Motorkontor case

# Introduction

Create a Server-Side application using Blazor Server .net5.

The application should facilitate work done by office personnel in a ‘Motorkontor’ (Vehicle Registration Office).

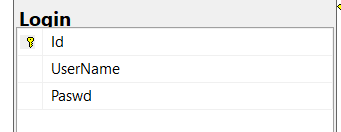
The application should have a Navigation Bar and a Main Content Area (the place to show pages).

# Database

Using below listings create a database. Name it ’Motorkontor’.

Create *relations* between tables (1:1 og 1:M). You’ll need *foreign keys*.

The Login table will serve as your account. You will hook that up to the Customer table.



|  |
| --- |
|  |

CRUD (Create, Read, Update and Delete):

On tables: Login, Registration, Customer, Address, vehicle

Other tables are just prefilled: (your database script must fill data into Zipcode, Fuel (eg, Gasoline, Diesel, EV, Hydrogene), Category (eg. Van, Car, Lorry, Truck…)).

# Queries

Notice:

* All selects must have field names. It is not allowed to use ‘\*’ for all fields, list them.
* All input values must be using Parameters to prevent Sql Injections.
* You must show knowledge of using Stored Procedures. At least one Insert and one Update must be executed by using a Stored Procedure.
* Show use of at least one View: Get Customers along with relevant data for a registration.

Necessary functionality that the personnel must be able to perform:

* Select all data/fields from all tables
* Select all data/fields from all tables based on an Id (leave out tables Zip, Category and Fuel – unless you use them in certain components).
* Insert rows on: Customer, Address, Registration and Vehicle.
* Delete rows on: Customer, Address, Registration and Vehicle.
* Update rows on: Customer, Address, Registration and Vehicle
* Get all Vehicles based on a certain fuel. (Bonus: select all Registrations with a certain fuel).
* Get all Customers from a certain city. (Bonus: get all Registrations with customers from a certain city

All functionality/queries must be used somewhere in the program.

We assume a functional GUI.

# Indexes

At least one additional (non-clustered) index on at least one table.

# Login

The user must be able to login.

Your login should hold some kind of a session id which is used to authorize you to use the pages you are accessing.