Laravel Project Assignment: Building a Comprehensive Management System

Objective

Students will create a management system to learn the fundamentals of Laravel, including routing, controllers, models, form request, JSON Resource (if SPA), views, migrations, seeder, command and database relationships (with proper integrity and indexing). They will progress to more advanced concepts like Eloquent ORM, middleware, and CRUD operations for complex data models.

Project Title: Library Management System

Overview:

Create a library management system where users can manage books, authors, publishers, and borrowers. The system should include user roles (admin and user) with different permissions.

Assignment Deliverables:

1. **Functional Features**
   - **Admin Panel:** Manage books, authors, publishers, borrowers, and borrowing transactions.
   - **User Panel:** View available books, borrow books, and view borrowing history.
   - **Authentication:** Use Laravel Breeze or Laravel Jetstream for login, registration, and user roles.
2. **Database Relationships:**
   Design the database to include the following relationships:
   - **One-to-Many**: One publisher can publish many books.
   - **Many-to-Many**: Books can have multiple authors, and authors can write multiple books.
   - **One-to-One**: Each user profile has one corresponding profile detail (e.g., address, contact info).
   - **Polymorphic Relationship**: Books, publishers. and authors can have multiple reviews (Users can leave reviews on books, publishers, and authors).
3. **CRUD Operations:**
   Implement complete CRUD operations for the following:
   - Books
   - Authors
   - Publishers
   - Borrowers

     Ensure validation is implemented for all forms.
4. **Advanced Features:**
   - **Middleware:**
     - Restrict admin features to admin users only.
     - Allow book borrowing only for logged-in users.
   - **Search and Filter:**
     - Search books by title, author, or publisher.
     - Filter books by availability, **ISBN** Number, or genre.

    - **Pagination:** Add pagination to the book listing.

5. **Front-end Integration:**
   Use **Bootstrap** or **Tailwind CSS** to create responsive and user-friendly layouts for the following:
    - Home page listing available books.
    - Admin dashboard.
    - User profile page.
6. **Tasks for Students**
   Students will implement the following to understand Eloquent relationships:
    - Set up migrations for all models.
    - Define relationships in models (e.g., hasMany, belongsTo, belongsToMany, morphMany).
    - Seed the database with dummy data using factories and seeders.
    - Create Resource/RESTful (if SPA) controllers and routes.
7. **Testing:**
   Write basic unit tests and feature tests for:
    - User registration and login.
    - CRUD operations for books.
    - Borrowing and returning books.

Grading Criteria:

Criteria	Weight
Database design & relationships	30%
Functionality (CRUD, roles)	30%
Front-end design & usability	20%
Advanced features & testing	10%
Code structure & documentation	10%

Submission Guidelines:

- Upload your code to a GitHub repository and share the link.
- Include a README.md file with:
    - Project description
    - Installation steps
    - Brief usage guide
- Include screenshots of your application in the README.md.

Bonus Challenges (Optional):

- Add real-time notifications using **Pusher** or **Laravel Echo** for borrowing approvals.
- Implement an API endpoint to list books with their authors and publishers in JSON format.
- Integrate a simple Vue.js or React front-end for book borrowing.