FAST NUCES – Fall 2017 IPT – Assignment 4

Due on: Sunday 10th December 2017 by 17:00 hours

Please submit via SLATE. Please include a single .zip/.rar file (source code only, no binaries) and name your submitted file as "Assignment4IPT<your reg #>"

1. Consider the Catalog Database shared in Week13
2. Create a new table called Product with the following attributes:
    a. Product Name
    b. Primary Product Category
    c. Product Description
    d. Active (whether the product is active or not)
    e. Product Price
    f. Other product attributes; Use an XML/JSON column to store one or more of the following product attributes:
        i. Available colors - **mandatory**
        ii. Available sizes - **mandatory**
        iii. Dimensions (length x width/depth x height and units in cm/inches/meter/ft) - **mandatory**
        iv. In case of electrical product: input watts
        v. In case of kitchen appliances or cloth: Material type
        vi. Warranties/Guarantees where applicable
3. In addition to the relationship between Category and Product depicting the primary product category (2b above), a product may be associated with one or more secondary product categories.
4. Populate this database with about 50,000 products, each with 0 to 5 secondary categories.
5. Make a browser based application with MVC and Entity Framework with a search screen to search for products with following filters:
    a. A text box that will do search on product name, description and/or product categories.
    b. Allow user to specify category. Should show indented list of all categories. Default is 'All Categories'. If a specific category is specified, show all products that have either this primary category or one of the secondary categories.
    c. A price filter with options to search for product up-to a certain price, at least a certain price or a price range. Default is 'No price filter'
    d. A filter for colors or sizes. Default is 'All Sizes' and 'All Colors'
6. The result should be shown in a grid/table. The default sort order is 'Product Name'. The user should also be able to specify Sort order (consider providing these options: Product Name, Price Low to High, Price High to Low)
7. The results of the search should be paged.
8. Include the following in the result:
    a. Product Name
    b. First 100 characters of description
    c. Product price
    d. All categories to which this product belongs to…
    e. All sizes…
    f. All prices…
    g. Dimensions and its unit
9. Carry out the above using two different techniques:
    a. Use EF & LINQ to filter and Use PagedList.Mvc package for pagination

b. Create a custom stored procedure that is sent all filter parameters and sort order as well as current page to be retrieved and the page size. The stored procedure should return the resultant page as well as total number of records; Alternatively, create two stored procedures the first one will provide the total records and the second one will return the results of the given page.
(References: https://basquang.wordpress.com/2011/03/18/stored-procedures-paging-solution-in-asp-net-mvc-2/
http://aspmvcsnippets.com/Article/Stored-procedure-with-pagination.aspx)

10. Provide an 'Export Summary' button that should produce a summary for the search criteria and show it as a popup:

    Category Name **Total # of Products** **# of Products Matching the Criteria**

Carry out the Export work by using LINQ to query the database and store (either ALL or) qualifying products data in a collection and use LINQ with collections to generate the summary. In the above sample, if a product belongs to 3 different categories, it would come three times in the LINQ query and would be part of the counts of three different categories.

Objectives

A. To assess LINQ skills
B. To assess Stored Procedures skills