Team Streams (10) project proposal

Project: Yomiru

Simon Fraser University

CMPT 276: Introduction to software engineering

Parsa Rajabi

Submitted to Vincent Hu

Date

13 June 2025

Yadav Singh (Project Manager)

Eric Stad (JavaScript Development lead)

Jaskirat Kaur (Design/UX lead)

Karan Passi (Quality Assurance Tester)

https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams

Overview

Intention:

The idea for this project came from most of our team being anime fans with a common issue: What to watch or read (manga). Since the covid era the anime world has been on the rise with new releases coming out almost daily. There is an overload, that is further worsened by random reels on social media showing us a clip with no titles given. So that's where our project comes in. Yomiru is an all-in-one anime suggestion tool based on the user's tastes as well to find that anime that has no title by just uploading a screenshot to our website and getting a description of the anime and links on where to watch it. Thus, making the decision what to watch easier and cutting down the hours gone by of wondering 'I don't know what to watch'.

Potential users:

1) **New to anime user – Discovering new anime or manga based on their general preferences given i.e.) genre through the suggestion interface we intend to implement.**
2) **Hardcore/ long time anime or manga fan- This user would use our webpage to find new anime that may interest them, that they may have overlooked or not seen before. This would also be achieved through the suggestion feature of the webpage.**
3) **Manga purist – This user can look for the latest releases of their favorite niche manga's keeping up to date as well as finding a place to get the latest copy. Or even find where that obscure manga panel screenshots originates from using our anime finder feature powered by Trace Moe.**
4) **Anime only viewer- User that only watches the most popular mainstream anime, they can find this out by looking at the most watched for the week. This need to keep up with the latest mainstream can be achieved by looking our top 10 dynamic anime most watched board.**

5) **The casual fan – This user is a person that watches anime occasionally, this site would keep them up to date and help them commit to watching a new anime by cutting down the work needed to decide on what to watch. Our suggestion feature comes in useful here for this purpose, and the Trace Moe feature can help with identifying the random unnamed anime from social media that brings them back into anime.**

Personas:

1. **New to anime user: Ian – Ian is a 38-year-old dad, whose son has recently been developing an interest in anime, Ian wants to relate with his son by getting into anime.**
2. **Hardcore/ long time anime/ manga fan: Hino is a 34-year-old software engineer who lives for anime/manga and has a huge collection of manga and anime DVDs. She uses Yomiru to search for super hidden and underrated manga/anime that she has never read/watched.**
3. **Manga purist (user exclusively reads manga): Chloe- 16-year-old Chloe is an old soul she lives for a fresh copy of manga on release which is why she uses Yomiru to find out when the latest chapter drops and where she can buy a copy.**
4. **Anime only enthusiast: Keshav- 27-year-old Keshav is a busy engineer with friends who are super into anime, he likes anime but lacks the time to get into the deep niches, he uses Yomiru to find out what's the latest trending anime to watch in his spare time to keep up with his friend group.**
5. **Casual fan (watches both manga and anime go with the mood on watch to watch): Sarah is a 20-year-old college student who engages in anime and manga occasionally. She uses the Yomiru swiping feature from time to time to choose anime and manga based off of what she is in the mood for.**

APIs:

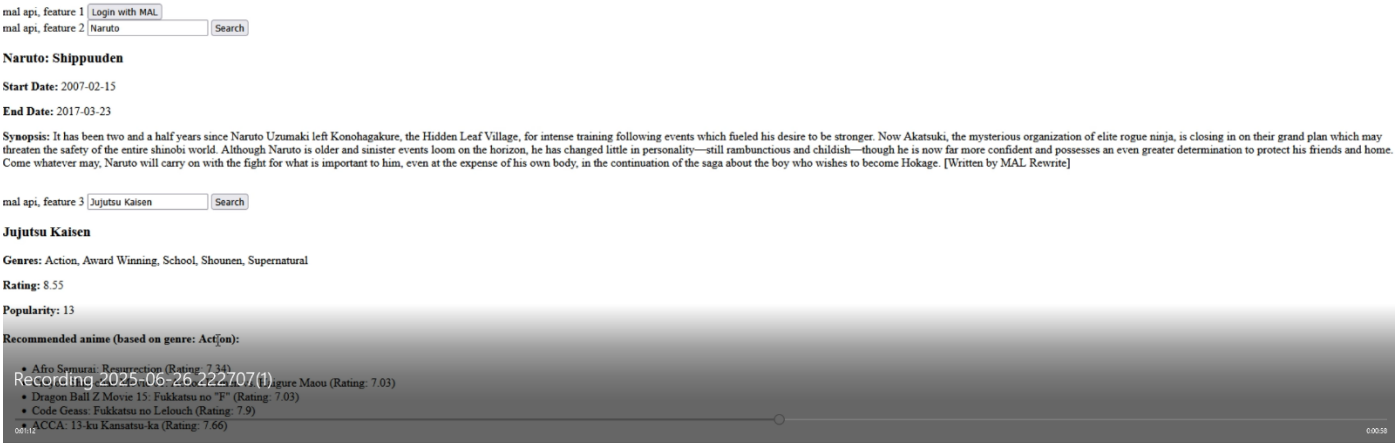
Primary

1. Myanimelist API (MAL)

   The MyAnimeList (MAL) API gives developers access to a huge manga/anime database and some user features. The list of manga/anime is over 100 000+, and some features include recommendations, user integration with their profile, providing ratings and reviews, etc. This API is free.

Features:

1. Access to existing user accounts (for adding anime suggestions to their watchlist on myanimelist).
2. Access to release date/ information from MAL API.
3. Getting information about an anime/manga (genre etc) and suggests anime/ manga to a user based on preferences they give for the session.



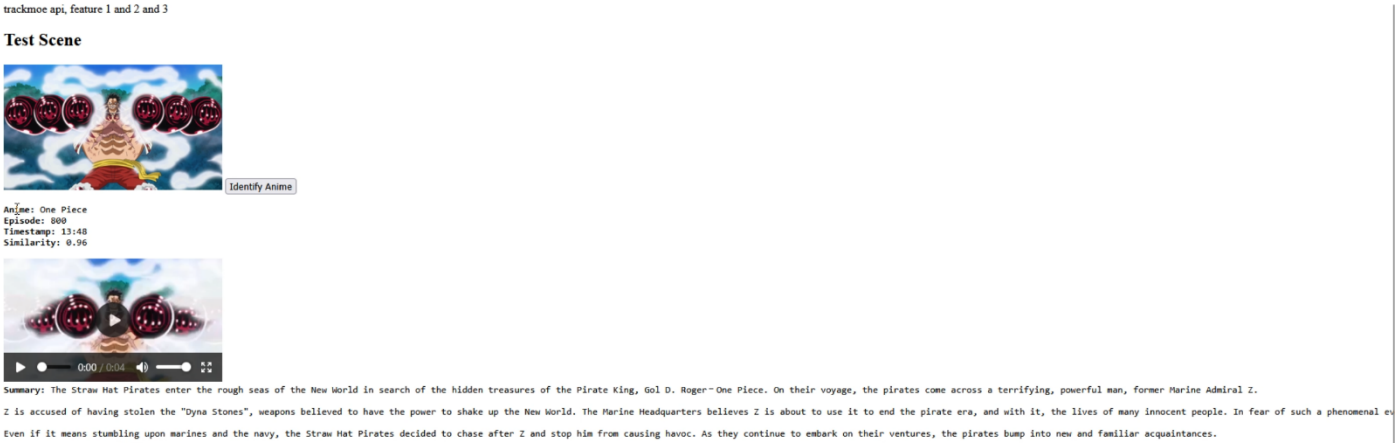
2. Trace Moe (API)

   The Trace Moe API is a reverse image search engine for anime scenes. Trace Moe API achieves this by taking a picture and identifying what anime it is from, episode number, timestamp, providing a clip review, etc. Combined with MAL, it provides a summary of the anime from a specific scene that it identifies. This API is free.

Features:

1. Given an image from an episode of an anime – returns the anime the image is from.
2. Gives the exact time stamped clip from the episode.

3. Combining with MAL API to find the anime the scene is from and using the data from MAL give a summary/ synopsis of the anime.



Secondary/back-ups

1. Gemini

   The Gemini API is a conversational AI where you can query to generative information for you. In our case, Gemini AI can give character info given a name/image of an anime, and provide a Wiki link for providing further information. Additionally, it can assist with searching and suggesting anime. This API is free.

Features:

1. In conjunction with Trace Moe, finds the character and prompts to takes the user to character wiki page.
2. Used as a scraper to search the web for the best places to watch the anime chosen (either from the scene finder or from the recommendation page)
3. Creates summaries for animes/ manga suggested.

2. Studio Ghibli API

The Studio Ghibli API provides information on Studio Ghibli films, characters, and locations. Ghibli films are legendary in the world of anime, and can help enrich our app by showing voice actors, locations, fun facts, etc. for Ghibli content. This API is free. (edited)

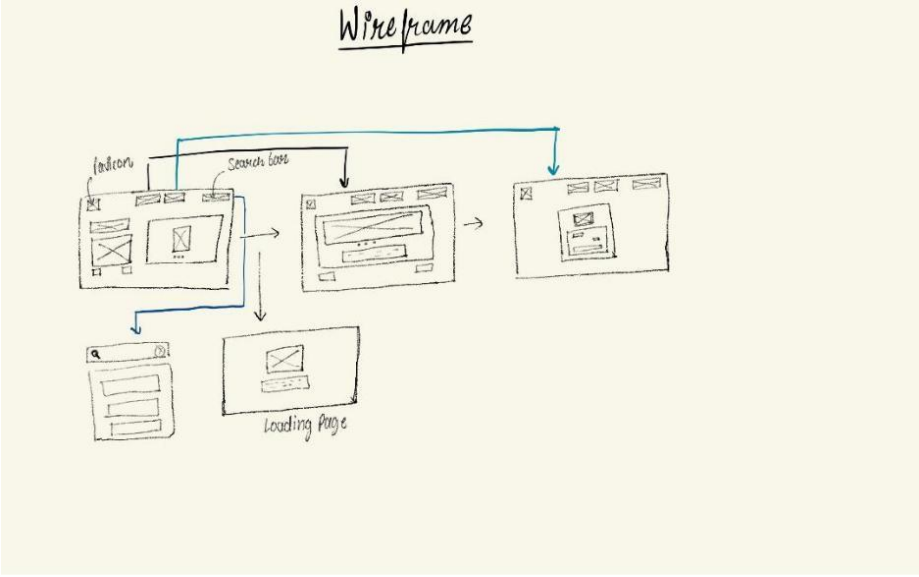
Features: (All for specifically Ghibli studios)

1. Character and location data for characters within the Ghibli universe
2. Insights into the inner workings/ behind the scenes related to Ghibli productions
3. Use /people and attach fun facts like voice actors (can be scraped from IMBD or Wikipedia.)

#	API	Feature Description	User Story
1	Gemini	Scrape the web for the best places to watch the anime (from scene finder or recommendations)	*As a busy viewer, I want to know where to watch anime immediately, so I don't waste time searching.*
2	Trace Moe	Identify an anime from an uploaded screenshot	*As a casual fan, I want to upload a screenshot and find the anime name so I can watch the anime from the screen.*
3	MyAnimeList	Add anime suggestions to a user's watchlist	*As a new anime fan, I want to save a suggested anime to my MAL account so I can watch it later.*

#	API	Feature Description	User Story
4	Gemini	Generate summaries for suggested anime/manga	*As a casual viewer, I want a short summary of suggested anime, so I know if I'll enjoy it.*
5	Trace Moe + MAL	Identify anime from image and use MAL to display full details (summary, rating, genre, etc.)	*As a hardcore fan, I want full details from a screenshot so I can dive into a new anime quickly.*
6	MyAnimeList	Access release dates and information about anime/manga	*As a manga purist, I want to know when a new chapter is coming out so I can be up to date.*

Low- fidelity story board:



Front-end technology stack used:

- React- We plan on using react as the base framework, due to the extensive libraries on offer as well as the ease of use/ familiarity team members have with the framework. We also kept testability in mind and the ease and speed react based frameworks can be tested.
- Tailwind- Tailwind was chosen as the main styling tool, due to the flexibility and control over UI design it offers.
- GitHub issues for CI/CD pipeline – Group members are familiar with it and it increases
-
- convenience
- Testing platforms
    - Jest – due to the ease of testing JavaScript competent by component
    - React testing library – JavaScript and UI testing chosen for the proximity of it to the React framework
    - Cypress – For the complete end-to-end testing of the project, we choose this due to it being the most comprehensive for the purpose of the project while maintaining an ease-of-use characteristic.

-Design

- Low fidelity – mix of physical white boards(planning), and digital stylus and tablets for finalization, chosen for the ease and comfort of the teams' planners and designers.
- High fidelity – Figma was chosen due to the easy to access nature remotely by team members as well as all team members being comfortable with it.

- Web- hosting

- Netlify, we chose this as the hosting platform due it is fitting our needs for posting a web project with not a lot if any server-side resources. It is also free to use.

offer as well as the ease of use/ familiarity team members have with the framework. We also kept testability in mind and the ease and speed react based frameworks can be tested.

- Tailwind- Tailwind was chosen as the main styling tool, due to the flexibility and control over UI design it offers.
- GitHub issues for CI/CD pipeline – Group members are familiar with it and it increases
-
- convenience
- Testing platforms
    - Jest – due to the ease of testing JavaScript competent by component
    - React testing library – JavaScript and UI testing chosen for the proximity of it to the React framework
    - Cypress – For the complete end-to-end testing of the project, we choose this due to it being the most comprehensive for the purpose of the project while maintaining an ease-of-use characteristic.

-Design

- Low fidelity – mix of physical white boards(planning), and digital stylus and tablets for finalization, chosen for the ease and comfort of the teams' planners and designers.
- High fidelity – Figma was chosen due to the easy to access nature remotely by team members as well as all team members being comfortable with it.

- Web- hosting

- Netlify, we chose this as the hosting platform due it is fitting our needs for posting a web project with not a lot if any server-side resources. It is also free to us.

Algorithm - We plan to implement an algorithm where there are two lists. The first list includes similar anime to what the user has liked earlier by using the MAL API and filtering it out based on one or two tags. Another list will contain a randomized list of trending and genre diverse anime. A display queue will be created to randomly alternate entries from the first and second lists so that the anime recommendations are not just an echo chamber of the first anime that the user has liked. Initially for the first anime that is displayed the anime will be chosen from the second list.