Description

This continuous assessment (project) is a group project, and in each group, there should be four members.

Project Specification

The purpose of this assignment is to create an **end-to-end EV Charging Station Booking System using client-server architecture**. The system should have a web application for back-office work and charging station operators, and a **mobile application for EV owners.**
Here's an overview of the required functionality:

a) Web Application:

a. User Management:

I. Create web application users with two distinct roles. Backoffice and Station Operator.
II. Only Backoffice users should have access to system administration functions and Station Operator can access the mobile for EV operations

b. EV Owner Management

I. Create, update, and delete EV owner profiles using NIC as the primary key.
II. Enable activation and deactivation of EV owner accounts.

c. Charging Station Management

I. Manage charging station details, including:
Creating new charging stations with location, type (AC/DC), and available slots.
II. Updating station details and schedules (availability of slots).
III. Deactivating stations (cannot deactivate if active bookings exist).

d. Booking Management

Manage charging bookings, including:
I. Creating new reservations (reservation date/time within 7 days from the booking date).
II. updating reservations (at least 12 hours before the reservation).
III. Canceling reservations (at least 12 hours before the reservation).

e. User Interface

I. Develop user interfaces using **Bootstrap 5** or **Tailwind CSS** for a modern, responsive appearance and **React.js** also allowed.

Mobile Application: Mobile apps must be pure Android with SQLite Db(no frameworks)

- Mobile applications are intended to provide functionalities for both EV owners and station operators.
  - **a. User Management:**
    - i. Allow EV Owners to create their own accounts (using NIC as the PK).
    - ii. Allow EV Owners to update and deactivate their accounts.
    - iii. Deactivated accounts can only be reactivated by a back-office officer.
    - iv. It should maintain local db for User management.
  - **b. Reservation Management:**
    - i. Allow EV owners to create, modify, and cancel reservations, providing a summary before confirmation.
    - ii. Once the booking is approved, it should be able to generate the QR code.
  - **c. View Bookings:**
    - i. Enable EV owners to view upcoming bookings and past charging history.
  - **d. Dashboard**
    - i. This is a home screen.
    - ii. It should contain how many pending reservations
    - iii. Count of Approved future Reservation
    - iv. Show nearby charging stations on a map (Google Maps API).
  - **e. EV Operator**
    - i. EV Operator (Station Operator) can login to mobile by its login.
    - ii. They could read the QR code and confirm the data retrieved from the server
    - iii. Finalize the business once EV operation is done.

b) Web Service

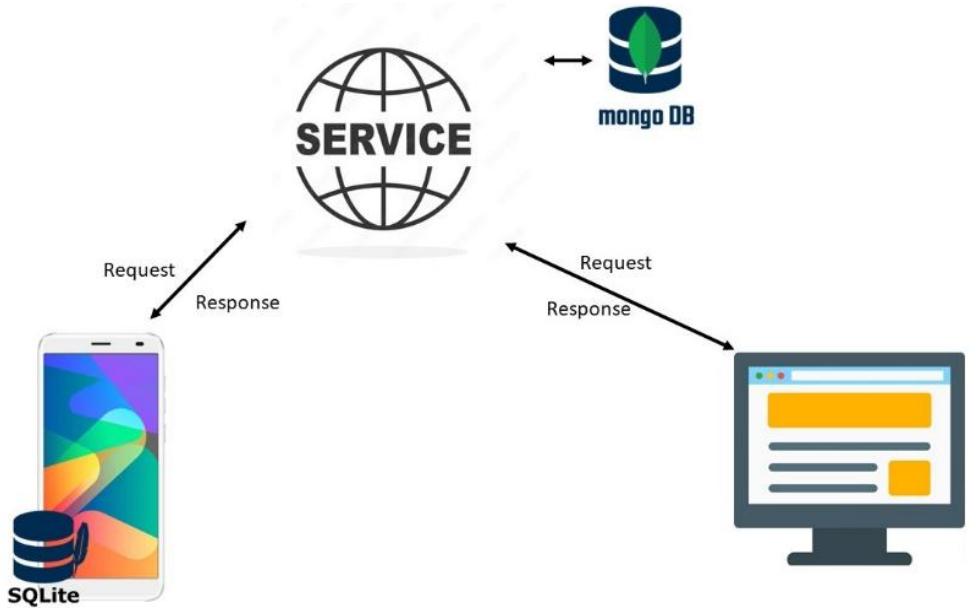
A central web service will handle all client requests.

- All business logic must be implemented here (FAT Service).
- The service should expose using **Web API based on C# language**.
- Server-side data storage must use a **NoSQL database**.
- The system must run on a centralized service running on Windows IIS Server.

The **web application** and **mobile application** should be limited to **user interface functionality** with Database and interact exclusively with the web service.

This system incorporates **web and mobile applications that connect to a centralized web service running on IIS**. Both client applications interact with the web service to access processed data and business logic. The server-side data storage utilizes a NoSQL database.

Therefore, it should develop an Android mobile application (Pure android and No frameworks) and web applications which have only interfaces to interact with Web service to facilitate data and Business logic and the NoSQL database. Here is the high-level diagram.



Project Scenario:

In this project scenario, the Backoffice team is responsible for managing the registration of EV charging stations and maintaining their schedules. Station operators, who can access both the web application and the mobile application, are responsible for updating slot availability and monitoring bookings. EV owners, on the other hand, can conveniently use the mobile application to reserve charging slots, modify existing reservations, and review their charging history. Once a booking is confirmed, the EV *owner receives a confirmation and can access the details directly on their dashboard*. If circumstances change, cancellations can be easily made either through the mobile application or with the assistance of a station operator.

What do you have to hand in?

1. Project Code AND a screen shot: Provide a single zipped file containing all directories/files in your project and a detailed report. You should also include a screen shot of the main opening menu/screen of your app. Include IT number in the zipped file name e.g. IT15895623.zip

Note: Code that does not include the following will not be marked

- **comment header block on each .cs file.**
- **Inline comments at the beginning of each method.**
- **Unique screenshots of the application.**

2. You should write a detail report that describes your app with following Criteria.

- Screen shots of all UI's
- Application high level diagram, Use case Diagram and DFD diagram.
- Database design
- Source code **(not screen shots**). Copy and paste your codes.
- All the references
- Individual contribution
- Challenges

Marks Allocation

This group project assignment is worth **20%** of your overall final mark in this module. Mark distribution is as follow.
Note: No exemptions are approved. It should clearly mention the **c**ontribution of each student. (Should this goes to the report or separate document?)

Project report	5%
source code, and demo/viva (participation required)	15%

Due Date

Please submit the project source code and report on or before **2nd of October 2025 11.59 pm**.

What is Viva for the remaining assessment of the app?

Viva session will test your knowledge in your own app. It will be done as a supervised test (After submission).
If you are absent for the viva session, your assignment will not be marked.

Instructions

- All students are expected to submit assignments within the given deadline.
- late submissions are not accepted.
- Viva is treated as formal exam. No additional dates will be provided. You will get the date of viva at least a week before.
- Any code snippets that are not directly written by you (e.g., used from a tutorial) must be referenced as such within your code. You must directly comment the code to explain its source**. Failure to reference code that is not yours will be treated as plagiarism.**
- The app must be your own work. **Any plagiarism found will get zero marks**, and the plagiarism escalated as per DIT assessment regulations.