



SWINBURNE  
UNIVERSITY OF  
TECHNOLOGY

**Swinburne University of Technology**  
*School of Science, Computing and Engineering Technologies*

**Assignment 1**  
**COS30020 Advanced Web Development**

Semester 2, 2021

**DUE ON:** 5:00 PM 9 September 2021

**Assignment Worth:** 40% of total marks

**Assignment requirement specifications:**

The aim of this assignment is to create a job posting system. This system will allow users to enter details about a job vacancy, and saved into a text file. These postings can also be searched using various criteria and all matched postings will be displayed on a web browser.

For this assignment, you will need to create all the appropriate HTML, CSS, PHP and text files. As with lab activities, you should save and test all your HTML, CSS and PHP files on Mercury in a directory "~/cos30020/www/htdocs/assign1". The text data file "jobs.txt" should be in the directory "jobposts" under "~/cos30020/www/data". The sample screen shots are provided only to illustrate the information requirement. It is **expected** that you create a much better user interface for this assignment.

**Task 1 – Home Page** `index.php`

This web page contains:

- Your name
- Your student SIMS number
- Your email address
- A statement: "I declare that this assignment is my individual work. I have not worked collaboratively nor have I copied from any other student's work or from any other source."
- A link to the post job vacancy into the system (`postjobform.php`)
- A link to the job vacancy searching process (`searchjobform.php`)
- A link to "about.php" page that tells us what you have attempted.

**Sample screen shot of the home page:**

**Job Vacancy Posting System**

Name: John Smith  
Student ID: 1234567  
Email: [1234567@swin.edu.au](mailto:1234567@swin.edu.au)

I declare that this assignment is my individual work. I have not worked collaboratively nor have I copied from any other student's work or from any other source

[Post a job vacancy](#)  
[Search for a job vacancy](#)

[About this assignment](#)

**Task 2 – Post Job Page** `postjobform.php`

This web page contains the blank form except for the date that enables a job posting to be submitted and saved. The requirements are as follow:

Req. 1. Job vacancy data contains:

- Position ID (Text input type. NOT NULL, should be unique e.g. P0001). The code is 5 characters in length. It must start with an uppercase letter “P” followed by 4 digits.
- Title (Text input type. NOT NULL, e.g. “IT Manager”). The title can only contain a maximum of 20 alphanumeric characters including spaces, comma, period (full stop), and exclamation point. Other characters or symbols are not allowed.
- Description (Text area type. NOT NULL, e.g. “This position promotes and supports the use of information technology throughout the organisation.”). The description can only contain a maximum of 260 characters.
- Closing Date (Text input type. NOT NULL, should initially contain the current date of the server in dd/mm/yy format e.g. 26/08/13 or 26/8/13. This can be edited by the user.)
- Position (Radio button type with 2 options: Full Time or Part Time)
- Contract (Radio button type with 2 options: On-going or Fixed term)
- Accept Application by (2 checkboxes, Post, Email)
- Location (Select option type, should initially be 3 dashes. The options are ACT, NSW, NT, QLD, SA, TAS, VIC and WA)

Req. 2. POST method for form submission is used.

Req. 3. Link to return to the Home page is provided.

**Sample screen shot of the job vacancy posting form:**

### Job Vacancy Posting System

Position ID :

Title :

Description:

Closing Date:

Position: ☐ Full Time ☐ Part Time

Contract: ☐ On-going ☐ Fixed Term

Application by: ☐ Post ☐ Mail

Location:

All fields are required. [Return to Home Page](#)

**Task 3 – Process Post Job Page** `postjobprocess.php`

This PHP page checks the input data, writes the data to a text file and generates the corresponding HTML output in response to the user's request. The requirements are as follow:

Req. 1. All fields are mandatory.

- The PHP page should not allow saving of job vacancy to the text file if any of these fields are not supplied or not valid (see rule in Task 1 – Step 1), and provide a proper error message to the user that includes links to return to the Home page and Post Job Vacancy page.
- The date must also be validated to conform to the dd/mm/yy format.
- Furthermore, the Position ID must be checked for its uniqueness within the text file.

Req. 2. Creation of the directory named “jobposts” on Mercury server inside “cos30020/www/data” to store the job vacancy text file (“jobs.txt”) is handled AUTOMATICALLY by the PHP script, if it does not exist.

- Req. 3. Each job vacancy record should be saved in the text file called "jobs.txt". Each record should be saved on one line, which means the next record should be saved in a new line ("n"). Each field in a record should be separated by a tab ("t"). The accept application by is stored as 2 separate columns.
- Req. 4. A confirmation message should be generated for this entry followed by a link to return to the Home page, once the job vacancy is stored successfully in the file,

#### Task 4 – Search Job Vacancy Page `searchjobform.php`

This PHP page contains a form that accepts a job title search string. The requirements are as follow:

- Req. 1. The search job vacancy page should contain an input text field for typing in the job title for searching the job vacancy information which was already saved in "jobs.txt".
- Req. 2. GET method for form submission is used.
- Req. 3. Link to return to the Home page is provided.

**Sample screen shot of search job vacancy page:**

### Job Vacancy Posting System

Job Title:

[Return to Home Page](#)

#### Task 5 – Search Job Vacancy Page `searchjobprocess.php`

This PHP page checks the job title search string, reads the data from the job vacancy text file, searches for the occurrence of the job title string in each job vacancy record and generates the corresponding HTML output in response to the user's search request. The requirements are as follow:

- Req. 1. The job title search string and the existence of the "jobs.txt" file are validated. If either validation fails, provide a proper error message to user that includes a link to return to the Home page and Search Job Vacancy page.
- Req. 2. All job vacancy records are searched. Within string matches are allowed. For example, "IT Manager" will also be matched to "Senior IT Manager"
- Req. 3. The summary of the requested job vacancy information should be generated including links to return to the Home page and Search Job Vacancy page if found. In cases where there are more than one job vacancy information located, all matches must be listed.
- Req. 4. If there is an error, a proper error message is generated that include links to return to the Home page and Search Job Vacancy page.

**Sample screen shot of search job vacancy information page:**

#### Job Vacancy Information

Job Title: IT Manager

Description: "This position promotes and supports the use of information technology throughout the organisation."

Closing Date : 26/8/2013

Position: On-going - Full time

Application by: Email

Location: VIC

[Search for another job vacancy](#)

[Return to Home Page](#)

**Task 6 – About Page** `about.php`

This is a PHP webpage for presenting what you have done based on the question provided.

Req. 1. List your answers in bullet point for each question. The requirements are as follow:

- What is the PHP version installed in mercury? (Use PHP function to generate this answer)
- What tasks you have not attempted or not completed?
- What special features have you done, or attempted, in creating the site that we should know about?

Req. 2. Provide a screen shot for the following:

- What discussion points did you participated on in the unit's discussion board for Assignment 1? If you did not participate, state your reason.

Req. 3. Provide a link to return to the Home page.

**Task 7 – Advanced Search Job Vacancy Page** `searchjobform.php`

Req. 1. Modify `searchjobform.php` to support additional search criteria

- a) By position
- b) By contract
- c) By application type
- d) By location

Req. 2. If no criteria were selected, it means any value applies.

**Task 8 – Order and Display Job Vacancy Result Page** `searchjobprocess.php`

Req. 1. Modify `searchjobprocess.php` to display job vacancy from the most future closing date until today's date.

**Submission & Assessment Process:**

Create your site structure as described below.

<code>assign1/</code>	assignment folder
<code>index.php</code>	Home page
<code>postjobform.php</code>	Post a new job vacancy page
<code>postjobprocess.php</code>	Process job vacancy data
<code>searchjobform.php</code>	Search job vacancy page
<code>searchjobprocess.php</code>	Process job vacancy search data
<code>about.php</code>	About page
<code>style.css</code>	CSS style sheet rules for your stylesheet
<code>style/</code>	Folder for any stylesheet images (optional)

Notes:

- PHP files should only be in the base "assign1/" folder – not anywhere else.
- All links to your files, including data files, should be relative.
  - Links must not be absolute, as these links will be broken when files are transferred for marking.
  - If links are broken, or an assignment does not run on the mercury server when we unzip it, the result will be a fail for this assignment.

**Assignment Submission:**

The assignment should be submitted as an individual work, using ESP (Electronic Submission Processor). ESP URL: (<https://esp.swin.edu.au/>).

Please compress all your PHP, support files and a video showcasing all the functionality as a zip file named "php\_assignment\_1.zip" to submit via the ESP tool. The video should clearly show your assignment working on Mercury. You can submit more than once via ESP and the current new submission overwrites the previous one.

*\* Make sure to test your program on the Mercury server before your submission.*

*\* Full marks for each task will not be awarded if there are errors or issues in the way the code is implemented and/or if usability is not considered properly.*

*\* If your assignment cannot run on Mercury, your result will be 0 marks for this assignment.*

### Provisional Marking Scheme

The assignment will be marked out of 100.

Task		Mark
Task 1:	Home Page ( <code>index.php</code> ) a. Page contained information shown in the screen shot b. Links worked using relative addressing	2
Task 2:	Post Job Vacancy Page ( <code>postjobform.php</code> ) a. Form contained all information shown in the screen shot b. Date field contained the server date c. Web page used the POST method d. Links worked using relative addressing	5
Task 3:	Process Post Job Vacancy Page ( <code>postjobprocess.php</code> ) a. Directory/file exist checked before directory creation b. Directory created using the correct path c. Data saved with appropriate tab and line break d. File opened and closed properly e. Input data validated f. Position ID is verified to be unique g. Appropriate error messages generated if errors occur h. Links worked using relative addressing	35
Task 4:	Search Job Vacancy Page ( <code>searchjobform.php</code> ) a. Form contained information similar to the screen shot b. Web page used the GET method c. Links worked using relative addressing	3
Task 5:	Search Job Vacancy Result Page ( <code>searchjobprocess.php</code> ) a. Search string and file existence validated b. Search performed correctly and summarises correctly c. File opened and closed properly d. Appropriate error messages generated if errors occur e. Links worked using relative addressing	20
Task 6:	About Page ( <code>about.php</code> ) a. All questions answered b. Links worked using relative addressing	5
Task 7:	Search Job Vacancy Page ( <code>searchjobform.php</code> ) a. By position b. By contract c. By application type d. By location e. Allow no criteria	10
Task 8:	Search Job Vacancy Result Page ( <code>searchjobprocess.php</code> ) a. Properly sort the result b. Display only those job vacancies that had not closed as of today	10
	Overall Quality a. User interface layout improved beyond the sample screen shot b. CSS used for colours and styles (must have good colour contrast) c. All pages validated (no partial mark) d. PHP codes commented and indented properly (see lab work)	10
Total		100

\* The marking scheme only provides a guide, it is expected that implementation demonstrates good coding and testing procedure.

\* Full marks for each task will not be awarded if there are errors or issues in the way the code is implemented and/or if usability is not considered properly.

\* If your assignment cannot be properly tested, your result will be 0 marks for this assignment. Note that JavaScript support in the browser will be disabled during testing.

**\*The assignment is your original work. If your assignment is not your original work, your result will be zero marks for this assignment.**

\* If your assignment cannot run on Mercury, your result will be 0 marks for this assignment.