

Term Project Report

COP 4710

Application: College Event Website (CEW)

Format and Submission Summary

- Groups: all groups
- Format: report file in pdf
- Software: in appropriate format, e.g., '*.sql,' '*.c,' '*.txt,' etc.
- One compressed file: all contents in one folder (include all files organized in subfolders), compressed to 'CEWGroup.zip,' where *Group* is the number assigned to your group, if any, if not it is the appended last names of all members (no space in between).
- Submission: upload the file to the 'Term Project' assignment by the due date. ONE group, ONE file.

Report Contents

- Project Title and Group Information:
 - Title: 'Group Report,' Project Name, e.g., 'College Event Website'
 - Course information: course number, section, term
 - Group information: group members
 - Table of contents: GUI, ER-Model, constraints, relational data model, 3NF decomposition (if required), demo, etc.
- Project description, additional assumptions, if any (to simplify the implementation or to make the project more realistic).
- GUI: Platform, languages, DBMS, screen shots of 'Create RSO,' 'Create Event,' 'View Events,' etc.
- ER-Model: ER diagram, constraints captured in the ER-model and other general constraints to be enforced by other means, such as assertions, triggers...
- 3NF decomposition (if required)
- The relational data model in SQL: 'CREATE TABLE,' 'CHECK,' 'CREATE ASSERTION,' 'CREATE TRIGGER,' as required.
- Populating tables with sample data: for example: two universities, 20 students, 3 RSOs, five events, three comments, etc. IMPORTANT: integrity of data must be maintained, i.e., all constraints must be enforced, including no overlapping events, active RSO must have at least five members...
- SQL examples and results:
 - SQL statement to insert a new RSO (part of the processing of the 'Create RSO' form). Show results.
 - SQL statement to insert a new student to an existing RSO (part of the processing of the 'Join RSO' form). Show results.
 - SQL statement to insert a new event (part of the processing of the 'Create Event' form). Show results.
 - SQL statement to insert/update a (new) comment (part of the processing of the 'Create/Add/Modify Comment' form). Show results.

- Several SQL queries to display events—public, private, and RSO-- (part of the processing of the ‘View Event.’ request by a user with a specific role). Show results.
 - SQL statements of interest (optional), e.g., advanced SQL queries
- Constraint Enforcement: Show SCREENSHOTS of error messages/warnings when the following events/attempts are made:
 - A new event to be held at the same location and overlapping times with an existing event: Show error message with enough detail such as the conflicting event, time, location, etc.
 - An admin who is not the Admin of the RSO attempts to create an event for that RSO: Show an error message.
 - An INSERT of a member of an RSO with 4 members: Show the status of the RSO changing to ‘Active.’ A DELETE of a member of an RSO with 5 members: Show the status of the RSO changing to ‘Inactive.’
- Advanced Features: If implemented, briefly describe the features, and show screenshots.
- **If the demo was not as expected** (i.e., crash, something failed, produced wrong results, etc.), and your team promised the problems could be fixed, discuss what you did here, including the new code, the results and their screenshots.
- Conclusion/Observation:
 - Database performance: query response time, suggested indexes
 - Desired features/functionalities: security (login), event feed from university websites (e.g., XML feeds from <http://events.ucf.edu/>), social network integration (Facebook, Twitter, etc.)
 - Problems encountered, things that have been learned from the project, more things/skills needed to master to build a more advanced database app. In retrospect, what are the things you would have done differently to implement the project?

Folder Contents

- Group Report
- Subfolder: contains the source code and (if open-source) required libraries to run the app (in appropriate format, e.g., ‘*.sql,’ ‘*.c,’ ‘*.txt,’ etc.). A ‘*.sql’ file containing SQL statements to set up the environment (database name, user, password, etc.) and create the database (CREATE TABLE, but no data needed, i.e., no INSERT statements) is to be included here. (You can accomplish this by ‘exporting’ the database, not the sample data). Hint: back up (only the structure of) your database.
- Software installation: installation instructions

Submission

- One compressed file: all contents in one folder (and subfolders), compressed to ‘CEWGroup.zip’
- Submission: upload the file to the ‘Term Project’ assignment by the due date. ONE group, ONE file.