

Advanced Web Programming
COMP1230
Assignment 2 Version 1.0

Objectives

Upon completion of this assignment, students will be able to

- Use server-side scripting language PHP to implement a database-driven solution.
- Design, implement and test a solution based on the given requirements.

Note

Please note that failure to follow the guidelines and/or sloppy, unprofessional writing and/or code will count against your grade. So be sure to read and follow the guidelines carefully. **The longer you wait, the greater the crisis you are likely to create for yourself.**

Overview

The accounting firm application you have implement requires an upgrade by integrating the flat file to the database. Your task is to replace the data storage from flat file to the MySQL database. There are no requirements for changing the interface, but there are a few new features you are asked to implement.

Requirements

All mentioned functionalities in the first project must stay intact. The only change to existing functionality is to covert the storage to the MySql database.

List of newly requested features.

- Implement authentication and authorization. The application must be password protected and unauthorized access to any page should be redirected to the login page showing an error message indicating login is required. (Please hard code the username and password in the login form for instructor access)
- Implement User management module: The module will be used to manage user account (company employee) who will have access to the application (from login form)
- Log all operations (add/update / view per model) on all four modules (User, Client, Notification, ClientEvent) **Format:** user , module name, action, DateTime, IP
- Add database backup feature. When backup (all table/data) is complete authorized users should be able to download the backup file. (each backup file must have a unique name)
- Restore the backup feature. An authorized user should be able to restore the database from backup file.

User characteristics (minim fields) to store:

- * First Name
- * Last Name
- * Email
- * Cell number
- * Position: Manager, Senior Accountant, Junior Accountant, Chartered Accountant, BookKeeper
- Username
- Password (must be hashed)
- Status – Suspended / Active
- Picture (file must be uploaded and resized – keep the original picture as well)

Rubrics

Your assignment will be marked according to how well it meets the following criteria:

- Originality, being able to explain your code
- Overall design
 - User-friendly (having a landing page, easy to navigate, ...)
 - Accessibility
- Code indentation
- Code documentation comment
- Reusability (use of functions)

Total mark: 100 %	
Any missing module from the first assignment result in losing 15 marks	
Any missing feature from the first assignment results in losing 2 marks	
Accessibility – easy to use	2
Error-free	15
Code documentation – comment	2
Valid HTML5 and CSS code	5
Variable and function name (descriptive names)	3
Database Restore	10
Database backup	10
Modular code – reusability – functions.php	7
Perform CRUD - User (create, retrieve, update and status update and picture)	15
Authentication and Authorization	15
Log all operations for User, Client, Notification, ClientEvent modules	16
Total	

Submission:

1. All your files **must** be uploaded to GBLearn account under the following directory:
public_html/comp1230/assignments/assignment2/
2. Login to <https://my.gblearn.com> and submit your **functions.php** file which must contain all the functions used in your assignment the due date
3. **Assignments not submitted on my.gblearn.com and or uploaded to a wrong directory will not be marked**
4. You must test your code before uploading to gblearn and submitting the functions.php file to my.gblearn.com . Please make sure there are no parse errors. Any parse error will result in 0 grade automatically. **No exceptions.**

10% Bonus Mark

Use a third party MVC framework such as Laravel to get a 10% bonus mark.