1. What tech stack will you use for your final project? We recommend that you use React and Node for this project, however if you are extremely interested in becoming a Python developer you are welcome to use Python/Flask for this project.   
     
   I enjoy using React and Node. I feel I could use more practice with Python/Flask, but overall I think the app itself would benefit from React over Python. I’ll be using React and it’s component style structure to flesh out the overall design and structure of Kamikazee Karaoke.
2. Is the front-end UI or the back-end going to be the focus of your project? Or are you going to make an evenly focused full-stack application?  
     
   My current plan is to make an evenly focused full-stack application. The overall idea behind the app is to have users who can sign in, a way to track what songs a user has sung/will sing and possibly even a way for a user to connect to other users. This will require backend design as well as front end design. Overall I would like to have a table for users, one for songs/artists/genres and possibly one for conversations.
3. Will this be a website? A mobile app? Something else?  
     
   I think this would be best described as a hybrid of the two. I would like to have a website for when a user isn’t on their phone/tv, but I would definitely prefer to see this on someone’s mobile device. I would love to have access to a user's mic/camera so if features like connecting with community members or creating Karaoke Rooms become a thing, then users can share their voice and their face. Of course, the option for avatars then becomes a possibility. Maybe something like the iPhone’s memoji’s could be used in place of avatars created by a user.
4. What goal will your project be designed to achieve?  
     
   Karaoke for fun from home. I have a problem with not being able to go out and have a Karaoke session these days. It created a social environment where I could show off my skills, or lack of skills. I love making people laugh and making people sing along with what I’m up there singing. This is to fill the void that covid created in the world musically. It’s created a larger void in places, but in general that’s the goal of Kamikazee Karaoke.
5. What kind of users will visit your app? In other words, what is the demographic of your users?  
     
   This will hopefully connect with people of all ages. A lot of people in every age group love to sing and with the covid pandemic it’s created a lack of Karaoke options. I know when I was single and going to Karaoke EVERY week it was the highlight of my week. Not only did I challenge myself to stand in front of an audience, but I also felt more sociable while doing so, of course the booze helped I suppose.
6. What data do you plan on using? How are you planning on collecting your data? You may have not picked your actual API yet, which is fine, just outline what kind of data you would like it to contain. You are welcome to create your own API and populate it with data. If you are using a Python/Flask stack are required to create your own API.   
     
   I know I intend to use the youtube api to pull video’s of karaoke songs, possibly even video’s of songs themselves for the user to listen to prior to. After youtube the intention will be for lyrics to be pulled so the user can go over them before they sing the song. Will likely use the lyrics.ovh api. It doesn’t appear to have as many restrictions as others I’ve seen. Then after that I will use either Apple Music, or Spotify if I can’t find the full song on YouTube, which is highly possible. I will need to create my own API for user information storage/tracking. I would like to track what a user sings and maybe ranking on how they did. Possibly even user interaction ratings.
7. In brief, outline your approach to creating your project (knowing that you may not know everything in advance and that these details might change later). Answer questions like the ones below, but feel free to add more information:
   1. What does your database schema look like?  
        
      Songs: Artist | Song | Genre | Year  
      Users: Username | Password | First Name | Last Name | Email | DOB (Resetting Passwords) | Security Question | Security Answer (Possibly these instead of DOB, or both just to verify)
   2. What kinds of issues might you run into with your API? This is especially important if you are creating your own API, web scraping produces notoriously messy data.  
        
      The API I create will likely have issues with little. It will be there just to have data in the API already. The songs listed in the database will be there for control over what the user picks, so we don’t have any songs that are selected with no Karaoke option. This will also limit the songs users can select from, but the list should be large enough that it won’t matter that users won’t be able to select more than the artist/genre/year.
   3. Is there any sensitive information you need to secure?  
        
      I don’t think I’ll need to store much other than a username, password, DOB and first and last name.
   4. What functionality will your app include?  
        
      Random number generator that’s the length of the artist/song/genre database. A grouping function for genre and artist. GET, POST, DELETE and possibly PATCH methodology.
   5. What will the user flow look like?  
        
      Landing Page > Select Genre/Artist > Kamikazee or Select Song from Criteria > Pull song and Sing or Practice  
      Landing Page > Select Kamikazee > Pull song and sing or practice  
      Landing Page > User Information > View/Edit user’s information (Includes last x songs sung)
   6. What features make your site more than a CRUD app? What are your stretch goals?  
        
      This will be more than just a CRUD app because it’s about more than CRUD. It’s about user interaction with the app more than anything. The intention of this app is to make it accessible to all and make it so users can have a good time. It’s less about the program and more about the user. Stretch goals include the use of sockets to connect people. Perhaps broadcasting their karaoke performance by audio and video, or even just audio. Including avatars to make it more anonymous.