

COS30020 Advanced Web Development

Assignment 3: System development project 2

Background Description

With the Status Posting System completed (Assignment 2), 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.

- Page contents:
 - o Your name
 - Your student ID number
 - o 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:
 - o 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.
- o 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

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.

- Page contents:
 - o Email address input field
 - Profile name input field
 - Password and confirm password input field
 - o Register button using POST method for form submission
 - Link to return to the Home page
 - o 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.
 - o 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)



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.

- Page contents:
 - o Email address input field
 - o Password input field
 - o Login button using POST method for form submission
 - o 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.
 - o 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.



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:
 - o profile name of the logged in user
 - o 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
 - o links to 'Add Friends' (Task 5) and 'Log out' (Task 6)

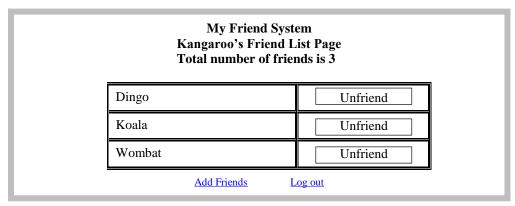


Figure 4: 'friendlist.php'

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.

- · Page contents:
 - o 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
 - o links to 'Friend List' (Task 4) and "Log out' (Task 6)

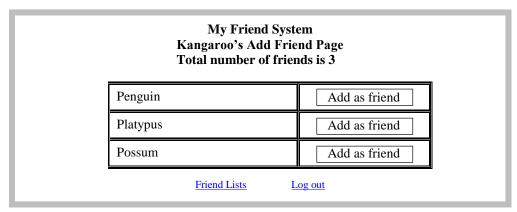


Figure 5: 'friendadd.php'

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:
 - o Clears all session variables
 - o 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.

- Page contains:
 - o 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 3?

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] (See Figure 6)

Enhance friendadd.php that will only display 5 names at a time with provision to go for previous and next page. Direct page number selection is not required in this task.

Requirements:

- Limit to only 5 names per page
- Support 'previous' and 'next' links, 'Previous' link must not appear if the first page is displayed, and the 'Next' link must not appear on the last page

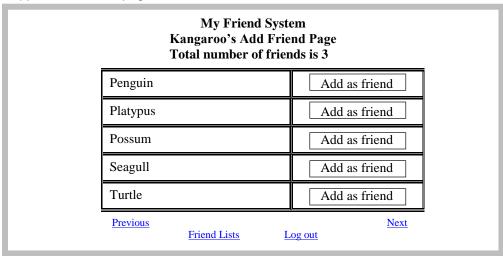


Figure 6: 'friendadd.php' with paging

Task 9: Mutual Friend Count in Add Friend page [friendadd.php] (See Figure 7)

Enhance friendadd.php to include mutual friend count information.

Requirements:

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

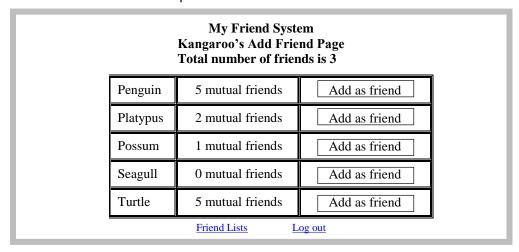


Figure 7: 'friendadd.php' with mutual friend count

Submission & Assessment Process:

Create your site structure as described below.

assign2/ You must have this folder – case sensitive!

index.php Home page login.php Login page

signup.php Sign up (registration) page

friendlist..php List friend page friendadd..php Add friend page logout.php Log out script about.php About page

####.php Other PHP script only pages

functions/ Folder for any functions or includes for code reusability (optional)

images/ Folder for any images used for page content (optional) style.css CSS style sheet rules for your stylesheet (optional)

style/ Folder for any stylesheet images (optional)