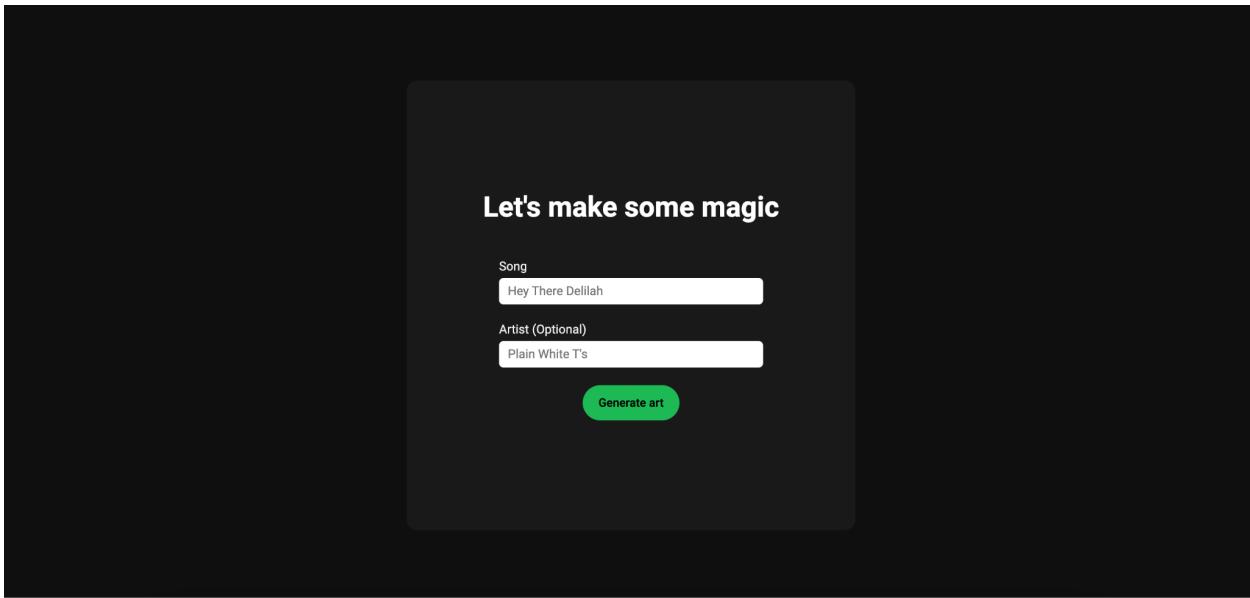


NETS 1500 User Manual

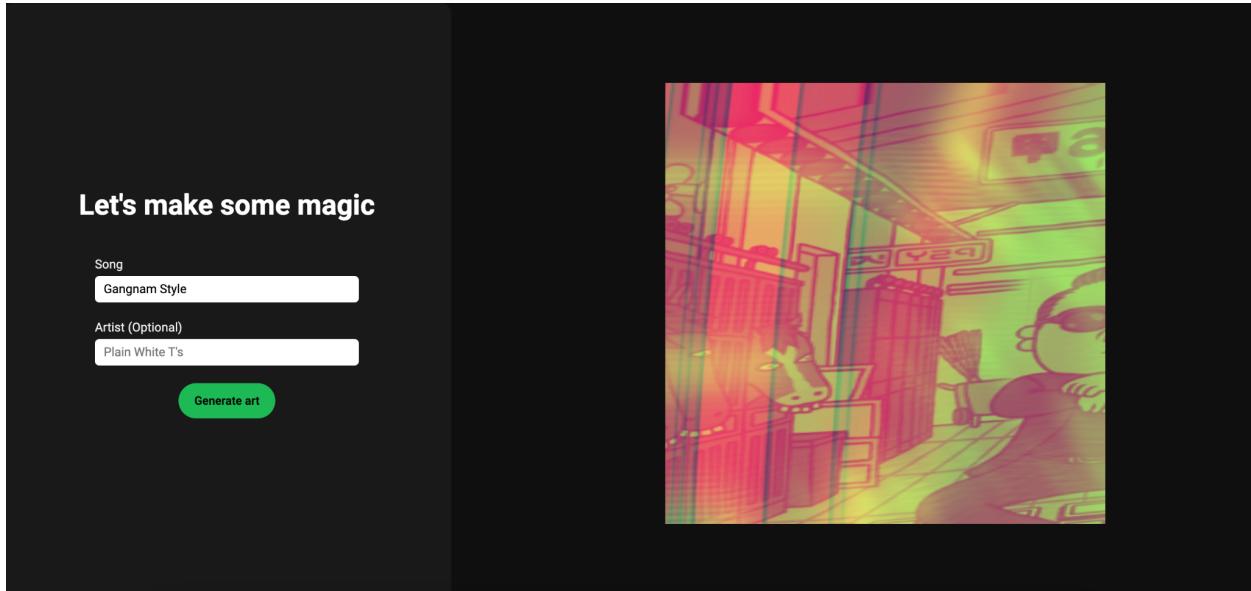
Load <http://127.0.0.1:3000/index.html> into your browser to access our site.

Landing Page

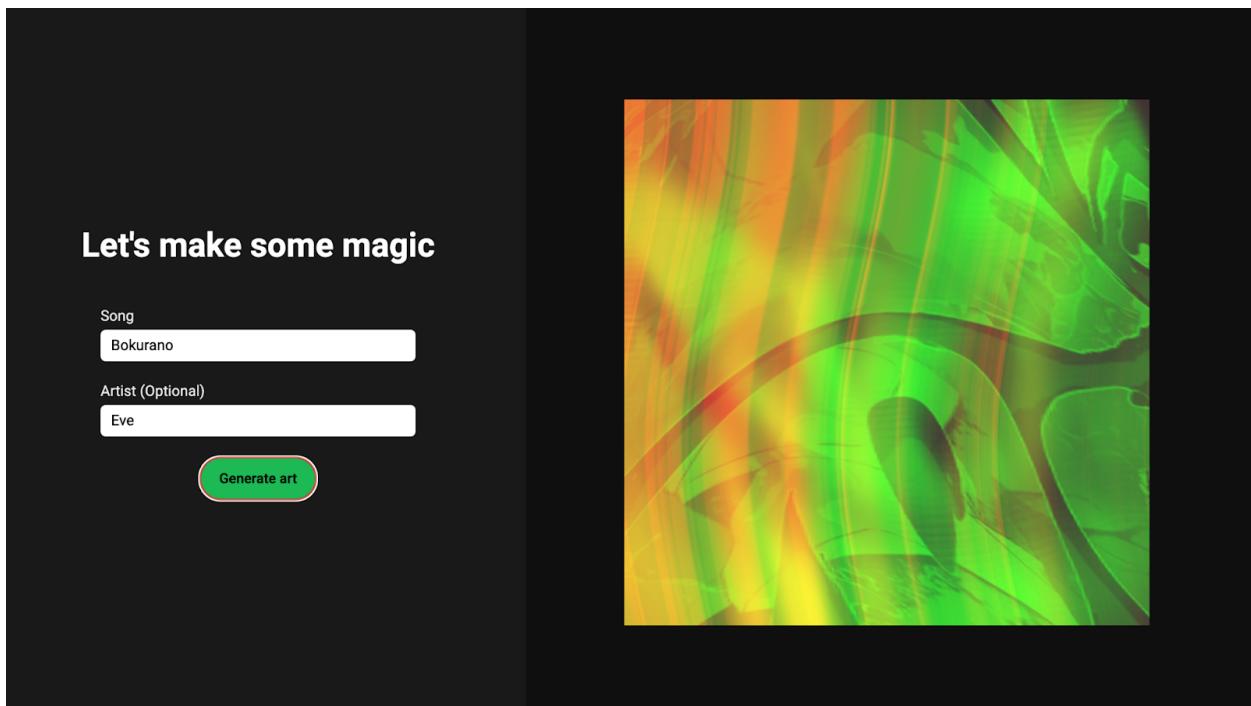


1. Type the name of your song query (e.g. "Gangnam Style", "Hey There Delilah").
2. Optionally, you may add an artist as well (e.g. song: "Bokurano", artist: Eve).

Song Page

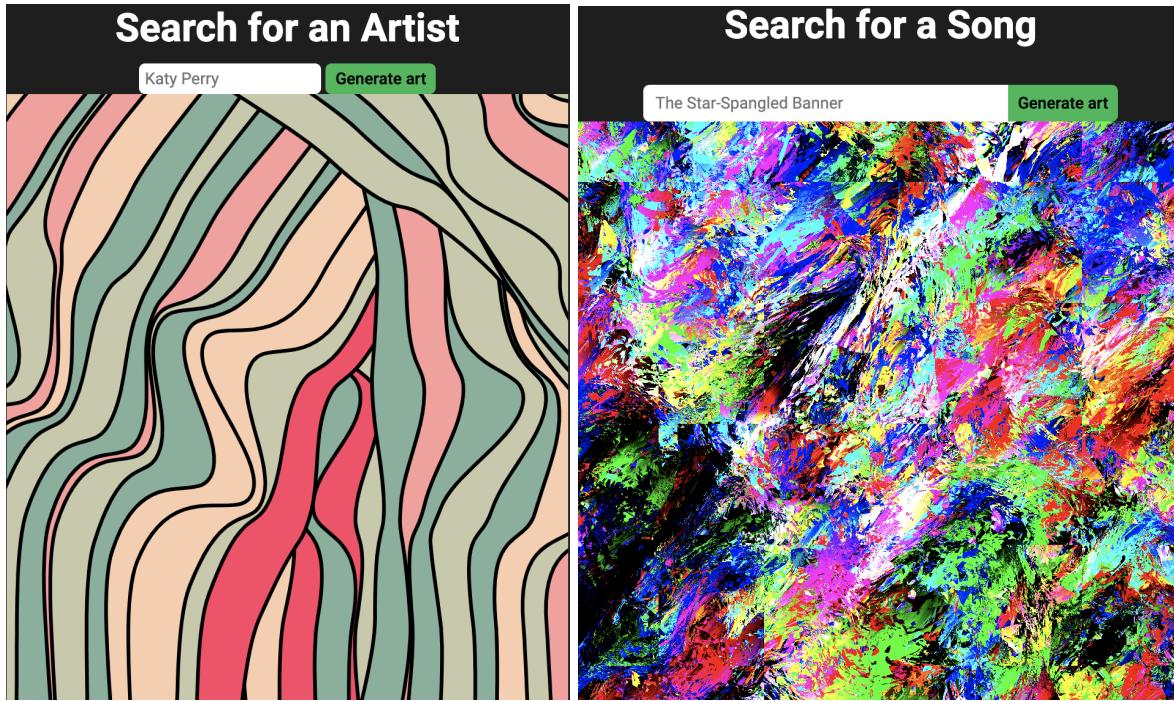


Song Page + Artist

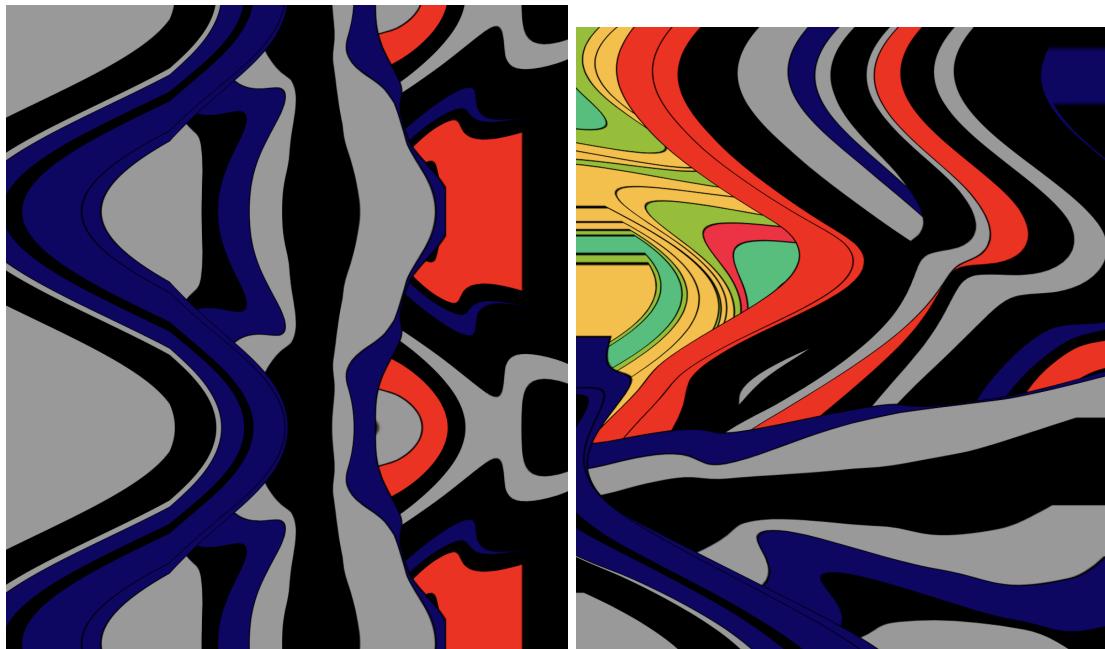


In Progress Screenshots

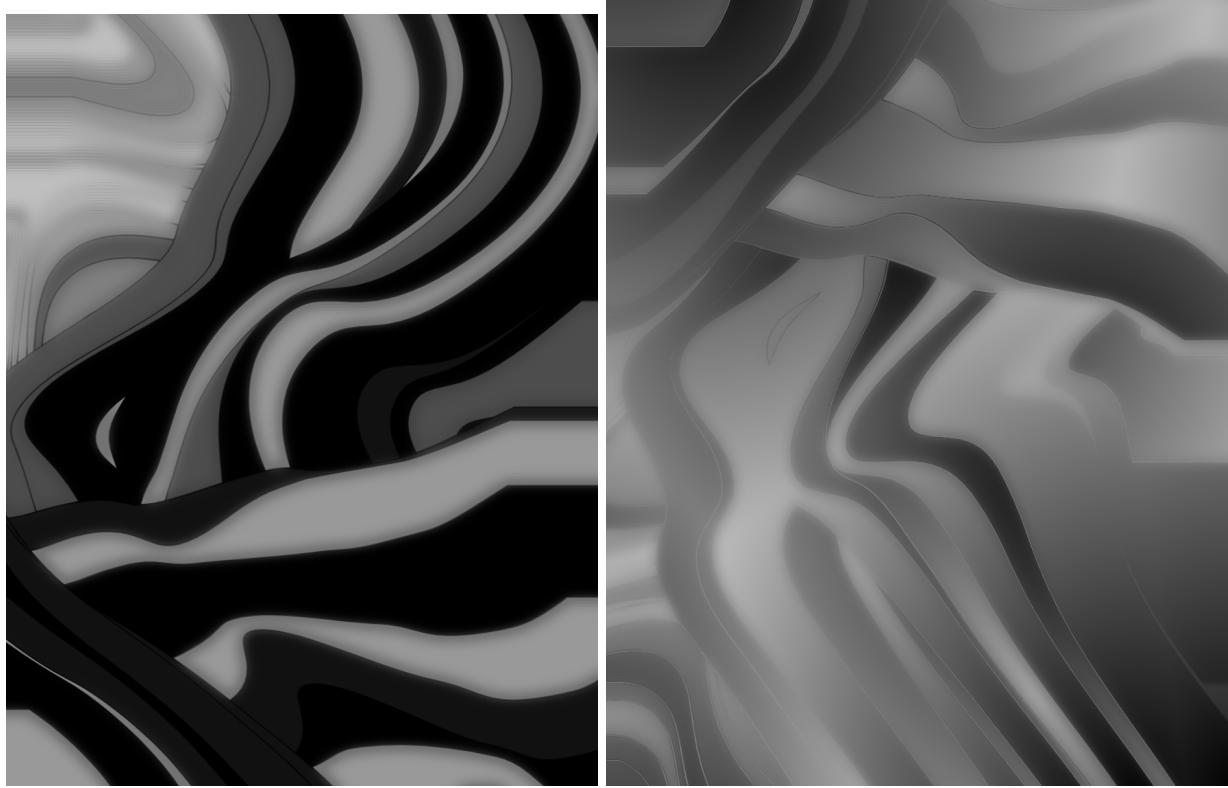
Some in progress captures.



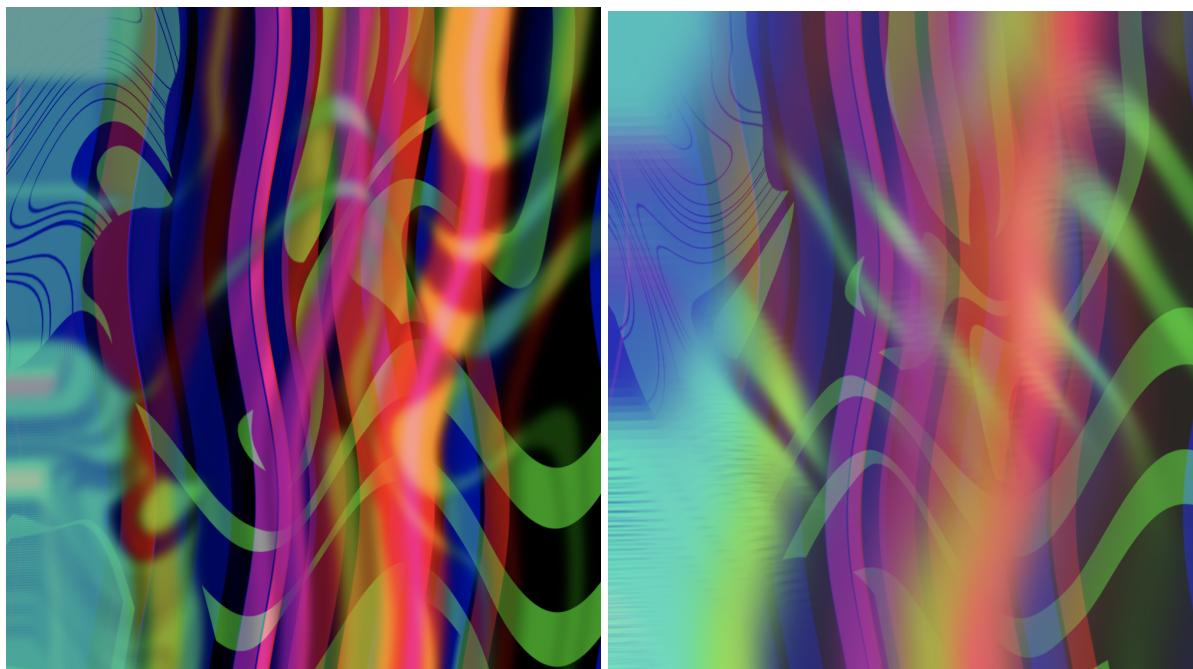
Some pictures of fill in shaders from when we first figured out how to load in shaders. We made several variations of this image to act as experimental album covers.



First instance of layering shaders successful. Took me (Jackie) a while to figure out how p5js worked with them since we get no notifications of bugs when working in GLSL (or WebGL in this case), but we figured it out eventually. Above is one of the featured shaders of the final layered shader, which is a simple sine wave shader.



Added a bloom filter and worley noise to the layered shader. I originally wanted to use the worley noise to create a cell-like structure across the entire canvas, but I could not figure it out in time because the shader kept on throwing runtime visual errors when I tested it, so I instead opted to use a cosine function to make a ribbon-like sheen across the entire image.



Final shader with the test album cover. As you can observe, we added more variety to the shaders based off the parameters that would be substituted with the actual album cover variables. For these shaders, we used speechability, danceability, and valence, all extracted from the Spotify API via web scraping. This would ultimately be the representation of each album cover's influence on its shader form.