WESTERN SYDNEY UNIVERSITY

SCHOOL OF COMPUTING, ENGINEERING & MATHEMATICS
COMP2020: Technologies for Web Applications

Autumn 2022: Web Application Assignment

Due by 7:00pm on Friday 3rd June 2022 **Assessment Weight: 30%**

A. Requirements

a) ALL instructions given in this document MUST be followed to be **eligible** for full marks for the Web Application Assignment. This document has nine (9) pages in total including three (3) Appendices.
b) This assignment is **NOT** a group assignment; collusion, plagiarism, cheating of any kind is not acceptable. As part of your submission you MUST certify that all work submitted is your own. If you cannot honestly certify that the work is your own then do not submit the assignment. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information).
c) All assignment submissions will be checked for academic misconduct using the MOSS program from Stanford University.
d) Design the web pages with ease of navigation and operation, attractiveness, and accessibility in mind. Images other than those provided in the assignment zip file, if any, may also be used in the assignment.
e) Your code must guard against SQL injection and Cross Site Scripting attacks. That is, sanitise user input.
f) All assignment files are to be uploaded in the **project** folder in your TWA web site on the TWA server as follows:
   - php and html files in the `project` folder
   - css files in the `project/css` folder
   - images in the `project/images` folder
   - javascript in the `project/javascript` folder.

   Note: Compressed archive files (eg, zip, tar etc) are not acceptable and will not count toward submission requirements

g) Complete the full submission process before the due date and time. See section D for details of the submission process.
h) All styling and page layout must be achieved using CSS. The use of Bootstrap or other frameworks is not permitted.
i) jQuery or similar are not permitted.

For the problem definition described in section B you must
j) include your authorship details at the top of **each file** in coded comments.
k) **reference** all sources that you used for inspiration of your solution as per Section C of this document.
l) ensure that your web application renders correctly in Chrome and runs correctly from the TWA web server.

B. Web Application Assignment Details

B(i) - Background information and description

Dunder Mifflin Paper Company, Inc., is a paper supply company with its corporate office in New York and six branches in eastern states of the USA. The CEO of Dunder Mifflin is David Wallace. David Wallace believes that one of the problems facing the company is a lack of accountability of each staff member. To partially address this issue David requests the creation of a web-based Staff Performance Review system that can be used by corporate office to better track staff performance year on year. The Staff Performance Review system will first be trialled in the most dysfunctional, yet somehow most successful, branch in Scranton, Pennsylvania. Michael Scott, the Scranton branch manager and author of *Somehow I Manage*, does not like technology, and believes that his current system of using post-it notes on his computer monitor is sufficient. However, David Wallace insists that Michael's current system does sometimes [often] result in staff performance reviews becoming lost and there is a lack of ability to be able to accurately track [at all] when reviews are due to be completed and if they have been completed [never]. David believes that these general problems can be rectified by online lodgement of performance review forms which will enable better tracking by staff supervisors. For the web application project in TWA your task is to implement this simple online Staff Performance Review system that supports the Dunder Mifflin Staff Performance Review process. Michael has nicknamed the system Dunder Mifflin Super Snooper.

In this assignment, you will create a web-based application that supports the Dunder Mifflin performance review process. In general, a performance review involves the supervisor of an employee judging the performance of the employee based upon certain criteria. After the review has been completed by the supervisor, the employee that is the subject of the review is able to view it to determine whether they agree with the review. The system will allow employees to **view** their own reviews (past and present) and for supervisors to **carry out** performance reviews of staff members they supervise. The system will utilise a MySQL database, performancereview,

to store the data associated with each employee and their performance review(s). For this assignment you need to design and implement the PHP web pages that support the required functionality for the performance review system as described in section B(ii). The Performance Review database is described in section B(iii) of this document.

B(ii) – Functional Requirements

Your Web Application **must**

- be coded using HTML 5, CSS, JavaScript, and PHP as necessary. Note: all five files described below must be PHP files to achieve the server-side functionality.
- provide easy-to-use navigation for the user as described in the following page descriptions.
- provide the following page content and functionality for each page as described.

Employee login page (index.php).

The purpose of this page is to provide a login facility for employees (both supervisors and other staff) that need to use the Dunder Mifflin Staff Performance Review system. No other pages within the system should be accessible unless the employee has successfully logged in first.

Page content and functionality:

The page will:

1. have no direct hypertext links to any other pages of the system
2. include the following two paragraphs:

   > The Dunder Mifflin performance planning and review process is intended to assist supervisors to review the performance of staff annually and develop agreed performance plans based on workload agreements and the strategic direction of *Dunder Mifflin*.
   >
   > The Performance Planning and Review system covers both results (what was accomplished), and behaviours (how those results were achieved). The most important aspect is what will be accomplished in the future and how this will be achieved within a defined period. The process is continually working towards creating improved performance and behaviours that align and contribute to the mission and values of *Dunder Mifflin*.

3. include a postback login form that contains
   - a text box to capture the **employee ID**
   - a password box to capture the employee **password**
   - a submit/log in button.

When the form is submitted by the user the page will need to **authenticate** the credentials (**`employee ID`** and **`password`**) as supplied by the user in the login form against the records in the **`employee`** table of the database. Make sure that the user input is sanitised.

The page will allow or deny access to the relevant pages depending on the result of authentication as follows:

a. **Successful login attempt**: Successful authentication should automatically redirect the user to the **Choose Performance Review** page.

b. **Failed login attempt:** Unsuccessful authentication should automatically redirect to the login page so that the user may try to login again. An appropriate error message must be displayed to the user if the login attempt fails. The message display must be implemented using appropriate PHP server-side code; it is not to be a JavaScript alert (or any other browser side component). The message should be displayed in an appropriate location on the login page in a suitable colour.

Note on Employee Login Credentials:

The passwords that are stored in the **`employee`** table of the database are encrypted using the sha256 algorithm (the passwords are not salted - to decrease the complexity of implementation). A list of employee `employee IDs` and (plain text decrypted) `passwords` can be found in **User Credentials in appendix 3**.

Choose Performance Review Page (chooseReview.php)

The purpose of this page is to allow the logged-in employee to choose a performance review that they wish to view. The content of the page will be slightly different depending on the type of employee that is logged in (ie, whether they are a supervisor or not); the differences are described below. The page will list performance reviews from the database for the logged-in employee. The performance reviews may be either **completed** or **current** (ie, not completed). The logged-in employee will be able to choose a performance review to either **process** (for current reviews) or **view** (for completed reviews).

General content that must be displayed on this page:

- Navigation to the other pages of the system *as appropriate* including a **`Log Off`** link (this link must run the logoff.php page).
- current server date
- logged-in user's **name**

Content that must be displayed on this page **for all logged-in employees**:

The first section of the page will list all the performance reviews that exist in the database that are about the logged-in employee. To do this display the ***year of review*** and the ***date completed*** for each performance review of the logged-in employee. Display this data in reverse order of the year of review so that the most recent year is listed at the top. Each **year of review** is to be a **hypertext link** to the *View Performance Review* page. When clicked, the link must pass the **review id** of the review to the *View Performance Review* page.

Note 1: from the way that the list is displayed it must be clear to the user which performance reviews are completed and which are still current. This will make it easier for the user to decide which performance review they wish to view.
Note 2: viewing of a performance review does not occur on this page but on the *View Performance Review* page.

Content that must be displayed on this page for a logged-in employee that is a **supervisor**:

The second section of the page will only be displayed if the logged-in employee is a **supervisor.** This section will display **summary** details of all performance reviews of **employees in the database for whom the logged-in employee is the supervisor.**

For **each performance** review that belongs to **this logged-in supervisor** display the Employee's Surname, Employee's Firstname, year of review, review id, employee id, completed status, and date completed. Each **Employee's Surname** is to be a **hypertext link** to the *View Performance Review* page. When clicked, the link must pass the **review id** of the review to the *View Performance Review* page.

Notes 1 and 2 above also apply to this output data.
Note 3: this section of output should be displayed in two parts/sections/groups. Firstly, list the reviews that are current (not yet completed). Secondly, list those that have been completed. Within each of these groups display the lists in reverse order of review year. The user will then be able to find the performance review more easily they wish to view.

This page must only be accessible by a logged-in user. If a user tries to access this page and they are not logged in, they must be automatically redirected to the **logoff.php** page (which redirects to the login page) and then display an appropriate error message on the **login** page indicating that the error has occurred.

View Performance Review page (viewReview.php).

The purpose of this page is to display a single performance review that was chosen by the logged-in employee on the Choose Performance Review page. Both **completed** and **current** performance reviews can be displayed on this page but the way they are displayed is slightly different. The choice of performance review to display is made by the logged-in employee on the Choose Performance Review Page (described above). Completed performance reviews **cannot be modified** by the logged-in employee. Current (ie, not completed) performance reviews can be **accepted** by the employee on this page (see verification section below).

General content that must be displayed on this page:

- Navigation to the other pages of the system *as appropriate* including a Log Off link (this link must run the logoff.php page).
- current server date
- logged-in user's **name**

Content that must be displayed on this page for the **chosen** performance review:

a) **Employee information section**: Employee ID, Surname, First name, Employment mode, Review Year
b) **Ratings Information section**: the rating for each criteria: Job Knowledge, Work Quality, Initiative, Communication, Dependability
c) **Evaluation section**: Additional Comments, date that the review was completed.
d) **Verification section** (*this section is only displayed if the chosen performance review is **current**. This section is a small postback form. The data in point ii below can be changed by the user*):
    i. Display the following two paragraphs:

```
Thank you for taking part in your Dunder Mifflin Performance Review. This review is an important
aspect of the development of our organisation and its profits and of you as a valued employee.
```

By electronically signing this form, you confirm that you have discussed this review in detail with your supervisor. *The fine print: Signing this form does not necessarily indicate that you agree with this evaluation.*

ii. an appropriate input device that enables the employee to agree with/accept the above statement,
iii. a Submit button.

Submission of form

The form must use postback (ie, submit to the same page). The form can be submitted by clicking the Submit button. The Submit button updates the acceptance of the performance review by the employee in the database.

This page must only be accessible by a logged-in user. If a user tries to access this page and they are not logged in they must be automatically redirected to the **logoff.php** page (which redirects to the login page) and then display an appropriate error message on the **login** page indicating that the error has occurred.

Create Performance Review page (createReview.php).

The purpose of this page is to enable a supervisor to create and edit a performance review about a staff member that they supervise. Only supervisors are allowed to access this page. On this page the supervisor will choose the staff member they wish to create the review for and the year of review. After doing so the supervisor will be able to create a new performance review for the employee by entering data into a second form. Note that not all values in the second form are allowed to be changed by the supervisor – read the information below carefully.

General content that must be displayed on this page:

- Navigation to the other pages of the system *as appropriate* including a Log Off link (this link must run the logoff.php page).
- current server date
- logged-in user's **name**

Initial page content:

The page needs a postback form that contains

a) an appropriate input device to choose the employee for whom the performance review is to be created
b) an appropriate input device to enter the year of review
c) a submit button

Validation of user input on first form

Client-side input validation using **JavaScript** should be implemented as follows:

a) Both inputs are mandatory fields.
b) The year of review must be a number in the range 2022 to 2030

Note 1: Error messages due to the above client-side validation must be DOM notifications (ie, no alert boxes are to be used).

Secondary page content:

When the first form is successfully submitted a second postback form is to be displayed that contains information about the **chosen** employee. The form will need to contain:

a) **Employee information section** (*the data in this section is retrieved from the database and is **not allowed** to be changed by the user*): Employee ID, Surname, First Name, Job Title, Department Name, Review Year
b) **Ratings Information section** (*the data in this section **can be** changed by the **supervisor***): Ratings for: Job Knowledge, Work Quality, Initiative, Communication, Dependability.
c) **Evaluation section** (*the data in this section **can be** changed by the **supervisor***): Additional comments
d) **Verification section** (*the data in i below **can be** changed by the **supervisor***):
    i. an appropriate input device that enables the **supervisor** to indicate that the review is complete,
    ii. a Save Review (Submit) button.

Validation of user input on second form

Client-side input validation using **JavaScript** should be implemented as follows:

c) Each **rating** must be a number in the range 1 to 5. Not mandatory field.

Note 2: Error messages due to the above client-side validation must be DOM notifications (ie, no alert boxes are to be used).

Server-side input validation using **PHP** should be implemented as follows:

a) Each **rating** must be a number in the range 1 to 5. Not mandatory field.

b) **Additional Comments** for employee may only contain alphanumeric [“0” to “9”, “a” to “z”, “A” to “Z”] characters, spaces [“ “], hyphens [“-“], commas [“,“], period [“.”] and exclamation marks [“!”]. Not mandatory field.
c) **All form data must be sanitised**

Note 3: Error messages due to the above server-side validation must be DOM notifications (ie, no alert boxes are to be used).
Note 4: the database must not be updated if there is an error with the user input.

Submission of second form
The form must use **postback**. The form should only be allowed to submit when all data in the form is valid (according to the above **client-side** validation rules). The form can be submitted by clicking the Save Review button. The Save Review button enables the supervisor to enter data in the performance review and save it into the database for later editing. If the form data is valid then insert it into the database as a new performance review and display a message indicating that the performance review has been created on the current date, by the logged-in supervisor.

This page must only be accessible by a logged-in user. If a user tries to access this page and they are not logged in, they must be automatically redirected to the **logoff.php** page (which redirects to the login page) and then display an appropriate error message on the **login** page indicating that the error has occurred.

Logoff page (logoff.php).
The purpose of this page is to log the user off the system and automatically return them to the login page. The page must not display anything to the user. The page should remove all session variables and end the session (if sessions were used). It should then redirect to the login page. This page can be accessed from any of the pages as detailed above.

B(iii) – Database Description

1. Tables in the ***performancereview*** database are described in the **Data Dictionary in appendix 1**.

2. You have been provided with your own copy of the ***performancereview*** database on the TWA server. To access this database, you need to use a username and password. Details on how to connect to your copy of the database are in **appendix 2**.

3. The tables within the ***performancereview*** database have already been populated with some data. Use the `allTables.php` script to view the data (make sure you use the connection information as indicated above). This script is found in the project zip file.

4. A list of usernames and [plain text decrypted] passwords can be found in **User Credentials in appendix 3**.

B(iv) – HTML, CSS, JavaScript, and PHP files

All page styling is to be achieved using CSS. Create a CSS file called projectMaster.css. Add your CSS rules to this file. You may create additional CSS files if you wish. All CSS files should be uploaded to the `project/css` folder of your TWA website.

Some functionality in the web site will need to be achieved using JavaScript (eg, the client-side validation). Create a JS file called projectScript.js. Add your JavaScript to this file. You may create additional JavaScript files if you wish. The JavaScript files should be uploaded to the `project/javascript` folder of your TWA website.

All pages listed in section B(ii) will need to be PHP files to achieve the required server-side functionality. Additional PHP and HTML files may also be created as part of your solution if desired **if doing so does not contradict/negate the stated page requirements**. PHP and HTML files should be uploaded to the `project` folder of your TWA website.

C. Referencing

Referencing must follow the guidelines given in Section 2.5.3 of the TWA Learning Guide.

D. Submission Instructions

To submit your Assignment, you must do the following by the due date and time specified on page 1 of this document.

1. Upload all assignment files in the **project** folder in your TWA web site on the TWA server as follows:

    a. php and html files in the `project` folder

b. css files in the `project/css` folder

c. images in the `project/images` folder

d. javascript in the `project/javascript` folder.

2. Run the submission script located at

   `http://twaaut.cdms.westernsydney.edu.au/submit/submit.asp`

   As part of the submission, you will be prompted for your TWA website username and password. You will then be asked to read the WSU policy on misconduct and certify that work submitted by you is your own work. This action will be logged in a database for future reference and is deemed to be evidence that you claim that your work is original. Next, you will need to select from a drop-down list the Assessment you are submitting, eg, Assignment 1, and click the *Submit Assessment* button. The web page will then display a listing of the files you have submitted along with a receipt number. You should print this page for proof of submission.

E. Marking Criteria and Standards

The marking criteria and standards for the Web Application Assignment are published in Section 2.5.3 of the Learning guide and will be used to assess your assignment submission according to the specific weightings identified in the table below:

CRITERIA	WEIGHT
CODE FUNCTIONALITY/CORRECTNESS	60%
WEB PAGE DESIGN	25%
FORM DESIGN	10%
CODE READABILITY	5%

~~ Appendices start on the next page ~~

Appendices

Appendix 1 – Performance Review Data Dictionary

The `performancereview` database consists of 4 tables. Each table is described below. Primary and foreign keys are also indicated. A description of how to connect to your copy of the database is given at the end of the document.

Table Name: employee

This table provides details about employees that can access the system including their login credentials (employee_id and password) and their personal details which are used in performance reviews.

Note: passwords are encrypted using the sha256 algorithm. For testing purposes, a copy of the plain text passwords is given in the User Credentials document.

Your database credentials only have **Select** privileges for this table.

Column	Type	Null	Default	Comments
id	int(11)	No		This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically.
employee_id	varchar(8)	No		Staff member's unique employee ID number. This is used for authentication. *Primary Key*
surname	varchar(50)	No		Staff member's surname
firstname	varchar(50)	No		Staff member's first name
password	varchar(255)	No		Staff member's password. This is used for authentication. The value stored in this field is encrypted using the sha256 algorithm.
job_id	varchar(10)	No		Unique identifier for the job they are employed to undertake. *Foreign Key*
email	varchar(255)	No		Email address
department_id	varchar(20)	No		The id of the department to which the employee is assigned. *Foreign Key*
supervisor_id	varchar(8)	No		The employee id of the staff member's supervisor. *Foreign Key*

Table Name: department

This table identifies the departments that exist in Dunder Mifflin and the head of department (the head of department is also the supervisor for the department) for each department.

Your database credentials only have **Select** privileges for this table.

Column	Type	Null	Default	Comments
id	int(11)	No		This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically.
department_id	varchar(20)	No		Unique identifier for the department. *Primary Key*
department_name	varchar(100)	No		Name of the department
department_head	varchar(8)	No		The employee id of the head of the department

Table Name: review

This table gives details of an employee's performance review for a particular year.

Your database credentials have **Select, Insert** and **Update** privileges for this table.

Column	Type	Null	Default	Comments
review_id	int(11)	No		This is an auto incrementing number to uniquely identify a table row. **You do not insert this number into the database it is determined automatically.** Unique identifier for a review. *Primary Key*
employee_id	varchar(8)	No		Employee id for whom the review is about. *Foreign Key*
completed	char(1)	No	N	An indicator for whether the review has been completed. Possible values are Y or N
review_year	smallint(6)	No		Year for which the performance review was undertaken
job_knowledge	tinyint(4)	Yes	*NULL*	Rating [1 to 5] for job knowledge
work_quality	tinyint(4)	Yes	*NULL*	Rating [1 to 5] for work quality
initiative	tinyint(4)	Yes	*NULL*	Rating [1 to 5] for initiative
communication	tinyint(4)	Yes	*NULL*	Rating [1 to 5] for communication skills
dependability	tinyint(4)	Yes	*NULL*	Rating [1 to 5] for dependability
additional_comment	varchar(1000)	Yes	*NULL*	Additional comments that the supervisor wants to make about the employees performance in any of the ratings criteria.
accepted	char(1)	No	N	Indicates that the employee accepts ('Y') the performance review or 'N' otherwise
date_completed	date	Yes	*NULL*	The date upon which the performance review was completed by the supervisor
date_accepted	date	Yes	*NULL*	The date upon which the performance review was accepted by the employee

Table Name: job

This table provides details about the job that each employee can undertake at Dunder Mifflin.

Your database credentials only have **Select** privileges for this table.

Column	Type	Null	Default	Comments
id	int(11)	No		This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically.
job_id	varchar(10)	No		Unique identifier for the job title. *Primary Key*
job_title	varchar(100)	No		Job title

Appendix 2 – Connecting to your Performance Review Database

You have your own copy of the ***performancereview*** database. To access this database, you have to use a MySQL username and password. The following generic connection information can be used to connect to your ***performancereview*** database from your php scripts:

Database name: ***performancereview###***
Username: twa###
Password: twa###XX
Server: localhost

where ### is your twa site **number**, and XX refers to the first two characters of your twa site password.

For example, if your TWA site is twa999, and your password is abcd7890, then the following would be your connection information:
Database name: ***performancereview999***
Username: twa999
Password: twa999ab
Server: localhost

Hence, to connect to the ***performancereview999*** database from your php script you would require code similar to the following:

```
$dbConn = new mysqli('localhost', 'twa999', 'twa999ab', 'performancereview999');
if ($dbConn->connect_error) {
    die('Connection error (' . $dbConn->connect_errno . ')'
    . $dbConn->connect_error);
}
```

Note: The tables within the database have already been populated with data. Use the allTables.php script to view the data (make sure you use the connection information as indicated above in the script).

Appendix 3 – User Credentials for Performance Review web application

User Credentials for Dunder Mifflin Staff Performance Review system.

The passwords stored in the **password** field of the **employee** table are encrypted using the sha256 algorithm. Below are the plain text passwords for these users.

employee_id	Plain text **password**
DM001	bestBoss
DM002	athlead
DM003	beets
DM004	beesly
DM005	NardDog
DM006	bobVance
DM007	Actually
DM008	genious
DM009	Sprinkles
DM010	CostaRica
DM011	Jim
DM012	Carrot
DM013	planking
DM014	darryl
DM015	hide
DM016	pam
DMCEO	Suck It