Final Exam

Term: Summer 2023 **Professor:** Mahboob Ali
Course: PROG32356 – .NET Tech. using C# **Email:** mahboob.ali@sheridancollege.ca

Please Be Advised That:

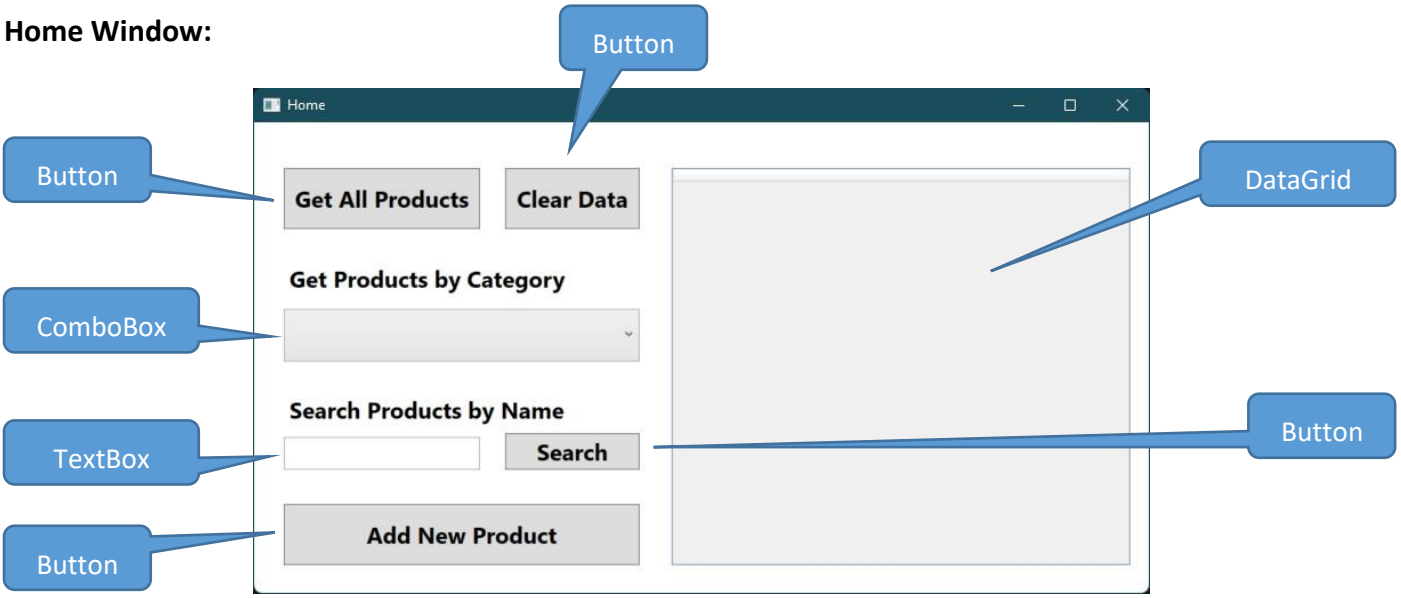
1. Exam must be completed as an individual effort. Do not collaborate with anyone or share it with any individual, party or entity.
2. Do not share this exam with anyone or any 3rd party without the written consent of the professor.
3. ZIP the project and upload it to SLATE by due date/time, mentioned in SLATE.
4. All online submissions will be done via SLATE (Email submissions will NOT be accepted).
5. Corrupt/incorrect submissions will be graded as zero.
6. This is a 3-hour exam which will be attempted over the period of 3-days. Therefore:
    a. No late submissions will be accepted as it is an exam.
    b. No 3-day grace period is allowed.
    c. No extensions will be provided to students with accommodations.
7. Make sure your laptops are in good working condition. If something does not work, or any application or system crashes, you will be responsible to fix it and submit your work on time. No excuses will be accepted during the exam.
8. Make sure your Visual Studio is in good working condition. No excuses will be entertained regarding issues with Visual Studio.
9. Please refer to the Academic Integrity Policy.
10. Refer to the School of Applied Computing's Academic Procedures for Evaluations for more details regarding missed work: Procedures for Evaluations.
11. If you take code-snippets from external sources such as StackOverflow, make sure to provide references to the source's webpage as comments in your code.

Copyright Disclaimer:

1. Make a ***WPF App (.NET application)*** in Visual Studio and name it as **FinalFirstnameLastname**.
2. The app should be able to display and add products from the Northwind database.
3. **Download the *Northwind-Sample-Database-for-LocalDB.sql* file and use the SQL script it contains to create and populate the database.**
    - When creating a new database, it must be named ***Northwind***, otherwise, the script will not run properly.
    - Use **LocalDB** (meaning like MSQLLocalDB) as database source unless you've got written permission from your professor to use a different data source such as SQL Server or MDF file.

4. You can use Connected, Disconnected or typed dataset for this.
5. The application should have two windows:
    - Home | Add Product

Home Window:



1. This window has:
    - `Button`s: four buttons to get all products, clear fields, search product and add new product.
    - `ComboBox`: to display list of categories.
    - `TextBox`: to enter the product name to search for it.
    - `DataGrid`: to display the fetched data of products from the Products table.
2. When the user clicks the "Get All Products" button, fetch all the products data from the Products table and display it in the `DataGrid`. Also, populate the categories `ComboBox` with categories.
3. The "Clear Data" button clears all the data from the window, that is, it clears the `DataGrid`, the `ComboBox` and any text from the `TextBox`.
4. The user can get all the products from a selected category. When the user selects a category name from the categories `ComboBox`, fetch and display all the products that belong to the selected category in the `DataGrid`.
5. The user can also search for a product by name. Fetch and display all the products in the `DataGrid` that match the name entered by the user.
    - ***Partial name matches should also fetch the products.***
6. On clicking the "Add New Product" button, a new window should be displayed which can add a new product to the Products table.
    - Display the Add window as a dialog box. User should not be allowed to go back to the Home window as long as the Add window is open.

Add New Product Window:

1. This window has:
    a. Two `TextBox`es: to enter the product name and price.
    b. `ComboBox`: to display the list of categories.
    c. `Button`s: to add the above information to the database and to close the window.
2. On the initial window load, the categories `ComboBox` is filled with category names.
3. The user can type the name of the product and its price in the textboxes.
4. Select the category from the category `ComboBox`.
5. Then, clicking the "Add Product" button adds the information to the Products table.

Marks Breakdown:

- 10% - for correct submission.
- 20% - for proper implementation of EF.
- 40% - home window.
- 30% - add product window.

Submission:

1. Once done, ***ZIP the solution folder*** and upload it to **Assignments** on SLATE.
    a. ***Double-check your submission by downloading it and running it.***
2. You are to submit .ZIP and .TXT files, separately:
    a. Upload the **.ZIP** file of your assignment to SLATE.
    b. Upload the **.TXT** files to SLATE.
3. You must copy and paste all of your source code from your C# files into separate plain text files.
    a. You can copy and paste the source code into Notepad.
    b. Create a separate .TXT file for each source code file.
    c. You do not need to copy the XAML source code. Only copy/paste the .CS source code.
4. You don't have to format this code - it's used by TurnItIn (the originality checker in SLATE, which is a piece of software that checks your submission for plagiarism against other submissions in the college, in other colleges, from the web, and various other sources).
5. Submit this document in addition to your assignment ZIP file.
    a. DO NOT add it inside your zip/rar file - it must be a separate file.
    b. This is used for TurnItIn (it won't examine the contents of zip/rar files).

6. **Note:**
    a. If these submission instructions are not followed, Grade 0 will be granted.
    b. If TXT files are not provided, Grade 0 will be granted.