Web App

CMPE 138 Individual Project

Objective: Implement an interesting web application with a database back-end. This may be a novel application you come up with, or a "clone" of an existing web app (bank, amazon, etc.). If you choose the latter option, it might be better to choose a more obscure app that does not have existing solutions online. This project will help you learn, but not if you copy someone else's ideas (also, doing this is a case of plagiarism and will earn you an F in the assignment and potentially the class – cheating will NOT be tolerated).

Team size: 3

General description: Design or clone an existing web app which stores its data in a relational database. The database and web app have to have enough complexity to warrant a team of 3 working on the project throughout the semester (5-15 tables, appropriate constraints, etc). You can choose whatever technologies and languages you choose to complete this project, with the restriction that you should use an RDBMS (not, e.g., a NoSQL or column store engine). In particular, you can use this project to practice your general SQL knowledge in a MySQL or Oracle environment. The following tasks should be completed:

1. Project proposal: submit a 1-2 page report describing the application you would like to create or clone. The report should include enough detail of the application and its functionality to understand the relations necessary to implement its functionality. A preliminary list of these relations should be listed in the report.
2. Decide an appropriate database schema. Create an ER diagram for the database, which you will include in the report.
3. Create the database and its tables. You should have 5-15 tables in the database, with a variety of constraints between them. Populate the tables with some test data. There should be enough data in each table to test each of the relational constraints you specify in the ER diagram.
4. Write 30-50 queries that test all the functionality of your database. Execute these queries and capture their output.
5. Create the web app, which should have enough complexity to make use of all relations in the database.
6. Write a report in no more than 10 pages, in which you describe the purpose of the web app, choices in architecture, intended available functionality and features, design of the database (including ER diagram), and other aspects of the architecture or implementation that you think are worth mentioning.
7. Demo the project. You will have 10 minutes to present your app and its functionality. Each team member should present some of the functionality.

Deliverables: The following items should be included in a *zip* or *tar.gz* archive and uploaded to Canvas. The archive should be called *project.zip* or *project.tar.gz*.

- A *database creation script* (or set of scripts) that creates the database tables and loads appropriate test data for the database.

- Your query execution log, in a file named *queries.log*, containing the 30-50 queries you wrote to test the functionality of your database. The file should be a text tile. Each query should be printed out, followed by the output it returned.
- A *pdf* version of your report, named *report.pdf*.
- A sub-directory named *web-app*, containing the code for the web app. You should include a README.txt file with instructions on running/installing the web app and its database.

Evaluation: The projects will be evaluated based on database design, correctness, documentation, presentation, and appropriate evaluation/testing. The following grading rubric will be used for the project.

- Database design (40 pts)
    - Does it capture all the attributes correctly?
    - Does it allow efficient queries without unnecessary data duplication?
    - Is the ER diagram correctly drawn? Does it capture necessary constraints?
    - Is the diagram correctly translated into database tables? Does the database creation script work as expected?
    - Does the schema support the required queries?
- Database queries (20 pts)
    - Do all the requisite queries return appropriate data?
    - Is the set of queries sufficient to comprehensively test the intended functionality in the database?
- Web app (20 pts)
    - Was the application complex enough given the team size?
    - Did all team members participate in the application demo?
- Reporting (10 pts)
    - Does the report file exist and include all appropriate information?
- Package (10 pts)
    - Are all the files/directories in the archive named correctly?
    - Are there missing files?