

# Sprint-3 Artifact

**Members:** Michelle Chu, James Ooi, Jeng-Rung Tu, Karsten Widjanarko, Yuteng Wu

## Features to implement in next sprint:

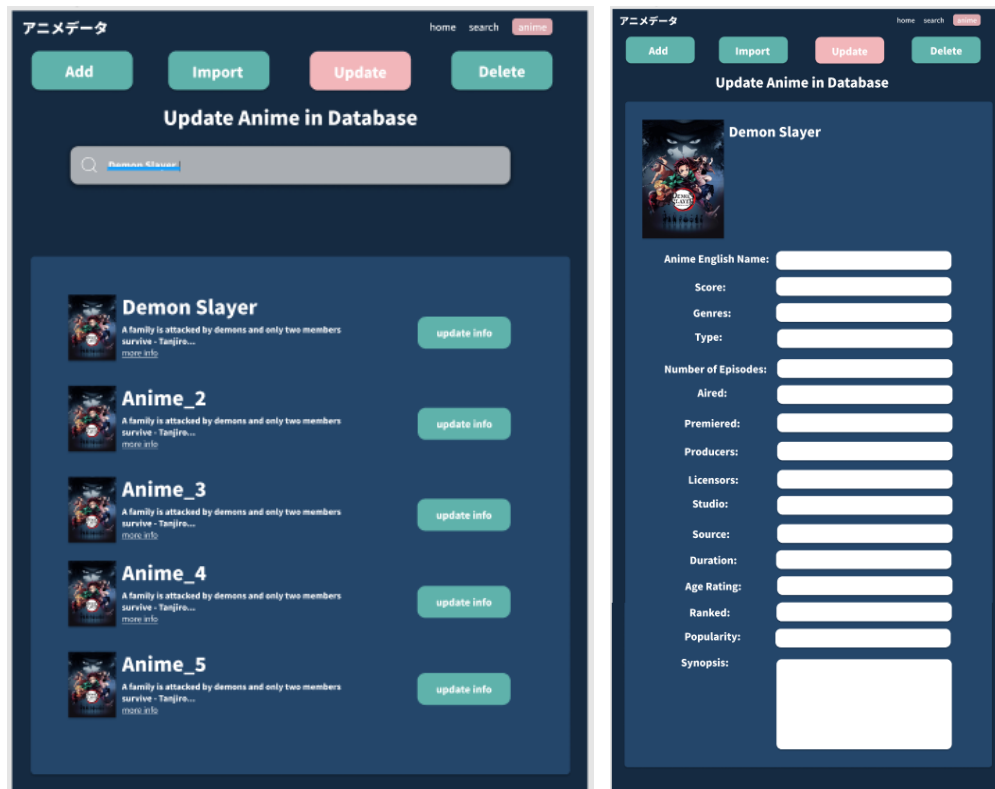
- **Feature 1:** As a user, I want to insert a new anime and any possible information I know about that anime so that the database can stay current.
- **Feature 2:** As a user, I want to delete an anime from the database so that I can get rid of non-existing anime.
- **Feature 3:** As a user, I want to update any information changes for an existing anime.
- **Feature 4:** As a user, I want to import any new changes to the database into a new csv file so that I can save my changes.
- **Feature 5:** As a user, I want to backup the current csv file so that when I make any changes, I can revert back to a previous version of the database.

## GUI Design:

**Insert And Delete:** Note that the actual web pages will be cut off, but the user will be able to scroll down.

The image displays two mobile application mockups for an anime database. The left mockup is titled 'アニメデータ' (Anime Data) and features a navigation bar with 'home', 'search', and 'anime' tabs. Below the navigation bar are four buttons: 'Add', 'Import', 'Update', and 'Delete'. The main section is titled 'Add Anime to Database' and contains a form with the following fields: Anime English Name, Score, Genres, Type, Number of Episodes, Aired, Premiered, Producers, Licensors, Studio, Source, Duration, Age Rating, Ranked, Popularity, and Synopsis. There is also an 'Add Anime Image' section with an 'upload image' button. The right mockup is also titled 'アニメデータ' and has the same navigation bar and buttons. The main section is titled 'Delete Anime in Database' and features a search bar. Below the search bar is a list of anime entries, each with a thumbnail, title, description, and a 'remove data' button. The entries are: Demon Slayer, Anime\_2, Anime\_3, Anime\_4, and Anime\_5.

**Update:** Note that the actual web pages will be cut off, but the user will be able to scroll down. When user clicks on update info, current selection will expand to input info.



**Import:**

**Backup:**

## User Test Cases

- **Feature 1 Test Cases (Insert):** As a user, I want to insert a new anime and any possible information I know about that anime so that the database can stay current.
  - **Test case 1:** As a user, in the anime page, I click on the “Add” button.
    - **Correct Output:** The website displays a table with all the possible anime information I can add. On top, there is a trash can button in case I don’t want to add the anime.
  - **Test case 2:** As a user, I fill out all the information about an anime except the anime name.

- Correct Output: The website does not allow me to add the listing to the database because I don't have an anime name.
  - **Test case 3**: As a user, I only fill out the anime name and press add.
    - Correct Output: The website displays the new anime I added, but all the other fields will be set to default values.
- **Feature 2 Test Cases (Delete)**: As a user, I want to delete an anime from the database so that I can get rid of non-existing anime.
  - **Test case 1**: As a user, in the anime page, I click on the "Delete" button.
    - Correct Output: The website displays a search bar where I can search up the existing anime in the database.
  - **Test case 2**: As a user, I find the anime I want to delete and press the "Remove Data" button.
    - Correct Output: The website no longer displays the selected anime. Also, that anime's row in the database is deleted.
- **Feature 3 Test Cases (Update)**: As a user, I want to update any information changes for an existing anime.
  - **Test case 1**: As a user, in the anime page, I click on the "Update" button.
    - Correct Output: The website displays a search bar where I can search up the existing anime in the database.
  - **Test case 2**: As a user, I find the anime I want to update and press the "Update Info" button.
    - Correct Output: The website displays all the information fields for that anime.
  - **Test case 3**: As a user, I fill out the "Genre" and "Name" fields for an anime.
    - Correct Output: The website updates the genre and name for that anime, and the rest of the existing information fields for that anime remains the same.
- **Feature 4 Test Cases (Import)**: As a user, I want to import any new changes to the database into a new csv file so that I can save my changes.
  - **Test case 1**: As a user, I edited an existing anime record and just inserted a new anime record and deleted an existing anime record.
    - Correct Output: After making these changes, the webpage reflects the edits I just made by showing me a redo button.
  - **Test case 2**: As a user, I refresh the page after making changes.

- Correct Output: After I refresh the page, my edits should stay in the webpage I am looking at.
- **Feature 5 Test Cases**: As a user, I want to backup the current csv file so that when I make any changes, I can revert back to a previous version of the database.
  - **Test case 1**: As a user, I edited an existing anime record and just inserted a new anime record and deleted an existing anime record.
    - Correct Output: After making these changes, the webpage reflects the edits I just made.
  - **Test case 2**: As a user, I refresh the page after making changes.
    - Correct Output: After I refresh the page, my edits should stay in the webpage I am looking at.
  - **Test case 3**: As a user, I'm regretting the edits I made and wish to revert back to the oldest version of data so I click the revert changes button on the menu bar
    - Correct Output: After I revert changes, my web page reflects the original backup data I had before I made any changes and displays a undo button.

- **TODO List**

**Done list of last sprint:**

- Fix the Features List and message TA on Slack.
  - [finished by Michelle Chu and verified by everyone]
- Design GUI for home page.
  - [finished by James Ooi and verified by everyone]
- Frontend program for search feature.
  - [finished by James Ooi and Karsten Widjanarko verified by everyone]
- Backend program for search feature.
  - [finished by Michelle Chu and verified by everyone]
- Backend program to read the data from csv.
  - [finished by Jeng-Rung Tu and Yuteng Wu and verified by everyone]
- Backend program to convert the csv data to json object.
  - [finished by Jeng-Rung Tu and Yuteng Wu and verified by everyone]

**ToDo task list for the next sprint:**

- Frontend program for insert function.
- Frontend program for update function.

- Frontend program for delete function.
- Frontend program for studio search.
- Frontend program for producer search.
- Frontend program for score search.
- Backend program for insert function.
- Backend program for update function.
- Backend program for delete function.
- Backend program for backup function.
- Backend program for import function.
- Backend program for studio search.
- Backend program for producer search.
- Backend program for score search.