**Assignment – The Two APIs** [v2]

**To write two Web APIs:**  
1. GET: ItemsByName -Implemented  
2. DELETE: Category -Implemented

**Mandatory conditions:**  
1. Use C#, ASP.Net Core 2.2/3.1 or Framework 4.7.2 with Entity Core/Framework (EF). -Used .Net Core  
2. Follow multi layered architecture (mandatory) with SOLID principles/design patterns wherever applicable. -Followed All Design Principles.  
3. Name is required. Accepted string length is min = 3 and max = 12.   
4. The GET API ‘ItemsByName’ should get a list of Item(s) with ‘Category Name’, ‘Sub-Category Name’, ‘Item Name’ and ‘Item Description’   
 where ‘Item Name’ is like the name-string provided in query string.  
 It should get all records if ‘Item Name’ query string is not provided.

-Implemented the Above Function but could not return some foreign key values. Some issue from the .net Core Side. Also tried to implement this from Stored Procedure usage  
5. The GET API ‘ItemsByName’ API should have paging capabilities. The default load should be page 1 (first page) with 10 records. – Implemented  
6. For DELETE Category API, delete the Category by ‘Category Name’. -Implemented  
7. Database tables should have foreign key references. -Implemented  
8. Write unit tests using Nunit/Xunit framework and Moq for mocking the dependencies. -Not Implemented. Don’t have much hands on NUnit.  
9. Push the code (including SQL scripts and screenshot of Swagger) in any free GIT repo for Code Review. Do not use Harman or any company name in the code/repo. -Successfully Pushed the code to GitHub.  
10. Implement Swagger. -Implemented

**For the above assignment, create three SQL tables:**  
1. Category  
 Id (pk)  
 Name  
2. SubCategory  
 Id (pk)  
 CategoryId (fk)  
 Name  
3. Item  
 Id (pk)  
 SubCategoryId (fk)  
 Name  
 Description-Implemented the above SQL Database. Attached in the Github Link are the scripts and ER Diagram.

**Assignment Note:**  
- Please make sure to comment-document your code.  
- Do not writer unwanted code/API. Remove unwanted code.