**Movie Reviews: A Simple Movie Review Web Application**

**Overview**

**Movie Reviews** is a web application that allows users to register, log in, search for movies, and write/view reviews. This project was developed as a part of my capstone project, and it uses basic tools and concepts to demonstrate full-stack functionality.

**Features**

1. **User Authentication:**

• **Registration:** New users can create an account by providing a username and password.

• **Login/Logout:** Registered users can log in to access the application and log out when finished.

2. **Movie Search:**

• Users can enter a movie title to search for movie details.

• The application uses the **OMDb API** to fetch information such as the movie title, release year, plot summary, and poster image.

3. **Review System:**

• Logged-in users can write reviews for a movie.

• All reviews for a specific movie can be viewed by anyone using the app.

**Standard User Flow**

1. **Home Page:**

When users visit the site, they see a welcome message along with options to log in or register. Once logged in, a search form is available to find movies.

2. **Registration & Login:**

• **New Users:** Register with a username and password.

• **Existing Users:** Log in using their credentials.

3. **Movie Search:**

Users enter a movie title in the search form. The app then sends a request to the OMDb API (using the stored API key) to retrieve and display movie details.

4. **Writing a Review:**

From the movie details page, users can click to write a review. The review is then saved and associated with the logged-in user.

5. **Viewing Reviews:**

Users can view all reviews for the movie on a dedicated page.

6. **Logging Out:**

At any point, users can log out to end their session.

**API Integration**

• **OMDb API:**

The application integrates with the OMDb API to fetch movie details. The API key is stored as an environment variable (loaded in **app.py**), ensuring secure access to the movie data. This API returns essential movie information such as the title, year, plot, and poster image.

**Technology Stack**

• **Backend:**

• **Python** with the **Flask** framework is used to handle HTTP requests, define routes, and render HTML templates.

• **Database:**

• **PostgreSQL** is used to store user data and movie reviews, accessed via SQLAlchemy (an ORM).

• **Authentication:**

• **Flask-Login** manages user sessions.

• **Flask-Bcrypt** securely hashes passwords.

• **API:**

• **OMDb API** provides movie details, which are integrated into the app.

• **Project Structure:**

• Everything is contained in the **app.py** file to keep the project simple and manageable for a beginner.

**Additional Notes**

• **Single File Simplicity:**

This project uses only the **app.py** file to include configuration, database models, user authentication, API integration, and route definitions. This keeps the project structure very simple and ideal for learning purposes.

• **Local Development:**

The application is developed and tested locally on a MacBook Pro.

• **Future Enhancements:**

As I gain more experience, I may separate the project into multiple files for better organization, add improved error handling, and enhance the UI with additional styling.

**Conclusion**

**Movie Reviews** is a beginner-friendly web application that demonstrates essential web development concepts such as user authentication, API integration, and CRUD operations for reviews. With everything contained in a single **app.py** file, this project provides a clear and straightforward example of a full-stack web application built using Python, Flask, PostgreSQL, and the OMDb API.