

# Title: TopReview

## Team Members:

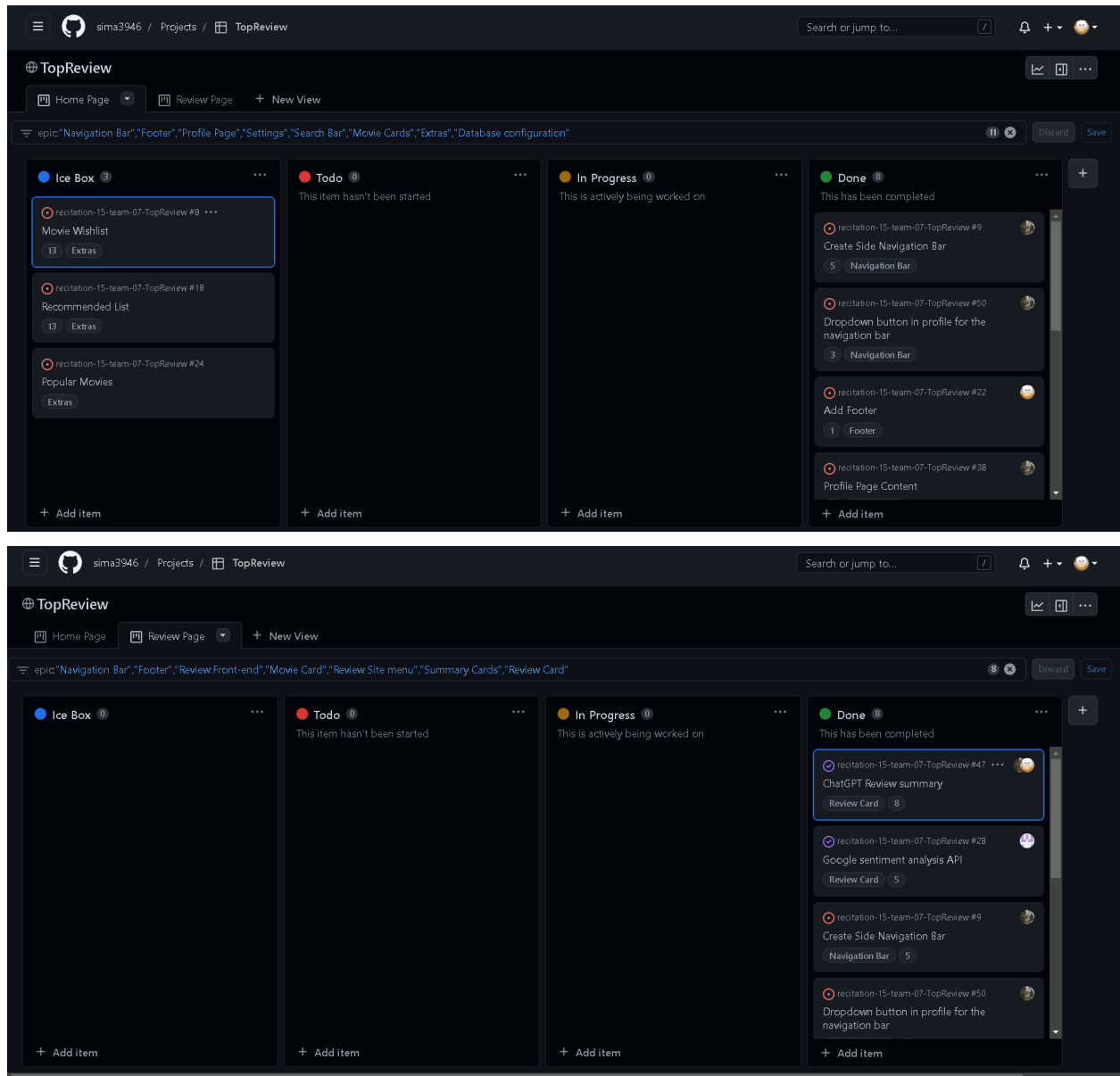
- Gunhi Kim; guki7733 ; [guki7733@colorado.edu](mailto:guki7733@colorado.edu)
- Tyler Kloster; tkloster01; [tykl4001@colorado.edu](mailto:tykl4001@colorado.edu)
- Siranush Mazmandyan; sima3946; [sima3946@colorado.edu](mailto:sima3946@colorado.edu)
- Alex Pham; avqpham01; [alph4995@colorado.edu](mailto:alph4995@colorado.edu)
- Kira Velez; kive7791; [kive7791@colorado.edu](mailto:kive7791@colorado.edu)

## Description: 200-word summary

Some of the highest-grossing movies to date are incredibly long. In fact, over the past few decades, the average length of movies has been increasing. With such a commitment, movie watchers need an easy way to know whether a movie is worth watching. This is often done by looking at summaries. More importantly, however, we look at reviews for guidance. With each movie racking up hundreds of reviews, finding a movie is an issue. TopReview is dedicated to making the process easier. The process is made substantially easier by aggregating these hundreds of reviews into three categories - negative, neutral, and positive. Drawing from open-source sites such as TMDb and Letterbox, TopReview first sorts each review into one of these three categories using sentimental analysis. Google Cloud provides this analysis and is a natural language processor that deduces digital text's overall tone and underlying sentiments. Once categorized, the reviews are funneled into ChatGPT along with a prompt of our making to create a comprehensive summary for each category. If you are looking for movies to watch, look no further than TopReview. Finding movies has just been made easy.

## GitHub project board:

- Link to your Project Tracker (for instructor & TAs):  
<https://github.com/users/sima3946/projects/1>
- Screenshot showing your project in your project tracker:



Video:

Link to the video: <https://youtu.be/vx3jx8SIKys>

VCS:

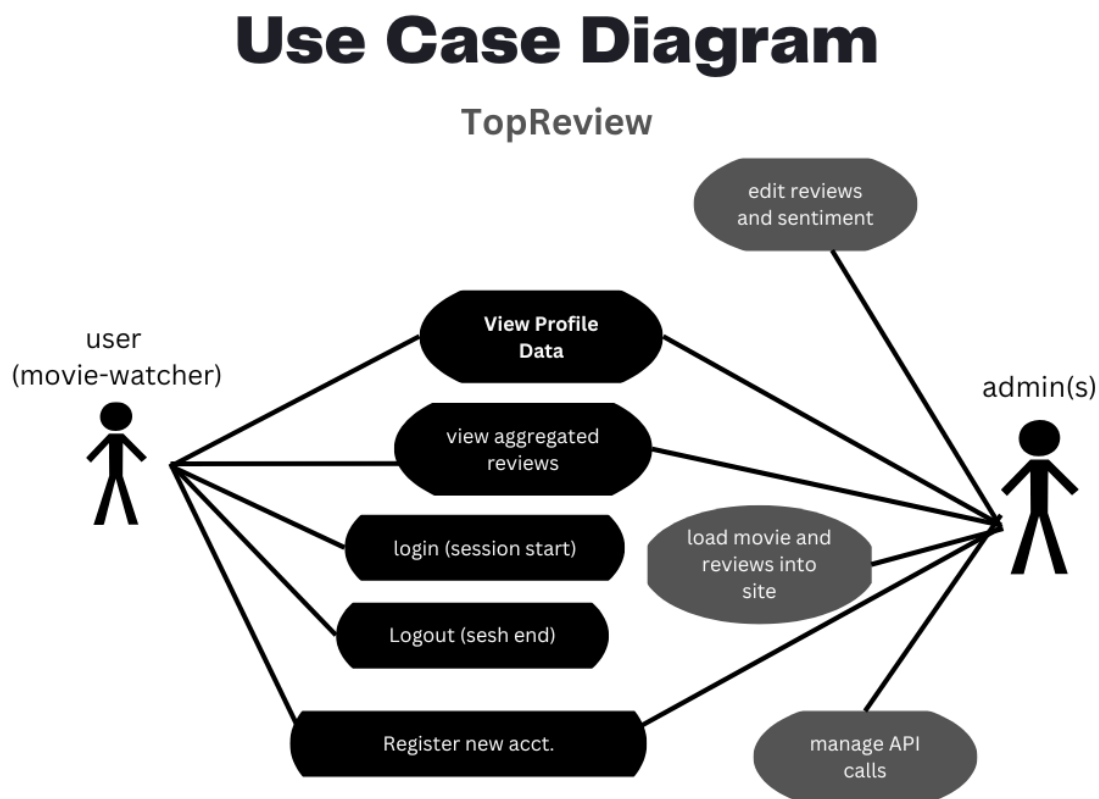
Link to your git Repository:  
<https://github.com/sima3946/recitation-15-team-07-TopReview>

## Contributions:

Gunhi Kim:	For TopReview, my main contribution was the backend development regarding populating the database for movies and movie reviews. Doing this required me to adjust our database to ensure that user, movie, and movie review data were stored and pulled correctly. I also built endpoints that called on the following external APIs: TMDb (The Movie Database), LetterBoxd, and Google Cloud. Additionally, I was able to draft a project outline and presentation slides for the group's presentation.
Tyler Kloster:	For the first week of this project, I made the front-end design for the registration page. I also implemented the backend API calls for the login and register pages. In week two, I integrated these pages with a homepage that I made. I also worked with Gunhi to rework the TMDb API calls to work with the database and added this movie information to the cards on the home. In week three, I made the review page allowing you to inspect the movies further and show the reviews from chatGPT. This week I did further work on the review page.
Siranush Mazmandyan:	After creating the GitHub repository, my work was first to establish a working navigation bar that would redirect users to respective pages through icons. Then I created the Profile page, API calls, and Logout API, and debug issues found on the Home page. I also worked on correctly initializing the movie database upon reaching the Home Page. My work was also mostly focused on leading meetings, taking notes during the TA/Group meetings, and creating Release Notes. In the final week I worked solely on deploying the application and final revisions.
Alex Pham:	First, I helped with structuring the project. Also, my contribution was split between two areas: (1) creating the login page and (2) backend development relating to making calls to a sentiment analysis. My work on the login page involved a lot of front-end work. I also made an API call to google cloud to give a score for each review. The score allows me to write a query to organize reviews into negative, neutral, and positive. Finally, I did a lot of debugging with review calls.

Kira Velez:	My contribution was planning and structuring the project, creating a footer, helping create the Review page and layouts of other pages, debugging, and creating the ChatGPT API. For the first week, I helped structure the project by using Canvas collaboration, updating the project board, and working on the initial structure of files. In the second week, I created the footer and debugged CSS files. During the third week, I helped debug the database code and started working on the ChatGPT API. For the final week, I finished the ChatGPT API and set up the final look for the Review page.
-------------	--

Use Case Diagram:



## Test Results:

### Test Cases

1. Test to see when the user inputs a username and password to the register page, it hashes the password and stores it

Test to see if you can then log in with this account

Test on both the register and login page that if you don't fill a field, you won't move on

Test to see if your username is already taken you can't register using that name

Test to see if login is case sensitive

Observations: users implemented their own uses of passwords and usernames which did not cause any issues other than a username being taken, which was taken care of by having the login page be refreshed without saving the user's data. Otherwise the users were consistent with the use cases as predicted.

2. Test to see if all the movie pages are accessible and show the appropriate reviews.

Go to all the given websites and see if the scores match up and the review compilations seem appropriate

All information in the table gets portrayed on the page correctly

Observations: The users were able to click on their movies of choice and see the summarized reviews as done through the ChatGPT API. The users, for those who chose to do so, were able to visit the websites of TMDb and LetterBoxd and see that the reviews given matched with those given through our website. There were no issues and the users proceeded as the use cases predicted.

3. Test to see if you can get to the profile page from the home page

See if the username and password shown is correctly

See if you can then logout and login and the information is different

Observations: The users were able to access the profile page and review their usernames and password with no issues. If they chose to logout, then they were able to do so; when other users logged in, their own usernames and passwords were shown as well. These users acted upon the predicted course of action as determined by our use case.

4. Test to see that the reviews generated on the review page makes sense and the positive section correlates to a positive review and so on.

Make sure all the movie information given on the page is correct, for example the title and image

Observations: Users, when reading the reviews and asked if the reviews were comprehensible, agreed that the reviews were easy to understand and gave information that could have helped in deciding whether or not they would watch the movie given. Some users wished to see the description of the movie (aside from the title) for more

information, however we were unable to implement this because of a shortage of time and workforce.

## Deployment

Link to deployment environment: <http://recitation-015-team-07.eastus.cloudapp.azure.com:3000/>

For this app to run, Docker must run the files of the application folder (which can be done using Microsoft Azure). The application was deployed using Azure through a virtual Linux machine that then allows the user to access the website after running Docker through the terminal.