

Meetings (please add to if information not included):

2-5:

- Talked about a proposal.
- Formed tentative plan of action.
- Wrote self introductions on proposals.
- Shared tutorials and tools to get started.

2-12:

- Go over tools needed.
- Assign initial tasks.
- Georgia: Create a test app on command line for the merge sort (can be in C++, Python, JS, or whatever) that takes in a text file of entries and then prompts the user with several matchups asking which is better, and then taking that user input and outputting the sorted results into another text file. Email Professor about milestones.
- Shreyak: Design the front end for the userpage, which will be a list of playlists that has under the title of each entry: # of songs, # of matchups generated, and progress bar. Add a plus and minus button that when an entry is highlighted, clicking the minus button will delete the entry. Clicking the plus button will create an entry. For now, populate the entries with template data, Spotify integration will come later.
- John: Help Shreyank with above
- Milestones are due on Feb 19

2-17:

- Talked about collaborative sorting in the future.
- Talked about landing page design, where sort will occur.
- Everything is saved by session for now, not by account.
- Focus on individual sorting. Save collaborative sorting if single user sorting works perfectly in the future.
- Have profiles that save playlists per user (possibly).
- **Make sure you have an issue assigned to you in Github in one of the milestones.**
- Maybe include information on previous matchups to aid in the amount of matches to be made. For instance, if one song is chosen over multiple other songs and is “at the top of the list,” and another song is chosen over that song, that new song “beats” all the songs the previous song that was on top beat. This behavior emulates a priority queue.
- Possibly need to keep information on previous matchups.
- End of sprint 1 is 2-18 (make sure proposal milestone is complete with issues assigned)

3-1:

- About me cards or scrollable events
- Need to solidify theme to help with formatting and inlining
- Need to link to spotify on export/edit playlist page, edit dialogue is merge sort could occur
- Dialogue box, circle for song (with metadata), and choose between songs (left and right)
- Nav bar vs. side bar (both)
- Mirror Spotify's design and format

- Need a way to log in/log out, link with Spotify (Django stores metadata about spotify data)
- Kudos on John's Export/Edit page
- Frontend: Get a theme in place, start integration with backend
- Backend: work on converting merge sort function to python, integration with backend, continue working on API components

3-12

- Working on implementing theme
- Need prototypes for sorting functions in python
- Work on merging repos (frontend and backend)
- Google takes care of username/password, use Django authorization, start out with Django sessions (for testing), maybe improve later on
- Set up webpack (shown on first couple of videos in tutorial)
- Will merge frontend and backend next week with navbar and theme (for now it is frontend built in tutorial, just for testing)
- Authentication expires every hour, need new save method
- Need to reauthenticate again every time playlist modified
- How to save a playlist, create a playlist in Spotify or save in a database?
- Spotify returns a json file as playlist, modify it (sort), and export back to Spotify

3-26

- Jonathan updated Backend readme to run code for frontend
- Need a way to save the progress of the merge sort. Options: create a json file or use iterative merge sort method
- Need to figure out how to merge repos without losing commit history or find a way to submit both for him (but merge to give functionality)
- Sprint 3 due 4/1/2021
- Submit backend as branch to frontend and manage conflicts individually
- Merge function needs to output progress (how many match ups left) and save and quit option (iterative merge sort function for Georgia's sprint)
- Page for sorting or modal for sorting page or dialog with 2 cards with album cover and ability to play song (John's sprint)
- Work on frontend bugs (Shrey sprint)
- Merge repos and manage merge conflicts, debug front end, look at merge sort repo example (Jon's sprint)
- Pull code from where you need it, deal with conflicts later

From class notes:

- Final project report due April 29
- Dates for possible presentation: Apr 13, 15, 20, 22, 27, 29
- Can use initial report as template

4-15

- Song models are changed in database to sort songs
- Each button has API call to backend to sort songs
- Intermediate testing to be noted later along with more extensive testing after merge sort has been implemented
- Move everything to MergeSortMusic CS340 repo after viable model made

4-23

- Go through slides to decide who is presenting what
- Go through mock demonstration for presentation