SCHOOL OF GEOGRAPHY UNIVERSITY OF LEEDS	UNIVERSITY OF LEEDS

COURSEWORK COVERSHEET

Student ID number	2	0	1	9	0	2	5	3	8
Module code	GEOG5870M								
Module title	Web-based GIS (23078)								
Assignment title	Final Project								
Marker									
Declared word count	971								

Use of Generative Artificial Intelligence (Gen AI) in this assessment – mark one box as appropriate	
I have made no use of Gen AI	√
I have used Gen AI only for the specific purposes outlined in my acknowledgements	

By submitting the work to which this sheet is attached you confirm your compliance with the University's definition of Academic Integrity as: "a commitment to good study practices and shared values which ensures that my work is a true expression of my own understanding and ideas, giving credit to others where their work contributes to mine". Double-check that your referencing and use of quotations is consistent with this commitment.

You also confirm that your declared word count accurately reflects the number of words in your submission, excluding the overall title, Gen AI use acknowledgement, bibliography/reference list, text/numbers in tables and figures (although table and figure captions are included in the word count).

Development of Leeds Leisure Centres Website Based on Leaflet

1. Introduction

This project aims to develop an interactive website that helps residents and visitors quickly find information about leisure centers in Leeds. The website provides detailed information, including facilities, opening hours, contact details, and location. With an intuitive map navigation system and searchable directory, it aims to offer users a convenient way to choose and find suitable facilities. The project uses Leaflet as the map display library, and combines HTML, CSS, and JavaScript to implement dynamic interactive features, greatly enhancing user experience.

2. Requirements Analysis

A brief requirements evaluation revealed the following user groups:

- **Parents with children:** Looking for safe, nearby swimming or play facilities.
- **Office workers:** Interested in viewing real-time fitness class schedules.
- **New residents or visitors:** Unfamiliar with local leisure options.

Functional Requirements:

- Provide an interactive map showing the locations of all leisure centers in Leeds.
- Filter leisure centers based on service types (e.g., gym, swimming pool, classes), and provide a full directory with detailed information.
- Display each leisure center's opening hours, allowing users to view them without downloading PDF files.

These requirements provide a foundation for the frontend layout and database design.

Data Requirements:
The website’s data comes from the Leeds City Council, including information on the locations, facilities, and opening hours of leisure centers. This data is imported into a PostgreSQL database via pgAdmin. The front-end makes AJAX requests to PHP scripts, which query the database and return the data in JSON format.

3. Technologies and Tools Selection

Frontend Technology Stack:

(1) Leaflet: An open-source JavaScript library for building interactive maps. It is suitable for displaying the geographic locations of leisure centers.

(2) HTML/CSS: Used for constructing the website's basic structure and style, ensuring clarity and responsiveness.

(3) JavaScript/jQuery: Used for implementing dynamic features, such as showing and hiding maps, loading data, and user interaction.

Backend Technology:
The backend uses PHP for data retrieval, filtering, and structuring it into JSON format. jQuery is used to simplify DOM manipulation and AJAX requests. Spatial data storage and querying are handled by PostgreSQL.

4. Implementation Process

4.1 Page Structure

The website consists of four main pages:

(1) Homepage (index.html): Provides an overview of the website, navigation, welcome banners, and feature highlights. Users can navigate between pages via a navigation bar.

(2) Map Page (map.html): Embeds a Leaflet map displaying the spatial distribution of all leisure centers. Clicking on a marker shows detailed information.

(3) Listing Page (centres.html): Displays basic information about all leisure centers. Users can filter the list dynamically by entering keywords or selecting service categories (e.g., swimming, fitness). The page sends requests to the backend via JavaScript and AJAX, updating the list in real-time based on user input.

(4) Opening Hours Page (Opening Hours.html): Lists the opening hours of each center. Users can filter and view the detailed opening hours to plan their visit.

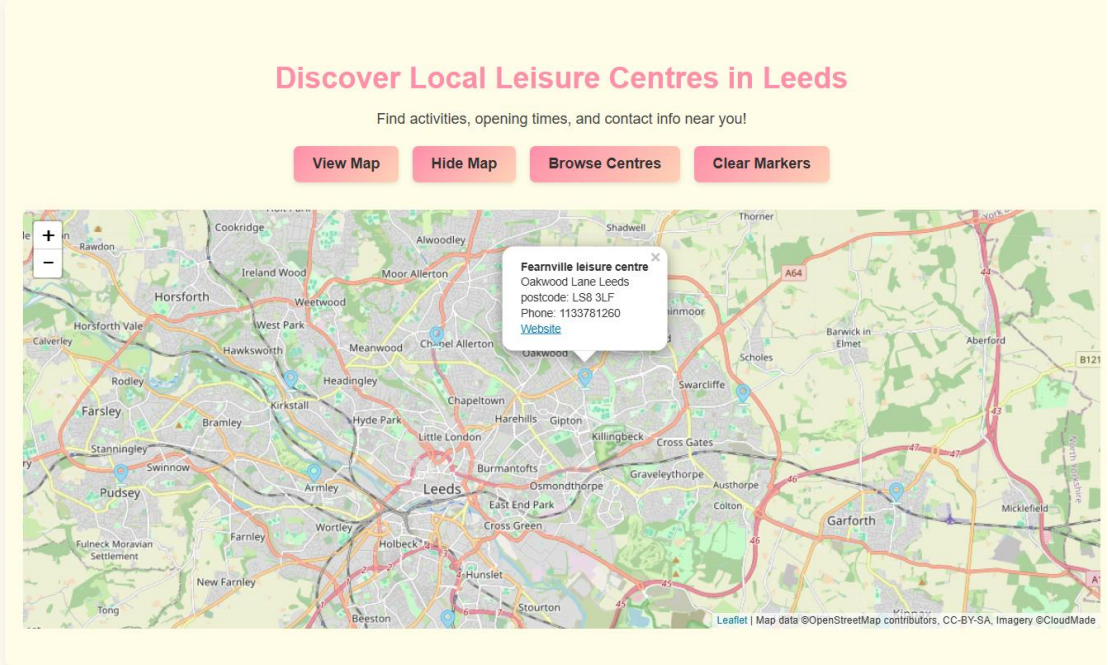
4.2Interactive Button Features

The website’s interactive features are mainly implemented using JavaScript (script.js) in combination with jQuery, improving responsiveness and user experience.

(1) Show/Hide Map Feature

This feature allows users to dynamically show or hide the homepage map area, enhancing visual clarity and usability.

- Clicking the #showMapButton triggers the showMap() function, which sets the height of the #map-container to 450px and initializes the Leaflet map if it hasn't been initialized yet.
- Clicking #hideMapButton calls the hideMap() function to collapse the map container by setting its height to 0px.



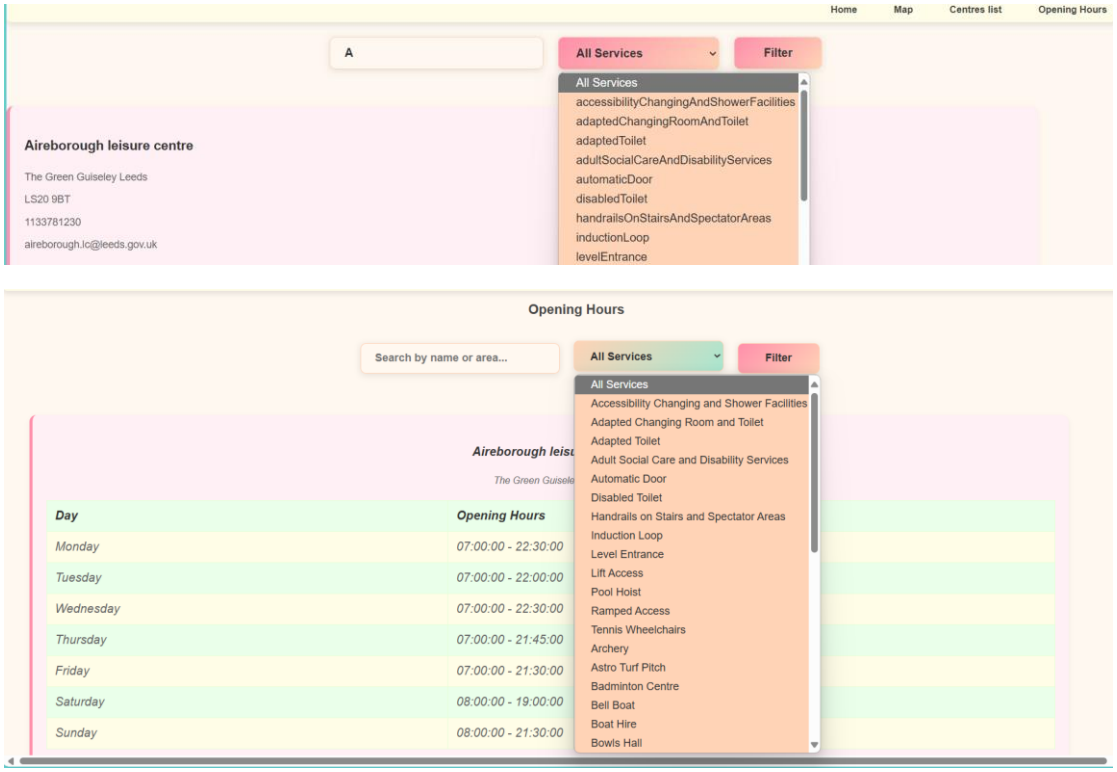
(2) Map Browsing and Interaction

Clicking the “Browse Centres” button opens the map view where all leisure centres are shown as markers. Clicking a marker opens a popup with centre details such as name, services (e.g., swimming, gym, classes), and contact info. The “Clear Data” button removes all markers to declutter the map.

Implementation Steps

- **Data Fetching:**
  The fetchData() function sends an AJAX request to fetchData.php upon clicking the “Browse Centres” button, retrieving centre data in JSON format. The data (e.g., name, location, address, phone, website, opening hours) is stored in the global array leisureCentresData for later use.
- **Plotting on Map:**
  The plotLeisureCentres() function displays the centres on the map by creating markers and binding popups with detailed centre information.
- **Button Binding & Event Handling:**
  Upon page load, window.onload binds actions to the “Browse Centres” and “Clear Data” buttons:
  the former loads data and expands the map, while the latter clears all markers via clearData().

(3) Leisure Centre Filtering Feature



Users can filter leisure centres in Leeds using a top search bar and service category dropdown.

Implementation Steps

- **Input & Trigger:**
  Users type keywords in #searchBox, select a service from #serviceFilter, and click the filter button to trigger the filterCentres() function.
- **Sending Request:**
  JavaScript sends a POST request to fetchData.php via $.ajax(), passing the keyword (search) and selected service (service).
- **Backend Processing:**
  The PHP script escapes input using pg_escape_string() to prevent SQL injection. It constructs SQL queries based on the input:
  - If a keyword is provided, it uses ILIKE for fuzzy matching against names or addresses.
  - If a service is selected, it filters for centres offering that service.
- **Data Return & Display:**
  The server returns matching centre info (name, address, contact, hours), and JavaScript dynamically updates the webpage with the filtered results.

5. Issues and Improvements

Despite its success, the system has limitations:

- **Lack of mobile responsiveness**: Elements may overlap or shrink on small screens. Future updates will include media queries and improved touch controls.
- **Static content**: No real-time updates on class availability or temporary closures. Integrating an API connected to internal centre systems could solve this.
- **No personalization**: All users see identical results. Future versions may add a login system to save preferences and offer tailored suggestions.

6. Conclusion

This website effectively visualizes leisure centres in Leeds, helping users locate recreational services easily. By integrating Leaflet, PHP, and AJAX, the platform delivers responsive data interaction and improves the overall experience. Future enhancements like real-time updates and personalization can further increase its value.

7.Reference

Mozilla. No date. *Window: load event – Web APIs*. [Online]. [Accessed 7 May 2025]. Available from: https://developer.mozilla.org/en-US/docs/Web/API/Window/load_event

W3C. 2014. *HTML5 Specification*. [Online]. [Accessed 7 May 2025]. Available from: https://www.w3.org/TR/html5/

W3Schools. No date. *AJAX Introduction*. [Online]. [Accessed 7 May 2025]. Available from: https://www.w3schools.com/xml/ajax_intro.asp

Yenduri, R. and Al-khassaweneh, M. 2022. PHP: Vulnerabilities and Solutions. *Proceedings of the 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. [Online]. pp. 391–396. [Accessed 7 May 2025]. Available from: https://doi.org/10.1109/MIUCC55081.2022.9781790