

Co-Lab- the professional collaborative songwriting suite

1. The techstack I will be using is full-stack Node.JS, with an express back-end and a React front-end. I will also be using a few external APIs for added features in the front end (lyrics/quote searching for example)
2. My project will be evenly focused between front- and back-end technologies. Since it has a lot of moving parts and a collaborative nature, the backend will be a bit more involved than the front.
3. As a first version, this will be a website that may have the potential to grow into a mobile app, but for now it will be a web app.
4. The goal of Co-Lab is to bring songwriters together for the entire songwriting process, without them having to be in the room together. As of now, people can video chat while on a google doc together to jot down ideas, but this takes it to another level. Co-Lab focuses on the brainstorming aspects primarily, with tools to reference other lyrics by keyword, to look up inspirational quotes to find hook ideas, to create different ideas for arrangements, and to jot down notes for lyrics. This allows users to collaborate remotely, and share ideas as the song grows and changes
5. My target demographic, for now, is songwriters of really any genre. It has the potential to grow in the future to include other creative disciplines, but for now songwriting is where I am focused, as it is something I know and love personally.
6. I plan to use some of my own data that I will create, and I will also use the front-end to collect user data and song/project data. The API i will mostly be pinging is one that I will create. It will have RESTful routing, and really be used to maintain the entire app infrastructure, from user data to arrangements in a project.
7. a) My database schema is actually going to have to be quite large with lots of join tables. There will have to be user tables to manage basic user data, along with managing user collaboration requests. There will also have to be data to keep track of individual projects, and then tables to manage the data of the details within the project such as arrangements, rich text editors, and file management. Since collaboration is not required to use the tool, some projects will not have them and therefore those will need to be separated from those that do.
b) I think i may have issues storing and accessing some of the more complex data that has to do with the arrangements and the text editor. Storing the formatting from a text editor is possible, as I know Google docs does it frequently, but it will be expensive to do so. Also, since I likely won't be implementing a web socket, having the project be collaborative will mean having to constantly ping the API and check for differences every few seconds so that the page rerenders any changes that may have been added.
c) I will need to secure user passwords for sure, but also, each individual project needs to be secured since there can be copyright implications involved. I have to really make sure my auth is strong for each project, and that my database cannot be hacked to retrieve any lyric data from the rich text editor.
d) My app will feature a user dashboard, which will show users what projects they are working on, in such a way that those projects are separated between those they started, and those they were invited to collaborate on. Within any project, there will be modules

to aid with different aspects of the songwriting process, including for example a drag-and-drop arrangement maker to help brainstorm the order of the song fragments. It will include auto-save functionality and auto-rerender functionality as well to make the whole process truly collaborative.

e) User flow- reach site and find landing page with images to show how it works. Click log-in or sign-up to enter into the dashboard. Browse their projects, or start a new one. Once on a project, navigate to the specific project suite and load all necessary resources relating to that project. Click any module to load into that resource, and then save or cancel to go back to the project suite. Master save to save all modules in the project at once, and then navigate back to dashboard.

f) The use of multiple modules makes it more than CRUD. I also may decide to eventually implement a socket.io live chat function so any collaborators on a project can chat directly within the project suite. I also plan to implement a file-upload module so that people can upload any audio files they may have recorded.