CSIS 3280 Final Project

Each group member must contribute working code to your project. Group members found not contributing code should not be given free mark! Projects that do not run after loading the database will be given a ZERO mark.

Project Requirements

The Group Project is a project that should cumulatively cover all the topics in the course, it is to model an application with a small real-world use case. The project will be a Web based Object-Oriented PHP app and will connect to a local database using PDO (PHP Data Object). You must have **at most two people** in your project. If you work in a team of two, appoint one of you as the team leader who will communicate with me and submit the project later. Please contact the instructor if your team wants to create a project using Laravel framework.

Requirements

Data and Program Requirements:

- The database:
  - Except if it is an associative table that breaks the M:N relationship, each table/entity in the database should have at least five columns
  - If you are working alone, the database should have at least two entities
  - If you are in a team of two, the number of entities > count($members). One of the entities can be an associative entity
  - If you are in a team of three, the number of entities > count($members). One of the entities must be an associative entity
- Entities must be stored in a Database and accessed using PDO.
- The application must support CRUD (Create, Read, Update and Delete) Items using PDO. This must be implemented for each entity. It does not mean that you should have the same list view, add view and edit view for each entity. It should be a complete web application, not a data view and entry application.
- The application must have multiple HTML forms and pages.
- The user must be able to search and lookup for records from the database.
- Statistics must be shown for at least one entity.
- All input must be validated, and the appropriate events handled; proper English must be used.
- The app must be easy to use and visually pleasing, professionally designed with effective use of layouts, text, page and content elements and user input elements.
- Users must be able to login; their credentials must be encrypted. The application must use sessions logging in and logging out.
- The above requirements are for passing the project. You should try your best to make your project stand out by integrating new technologies or topic that are not covered in the class.
- Additional features that extend beyond what we covered in class will be rewarded.

Project Documentation Requirements

1. **Project description (3-5 pages)**
   You must describe your project and explain how all the entities interconnect to create your application. In addition, you need to:
   - Specify your group members and the entities they are responsible for.
   - Include a list of features you implement and why these add value to your project.
   - Include an installation manual, how to set up (database, folder structure, web alias, etc).
   - Include a list with specific examples of how the technical concepts were implemented (see requirements)
   - Include a list of the technologies implemented as an extension beyond what was covered in the classroom.
2. **Class diagram**
   Specify the entities, the PK and FK and cardinalities. You can use tools like Software Ideas Modeller or any other UML drawing tools like lucidchart.com to create your model
3. **Meeting minutes**
   For those working in a group, every time the group meets, be sure to record the meeting minutes. Be sure to include who was present and provide details on:
   - Attendance (who is present for your meeting)
   - What was done since the last time the group met (per person)?
   - What is working well?
   - What is not working well?
   - What will be done before the next time the team meets?
4. **User manual (1 page)**
   Provide a concise user manual how to install, use and operate the application.
   - Include visuals wherever applicable
   - Must cover all the actions a user can perform (CRUD and Search etc...)

Bonus requirements

- You store the code of your implementation in your GitHub or BitBucket in which the instructor was invited as one of the contributors. Commits were done from each member of the team from the starting date of the project, not just before the dateline for submission.
- The project is hosted online (see note about the controller file). You may want to look at https://platform.sh/, https://www.000webhost.com/ or any other free cloud platform provider.

References

You must provide references for any other materials that were used or consulted for your project. This includes stack overflow, w3schools, LinkedInLearning.com etc...

Submission

Your Team submission should include a folder with the naming convention TeamNumberXX, with XX denotes your team number. You should have one controller file called "TeamNumberXX.php". Note: if you are hosting your app online, you need to change the main controller file name into index.php.

You must provide the appropriate SQL for your project in the data directory with the filename extension .sql. Your program structure should be similar to the kinds of programs we have been doing in class using MVC where applicable.

Grading

The project will be graded on a scale of 20 points.

Criteria	Grading
The web application produces errors or warning messages. The output of var_dump or print_r are visible on the web app. The web app creates folders and files automatically.	-16
Project submitted and named properly with all assets to Blackboard by Team Leader, file is named according to the naming convention	1 point
Project Description, installation manual, Class Diagram, Meeting Minutes, all completed with relevant details. Class Diagram is accurate and includes all properties and cardinality. User Manual – Concise, to the point, graphical, labeled properly.	3 points
Good program structure is used, all functions were described with comments, and comments are used where applicable. Naming conventions are followed.	2 points
The database was designed following the requirements. The tables/entities were sound and were designed to reflect real life web application. The complexity of the database should be similar to or greater than any database design used in the class practice. The SQL script was included in the project submission.	3 points
CRUD operations are implemented using PDO, DAO and prepared statements. The errors are captured and logged. The user must be given positive confirmation of a CRUD operation. Statistics are coded using SQL queries and displayed.	5 points
All input were validated according to the expected input. The user must be prompted for corrections and stack traces must be logged to a file.	2 points
HTML Forms were coded properly. The layout is efficient and easy to use, and interface controls are intuitive.	2 points
The overall quality and complexity of the submitted project should be greater than any class demo practices and assignments	2 points
Bonus requirements	2 points

Some advice

- Have a consensus among the team members with regards to your data types and formatting of your data for your database. After your team agree with the database:
    - Start with small working code that can perform CRUD to the database
    - Save it as version X of your app
    - Add new feature
    - Repeat
- Take the perspective of a user using your application when you test it.
- Be sure that your tests cover all the CRUD operations.
- Try to run your project on another computer that was not in developing it or has never been used to run it prior to submission.
- Do not wait till the last minute to put together your submission, if your project artefacts are missing you will receive zero for them and if you hand in a poorly implemented useless trivial application that could have been programmed in a few hours you will also receive a zero.