# INFO5094 LAMP 2 Group Project

## Groups

**This is a group project. Each group is to consist of 3 to 6 students.**

Students may form their own groups and one member of the group is to email a list of the names and student numbers of the group to the Steve Sharpe by 5:00 PM Friday, January 29, 2021.

Any students who are not part of a group may be assigned to groups at the course instructor’s discretion. No changes in group membership will be permitted once the groups are set up.

The intent is that each student in a group will receive the same mark for each part of the project. However, each student’s participation in the project will be taken into account based on the participation form that is to be submitted in paper form for each project part submission. Students who do not participate in a part of the project will receive a mark of zero (0) for that part of the project. A copy of the form is at the end of this document.

## Due Dates

This project consists of 3 parts with the following due dates. Each part must be submitted to the correct FOL drop box by its specified due date and time. No late submissions or re-submissions will be accepted.

**Part 1** due by 11:59 PM on Friday, February 19, 2021 in the FOL submission folder

**Part 2** due by 11:59 PM on Friday, March 13th 2020 in the FOL submission folder

**Part 3** due by 11:59 PM on Friday, April 10th 2020 in the FOL submission folder

NOTE: Make sure that a submission is placed in the correct drop box. Submissions placed in the wrong drop box (whether that drop box is within this course’s FOL site or another course’s site) will not be accepted.

NOTE: Make sure that the correct file is uploaded. Students can download the files they upload and check that they are correct.

## Grading Information

* Any submission that produces fatal errors of any kind in any part of the submitted code will cause that part to receive a grade of 0. A fatal error is one that causes an application, or any part of an application to terminate abnormally.
* Any submission that produces warning or notice messages that are not fatal errors will be subject to penalties based on the impact of the defect that caused the message. The penalties will be applied to the marks for the portion of the application affected by each warning or notice message and may be partial or total loss of marks for that portion of the application.
* This project must be implemented using PHP, JavaScript, CSS and HTML only. No other programming or scripting language is permitted.
* **Part 2** of this project requires the use of AJAX after the initial page load. Failure to use AJAX as instructed in these Parts or sections will result in a deduction of 50% of the marks available for those Parts or sections. This includes the use of any library other than jQuery to perform the AJAX operations or using a server to client data format other than specified.
* All source files must contain a start of file comment identifying the file, its purpose and the author(s) of the file. An overall deduction of 10% of the total available marks for the part of this project will be deducted if any source file does not contain this comment.
* Students are required to use either the object oriented version of the mysqli or the PDO PHP database access library for all database operations. Use of any other database access library (including the procedural version of the mysqli library) is not permitted. An automatic deduction of 50% of the total marks available for the part of this project in which any other database library is used will be assessed for its use.
* Follow the project instructions carefully, failure to follow the instructions for this project will result in the loss of marks, whether the submitted code appears to do what is expected or not.
* The project will be marked using an Ubuntu 20.04 virtual machine. No other development environment will be considered during marking. Make sure your group’s project works correctly on Ubuntu 20.04
* This is a reminder of the academic integrity policies that are applicable for the program.

## Introduction

In this project, you will build a web application to help a company manage its human resource information. The application you will create is a prototype/proof of concept for a much larger and more complex system to be built in the future.

The 3 parts of this project will implement the following functionality:

1. Part 1
   1. Build a tool that generates a CSV file of random employee data for test purposes. We suggest an array of given names and an array of surnames that you can randomly select from. We suggest an output of 400 employees.
   2. Define the needed database.
   3. Upload the employee data in a CSV format text file to the web server. Assign each employee a unique employee number *if one has not been included in the CSV file*.
   4. Import the uploaded employee data into a MySQL database
   5. Produce a message for any duplicate employee names in the upload file. (It could happen.)
   6. List the raw employee data from PHP. This could be done in a simple <table> for now.

1. Part 2

Section C of this part of the project requires the use of AJAX. The jQuery library MUST be used to perform all AJAX operations.

* 1. Activate HTTPS in Apache2. The use of a self-signed certificate is okay here. From the PHP code, detect that HTTPS is active and if not, redirect the user to the https:// version of the site.
  2. Add login and logout pages that use PHP session
  3. Display the data for a selected employee

Using PHP create a web form that includes a method for selecting one employee from all employees stored in the database.

This method can either be a tabular list of available employee names with a radio button beside each employee name OR a select dropdown of employee names.

Form submission MUST use AJAX to send the selected employee to the server and to receive the employee data to display. The employee data must include at least the raw data items from the CSV upload.

You may use either HTML or JSON formats for the AJAX response from the server.

* 1. Edit the data for a selected employee.
  2. Allow data to be entered for a new employee.

1. Part 3
   1. Add CSS to improve the application's look and feel.
   2. Perform the following calculations for a given employee
      1. Annual salary or hourly rate history
      2. Earliest unreduced retirement date

Detailed instructions for Part 1 are given below. Parts 2 and 3 to follow.

Some of the instructions for this project are very explicit, telling you what to do and how to do it. Other instructions only tell you what to do, allowing you to choose how to do them. Be careful that you do what is asked for.

## Background Information

Here is some background information on the company. How flexible and configurable you make this system is up to you.

The company employs full-time (FT) and part-time (PT) employees. The PT employees are "regular," that is they work a set number of hours every week, up to a limit of 24 hours/week.

Employees are paid on a grid, from Level 1 to Level 9. Depending on their qualifications and experience, an employee could be hired to start anywhere from Level 1 to Level 5 on that grid. For each year of service, an employee progresses to the next level. No employee can exceed Level 9.

Each level has a corresponding FT annual salary and a PT hourly rate. The rates may change from time to time.

FT Annual Salary Grid:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Salary up to December 31, 2017 | Salary from January 1, 2018 to March 13, 2020 | Salary from March 14, 2020 to December 31, 2020 | Salary from January 1, 2021 to December 31, 2021 |
| Level 1 | $60,000 | $61,200.00 | $62,730.000 | $63,357.30 |
| Level 2 | $62,000 | $63,240.00 | $64,821.000 | $65,469.21 |
| Level 3 | $64,000 | $65,280.00 | $66,912.000 | $67,581.12 |
| Level 4 | $66,000 | $67,320.00 | $69,003.000 | $69,693.03 |
| Level 5 | $68,000 | $69,360.00 | $71,094.000 | $71,804.94 |
| Level 6 | $70,000 | $71,400.00 | $73,185.000 | $73,916.85 |
| Level 7 | $72,000 | $73,440.00 | $75,276.000 | $76,028.76 |
| Level 8 | $74,000 | $75,480.00 | $77,367.000 | $78,140.67 |
| Level 9 | $76,000 | $77,520.00 | $79,458.000 | $80,252.58 |

Hint: You'll want to create a MySQL table for the salary ranges. You won't really need to use it until Part 3. No hardcoding, please! The design is up to you.

PT Hourly Rate:

The corresponding PT hourly rate can be calculated by taking the annual rate, dividing by 261 days/year, and then dividing by 8.

Earliest Unreduced Retirement Date:

An employee can retire at the earliest of (a) the date they turn Age 65 (b) the day the sum of their age plus their service equals 85.

## Data File Format

The data for an initial employee upload is provided in CSV format files.

NOTE: the CSV format uses the comma (,) character to separate data items on a line, when a comma appears within a data item, that item must be enclosed in quotes in the file. The enclosing quotes are not part of the data and must not be put into the database.

The first line of a CSV file indicates the field headings.

For example:

**EmployeeID,Surname,GivenName,MiddleName,BirthDate,Gender,HireDate,InitialLevel  
42,"D'Asterix",Darryl,,1972-04-01,M,2005-09-23,5**

Note that the first CSV file uploaded may not have any employee IDs assigned. What other fields might not have values?

Question to consider: Have I left out any useful fields above?

## General Hints

1. Every group will probably have a different database design, which is perfectly OK. However, if your group decides that your original database design is not feasible in parts 2 or 3 of the project you may change it later.
2. If you do make changes to the database in part 2 or 3, go back and modify the file upload/import functionality and database creation script in part 1 to match the new database design. Failing to keep the upload/import functionality and database creation script up to date will prevent proper marking of part 2 and/or 3 since a new, clean database may be created and loaded with data when marking each part of the project.
3. Use relative paths for accessing files and directories in your site. Do not assume that your application will be installed in any particular directory.

## Common Specification Information

The information in this section is given to remind you of what is expected when processing data in a web application to preserve the integrity of the data used by the application. Failure to use this information will cost you marks in the parts/sections of the project where related operations occur unless there are specific notes in a part/section telling you otherwise.

When you are told to validate a data item from a web form, you must check for:

1. The data item exists.
2. The data item does not begin or end with spaces or tabs.
3. If the data item is a required item, it must not be empty.
4. The data item must not exceed its maximum length and/or must be within its specified range of values.

When inserting or updating database records, ensure that the record is properly protected from malicious content.

## Submission Requirements

Each part of this project is to be submitted to the appropriate FOL drop box by the deadline for that part of the project.

Each submission is to contain your entire project to date (the submission for part 1 will contain part 1, the submission for part 2 will contain parts 1 and 2, and for part 3 all 3 parts must be submitted.

Submit a single zip file containing all of the files and directories necessary to run your project. The zip file name must include the part of the project being submitted and the name or number of the group.

You may submit as many times as you like, but only the LAST file submitted will be marked.

## Part 1 Requirements

[marked out of 100]

1. Generate CSV file [this section is 15 marks]

* Define a site structure for your project such that the files and directories for each part of the project are within separate directories within the overall site directory. The overall site directory is to contain only an index.php file that produces a menu allowing users to access the functionality of the project. There may also be directories for common CSS, JavaScript and Images in the site directory. CSS, JavaScript or Images that apply only to one part of the project must be stored in directories within that part’s directory.[5 marks]
* Write a tool in PHP that will generate random employee data.[10 marks]

1. Define Database [this section contains 20 marks]

Create a MySQL database for your project to use. By all means use a tool such as PHPMyAdmin to assist you with this. This database needs to meet the following specifications:

* 1. [1 mark] The project's database must not contain any tables other than those required for the project.
  2. [1/2 mark] The name of the database must reflect the purpose of the database.
  3. [1/2 mark] The database must use the UTF8 character set.
  4. [2 marks] Create a non-DBA database user account for your application to use to access the database. This user account must only have access to the database for this project and publicly available databases.
  5. [13 marks] In the project database create the tables necessary to store employee data. When creating the tables, pay attention to the types and sizes of the data. The table schemas must include the definition of relationships between the tables.
  6. [3 marks] Create a MySQL script file that can be used to create or recreate your project’s database. (Hint: PHPMyAdmin can do it, if you know where to look.) This script file will be used to create an instance of your database for marking, MAKE SURE IT WORKS PROPERLY. This file must include the following, in the given order:
     1. Delete an existing instance of your database, if one exists
     2. Create a new instance of your database.
     3. Delete your database user account, if it exists
     4. Create your database user account
     5. Assign all privileges on your project database to your database user account
     6. Create your project’s database tables within your project’s database.

1. Upload the employee CSV file and import its contents into the database [this section contains 35 marks]

Write a PHP web program that will allow a user to select a CSV file containing the data about employees to upload to the web server and then import the contents of the file into the database.

Note: a CSV upload should delete any existing employee data in the table.

The PHP code must meet the following specifications:

* 1. [3 marks] The program must display a web form that allows the user to select a single CSV file containing path information and upload it to the server. The form must include a brief explanation about how to use the form.
  2. [1 mark] After submitting the form, the uploaded file’s existence in the temporary directory is to be confirmed and all of the metadata about the file is to be checked and/or validated.
  3. [2 marks] If there are any checks or validations that fail in b. above, the upload form is to be re-displayed with appropriate error messages. Processing must not proceed any further until a successful upload is achieved.
  4. [2 marks] Once a successful upload is achieved (no check or validation failures in b. above), the uploaded file is to be moved from its temporary location to a permanent location. During the move, the file is to be given a unique, but meaningful name. Keep track of the new name as you will need to use it to import the data into the database. The new name will also need to be associated with the path information stored in the database.
  5. [28 marks] Using the file in its permanent location, open the file and read it in one line at a time, populating the database as each line is read in. Validate each line’s fields to ensure that they contain valid data, based on the Data File Format section above. All database operations (open, insert, etc.) are to be checked for errors. If any validation or database error is detected, the entire path import is to be aborted with no records for the path being left in the database tables.

HINT: Trying to process CSV file lines using standard file input techniques is error prone, PHP includes some special functions for reading data from CSV files that can assist you. Look at [www.php.net](http://www.php.net/).

HINT: The PHP empty function won’t help with checking the validity of fields (especially numeric data), you will have to use other functions to help you check the correctness of the data from the CSV files.

1. The contents of the CSV file are to be inserted into the MySQL database.[15 marks]

HINT: Check out the use of MySQL transactions, they can help you meet the error handling requirement described in this section (mysqli and PDO have special functions for transaction processing).

1. In part D above, be sure to detect if there are any duplicate employee names.[5 marks]
2. Display the raw employee data [10 marks]

Knowing that Part 2 is coming, you could display the raw employee data using a <table> or perhaps a dropdown.

## Participation Form

The next page contains the group participation form. Each group must submit one (1) copy of this form on paper at the beginning of the first class following the due date for each part of this project. Electronic copies of the form WILL be accepted.

Each submitted form must contain the name of each member of the group who participated in the part of the project to which the form pertains. Along side each student’s name, he or she is to write a list of what work he or she performed for this part of the project and then sign the form in the provided space.

Any group member whose name does not appear on the form, or who does not sign the form will receive an automatic grade of zero (0) for that part of the project. In addition, poor participation by any group member may results in a reduced mark provided that other members of the group can provide documented evidence of poor participation.

Failure to submit a participation form for any part of the project at the beginning of the first class following the due date of any part of the project will be considered to mean that all students in the group participated equally in that part of the project.

### Group Member Participation Form

Group Name/Number Project Part Date

|  |  |  |
| --- | --- | --- |
| Member | Tasks Performed | Signature |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |