1) **Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).**

   When it comes to our original project proposal, there have not been many changes when it comes to the intended end goal of our project. We still continued to implement our project with the intent of simplifying fact-sharing in Dungeons and Dragons campaigns. We did not have enough time to successfully implement the front end for campaign management, though the backend code is implemented for integrating this in the near future.

2) **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**

   Because our intended goal was to improve fact-sharing between Dungeons and Dragons campaigns we do feel that our web application successfully achieves its intended purpose. As our application is dynamic, data will simultaneously be updated depending on users' interactions with different facts and artifacts.

3) **Discuss if you changed the schema or source of the data for your application**

   We did not really change the source of the data for our application, because this data is reliant on individual Dungeons and Dragons campaigns. Mainly for our mockup, we scraped Wikipedia to generate our mockup data for large-scale testing.

4) **Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?**

   We did not make too many changes but one big change we made was moving away from our original design of using weak entity relations between facts, artifacts, and campaigns. We ended up doing this following the advice from our TA, and it simplified our database structure. In doing so, we avoided having to use multiple keys to identify something, which can be complicated with API calls.

5) **Discuss what functionalities you added or removed. Why?**

   We kept the basic functionality we had described, such as creating and removing artifacts and facts and searching artifacts and facts based on keywords. The only setback we did have was that we did not have enough time to implement the

interfaces to manage the different campaigns, so different functionalities such as adding or removing these were not implemented.

6) **Explain how you think your advanced database programs complement your application.**

   Our advanced database programs complement our application with our search feature making it much easier to look through all artifacts to look for any specific entry that we would like. In addition, our implementation of a friends list will allow users to find and message each other or quickly find artifacts written by them that the user has access to in the future.

7) **Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.**

   Justin: One of the biggest technical challenges we faced was using SQL within a Node.js application. Although the team has had previous experience working with ORMs such as Sequelize and Mongoose, we had not previously worked with using raw SQL queries to interface with the database. We looked extensively into how to do so defensively, including using parameterized queries to mitigate potential dangers caused by SQL injection attacks.

   Rishi: Using Google Cloud was a technical challenge because it was the first time we were using the platform. Tasks such as navigating Google Cloud's complex interfaces were a new experience in addition to identifying what features we needed. There were also security issues because the database could be exposed to the public. We were however able to take advantage of google cloud's features such as the IP Whitelist and backup options to create multiple copies of our database while we worked so we could restore at any time.

   Cristian: A technical challenge we faced was dictating the different API endpoints necessary, and making sure that these endpoints are quick and efficient enough to generate the desired results without having to crash our application. Because we had a number of different APIs implemented, it was vital to make sure that all of these ran smoothly.

   Nate: A technical challenge that the team encountered was figuring out how to connect the backend to the front end. Specifically laying out how and what we

needed to fetch for any particular function was tricky to implement at first but after figuring out the first few we realized patterns we could use to make everything else easier.

8) **Are there other things that changed comparing the final application with the original proposal?**

   There were simple differences between the UI diagram we had generated and the final result, but aside from this, there were not many changes from our final application and the original proposal.

9) **Describe future work that you think, other than the interface, that the application can improve on.**

   While we had intended to implement user authentication as an extra feature, we hope that if we continue to work on this project in the near future we can successfully implement this. We have the backend authentication verification code implemented, but just needed some more time to successfully add this new feature, as we still needed to fix minor bugs with our intended goal.

10) **Describe the final division of labor and how well you managed teamwork.**

    Cristian, Nate, and Rishi worked on a good chunk of the front end, implementing and designing the various React components. Rishi also added support for using the third-party Markdown library, laying out the final interface,. and integrating them to update dynamically with calls to the backend endpoints. Justin focused on designing and implementing these endpoints, and integrating the SQL queries using a Node.js SQL driver and Express.

    Cristian and Nate each also wrote the advanced queries, Cristian writing the advanced query for finding all other users that a given user can interact with, and Nate writing the search artifacts and facts by title and entry query.