

Project Proposal

By default, Spotify uses the album art of songs in a playlist to auto-generate the cover of personalized playlists. We would like to create a website that allows users to generate their own customized playlist art by using Spotify Web API and an image generator API, based off of the music within an individual playlist. With our Spotify album art generator, users would be able to create a custom image by modifying various elements including font, color, and other variables. Then the image could be downloaded to the user's device or automatically uploaded to Spotify.

APIs (not all will be used, these are just to show that it's possible):

- Spotify Web API (Get A Playlist, Upload a Custom Playlist Cover Image, Get An Album)
- Fractal image generator <https://github.com/rafgraph/fractal>
- Add Text to Image API <https://cloudinary.com/cookbook/tag/text>
- Sirv Text Overlay API <https://sirv.com/help/articles/dynamic-imaging/text/>
- moesif Text Overlay API <https://textoverimage.moesif.com/>
- List of Image Generator APIs <https://blog.api.rakuten.net/top-image-generation-apis/>
- AI color scheme generator <http://colormind.io/>
- Image Color Extraction API <https://imagga.com/solutions/color-api>
- Npm Image Downloader <https://www.npmjs.com/package/image-downloader>
- IBM Watson Tone analyzer <https://www.ibm.com/watson/services/tone-analyzer/>
- Ganbreeder for combining images <https://github.com/joel-simon/ganbreeder>
- DeepAI text to image API <https://deepai.org/machine-learning-model/text2img>
- Avatar generators API <https://rapidapi.com/search/avatars>
- Trianglify API <https://github.com/qrohlf/trianglify>

Intended Audience: Spotify users

Potential Risks: API call limits, which can be limited by using dummy information to build the website. Use MongoDB to cache json data from API calls.

On the front end, Nick and Spencer will collaborate to execute the design and wireframe of the website splitting up the necessary work to create a seamless and functional website. Because Spencer is working with his CS411 team on the backend, we will prioritize the creative design of the front end CSS and HTML, as well as website infrastructure. He and his team will configure the APIs to retrieve the necessary information from Spotify with OAuth, and any other APIs used in generating the image and making the site operational.

Initial Proposal Ideas:

1. Custom Spotify album art generator. Using Spotify API and an image generator API, generates a custom spotify playlist cover based on your own playlist. Title on the cover will be from the title of the playlist. Could include different variables for color, font, etc.
2. Screentime visualizer - enter your device screentime and see a visualization of the multitude of other things you could be doing with your time. For example, 20 hours of screentime would have equaled 20,000 pushups or 2 books, etc.
3. Waveform scroll thing - see a visual of the sound levels of a video/song before watching it, to prevent surprises in audio level changes for people with sensitive hearing or something. Maybe useful for youtube videos or tiktok or twitter (Chrome Extension)
4. Redlist website - BU short film club - member profiles, projects and events, etc. (uses wordpress so probably not allowed)
5. Student Link refresh - single login at the beginning, better navigation, relevant info shown on homepage, better scheduling system, etc.
6. Chrome extension to turn any page into a book with pages
7. Cocktail Drink Builder - enter your ingredients and get recipes for drinks (and search for a drink and see what ingredients are needed)
8. Integrated zoom website - allows side by side note taking whiteboard option. Allows pausing, rewind, slowdown during live zoom sessions. Upload files and screenshots from zoom... Zoom Livestream API

Questions for jon:

1. Can we do a chrome extension (numbers 2, 3 and 6), which uses HTML, CSS, and JS?
2. Redlist website for BU - is there a conflict hosting a non wordpress website with BU IT services...
3. Can we "redo" the student link even if it isn't entirely functional (I'm guessing Kerberos/DUO is a big hurdle, we wouldn't necessarily be able to present real student data)?