Movie Rental System

Contributor: Krishna karthik Penmetsa

1. Introduction:

This Movie Rental System is a Django-based web application designed to provide users with the ability to browse, search, filter, and rent movies. The platform includes user authentication features, responsive design, and dynamic movie management using a JSON file for data storage. It supports search and filter options by title, genre, rating, and release year, with rental functionality ensuring that a movie can be rented by only one user at a time.

2. How to Execute the Project:

- Python Environment Setup: check If there is python is available in the system or not and the if it is not there please install python.
- Create virtual environment using python -m venv venv command.
- Activate it using:
- Install Django using: pip install Django
- Apply migrations using: python manage.py migrate
- And run server using: python manage.py runserver
- To see our database that was created use python manage.py createsuperuser

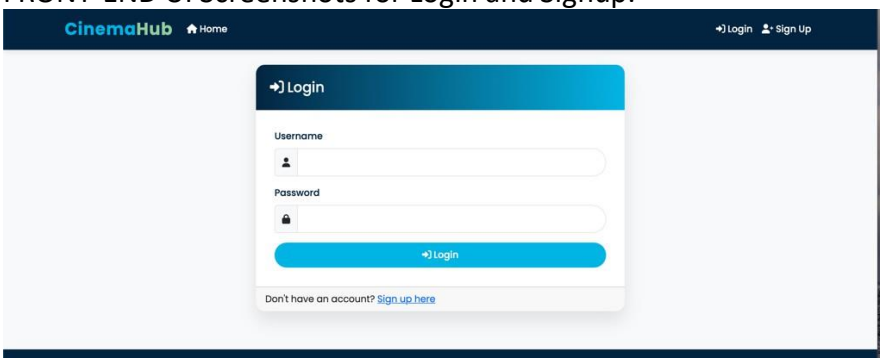
3. My Contribution:

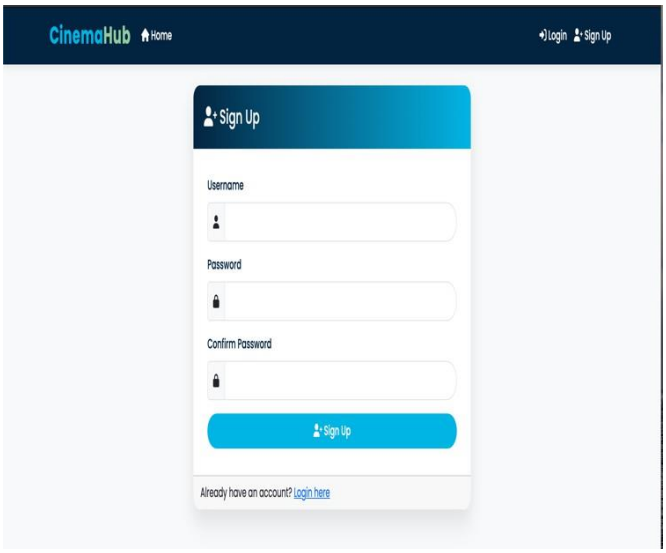
- I was responsible for setting up the environment and creation of project and application.
- I was responsible for writing the JSON file
- I worked on the creation of basic structure required and the forms that link to the backend.
- We worked as a team to link the backend but there were some minor issues we faced during creation of models and views.
- I tested the models and the data in it using a free software called the sqlviewerapp with which I saw the structure and the data entered into it after creating the super user.

4. Execution (screenshots):

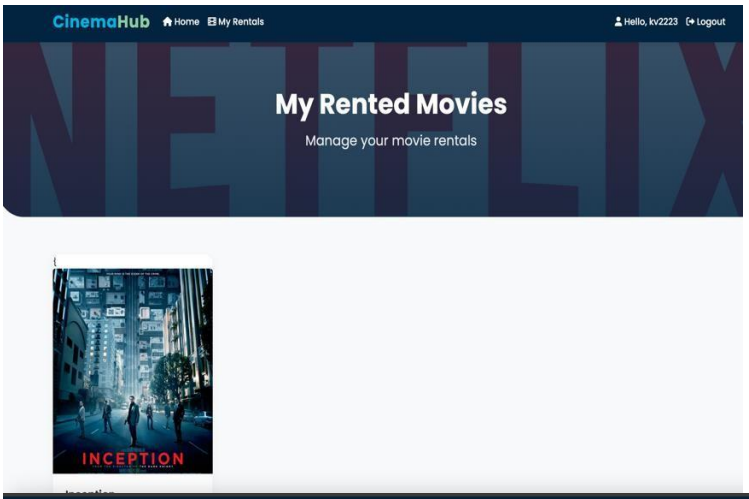
Below are successful execution screesnhots for our project.

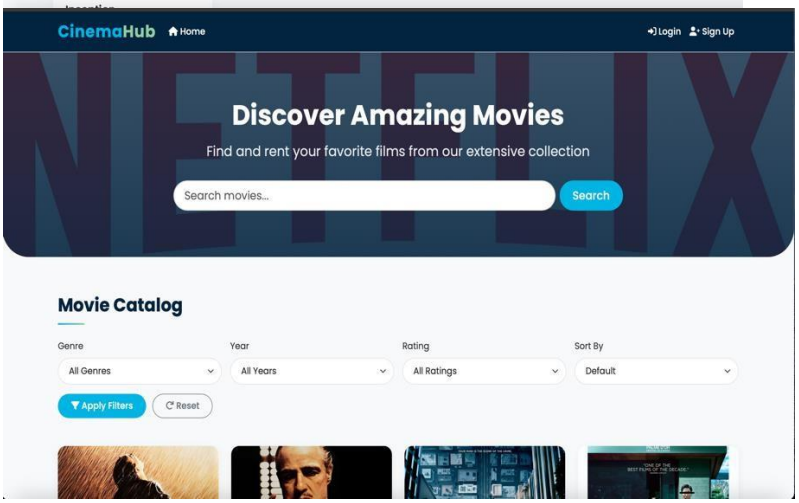
- FRONT-END UI Screenshots for Login and Signup:



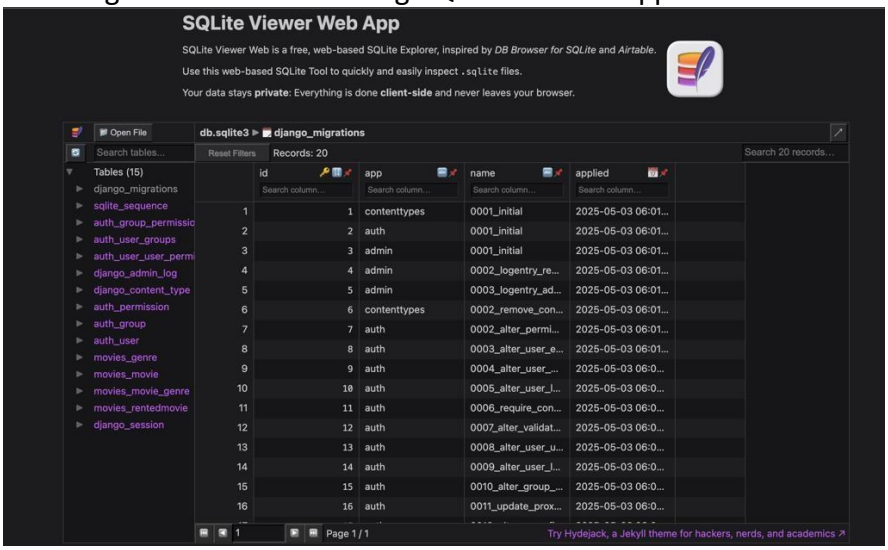


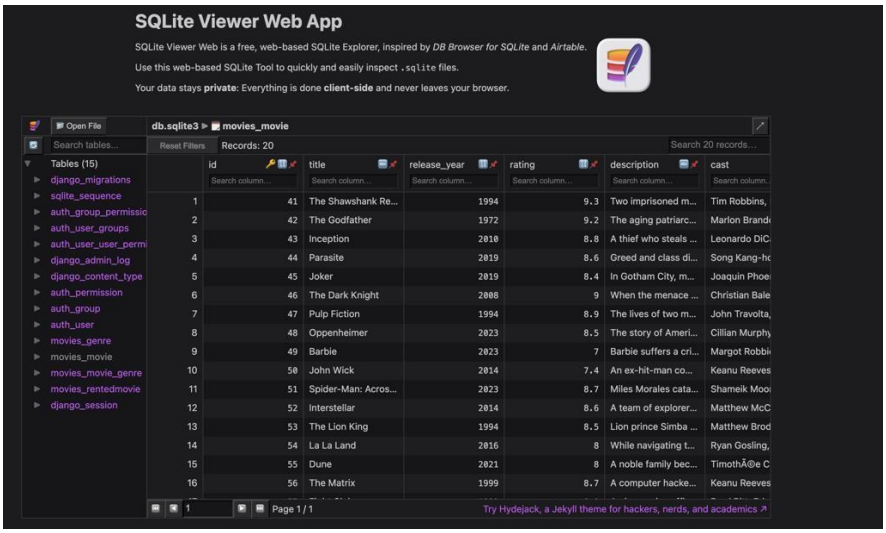
- FRONT-END UI:





- Checking model structure using SQL viewer webapp:





5. Conclusion:

The Movie Rental System provides a fully functional platform for browsing and managing movie rentals using Django. My contributions focused on core user functionalities like search, filtering, rental logic, and project documentation. This ensured a smooth and user-friendly experience aligned with the project's technical and functional requirements.

6. References:

- **Design using:**
  - **https://getbootstrap.com/**
  - **https://www.w3schools.com/css/**
- **Testing: https://sqliteviewer.app/**
- **Sample codes:** https://www.w3schools.com/django/