

Title: Purrfect Movie Recommender

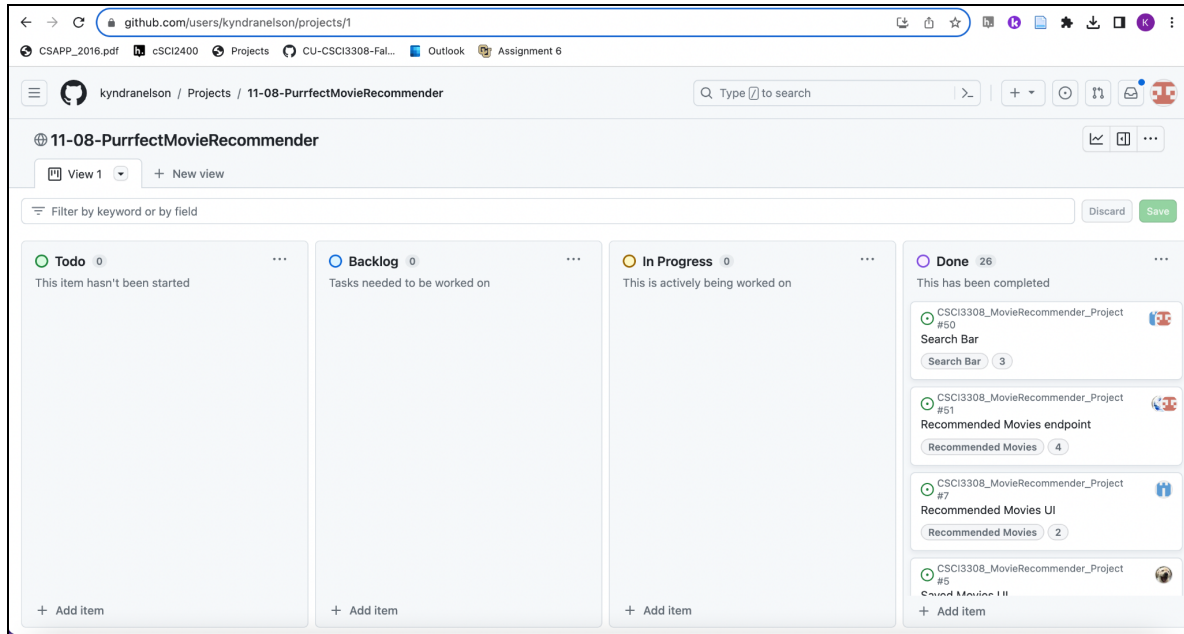
Who: Robin Luo, Mathis Fitwi, Matt Corley, Micah Dew, Kyndra Nelson

Project Description: This is the Purrfect Movie Recommender. As a team of 5, we created a WebApp that allows unique users to register, login, save, and mark movies as watched. Whether or not the user is logged in or not, they are still able to use the WebApp. The user does need to be logged in to use the save and watch movie buttons though. The API that was used was the TMDB movie database in order to get the movies for each component of the website. Each user can browse through a series of popular movies, highly rated movies, and even search movies they are interested in and save them for later or mark the movie as watched. Each movie is displayed in a movie card-like format that shows its release date, genre, rating, and description for the user's curiosity. The user is also able to locate the saved and watched movies if they want to see what they added to either list. Also on these said pages, if the user made a mistake and accidentally added a movie to either page, there is a delete button that will remove it from the database and from that page.

However, all of this movie storage and browsing is also very important to the main feature of the WebApp, which is its ability to recommend movies. Using all of the movies you saved and marked as watch for later, it takes the genres of your movies and gathers some popular movies based on your taste and displays them for you to add to your account. Once the user is done using the website, feel free to logout.

Project Tracker - GitHub project board:

<https://github.com/users/kyndranelson/projects/1>



Video: <https://vimeo.com/893606006>

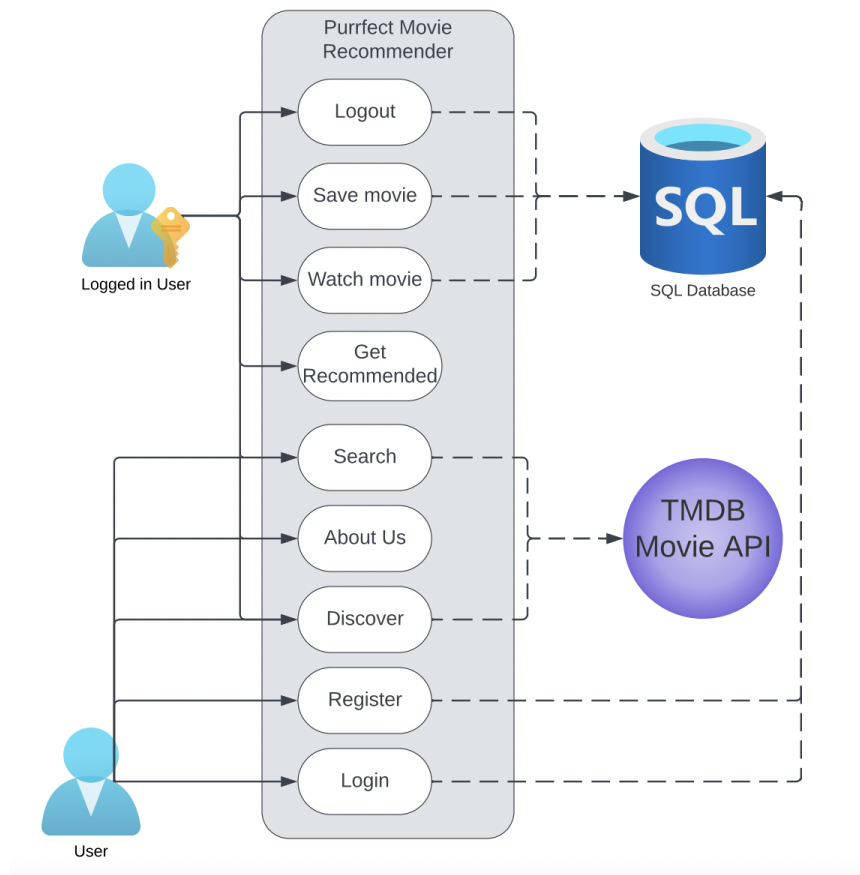
VCS Link: https://github.com/kyndranelson/CSCI3308_MovieRecommender_Project

Contributions:

- Micah - I contributed to various things in this project. I helped on the frameworks for most of the UI pages, so partials and pages. I made the tables and database for our web app's data. I also was a part of a lot of the API endpoints that are the foundation of the functionality of the website and its features. Endpoints such as the save and watched movies buttons and storage and the delete watched movies button. As well as some queries for other endpoints such as the recommender.
- Robin - The contributions I made were finding and using the movie database and using the API calls to get data from the database. I also created the carousel function on the discover movies page and made a search bar so that users could look up specific movies if they wanted too. Towards the end I worked a lot on quality of life changes such as changing the appearance of the save and watched buttons after you pressed it and fixed some of the stuff that would cause internal server errors

- Kyndra - I worked mostly on diagrams needed for project labs and the project overall, how the website looked, the logout API, weekly deliverables, and other small things that needed to be done. In terms of the diagrams, I created the use case, Wireframes, and the Architecture diagram. For how the website looked, I created the logo for the website from scratch and implemented it, as well as dabbled a lot in the style.css file to get pages like the logout, login, register, about me, and the discover page to look more professional. The other small tasks are mostly bug fixes and implementing other features which can be found on the project board.
- Matt - I mainly focused on the backend side of the project. I started out by working on the login routes which I took inspiration from previous labs. I then wrote out test cases for our server routes to ensure they worked correctly. I helped on some one-off bugs throughout the project as well like helping to get the logout to work and getting certain routes to render when they were crashing. Finally, I wrote out two versions of the recommendation system. The first attempt was quite buggy and would only work under certain conditions. My newly revised recommendation system works a lot better, and gives the user feedback if there is not enough data yet. It also references *why* a movie was recommended to provide extra feedback to the user.
- Mathis - I worked around some things. Some are helping out with the architecture diagram and wireframes. Getting the login and registration routes to work on those pages and a simple design layout of both pages. I helped get the TMDb Api call to work for the carousel functionality in the discover page. Towards the end though I didn't do as much as I like. Though, I created the slides for the presentation and helped with hosting our site on Azure's cloud.

Use Case Diagram:



Test results:

The user overall followed the instructions properly in the original [test plan](#). They also dabbled around with other buttons as they were following these instructions because they were instructed they could test other stuff while they were doing the original test.

Test plan 1: Check login and users system to check if navbar works properly

- What are the users doing?
 - They are making sure that the login page works and everything shows up properly for if the user is logged in or not. The user just did not use the given account(Tester) to check and see if the page works. The user made their own account then added and saved movies from the discover page.
- What is the user's reasoning for their actions?
 - The user's reasoning was that they wanted to create their own account to super make sure that the website is working properly.
- Is their behavior consistent with the use case?

- The user's behavior is mostly consistent with the use case, the user just made the change to use the register page as well as all of the other steps of testing the login page.
- If there is a deviation from the expected actions, what is the reason for that?
 - The user wanted to create their own account instead to also test the register page.
- Did you use that to make changes to your application? If so, what changes did you make?
 - No changes were made.

Test plan 2: Add and Remove movies

- What are the users doing?
 - The user used the same account that they had created just before this instead of the testing account. The user saved a bunch of movies and a bunch of watched movies with the buttons near the bottom of the movie cards. They then checked if they were in the saved and watched movie routes and deleted a couple movies from both routes. After that, they went back to the routes to check if they were successfully deleted or not.
- What is the user's reasoning for their actions?
 - The user's stated that they wanted to actually follow the instructions this time.
- Is their behavior consistent with the use case?
 - The user's behavior is consistent with the use case; they just forgot to logout of their account at the end.
- If there is a deviation from the expected actions, what is the reason for that?
 - The user has just forgotten to logout of their account.
- Did you use that to make changes to your application? If so, what changes did you make?
 - No changes were made.

Test plan 3: Recommendation system

- What are the users doing?
 - The user used the same account that they had created just before this instead of the testing account. The user added movies to their saved and watched movies list in order to test the recommendation system then clicked the button to get the recommended movies.
- What is the user's reasoning for their actions?

- The user's stated that they wanted to add as many movies that they liked as they could in order to get the best recommendations.
- Is their behavior consistent with the use case?
 - The user's behavior is consistent with the use case.
- If there is a deviation from the expected actions, what is the reason for that?
 - No deviation.
- Did you use that to make changes to your application? If so, what changes did you make?
 - No changes were made.

Test plan 4: Search Button functionality

- What are the users doing?
 - The user is testing the account being logged out first then logging in and testing the search function. The user first searched for "IT" and found the correct results they were looking for. They also noticed that when logged out the saved and watched movie buttons do not show up. The user then logged into their account and searched up the same movie. They then added this to their saved movies, checked the saved movies tab, and found the movie.
- What is the user's reasoning for their actions?
 - The user stated that they wanted to make sure that the movie would save correctly again even when looking up the movie in the first place.
- Is their behavior consistent with the use case?
 - The user's behavior is not very consistent with the use case.
- If there is a deviation from the expected actions, what is the reason for that?
 - The user wanted to make sure that the save movie button would still work even when searching up a movie. They still hit all of the things that needed to be tested for this feature of the website though.
- Did you use that to make changes to your application? If so, what changes did you make?
 - No changes were made.

Deployment: Link to deployment environment or a written description of how the app was deployed and how one might access/run the app. The app must be live, working, and accessible to your TA.

Deployed Application Link:

- Cloud Link:
 - <http://recitation-11-team-08.eastus.cloudapp.azure.com:3000>

How to run Application Locally:

- git clone the repository within the github
 - https://github.com/kyndranelson/CSCI3308_MovieRecommender_Project#csci3308_movierecommender_project
 - git clone
git@github.com:kyndranelson/CSCI3308_MovieRecommender_Project.git
 - make sure to have working ssh key within all of this
- Within CodeComponents folder:
 - docker compose up -d
 - use this link in address bar within google: <http://localhost:3000/>
 - the link should reroute to the Discover page
- Enjoy!