

# Vehicle Rental Management System

## Scenario

The goal of this project is to build a vehicle rental management system using ASP.NET MVC. The system will allow car rental companies to manage their fleet of vehicles, reservations, customers, and billing.

## Features

1. Login and Registration: Users will be able to register and log in to the system.
2. Dashboard: A dashboard will be created for users to view important information about their fleet, reservations, and customers.
3. Manage Vehicles: Users will be able to add, edit, and delete vehicles from their fleet. The system will also track vehicle availability.
4. Manage Reservations: Users will be able to create, modify, and cancel reservations for customers. The system will track reservation dates, vehicle details, and customer information.
5. Customer Management: Users will be able to manage their customer database, including adding new customers, editing existing ones, and deleting old ones.
6. Billing: The system will automatically calculate the total bill for each reservation, including any taxes or additional charges.
7. Reports: Users will be able to generate reports based on various criteria, such as reservation dates, vehicle types, or customer demographics.

## Group Tasks

### 1. Project Design:

Set up group meeting and discuss the tasks of the project. Assign the tasks to each team members. Based on the features above, design the database schema for the system and design the UI for the page.

How to submit (only **one submission** for a team)

Follow the Project Design Template structure, finish and submit the Project Design Document before the due date.

Total 100 points:

30 points for tasks, 30 points for UI design, 30 points for Database design, 10 points for grammar.

## **2. Develop the website:**

Develop the ASP.NET MVC application, including the controllers, models, and views.

Connect Database and implement CRUD.

Build the UI using HTML, CSS, and JavaScript.

Implement user authentication and authorization.

Test the system thoroughly to ensure it is bug-free and user-friendly.

Deploy the application to Azure and make it accessible to users.

**Bonus:**

Integrate third-party APIs, such as payment gateways or map services.

## **3. Push your code to GitHub**

Create a repository for this project in one teammate's account, and push code to this repository.

**Bonus:**

Before building the project, create one branch for each team member. Build every feature or module in a different branch. After finish, merge to master branch.

How to submit: (only **one submission** for a team)

1. Record the presentation and submit the recording file in D2L.

The presentation should include all your team work: about the project design, run your system on Azure, explain how to use your system.

2. Paste the link on Azure and GitHub repository link in comment.

There should be **two links** in comment.

Total 100 points:

70 points (10 points for each feature) + 30 points (10 points for UI, 20 points for code)