Movie Favorites App

Project Description (Written by YenChen)

Users can utilize this application to search for movies in the IMDB database, save their favorite movies to a "My Favorites" list, add movies to their favorites, view their favorite movies, and remove movies from their favorites. The application uses Flask as the web framework and SQLAlchemy to interact with a database.

User Stories

Written by YenChen	Written by YuLun
1. User can view all favorite movies 2. User can view the create-new-favorite-movie page 3. User can create a new movie to the favorites 4. User can view details of a specific movie in their favorites 5. User can remove a specific movie in their favorites 6. User can retrieve details for editing a specific movie in their favorites 7. User can update the details of a specific movie in their favorites	1. User can search movies from The Open Movie Database by inputting keywords on the search bar 2. User can add movies from the search results directly to their favorites page 3. User accesses their favorite movie table, the page smoothly retrieves and renders the data from the database.

Challenges & Solutions

	Challenges	Solutions
Yen Chen	1. A member dropped out, and due to everyone's busy schedules, the project faced a productivity crisis. 2. Because I made modifications to the .bash_profile while writing JavaScript, I couldn't locate the Python path when working on this Movie Favorites App.	1. Given my faster coding pace, I assumed the departing member's tasks, wrote meeting records, and clarified responsibilities. 2. In the .bash_profile, I used the following command to reposition the Python path: `export PATH=$HOME/bin:/usr/local/bin/python3:/usr/bin:$PATH`.
Yu Lun	1. At the beginning, our team was not proficient in using the Flask backend framework, and we faced challenges in properly distributing data across various database tables and managing the database efficiently. 2. To have a smoother presentation, I simplified the interface to showcase the features I developed. During this process, I spent time understanding the CORS logic, which enabled me to successfully integrate backend APIs and effectively run the frontend features.	1. After studying the framework's official documentation and watching tutorials on YouTube, I gained a foundational understanding of how to apply this skill to our project. With this knowledge, I can come up with strategies for constructing an appropriate database structure and data flow, and subsequently address bugs originating from the database 2. I delved into various documents and articles that explained the use of CORS in data requests from the server. This additional learning allowed me to effectively integrate my frontend experience with my newfound understanding of backend processes.

Tools (Written by YenChen):

1. Flask: Used as a web framework for building the backend of the application.
2. SQLAlchemy: Used to interact with the database.
3. Git: Used for version control.
4. Pipenv: Used for virtual environment and dependency management.
5. JSON: Used to pass data between different parts of the application.
6. Postman: Used to test the correctness and availability of APIs.

Division of tasks:

YenChen	YuLun
1. Implement **seven** functionalities. 2. Establish a database to store **favorites**. 3. Alternate hosting meetings. 4. Meeting documentation. 5. Complete portions of Summary, Demo, Project Presentation. 6. Discussed the app design and drew the design diagram.	1. Implement **three** functionalities. 2. Managing our project's github repo. 3. Establish a database to store **search results**. 4. Alternate hosting meetings. 5. Complete portions of Summary, Demo, Project Presentation. 6. Discussed the app design.