

Library Management System with Borrowing Feature

You have been hired by a small-town public library to develop a Library Management System. The goal is to create a digital platform where the library staff can manage the book inventory and users (clients) can borrow and return books online. To ensure secure access, both library staff (admins) and regular users (clients) must be able to register and log in through a custom login and signup page — usage of Django's default admin panel is strictly prohibited for this functionality. User authentication should be properly set up, with restrictions on password strength (e.g., minimum 8 characters, must include letters and numbers) and unique usernames.

Project Requirements:

You are required to build a Django project named `library_project` and an app named `library`.

The system must handle book management, user authentication, and book borrowing/returning operations based on the following specifications:

1. User Management:

- Implement a custom user registration (signup) page where users can create an account by providing a username, email, and password.
- Implement a login page where registered users (both admins and clients) can securely log in.
- Add authentication and validation:
 - Passwords must be at least 8 characters long and contain both letters and numbers.
 - Username must be unique.
- Upon successful registration/login, users should be redirected to their respective dashboards.
- Use role-based access control:
 - Admin users should be able to manage all books and view all borrow/return activities.
 - Client users should be able to view available books, borrow books, and return books they have borrowed.

2. Define the following two models:

- **Book Model:**
 - title (CharField)
 - author (CharField)
 - publication_date (DateField)
 - ISBN (CharField)
 - genre (CharField)
 - is_available (BooleanField, default=True)
- **Borrower Model:**
 - name (CharField)
 - email (EmailField)
 - borrowed_book (ForeignKey to Book, with on_delete=SET_NULL, null=True)

- borrow_date (DateField, auto_now_add=True)
- return_date (DateField, nullable=True)

3. Business Logic Rules:

- When a user borrows a book, the book's is_available status must automatically become False.
- When a user returns a book, the return_date must be recorded, and the book's is_available status must be updated back to True.
- Only books that are currently available (is_available=True) should be displayed in the list of borrowable books.

4. Functional Requirements:

- **Book Management (for Admin users):**
 - Add new books to the system.
 - Edit existing book details.
 - View a list of all books.
 - Delete books.
- **Borrowing Books (for Client users):**
 - View a list of available books.
 - Borrow a book (only if it is available).
 - After borrowing, the book should not appear in the available list.
- **Returning Books (for Client users):**
 - View the list of books they have borrowed.
 - Return a borrowed book, which updates the return date and makes the book available again.

5. Access Restrictions:

- Only authenticated users should be able to access book borrowing and returning pages.
- Unauthenticated users trying to access protected pages should be redirected to the login page.
- Admin-only sections (like book management) must be restricted to users with admin roles.