



---

## COMP1230 (Advanced Web Programming)

---

Final Project – Fall 2021

**Due date: December 10, 2021, by 11:59 pm.**

## Objectives

---

To give students the experience of working in a team and undertaking and completing a project, applying the skills learned throughout the semester.

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

---

An accounting firm must notify its clients about the due dates to file their HST return and the cooperate tax. The due date of the return is determined by the client's reporting period that can be monthly, quarterly and annually. CRA demands penalties and interest when an individual fails to file their taxes on time.

The accounting firm hires you to build an application to keep track of their clients with the option to inform their clients about their filing periods. The company would like to set a schedule for each client and send an automatic notification, either email or SMS, before the filing due date.

The application you are building must have the following management modules:

- Clients Manager – To manage client's information
  - Adding a new client
  - Modifying the client's information
  - Update status (inactive or active)
- Notification Manager – this module provides functionalities to
  - Add new notifications type (email, SMS, Push Notifications,...)
  - Modify a notification – update status (enable /disable) and update notification name.
- Client's Notification Manager - managing clients' notification types and their frequency.
  - Manage client's notifications (add / update)
  - Change status of a client's notifications. (enabled / disabled)
- Employee – Manage company's employee
  - Add a new employee
  - Modify the employee information
  - Update status (inactive or active)

Tip: Actually sending the notification is only a part of the bonus, and not required.

Tip: You will have 7 webpages in the end, the 4 modules explained here plus the Login page for employees, and Display Log Data page, and a page listing your group members which are explained below.

## Requirements

---

Only authorized users with the correct username and password can access the application. Except for a login page, there are no public pages. **You only need a login page for users that are employees, regardless of their position.**

For this application, you must use PHP and store the information in the database (MySQL)

Clients characteristics (minimum fields) to store:

- \* Company name

Tip: These are the minimum fields, so you can should also add fields for primary keys in these tables using a format like tablename\_id.

- \* Business number
- \* Contact's first name
- \* Contact's last name
- \* Phone number
- \* Cell number (to use for SMS notification)
- \* Carriers ([See the list and domain name here](#))
- HST number
- Website
- Status – Inactive / Active (default Active)

Notifications characteristics (minim fields) to store

- \* Name (to identify the notification)
- \* Type – example SMS, email
- Status – Enabled \ Disabled (default Disabled)

Tip: For example, the names of the notifications can be Inform of Due Date, Penalty Incurred, and Interest Accrued (based on description in the overview)

Clients events characteristics (minim fields) to store

- \* Client (client id)
- \* Notification (notification id)
- \* Start date/time
- \* Frequency – example, every 30 days/ every 120 days / every 365 days – the client should add any integer as the number of days.
- Status – Inactive / Active (default Active)

Employee characteristics (minim fields) to store:

- \* First Name
- \* Last Name
- \* Email
- \* Cell number
- \* Position: Manager, Senior Accountant, Junior Accountant, Chartered Accountant, Book Keeper
- \* Password (must be hashed)
- Picture (Good option to have - uploaded and resized plus keep the original image)
- Status – Inactive / Active (default active) – **Employee cannot log in when the status is inactive.**

Tip: If you (optionally) add a Picture field, you can use a varchar to store the name of the image and include the jpg file in your htdocs. Then you can query the image name from the database to display it on the employee webpage.

Tip: It is an auth page for employees, using the username and pass in that employee's record in the table. Hard code the username and pass for one employee for us so we can use your application.

- Implement authentication and authorization. Your application must redirect any unauthorized access to the login page with an appropriate error message. (Please hardcode the username and password in the login form for instructor access. You will receive zero grades if your instructor cannot log in. )
- Log (database) all operations (add/update/view per model) on all four modules (Employee, Client, Notification and ClientEvent) **Format:** Employee ID, module name, action, DateTime, IP

1, client, add client, 2021/11/12 11:31:56, 99.240.11.100  
 1,client, update client, 2021/11/12 21:31:56, 99.240.31.100  
 12,notification,diable status, 2021/11/22 11:21:16, 9.240.31.100,

Create a page to view the log data. (Include a link to this page on the navigation).

Tip: Use primary keys to connect the two tables here.

- Changing a client's status to Inactive should change the status of all the events that belong to that client, and vice versa
  - For example, you may implement logic in your code to prevent events from firing (like preventing notifications from appearing)
  - Or for example, you may update the status of all the client's events who have had a status change from active to inactive, or vice versa.
- You must use PHP for data validation (No JavaScript validation allowed –Might not be practical but trust me, you will learn more this way)
- Users must be able to perform the following operations for each module.
  - Add/ View / Edit / Search / Change Status.
- Any status change must be confirmed before proceeding with the action. "Are you sure?" message
- All mandatory fields \* must have value to update any record.

## 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
  - Accessibility
- Code indentation
- Code documentation comment
- Reusability (use of functions)

Total mark: <b>100 %</b>	
Accessibility – easy to use - Add PHP show source for ALL files	<b>3</b>
Error-free	<b>10</b>
Code documentation – comment	<b>2</b>
Valid HTML5 and CSS code- add the HTML5 validation script in all files that contain HTML	<b>5</b>
Variable and function name (descriptive names)	<b>3</b>
Modular code – reusability – functions.php	<b>7</b>
Perform CRUD - Client model (create, retrieve, update and status update)	<b>12</b>
Perform CRUD - Notification model (create, retrieve, update and status update)	<b>12</b>
Perform CRUD - Client and Notification model (create, retrieve, update and status update)	<b>12</b>
Perform CRUD - User (create, retrieve, update and status update and picture)	<b>12</b>
Authentication and Authorization	<b>12</b>
Log all operations for User, Client, Notification, ClientEvent modules	<b>10</b>
<b>Total</b>	<b>100</b>

## Bonus Mark

- Add database backup feature. Only an authorized user can download the backup file when backup (all table/data) is complete. (each backup file must have a unique name) **+5**
- Restore the backup feature. An authorized user should be able to restore the database from a backup file. **+5**
- Use a third party MVC framework such as Laravel **+10**
- Send SMS and email **+ 10**

## Submission:

1. Every member must upload the working application to their GBLearn account under the following directory: **public\_html/comp1230/assignments/assignment4/**
2. Create an HTML file called **member.html** and upload it to the assignment4 folder (create a link to this file from the application footer (open in a new window.)
  - a. Add the member information in a tabular format:
    - Student id
    - GBLearn URL to member assignment four folder
    - Email address
    - List of tasks completed by the member.

**Missing any of the above instructions will result in getting a zero mark.**

2. **Only one member** must log in to <https://my.gblearn.com> and submit the **functions.php** containing all the functions used in the assignment.
3. **Assignments not submitted on my.gblearn.com or uploaded to the wrong directory will receive a zero grade.**
4. The application must work for every member of your team on their GBLearn account. Any error will result in 0 grade automatically for the member. **No exceptions.**