A company called LIT Realty has recently formed in Limerick and has decided to base itself in the industry incubation centre in LIT's Moylish Park campus. LIT Realty sell property all over Ireland and currently advertise their properties in the national press. To facilitate these advertisements, LIT Realty have developed a complete database of their properties and their selling agents. You must develop a complete web application for LIT Realty that will offer the following functionality. I have broken the functionality down into four categories.

Category #1: Agent Functionality (30%)

1. Log-in and log-out feature for agents (each agent must be authenticated using their user-name and password from the database. Once logged in, each agent must able to:
    1.1. View, edit, delete 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. Any deletion must require the agent to confirm whether they are sure they want to proceed with this deletion or not.
    1.2. Each property that appears on the website has a vendor (who has trusted LIT Realty 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. Only authenticated agents can view vendor information. It is possible that one vendor may be selling more than one property. There is currently no "vendor" table within the database.
    1.3. Once an agent has been authenticated, every subsequent page they visit must display their (profile) picture.

Category #2: Customer Functionality (25%)

1. Every customer will be able to search the database for a property based on its price and location. The search results should be presented in tabular form. This table must include a thumbnail image for the property. 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 property in question (this is in effect a drill-down: extra information such as the square footage of the house, property style, property-type, garage type, number of bathrooms, number of bedrooms as well as details of the agent responsible for selling the house should be displayed – you should also display the larger images for the property). You must also mark on Google maps the location of the property
3. On the drill-down page for each property a customer should be able to add a property 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 property 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. The ability to view the most recently added properties to the system. This is a list of **any** properties in the database (regardless of their location/price etc) which have been added in the last 7 days.

Category #3: Unique Feature (30%)

You must add a unique feature to this assignment. The feature you add must complement the existing functionality.

For example, you could consider using a 3rd party API such as (but not limited to), Twitter (https://dev.twitter.com/), Google (https://developers.google.com/), Facebook (https://developers.facebook.com/), FourSquare (https://developer.foursquare.com/), Yelp (http://www.yelp.com/developers/) etc.

However, the unique feature **should include some custom code**. Incorporating API's/code from online will only get you so far.

Weekly Demonstration of Your Work in Class (15%)

You are required to demonstrate your code weekly in class to me between now and the final submission on December 4th 2018.

Failure to demonstrate your work in a given week will see you forfeit your marks for that demonstration. The demonstrations will take place on the following days.

Demo	Date	Marks
#1	23/10/2018	2%
	30/10/2018 (Reading Week)	
#2	6/11/2018	4%
#3	13/11/2018	3%
#4	20/11/2018	3%
#5	27/11/2018	3%
	4/12/2018 (final submission there will be no demo on this day).	N/A

Clear progress will be expected from week to week.

Note:

Once the submission date has expired, I will be testing your code/project on my laptop/office 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.

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.

Inserts/Updates will affect multiple tables in the database.

Only authenticated agents should be able to access features 1.1 and 1.2.

Customers are not expected to log-in and LIT Realty do not store any details about them other than their favourite properties (which may not be stored on the server).

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 XXS and all data must be validated before entering the database.

All passwords stored in the database must be encrypted. Currently, the passwords of all agents appear in plain text – this will have to be addressed.

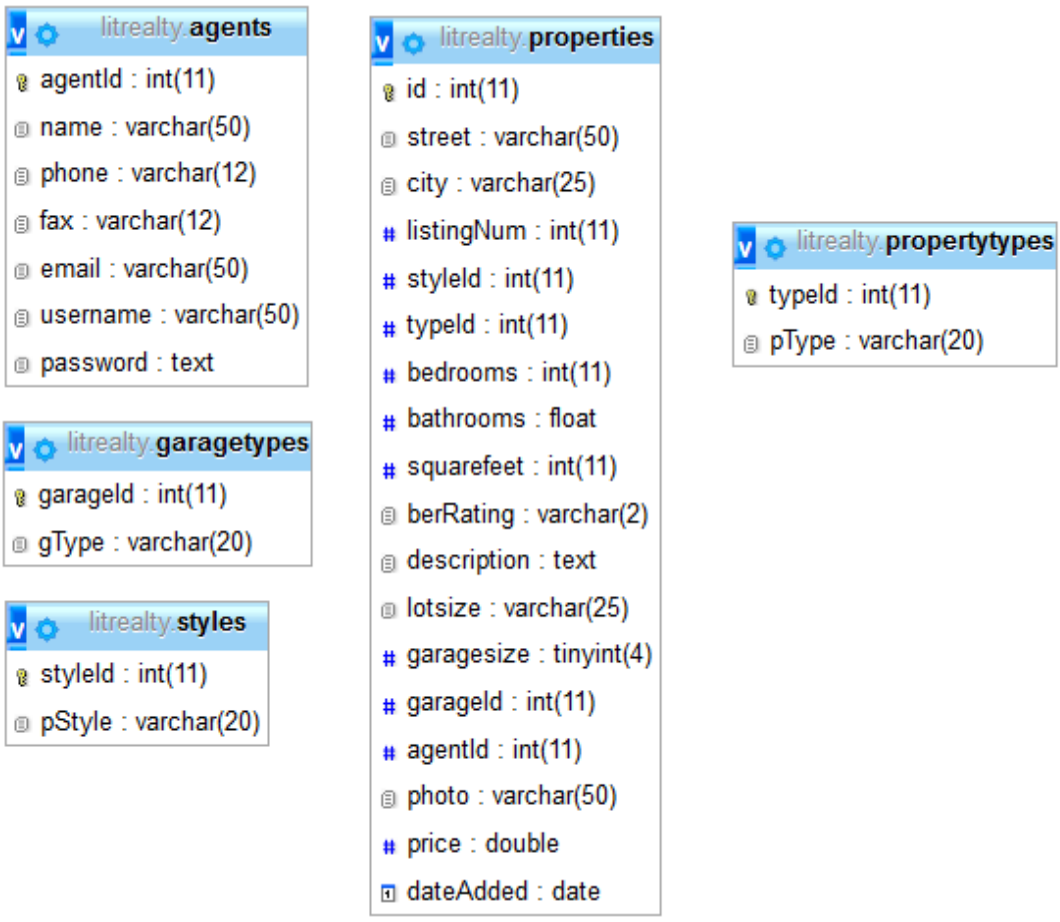
You are permitted to change the structure of the database to achieve your aims.

Your application must be as user friendly and intuitive as possible.

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).

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 – this is something you may decide to change.

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	Befast	784571	1	2	3	2	1900		Lovely home in a great neighborhood. Plenty of spa...	80x110	1	1	2	784571.jpg	200800	2016-11-01

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	email	username	password	agentImage
1	Sue Roberts	555-1234	555-9876	sue@homesellers.com	Sue.Roberts	suepass	1.jpg

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

The deadline for this assignment is Tuesday, December 4th at 6pm.

I have created a private GitHub repository for each of you (https://github.com/lit-alan/KNum). You must use this repository for your work during this assignment. I require you 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 2% for every weekly commit you miss.

In reality, you will more than likely commit your work more than once a week.