

BIT608 Web Programming

Assessment 2

Weighting

40%

Learning outcomes

2. Develop server-side web content using contemporary scripting languages that interacts with the client side.
3. Utilise applicable standards and protocols and address the security issues associated with web development.

Instructions

Complete and submit your assessment according to the Open Polytechnic's [Assessments webpage](#). This includes information on academic integrity, word limits and referencing.

- Include your name, student number and the assessment number.
- Number your pages.

Submission

- Submit your assessment in one zip file.
- Submit your work through your iQualify course.
- Emailed assessments will not be accepted.
- You will receive an automated notice following submission.

By submitting your assessment, you confirm that it is your original work.

Overview

The goal of the three assessments is to build a web application based on the provided design brief. You'll be focusing on various aspects of developing and publishing a web application by using a case study that reflects a typical set of requirements.

This assessment involves interpreting the requirements given in the design brief for the web application.

Resources you will need to complete the three assessments:

- the Motueka Bed and Breakfast Design Brief
- the partially completed example system source code is in *Motueka.zip*
- the *Motueka.sql* file in *Motueka.zip* needs modification to incorporate the required changes to the system database.

Note:

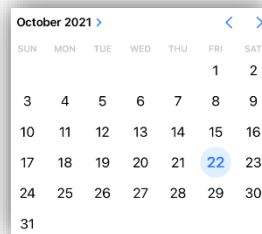
Ensure that your file contains the 'www' website document root folder containing all your PHP pages. Ensure the *Motueka.sql* file in your zip is the updated version that includes all your changes to the database. Ensure your app uses the default MySQL username and password (both 'root') so the grader can run your code if necessary.

Part 1: Client-side development from requirements

Read *The Motueka Bed and Breakfast Design Brief* before completing the following tasks.

Task 1

Integrate a JavaScript date & time picker into the *Make a booking* form you created in Assessment 1. You may decide to use vanilla JavaScript or a library to implement this requirement into your web application. Date and time pickers provide a visual means of selecting a date. If you have already used the rather basic HTML5 date and time picker, remove this and replace it with a JavaScript date and time picker.



Example of a date picker

(10 marks)

(Part 1 total: 10 marks)

Part 2: Server-side database development and security

This part involves interpreting the requirements for server-side purposes, handling the data and some security aspects. For this part, you are focusing only on the server-side aspects of the design brief.

In Assessment 1, Part 2, Task 1, you prepared a 'booking' table with five columns (Field Name, PK/FK, Data Type, Size, Required). Use that information to create the database tables for storing the various pieces of information coming from the forms. Based on the page views provided in the design brief, there are currently three subsystems (Bookings, Customers and Rooms).

The SQL DDL file (*Motueka.sql*) provided in the *Motueka.zip* file creates and populates the Customer and the Room tables. You need to add the Bookings table and sample data.

Task 1

Use the table created in Assessment 1, Part 2, Task 1, to produce the SQL DDL to create the *Booking Database Table*. There should be at least six fields just for the Bookings data (including the primary key) and two additional fields for foreign keys to link the *Rooms Table* and the *Customers Table*. These relationships will be necessary for the SQL simple joins.

(10 marks)

Task 2

Add at least two sample data records using SQL statements in the DDL file. Ensure that the foreign keys relate to existing primary keys in the test data for the other tables.

Ensure that there are no errors when the DDL file runs and that it creates the expected table and data.

(10 marks)

Task 3

Your web application could potentially use a payment function to accept credit cards or integrate PayPal and other payment methods. It is crucial to ensure that access to the server-side is restricted and controlled.

Describe how you can use encryption, digital certificates, and passwords to protect parts of your web application.

(10 marks)

(Part 2 total: 30 marks)

Part 3: Server-side PHP development – Managing bookings

In this part of the assessment, you will implement the server-side PHP pages for the booking system.

- For the purposes and scoping of this web application exercise, **you will not include any user access control apart from the basic login.**
- Use the existing PHP files for the Room and Customer subsystem as example code to help with implementing the additional pages.
- Use the HTML pages you created in Part 1, Question 1 above, as the starting point for your PHP pages.

Task 1: List bookings and view bookings (R of CRUD)

- a) Create a PHP web page to list the current bookings. Add three action links at the end of each booking line that let the admin manage a booking (view, edit, delete).

(5 marks)

- b) Create a PHP web page that shows a SINGLE booking. Ensure this page links back to the listing page above.

(5 marks)

Note: Refer to *listrooms.php* and *viewroom.php* for example code.

Task 2: Delete bookings (D of CRUD)

Create a page that lets the admin preview a booking and then request to delete or cancel the booking. Ensure this page links back to the listing page above.

Note: Refer to *deleteroom.php* for example code.

(5 marks)

Task 3: Edit bookings (U of CRUD)

Create a page that lets the admin edit an existing booking and then request to delete or cancel the booking. Ensure this page links to the listing page above.



Note: Refer to *editroom.php* for the example code.

Please ensure any data sent by the above forms are checked, cleaned, and verified before saving it to the database.

(5 marks)

(Part 3 total: 20 marks)

Part 4: Server-side PHP development – Create bookings

Locate your *Make a booking HTML* form created in Part 1, Question 1 to aid in creating this page.

At this stage, the *Make a booking* form should potentially have jQuery for handling AJAX and the date picker on the client-side.

This form now has two features:

- making a booking
- searching room availability.

Task 1: Create bookings (C of CRUD)

Create a PHP web page that lets a user create a booking.

Note: Refer to *addroom.php* for example code. This feature will require you to list the current rooms for the select (drop-down) input. The room information is retrieved from the rooms table and used to populate the select (drop-down) input.

When you save the booking data, you will need to record the *Room ID* and the *Customer ID* along with the booking, which will link a *Room* and a *Customer* to the booking. A Customer MUST exist to create a booking.

(10 marks)

Task 2: Check available rooms (R of CRUD)

Include a *search for room availability form* in the *Make a booking* webpage so that users can search for rooms available within a specified date range.

Note: Refer to *listcustomers.php* and *customersearch.php*, for example Ajax usage. Use the page view *Search for room availability* within the design brief as a guide.

Clicking the search button starts an Ajax request to return a list of available rooms. Use an SQL query that lists only the available rooms, similar to the SQL example below.

```
SELECT
FROM room
WHERE roomID NOT IN (SELECT roomID
                     FROM booking
                     WHERE checkin >= [fromdate]
                     AND checkout <= [enddate])
```

You will still need to fit this SQL to your code. Make sure you replace [fromdate] and [enddate] in the SQL with relevant PHP variables.

The *Search availability* feature uses Ajax, so you need to create an additional file, *roomsearch.php*, in PHP for the Ajax handler. This handler only needs to return the above query results so that the Customer can identify available rooms.

(20 marks)

(Part 4 total: 30 marks)

Part 5: Server-side PHP development – Logging in

Task 1: Logging in

Create a page with PHP that lets a user log in to the web application. Ensure that the login can distinguish between customer and admin logins and give page access accordingly.

- Require the user to log in first before accessing the booking pages.
- Provide an option for logged in users to log out again.
- Encrypt any passwords stored in the database for security purposes.

Note: Refer to *checksession.php* for an example session handling code and *index.php* on how to start using it.

The login page should interact and work with the existing BnB code from the *Motueka.zip* folder.

(10 marks)

(Part 5 total: 10 marks)

Marking schedule

Part 1: Client-side development from requirements Task 1	8 – 10 Marks	6.5 – 7.5 Marks	5 – 6 Marks	4 – 4.5 Marks	1 – 3.5 Marks
Task 1	A suitable embedded date & time picker works faultlessly as intended.	A suitable date & time picker is embedded in the page but may have one apparent fault.	A suitable date & time picker is embedded in the page but may not fully function correctly.	Any date & time picker is embedded in the page but does not fully function correctly.	An attempt is made to include any date & time picker which has no function.
Part 2: Server-side database development and security Task 1	8 – 10 Marks	6.5 – 7.5 Marks	5 – 6 Marks	4 – 4.5 Marks	1 – 3.5 Marks
Task 1	The SQL DDL creates the <i>Booking Database Table</i> . With no errors. At least six fields plus the two additional fields are created.	The SQL DDL creates the <i>Booking Database Table</i> . With one error. Five to six fields plus at least one of the additional fields are created.	The SQL DDL creates the <i>Booking Database Table</i> . With some errors. Four to six fields plus at least one of the additional fields are created.	The SQL DDL creates the <i>Booking Database Table</i> . With many errors. There may not be three to six fields or the two additional fields.	An attempt is made at producing the <i>Booking Database Table</i> . There may not be six fields.
Part 2: Server-side database development and security Task 2	8 – 10 Marks	6.5 – 7.5 Marks	5 – 6 Marks	4 – 4.5 Marks	1 – 3.5 Marks
Task 2	At least two sample data records are created. The foreign keys relate to existing primary keys.	At least two sample data records are created with some incomplete data. The foreign keys may not relate to existing primary keys.	Two sample data records are created with some incomplete data. The foreign keys may not relate to existing primary keys.	One sample data record is created. The foreign keys do not relate to existing primary keys. Many errors when the DDL file runs.	There was an attempt to create a sample data record. The DDL file may not run.

	No errors when the DDL file runs.	One error only when the DDL file runs.	Some errors when the DDL file runs.		
Part 2: Server-side database development and security Task 3	8 – 10 Marks	6.5 – 7.5 Marks	5 – 6 Marks	4 – 4.5 Marks	1 – 3.5 Marks
Task 3	Many aspects of data security are clearly explained, fully understood, and linked to the student's web application.	Data security is clearly explained with some links to the student's web application.	Data security is described well with some links to the student's web application.	Data security is described briefly with little link to the student's web application.	A brief definition of some data security terms is given.
Part 3 (Task 1): Server-side PHP development – Managing bookings Task 1(a)	4 - 5 Marks	3.5 Marks	2.5 - 3 Marks	2 Marks	1 – 1.5 Marks
Task 1(a) – Create Booking (R of CRUD)	A current bookings page is created and correctly lists all bookings. It links each booking to the view, edit and delete pages.	A current bookings page is created and lists all bookings. An error may be made. It links each booking to two out of three view, edit and delete pages	A current bookings page is created and lists all bookings. Some errors may be made. It links each booking to one of the view, edit and delete pages	A current bookings page is created and lists some bookings. Significant errors may be present.	A current bookings page is created with little or no functionality.
Part 3 (Task 1): Server-side PHP development – Managing bookings Task 1(b)	4 - 5 Marks	3.5 Marks	2.5 - 3 Marks	2 Marks	1 – 1.5 Marks
Task 1(b) – Create Booking (R of CRUD)	A single booking page is created and functions correctly. It links to the listing page.	A single booking page is created and functions correctly. It links to the listing	A single booking page is created and functions correctly.	A single booking page is created with little or no functionality.	An attempt is made to create a single booking page.

		page. One error may be made.	An error may be made, or the listing link may be incorrect.		
Part 3 (Tasks 2,3): Server-side PHP development – Managing bookings Task 2	4 - 5 Marks	3.5 Marks	2.5 - 3 Marks	2 Marks	1 – 1.5 Marks
Task 2	A delete booking page is created that allows an admin to preview, delete or cancel a booking. The page links to the listing page.	A delete booking page is created that allows most functions to work correctly. The page links to the listing page.	A delete booking page is created that allows some functions to work correctly. The page links to the listing page.	A delete booking page is created and links to the listing page.	A delete booking page is created with little or no functionality.
Part 3 (Tasks 2,3): Server-side PHP development – Managing bookings Task 3	4 - 5 Marks	3.5 Marks	2.5 - 3 Marks	2 Marks	1 – 1.5 Marks
Task 3	An edit booking page is created that allows the admin to edit, delete or cancel a booking. The page links to the listing page.	An edit booking page is created that allows most functions to work correctly. The page links to the listing page.	An edit booking page is created that allows two functions to work correctly booking. The page links to the listing page.	An edit booking page is created that allows any one function to work correctly. The page links to the listing page.	An edit booking page is created with little or no functionality.
Part 4: Server-side development – create bookings Task 1	8 – 10 Marks	6.5 – 7.5 Marks	5 – 6 Marks	4 – 4.5 Marks	1 – 3.5 Marks
Task 1	The create a booking page is created and works well with logical structure and well-designed code.	The create a booking page works as expected but contains an error.	The create a booking page mostly works with some errors.	The create a booking page partially works with many errors.	An attempt at creating a booking page is made.

Part 4 Server-side development – create bookings Task 2	16 – 20 Marks	13 – 15.5 Marks	10 – 12.5 Marks	8 – 9.5 Marks	1 – 7.5 Marks
Task 2	The search feature works faultlessly with well-designed code.	The search feature works as expected but contains an error.	The search feature mostly works with some errors.	The search feature partially works with many errors.	An attempt at implementing a search feature is made.
Part 5 Server-side development – Logging in	8 – 10 Marks	6.5 – 7.5 Marks	5 – 6 Marks	4 – 4.5 Marks	1 – 3.5 Marks
Task 1	A fully functioning and well-designed log in page is created.	A fully functional yet unorthodox design approach is used to create a log in page.	A well-designed approach to creating a log in page is made but not yet complete.	A good approach to creating a log in page is made but not yet complete.	An attempt at creating a log in page is made.