CS 460 Lab#6 Process Notes

Database restoration from .bak file using AdventureWorks2014.bak sample db from Microsoft:

* Download .bak file or .zip that contains the .bak file.
  + Save to location that is not within the Users file structure.
    - This can cause permissions issues.
* Open Database Object Manager
  + Right click the desired db (local db for this instance).
    - Add Query…
  + We now need to find the logical names for the database and the log files that are stored within the .bak file. We do this with:
    - RESTORE DATABASE yourDBName
    - FROM DISK = ‘c:\pathToBakFile’
    - GO
  + Run query and this will get back the names we will need.
  + Then we run this code with the names:
    - RESTORE DATABASE yourDBName
    - FROM DISK = ‘c:\pathToBakFile’
    - WITH MOVE ‘logicalNameOfDatabase’ to ‘c:\pathToAppDataProjectFolder\yourNamedDB.mdf
    - MOVE ‘logicalNameOfLogFile’ to ‘c:\ pathToAppDataProjectFolder \yourNamedDB.ldf’
  + Run this.
* Reverse Engineer the database to create the model classes we will use.
* Project -> Add Item….
  + ADO.Net Entity Data Model
  + Code first from database
  + Select your database
    - Select what parts of the table you will want to use.
* The connection string is automatically generated and added to your Web.Config file!

Notes on JS

* For using JavaScript in MVC (or anywhere else for that matter) contain your javascript file in a $(document).ready(function () {your JS code…}) to make sure that the jQuery files have been loaded before the JavaScript to avoid errors
* Also in MVC use the @RenderSection(“scripts” , required: [true or false]) in your \_Layout.cshtml file.
  + Include this wherever you want to load the script pages that you specify in your individual View files.
    - You will probably want this to mainly be below all the other Layout scripts but that is up to you.
    - Remember that you will need to place all your JS BEFORE the jQuery files.
    - If you try to load your JS file from your individual page WITHOUT doing the @RenderSection in Layout file or just exclude the JS file from being associated with that code, you cannot use jQuery.
      * The JS file will be loaded in the body section of the code and thus, included within the \_Layout.cshtml files @RenderBody code section and obviously above all the script renders.

JS Code and some helpful tips:

* Spent a bit of time trying to get my landing page to have some nice functionality. Found a nice snippet of code for easily managing my div visibility when each div is connected to a particular button.
  + This code works because typically with nav buttons we will have them all contained within a list: ul>li. The ul will actually give each of these list elements an index. Though this we can use the indexs to point to our divs
  + Place all of the divs into one container div. Now we can access this divs children with indexes just like we do with the lists.
    - Note that this will only work out if there are the only things in the container div!!
      * This could be a problem and feels a bit hacky but we can improve on this later.

$(".banner-nav-link").on('click', function () {

/\*

This will get get the drop down menus which are equal to the index of

the button pressed. ie. mens-drop-down is in index 0 of the ul. This will get

the div which is in the 0 index position of the banner-drop-downs container.

\*/

var $dropDown = $("#banner-drop-downs").children().eq($(this).parent().index());

//Toggle that divs visibility

$dropDown.toggle("slide", {direction: 'up'}, 750);

//Hide the rest.

$dropDown.siblings().hide("slide", { direction: 'up' }, 750);

* + - * });