COMP721 Web Development
Assignment 1

Semester 1, 2024

DUE ON: 11:59 PM Monday 29 April 2024
Assignment Worth: 25% of total grade

NB: Assignments will be accepted up to five (5) calendar days late, but a penalty of 5% per day late will be imposed on the submission. This is an individual assignment. Students are referred to the school's policy on plagiarism. A confirmed plagiarism case will incur zero mark to all the involved students.

Specifications:

The aim of this assignment is to create a diary system for a social networking site. This system will allow users to post their status and save it to a database table. These posted status details can also be retrieved using text matching and all matched status reports can be viewed in the order they are posted.

For this assignment, you will need to create all the appropriate HTML, PHP files and database table(s). For HTML pages, you are free to either use plain HTML+CSS or use CSS frameworks such as Bootstrap. You should save and test all your HTML and PHP files on webdev.aut.ac.nz server in a directory "*/htdocs/assign1*". Some screenshots are provided to illustrate the information requirement. It is expected that you make a much better user interface for this assignment.

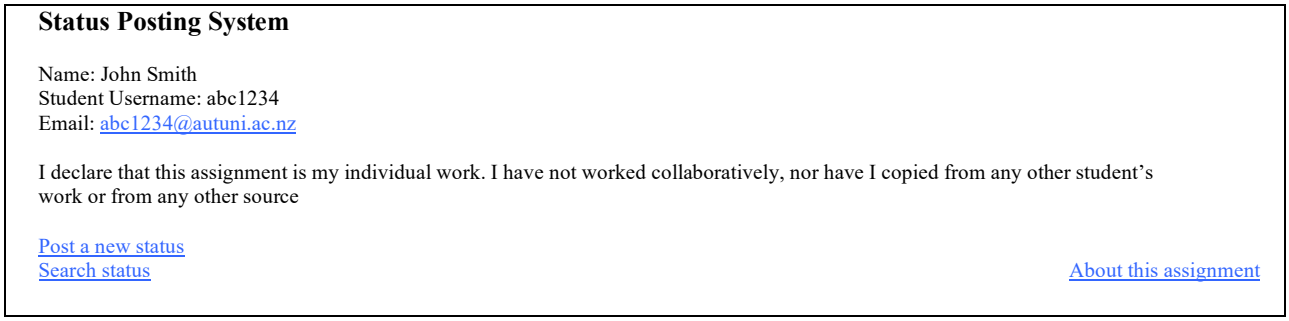
Task 1 – Home Page `index.html`

This web page contains:

- Your name
- Your student username
- 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 **relative** URL link (HTML anchor element) to the post status into the system (`poststatusform.php`).
    - The link must follow the syntax:
      …
    - The content of the anchor element must be: *Post a new status* as shown on the below sample screenshot.

- A **relative** URL link to the status searching process (`searchstatusform.html`)
    - The link must follow the syntax:
      …
    - The content of the anchor element must be: *Search status* as shown on the below sample screenshot.
- A **relative** URL link to "`about.html`" page that tells us what you have attempted (see Task 6).
    - The link must follow the syntax:
      …
    - The content of the anchor element must be: *About the assignment* as shown on the below sample screenshot.

Sample screenshot of the home page:



Task 2 – Post Status Page `poststatusform.php`

This web page contains the form that enables a status to be submitted and saved. Make sure all the required input elements on this page are inside a single form element, and clearly state the type, name and value of the input elements where necessary.

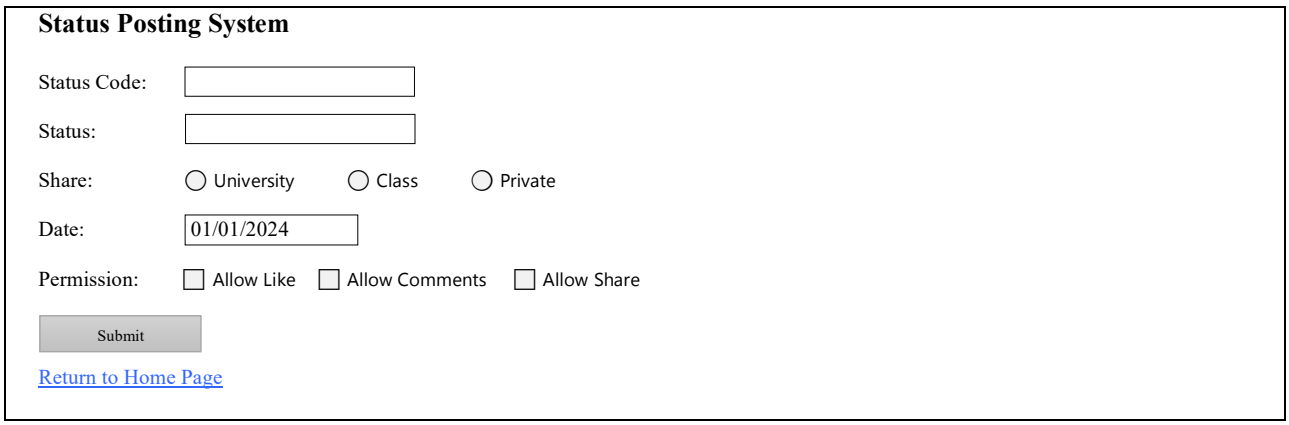
Step 1. Status data contains the following form elements:

- **Status Code**
  - Text input type.
  - Must set the `name` attribute to `"stcode"`: `<input type="text" name="stcode">`
  - NOT NULL.
  - Status codes should be unique.
  - The code is 5 characters in length. It must start with an uppercase letter "S" followed by 4 digits, e.g. `S0001`.
- **Status**
  - Text input type.
  - Must set the `name` attribute to `"st"`: `<input type="text" name="st">`
  - NOT NULL, e.g. `Doing my first assignment.`
  - The status can only contain alphanumericals, comma, period (full stop), exclamation mark and question mark. Other characters or symbols are not allowed. Note that spaces on their own cannot make up a status.
- **Share**
  - Radio button type.
  - Must include three options: `University`, `Class`, `Private`.
- **Date**
  - Text or date input type.
  - Must set the `name` attribute to `"date"`.
  - NOT NULL.
  - Must set the current server date as the default value.
    - *Hint: You can fetch the current server date using the `DateTime` class or `date()` function in PHP.*
  - Date must be in the format of `dd/mm/yyyy`, e.g. `01/01/2024`.
  - The date must be editable by users.
- **Permission**
  - Checkbox type.
  - Must include three selections: `Allow Like`, `Allow Comments`, `Allow Share`.

Step 2. POST method for form submission is used.

Step 3. Link to return to the Home page is provided and works as intended.

Sample screenshot of the status form (no style applied):



Task 3 – Process Post Status Page `poststatusprocess.php`
This web page checks the input data, writes the data to a database table and generates the corresponding HTML output in response to the user's request.

Step 1.

- Status Code and Status are mandatory fields.
- Must have PHP code to check if the database table exists and create the table if it doesn't. Remember to include any SQL commands in the sqlscript.txt file (refer to the marking scheme).
- The program should not allow saving of status to the database table if any of these fields are not supplied or not valid (see rule in Task 2 – Step 1).
- When needed, provide a proper error message (see the following instructions) to the user that includes links to return to the Home page and Post Status page.
    - If the user inputs a status code in a wrong format, the error message must contain at least one of these keywords: "S0001", "format". For example: *Wrong format! The status code must start with an "S" followed by four digits, like "S0001.*
    - If the user inputs a duplicated status code, the error message must contain at least one of these keywords: "unique", "duplicate", "exist". For example: *The status code already exists. Please try another one!*
    - If the user inputs a status in a wrong format or an empty string, the error message must contain at least one of these keywords: "format", "empty", "alphanumericals". For example: *Your status is in a wrong format! The status can only contain alphanumericals and spaces, comma, period, exclamation point and question mark and cannot be blank!*
- The date must also be validated using the `checkDate` function in the `DateTime` class.

Step 2. Once the status is stored successfully in the database, a confirmation message should be generated for this entry followed by a link to return to the Home page. Remember to include any SQL commands in the sqlscript.txt file (refer to the marking scheme).

- The confirmation message must contain at least one of these keywords: "success", "stored", "saved", "posted". For example: *Congratulations! The status has been posted!*

NB: Everything on this page including your confirmation and error messages must be inside a single HTML `div` element with a `class` attribute valued `content` (except the header and footer if you have them on your page).
E.g.:

```
<div class="content">
```

```
        <!--div class containing everything on the page except the
        header and footer-->
        <?php //embedded php code in the html file       ?>
</div>
```

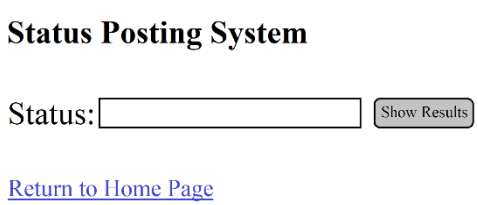
Task 4 – Search Status Page `searchstatusform.html`
This web page contains a form that accepts a status search input string. Users can type in a text string into the input text field to search for status record(s) containing the input string.

Step 1. The search status page should contain an input text field for typing in a status in order to retrieve any relevant records saved in the database table.
Step 2. GET method for form submission is used.
Step 3. Link to return to the Home page is provided.

The form elements of status search (input text field) and view status (button) must be identified by the name of `"Search"` and value of `"Show Results"` respectively.
E.g.:

```
<input type="text" name="Search">
<input type="submit" value="Show Results">
```

Sample screenshot of search status page:



Task 5 – Search Status Result Page `searchstatusprocess.php`
This web page retrieves an input string for the status search, reads data from the status database table, searches for the occurrence of the input status string and generates the corresponding HTML output in response to a user's search request.

Step 1.

- The status search string should not be empty.
- If validation fails, provide an appropriate error message to user that includes links to return to the Home page and Search Status page.
    - The error message must contain at least one of the keywords of "empty", "blank". For example: *The search string is empty. Please enter a keyword to search.*
- *Note: The input string should be used to match the* `status` *column instead of the* `status code` *column.*

Step 2.

- Before searching in the table, check if the table exists. If not, provide a message with a link directing to the post status page. The message must include the following wording: "No status found in the system. Please go to the post status page to post one."
- All the status records are searched based on keyword match.
- If there is no matching status, show a message to user that includes links to return to the Home page and Search Status page.
    - The message must contain at least one of these keywords: "not found", "no status", "not exist". For example: *Status not found. Please try a different keyword.*

- Remember to include any SQL commands in the sqlscript.txt file (refer to the marking scheme).

Step 3.
- When one or more matching records are found, the details of requested status information should be presented along with links to return to the Home page and Search Status page.
- If multiple records meet the search criteria, all matched records must be presented.

Sample screenshot of status search result information page:

Status Information

Status: Doing my first assignment
Status Code: S0001

Share: Public
Date Posted: August 22, 2017
Permission: Allow Like

Search for another status Return to Home Page

* Multiple search matches must be displayed when applicable. This screenshot only shows the case when one match is found.

NB: Everything on this page including your confirmation and error messages must be inside a single HTML div element with a `class` attribute valued `content` (except the header and footer if you have them on your page).
E.g.:

```
      <div class="content">
            <!--div class containing everything on the page except the header and
            footer-->
            <?php //embedded php code in the html file       ?>
</div>
```

Task 6: About Page `about.html`
This is a report webpage for presenting what you have done based on the question provided.

Step 1. List your answers in bullet point for each question. For readability, include the questions as well.
- What special features have you done or attempted in creating the website?
- Which part(s) did you have trouble with?
- What would you like to do better next time?
- What you have learnt along the way? Did you use any references or resources during this project? If so, please include the sources.

Step 2. Create a "Reset Database" button that drops the database table when clicked. The button request should be sent to "resetdb.php" for processing and a simple message showing if the request is successful should be returned.
Step 3. Provide a link to return to the Home page.

Submission & Assessment Process:
Create your site structure as described below.

assign1/	You must have this folder – case sensitive!
sqlscript.txt	Containing all SQL statements used in this assignment
index.html	Home page
poststatusform.php	Post a new status page
poststatusprocess.php	Process status data
searchstatusform.html	Search status page
searchstatusprocess.php	Process status search data
about.html	About page
resetdb.php	Reset database tables
style.css	CSS style sheet rules for your stylesheet
images/	Folder for any images used for page content (optional)
style/	Folder for any stylesheet images (optional)

Notes:

- PHP files should only be in the base "assign1/" folder – not anywhere else.
- Please make sure all links are working.
- CSS framework files and subfolders can be included

Assignment Submission:
The assignment should also be submitted as an individual work using the Assignment One link on Canvas (under the Assignments section). In this way, we can check the source code of your assignment project.

Please follow the steps below to submit:

1. Make sure your code works on the webdev server in the folder "assign1".
2. Download the folder 'assign1' (including all the files in it) to your local machine.
3. Make sure to put sqlscript.txt into the folder.
4. Zip this folder and rename it as `<Your AUT Network Username>.zip`. Upload it on Canvas to complete the submission.

** Make sure to test your program on the webdev server before your submission.*
** If your assignment cannot run on the webdev server, your result will be 0 marks for this assignment.*

Marking Scheme:
Please see next page.

Marking Scheme
Marking will be based on the detailed requirements as specified above for each task.
A general marking scheme is given below.
The assignment is marked out of 100.

Task		Mark
General:	**- Deduct 2 marks for each naming that does not follow the requirement (10 marks max)** - SQL script file (sqlscript.txt) containing all SQL statements used in this assignment such as CREATE TABLE, INSERT and SELECT. Statements should be separated by semicolons. If this file is not provided, 5 marks will be deducted.	
Task 1:	Home Page `(index.html)` a. Page contained information shown in the screenshot b. Links work as intended	2
Task 2:	Post Status Page `(poststatusform.php)` a. Form contained all information shown in the screenshot b. Date field shows the current date by default c. Web page used the POST method d. The link works as intended	8
Task 3:	Process Post Status Page `(poststatusprocess.php)` a. Existence of the database table checked. If database table doesn't exist, create the table. *E.g., query the INFORMATION_SCHEMA.TABLES.* b. Validation of input data a. Checks Status Code has a non-empty entry. b. Status Code follows the required format (refer to Task 2 Step 1). *Status code format can be validated on server side (using PHP) or on client side (using HTML pattern).* c. Checks status has a non-empty entry. d. Status follows the required format. e. Checks date has a non-empty entry. f. Date follows the required format. c. Status code is verified to be unique. *This requires database communication.* d. Error messages o Appropriate error messages generated when errors occur. o Link(s) work as intended in the error messages. e. Confirmation messages f. Appropriate confirmation message generated as expected. g. Link works as intended in the confirmation message.	40
Task 4:	Search Status Page `(searchstatusform.html)` a. The search status page contains information similar to the screenshot. b. GET method is used. c. Link works as intended.	5

Task 5:	Search Status Result Page `(searchstatusprocess.php)` a. Input search string is validated to be non-empty. A status with only spaces cannot be submitted and is considered invalid. b. Check the existence of the database table. If not, provide a message with a link directing to the post status page. c. Search results • Status search functions as intended. • When applicable, all matching records are displayed. • Links work as intended. d. Error message • When errors occur, appropriate error messages are generated. • Links work as intended.	25
Task 6:	About Page `(about.html)` a. All questions answered. b. Create a button that drops the database table. c. Link works as intended.	10
Overall Quality	Overall quality will be assessed based on the overall usability of the application including: a. Well-organized and well-styled user interface b. At least 10 CSS rules used for colours and styles	10
Total		100

	Undesirable Features	**Mark**
	1. Source code has poor or no indentation. 2. Source code has no or vague comments. An example of a bad comment is “this assigns a value to $a”. 3. Source code contains commented out code or code blocks. 4. Messages used for debugging and/or testing are included in the final version of the project.	–3 –3 –2 –2
Total		–10