

## SFWRTECH 4WP3 – Advanced Web Programming - Assignment #5

### Weight

4% of total course grade

### Due date

Monday February 28th at 11:59pm

### Primary learning objectives

- Use the model-view-controller design pattern in an application
- Create an application with CRUD database features

### Getting Started

Use **as5\_starter\_code.zip** on Avenue to Learn as a starting point for this lab. It contains a controller, view, and model that work together to provide basic CRUD functionality for a realtor database, as well as a script you can run (initdb.js) first for creating and populating the realtor database table.

### Requirements

For this assignment we will be working on the “Reatlors” page of our Hamilton Homes website that we have been creating since Assignment #1.

Add the following functionality to the MVC application. All of the functionality should be implemented in the expected way for an MVC application using the relevant concepts (controller, view, model, template object, actions).

- Column sorting
  - Each column header in the table of realtors should have an “up arrow” and “down arrow” image icon that when clicked causes the page to reload with the table sorted by that column (ascending or descending depending on the arrow clicked).
- Form validation
  - When updating or adding a realtor, perform form validation. If any form input is blank when the user clicks the submit button, that element is considered to have

a form validation error, the update or add should not take place, and instead the page should reload. When the page reloads, any form validation errors should appear as an unordered list below the form (e.g. "Specialty is blank"), any form inputs with form validation errors should be highlighted with a red border color, and the form inputs should be populated with the previously entered form input values. This form validation must be conducted on the server side, and the highlighting and populating of inputs must be done using the template object. No marks will be awarded for conducting this work with front-end JavaScript code.

- Likes column
  - Add a column to the realtor table called "likes". Note that to do so you will need to modify the database table definition itself as found in initdb.js. The likes field should be an INTEGER type value that is initially set to 0 for all realtors. The likes field and values for each realtor record should appear in the table when the user views the application as a new column, and the likes field must be added to the update and add functionality.
- Like feature
  - There should be a column in the table when the user views the application with the column heading "Like", similar to the Update and Delete columns, with a "Like" thumbs up image icon in each row of the column (it must be an image, find whichever you prefer online). When the user clicks on the "Like" image for a realtor record, the realtor's like field should be increased by 1 in the database, and the page should reload reflecting the new like count for the realtor.
- Layoffs feature
  - Create a "layoff" link below the table. When the link is clicked, any realtor with less than 2 likes should be deleted from the database table, and the page should reload.
- Random realtor feature
  - Create a "random" link below the table. When the random link is clicked, a randomly generated realtor should be inserted into the table, and the page should reload. The first name, last name, and specialties for the randomly generated realtor don't need to be real names, they can just be random characters, or you could pick randomly between an array of pre-defined values. The likes should make sense though (e.g. pick something on a range of 0 – 10 or something reasonable like that).

Use Bootstrap to nicely style the application. Add the navigation bar you have been using for your Hamilton Homes website to the page, with the Realtors navigation link active/highlighted.

No other libraries, frameworks or plugins except Bootstrap or jQuery can be used.

## Submission

Zip your solution as assignment5.zip and upload it to the dropbox on Avenue.

## Marking rubric

Component	Description	Marks
MVC design pattern	MVC design pattern used to implement the functionality (6 x 5 marks per functionality below)	/30
Column sorting	Column sorting functionality	/10
Form validation	Form validation feature	/10
Like column	Like column, add/update/read modified	/10
Like feature	Liking of realtors	/10
Layoffs	Laying off of realtors based on like count	/10
Random realtor	Inserting a random realtor	/10
Style	Neatly styled application with Bootstrap	/10
Total:		/100