Movie Rental System Documentation

Team Members:

Veekshitha Beladara Nanjegowda

Krishna Karthik

Sathvika Vaka

1. Introduction

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.

2. System Architecture

2.1 Technology Stack

The Movie Rental System is built using the following technologies:

- **Backend Framework**: Django 5.2
- **Frontend Technologies**: HTML5, CSS3, JavaScript
- **Database**: SQLite (default Django database)
- **UI Framework**: Bootstrap 5.3
- **Icon Library**: Font Awesome 6.0

2.2 Project Structure

The project follows Django's recommended project structure with the following main components:

- movie_rental/: Main project settings
- movies/: Primary application for handling movie-related functionality
- templates/: html -template for different pages
- static/: Static files (CSS, JavaScript, images)

3. Database Design

3.1 Data Models

The system uses three primary data models:

1. **Genre Model**
    - name: Character area for storage of style names
2. **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
    - cast: Text field for listing cast members
    - poster: Image field for storing movie poster image
    - poster_url: URL field for storing external poster image links
    - trailer_url: URL field for YouTube trailer links
    - is_rented: Boolean field indicating rental status
3. **RentedMovie Model**
    - user: Foreign key relationship with User model
    - movie: Foreign key relationship with Movie model
    - rented_date: Date and time field for rental timestamp

4. System Features

4.1 User Authentication

- The system implements a better user approval mechanism that includes:
- User registration with username and password
- Safe login and logout functionality
- Session management

- Access control based on certification status

4.2 Movie Catalog Management

4.2.1 Movie Listing

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

- Movie poster
- Title
- Release year
- Rating
- Genres
- Rental status

4.2.2 Movie Details

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

- Large poster display
- Title and release year
- Genre tags
- Rating
- Detailed description
- Cast information
- Trailer (if available)
- Rent/Return button based on current status

4.2.3 Movie Search and Filtering

The system offers robust search and filtering capabilities:

- Title-based search
- Genre filtering
- Release year filtering
- Rating filtering

- Multiple sorting options (rating, release year, title)

4.3 Rental Management

4.3.1 Renting Movies

Authenticated users can rent available movies:

- Rent button on movie cards and detail pages
- Automatic status update when a movie is rented
- Prevention of double-renting

4.3.2 Returning Movies

Users can return previously rented movies:

- Return button on rented movie cards
- Personal rental management page
- Automatic status update when a movie is returned

4.3.3 My Rented Movies

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

- List of all rented movies
- Rental date information
- Direct access to movie details
- Option to return movies

4.4 Admin Panel

The system includes a comprehensive admin panel for administrators:

- Complete movie and genre management
- User account management
- Rental record monitoring
- System-wide statistics and oversight

5. User Interface

5.1 Design Elements

The user interface employs a modern, responsive design with the following elements:

- Clean, minimalist aesthetic
- Card-based layout for movie display
- Color scheme based on dark blue, light blue, and white
- Responsive grid system that adapts to different screen sizes
- Intuitive navigation with clear visual hierarchy
- Interactive elements with hover and focus states

5.2 Navigation

The system provides consistent navigation through:

- Top navigation bar with links to main sections
- User-specific menu items based on authentication status
- Breadcrumb navigation on detail pages
- Back buttons for easy return to previous pages

5.3 Page Layout

5.3.1 Home Page

The home page features:

- Hero section with search functionality
- Filter and sort controls
- Movie grid display
- Specialized sections for top-rated and latest movies

5.3.2 Movie Detail Page

The movie detail page layout includes:

- Split-view design with poster on left, details on right
- Organized sections for different information types
- Embedded trailer (when available)
- Action buttons for renting/returning

5.3.3 My Rented Movies Page

The rental management page features:

- Grid layout of currently rented movies
- Rental date information
- Quick access to movie details and return functionality

6. Data Management

6.1 Data Sources

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

- Movie title, genres, release year, rating
- Description and cast information
- Poster image URL
- Trailer URL

6.2 Data Loading

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
- Handling external image URLs

7. System Access and Usage

7.1 Accessing the System

7.1.1 Regular User Access

Regular users can access the system through:

- Home page: http://127.0.0.1:8000/
- Login page: http://127.0.0.1:8000/accounts/login/
- Registration page: http://127.0.0.1:8000/signup/

7.1.2 Administrator Access

Administrators can access the admin panel through:

- Admin login: http://127.0.0.1:8000/admin/
- Credentials: Superuser username and password created during setup

7.2 User Workflows

7.2.1 New User Registration

1. Navigate to the Sign Up page
2. Enter username and password
3. Submit the form
4. Automatic login after successful registration

7.2.2 Movie Rental Process

1. Browse or search for available movies
2. View movie details
3. Click "Rent" button (must be logged in)
4. Movie is marked as rented and added to user's rentals

7.2.3 Movie Return Process

1. Navigate to "My Rented Movies" page
2. Locate the movie to return
3. Click "Return" button
4. Movie is removed from user's rentals and marked as available

7.2.4 Movie Search and Discovery

1. Use search bar for title-based search
2. Apply genre, year, or rating filters
3. Sort results by different criteria
4. Explore "Top Rated" and "Latest" movie sections

8. System Installation and Setup

8.1 Prerequisites

- Python 3.8 or higher
- Django 5.2
- Web browser with JavaScript enabled

8.2 Installation Steps

1. Clone the repository or extract the project files
2. Create a virtual environment:

```
python -m venv venv
```

3. Activate the virtual environment:
    - Windows: venv\Scripts\activate
    - Mac/Linux: source venv/bin/activate
4. Install dependencies:

```
pip install django
```

5. Set up the database:
6. python manage.py makemigrations

```
python manage.py migrate
```

7. Load movie data:

```
python manage.py load_movies
```

8. Create an administrator account:

```
python manage.py createsuperuser
```

9. Start the development server:

```
python manage.py runserver
```

9. Security Considerations

9.1 Authentication Security

- Password hashing with Django's built-in mechanisms
- CSRF protection on all forms
- Session security measures

9.2 Access Control

- Function-based authorization checks
- Template-level permission controls
- Admin access restrictions

10. Performance Optimization

10.1 Database Optimization

- Proper indexing on frequently queried fields
- Efficient query design in views
- Strategic use of Django's querysets

10.2 Frontend Performance

- Optimized image loading
- Responsive design principles
- Minimal JavaScript usage for essential functionality

11. Future Enhancements

The following enhancements could be considered for future development:

1. Implementing a rating and review system
2. Add payment processing for rent
3. Developing a recommendation engine
4. Implementing email notifications
5. Adding social sharing functionality
6. Create a user profile management system
7. Implementing content moderation for user-generated content

12. Conclusion

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.