Your final project will be an opportunity to create an application that both shows the skills you have learned or improved during the course of this class as well as to build a 100% made-by-you project to put into your portfolio. There are three parts to this project - be sure to **read the entire project description before you begin.** I recommend reading the application description, then the database description, then the planning description.

I will give you feedback after each of the first two parts, with the expectation that you will incorporate this feedback into the subsequent portions of the assignment.

When the project requires a written response, I will be expecting a substantial, thoughtful response. All databases and applications must run without error.

Part 1: Planning Phase (Final Project Planning Quiz)

In this phase, you will describe your planned database and application in detail. You will submit an ER diagram and answer some guiding questions to make sure you are on the right track. You may not use AI to generate your diagram or responses to the quiz questions.

Part 2: Database Creation (Final Project Database)

Your database should be contained in a single .sql file and contain the following:

1. At least four tables with data loaded into them.
    a. At least two tables must include foreign keys.
    b. At least one table should have a composite primary key.
        i. This composite key may not include an integer id field that uniquely identifies this table. It may be made of up integer ids that are foreign keys.
    c. Tables must be in third normal form.
    d. You must use at least one from each of the following datatype groups somewhere in your database:
        i. Date, Time, and/or Datetime
        ii. Numeric and/or Decimal
        iii. Enum
        iv. Boolean/TinyInt
    e. Your largest table should have at least 30 rows. You may use AI to generate the information that will be inserted into the tables, but you will be responsible for ensuring data is consistent within and across the tables.
2. At least one view or stored procedure that “reads” from your database and aggregates or summarizes useful information.
    a. Data should be viewer friendly and display information in an organized, easy to read manner.
    b. An aggregate function must be used in at least one get procedure/view.
3. At least one procedure that adds new data to the database.
4. At least one procedure to update a row. Changes should cascade.

5. At least one procedure to delete a row. Deletions should cascade.
6. You should build any other procedures, functions, and views needed to support your advanced feature.

<u>Part 3: Final GUI Application (Final Project)</u>

You application should be submitted in a zip file and include the following:

1. A readme file with detailed instructions on how to run your application and information about your advanced feature.
2. At least three distinct modules that separate the view, business logic, and data access layers.
    a. Only the data access layer should communicate with the database.
    b. Database connection information should not be hard-coded (with the exception of the database name). Connection information should be entered in the log-in screen. You may set defaults for the host and port, but these should be changeable.
    c. The data access layer should only use predefined stored procedures, functions, and views.
    d. The view layer must have a graphical user interface that is easy to read and understand.
    e. The view layer should provide the user with information about what is happening and what went wrong if results cannot be retrieved.
3. A log-in interface that precedes the presentation of any data.
4. The ability to add new information to at least one table.
5. The ability to update existing rows in at least one table.
6. The ability to delete rows in at least one table.
7. The ability to retrieve and view the data in the database.
    a. Any information that changes in the tables should be reflected in the views.
8. An advanced feature of your choice (examples: exporting to a csv or pdf, importing additional data from a csv using the application, allowing the user to filter a view on the client side, displaying charts or graphs of information in the database, connecting your data to data from an API, etc). This feature cannot be something that was part of the MRC project. You are expected to independently research and implement this feature.
9. As always, the application must run without crashing.