SCC S2 2020: Major Project Assignment

Due by 11:59pm on Friday 25th September 2020 **Assessment Weight: 50%**

A. Requirements

a) ALL instructions given in this document MUST be followed in order to be **eligible** for full marks for the Web Application Assignment. This document has six (6) pages.

b) This assignment is **NOT** a group assignment; collusion, plagiarism, cheating of any kind is not acceptable. As part of your submission you MUST certify that all work submitted is your own. If you cannot honestly certify that the work is your own then do not submit the assignment. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information).

c) All assignment submissions will be checked for academic misconduct by the use of the MOSS program from Stanford University. Details on MOSS can be obtained from the MOSS web site http://theory.stanford.edu/~aiken/moss/

d) Design the web pages with ease of navigation and operation, attractiveness and accessibility in mind. Images other than those provided in the assignment zip file (if any) may also be used in the assignment.

e) All assignment source files are to be compressed into an archive and uploaded to the **Major Project (50%) submission folder** on vUWS prior to the due date and time. You may create subfolders within this folder if you wish.

f) All styling and page layout must be achieved using CSS. The use of Bootstrap or other frameworks is not permitted.

g) All client-side validation must be achieved using JavaScript. The use of JQuery or other client-side validation scripts is not permitted.

For the problem definition described in section B you must

h) include your authorship details at the top of **each** file in coded comments;

i) **reference** all sources that you used for inspiration of your solution as per Section C of this document;

j) ensure that your web application renders correctly in Chrome and runs correctly from the XAMPP 7.4.8 or 7.4.9 apache web server.

B. Web Application Assignment Details

B(i) - Background information and description

Online medical service booking has become quite essential during the COVID-19 lockdown.

In this assignment you will create a web-based application for online medical service booking similar to HotDoc named ***MedBook***. The ***MedBook*** web application will allow

- members and non-members to search by clinic name
- members and non-members to view lists of clinic names
- members and non-members to view the details from selected clinic
- members and non-members to make bookings
- members to view their own bookings

This is a very simplified web application and leaves out aspects that would be required in a medical service booking. Read the Functional Requirements section (**section B(ii)** of this document) to determine what is specifically required for ***MedBook***. The MySQL database that supports the functionality of ***MedBook*** is named ***MedBook***.

The ***MedBook*** database is described in **section B(iii)** of this document. You will need to design and create the PHP web pages for this assignment.

B(ii) – Functional Requirements

The ***MedBook*** web application **must**

a) be coded using HTML 5, CSS, JavaScript, and PHP **as necessary**.
Note: all files described below must be PHP files to achieve the server-side functionality.

b) provide easy-to-use navigation for the user as described in the following page descriptions.

c) provide the following page content and functionality for each page as described.

Search Page (search.php)

Page purpose/description:

This page is a search page to find medical clinics. The search page can be used by both non-members and members.

Page content:

The page will have the following content:

1. Navigation to the other pages of ***MedBook as appropriate*** including a ***Log Out*** link (if the user is logged in) or a ***Log In*** link (if the user is not logged in).
2. member username (if the user is logged in)
3. An HTML form (the search form) which contains

   - a text box to enter the keyword(s) for the search (mandatory field)
   - a submit button

   As indicated above the search text box is mandatory. That is, a search may be performed only if a search keyword has been entered.
4. A ‘search results’ section that lists clinic names and corresponding suburbs that match the keyword(s) entered in the form.

Page functionality:

The Search page will

1. validate that a search keyword has been entered in the search form (**using client-side JavaScript**). Do not allow the form to submit unless a search keyword has been entered. Only letters and numbers are accepted in the search form.
2. use **postback** for the search form,
3. sanitise the submitted keyword(s) to avoid possible sql injection then search the ***MedBook*** database for records that contain the keyword(s) entered in the search form. The search should check for matches with clinic names. The search may find zero, one or many matches depending upon the search term(s) entered. The output from the search depends on the number of matches found:
   a. **One or more records found:** generate a summary list of matching records with clinic names and suburbs underneath the search form. The user will be able to choose the match they want to view more detail about from this list (see point 4 below) by clicking on the match. The summary list should be **displayed in groups** of clinic names and suburbs with address showing:

i. The name of the clinics that have matched with the search keyword(s) by clinic name
ii. Suburbs of the clinics
iii. Address of the clinics

b. **Zero records found:** display an appropriate message underneath the search form instead of a list of results

4. enable the user to choose a clinic match from the search results. The names of the clinic in the search results list is to be a **hypertext link**. When clicked, the hypertext link loads the Clinic Details page (**clinicdetails.php**) which displays more specific details for the chosen match. Description of the Clinic Details page are given below.

General Page Notes:

The design of this page is open to your creativity, but it must appropriately display all required information in a clearly readable, well organised and usable format. You need to present the use of PHP functions to prevent SQL injections and XSS.

ClinicDetails Page (clinicdetails.php)

Page purpose/description:

The purpose of this page is to display to the user the content of the **chosen** clinic. The **ClinicDetails** page can be used by both non-members and members.

Page content and functionality:

The page will:

1. include navigation to the other pages of ***MedBook as appropriate*** including a ***Log Out*** link (if the user is logged in) or a ***Log In*** link (if the user is not logged in).
2. display the member username (if the user is logged in)
3. display the name of the clinic, suburb, address
4. If the booking is made by a logged-in member, use a **postback** form which contains
   - a selection list for 5 types of general medical service: “nutrition support”, “dental care”, “pharmaceutical”, “physical therapy” and “diagnosis care” (mandatory and multiple is enabled)
   - a date picker for the booking date (mandatory)
   - a time picker for the booking time (mandatory)
   - a submit button

   The form data will need to be validated that all input devices have a value on the server side before being inserted into the database. The selection list is allowed to have multiple items selected so multiple services can be included in one booking.

   Note: the date and time pickers are not supported input types on Safari so you need to use Chrome or Firefox to test the form

   The booking will be added to the booking list table for that member once the postback form is successfully validated and processed. A confirmation message should be displayed and the page will redirect to **BookingList** page for that member.
5. If the booking is made by a non-member user, use a **postback** form which contains
   - an input text for patient name (mandatory)
   - an input text contact number (mandatory)
   - a selection list for 5 types of general medical service: “nutrition support”, “dental care”, “pharmaceutical”, “physical therapy” and “diagnosis care” (mandatory and multiple is enabled)

- a date picker for the booking date (mandatory)
- a time picker for the booking time (mandatory)
- a submit button

Note: the date and time pickers are not supported input types on Safari so you need to use Chrome or Firefox to test the form
The booking will be added to the booking table for that user when the postback form is successfully validated and processed. Validation rule should be applied to contact number: it should be a ten-digit mobile number starting with 04.
Once the booking is added to ***MedBook*** database, a confirmation message should be displayed and the page will redirect to **Search** page.

General Page Notes:

The design of this page is open to your creativity, but it must appropriately display all required information in a clearly readable, well organised and usable format.

Booking list Page (Bookinglist.php)

Page purpose/description:

Booking lists are collections of medical bookings. Booking lists can only be accessed by logged in members. Bookings can be added to booking lists. Booking lists cannot be deleted. Booking lists belong to a specific member; logged in members cannot access any other member's booking list(s). Non-members cannot access this page and hence cannot create or change booking lists.

Page content and functionality:

The page will:

1. include navigation to the other pages of ***MedBook as appropriate*** including a ***Log Out*** link,
2. display the member username for the logged in member,
3. display a list of all bookings(that belong to the logged in member). Each booking is to be a hypertext link that will open the **ClinicDetails** page so that clinic name and suburb from the chosen booking may be viewed as described in the **ClinicDetails** page.

General Page Notes:

The design of this page is open to your creativity, but it must appropriately display all required information in a clearly readable, well organised and usable format.

Member Login Page (login.php)

Page purpose/description:

The purpose of this page is to provide a login facility for members of ***MedBook***. Some of the functionality described in previous pages is only available to users after they have logged in.

Page content and functionality:

The page will:

1. include navigation to the other pages of **MedBook *as appropriate***,
2. include a **postback** login form which contains
   - a text box to capture the **member username**
   - a password box to capture the **member password**
   - a submit/log in button

When the form is submitted by the user the page will need to **authenticate** the member credentials (`username` and `password`) as supplied by the member in the login form against the records in the `membership` table of the ***MedBook*** database The passwords from the membership table are encrypted with sha256 algorithm. You need to hash the password form field from the login form to compare with the encrypted password in the membership table. Suppose your password form field is named as “password”, you can retrieve the value by

```
$pword = $_POST[“password”];
$hpass = hash(“sha256”,$pword);
```

The $hpass is the hashed (with sha256 algorithm) password and you can use it as a part of WHERE clause in the SQL statement for credential matching.

The page will allow or deny access to the subsequent pages of ***MedBook*** depending on the result of authentication as follows:

a. **Successful login attempt**: Successful authentication should automatically redirect the member to the **Search** page.

b. **Failed login attempt:** Unsuccessful authentication should automatically redirect to the login page so that the user may try to login again. An appropriate error message must be displayed to the user if the login attempt fails. The message display must be implemented using appropriate PHP server-side code; it is not to be a JavaScript alert (or any other browser side component). The message should be displayed in an appropriate location on the login page in a suitable colour.

3. The provision for displaying error messages as detailed below

Member Login Credentials:

The passwords that are stored in the membership table of the MedBook database are encrypted using the sha256 algorithm (the passwords are not salted). A list of member usernames and plain text passwords can be found in the MemberLoginCredentials document.

Note: You need to present the use of PHP functions to prevent SQL injections and XSS.

Log Out Page (logout.php)

The purpose of this page is to log the member off the web application and automatically return them to the **Search** page. The Log Out page must not display anything to the user. The page should remove all session variables and end the session (if sessions were used). It should then automatically redirect to the **Search** page. This page can be accessed from any of the pages as detailed above.

B(iii) – Database Description

To connect to the ***MedBook*** database use the following in your php script

```
$dbConn = new mysqli("localhost", "root", "", "MedBook");
if($dbConn->connect_error) {
  die("Failed to connect to database " . $dbConn->connect_error);
}
```

Notes

- The tables within the ***MedBook*** database have already been populated with some data. Use the supplied `allTables.php` script to view the data (make sure you use the connection information as indicated above).
- Tables in the ***MedBook*** database are described in the ***MedBook* Data Dictionary** document.
- A list of user names and plain passwords can be found in the **Member Credentials** document.

C. Referencing

Referencing must follow the guidelines given in Section 2.5.2 of the TWA Learning Guide.

D. Submission Instructions

To submit your Assignment, you must do the following by the due date and time specified on page 1 of this document.

1. Compress all the resource files (HTML, CSS, Javascript, PHP) with the signed academic integrity form in an archive and upload it in the **Major Project (50%) submission folder** on the vUWS.
2. All of the submitted files should be tested and validated on your own XAMPP 7.4.8. or 7.4.9 apache and MySQL server before the submission.

E. Marking Criteria and Standards

The marking criteria and standards for the Web Application Assignment are published in Section 2.5.2 of the Learning guide and will be used to assess your assignment submission according to the specific weightings identified in the table below.

Code functionality/correctness	60%	Web Page Design	25%
Form Design	10%	Code Readability	5%