Movie Master

The Team:

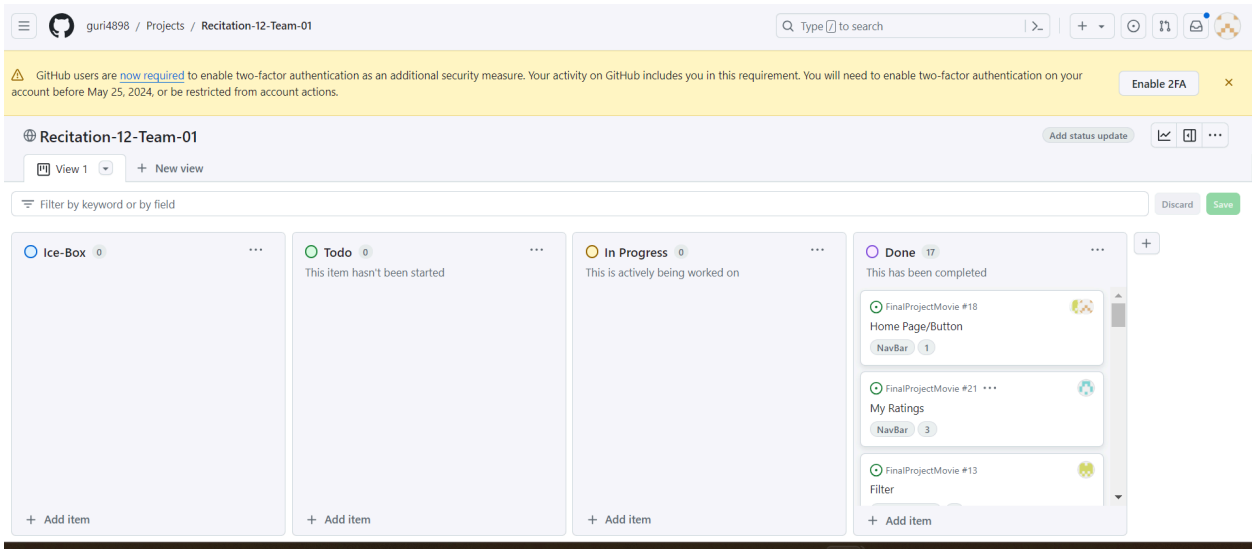
Andrew Hyunh, Grace Waida, Audrey Ly, Guillermo Rivas, Bhoopender (Honey) Singh, Sergio Lopez

Project description (Andrew):

For our final project, we made an application called Movie Master that allows users to rate, review, and discover new movies that they might've not found before. Users are able to friend other people, filter movies by genre, year or director, and search for specific movies that they want to read about. If there is a movie that they want to review that does not exist on our application, they can fill in the movie data on their own to have it added to the app. The application makes use of user sessions so that once someone logs in, their session is saved on the backend and every action they take is associated with their account.

The app is intended for anyone to use, whether it be movie enthusiasts or casual movie enjoyers. Each movie rating is based on user reviews, not critic reviews, so people can gain an understanding of how the general population enjoys the film rather than professional critics.

GitHub project board:



[Link to Project Board]

Video link (Sergio):

https://drive.google.com/file/d/1Zs9cdBmCMd8npMPxM1FRtxuwI5G8ZYUy/view?usp=share_link

VCS

https://github.com/guri4898/FinalProjectMovie

Contributions from each team member (everyone):

Sergio: Account settings page (UI, styling and backend), Add review (UI, styling of the details movie page and backend to add a new review and rating), regex for email and password. Message partial and implementation in login, register and account settings. Test cases for registration and a couple of more test cases. Saving user information.

Guillermo: I helped format the Home Page, specifically the filter feature. I also worked together with other teammates to style the Home Page using Cards and Carousels from Bootstrap to implement our movies.I was also in charge of running Lab 13 which involved hosting and setting up the environment to host our application on the Azure Web Service. The technologies that I worked with were HTML, JavaScript, Bootstrap, CSS, Azure Web Services, Handlebars, etc.

Andrew: The features that I worked on were the register page, the friend feature, the add movie feature, the page to display individual movie reviews and ratings, and the favorite movie feature. I also designed the database for our project, and I helped my other team members debug the issues they were facing while working on their features. I used HTML, CSS, handlebars, SQL, Javascript, and Postgresql to implement all the features that I was responsible for.

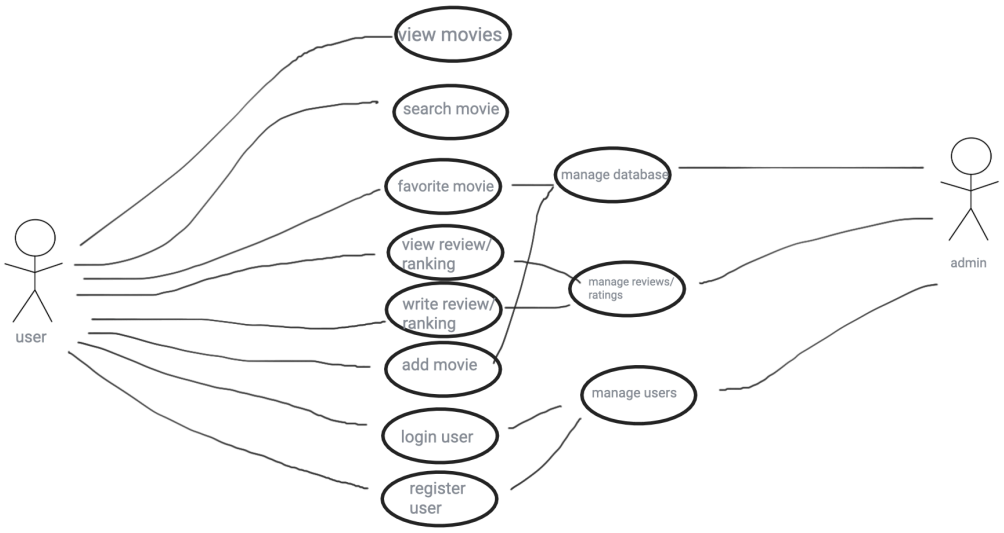
Grace: I primarily worked on the formatting and look of the website. In the beginning I worked on the basic user interfaces such as the login page, logout page, image display, navbar, adding movies to favorites with Andrew as well as helping out a bit on the home page. The technologies that I used were HTML, handlebars, and Javascript to help with implementing all of the features I worked on.

Audrey: I worked on the home page, search bar, and UI design on multiple different pages. For the search bar, I worked on the front end and back end, writing the code to make the home page rerender when a search is made and designing how the movies showed up when you search a movie. With that, I worked on many individual pages and formatted the look of pages like the

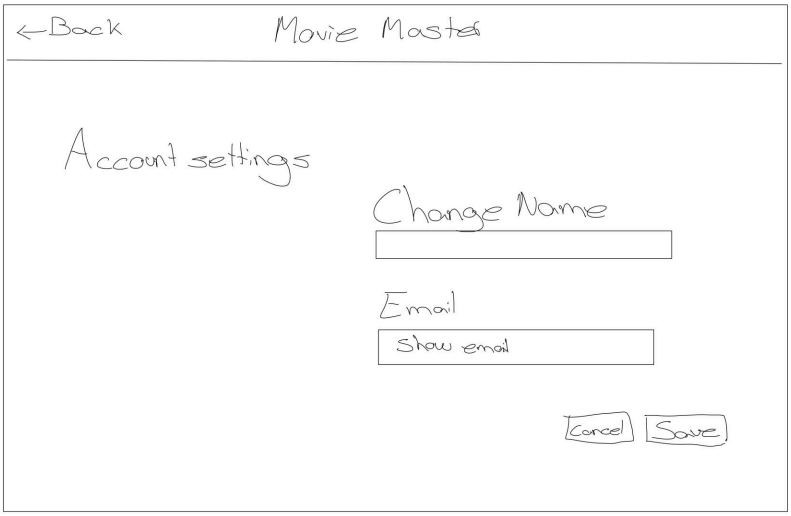
add movie page and the main home page. THe technologies I used were HTML, handlebars, Javascript to implement the features I worked on.

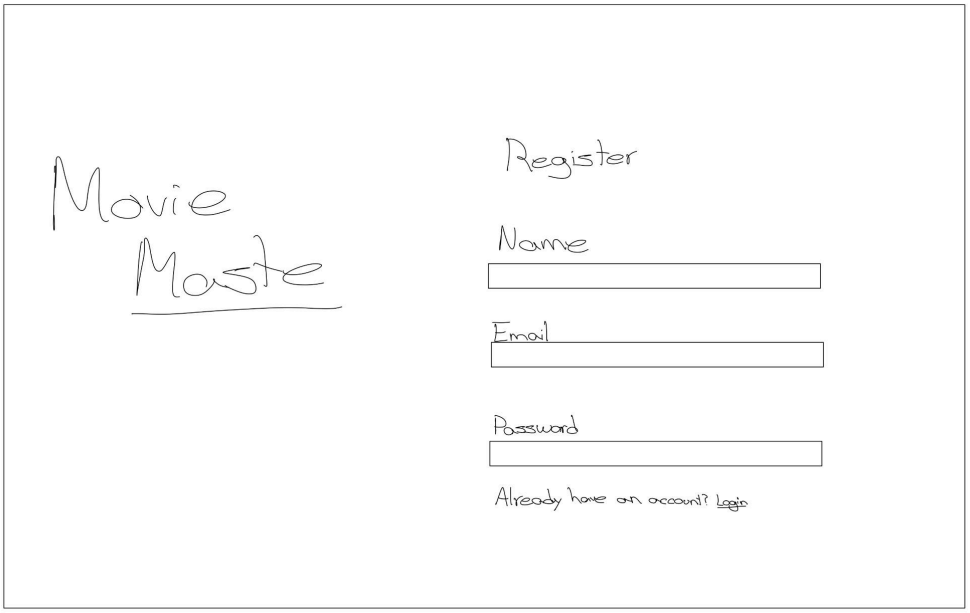
Honey: nothing

Use Case Diagram (Audrey):



Wireframes (sergio):




Movie
Maste
Register
Name
Email
Password
Already have an account? Login


Movie
Maste
Login
Email
Password
Don't have an account? Register


MovieMaster
Search bar
Account button
Filters
Highest rated movies
Movie 1
Movie 2
Movie 3
Movie 4


←Back
MovieMaster
Account button
Title of the Movie
Add to favorites
Picture of the movie
Description of the movie
Rating over 5
Reviews/Discussion
User 1
User 2
User 3
Spoiler safe
No spoiler safe

Test Results (Grace):

For the test cases for our project:

- Default case: This test case is implemented to make sure that the server is running and will return a success message if it is working.
- Adding User: This test case is used to make sure that we are able to add a user when they are registering on our website. This is extremely important because if you are unable to add a new user then you aren't able to create accounts and then the user cannot move past the register page because you need to be able to log in in order to use any other part of the website.
- Logout: Make sure that the user is able to log out fully. This is important for security and user experience making sure that their account information and past activity on their account is safe.
- Verify the login page was rendered correctly: This makes sure that the login page is rendered correctly and the login page is displayed and is able to be accessed by all users. This allows the user to not have any issues when trying to login.
- Logging in with Failed Credentials: This tests the behavior that the system will have when a user tries to login with credentials that do not exist or are invalid. This makes sure that the system knows what to do when incidents arise like this.

Observations:
These observations are from the initial testing of the website and seeing how the user interacts with the website.

- When we had friends test out our application all users were able to make the switch from the login page to the register page. If users did not create a password that met the criteria for the guidelines then it would not be accepted because of regex. When users were prompted with this they were able to understand and change the requirements to create an account. They then were able to navigate through to the login page and successfully login. Once in the account the users were able to see a carousel that displayed movies, most of the time the users clicked on the carousel and floated between the given options. They then would look at the options they had on the drop down menu such as add movies, favorite movies, etc. Moving from this we would change this so that all the options are displayed in the beginning and not make the user have to click on the button to then display. Most of the behavior with the users was very standard; they would explore the application and click on all of the functions to see what each of them did. Many if not all users did not try to change their email when in account settings but it was a nice feature to implement for when it will be needed. The behavior displayed by the users did not deviate from what we expected. We primarily expected the users to click on

buttons and navigate through the website. Moving to the future, we may tell a few of our testers.

Deployment (Guillermo):

http://recitation-12-team-01.eastus.cloudapp.azure.com:3000/