Project: Book Rental System

Objective:

Create a Book Rental System where users can register, manage their books, and rent books from other users. This project will involve database relationships, file uploads using Multer, and rendering views with EJS.

Requirements:

1. User Authentication:

- **Register:** Users can register by providing a username, email, and password.
- **Login:** Registered users can log in using their email and password.
- **Forgot Password:** Implement a feature to allow users to reset their passwords.

2. User Management:

- **Dashboard:** After logging in, users should have access to a dashboard where they can manage their books.
- **Add Book:** Users can add new books to their collection with details like title, author, description, thumbnail image, and a gallery of images.
- **Edit/Delete Book:** Users can edit or delete their books.
- **View Available Books:** Users can view all books available for rent, excluding their own.

3. Book Rental:

- **Rent a Book:** Users can rent a book by selecting it and entering the renting period (start date and end date).
- **Validation:** Ensure that the selected book is not already rented for the specified period.

- **Rental History:** Users can view their rental history, including the books they've rented and the renting periods.
- **Rate and Review:** After renting a book, users can leave a rating and review for the book. Other users can view these ratings and reviews when browsing available books.

4. File Uploads with Multer:

- **Thumbnail:** Allow users to upload a thumbnail image for their books.
- **Gallery:** Allow users to upload multiple images to a gallery for each book.
- **File Handling:** Use Multer to handle file uploads, storing images in a `public/uploads/` directory.

5. Database Design:

- **Relationships:**
  - A **User** has many **Books**.
  - A **Book** belongs to a **User**.
  - A **Book** has many **BookImages**.
  - A **BookImage** belongs to a **Book**.
  - A **Rental** belongs to both **User** and **Book**.
  - A **User** can rate and review many **Books**, and a **Book** can have many ratings and reviews from different **Users** (many-to-many relationship).

6. Views using EJS:

- **Register/Login/Forgot Password:** EJS templates for user authentication.
- **Dashboard:** Display user-specific data like their books and rental history.
- **Add/Edit Book:** Forms for adding and editing books with file upload options.
- **Available Books:** A view to display all available books with their details, an option to rent, and the ability to see ratings and reviews.
- **Rental History:** Display the user's past and current rentals.

- **Ratings and Reviews:** A form to submit ratings and reviews and a section to display them.

7. Routing:

- **User Routes:** For authentication (register, login, forgot password) and dashboard management.
- **Book Routes:** For CRUD operations on books.
- **Rental Routes:** For handling the renting process and viewing rental history.
- **Rating and Review Routes:** For handling the submission and display of ratings and reviews.

8. Validation:

- **Form Validation:** Ensure all required fields are filled out and valid.
- **Image Validation:** Validate the type and size of uploaded images.
- **Rental Validation:** Ensure that a book is not double-booked for the same period.
- **Rating Validation:** Ensure that only users who have rented a book can leave a rating or review.