Movie Rental System

Project Requirements

This project is a **Django-based Movie Catalog** that allows users to **browse, search, filter, rent movies**, and manage their rented movies. The project follows these guidelines:

- The project must be developed using **Django for backend** and **HTML, CSS, and JavaScript for frontend**.
- A **JSON file** will be used to store movie data.
- The project must contain at least **five pages**:
  - **Home Page (Movie Listing)**
  - **Login Page**
  - **Movie Details Page**
- User authentication (login/logout) is mandatory.
- A **movie rental system** must be implemented where a rented movie is **marked unavailable** for other users.
- The project should have a **responsive design** using CSS (Bootstrap or Tailwind optional).

Features Breakdown

1. Home Page (Movie Listing Page)

- Displays a **list of movies** with their **poster, title, genre, release year, and rating**.
- Includes a **search bar** to find movies by title.
- Users can **filter movies** by **genre, release year, and rating** (dropdown selection).
- Displays **"Top Rated Movies"** (rating > 8.0) and **"Latest Movies"** (released in the last 5 years).
- A "Sort by" dropdown for sorting movies by **rating, release year, or A-Z**.
- A **hover effect** to show movie posters when hovering over titles.
- A **"Rent" button**, which is **disabled** if the movie is already rented by another user.
- If a user **clicks on a movie**, they are taken to the **Movie Details Page**.

2. Login & Logout Page

- Users must **log in** to rent movies.
- **Sign-up option** for new users (username, email, password).
- Displays login errors for incorrect credentials.
- A **logout button** to end the session.

3. Movie Details Page

- Shows **detailed information** about the movie:
  - Title, genre, release year, rating, description, and cast.
  - A movie trailer (if available, embedded YouTube video).
  - "Rent Movie" button (disabled if already rented).
  - If the user rented the movie, they see a **"Return Movie"** button.
- Only **logged-in users** can rent movies.

Backend Details

- The backend is built using **Django**.
- A **JSON file (`movies.json`)** contains all movie data.

Navigation & Design

- A **navbar** allows easy navigation between **Home, My Rented Movies, Login/Logout**.
- Responsive design for **desktop users**.
- Clean and **user-friendly UI** with CSS.

Project Rules:

- No AI tools (such as ChatGPT, Bard, etc.) may be used for this project. If code similarity is detected, the project will not be graded.
- Copying code from any other source or team in the class is prohibited.
- Points will be deducted if any website functionality fails during the group presentation.

Presentation and Team Responsibilities:

- All group members must be present and able to answer questions from the instructor during the presentation. If a student cannot answer, the entire team will lose points.
- Each member should have a clear understanding of the functionalities and technologies used in the project.

All these instructions are guidelines or examples for you to follow or have an understanding of the requirements.

Let me know if you need any further clarification or assistance with your project.
Good Luck!