Lab 6 - The Boys

Platform: **1**
Platform: 1
Programming Languages: 1

Feature List: **1**
Important features: 1
Important Fields to Search: 2
User test cases: 2
GUI Design: 3
Taskboard: 6
Sprint - 1 6
Sprint - 2 7
Sprint - 3 7
Sprint - 4 7
Sprint - 5 8

[Anime Recommendations Database]

Data we have:

1. Anime names
2. Genre
3. Type
4. Ratings
5. Number of episodes
6. Members

Platform:

Discord (through a bot)

Programming Languages:

Java - IntelliJ IDEA

Feature List (Questions of Interest):

Important Features:

1. Sort alphabetically.

2. Sort based on the ratings.
3. Sort based on most watched.
4. Sort based on genre(alphabetically or selected genre)
5. Sort by number of episodes
6. Sort by type (TV, OVA, Movie. etc)
7. Add new anime to the CSV
8. Update ratings

Important Fields to Search:

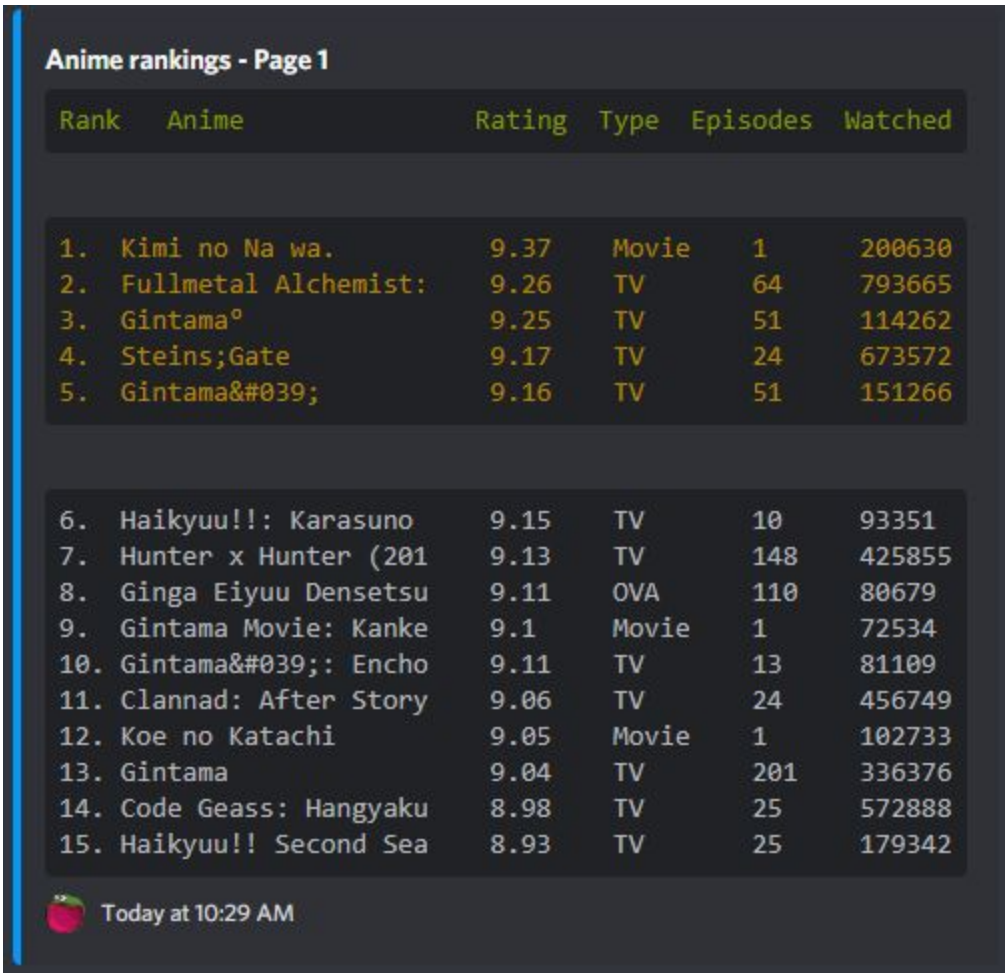
- Show/movie title
- Search by show/movie genre
- Search ratings

User test cases:

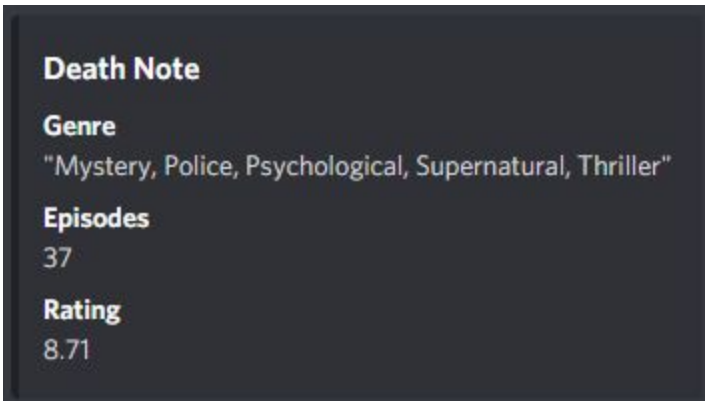
- Search using text input
    - As a user, I want to search for specific genre, rating, or anime.
    - Expected output: It will list out what we searched in a list.
- Display results sorted by x
    - As a user, I want to see the results of the csv file sorted by most/least watched, most/least episodes, highest/lowest rating, or by name.
    - Expected output: Show a list of the items as expected by the selected sort.
- Using react buttons to navigate through sections
    - As a user, I want to go to the next page of the list.
    - Expected output: Clicking the buttons navigates to the next page or the previous page of the list.
- Remove anime from the list
    - As a user, I want to remove a specific anime.
    - Expected output: Removing an anime will completely remove it from the list and from the csv file.
- Update anime statistics
    - As a user, I want to update specific parameters for x anime.
    - Expected output: Specific stats of the anime will change to whatever we specify them to change to, aka update the values.
- Analytics to show
    - As a user, I want to see pretty graphs of the contents
    - Expected output: It will show a picture graph of the stats on the Anime objects
- Analytics for the top10 anime
    - As a user, I want to see how the top 10 anime compare to each other in ratings, episodes, and number of people that have watched it.

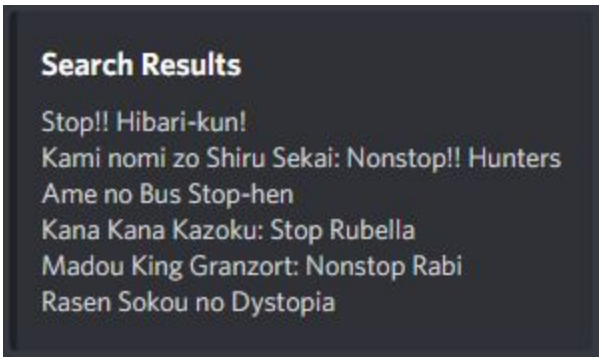
    - Expected output: It will show a picture graph of the stats and how they compare to each other.
- Favorite command
    - Expected output: As a user, I want to enter an anime title to add it in a favorite list.
- Favorite remove command
    - Expected output: As a user, I want to enter an anime title to remove it from a favorite list.
- Favorite export command
    - Expected output: As a user, I want to export a favorite list to import in another instance.

GUI Design:

- Showing list of anime
    - Expected output: A message that has a complete list in order ranked from best to worst, and it shows the values in a column.
    - 

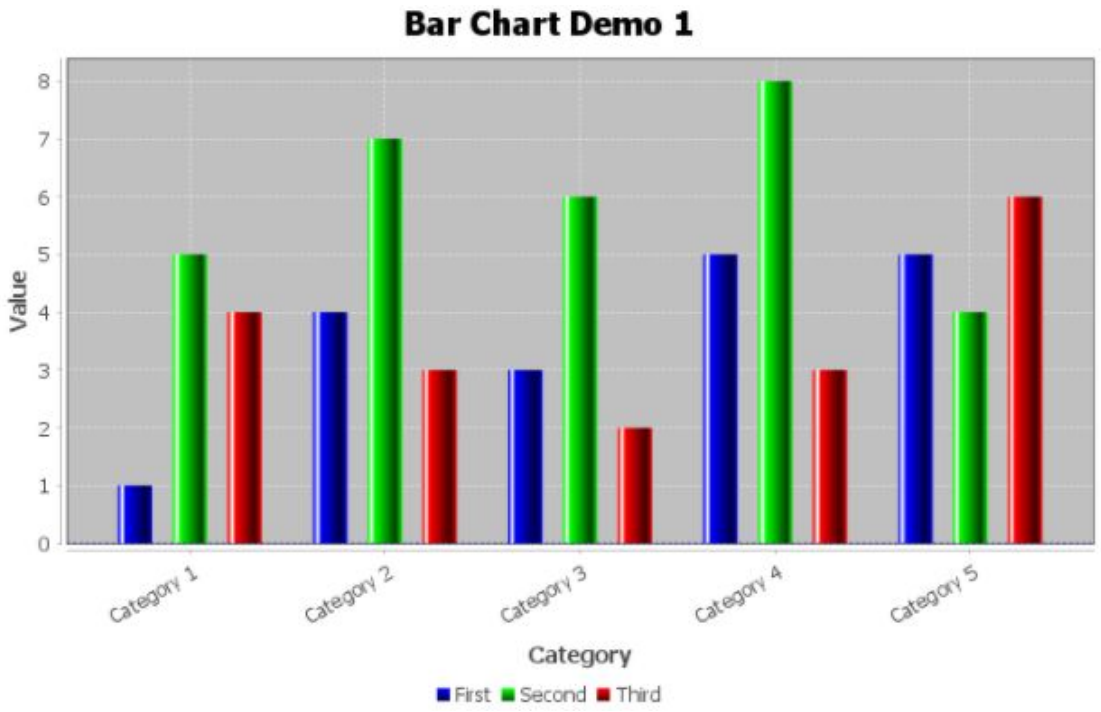
- Search shows a list of the results or matching title
    - Expected output: A message that shows a list of the search results.

- 

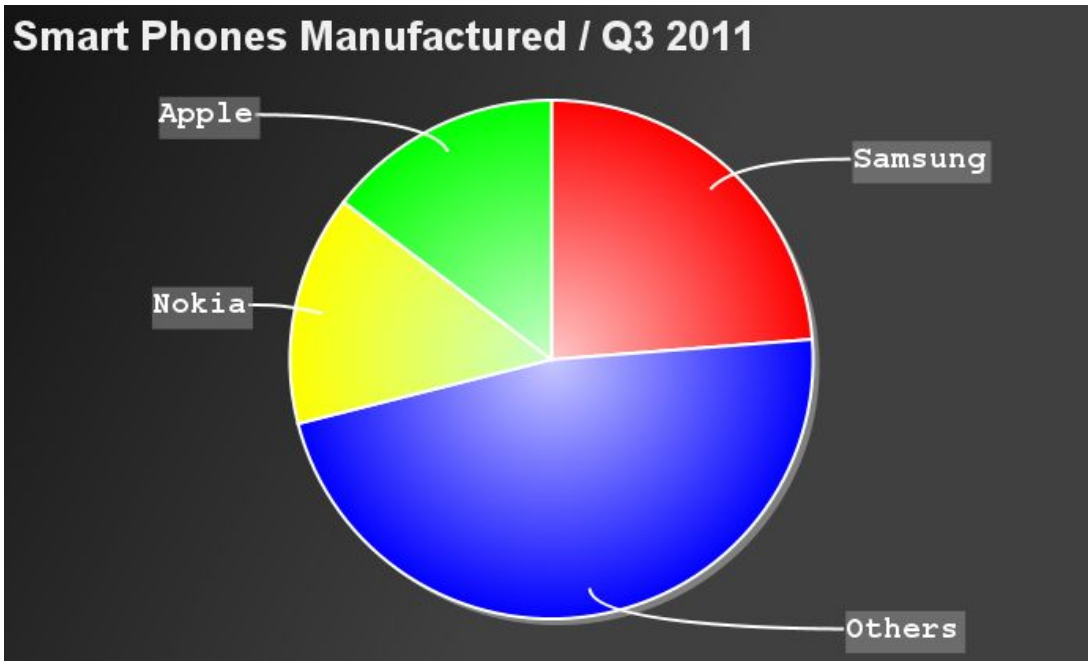
- 

- Analytics displayed
  - Expected output: A message will be displayed explaining what you are seeing while also showing a picture of the stats.

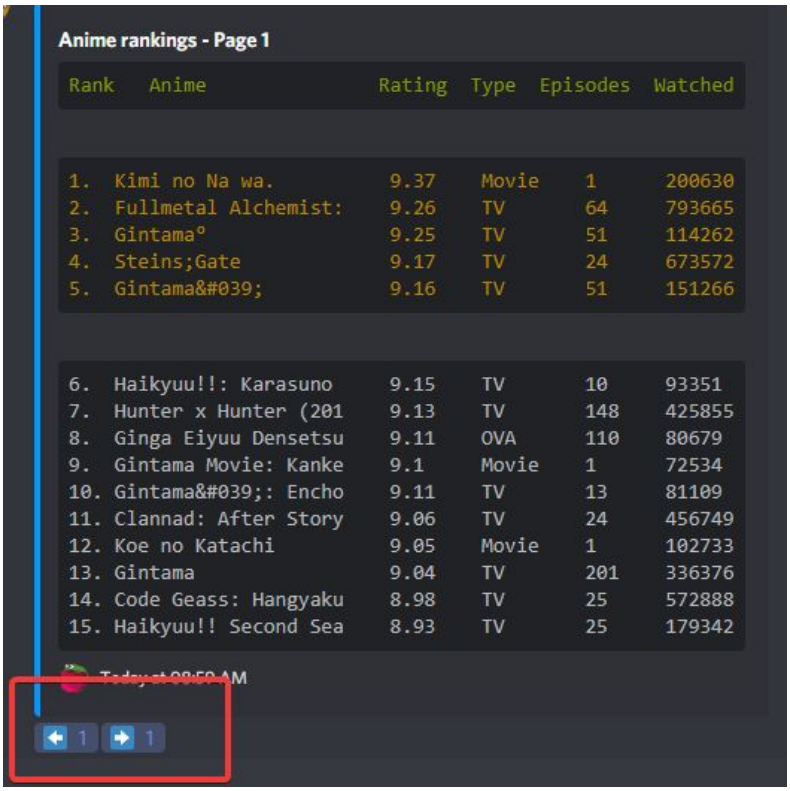
  - This is an example of what the graphs should kind of look like



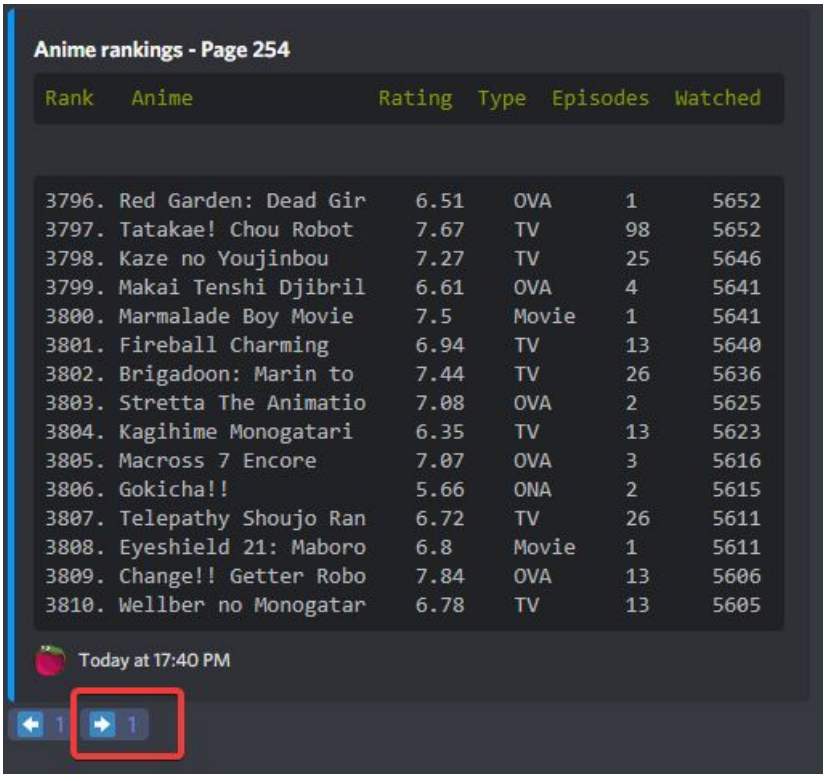
- This is an example of what the pie chart should kind of look like

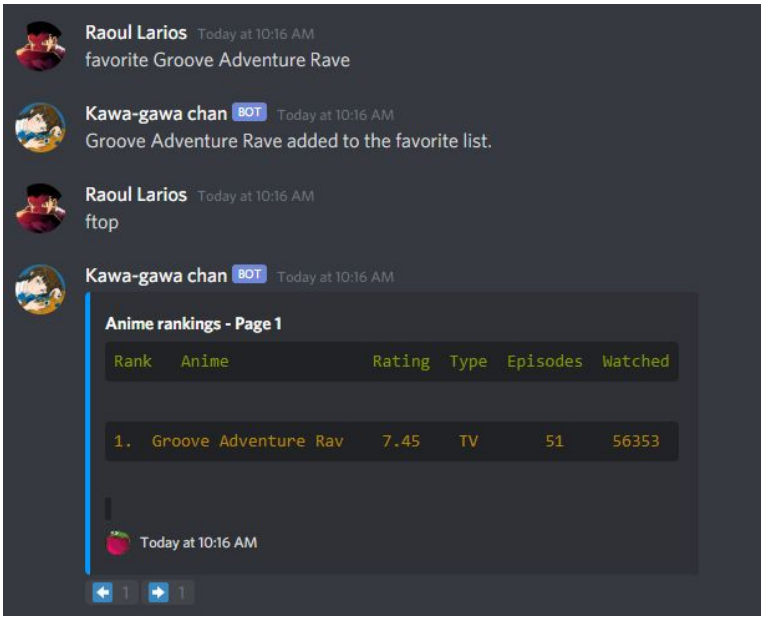


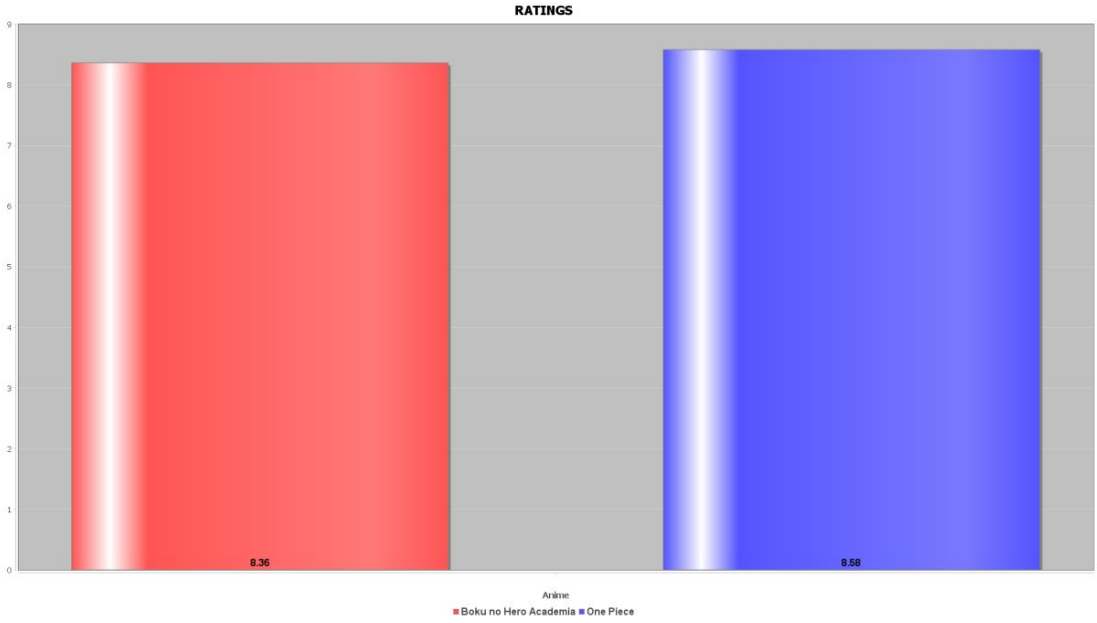
- Clickable emoji under messages
  - Expected output: Emoji will show up on messages and they will be clickable and have some sort of interaction with the messages.



- Switch to next page when emoji is clicked

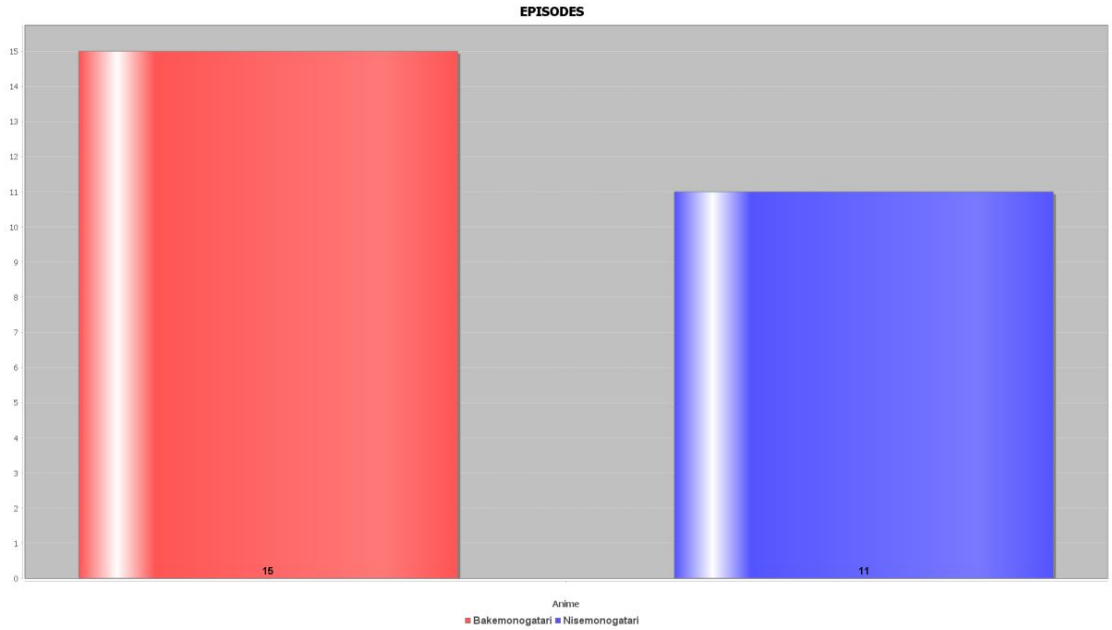


- Showing list of favorited anime
    - Expected output: A message that has a complete list of anime favorited
    - 

- Implementation of bar graph comparison. Input: rgraph boku no hero academia & one piece
    - Expected output: bar graph with the rating comparison.
    - 

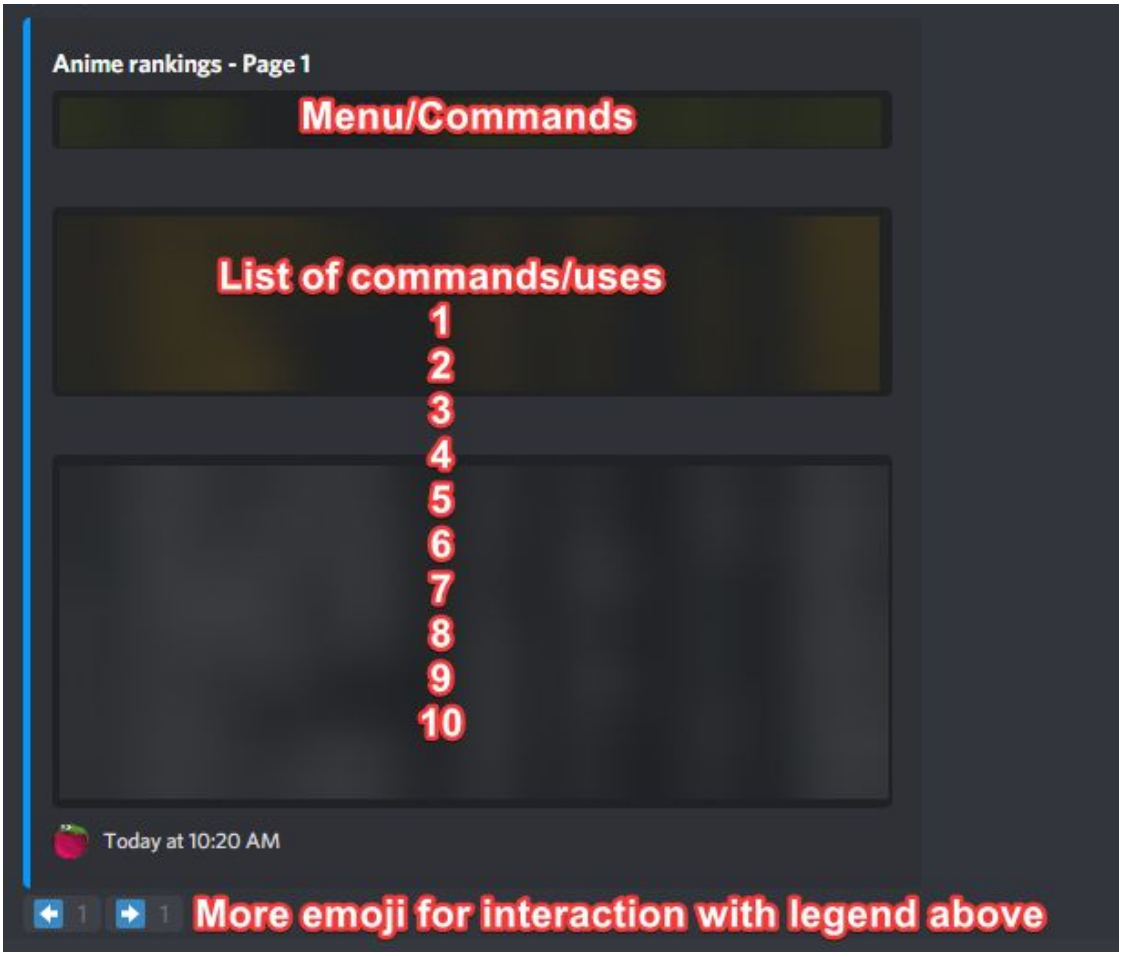
- Input: egraph bakemonogatari & nisemonogatari

- Expected output: The number of episodes shown for both animes.



- Input: Typing “random” into comment showcases a random anime from the list for the user to watch
  - Expected output: Message displaying a random anime.

- Menu with commands and emotes guide

- Expected output: Menu will pop up with relevant information towards using the bot and also have clickable emoji for quick action towards displaying specific functions of the bot.
  - 

-

Taskboard:

Sprint - 1

- Made bot
  - Group work
- Successfully connected bot to discord
  - Group work
- Successfully logged and replied to test command
  - Group work

Sprint - 2

- Parser for CSV file

    - Arturo
- Display results in sorted order of most watched
    - Mario
- Made Anime class to create anime objects to be used to identify
    - Brian

Sprint - 3

- Made save feature to store sorted lists to new csv file
    - Brian
- Made Rating class to create rating objects to be used to identify
    - Brian
- Created command prompts to bring up top watched, and top rated. i.e. typing in “topw” or “topr” in the message box
    - Mario
- Created a command prompt to backup the list to a written file
    - Mario
- Implemented Search function. When using “$” to list all the animes with the word and “!” which shows the anime and the information within the anime
    - Arturo
- Implemented inserting and deleting Anime objects to and from the list, which then are saved to the csv
    - Raoul
- Implemented updating values for Anime objects (Columns with editable content: episode, rating, and watched)
    - Raoul

Sprint - 4

- First of all, research analytics and find ways to display graphs
    - Raoul
- Get accustomed to displaying analytics and start brainstorming analytics to present
    - Raoul
- Show graph of stats put together to compare the top10
    - Arturo
- Created ratingList class for rating.csv file
    - Brian
- Tidied up code
    - Brian
- Implemented more functionality to updating the Anime Objects and cleaned up code (Editable columns are now: Episodes, ratings, watched, types, and genres)

    - Mario

Sprint - 5

- Implemented interactive emoji buttons to browse through features
    - Raoul
- Implemented favorite/bookmark feature with add, delete, export, and sort function.
    - Brian
- Implemented a “random” command to select a random anime for the user to watch
    - Mario
- Implemented a specific use case for rating and episode. bar graph for comparison. Can input multiple animes to compare using “&”. Using “rgraph” before the image shows the ratings of the anime. “egraph” compares the number of episodes.
    - Arturo

Sprint - 6

- Add comments to elaborate on code segments for understandability
- Make anime update in favorite and anime list if it is located in both lists
- Update general look on GUI with menu and add more functionality with emoji
- Show analytics of the distribution of ratings on specific anime
- Show analytics for ratings across animes
- Show pie graph on different genres