

DJANGO FINAL PROJECTS

1. Online Book Review System

Use Case: A platform where users can browse books, leave reviews, and rate them.

Requirements:

- **Introduction & Project Structure:**
Create a project bookreview and an app reviews.
- **Models & ORM:**
 - Book model (title, author, description, cover_image).
 - Review model (ForeignKey to Book, user, rating, comment).
- **Admin Panel:**
Register models, add list filters (by rating, author).
- **Forms:**
 - Manual Form for search by book name.
 - ModelForm for adding reviews.
- **CRUD:**
 - Users can add/update/delete their reviews.
 - `get_object_or_404()` to fetch book/review.
- **Authentication:**
 - Users must login to post reviews.
 - Staff can delete any review.
 - Flash messages for success/error.
- **Advanced Models:**
 - One-to-Many (Book → Reviews).
 - Many-to-Many (Book categories).
- **File Uploads:**
Book cover upload.

- **CBVs:**
Use ListView for books, DetailView for reviews, CreateView for adding reviews.
 - **Pagination & Filtering:**
Paginate book list, search books with Q(title__icontains).
 - **Permissions:**
Users can only edit/delete their own reviews.
 - **Signals:**
Create a user profile automatically after registration.
 - **Email Integration:**
Send email notification when a new review is added.
-

2. Student Course Management System

Use Case: A portal for students to register for courses, upload assignments, and track grades.

Requirements:

- **Models:**
 - Course (title, instructor, description).
 - StudentProfile (user, roll_number).
 - Enrollment (student ↔ course ManyToMany with extra fields).
- **Admin:**
Search by roll number, filter by course.
- **Forms:**
 - Manual form for course enrollment.
 - ModelForm for assignment uploads.
- **CRUD:**
Students can enroll/drop courses.
- **Authentication:**
 - Login/Logout/Register.
 - Different views for staff (instructors) vs students.
- **Advanced Models:**
 - Many-to-Many (Students ↔ Courses).
 - ForeignKey (Course ↔ Instructor).

- **File Uploads:**
Students upload assignments (PDF).
 - **CBVs:**
 - ListView → courses.
 - DetailView → course details.
 - CreateView → upload assignment.
 - **Pagination & Filtering:**
Show only 5 courses per page. Filter by department.
 - **Permissions:**
Only instructors can grade assignments.
 - **Signals:**
Automatically create StudentProfile after user signup.
 - **Email Integration:**
Email student when an assignment is graded.
-

3. Job Portal with Resume Upload

Use Case: Job seekers can apply to jobs, and employers can post jobs.

Requirements:

- **Models:**
 - Job (title, company, description).
 - Application (ForeignKey → Job, applicant, resume file, status).
- **Admin:**
Employers can manage jobs and applications.
- **Forms:**
 - Manual form for job search.
 - ModelForm for applying to jobs.
- **CRUD:**
Employers → Post/Update/Delete jobs.
Applicants → Apply, withdraw applications.
- **Authentication:**
 - Employers vs Applicants (different dashboards).

- **Advanced Models:**
One-to-Many (Job → Applications).
 - **File Uploads:**
Applicants upload resumes (PDF/DOC).
 - **CBVs:**
 - CreateView → Job posting.
 - ListView → Jobs.
 - DetailView → Job details.
 - **Pagination & Filtering:**
Paginate job list, filter by location or keyword.
 - **Permissions:**
Only job poster can edit/delete the job.
 - **Signals:**
Auto-create employer profile on signup.
 - **Email Integration:**
Send confirmation email when a user applies for a job.
-

4. Event Management System

Use Case: Users can create, register, and manage events.

Requirements:

- **Models:**
 - Event (title, date, organizer, image).
 - Registration (ForeignKey → Event, user, status).
- **Admin:**
Filter events by date.
- **Forms:**
 - Manual form for searching events.
 - ModelForm for event creation.
- **CRUD:**
Users → Create, update, delete events.
Others → Register/unregister for events.

- **Authentication:**
Login required for event registration.
 - **Advanced Models:**
Many-to-Many (Users ↔ Events through Registration).
 - **File Uploads:**
Upload event posters.
 - **CBVs:**
Use ListView, DetailView, UpdateView.
 - **Pagination & Filtering:**
Show 10 events per page, filter upcoming vs past events.
 - **Permissions:**
Only event organizer can edit/delete the event.
 - **Signals:**
Send notification when a new registration is made.
 - **Email Integration:**
Email reminder to registered users before event day.
-

5. Blogging Platform with Comments

Use Case: A blog website where users can post blogs and comment.

Requirements:

- **Models:**
 - Post (title, content, author, image).
 - Comment (ForeignKey → Post, user, text).
- **Admin:**
Filters for posts by author.
- **Forms:**
 - Manual form for blog search.
 - ModelForm for comment submission.
- **CRUD:**
 - Author → Create, update, delete posts.
 - Users → Add, edit, delete their comments.
- **Authentication:**

- Login required for posting/commenting.
- **Advanced Models:**
 - One-to-Many (Post → Comments).
 - Many-to-Many (Post categories/tags).
- **File Uploads:**

Blog image uploads.
- **CBVs:**
 - ListView → Blogs.
 - DetailView → Blog with comments.
 - CreateView → Add post.
- **Pagination & Filtering:**

Paginate blogs, filter by category.
- **Permissions:**

Only author can edit/delete their posts.
- **Signals:**

Auto-create profile for new user.
- **Email Integration:**

Send email notification to subscribers when a new blog is posted.