

Movie Rental System - Project Contribution

Contributor: Veekshitha Beladara Nanjegowda

Role: Backend Development

Responsibilities

- Designed and implemented Django models (`Genre`, `Movie`, `RentedMovie`).
- Managed user authentication and session handling (login, logout, register).
- Built the logic for renting and returning movies using Django views and forms.
- Developed a custom Django management command `load_movies` for importing movie data from a JSON file.
- Configured Django admin interface for managing genres and movies.

Documentation

The Django framework was used to create the extensive web application that lets users search, find, filter, and rent movies. Users can manage the price, view comprehensive information, and identify films using the system. This offers a thorough analysis of the documentation system's functionality, design, and implementation.

The Movie Rental System is built using the following technologies:

movie_rental: Main project settings

movies: Primary application for handling movie-related functionality

Genre Model

Movie Model

title: Character field for movie title

genre: Many-to-Many relationship with Genre model

release_year: Integer field for movie release year

rating: Decimal field for movie rating

description: Text field for movie description

poster: Image field for storing movie poster image

RentedMovie Model

movie: Foreign key relationship with Movie model

1 User Authentication

2 Movie Catalog Management

2.1 Movie Listing

The homepage shows a grid of movies available with necessary information:

Movie poster

Genres

2.2 Movie Details:

The movie details page provides comprehensive information about a selected movie:

Genre tags

2.3 Movie Search and Filtering:

Genre filtering

3. Renting Movies

Authenticated users can rent available movies:

Rent button on movie cards and detail pages

Automatic status update when a movie is rented

3.1 Returning Movies:

Users can return previously rented movies:

Return button on rented movie cards

Automatic status update when a movie is returned

3.2 My Rented Movies:

A dedicated page displays all movies currently rented by the user:

List of all rented movies

Direct access to movie details

Option to return movies

4 Admin Panel:

The system includes a comprehensive admin panel for administrators:

Complete movie and genre management

Card-based layout for movie display

User-specific menu items based on authentication status

Movie grid display

Specialized sections for top-rated and latest movies

5. Movie Detail Page:

The movie detail page layout includes:

5.1 My Rented Movies Page:

Grid layout of currently rented movies

Quick access to movie details and return functionality

The system loads movie data from a JSON file with the following structure:

Movie title, genres, release year, rating

A custom Django management command (load_movies) processes the JSON file and populates the database with movie information, including:

Creating genre entries

Loading movie details

6 Administrator Access:

Administrators can access the admin panel through:

Admin login: <http://127.0.0.1:8000/admin/>

6.1 Movie Rental Process :

Browse or search for available movies

View movie details

Movie is marked as rented and added to user's rentals

6.2 Movie Return Process:

Navigate to "My Rented Movies" page

Locate the movie to return

Movie is removed from user's rentals and marked as available

6.3 Movie Search and Discovery:

Apply genre, year, or rating filters

Explore "Top Rated" and "Latest" movie sections

Load movie data:

```
python manage.py load_movies
```

Create an administrator account:

7. Authentication Security

Admin access restrictions

The film rental system offers a comprehensive solution to manage film rental with a focus on user experience and functionality. Spontaneous interfaces, strong search functions and uninterrupted price management make it an effective tool for both users and administrators. The modular architecture of the system provides easy maintenance and future promotion.