|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **INF 354 Homework Assignment** | | | |
| **Assignment #:** | **9** | **DUE DATE:** | **Next Class** |
| **Topic:** | Mini-Project | | |
| **Instructions:** | ***Work in groups of 4-5.***  **NB: Base this homework off your final year project as a practise if possible.**  Create a SQL Server Database with the following [3]:   1. Main object table (e.g. Client). [0.5] 2. Look up table to the main table (e.g. Title). [0.5] 3. “Another” main Object table (e.g. Furniture). [0.5] 4. Look up table for “another” main object table (e.g. Furniture Type). [0.5] 5. Resolve the many-to-many relationship between the two object tables with an extra “datetime” attribute (e.g. ClientFurnitureRent). [1]   Build an ASP.Net WebForm application with the following functions [16]:   1. CRUD the two look up tables. [1] 2. Register client function including the validation that the client doesn’t already exist in the database. [1] 3. CRUD main object and “Another” main object table with lookup table in a drop down list. [2] 4. Fuzzy search on all attributes of main object and “Another” main object table. Order by ascending. This can be in two different web pages. [2] 5. Assign “another” main object to a searched main object. (e.g. Client “Bob” Rented a Couch) [1] 6. Search for one main object that has a relationship with another main object through the “another” main object (e.g. All other clients that have rented out the current client’s furniture). [2] 7. Display all the look up table attribute values of “Another” main object table from a specific main object record. This must be done for a specific day (e.g. Yesterday, Client x took out the following furniture types: … etc.). [2] 8. Create a dashboard web page that displays the relationships between the two lookup tables (e.g. Graph displaying counts of client titles for different furniture counts)    1. Code [3]    2. Display of Graph [2]   *Note: you are welcome to use graph.js, D3, Google Charts, etc.*  UI Design needs to be responsive taking a “mobile first” design approach. [4]   1. Mobile responsive for extra small screen. [1] 2. Mobile responsive for small screens. [1] 3. Mobile responsive for medium phones. [1] 4. Mobile responsive for large phones. [1]   If the following technologies are not used, no marks will be awarded:   1. Linq and/or Lambda 2. Entity Framework 3. ASP.Net (either MVC, or just normal Webforms) 4. Bootstrap | | |
| **Reference materials:** | **Google, Linq, Lambda, Entity Framework, Bootstrap**  [**http://Getboostrap.com**](http://Getboostrap.com) | | |

**RUBRIC**

|  |  |
| --- | --- |
| **Outcome** | **Mark** |
| **Marked assignment** | **20** |
| **Q & A (5 questions at 0.2 per question)** | **1** |
|  |  |
| **Mark \* Q&A** | **20** |