IS601 - Web System Development
SECTION: 104

REPORT

On

MOVIE RENTAL SYSTEM

By

NAME	UCID	EMAIL
Kenil Pravinbhai Avaiya	Ka683	ka683@njit.edu
Neel Chimanbhai Barvaliya	nb585	nb585@njit.edu
Sneh Vora	sv992	sv992@njit.edu

CONTENTS

NO	CONTENTS	PAGE NO.
1	Introduction	1
2	Software Specifications	3
3	Instruction	4
4	Web-site photos	6

INTRODUCTION

The Movie Rental System is a Django-based full-stack web application. It is designed to simulate a real-world online movie rental platform, where users can register, log in, browse a catalog of movies, and rent or return them based on availability. This project was created to demonstrate the practical application of both backend and frontend technologies in an integrated environment.

The system offers a clean and responsive user interface along with dynamic backend features such as user authentication, session management, and JSON-based data storage. The primary objective of this project is not only to implement core functionalities but also to reinforce concepts such as routing, data filtering, modular coding, and user-friendly design.

Project Structure

The Movie Rental System project is organized into two main parts: the **backend** (Django) and the **frontend** (React + Tailwind CSS).

Below is the detailed folder structure:

```
movie-rental-system/
├── backend/
│   ├── backend/
│   ├── movies/
│   ├── manage.py
│   ├── requirements.txt
│   └── Dockerfile

├── frontend/
│   ├── public/
│   ├── src/
│   ├── package.json
│   ├── tailwind.config.js
│   ├── vite.config.js
│   └── Dockerfile

├── docker-compose.yml
└── README.md
```

Important notes

- **Backend Technology:** Django for handling API requests, user authentication, and movie rental logic.
- **Frontend Technology:** React.js and Tailwind CSS for a modern and responsive UI.
- **Deployment and Environment:** Docker and Docker Compose for easy deployment and isolated environments

Key Features

- **User Authentication**: Users are required to register and log in to access the rental service. Secure session management ensures the protection of user data.
- **Home Page (Movie Listing)**: The home page presents a searchable and filterable list of movies, enabling users to sort the results based on their preferences.
- **Movie Details Page:** This page provides comprehensive information about a specific movie, including trailers and the option for logged-in users to rent or return the movie.
- **Movie Rental Logic:** Rented movies are marked as unavailable to other users and are displayed in the user's rental history.
- **Responsive Design:** The website is constructed using HTML, CSS (with optional Bootstrap or Tailwind), and is designed to be responsive, ensuring optimal functionality on both mobile and desktop devices.

TECHNOLOGIES AND TOOLS USED

The project integrates various frameworks and technologies to achieve the end goal of developing a functional web system.

Django (Python):

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel.

Django is a full-featured web framework that follows the Model-View-Controller (MVC) architectural pattern. It provides a set of tools and libraries for building web applications, including an ORM, a templating engine, and a built-in admin interface.

We utilized Django as the primary backend framework. It offers pre-built functionalities for routing, views, models, form handling, and particularly user authentication. This framework facilitates rapid development and ensures the organization of code into clean, modular components.

Python Virtual Environment (venv):

A virtual environment is used to isolate project dependencies, ensuring that the project runs with the correct versions of required libraries without affecting the global Python setup.

Frontend – React.js, Tailwind CSS, and Vite:

React.js, a popular JavaScript library for building user interfaces with a component-based architecture, is used for the frontend. Vite, a build tool, provides fast development and hot module replacement. Tailwind CSS, a utility-first CSS framework, aids in rapidly building responsive and modern user interfaces.

Key frontend functionalities include:

- User Interface: Interactive components for movie listing, details, and rental actions.
- Functionality: Search and filter functionality, conditional rendering based on login status or rental state.
- Technical Stack: React frontend with Django backend via API calls, Tailwind CSS for responsive layout.

JSON File (movies.json):

All movie data is stored in a JSON file rather than a traditional SQL database. This approach is suitable for small to mid-sized datasets and simplifies deployment by avoiding the need for database setup. The JSON file contains movie titles, genres, ratings, availability status, and additional information.

GitHub:

Version control is managed with Git, enabling collaboration and change tracking. GitHub hosts the codebase and maintains project history backups.

INSTRUCTION

To execute this project, we adhered to the following instructions:

Prerequisites

Make sure you have the following installed:

- Node.js and npm
- Python 3.x
- pip (Python package manager)
- Git
- Any code editor, such as Visual Studio Code (VS Code), is acceptable.

This guide walks you through how to set up and run the **Movie Rental System** application (ReactJS + Django REST API) locally on your computer.

Step 0 – clone the repository

Open your terminal and run:

- **git clone https://github.com/NeilNeel/movie-rental-system.git**
- **cd movie-rental-system**

Step 1 – open the project folder in the code editor(VS code)

Open two VS Code terminals and, navigate as follows:

- **terminal 1**: For the frontend folder
  - **cd frontend**

- **terminal 2**: For the backend folder
  - **cd backend**

Step 2 – Set up and run frontend

In the frontend folder terminal:

- **npm install**
- **npm run dev**

Your frontend is now running at: http://localhost:5173

Step 3 – Set Up and Run the Backend (Django)

In the **backend** folder terminal:

1. Create and Activate Virtual Environment
   - **python3 -m venv env**
   - **source env/bin/activate (For Mac/Linux)**
     or
   - **env\Scripts\activate (For Windows)**

2. Install Required Python Packages
   - **pip install -r requirements.txt**

3. Apply Database Migrations
   - **python manage.py makemigrations**
   - **python manage.py migrate**

4. Run the Django Server
   - **python manage.py runserver**

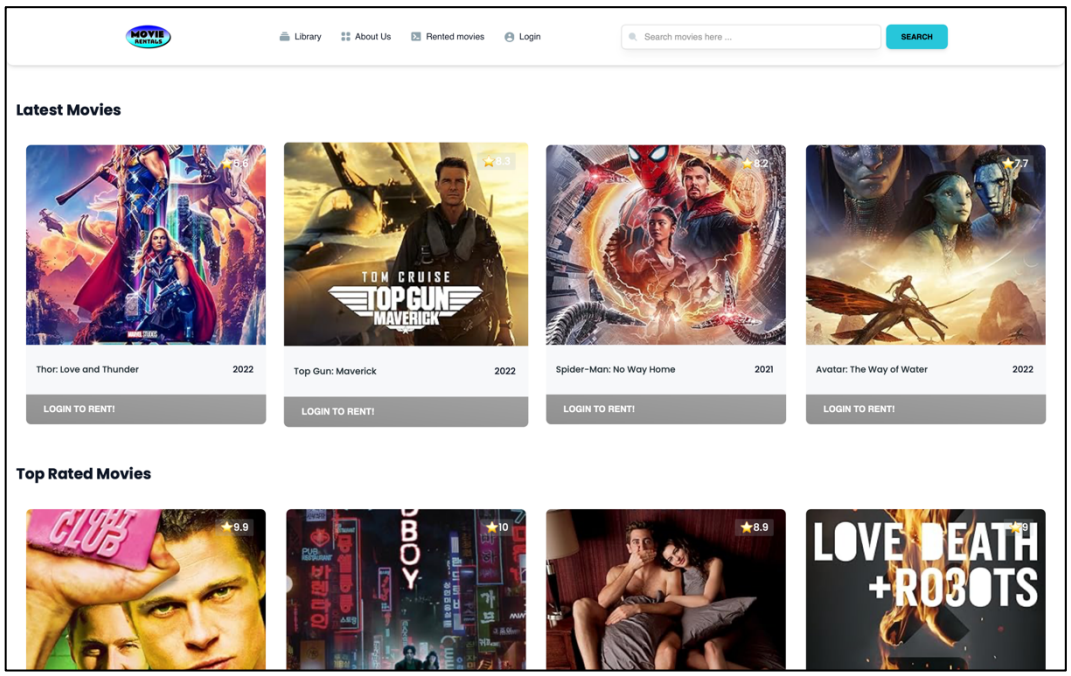
Your backend is now running at: http://127.0.0.1:8000

Final Notes

- Ensure the simultaneous operation of both frontend and backend servers.
- Launch your web browser and navigate to the URL: http://localhost:5173/
- Upon successful access, you will be able to engage with the comprehensive movie rental system application.

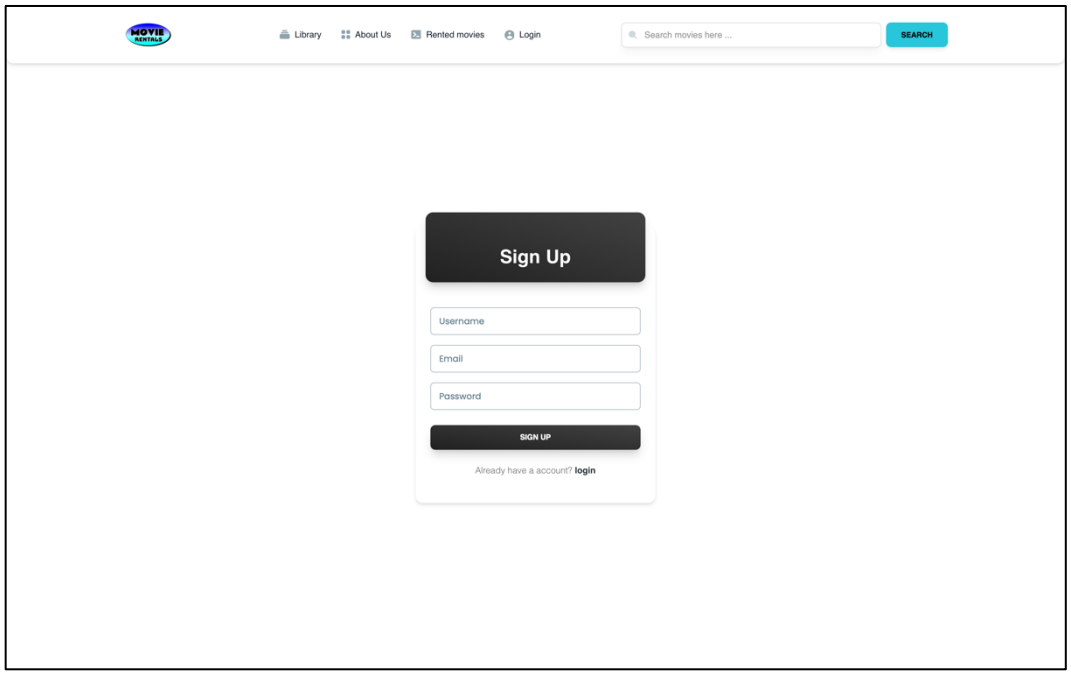
WEBSITE SCREENSHOTS

Main page (Home page): This is the page that you see when you access the website. It includes navigation of the different pages, as well as the latest movies and top-rated movies, and a search functionality.

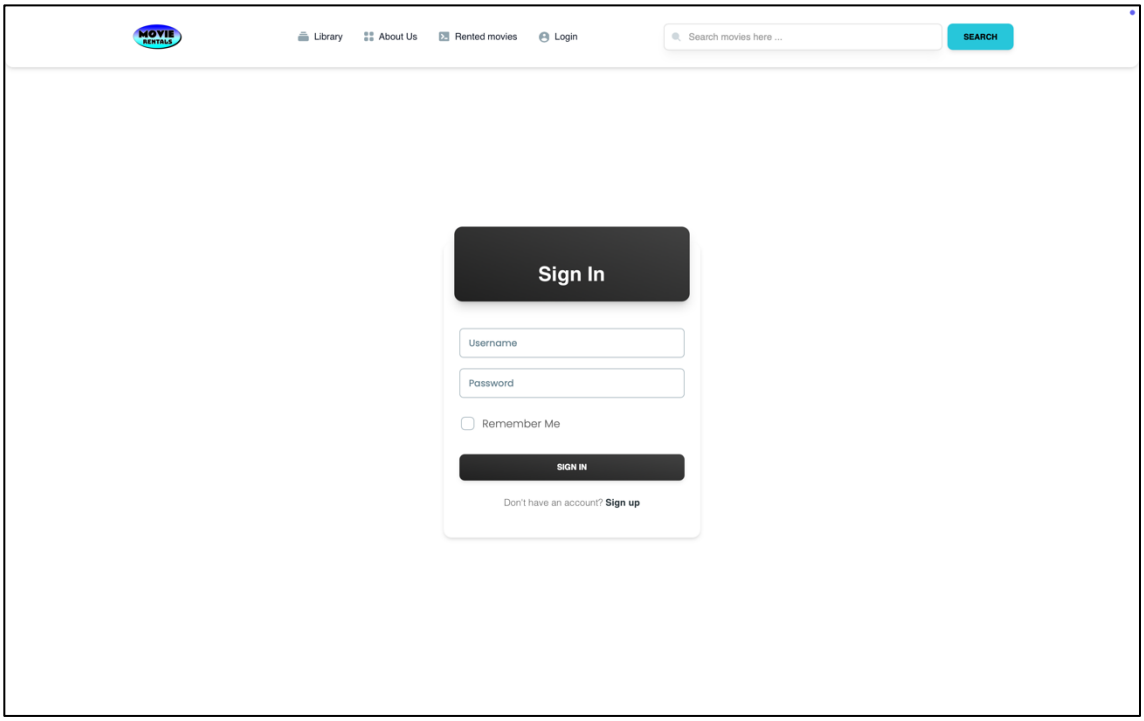


Home page

Login and sign-up page: if you have to rent a movie, then you have to first make a sign-up and then use that username and password to log in. A login is required to rent the movie.

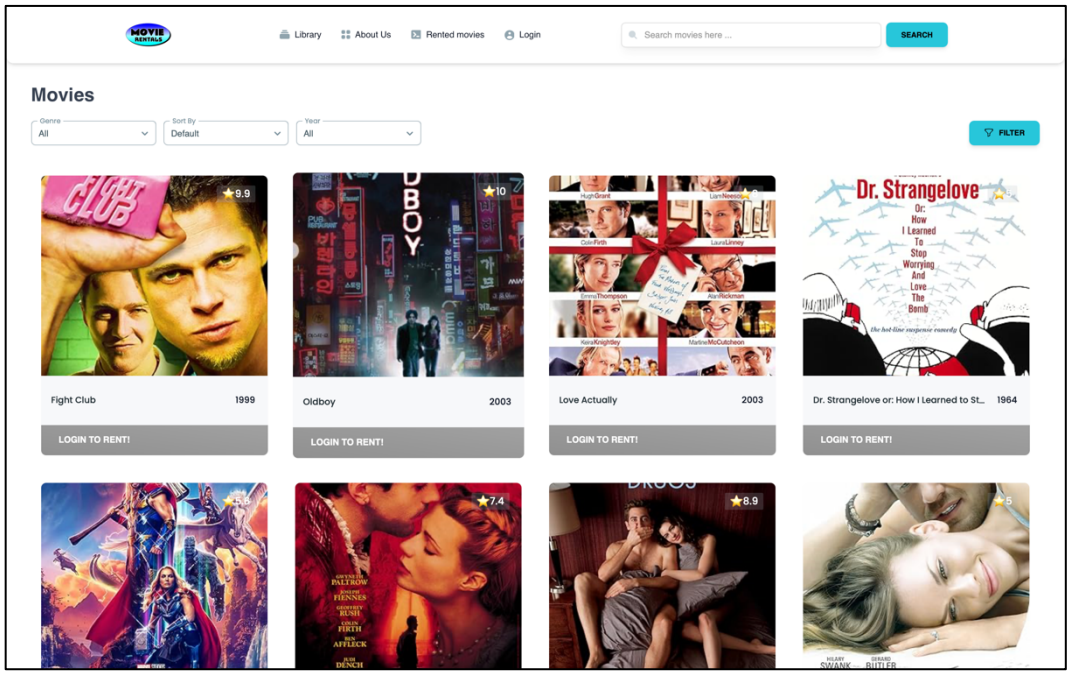


Sign-up page



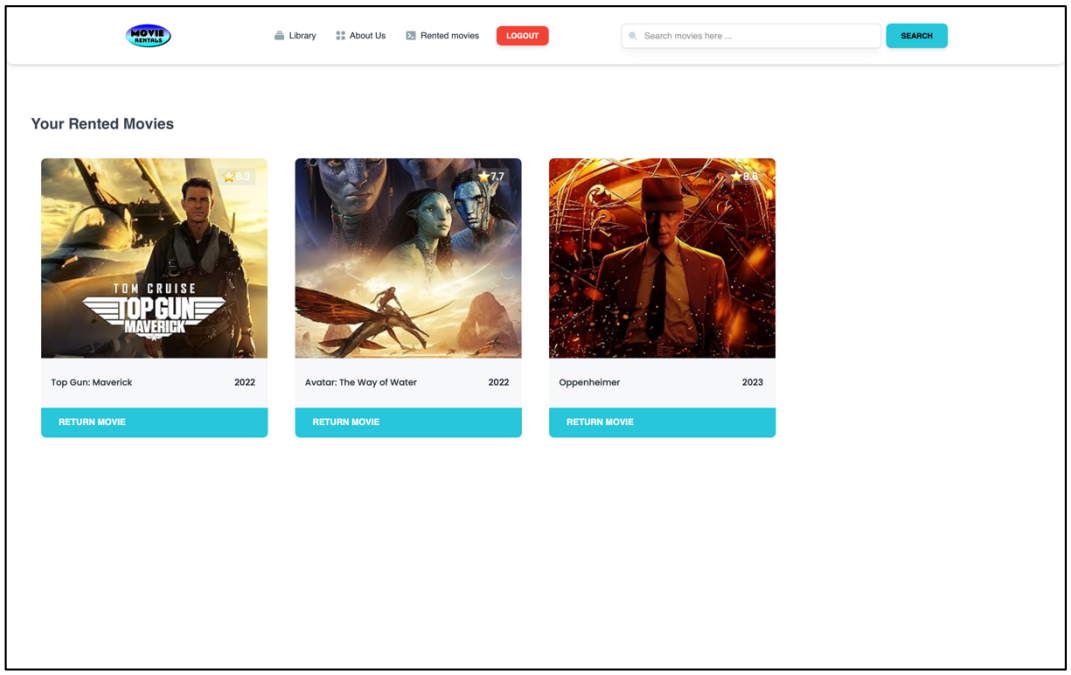
Login Page

Library page: This webpage presents a comprehensive list of all movies featured on the website. Additionally, you have the option to apply filters to refine your search and locate specific movies.



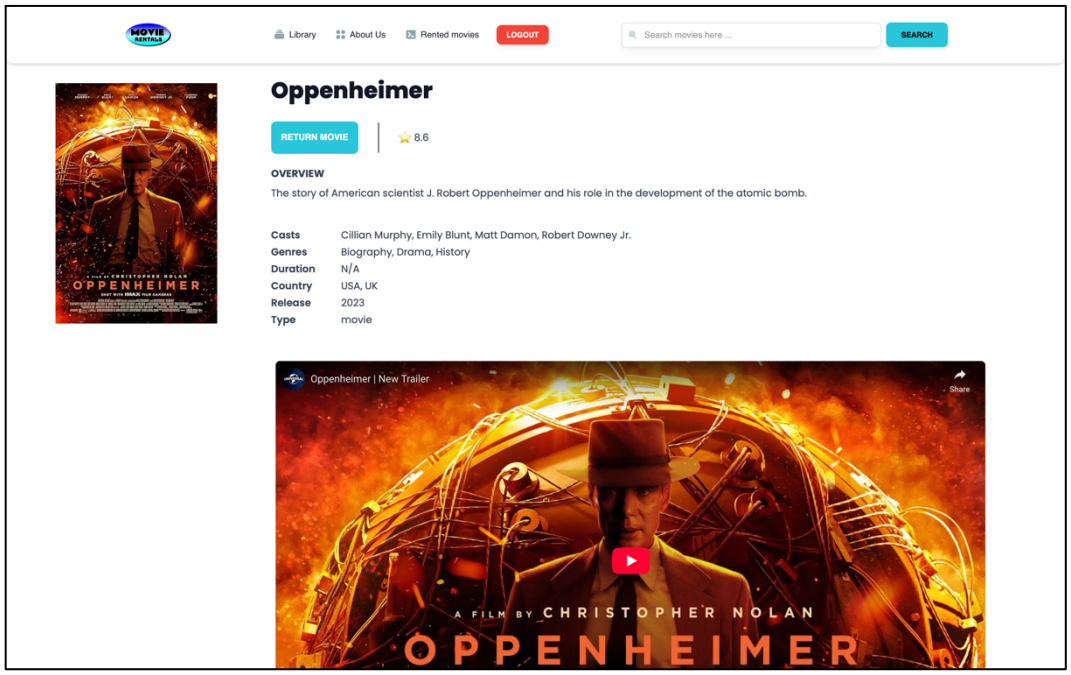
Library page

Rented Movie page: This webpage presents a list of movies rented by the login user.



Rented Movie page

Movie Description page: This webpage provides a comprehensive movie description with an overview, cast list, duration, genre, and trailer for the selected movie.



Movie Description page