A company called Irish Home Listings (IHL) has recently formed in Limerick with the intention of advertising property online and currently advertise their properties in the national press. To facilitate these advertisements, IHL have developed a complete database of their properties and their selling agents. You must develop a complete enterprise application for IHL that will offer the following functionality. I have broken the functionality down into two main categories.

Front Office Functionality (35%)

1. Every user will be able to search the database for a property based on its price, location, number of bedrooms, number of bathrooms (feel free to expand the search criteria if you wish) - consider using Data Tables (https://datatables.net/) to help you refine your search results.
2. When the user views a property online its location must be displayed on a (Google) map.
3. A user should be able to add a property to a list of their "favourites". This list can be viewed **at any time** by the user and you must also provide the ability for the user to remove any property from their list of favourites. The list of favourites must also be available to the users 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. Create a process to allow users to register with the application. There is currently no "users" table within the database. For now you can store passwords in plaintext.
5. Registered users should receive an email when similar local properties come to the market to those that they have viewed in the past.
6. The application should make recommendations to users based on their search history.
7. Registered users should be able to make notes on individual properties. These notes are private and only the user will see them.
8. You are required to add two other **unique** features to the front-office.

Back-office Functionality (50%)

The back-office is a part of the application that a general user does not see nor interact with. Because of this, all back-office functionality would normally be protected with an authentication and authorisation mechanism. We will look at implementing security/AuthX/AuthN mechanisms in Assignment Three.

1. View, edit, and insert a property to the database. An insertion/update must also include the ability to upload a new/updated image(s) for the property in question. A property can have multiple images associated with it and your update/insert features must cater for this requirement.

2. Agents should be able to archive a property which will effectively remove it from the main listings but will not delete it from the database. Perhaps the property has been sold or removed from the market hence the need to archive it.
3. Each property that appears on the website has a vendor (who has trusted IHL to sell their home). You can assume that each property has one vendor and that, the agent who is responsible for selling the property will manage their details. It is possible that one vendor may be selling more than one property. There is currently no "vendor" table within the database.
4. You are required to add three other **unique** features to the back-office.

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 each week will see you forfeit the marks for that demonstration. The demonstrations will take place during your scheduled practical class for the weeks specified.

Demo	Week Beginning	Marks
#1	7/11/2022	5%
#2	14/11/2022	5%
#3	21/11/2022	5%

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.

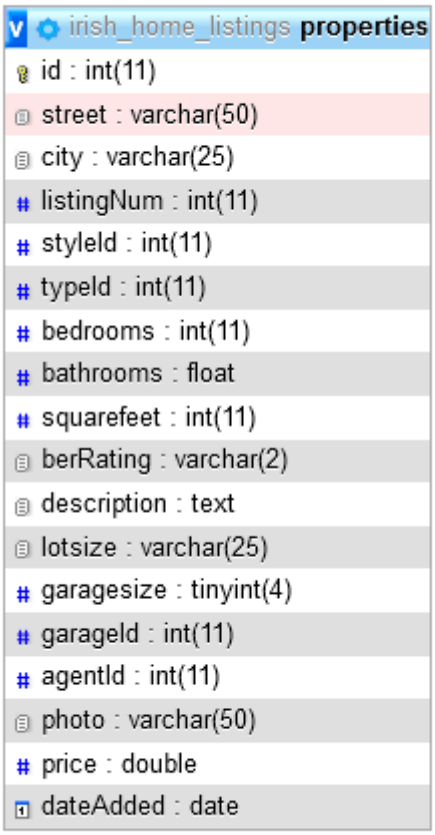
Some markdown tutorials include….

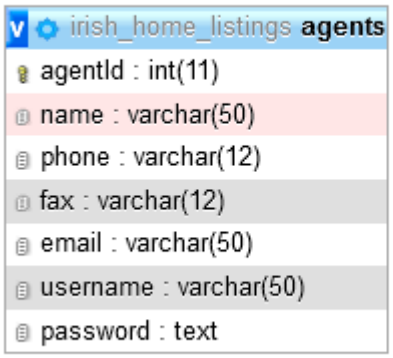
https://guides.github.com/features/mastering-markdown/

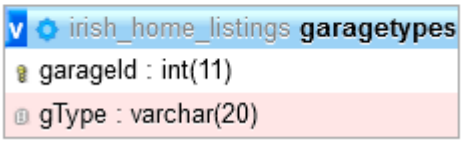
https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

https://www.markdowntutorial.com/

An overview of the database is as follows:







The following is a breakdown of the structure of each table in the database along with a sample record. There are no relationships between any of the tables – it is highly recommended that you address this.

Properties Table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**id**	int(11)			No	*None*	AUTO_INCREMENT
2	**street**	varchar(50)	utf8_general_ci		Yes	*NULL*	
3	**city**	varchar(25)	utf8_general_ci		Yes	*NULL*	
4	**listingNum**	int(11)			Yes	0	
5	**styleId**	int(11)			Yes	0	
6	**typeId**	int(11)			Yes	0	
7	**bedrooms**	int(11)			Yes	0	
8	**bathrooms**	float			Yes	0	
9	**squarefeet**	int(11)			Yes	0	
10	**berRating**	varchar(2)	utf8_general_ci		No	*None*	
11	**description**	text	utf8_general_ci		Yes	*NULL*	
12	**lotsize**	varchar(25)	utf8_general_ci		Yes	*NULL*	
13	**garagesize**	tinyint(4)			Yes	0	
14	**garageId**	int(11)			Yes	0	
15	**agentId**	int(11)			Yes	0	
16	**photo**	varchar(50)	utf8_general_ci		Yes	*NULL*	
17	**price**	double			Yes	0	
18	**dateAdded**	date			No	*None*	

id	street	city	listingNum	styleId	typeId	bedrooms	bathrooms	squarefeet	berRating	description	lotsize	garagesize	garageId	agentId	photo	price	dateAdded
1	88 Lagmore Glen	Belfast	784571	1	2	3	2	1900		Lovely home in a great neighborhood. Plenty of spa...	80x110	1	1	2	784571.jpg	200800	2022-10-01
2	26 Lawrence Hill	Derry	598795	5	1	7	4	2500	0	This elegant red-brick terrace house, dating from ...	80x120	2	2	2	598795.jpg	129900	2022-10-05

Agents Table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**agentId**	int(11)			No	*None*	AUTO_INCREMENT
2	**name**	varchar(50)	utf8_general_ci		Yes	*NULL*	
3	**phone**	varchar(12)	utf8_general_ci		Yes	*NULL*	
4	**fax**	varchar(12)	utf8_general_ci		Yes	*NULL*	
5	**email**	varchar(50)	utf8_general_ci		Yes	*NULL*	
6	**username**	varchar(50)	utf8_general_ci		No	*None*	
7	**password**	text	utf8_general_ci		No	*None*	
8	**agentImage**	text	utf8_general_ci		No	*None*	

agentId	name	phone	fax	username	email	password
1	Sue Roberts	061 451890	061 451880	Sue.Roberts	sue@ihl.ie	suepass
2	Natasha Watkins	061 451891	061 451881	Natasha.Watkins	natasha@ihl.ie	natashapass
3	Chris Clarkson	061 451892	061 451882	Chris.Clarkson	chris@ihl.ie	chrispass

Property Types Table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**typeId**	int(11)			No	*None*	AUTO_INCREMENT
2	**pType**	varchar(20)	utf8_general_ci		Yes	*NULL*	

typeId	pType
1	Residential-single
2	Residential-multi
3	Commercial

Styles Table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**styleId**	int(11)			No	*None*	AUTO_INCREMENT
2	**pStyle**	varchar(20)	utf8_general_ci		Yes	*NULL*	

styleId	pStyle
1	Bungalow
2	Semi Detached
3	Detached
4	Cottage
5	Terrace
8	Duplex
9	Condo
10	Apartment
11	Other

Garage Types Table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**garageId**	int(11)			No	*None*	AUTO_INCREMENT
2	**gType**	varchar(20)	utf8_general_ci		Yes	*NULL*	

garageId	gType
1	attached
2	detached
3	carport

Note:

Once the submission date has expired, I will be testing your project on my computer. It is important that you continually test you code/project on machines other than the one you are developing it on.

Your solution **must use JPA and connection pooling**.

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

Inserts/Updates will affect multiple tables in the database.

You are encouraged to change the structure of the database to achieve your aims.

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

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

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

The deadline for this assignment is 23:59 on Friday, November 25th. 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 (ideally you should look to commit after you implement every Use Case). Failure to do so (commit once a week), will see your final mark reduced by 3% for every weekly commit you miss.

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