

# Open Library Django Project ([View Live](#))

---

## Overview

Open Library is a Django demonstration project showcasing how Django can be used to build a simple library system. It extends the default Django user model to classify users as general users, staff members, and administrators, each with varying levels of access to features within the system.

## Features

### Authentication and Authorization

- System uses Django's built-in authentication system.
- Permissions are set based on user type to restrict access to certain views and functionalities.
- Anyone can register a new account.
- User can login using the username or email.
- Reset password through email.

### User Types

#### 1. General Users

- Can browse books in the library.
- Can search for books by title, author, or genre.
- Can filter books by name (default), rating, publication, and upload date.
- Can view the details of each book.
- Can save books as favorites.
- Can put a review on a book.
- Can update and delete (own) account.

#### 2. Staff Members

- All features are available to general users.
- Can add new books to the library.
- Can edit existing book information.
- Can delete a book.

#### 3. Administrators

- All features are available to staff members.
- Can manage user accounts.
- Can assign user roles (general user, staff, admin).
- Have control over all reviews.

### Book Management

- Books are stored with information such as title, author, genre, cover image, publication date, and description.
- CRUD operations for books (Create, Read, Update, Delete) are available to authorized users.

# Project Structure

## Models

- **Book:** Represents a book in the system with attributes like title, author, genre, etc.
- **FavoriteBook:** Represents a favorite book saved by a user.
- **Review:** Represents a review of a book by a user.

## Views

- **Catalogue Views:** Handles views related to browsing, searching, adding, editing, and deleting books.
- **User Management Views:** Handles views related to user registration, profile management, and user list/detail.

## URLs

- URLs are configured to route requests to appropriate views for catalog browsing, book management, user management, etc.

## Installation and Usage

1. Clone this repository.
2. Create a virtual environment and install dependencies using

```
pip install -r requirements.txt
```

3. Apply migrations using

```
python manage.py migrate
```

4. Create a superuser using

```
python manage.py createsuperuser
```

5. Run the project on the server. In the case of local development servers,

```
python manage.py runserver
```

## License

This project is licensed under the [MIT License](#).

