1. Project Team Repo Submission

A short project report (less than 4 pages) submitted on behalf of the team in the main project repo following the structure for D1 with a file that is named “deliverable-2". This report contains

1. Summary of your software (you can reuse the same submission in D1 if it hasn’t changed).

a. One paragraph for the problem & partner (if applicable)

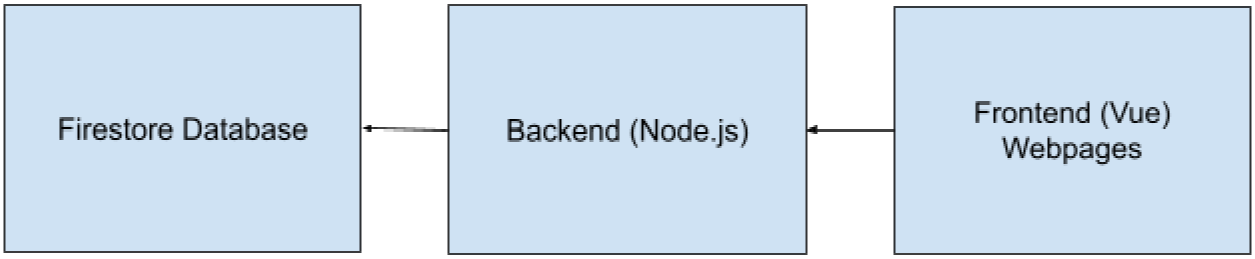
Simulence is a web application with the goal of being a platform that indie developers can use to upload their games, for which they will pick a group of playtesters to QA test their game and provide feedback based on surveys, recordings, and interviews. The hope is that this provides a more accessible way to playtest games and refine them, without going through the expensive effort of hiring a full-fledged QA team.

b. One paragraph introducing any existing software/infrastructure (if applicable)

- Existing stack: vue, node.js, firestore
- Webpack for development, firebase for deployment
- Auto-deployment already setup and documented for replication
- Existing software: able to create users (both developers and testers), create surveys, fill out surveys and save the response in a database, and view survey responses individually.

What currently exists is a website where users can sign up as playtesters or developers, the latter being able to create webpages for their games, and add any existing tester to work on it. The developer can create surveys that the playtesters can then respond to, and have their responses saved in a database, but the actual implementation of the survey analytics has yet to be added. The existing web application uses Vue.js for the frontend, Firebase for back-end and deployment, runs in Node.js, and uses Webpack for development.

2. 1-4 paragraph(s) on how you decided to divide the project and why. Ideally, this will be accompanied by a software architecture diagram and how each component connects to the others.



We decided to divide our project along three subteams: Front-End UI, Back-end and database functionality, and CSS improvements. The division between front and back end functionality was done to mirror standard web development practices (because it does make sense to separate server and client code in general!). The addition of a separate CSS team was done to satisfy a request from our partner, who has said improvement of Simulence's UI look-and-feel was a priority. Also, we noticed that the previous team used basically raw CSS to style the entire project, and saw an opportunity to improve it (and learn a new framework) by employing Bootstrap.

This also helps divide the project along different skillsets – the front-end team works largely in Type/JavaScript using Vue, the back-end team works in Node.js using JavaScript, and the CSS team works mostly using CSS (with plans to upgrade to Bootstrap). Since the three sections use pretty different technical skills (with some overlap), splitting the work along these lines lets our project members focus on honing a few skills each – instead of overextending ourselves trying to work across the entire stack. This also (mostly) mirrors what real tech companies do. Furthermore, the overlap allows for co-operation between different teams when issues arise, as everyone should be well-equipped to deal with other areas of the program.

In short, our divisions were made to help us adapt to a new stack by controlling complexity, hold to the standard separation of concerns in web applications, and accommodate the priorities of our partner.

3. One paragraph for part(s) each sub-team is responsible for.

Sub-team 1 was responsible for implementing the ability to view the analytics for the survey responses on the Developer's end, as well as designing how the analytics would be viewed. Although surveys could be created, stored in the database, and recorded in the Developer's Analytics page for their game, the results could not be viewed. The task was to create a mock-up for how the survey analytics are to be viewed, where clicking the desired survey on the surveys page will display sample questions and answers that accurately mimic the types of feedback surveys game developers would wish to create and hand out. For easy understanding, any questions using multiple-choice, multi-select, dropdown options, and a linear scale have their responses displayed through a bar graph, implemented with the chart.js library, that explicitly numbers the amount of times each choice was picked. Each question that requires a text response has their answers displayed through a Text Accordion that encompasses all users who took the survey, where clicking them will open up or close their answer. To make things more aesthetically-pleasing, Bootstrap was used for the styling and functionality of the Accordion.

Sub-team 2 was responsible for developing the logic/backend for the new survey analytics functionality. This encompasses linking the data stored in the database with the frontend, providing the necessary data processing and retrieval functionalities. Specifically, we are tasked with implementing CSC exporting, enabling users to export survey responses for further analysis or record-keeping. Additionally, we are responsible for writing functions to retrieve responses corresponding to a specific question within a given survey, as well as retrieving different survey analytics depending on the type of question (i.e. multi-choice, dropdown, text-based, multi-select, and rating scale questions). These functions will eventually be called by the frontend written by sub-team 1.

Sub-team 3 was focused on improving the user interface of the web application. This included both improving on the CSS to touch up on the design of each component, as well as rearranging and adding new components to improve the overall user interface. Our team found that upon testing of the previous group's application that the visuals of the web application lacked presentable user interface, as well as user experience, in many of the different tabs, such as the survey, testing, and calendar sections. As such, we decided that it would be reasonable to dedicate a team to the aspect of the user experience of the web application.
We also additionally created a few more profile pages, and restyled a few existing pages with unimplemented functionality to better fit what we plan to implement in the future.