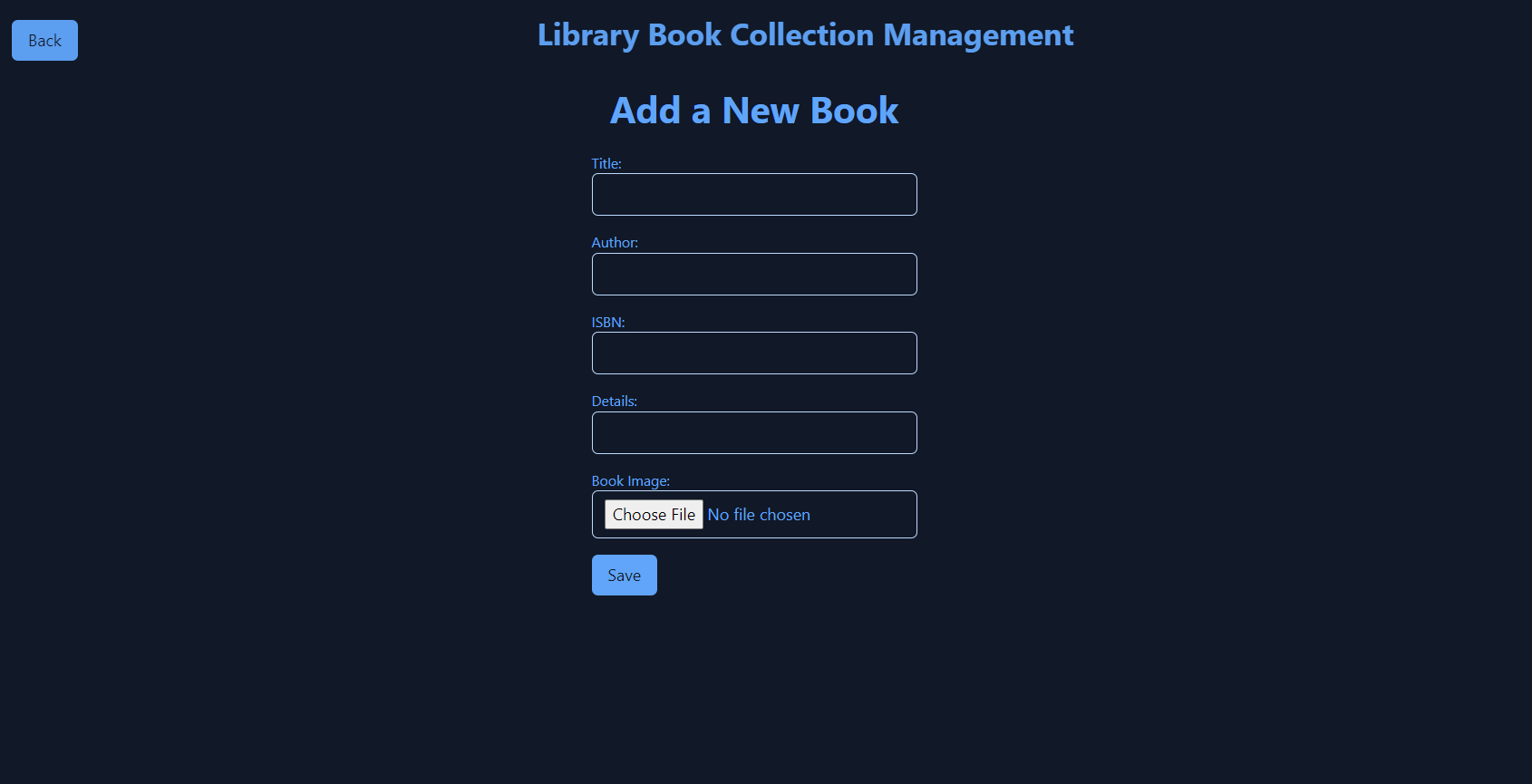


## Layout admin (SuperAdmin)



# CRUD Implementation

## Create:



Code: <div class="container mx-auto flex items-center justify-center ">

        <form method="post" action="{{ url('/books') }}" class="max-w-lg bg-gray-900 p-6 rounded-md" enctype="multipart/form-data">

            @csrf

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Title:</label>

                <input type="text" name="title"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Author:</label>

                <input type="text" name="author"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">ISBN:</label>

                <input type="text" name="isbn"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Details:</label>

                <input type="text" name="details"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Book Image:</label>

                <input type="file" name="image"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200">

            </div>

            @auth

           <input type="hidden" name="user\_id" value="{{ auth()->user()->id }}">

            @endauth

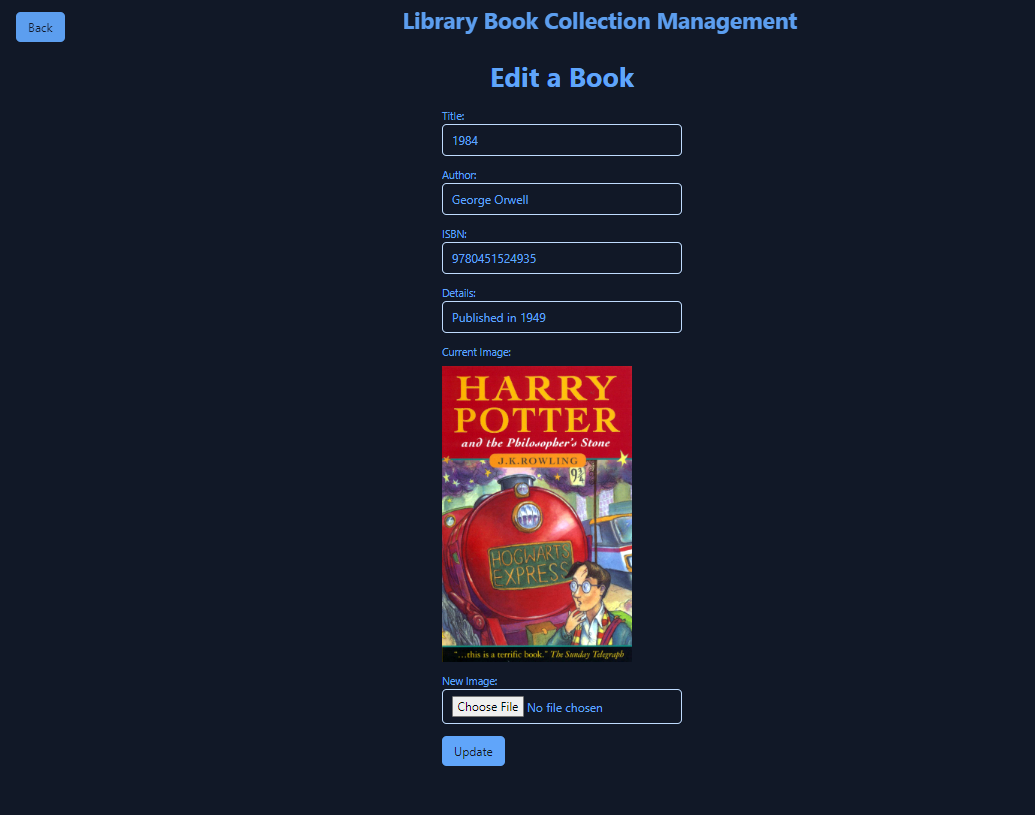
            <button type="submit"

                class="bg-blue-400 text-gray-900 px-4 py-2 rounded-md hover:bg-blue-500 transition duration-300">Save</button>

        </form>

    </div>

## Update:



Code: <div class="container mx-auto flex items-center justify-center">

        <h1 class="text-4xl font-bold text-blue-400">Edit a Book</h1>

    </div>

    <div class="container mx-auto flex items-center justify-center ">

        <form method="post" action="{{ url('/books/' . $book->id) }}"

            class="max-w-md bg-gray-900 p-6 rounded-md" enctype="multipart/form-data">

            @csrf

            @method('put')

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Title:</label>

                <input type="text" name="title" value="{{ $book->title }}"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Author:</label>

                <input type="text" name="author" value="{{ $book->author }}"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">ISBN:</label>

                <input type="text" name="isbn" value="{{ $book->isbn }}"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Details:</label>

                <input type="text" name="details" value="{{ $book->details }}"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200"

                    required>

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">Current Image:</label>

                @if ($book->image)

                <img src="{{ asset('images/' . $book->image) }}" alt="Current Book Image"

                    class="mt-2 max-w-xs">

                @else

                <p class="text-blue-400">No image available</p>

                @endif

            </div>

            <div class="mb-4">

                <label class="block text-sm text-blue-400">New Image:</label>

                <input type="file" name="image"

                    class="w-full px-3 py-2 border rounded-md text-blue-400 bg-gray-900 focus:outline-none focus:border-blue-400 border-blue-200">

            </div>

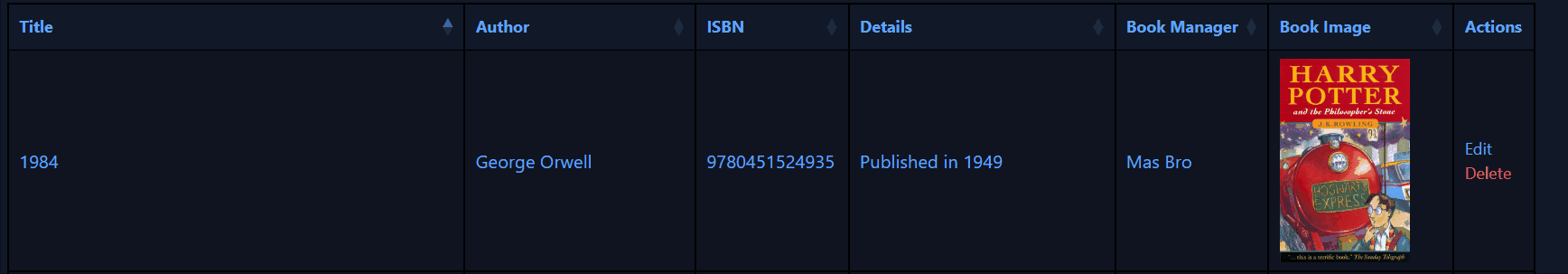
            <button type="submit"

                class="bg-blue-400 text-gray-900 px-4 py-2 rounded-md hover:bg-blue-500 transition duration-300">Update</button>

        </form>

    </div>

Delete: https://chiknwy.github.io/bruh/assets/delelte.gif



Routes:  
Route::get('/', [HomeController::class, 'landingpage']);

Route::get('/landing', [LandingController::class, 'index']);

Route::get('/about', [HomeController::class, 'about']);

Route::get('/marketplace', [HomeController::class, 'marketplace']);

Route::get('/features', [HomeController::class, 'features']);

Route::get('/lib', [HomeController::class, 'lib']);

Route::post('/lib', [BookController::class, 'store']);

Route::get('/books', [BookController::class, 'index']);

Route::middleware(['auth'])->group(function () {

    Route::get('/books/create', [BookController::class, 'create']);

    Route::post('/books', [BookController::class, 'store']);

    Route::get('/books/{id}/edit', [BookController::class, 'edit']);

    Route::put('/books/{id}', [BookController::class, 'update']);

    Route::delete('/books/{id}', [BookController::class, 'destroy']);

});

Route::get('/admin', [BookController::class, 'admin']);

Route::get('/admin/create', [BookController::class, 'createUser']);

Route::post('/admin/store', [BookController::class, 'storeUser']);

Route::get('/admin/edit/{id}', [BookController::class, 'editUser']);

Route::put('/admin/update/{id}', [BookController::class, 'updateUser']);

Route::delete('/admin/delete/{id}', [BookController::class, 'deleteUser']);

Route::get('/profile', [BookController::class, 'profile']);

Route::get('/uts', [BookController::class, 'uts']);

Auth::routes();

Route::post('/logout', [LoginController::class, 'logout'])->name('logout');

Route::get('/home', [LandingController::class, 'index']);

# BookController (also code for the delete):

class  BookController extends Controller {

    public function index()

    {

        $books = Book::when(request('filter'), function ($query) {

            $query->where('title', 'like', '%' . request('filter') . '%');

        })->paginate(100);

        return view('books.index', compact('books'));

    }

    public function create() {

        return view('books.create');

    }

    public function store(Request $request)

{

    $request->validate([

        'title' => 'required|max:255',

        'author' => 'required|max:255',

        'isbn' => 'required|numeric',

        'details' => 'required',

        'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',

    ]);

    // Set the user\_id before saving the book

    $book = new Book($request->all());

    $book->user\_id = auth()->user()->id;

    $imageName = time().'.'.$request->image->extension();

    $request->image->move(public\_path('images'), $imageName);

    // Store the book data including the image name

    $book->image = $imageName;

    $book->save();

    return redirect('/books')

        ->with('success','You have successfully uploaded the book.');

}

    public function show($id) {

        $book = Book::find($id);

        return view('books.show', compact('book'));

    }

    public function uts(){

        return view('uts.uts');

    }

    public function edit($id) {

        $book = Book::find($id);

        return view('books.edit', compact('book'));

    }

    public function update(Request $request, $id)

    {

        $request->validate([

            'title' => 'required|max:255',

            'author' => 'required|max:255',

            'isbn' => 'required|numeric',

            'details' => 'required',

            'image' => 'image|mimes:jpeg,png,jpg,gif,svg|max:2048',

        ]);

        $book = Book::find($id);

        // Update other fields

        $book->title = $request->title;

        $book->author = $request->author;

        $book->isbn = $request->isbn;

        $book->details = $request->details;

        // Check if a new image is provided

        if ($request->hasFile('image')) {

            // Delete the existing image if it exists

            if ($book->image) {

                Storage::delete('public/images/' . $book->image);

            }

            // Move the new image to the storage

            $imageName = time().'.'.$request->image->extension();

            $request->image->move(public\_path('images'), $imageName);

            // Update the book with the new image name

            $book->image = $imageName;

        }

        $book->save();

        return redirect('/books');

    }

    public function destroy($id) {

        $book = Book::find($id);

        // Delete the associated image from local storage

        $imagePath = public\_path($book->image); // Assuming image\_path is the attribute storing the image path

        if (file\_exists($imagePath)) {

            unlink($imagePath);

        }

        // Delete the book record

        $book->delete();

        return redirect('/books');

    }

    public function admin() {

        $admin = auth()->user();

        $users = User::all();

        if (auth()->check() && $admin->admin == true ) {

            return view('books.admin', compact('users'));

        } else {

            return redirect('/books');

        }

    }

    public function createUser()

{

    return view('admin.create');

}

public function storeUser(Request $request)

{

    $request->validate([

        'name' => 'required|max:255',

        'email' => 'required|email|unique:users',

        'password' => 'required|min:6',

        'admin' => '', // You might need to adjust this validation based on your admin field type

    ]);

    $user = new User($request->all());

    $user->password = Hash::make($request->password);

    $user->save();

    return redirect('/admin')->with('success', 'User created successfully.');

}

public function editUser($id)

{

    $admin = auth()->user();

    $user = User::find($id);

    if (auth()->check() && $admin->admin == true ) {

    return view('admin.edit', compact('user'));

    }else{

        return redirect('/books');

    }

}

public function updateUser(Request $request, $id)

{

    $request->validate([

        'name' => 'required|max:255',

        'email' => 'required|email|unique:users,email,' . $id,

        'password' => 'nullable|min:6',

        'admin' => 'numeric',

    ]);

    $user = User::find($id);

    $user->fill($request->except('password')); // Exclude password from fill

    if ($request->filled('password')) {

        $user->password = Hash::make($request->password);

    } else {

        $user->password = $user->getOriginal('password');

    }

    $user->save();

    return redirect('/admin')->with('success', 'User updated successfully.');

}

public function deleteUser($id)

{

    User::find($id)->delete();

    return redirect('/admin')->with('success', 'User deleted successfully.');

}

public function profile()

{

    $users = auth()->user();

    return view('books.profile', compact('users'));

}

}

Pagination and filtering:   
<https://chiknwy.github.io/bruh/assets/Pagination.gif>

https://chiknwy.github.io/bruh/assets/search.gif



I’d use datables for the pagination and filtering

# Relationship database concept

## Model Book.php

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Book extends Model {

    protected $fillable = ['title', 'author', 'details', 'isbn', 'image'];

    public function user()

    {

        return $this->belongsTo(User::class);

    }

}

## Migration

public function up()

    {

        Schema::create('books', function (Blueprint $table) {

            $table->id();

            $table->string('title');

            $table->string('author');

            $table->string('isbn');

            $table->text('details');

            $table->string('image')->nullable();

            $table->foreignId('user\_id')->constrained()->nullable()->default(null);

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('books');

    }