

# Clinical Information Systems (BIOM9450)

## Tutorial 5: Server-side Scripting PHP

**Due** 5 pm, 2<sup>nd</sup> November, 2020.

Submit all source code as well as your database file (project.mdb) and written report as a single .zip file to Moodle. Your report highlight the most important parts of your well commented code, and should contain screen shots of any messages generated by your code, and a description of your design choices and what motivated them.

### Objectives

Use PHP to dynamically retrieve and display information from an SQL database, and to insert information into the same SQL database.

### Background

So far you have learned how to:

- Created a webpage for a fictitious workshop website.
- Add JavaScript to perform client-side form validation of a registration form.

This week you will use PHP to capture and process the information entered in the registration form. When the registrant submits their form, you will check if they have previously been banned from registering for the workshop, store the registrant information permanently in an SQL database, and display it to the registrant after they submit their information.

### Tasks

- Use your JavaScript/HTML pages implemented in Tutorials 3 and 4 to capture the participant's registration details; requesting their email address, password, given and family names, date of birth, sex, and whether they wish to receive marketing material. You must perform the same form validations on the text entered, as per Tutorial 3.
- Design a table called 'Registration' in the 'project.mdb' Microsoft Access database which has been placed in your student folder on the 'G:' drive. You must keep this file in this location, as that is how the server knows how to find it when you use the ODBC Data Source Name (DSN) alias in your PHP code (explained next). When making a connection to the database, use the following PHP code:

```
$conn = odbc_connect('z1234567', '', '', SQL_CUR_USE_ODBC);
```

where z1234567 is your student number; this is an alias which the server uses to know where the database file is stored. We have already linked the DSN alias to the file on the server for you.

- Manually edit the database to insert at least one banned registrant – i.e., there should be a field in the 'Registration' table that indicates if a registrant is banned or not, or alternatively a separate table may be created that contains the details of the banned registrants.
- On submission of a valid form (by clicking the 'Register now!' button), you should use the POST HTML method to direct the user to a new page which confirms that their registration was successful (or not).
- If a user is banned (case-insensitive match on given AND family names, OR case-insensitive match on email), then the page should tell the user that they are banned from registering and that their registration is unsuccessful.
- If they are successful in registering, the confirmation page should greet them with a congratulating message. Their name, date of birth (with the days and month of their date of birth hidden with 'x's) and their email address should also be printed (see Figure 1(b)). The following additional tasks should also be performed:
  - Enter the successful registrant's details and the date on which they registered in the Access database; however, first check that the registrant has not already registered previously; if they have, display an information message to them (Fig. 2).
  - You may match people in the database by either a case-insensitive matching of both their given AND family names, OR case-insensitive matching of their email address.
  - Also display a complete list of all people (names only, and include on the list the current registrant) who are currently registered in the database.
- If any registrant is banned a customised message should appear (Fig. 3).

### Login details:

Email  No errors  
Password  No errors  
Re-enter Password  No errors

### User details:

First Name  No errors  
Last Name  No errors  
DOB (dd/mm/yyyy)  No errors  
Gender  No errors

☒ I want to receive marketing material

(a)

## Congratulations!

Your registration was successful

### Details:

Name: Heba Khamis  
DOB: xx/xx/1984  
Email: h.khamis@unsw.edu.au

### Registered Users:

Name	Email
Heba Khamis	h.khamis@unsw.edu.au
Nigel Lovell	n.lovell@unsw.edu.au

(b)

Fig. 1. Registration form completed correctly

### Login details:

Email  No errors  
Password  No errors  
Re-enter Password  No errors

### User details:

First Name  No errors  
Last Name  No errors  
DOB (dd/mm/yyyy)  No errors  
Gender  No errors

☒ I want to receive marketing material

(a)

## Duplicate Registration!

Bob Carr, you have previously registered!

### Registered Users:

Name	Email
Heba Khamis	h.khamis@unsw.edu.au
Nigel Lovell	n.lovell@unsw.edu.au

(b)

Fig. 2. Duplicate registration – this matched because [h.khamis@unsw.edu.au](mailto:h.khamis@unsw.edu.au) was already in the database.

### Login details:

Email  No errors  
Password  No errors  
Re-enter Password  No errors

### User details:

First Name  No errors  
Last Name  No errors  
DOB (dd/mm/yyyy)  No errors  
Gender  No errors

☒ I want to receive marketing material

(a)

## Registration Failed!

Robot Robot (robot@server.com), you are banned from registering for this workshop!

(b)

Fig. 3. Message printed for banned registrant.