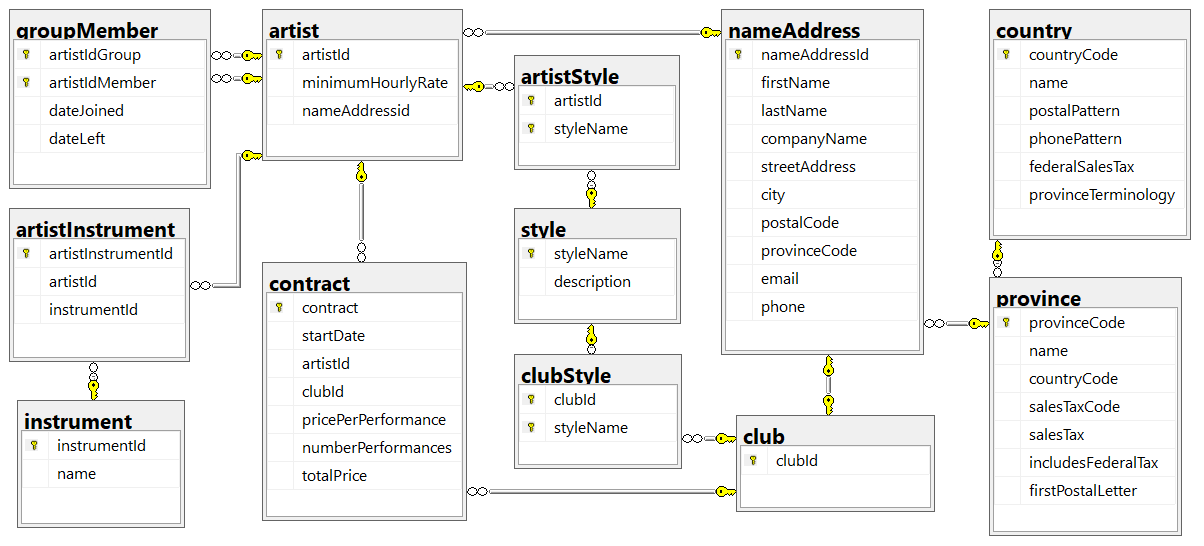
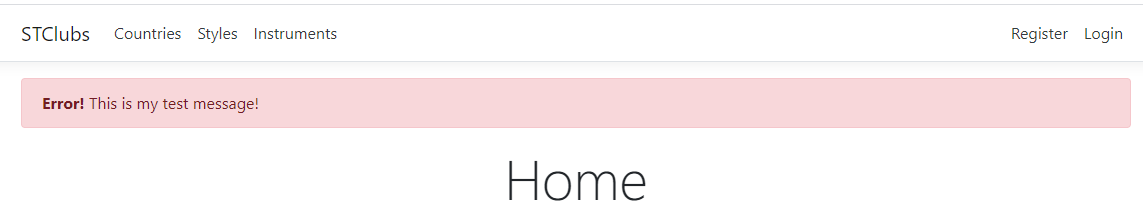
## Ms Web Tech Assignment – Persistence, sorting & filtering

In this assignment, you will pass & persist (remember) a key. When the user selects a **Country**, you will provide a link to list all **Province** records for it. New and updated provinces will be assumed to be for the selected country, so CountryCode will not be an input field. As always, where you see “XX”, substitute your name.



### Project Setup

1. Continue using your ***XXClubs*** project from assignment 1.
2. On ***~/Views/Shared/\_layout.cshtml***, display the TempData variable “message” right before content pages sections. Use bootstrap alerts: <https://www.w3schools.com/bootstrap/bootstrap_alerts.asp>. This should be partial view called Message. To test add TempData["Message"] = "This is my test message!"; to your Home/Index **action**.



1. **Add support for Session variables to your site’s Startup.cs.**  This is in bold because a few students last year never did this and their session variables did not work. Rather than get rid of the session stuff, they doubled-up and used cookies as well. That did not turn out at all well: one set tripped over the other.

### XXCountryController

1. In the Index view for this controller, make the country code a hyperlink that passes it to the Index action of the ***XXProvinceController***.
   1. Order the country listing by name
   2. Country details should display list of provinces if any.   
      **List of provinces should be partial view** since it is going to be used in two different places.

### XXProvinceController

1. Generate a controller for the *province* table, called ***XXProvinceController***, with full CRUD support and views. Make sure that create, edit, delete and details pages all work.
2. In the *Index* action:
   1. If the *CountryCode* is in the URL or a QueryString variable, save it to a cookie or session variable.
   2. If there is no *CountryCode* passed in the URL or QueryString, see if there is a cookie or session variable with it.
      1. If there’s no *CountryCode* in the cookie or session variables either, return to the ***XXCountryController*** with a message asking them to select a country if they want to see country specific provinces.
   3. Regardless of where you got the *CountryCode* from, use it to filter the listing to only show provinces on file for that country
      1. Order the listing by *name*
   4. If the country name had also been passed, persist it the same way … otherwise, fetch the record from the country table, then extract & persist its name.
   5. Do not show the *CountryCode* … instead, show the country’s name in the browser’s title and in the view’s heading … “Provinces for ‘Canada’”.
3. In the Create and Edit views:
   1. Do not show an input field for *CountryCode* … the user has requested provinces for a specific country, so stay in that mindset.
      1. When posting back, the *CountryCode* field still has to be inserted into the new/modified record, so hide it and its value on the views. A hidden field without a value posts back as an empty string, killing numeric receiving parameters.
   2. Ensure that there is no province on file with the same name or the same province code, unless you are updating that record.
4. In all views for this controller:
   1. Show the ProvinceCode, but only allow Create to change it.
   2. Modify each browser title and page heading to indicate its purpose and the country the province is for, like “Add a Province to Germany”, or “Details of a Province in France” and so on … watchyourpunctuation.
   3. Do not show the *CountryCode* on any view.
5. For fun, use the province terminology for the selected country, such as “Prefecture” in France or “Metropolitan Area” in Japan. Where should this term be found?

### Hand In

1. Zip and upload your project folder to the [D2L upload folder](https://www.econestoga.ca) (course tools 🡪 assignments) for this assignment. Only the last upload is retained.