

A second hand car dealer called 'Moylish Motors' has recently started to trade in Limerick and has decided to base itself in the Moylish area of the city. They have a number of vehicles on their books (the details of which are stored in an Excel spreadsheet) which they have extensively photographed. They are currently operating without an online presence, which they feel, is hampering their business. They have approached you to build a web application with both front and back office functionality.

Front Office Functionality (20%)

1. Every customer will be able to search the database for a vehicle based on its price, make, model and year (feel free to expand the search criteria if you wish). The search results should be presented in tabular form. This table must include a thumbnail image for the vehicle. As an extra option, the customer should also be able to refine their search results - consider using Data Tables (<https://datatables.net/>) to help you refine your search results.
2. You must enable the thumbnail image so that it appears as a link that when clicked on, will provide extra information about the vehicle in question (this is in effect a drill-down: extra information such as the odometer reading, the engine size, the fuel type, the transmission type, the body type etc. As well as that, details of the salesperson responsible for selling the vehicle should be displayed. The larger images for the vehicle should also be displayed for the vehicle on the drill down page.
3. On the drill-down page for each vehicle, a customer should be able to add a vehicle to a list of their "favourites". This list can be viewed **at any time** by the customer and you must also provide the ability for the customer to remove any vehicle from their list of favourites. The list of favourites must also be available to the customer after their browser session has been terminated. Obviously, each customers list of favourites will be independent of each other. Assume that no customer will access the site from more than one computer.
4. You are required to add two other unique features to the front office.

Note: certain details like the buy price of the vehicles and those that relate to the previous owners are not intended for public display.

Back Office Functionality (35%)
--

The back office is a part of the application that the customer does not see nor interact with. Because of this, all back office functionality must be protected with an authentication and authorisation mechanism that you must develop (how many user roles will be in your application?). You have a free hand to develop whatever functionality you choose for the back office but it should be as comprehensive as possible.

For the purposes of this exercise, you can assume that the 'List Price' is 8% greater than the 'Buy Price' (except where models are marked as reduced to clear). The 'List Price' is only a guide however and is not the final sale price as discounts can be brokered. When a customer is purchasing a vehicle they will more than likely trade in a vehicle of their own and the value of this car (the trade in) will affect the final sale price.

3rd Party API (10%)

You must integrate a 3rd party API's such as (but not limited to), [Twitter](#), [Google](#), [Facebook](#), [FourSquare](#), [Yelp](#), [Ebay](#), [Motor Check](#) (€), etc., to your application.

A list of more general Motoring API's can be found [here](#).

Weekly Demonstration and Progress Report of Your Work in Class (15%)

You are required to demonstrate your code weekly in class to me between now and the final submission.

Failure to demonstrate your work in a given week will see you forfeit your marks for that demonstration. The demonstrations will take place during your scheduled practical class for the weeks specified.

Demo	Week Beginning	Marks
	26/10/2020 (Reading Week)	
#1	2/11/2020	3%
#2	9/11/2020	4%
#3	16/11/2020	4%
#4	23/11/2020	4%
	30/11/2020 (final submission there will be no demo on this week).	N/A

Clear progress will be expected from week to week. You must maintain an MD file for this project, which charts your development efforts on a weekly basis (a progress update). The marks allocated for the weekly demo will incorporate a mark for this file.

Some markdown tutorials include....

<https://guides.github.com/features/mastering-markdown/>

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

<https://www.markdowntutorial.com/>

Security (12%)

Sensitive information must not be transmitted in plain text and instead must be sent over HTTPS – using HTTPS for all connections is recommended.

All code must be resistant to SQLi and XSS and all data must be validated (and sanitised) before entering the database. It is recommended that you do some research into best practices for identification and prevention of both SQLi and XSS.

All passwords stored in the database must be encrypted.

Database Design (8%)

You are not being issued with a database for this assignment even though you must use one. You can assume that Moylish Motors have been effectively running their business through a spreadsheet. You will have to convert this spreadsheet to a MySQL database to build an effective web application. This will involve you:

- Dividing the information into a table(s)
- Turning information items into columns with specific types.
- Identifying primary keys for each of your tables.
- Identifying relationships amongst your tables.
- Apply the normalization rules.

The process of exporting of an Excel spreadsheet to a MySQL database (can be) an [automated one](#).

A poor database design will hinder your development and is something that you should give a lot of consideration to.

Note:

Once the submission date has expired, I will be testing your project on my computer. It is important that you bear this in mind when developing your work (do not use fully qualified paths etc. and make sure to test your application across multiple browsers).

Your solution must use **JPA** (complimented by the Repository Pattern) and **connection pooling**.

You **must use Tomcat** as your server/container.

All authorisation/authentication **must be implemented using Apache Shiro**. You may also decide to use Shiro for cryptography, and session management but are not required to do so.

All **erroneous conditions** must be handled **gracefully**.

Your solution **must** adhere to the **MVC** principles (no JavaScript should appear directly in your JSP's. Instead, provide links to the scripts themselves).

The design of your application must be **responsive** (this can easily be tested using Google Chrome) as well as being user friendly and intuitive.

The deadline for this assignment is 23:59 on Wednesday, December 2nd. I will be using GitHub classroom to manage this assignment and each of you will have your own private repository for this assignment. You are required to commit at least once a week to this repository. Failure to do so (commit once a week), will see your final mark reduced by 3% for every weekly commit you miss. *In reality, you will more than likely commit your work more than once a week.*

****I also require you to upload a copy of your final project to Moodle before the deadline expires****