

General Project Guidelines and Instructions of Video

Here are some general rubric items for grading the final submission of the “standard” course project. They may not cover every aspect of your project; however, you can follow them to improve your project. This list can also serve as a grading reference for other types of projects.

Out of the 100 points of the project grade, 40 points are for completing the proposed work (as a team) and 30 points are for the quality of the work (as a team). 5-10 points Therefore, the following list is also a guideline for how to achieve good completeness and quality, but it does not cover

1. Completeness.

- a. The website should be accessible, and each page should not collapse.
- b. Each call to the backend should result in a proper action; e.g., it is either correctly executed or the error is captured.
- c. A set of basic functionalities that are intuitive to the user should be implemented. Other functionalities or features would be counted towards the quality of the project.
- d. There should be at least one or two [user scenarios](#) such that describe a complete story about how a user interacts with the website and accomplish what s/he needs. For example, a user visits the website, navigate through pages and finally finds the info.
- e. The website is secure to the SQL injection attack and should have interactive performance (i.e. you should not need to wait for a few seconds to receive an output).

2. Database Design

- a. The final E/R diagram you submit with your project report should be consistent with the design of your website functionalities. It should support all basic functionalities, and ideally should be easily extensible to support future functionalities. But note that several SQL designs are possible converted from E/R diagram.
- b. The database schema should be consistent with the E/R diagram. Views, triggers, constraints, indexes should also be considered into the schema if necessary. Many database metrics, such as data redundancy, storage cost, query performance, etc, are significantly dependent on the database schema. We would like to see how you analyze your database schema and optimize it.

3. Dataset Quality

- a. A production site usually is supported by a large-scale dataset. The data source in your website database should contain at least a few thousands records, possibly more. You can use either real-world data or synthetic data.
- b. Most of the data should be meaningful, well-formatted and correct. It is often the case that you need to scrape data, and this procedure is crucial to the quality of your data. The data source is also important.

4. Submitted final files:

- a. They are described in the original project pdf. Please make sure to detail everyone's contributions in members.txt. In addition, submit a video for presentation of your project as follows.

Final Demo (Video)

The video should be submitted on gradescope by Friday 04/24 night.

This is in addition to the other materials mentioned on

<https://www2.cs.duke.edu/courses/spring20/compsci316/project/project.pdf> as we cannot have an in-person presentation.

The final submission of the project also requires a video **that should include both a short presentation and a demo**. It should be 5-7 min long in total. Here is a guideline for what you could include in your video.

1. A walkthrough of your website.

- a. Briefly describe the purpose of your website; e.g., who are the general users, why they need this website, and how they should use it.
- b. Introduce each functionality; e.g., what it is and how it is used. You can follow the order of how a user would interact with the website.

2. Briefly describe your database, queries, and dataset.

- a. Introduce which dataset you use and briefly describe how large it is and how you parse it into your database. List the main tables and state how they are related to each other if so.
- b. Explain your optimizations to the database if you have; e.g., indexes, table decompositions, materialized views, triggers, transactions and so on.
- c. Show us a slide about the dataset you used, its size (number of tuples in each table, number of attributes in each table etc). We hope your database would have a few thousand tuples (a few hundred thousands to a million would be even better!)

3. Introduce how your frontend interacts with the backend.

- a. For some text/table/graph shown on the page, you can briefly explain which query it is related to and how the frontend and backend process the query result.
- b. Discuss if you used PREPARED statements.
- c. Discuss if you guarded against SQL injection attack.

4. Show a slide about challenges and future work.

- a. What would you have done if you had more time, what you liked about your project the most, show us some fun pictures of your team (virtually) working together!