Web Application Assignment

Due: Sunday 11:59pm 12th of February 2023 – **Assessment Weight: 30%**

Note:

- Include comments for your student ID, Name, and Practical Class Time at the top of each source file created.
- All instructions given in this document must be followed in order to be eligible for full marks for the Web Application Assignment
- This assignment is **not** a group assignment; collusion, plagiarism, cheating of any kind will not be tolerated. Submitting your work to your TWA site signifies that the work uploaded is yours. If you cannot honestly certify that the work is your own then do not upload it to your TWA site. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information)
- All files must be uploaded to your TWA web site before submission due date
- Ensure all HTML written is valid. Use http://validator.w3.org to confirm before submission

---

Assignment Overview:

For this assignment, you are going to be build a B&B Booking System for a property portfolio. To do this, you are given a list of files that you will need to create. These will be a mix of HTML, CSS, Javascript and PHP files. Within some of your PHP files, you will be required to connect to and execute SQL statements on a database. This database will be given to you and will only be associated with your TWA site.

The concepts or skills within this assignment has been presented in lecture or has been part of a practical class exercises. You will be combining all these concepts and skills to build your basic B&B Booking System.

Design and Styling

For this major web assignment, you must design and style your own site. You can adopt what has been given in your Practical Class exercises or you can start from scratch. Either way, you must design and style your B&B Booking System so it is professional looking. Your chosen design and styling will be assessed.

Javascript and PHP Validation

All forms or where a user can provide input needs to be validated with both Javascript and PHP. HTML validation will not be accepted as a valid form of validation.

Web Application Assignment Database

You have been provided with your own copy of the database called **bnb_booking** on your TWA site. To access this database, you must use a username and password.

Continues next page

The following is a generic representation of the connection information to be used when connecting to your bnb_booking database (you do this with PHP code):

Database Name: bnb_booking###
Username: twa###
Password: twa###XX
Server: localhost

Where ### is your twa site number, and XX refers to the first two characters of your TWA site password.

For example, if your TWA site is twa**999**, and your password is **ab**cd7890, then the following would be your connection information:

Database Name: bnb_booking999
Username: twa999
Password: twa999ab
Server: localhost

Using this information, you will use similar code to that below to connect to your database:

```
$connection = new mysqli('localhost', 'twa999', 'twa999ab', 'bnb_booking999');
if($connection->connect_error) {
 // Echo out if it failed to connect to the database
 echo $connection->connect_error;
}
```

Once connected to your database, you will have access to the bnb_booking database and all its data. Figure 1 presents the Entity Relationship Diagram and the schema of your database.

You will can only access your database through your TWA site.

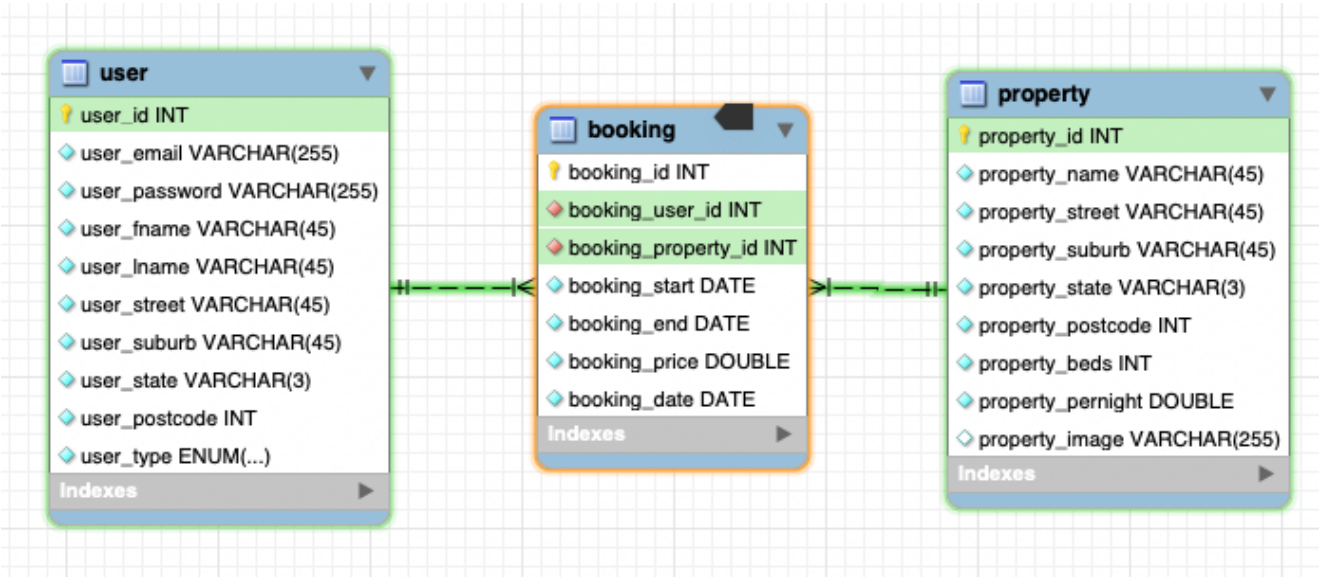

Figure 1

The bnb_booking Database Data Dictionary:

User Table			
Column	**Description**	**Type**	**Required**
user_id	This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically	INT	Yes
user_email	This is the email address of the user. It will be used as their login	VARCHAR(255)	Yes
user_password	This is an encrypted password. The encryption used is sha256.	VARCHAR(255)	Yes
user_fname	This is the first name of the user	VARCHAR(45)	Yes
user_lname	This is the last name of the user	VARCHAR(45)	Yes
user_street	This is the home street of the user, e.g. 4 Silly Street	VARCHAR(45)	Yes
user_suburb	This is the home suburb of the user, e.g. Parramatta	VARCHAR(45)	Yes
user_state	This is the home state of the user, e.g. NSW	VARCHAR(3)	Yes
user_postcode	This is the home post code of the user, e.g. 2000	INT	Yes
user_type	This is the type of user they are. There is 'general' for general user and 'admin' for admin user. 'general' is default.	ENUM('general', 'admin')	Yes

Booking Table			
Column	**Description**	**Type**	**Required**
booking_id	This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically	INT	Yes
booking_user_id	This is a foreign key for the user who place the booking.	INT	Yes
booking_property_id	This is a foreign key for the properter booked.	INT	Yes
booking_start	This is the start date of the booking reservation.	DATE	Yes
booking_end	This is the end date of the booking reservation	DATE	Yes
booking_price	This is the final price of the book. It should include GST, and any other calculations	DOUBLE	Yes
booking_date	This is the data the booking reservation was made.	DATE	Yes

Property Detail Table			
Column	**Description**	**Type**	**Required**
property_id	This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically	INT	Yes
propery_name	This is the name given to the property for display and searching purposes.	VARCHAR(45)	Yes
property_street	This is the street of the property, e.g. 4 Silly Street	VARCHAR(45)	Yes
property_suburb	This is the suburb of the property, e.g Sydney	VARCHAR(45)	Yes
property_state	This is the property state, e.g. NSW	VARCHAR(3)	Yes
property_postcode	This is the property post code, e.g. 2000	INT	Yes
property_beds	This is the number of beds at a property. Each bed can allow for 2 people.	INT	Yes
property_pernight	This is the price per night for the property.	DOUBLE	Yes
property_image	This is the relative path to the property image	VARCHAR(255)	No

Hint: Write your HTML, CSS first before you do any Javascript or PHP programming. Using this approach makes it easy to focus on design and allows you to *add* the programming logic later – it is considerably easier this way. Plus, if you fail to implement the Javascript or PHP you can still get marks for the HTML and CSS used to construct the page.

Navigation

The site must have a main navigation bar. Where you place this within your page is up to you. Every page should have this navigation bar. Pages which fail to have a navigation bar will lose marks. The navigation bar should display the following links:

- Home – links to index.php
- Properties – links to properties.php
- Bookings – links to bookings.php
  - This should only be present if the user is logged in
- My Profile – links to profile.php
  - *This should only be present if the user is logged in*
- Register – links to register.php
  - If the user is logged in this link should not be present
- Login – links to login.php
  - *If the user is logged in this link should change to 'Logout' (*logout.php*)*

Style and Design – styles.css

The styles.css file is your master CSS stylesheet for your site. Although you are allowed embedded and inline styles, a focus on external styles should be maintained. For maximum marks, your style and design should be responsive. This means it can be viewed both on a mobile screen and a desktop. Use the device preview available in Google Chrome under Developer Tools.

Javascript – actions.js
This file is to hold all your javascript required for your pages; all your form validations or actions in javascript.

Home page – index.php
This file is the home page to your site. This page only requires:

- A navigation bar
- A welcome message

List Properties – properties.php

a) This page lists the properties found within the database table *property*. The listing of the properties can either be in rows or a grid. Regardless of how the properties are displayed they need to show:
   - The property name
   - The number of beds
   - Address
   - Price per night
   - Image – images for each properties are found within the image directory of your assignment zip.

b) The page needs to allow the user to search for availability of properties between a date range (start and end date). If a property is not available for the entered date range the property is either removed or is marked as unavailable with its next availability shown. If a property is available a user can book the property by clicking a 'Book' button. This book button should take the user to the **book.php** page. To achieve the *availability* functionality a post-back must be used. This is where the page posts data back to itself and uses this data to refine the listing of properties. The data needing to be posted back is the start and end date. By using these two dates you can compare the start and end dates of properties to determine their availability.

c) When the page first loads the initial date range should be the present date and 3 days into the future.

Book Properties – book.php
A user must be logged in to see this page, if not they are redirected to login.php

a) After a user has clicked 'Book' on an available property they are taken to this page with the property ID, start and end date in the URL (GET request)

b) The details of the property of the chosen property, similar details to what is shown on the properties page, and the date range selected by the user are to be shown.

c) A total price calculation of the reservation needs to be made from the booking date range and the property per night cost. Remember it is nights, not days. For instance, if a user has entered in a date range of 5 days it would be 4 nights. A calculation of GST also needs to be added to the calculated price to show a total cost of the booking.

d) There is to be a confirm booking button which will take the user to **checkout.php**. It would be best to save the details of the booking in a session to be saved to the database after the checkout is completed.

Checkout – checkout.php
A user must be logged in to see this page, if not they are redirected to login.php

a) This page will present a form for payment details (part b of this page). Once the payment details have been validated the booking will be processed. Processing of the booking means the currently logged in user will have a new booking listed with all details of that booking (property booked) within the database.

b) For the payment details the following are required:
   - Credit Card Name

- Must be alphabetical characters and single inverted commas only
- Credit Card Number
  - Must be 16 digits long
- Expiry Date
  - Must be greater than current date
- CSV Number
  - Must be 3 digits long

These details are not to be stored, it is a mock payment form. However, the mock form must be validated before the order can be processed.

c) Once processed a thank you message is shown needs to be presented (**booking-confirmed.php**).

Booking Confirmation – booking-confirmed.php

A user must be logged in to see this page, if not they are redirected to login.php

a) This page should display a thank you message, along with the bookings details to show the user.

Booking History – Bookings.php

A user must be logged in to see this page, if not they are redirected to login.php

This page lists all the bookings placed by the currently logged in user. The page only needs the show following details for each booking:

- The Property
- The date the booking was made
- The start and end date of booking reservation
- The total cost of the booking

User Profile – profile.php

A user must be logged in to see this page, if not they are redirected to login.php

a) This page presents the currently logged in users' details within a form (except their user id, user type and password).

b) If the details have changed the user can update details and click submit/update to perform a **postback** and save the modified details to the database. This page requires the following validations before the **postback** can occur:
   - Basic form validation so all fields are required
   - The email address is a valid email address form
   - Post code must be 4 digits
   - State must be 3 characters or less (Hint: use a drop down)

User Registration – register.php

a) This is the user registration page. When users are registered through this page they become a 'general' user type. For this reason, your registration form must not have the user_type field displayed to the user and insert the value 'general' for user type. Also, the user_id field is not to be entered in by the user. The database automatically populates this field.

b) To process this form, you are to use a **postback** to submit the data back to **register.php**. Once the form is valid, a new user is saved to the database, and a new user session is created where you must redirect the user to the home page (index.php).

c) When processing the password field, it must be hashed (encrypted). The encryption algorithm to be used is sha256. The following code outlines how to use the hash function to perform this encryption algorithm:

Continues next page

```php
$password = "234jkldd";
$hashedPassword = hash('sha256', $password);
// $hashedPassword now contains:
// a5d95f8ebc8d8d592cfe772d33d8833909f526233d9ae12c32ada2bbea6a0bba
```

d) This form needs to be validated before submitting. The required validations are:
- All fields are required
- First name and last name must start with a capital letter
- The email address must be a valid email address
- Post code must be 4 digits only
- State must be 3 alphabet characters or less
- Password must be 8 characters long and must include at least 1 number [0-9]

User Login – login.php

a) This page will hold a login form containing a username (the user's email) and a password field. Although there are different types of users, both users can login through the same form.
- Remember, when checking if a user's password is valid you will have to encrypt it before you can check if the password is correct. This encryption is the same process found in **register.php**.

b) A **postback** must be used to login the user. If all credentials are valid, a user session is created and then the user is redirected to the home page (index.php). If the credential are invalid, the user is presented with an error informing them the username and password entered is incorrect.

There are four users already present in the database:
- User 1:
  - Username: john.smith@example.com
  - Password: password1
- User 2:
  - Username: sarah.smith@example.com
  - Password: password2
- User 3:
  - Username: margret@example.com
  - Password: password3
- User 4:
  - Username: admin@example.com
  - Password: password4

Logoff – logoff.php

This page is not a presentation page. When a user navigates to this page their user session is destroyed. Everything within should be destroyed. Once destroyed, the user is then redirected to the home page (index.php)

Submission Instructions

All files are required to be on your TWA site under the /project directory before the submission due date. Submitting your work to your TWA site signifies that the work uploaded is yours. If you cannot honestly certify that the work is your own, then do not upload it to your TWA site. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information).

IMPORTANT: Submission Script Details

1. Prepare a README.txt file that lists your own created usernames and passwords for markers to test your website. If you have some special notes for the markers, you can also include them in this file.
2. Upload all your practical files in the **/project** folder in your TWA website. Failure to upload the files to the correct location could result in a Fail Non-Submit. No zips are allowed. Your site must work on your TWA site without modification.
3. Run the submission script located at: https://twasum.cdms.westernsydney.edu.au/submit/submit.asp
4. As part of the submission, you will be prompted for your TWA website username and password. You will then be asked to read the WSU policy on plagiarism and certify that work submitted by you is your own work. This action will be logged in a database for future reference and is deemed to be evidence that you claim that your work is original. Next, you will need to select from a drop down list the 'project', and click the "Submit Assessment" button. The web page will then display a listing of the files you have submitted along with a receipt number. You can take a photo/screenshot of or print this page for proof of submission.

Notes:
The submission script may be run more than once if needed. A record of all submissions will be automatically created by the submission script in a set of web log files and in a database for future verification purposes. This submission script copies the files from your web site to a duplicate version on a marking web server. You will not be able to access the marking web server.