



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

**Assignment 2**

**COS30020 Advanced Web Development**

Semester 2, 2024

**Due:** Friday 08:00 PM 18 October 2024  
**Worth:** 45% of total marks for the unit

**Background Description**

With the Status Posting System completed (Assignment 1), you are now tasked to develop the My Friend System. The system is a simplified social network application, and will make use of MySQL table creation, MySQL database access from PHP and PHP sessions.

**Part 1: Home Page and Database Table Creation for the My Friends System**

This part creates the Home Page that allows user to “sign up” or “log in” to the website. It also creates database tables and index page for the My Friends System. Two database tables will be created, namely “friends” and “myfriends”, and these tables will initially be populated with at least 10 and 20 sample records respectively.

**Task 1: Home page [index.php]** (See Figure 1)

Create a web page that display student information, and database tables and records if they do not exist.

Requirements:

- Page contents:
  - Your name
  - Your student ID 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.”
  - Three links:
    - ‘Sign up’ hyperlink that links to the sign up page of the site (Task 2),
    - ‘Log in’ hyperlink that links to the log in of the site (Task 3), and
    - ‘About’ hyperlink that links to the about in of the site (Task 7)
  - A message to indicate the success or failure in creating the two tables and records insertion.
- Other page functionalities:
  - Create database tables within your ‘s<7-digit id number>\_db’ database.
    - Only create tables if they do not currently exist in the database using the command “CREATE TABLE IF NOT EXISTS”.
    - Create the table ‘friends’ having the following fields:
      - friend\_id – the unique numeric identifier, e.g. 1023. Set this field’s properties as ‘integer’, ‘not null’, ‘auto\_increment’ and ‘primary key’.
      - friend\_email – the unique email identifier used for logging in to the system, e.g., user@gmail.com. Set this field’s properties as ‘varchar’, ‘size=50’, and ‘not null’.
      - password – the password for the account , e.g. ‘007’. Set this field’s properties as ‘varchar’, ‘size=20’, and ‘not null’.

- `profile_name` – the profile name for the account , e.g. 'Nice Person'. Set this field's properties as 'varchar', 'size=30', and 'not null'.
- `date_started` – the date in which the account was created, e.g. 2012-9-22. Set this field's properties as 'date', and 'not null'.
- `num_of_friends` – the number of friends of the account, e.g. '18'. Set this field's properties as 'integer', and 'unsigned'.
- Create the table 'myfriends' having the following fields:
  - `friend_id1` – the numeric identifier of a friend, e.g. 1023. Set this field's properties as 'integer', and 'not null'.
  - `friend_id2` – the numeric identifier of a friend, e.g. 1033. Set this field's properties as 'integer', and 'not null'. This is use to represent the friend identified by `friend_id1`, therefore it cannot have the same value of `friend_id1`.
- Populate the 'friends' and 'myfriends' tables with 10 and 20 sample records respectively. These sample records must be readable data, they cannot be random text such as 'asdf', 'aaa' or similar.

### My Friend System Assignment Home Page

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.

Tables successfully created and populated.

[Sign-Up](#)
[Log-In](#)
[About](#)

Figure 1: 'index.php'

## Part 2: Sign Up and Log In to My Friend System

This part allows navigation to all respective PHP pages that add and edit records on the 'friends' table. Delete function is not supported.

### Task 2: Sign Up page [signup.php] (See Figure 2)

Create a self-calling web page that contains the sign up form and validation script.

Requirements:

- Page contents:
  - Email address input field
  - Profile name input field
  - Password and confirm password input field
  - Register button using POST method for form submission
  - Link to return to the Home page
  - validates the input data, writes the data to the 'friends' table, set up the session variable(s) and generates the corresponding HTML output in response to the user's input.
- On form submit validate the input data, set up the session variable(s) and generates the corresponding HTML output in response to the user's input.
  - Server-side validation requirements
    - Email must be a valid email format and not exists in the 'friends' table
    - Profile must contain only letters and cannot be blank
    - Password must contain only letters and numbers and both input must match
  - If input is valid
    - Add the data into the 'friends' table, including the current server date and a value of 0 to number of friends
    - Set the session to a successful log in status
    - Redirect to 'friendadd.php' (Task 5)
  - If input is invalid
    - Display the form again with the original input, except the passwords, and an error message indicating which fields are wrong.
    - Display a link to the 'Home' page (Task 1)

The screenshot displays a web form titled "My Friend System Registration Page". The form contains four input fields: "Email" with the value "kangaroo@zoo.com.au", "Profile Name" with the value "Kangaroo", "Password" with masked characters, and "Confirm Password" with masked characters. Below the fields are two buttons: "Register" and "Clear". At the bottom of the form is a blue underlined link labeled "Home".

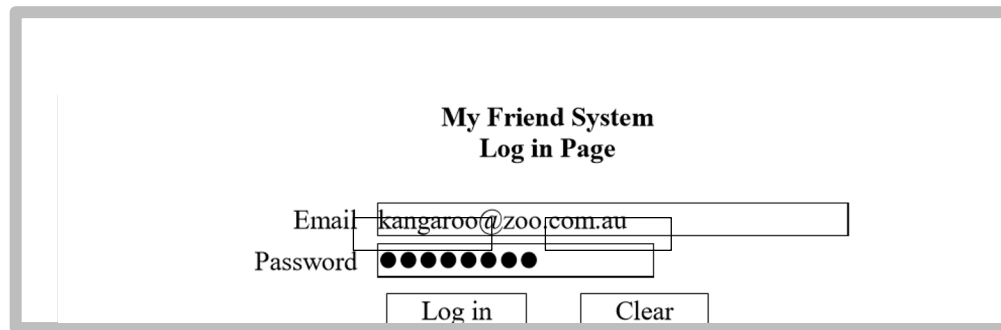
Figure 2: 'signup.php'

### Task 3: My Friend log in page [login.php] (See Figure 3)

Create a self-calling web page that contains the log in form and validation script.

Requirements:

- Page contents:
  - Email address input field
  - Password input field
  - Login button using POST method for form submission
  - Link to return to the Home page
- On form submit validate the input data, set up the session variable(s) and generates the corresponding HTML output in response to the user's input.
  - Server-side validation requirements
    - Email must exists in the 'friends' table
    - Password matches the value in the 'friends' table
  - If input is valid
    - Set the session to a successful log in status
    - Redirect to 'friendlist.php' (Task 4)
  - If input is invalid
    - Display the form again with the original input, except the password, and an error message.



The screenshot displays a web form titled "My Friend System Log in Page". It features two input fields: "Email" and "Password". The "Email" field contains the text "kangaroo@zoo.com.au". The "Password" field is filled with ten black dots. Below the input fields are two buttons: "Log in" and "Clear".

Figure 3: 'login.php'

### Part 3: User Interaction Pages of the My Friend System

This part allows navigation to all respective PHP pages that Add, Edit, and Delete the friend list.

#### Task 4: My Friend List page [`friendlist.php`] (See Figure 4)

Create web page that list the friends of the current logged in user with options to remove friends.

Requirements:

- Page contains:
  - profile name of the logged in user
  - a list of current friends arranged by profile name, showing
    - the profile name
    - unfriend button to remove them from the list and updates the number of friend count
  - links to 'Add Friends' (Task 5) and 'Log out' (Task 6)

**My Friend System**  
**Kangaroo's Friend List Page**  
**Total number of friends is 3**

Dingo	<input type="button" value="Unfriend"/>
Koala	<input type="button" value="Unfriend"/>
Wombat	<input type="button" value="Unfriend"/>

[Add Friends](#)      [Log out](#)

Figure 4: '`friendlist.php`'

#### Task 5: Add Friend List page [`friendadd.php`] (See Figure 5)

Create web page that list the all people who had registered, except friends of the current logged in user, with options to add friends.

Requirements:

- Page contents:
  - profile name
  - a list of registered uses arranged by profile name, showing the profile name
    - friend button to add them to the friend list, remove them from the current displayed list, and updates the number of friend count
  - links to 'Friend List' (Task 4) and "Log out" (Task 6)

**My Friend System**  
**Kangaroo's Add Friend Page**  
**Total number of friends is 3**

Penguin	<input type="button" value="Add as friend"/>
Platypus	<input type="button" value="Add as friend"/>
Possum	<input type="button" value="Add as friend"/>

[Friend Lists](#)      [Log out](#)

Figure 5: '`friendadd.php`'

**Task 6: Log out page [logout.php] (No HTML generated)**

Create script page that clears the PHP session (and associated data) and redirects the client browser (using a HTTP header response) back to the Home page.

Requirements:

- Other page functionalities:
  - Clears all session variables
  - Redirects to 'Home' page (Task 1). Tip: use `<?php header('Location: index.php'); ?>`

**Task 7: About Page [about.php]**

Create a report webpage for presenting what you have done based on the question provided.

Requirements:

- Page contains:
  - Listing of the answers to the following questions in bullet point.
    - 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?
    - Which parts did you have trouble with?
    - What would you like to do better next time?
    - What additional features did you add to the assignment? (if any)
  - Listing of links to the following pages
    - Friend List - `friendlist.php`
    - Add Friends - `friendadd.php`
    - Home page. - `index.php`
  - A screen shot of a discussion response that answered someone's thread in the unit's discussion board for Assignment 2?

## Part 4: Extra Challenge

---

**Note:** You will need to successfully complete the all four parts if you wish to achieve marks within the HD range.

This part is an opportunity to demonstrate deeper understanding of programming concept in the assignment using both PHP and MySQL.

### Task 8: Pagination for Add Friend page [`friendadd.php`]

Enhance `friendadd.php` that will only display 10 names at a time with provision to go for previous and next page.

### Task 9: Mutual Friend Count in Add Friend page [`friendadd.php`]

Display the mutual friend count for each profile name in the list.

### Submission & Assessment Process:

Create your site structure as described below.

<code>assign2/</code>	You must have this folder – case sensitive!
<code>index.php</code>	Home page
<code>login.php</code>	Login page
<code>signup.php</code>	Sign up (registration) page
<code>friendlist..php</code>	List friend page
<code>friendadd..php</code>	Add friend page
<code>logout.php</code>	Log out script
<code>about.php</code>	About page
<code>####.php</code>	Other PHP script only pages
<code>functions/</code>	Folder for any functions or includes for code reusability (optional)
<code>images/</code>	Folder for any images used for page content (optional)
<code>style.css</code>	CSS style sheet rules for your stylesheet (optional)
<code>style/</code>	Folder for any stylesheet images (optional)

Notes:

- PHP files should only be in the base “assign2/” 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.
- Additional files are allowed depending on adopted solution.
- JavaScript will be disabled in the browser during marking.

### Assignment Submission:

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

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

*\* If you do not submit all the files on ESP, your result will be 0 marks for this assignment.*

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





## Appendix: Provisional Marking Scheme

The following is the provisional marking scheme for Assignment 2.

Task		Mark
Part 1	Home Page and Database Table and Record creation	
Task 1:	Home Page ( <code>index.php</code> ) <ul style="list-style-type: none"> <li>a. Page contained information shown in the screen shot</li> <li>b. Links worked using relative addressing</li> <li>c. Database table created</li> <li>d. Record successful added</li> </ul>	12
Part 2	Sign Up and Log in to My Friend System	
Task 2:	Sign Up Page ( <code>signup.php</code> ) <ul style="list-style-type: none"> <li>a. Form contained all information shown in the screen shot</li> <li>b. Web page used the POST method</li> <li>c. Input data validated and added in the table</li> <li>d. Session variables successfully created</li> <li>e. Appropriate error messages generated if errors occur</li> <li>f. Links worked using relative addressing</li> </ul>	15
Task 3:	Log in Page ( <code>login.php</code> ) <ul style="list-style-type: none"> <li>a. Form contained all information shown in the screen shot</li> <li>b. Web page used the POST method</li> <li>c. Input data validated and checked against record in the table</li> <li>d. Session variables successfully created</li> <li>e. Links worked using relative addressing</li> </ul>	10
Part 3	User Interaction Pages of the My Friend System	
Task 4:	List Friend Page ( <code>friendlist.php</code> ) <ul style="list-style-type: none"> <li>a. Page contained information similar to the screen shot</li> <li>b. Unfriend button works</li> <li>c. Total number of friends updated</li> <li>d. Links worked using relative addressing</li> </ul>	15
Task 5:	Add Friend Page ( <code>friendadd.php</code> ) <ul style="list-style-type: none"> <li>a. Page contained information similar to the screen shot</li> <li>b. Add Friend button works</li> <li>c. Total number of friends updated</li> <li>d. Links worked using relative addressing</li> </ul>	15
Task 6:	Log out Page ( <code>logout.php</code> ) <ul style="list-style-type: none"> <li>a. Clears all session variables</li> <li>b. Redirect to home page</li> </ul>	2
Task 7:	About Page ( <code>about.php</code> ) <ul style="list-style-type: none"> <li>a. All questions answered</li> <li>b. Links worked using relative addressing</li> </ul>	10
Part 4	Extra Challenge – <b>no partial marks</b>	
Task 8:	Enable pagination in Add Friend Page ( <code>friendadd.php</code> )	8
Task 9:	Enable mutual friend count in Add Friend Page ( <code>friendadd.php</code> )	8
Overall	Quality <ul style="list-style-type: none"> <li>a. User interface layout improved</li> <li>b. CSS used for colours and styles</li> <li>c. All pages validated (<b>no partial marks</b>)</li> </ul>	5
Total		100

	Undesirable Feature	Mark
	1. Source code observes poor or no indentation	-2
	2. Source code has no or useless comments. An example of a bad comment is "this assigns a value to \$a"	-2
	3. Source code contains commented out code or code blocks	-2
	4. Application displays debugging or testing messages	-2
	5. Submitted zip file contains extra/unused files	-2
Total		-10

\* If your assignment cannot run on Mercury or is not placed in the right directory,, your result will be 0 marks for this assignment.