Web Development
Assignment 2

Semester 1, 2023

DUE ON: 11:59 PM, Friday 26 May 2023
Assignment Worth: 25% of total marks

NB: Assignments will be accepted up to five (5) days late, but a penalty of 5% per day (or part of a day) late will be imposed on either the team or the individual depending on the circumstances.

This is an individual assignment. Students are referred to the school's policy on plagiarism. A confirmed case will incur zero mark to all the involved students.

Naming Standards
All file names must be consistent with what is specified in the assignment instruction.

- For Task 2.1, you must use the same name "booking" for all types of files for this task. For example, booking.html, booking.php, booking.js.
- For Task 2.2, you must use the same name "admin" for all types of files for this task. For example, admin.html, admin.php, admin.js.
- For the MySQL table creation command, please write in mysqlcommand.txt.

1. Overview

The aim of this assignment is to develop a better understanding of building web applications using simple Ajax techniques, PHP, and MySQL. It is assumed that have the knowledge and programming skills to work with PHP and MySQL PHP on the server, as learned in Assignment One and the first part of this course.

For this assignment, you will need to create all the appropriate HTML, JavaScript, PHP files and database tables. You should save and test all your HTML and PHP files on *cmslamp14.aut.ac.nz* server in the directory "***/home/<your AUT network login>/public_html/assign2***".

2. Assignment Tasks

The assignment is to develop a simple web-based taxi booking system called **CabsOnline**. It allows passengers to book taxi services from any of their internet connected computers or mobile phones. The techniques you are going to use include the Ajax techniques (JavaScript/HTML, XMLHttpRequest, CSS, and DOM), MySQL and PHP. For client-server communication, you MUST use the XMLHttpRequest object.

Two components (booking and admin) of such an online service that must be completed for this assignment are specified in the following two sub-sections. Other components such as querying service for drivers, monitoring services for customers, payment processing, detailed processing for assigning taxi are not required in this assignment but you are free to extend as your hobby project.

2.1 The Booking Component

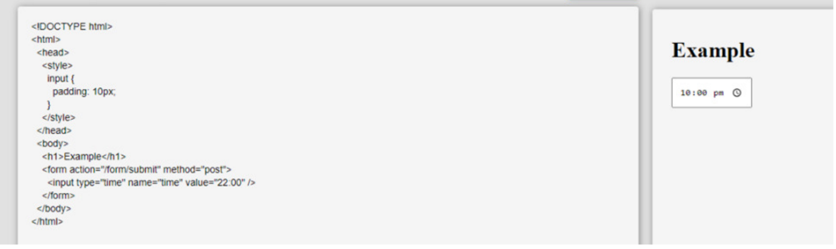
This component is used to allow a passenger to submit a taxi booking request in Auckland and its surrounding areas.

1) Design and create a MySQL table for storing information of all requests. For each request received, information including the generated booking number, customer's

inputs, the generated booking date and time, and the generated assignment status must be stored in this table.

2) On the client side, design an HTML page (booking.html) and a client-side program to take inputs for a booking request, and pass them to server for processing (using XHR object). You need to validate these inputs before sending them to the server:
    - **Customer Name** (Text input type. Must set its name attribute to “cname”: <input type=”text” name=”cname”>. NOT NULL)
    - **Phone Number** (Text input type. Must set its name attribute to “phone”: <input type=”text” name=”phone”>. NOT NULL). Phone number must be all numbers with length between 10-12
    - **Unit Number** (Text input type. Must set its name attribute to “unumber”: <input type=”text” name=”unumber”>. Optional)
    - **Street Number** (Text input type. Must set its name attribute to “snumber”: <input type=”text” name=”snumber”>. Not NULL)
    - **Street Name** (Text input type. Must set its name attribute to “stname”: <input type=”text” name=”stname”>. NOT NULL)
    - **Suburb** (Text input type. Must set its name attribute to “sbname”: <input type=”text” name=”sbname”>. Optional)
    - **Destination Suburb** (Text input type. Must set its name attribute to ”dsbname”: <input type=”text” name=”dsbname”>. Optional)[1]
    - **Pick-Up Date** (text or date input type. Must set its name attribute ”date”. NOT NULL, should initially contain the current date in dd/mm/yyyy format e.g. 01/05/2021. This can be edited by the user.
    - **Pick-up Time** (text or time input type. Must set its name attribute ”time”. NOT NULL, should initially contain the current time in 24h format HH:MM format e.g. 18:30. This can be edited by the user. **Please note that the pick-up date and time must not be earlier than current date and time on the client machine.**

      NB: The value attribute must be set to the 24-hour format. However, the time input type display format is either in 12 or 24-hour ***depending on OS locale*** *when viewing in Chrome. Therefore, the browser display could either be in 12 or 24-hour format. The following example is acceptable.*



```
<!DOCTYPE html>
<html>
 <head>
  <style>
   input {
    padding: 10px;
   }
  </style>
 </head>
 <body>
  <h1>Example</h1>
  <form action="/form/submit" method="post">
   <input type="time" name="time" value="22:00" />
  </form>
 </body>
</html>
```

3) On the server side, for each booking request, a server-side program is required to generate a unique and incremental booking reference number, booking date and time and a status with initial value “unassigned”, and add them together with customer’s inputs into the MySQL table. The booking reference number must follow the format of “BRN00001”, that is: the first three characters are upper case “BRN”, then the rest five character are digits.

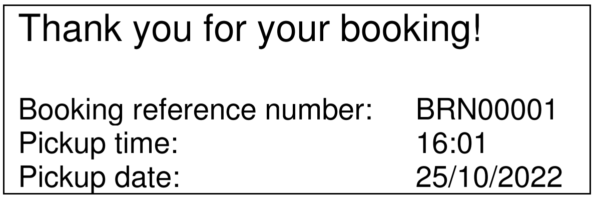
4) On the client side, the client-side program needs to display a returned confirmation message upon receiving the returned information from the server (as we are using

[1] *Note that it is not necessary to validate the genuineness of an address. If you do want to make the application more realistic, you can use* `google.maps.Geocoder()` *Web API. Please refer to Google Map API online document or https://codepen.io/pavenuto/pen/dXkgEz on how to do that.*

AJAX, the returned message should be inserted on the same booking.html page). The confirmation message must follow the following format requirements:

- Confirmation message must include keywords of “booking reference number”, “pickup time” and “pickup date”.
- The booking reference number must follow the correct format (as stated in Task 2.1.3)
- The pickup time must be displayed in the 24-hour format
- The pickup date must be displayed in the DD/MM/YYYY format

The confirmation message should be similar to the example below:



Thank you for your booking!

Booking reference number:	BRN00001
Pickup time:	16:01
Pickup date:	25/10/2022

You must make sure the message is inside a <div> element of the booking.html file. You may choose to use an appropriate tag for the message, but the id attribute must be set to “reference”. For example, you will need <p id=”reference”> if you choose to put the confirmation message in a <p> element.

2.2 The Admin Component

This component allows administrative people of CabsOnline to view those taxi booking requests that need to be assigned as soon as possible and to assign taxi for a particular booking request. Note that authentication is not required though it is necessary in the real application. The specific functions of this component include:

1) Design an HTML page (admin.html) that handles booking request searches. It should be realized by a single button to display pickup request information. This booking request search consists of two parts:
   - **A text field for booking request searching** (Text input type. Must set its name attribute to ”bsearch”: <input type=”text” name=”bsearch”>. Optional). User can either input a reference number or nothing:
     i. If user input is non-empty, it must be verified by checking the format of the reference number. When the input is not in the valid format, an error message is displayed accordingly.
     ii. After verifying the format of the reference number, the booking record associated with this reference number is returned by the server.
     iii. If user input is empty, then a list of bookings with a pickup time **within 2 hours from the current time** is returned by the server. Task 2.2.3 provides more details.
   - **A Button to submit the search request** (button input type. Must set its name attribute to ”sbutton”: <input type=”button” name=”sbutton”>). Once the button search booking is pressed, a client-side program issues a query request to the server (using XHR object);

2) On the server side, a server-side program responds to the search request by executing a query on the MySQL database to find the matching records.

- For input with a reference number, the record exactly matches the reference number should be returned or no record is returned if there is no match;
- For an empty input, the server-side program should execute a query on the database to find those “unassigned” booking requests with a pickup time **within 2 hours from the current time only**.

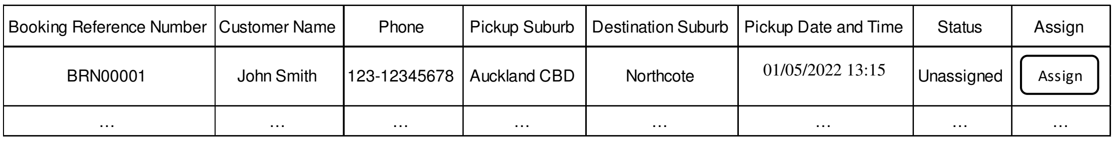
Information returned to the client should include the booking reference number, customer name, contact phone number, pickup suburb, destination suburb, and pickup date and time.

3) When getting a search request, the returned information must be displayed in a table inside a <div class=”content”> container in the admin.html file.

The following table header fields must be used:

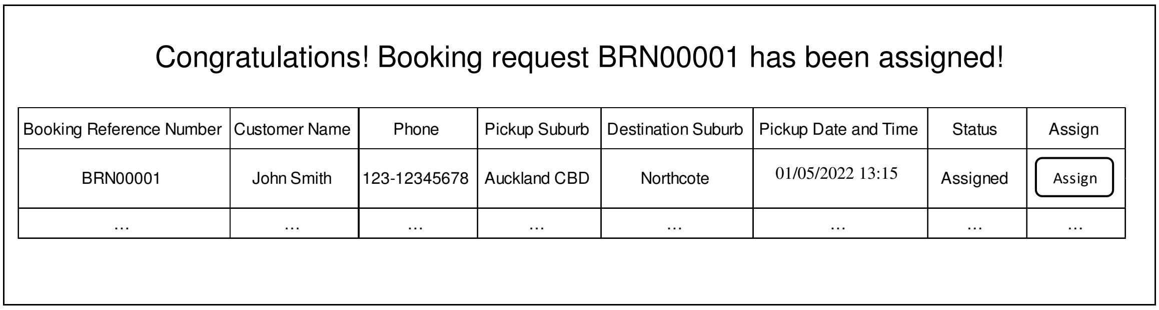
- Booking reference number
- Customer name
- Phone
- Pickup suburb
- Destination suburb
- Pickup date and time
- Status
- Assign

Each table row must contain one record. For each row, the “assignment” field must contain a button with name “Assign”. When the “assign” button is clicked, the status of this record (booking request) needs to be updated on both the database and the HTML page.
Below is a sample screenshot of the table:



Booking Reference Number	Customer Name	Phone	Pickup Suburb	Destination Suburb	Pickup Date and Time	Status	Assign
BRN00001	John Smith	123-12345678	Auckland CBD	Northcote	01/05/2022 13:15	Unassigned	Assign
…	…	…	…	…	…	…	…

4) In the same server-side program, it should also respond to taxi assignment requests. The booking reference number is sent to the server side when an ‘Assign’ button is clicked in admin.html. Upon receiving the client-side request, the server-side program should complete an update to the MySQL database to change the status from “unassigned” to “assigned” for the matching record. After that, a confirmation message should be returned and inserted into admin.html. The message must contain the booking reference number. You might optionally disable (gray-out) the corresponding ‘Assign’ button.
Below is a sample screenshot after the assignment:



Congratulations! Booking request BRN00001 has been assigned!

Booking Reference Number	Customer Name	Phone	Pickup Suburb	Destination Suburb	Pickup Date and Time	Status	Assign
BRN00001	John Smith	123-12345678	Auckland CBD	Northcote	01/05/2022 13:15	Assigned	Assign
…	…	…	…	…	…	…	…

Submission Requirements

You should ensure that all files used for the assignment sit in a directory called "*assign2*" (use this exact name, it is case sensitive and there is no space between *assign* and *2*) within your *cmslamp14* account. The directory should contain no sub-directories (i.e., all files are placed directly under the "*assign2*" directory).

Please follow the following steps to submit your assignment:

- Make sure your code works on the cmslamp14 server in the folder "assign2".
- Download the folder "assign2" (including all the files in it) to your local machine.
- Zip this folder and submit on Canvas, i.e. the file to submit should be in .zip format.

The files should include:

- two HTML files: booking.html and admin.html;
- any JavaScript files that you created
- any CSS files that you created
- any PHP files that you created
- a text file "mysqlcommand.txt" that includes the MySQL commands that you used to create the database
- a text file readme.txt that includes
    - a list of all the files in the system;
    - brief instructions on how to use the system.

For each submitted file, we require clear comments including student information, description of the file, and description of each function defined in this file.

The MySQL database that you use should be constructed in your *cmslamp14* account. After submission, you are not allowed to change any of the submitted files in the *assign2* directory on your *cmslamp14* account; time stamps will be checked.

Assignments that fail to follow "submission requirements" will <u>NOT</u> be assessed.

Notes:

- Please remove your database login details from the files before submitting to Canvas to avoid potential privacy breach.
- If your assignment cannot run on cmslamp14, your result will be 0 marks for this assignment.

Marking Scheme

Work will be assessed based on the quality and presentation. The assignment will be marked out of 100 and contribute 25% towards the final grade of this course.

Task	Assessment item	Received	Marks
2.1.1	Design and create a MySQL table		5
2.1.2	• Design an HTML page (booking.html) (2.5) • A client-side program to take inputs for a booking request and pass them to server for processing using XHR object. (2.5) • All required input items must be provided except the optional ones. (2.5) • The pick-up date and time must be no earlier than current date and time (2.5).		10
2.1.3	• Generate a unique and incremental booking reference number, booking date and time and assignment status (10) • Insert into the MySQL table (5)		15
2.1.4	• Confirmation message displays the following information: ○ Booking reference number (5) ○ Pickup time (2) ○ Pickup date (3) • Information displayed in the message follows the correct format (5)		15
2.2 Initial landing page	On the landing page (admin.html), the request searching functionality is realised. The following can be found on the page: • A text input field for searching. Name attribute is set to "bsearch". (5) • A search button where the name attribute is set to "sbutton". (5)		10
2.2 Booking Request Search & Taxi Assignment	When there is a non-empty input: • The input is verified by checking the format of the reference number. If the input is not in a valid format, an error message is displayed accordingly. (4) • A matched record is displayed on the page when the provided reference number already exists in the database. It should be presented in a table including the following information: (15 in total) ○ booking reference number (2) ○ customer name (2) ○ phone number (2) ○ pickup suburb (2) ○ destination suburb (2)		35

	o pickup date and time (3) o status (2) • An “assign” button should be present at the end of each table row. (2) • When clicking on the “assign” button, the record should change its status from “unassigned” to “assigned”.(5) • Once the taxi assignment is completed, a confirmation message is displayed. (3) When there is an empty input: • A list of bookings with pickup times within 2 hours from the current time is displayed on the page. (6)		
Overall quality	Comments are written in all files. For each file, the comments include descriptions of the file and all the functions used. Student information is included at the top of the files. Files should not include code that is commented out. The readme.txt file includes a list of all the files as well as brief instructions on how to use the system. Code quality. Evidence may include: - Consistent naming conventions for variables, methods and functions. - Indentation is used appropriately for good readability - Structured comments are presented throughout the files Usability of user interface - No redundant or misleading information can be found in the system. All information presented are consistent. - Buttons all function as intended. The actual functionalities match the text of the buttons.		10
	Total		100