WiSe 2023/24

Practical Experience in Digital Media II

App Development

Sarah Haq

“WeebNote”

Phuong Anh Bui, 3044825, Phuong.A.Bui@stud.leuphana.de

Jessica Maier, 3044981, jessica.maier@stud.leuphana.de

Our primary goal was to program an online application that helps you pick a random show to watch, when you don’t know what to watch and where to look. This generator should pick a random show for you as inspiration. If you like it, you can add it to a to-watch-list and communicate your intent to watch it to your friends. This way, it could also serve as a bonding experience and overall growing a community that enjoys the same thing we do: in this case anime shows, which are mainly japanese animated series. There are more than 10 million people worldwide paying for the largest streaming service that focuses on anime, so there are many potential fans and users of all age groups. So the main goals are to let users find entertainment quick and easy and connect with friends over their favorite shows. Our first user-feedback was primarily positive and motivated us to implement user profiles. The working title of our app is “WeebNote”.

The first user-contact is the login page, where we use the form widget streamlit provides. As this is an MVP we only check if the username is correct by comparing the typed in username to stored usernames in a deta database that we inserted: To test it, we used “ash” and “misty”. In general, we use deta (https://docs.streamlit.io/knowledge-base/tutorials/databases/deta-base), kindly recommended by Sarah Haq, as it is free and easy to connect with our python script to store and access the user data through the functions in our helpers file. It was important to add the secret data key in a .gitignore file to not publish it. Back to the app, there is also some explanatory text on this page, what the app is about and why you should sign up, though the code to sign up is a topic for future development of this app. After logging in, the definition “open_app” from a second python script gets called, where the code is stored independent from the code for the login page. So you get to the welcome page, which greets you with your username to indicate that you are logged in with the right account. With streamlit’s sidebar menu you can switch between pages, every page calls a separate definition that executes the content of that page. In the “Generate Anime” page, we created two tabs with different filters to adjust the randomization of your anime a bit. Tab 1 lets you decide which season and year the anime got published. We actually did not find this function in similar applications, even though it is common in the community to search for anime based on the criteria of season and year. Tab 2 is for including or excluding genres to your search with streamlit’s multiselect widget. When

you generate an anime in either tab, a button to add the title to your anime list appears. In the “My Anime” page you can look at your stored anime. The stored anime are saved in a deta database as a title and image and the username that saved the title, so that the anime corresponds to the user. When removing a show from your anime list by clicking on the “X” button under a show in the application, the entry in the database is deleted. In the page to match friends you can enter a username. It will then tell you through if-statements in the code, whether the user exists, you matched and if you matched, what shows you and your friend have both on your watching lists. The if-statements compare whether the username exists in the database, so the typed in username has to be identical, and then it compares whether there are identical titles on both of the lists of you and your friend. Sarah Haq showed us how to show the results to the user in bullet-points design-wise. The logout button also has its own page in the menu. As we saved our “logged in” status in a session state variable, when the user wants to logout, we set said session state to false and return to the login page. To code our app we mainly used the deta and streamlit documentations.

The genre and season filters use the JIKAN API, which is the unofficial API for myanimelist.com, the biggest and most active anime community and database. The main reason why we have decided to use this API instead of the official myanimelist API, is due to the fact that the official API has very limited functions and number of requests allowed per day. The JIKAN API does have its limitations too and occasionally it may happen that an error occurs, due to the limited number of allowed requests by the API. This error is however easily fixed by simply refreshing the page.

The code used for these functions within our app has been purely written by us, using the official JIKAN documentation. None of the code has therefore been outsourced.

Some limitations at this point were that it proved difficult to find a way to let two users communicate. In the future, we would love to implement something like a chat or contact function, so that they connect even more through the app and their shows. Another thing is that our iOs testers told us that the sidebar menu seems to disappear on mobile devices. We could recreate the error and find a similar complaint on the internet (https://github.com/streamlit/streamlit/issues/4907), so we assume that it is a bug from streamlit. We were also short in time to implement the idea that users can upload a profile picture, but it is possible with code from Sarah Haq's class, so that is also noted for future development. When not working with html, streamlit also has limited design options, but for our purposes, we are happy with the result design-wise. This was a fun and creative way to improve our programming skills with python this semester.

User feedback

After finishing the development of our MVP, we have gathered 6 participants to test out its functionalities and to collect feedback from, for the sake of future improvements.

We firstly asked the participants about their overall experience when navigating the app and how they would rate its user-friendliness. 5 participants viewed the UI as user friendly and described it as “intuitive” and “easy to navigate”. 2 users however mentioned that for the final product they would wish for the UI to be improved to be more user-friendly, but no specific aspects were directly mentioned. A different user suggested an to improve the UI on mobile devices as it was difficult for them to locate the side-bar. While the development of this MVP was mainly focused on desktop users, it could perhaps be useful to expand the instructions on the “welcome” page with additional notes for mobile users.

When asked about specific improvements for the app, 2 users left feedback. One suggestion was to implement a function that would let users generate titles from a span of years, instead of 1 specific year. This is an improvement that could be easily implemented within the seasonal filter. Another suggestion was to add a “ratings” filter in order to be able to get more popular titles during one’s search. As JIKAN API also offers a “ratings” filter, this improvement should also be possible to implement. Lastly, there was also the suggestion to be able to combine the genre and season filters together, instead of having to use them separately. This is however a limitation of the API itself and therefore it is not possible to combine the filtering options as of now. Possibly with a future update this could however change.

Overall, users appreciated the main functions of our app, which was to find new anime and to possibly match with friends. One User wrote, “I *think it's unique and definitely a problem solver for many anime watchers (be they casual or an avid watcher)*”. Other users also mentioned they liked the neutral design and easy and straightforward navigation through the individual pages.

Lastly 3 out of our 6 participants said that they would be highly interested in using the app, with one of them even mentioning that they found a title they would like to watch during their testing. 2 other participants did admit that they were interested, but that they would use the app only after further improvements and more development. Generally, participants seemed to be excited by the prospects of this app and were looking forward to its further improved versions.