

Assignment -03 [BookStock]

Task . 100 marks (Pass Marks: 40)

The objective is to build a web-based application to manage a library of books. The system must allow users to perform **CRUD** (Create, Read, Update, Delete) operations while handling data persistence via MySQL and managing assets (like book covers) through Laravel's file system.

1. Database & Schema (MySQL & Migrations)

You must create three migrations to establish a relational structure. Ensure the tables are created in an order that respects foreign key constraints (Categories/Authors first, then Books).

- **categories table:** id, name.
 - **authors table:** id, name, bio (text).
 - **books table:** * id, title, isbn (unique).
 - category_id (foreign key pointing to categories).
 - author_id (foreign key pointing to authors).
 - cover_image (string for file path).
 - description (text), published_at (date).
-

2. Routing & Controllers

- **Resourceful Routing:** Set up `Route::resource('books', BookController::class);`.
- **Logic Isolation:** The BookController must handle the "Heavy Lifting."

Create/Edit Methods: You must fetch all categories and authors from the database using Query Builder to populate the dropdown menus in your forms.

3. Data Handling (Query Builder)

You are strictly forbidden from using Eloquent Models (e.g., `Book::all()`). Use the DB facade for all operations.

- **Joins for Index:** When listing books, use a `leftJoin` or `join` to pull the Author and Category names so the user sees "Fiction" instead of "ID: 1".
-

4. Frontend (Laravel Blade)

- **Master Layout:** Use `layout.blade.php` to include your CSS (Bootstrap or Tailwind) and the `@yield('content')` directive.
 - **Dynamic Dropdowns:** In `create.blade.php` and `edit.blade.php`, use `@foreach` loops to generate `<option>` tags for Categories and Authors.
 - **Active State:** In the Edit form, ensure the correct Category and Author are "selected" using a ternary operator.
-

5. Forms & Validation

- **Security:** Every form must include the `@csrf` directive.
 - **Validation Logic:**
 - `title`, `author_id`, and `category_id` are **required**.
 - `isbn` must be **unique** in the `books` table (except when updating the same record).
 - `cover_image` must be an **image** (jpeg, png, jpg) and under **2MB**.
 - **Error Feedback:** Use `@error('field_name')` to display red validation messages under each input.
-

6. File Handling

- **Storage Path:** Save uploaded covers to `storage/app/public/covers`.
- **Naming:** Store the path in the database.
- **Public Access:** You must run `php artisan storage:link` to make the files accessible.
- **Display:** In your Blade table, render the image using the `asset()` helper: `src="{{ asset('storage/' . $book->cover_image) }}"`

নির্দেশনা:

১। ভালো করে অ্যাসাইনমেন্টের নির্দেশনা গুলো পড়ুন।

২। এই এসাইনমেন্টে আপনাকে UI প্রদান করা হবে।

৩। UI এর Resource URL:

resource link:

<https://drive.google.com/file/d/1sFaRPOlwShta11X2jldkgvH-TXFFW8Qc/view?usp=sharing>

৪। এরপরেও যদি অ্যাসাইনমেন্ট সম্পর্কিত কোন প্রশ্ন থাকে তাহলে অবশ্যই সাপোর্টে এসে কথা বলবেন

৫। অ্যাসাইনমেন্টের মার্কস ১০০ (পাস মার্কস ৪০)

৬। এসাইনমেন্ট জমা দেওয়ার শেষ তারিখ (28 Feb 2026) রাত 11.59 মিনিট পর্যন্ত।)

৭। এসাইনমেন্ট জমা দেওয়ার শেষ তারিখ অতিক্রম হবার পর জমা দিলে ৫০% মার্কস কর্তন হবে

৮। অ্যাসাইনমেন্ট এর সমস্ত সোর্স কোড গিটহাব এ আপলোড করতে হবে এবং readme.md file এ নিচের তথ্যগুলো প্রদান করতে হবে

প্রজেক্ট এর root ডিরেক্টরীতে একটি readme.md file তৈরি করুন

তারপর রেডমি ফাইলে নিচের দেওয়া তথ্য গুলো লিখুন

Assignment : কত নাম্বার এসাইনমেন্ট তা লিখুন

Name : আপনার নাম লিখুন

Email: (আপনার জিমেইল এড্রেস লিখুন যে যে জিমেইল দিয়ে আপনি কোর্স purchase করেছেন)

পুন:রায় গিটহাব এ readme ফাইল সহ কোড গুলো পুশ করুন

৯। এসাইনমেন্টের গিটহাব রিপোজিটরিটির লিংক কপি করুন এবং সেটি ওয়েবসাইটে জমা দিন (গিটহাব রিপোজিটরিটি অবশ্যই পাবলিক থাকতে হবে)