Pitch:
Music is something that everyone is passionate about. Our website will allow people to share their music opinions about what they care about with who they care about.

Functionality:

1. Users can leave reviews on songs they like
2. Users can search for songs
3. Users can choose to view by genre on main webpage
4. Users can score songs on scale from 1-10
5. Users can choose favorite songs
6. Users can share song pages

Components: (Once you've described the components, *include a diagram showing how they communicate or interact with one another)*

We will use ReactJS for the front end and NodeJS for the back end.

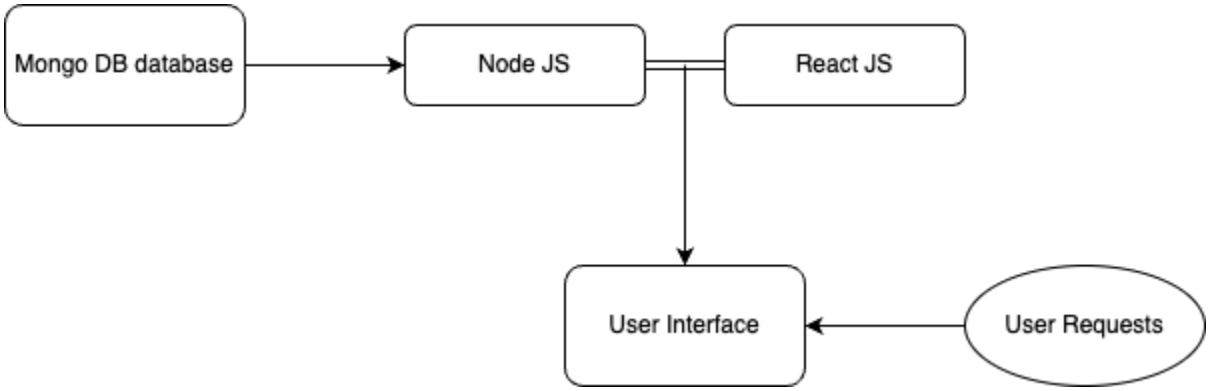
Programming languages - Our project will utilize javascript as the main language. We chose this because it allows the creation of a scalable and maintainable web application.

Libraries - We will incorporate the React JS framework to build the front end and Node JS for the backend. In addition, we will use MongoDB in order to implement a database to store the data our website will hold.

Testing - We will use unit testing in order to ensure each component of our program works the way we expect it to. We should be able to approach 100% test coverage with our website to ensure our website works well.

Interactions with other components - In order to update our webpage with the appropriate information, we will make HTTP requests (POST, GET, PUT, DELETE). This can be done through Node JS.

Diagram of web application



Continuous Integration:
To run our tests, we will be using Mocha and Chai, per the suggestion of our mentor. The style guide we will be following is camelCase. A library we can use to enforce the correct style is “prettier”. Test coverage is included in Mocha. Pull requests will be opened whenever someone completes one unit that the group assigned them. We will alternate who reviews each person’s code so we can get a fresh set of eyes on them. When there is a merge conflict a designated person will review it.

Schedule:

1st and 2nd weeks:- We will learn the React JS framework for frontend and Node JS frameworks for backend using Javascript, and then also learn MongoDB which we will use to create a database that will store the data our website will hold. Could go into 3rd week if behind. It would be helpful to start writing lines of code at this stage, so we can sketch out what we need to do using examples. Also work on home webpage that has a list of user selection song categories to rate from.

3rd + 4th weeks:- Work on individual webpages for song categories that have list of songs and individual webpages for songs inside each category’s list. I don’t believe this phase should take too long once we have an idea of how it works and what we need to do once we get one song rating review working properly. Get working on rating system as well as allow users to share songs via HTTP POST request which updates the database list of songs.

5th week:- Work on search bar for searching a song that might or might not exist(front-end and back-end). Hopefully, we have made enough progress in 3rd and 4th weeks on project. Can continue working if not finished. Also take a look at overall

functionality of our website with webpages and start running continuous integration tests using Mocha and Chai test development frameworks.
6th, 7th, 8th weeks:- We can begin wrapping up progress on our project if it is looking more complete. Now, the initial stages are complete. Continue debugging testing. Checking for errors in code. Don't think there will be too many edge cases to worry about right now. Make sure that the frontend and backend are interacting properly.

Risks:

1. Since we are all learning the language and libraries, it is possible that we fall behind. This could look like being on week 5 or 6 and having little to no code complete. The only possible solution to this is to devote more time to the project as well as supporting the team members who are falling behind. At most this should push us behind by one week. We have time built into our schedule for debugging so this will go into that time.
2. The libraries we decided to use are all new to us, so we might find that they do not adequately fill our needs. Since none of us have used javascript or nodejs/reactjs before, it is possible that a feature we want to implement is not able to be implemented. If this happens, we will find a different library that supports what we are trying to include in our project. It should not affect our schedule that much (only a couple days at most).
3. It may be difficult to implement a database using MongoDB. Since most of us have little no no experience with database design and implementation, this part may slow us down when developing our project. Our progress could slow down and put us up to a week behind, but this risk can be alleviated if we help each other to grasp database concepts and learn this new concept together.

Teamwork:

Our group is thinking about using a docker environment, we will need to decide this within the coming week. One way we plan to reduce friction is to have clear-cut roles so we are not pointing fingers and instead know exactly what went wrong. Right now the plan is for Diwakar and Rahul to work on the back end and Danny and Nico to work on the front end. This makes sense since Diwakar has experience on the back end and can support Rahul who is learning. Nico has the most experience on the front end so he should not have too much trouble but still has someone to assist him.