CS348 – Project – Stage 2

Due date: 12/8/2022 at 11:59 PM.

The main deliverables of Stage 3 are the following:

1. A document in your shared folder to describe what indexes you have on your tables and what query(s) and report(s) those indexes support. For each index, list the query(s) that benefit from the index and where the queries are used (e.g., a specific report).
2. A final demo of your project. You will record a 5- to 10-minute demo and save your demo (video) in your shared folder. In the demo, **show requirements 1 and 2 (in stage 1) how to use them**.
3. In the demo, **describe parts of your code where you implemented the following course concepts**:
    a. Using at least two of the following methods: **prepared statements**, **ORM**, and **stored procedures**. Each method should account for at least 20% of your database-access code (e.g., 80% prepared statements and 20% ORM). You may consider using ORM for data entry and updates as well as simple queries and use prepared statements for complex reports.
    b. The user interface of requirements 1 and 2 should retrieve data from the database if needed, such as when populating the items of a drop-down list (e.g., a drop-down list or a list of check-boxes to select a particular course. Such a list must be built dynamically using data from the database. The courses must not be hard-coded in the interface). In your demo, show how you build those interface components dynamically.
    c. Transactions and your choice of isolation levels (transactions and concurrency will be covered next week). If your application is designed for a single user you may discuss a version where multiple users can access the same data concurrently.
    d. Discuss the lessons learned during the project phases. What would you change if you could start over.
4. A url to your application (if available) and your application's code (e.g., a GitHub link).

Grading:

Stage 3 is worth 70% of the project grade. Your assigned TA will grade your demo and code. Grading will focus on the database concepts and whether you implemented them correctly in your project. For example, did you use the right indexes to support your report. Did you use transactions and appropriate isolation levels. Any user interface is acceptable as long as it is a GUI (web, mobile, or desktop application). Command-line interface is not acceptable (I have not seen any information systems that use command line interface!).

Extra credit:

1% of the course grade if you deploy your project (**database and application**) to Google Cloud Platform or any other cloud provider (e.g., Amazon AWS or Microsoft Azure). If you want to deploy your code to GCP you may use the app engine or compute engine (I think app engine is easier). If you do so, we expect a url to be available at the time of grading (between 11:59 pm on 12/8 and 11:59 pm on 12/15). Please remember to delete your instances after this date.

Deliverable	% of project grade
Indexes report. List queries that benefits from each index (be specific: report XXX or feat XXX). The indexes must be meaningful (important queries, reports, etc.). You must have a valid justification.	10%
Demo. A proof that requirements 1 and 2 are complete (data can be inserted, updated, and deleted). A report is displayed before and after some data is changed. Note. Remember to highlight the source code while showing/explaining the above features.	30%
Used two database access methods	10%
User interface built with data from the DB. If not applicable, 5% is added to 'using two access methods'	5%
Discussion of transactions, concurrent access to data, and the choice of isolation levels. If not applicable, 5% is added to indexes and 5% is added to demo.	10%
Discussion of lessons learned	3%
Copy of the code	2%
Total	70%
Url to the application. A test of the live application (data is entered, updated, or deleted and a report reflects the changes made by the TA). The demo proves the database and app are hosted by some cloud provider.	Extra credit (1% of the course grade).