



## Assessment Cover Sheet

Assessment	Midterm examination		
Assessment	Controlled	Individual	Not must-pass
Due Date	18 May	Course Code	IT8415
Course Title	Database Programming 2		
Internal Moderator's	Mr. Behrouz		
External Examiner's			

**Instructions:**

1. This cover sheet must be completed and attached to your assessment before submission.
2. The time allocated for this assessment is two months.
3. This assessment carries 100 marks and assessing CILO 1, CILO 2.
4. This assessment has 8 pages.
5. The **use of generative AI tools is strictly prohibited**.
6. References consulted (if any) must be properly acknowledged and cited.

Learner ID(s)		Date Submitted	
Learner Name(s)			
Programme			
Lecturer's Name			

By submitting this assessment for marking, I affirm that this assessment is my own work.

Learner

Assessor's Name			
Marking Date		Maks Obtained	

Comments:

# Borrow My Charger - Project Requirements Document

## 1. Overview

The "Borrow My Charger" system is a data-driven web application designed to facilitate electric vehicle (EV) charging point rentals. Homeowners with a charging point can offer it for paid usage by EV owners. The system will be developed using PHP, MySQL, HTML/CSS, and JavaScript. It will feature three user roles: Homeowner, Admin, and User, each with distinct permissions.

## 2. System Roles and Permissions

### 2.1 Homeowner:

- Can register, log in, and manage their charge point.
- Can edit their own charge point details (location, price, availability).
- Cannot edit or delete other homeowners' charge points.
- Can view booking requests.

### 2.2 Admin:

- Can manage all charge points (add, edit, delete).
- Can manage user accounts (approve, suspend, delete).
- Can generate system reports using SQL queries to retrieve data directly.

### 2.3 User (Rental User):

- Can register, log in, and search for charge points.
- Can book charge points by sending requests to homeowners.
- Can view and review their booking history.

## 3. Functional Requirements

### 3.1 User Registration & Authentication

- Secure registration and login system which prevent robot sign in
- Password encryption.
- Different roles with specific access permissions.

### 3.2 Charge Point Management

- Homeowners can:

- Add/edit their own charge point (only one per homeowner).
  - Set charge cost (price per kWh).
  - Set availability.
  - Upload a picture of the charge point, which will be stored in the system and linked to the respective charge point record.
- Admin can:
    - View, edit, and delete any charge point using SQL data retrieval.
    - Approve homeowner registrations.

### 3.3 Search and Filtering

- Users can search for charge points based on:
  - Location (address, postcode, lat/lng).
  - Price range.
  - Availability.
- **Live search implementation using AJAX for real-time filtering of charge points based on user input.**

### 3.4 Booking System

- Users can:
  - Book a charge point for a specific date and time.
  - Contact the homeowner via a form submission.
  - Users can view their booking requests with statuses such as Pending, Approved, or Declined.
  - Real time update: an AJAX-based polling mechanism could periodically check for status updates without requiring a page refresh.
- Homeowners can:
  - Approve or decline booking requests.

### 3.5 Interactive Map Integration

- Display available charge points using Google Maps API.

- Users' current location should be centered on the map.
- Filters integrated with map search.
- **Charge point location updates via AJAX without page reload.**

### 3.6 Pagination

- SQL-based pagination to retrieve and display charge points efficiently.
- Limit the number of records fetched per query to optimize performance.

### 3.7 Security Features

- Prevent SQL injection and cross-site scripting.
- Role-based access control for sensitive pages.

## 4. Non-Functional Requirements

### 4.1 Performance

- The system should support 1000s of users without performance degradation.
- Optimized SQL queries to handle large datasets.

### 4.2 Responsive Design

- Fully responsive for mobile, tablet, and desktop users.
- Intuitive UI with Bootstrap framework.

### 4.3 Usability

- Intuitive and easy-to-use interface for all user roles.

### 4.4 Maintainability

- Well-documented code with comments.

## 5. Technologies to be Used

- **Frontend:** HTML, CSS, JavaScript, AJAX
- **Backend:** PHP (MVC Architecture), MySQL
- **Hosting:** University Server
- The core PHP architecture **needs** to follow an **Object Oriented, Model-View-Controller** (MVC) Design Pattern approach. This is covered in the lectures and

workshops and you practice it by following the workshop material closely. **You are not allowed to use jQuery or a high level PHP framework.** You need to use and adapt the MVC templates that you use in the workshops and this is **very important!**

## 6. Submission Requirements

The system must be deployed and accessible on the designated server.

A ZIP file submission must include the following:

- All project files, ensuring completeness.
- A documentation file detailing the achievement level of each functionality and specifying the corresponding files within the MVC structure (Model, View, Controller).
- Clear screenshots of the database tables.

**The submission must include a direct link to the live system for evaluation.**

**The functionality of the system will be evaluated first, followed by a detailed assessment of the code to determine the final marks**

**For testing purpose be sure you have**

Home (charge point) owner	Rental user
<ul style="list-style-type: none"><li>• User ID (auto generated in the DB)</li><li>• Username: lee@lee.com</li><li>• Name: Lee Griffiths</li><li>• Password: 123456</li></ul> <p>With a charge point record:</p> <ul style="list-style-type: none"><li>• 5 The Crescent, Salford, M5 4WT</li><li>• Lat: 53.483710, Lng: -2.270110</li><li>• Price (for charging) £0.25 / kWh</li></ul>	<ul style="list-style-type: none"><li>• User ID (auto generated in the DB)</li><li>• Username: user@user.com</li><li>• Name: User Lee Griffiths</li><li>• Password: 123456</li></ul>

<ul style="list-style-type: none"><li>• <b>Admin user: User ID (auto generated in the DB)</b></li><li>• <b>Username: admin@admin.com</b></li><li>• <b>Name: User Lee Griffiths</b></li><li>• <b>Password: 123456</b></li></ul>	

## 7. Conclusion

This project aims to provide a seamless EV charger rental experience while ensuring data security, scalability, and performance. Each user role will have specific functionalities that contribute to the smooth operation of the system.

## Rubric

Functionality	Percentage	Excellent (5)	Good (4)	Satisfactory (3)	Needs Improvement (2)	Poor (1)
User Registration & Authentication	13%	Fully functional with secure validation and encryption	Functional but minor issues with validation	Basic functionality with limited validation	Issues in login/registration process	Registration/login not working
Role-based Access Control	10%	Fully implemented with restricted access per role	Mostly implemented with minor security issues	Some access control but with gaps	Limited access control implemented	No access control
Charge Point Management	12%	Fully functional with image upload and data integrity	Functional but minor issues with input handling	Basic functionality with partial features	Issues with data handling	Charge point addition not working
Image Upload	3%	Successfully stores and links image	Stores image but minor display issues	Basic implementation	Issues in image processing	No image upload functionality
Search and Filtering	15%	Fully functional with live AJAX filtering	Functional but minor performance issues	Basic filtering implemented	Search with minimal functionality	Search feature not working
Booking System	10%	Fully functional with notifications and status updates	Functional but minor delays in updates	Basic booking submission	Issues with booking process	Booking system not working

Status Updates	5%	Real-time updates via AJAX	Updates work but minor display issues	Status changes with manual refresh	Status updates delayed or incorrect	No status tracking
Interactive Map Integration	12%	Fully interactive with geolocation	Functional but minor accuracy issues	Basic map display	Issues in displaying map	Map feature not implemented
Charge Point Updates	5%	Real-time updates via AJAX	Updates work but require manual refresh	Basic updating	Issues in map updating	No map updates
Security Features	5%	Fully secure against all vulnerabilities	Secure but with minor gaps	Basic validation implemented	Limited security measures	No security measures implemented
Documentation	5%	Comprehensive and clear documentation	Good documentation with minor missing details	Basic documentation	Minimal documentation	No documentation provided
Test Accounts	5%	All required test users created and functional	Test users created with minor issues	Basic test accounts	Test accounts incomplete	No test accounts provided