**.NET CORE CA Project**

Project Specifications

The objective of this CA project is create a Shopping Cart web application using .NET Core.

You can assume that the users who are purchasing have already registered with your system. You just need to provision a login screen for the user so that you know which users have purchased what items. Also, assume that all users’ credentials are stored in the database (passwords must be hashed in the database and client should not send the user’s password in the clear [Used Bullet or star]).

The Login screen should look like this. Remember to perform proper validation on users’ inputs and display appropriate error messages.

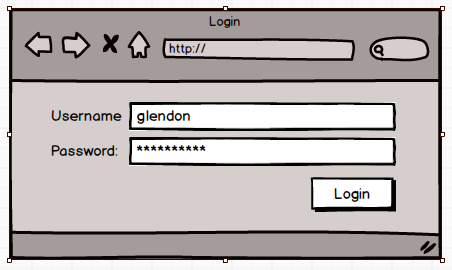


Figure 1:Login

Once logged in, the user will be brought to the Gallery page (Figure 2) to purchase.

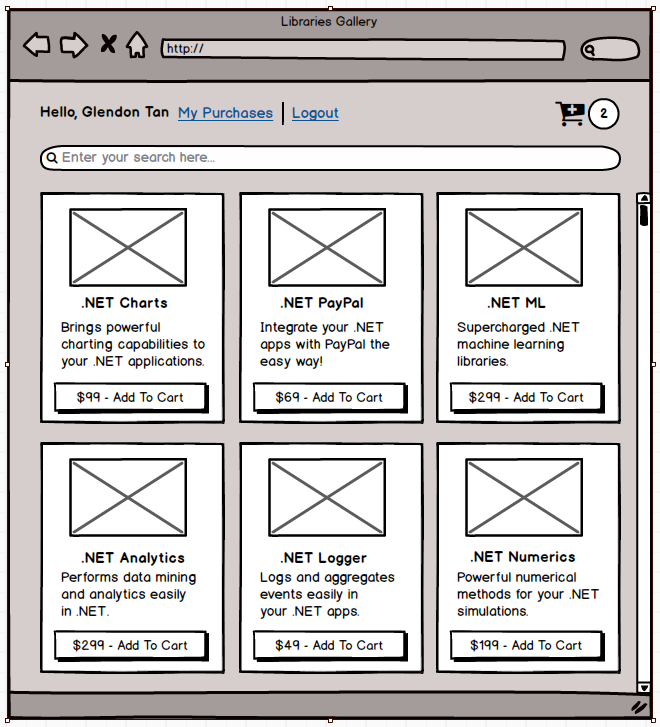


Figure 2: Gallery page

Display the user’s name on the left and a link for him to logout of your system. There should be a Search bar for users to search the products he is looking for. Once he hit Enter, display the list of products that satisfy his Search string (Figure 3). Once he has cleared the Search bar, list all the products **again**.

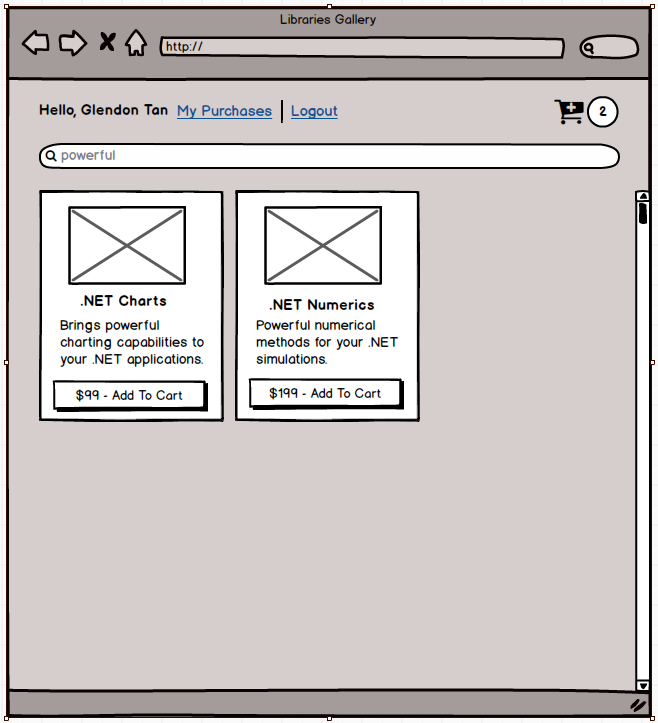


Figure 3:Search

If he clicks on My Purchases, he will be brought to My Purchases page to view all the purchases he has made.

When the user clicked on the “Add To Cart” button, increment the number beside the Cart icon. If the Cart is currently empty and he clicked on the same item twice, the number besides the Cart icon should be 2.

When he clicks on the Cart icon, display the products as shown in Figure 4.

Allow him to change the quantity of each of his selected product and re-compute the Total amount on the top right. If he clicks Continue Shopping, brings him to the Gallery page (Figure 2).

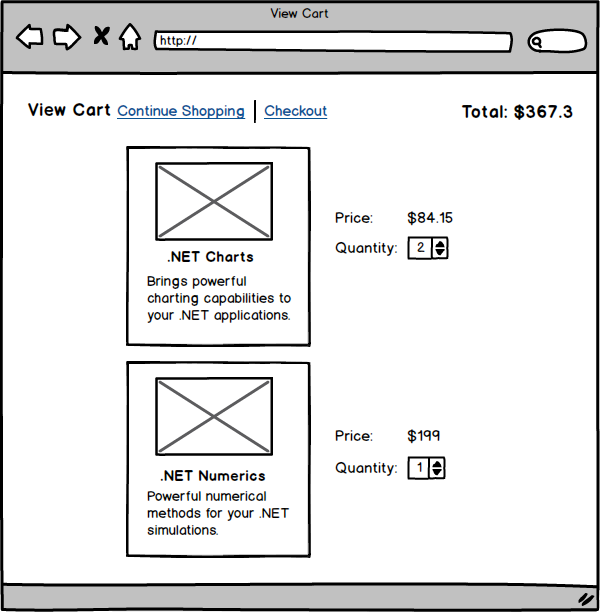


Figure 4: View Cart

If he clicks on Checkout, we assume that the purchase has been successful and brings him to My Purchases pages (Figure 6). Each purchased product will have a unique activation code. If the user has purchased multiple copies of a product, he would have multiple activation codes. Display multiple activation codes with a combo box.

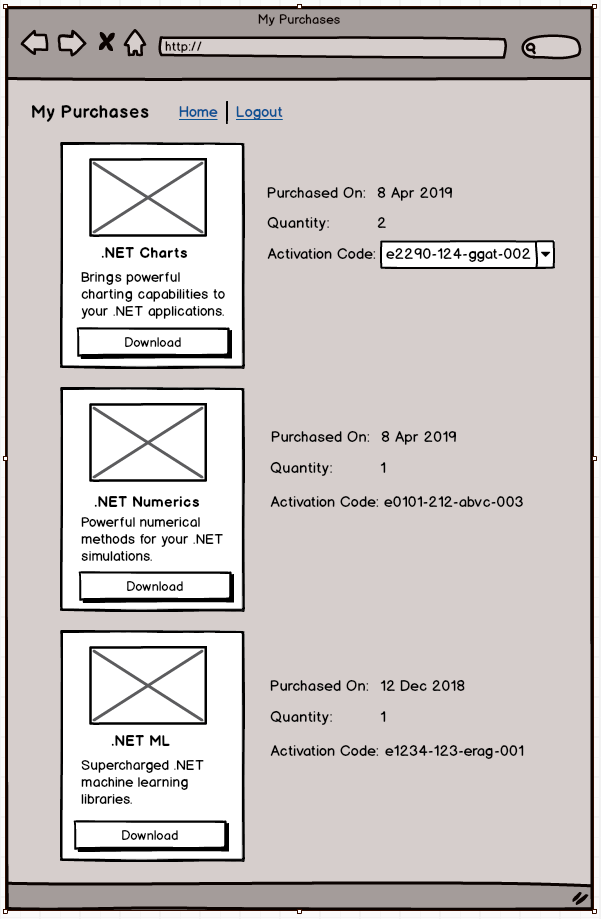


Figure 6: My Purchases

The program flow for a user that has logged into before putting items into his cart is depicted in Figure 7. Your program must follow this flow.

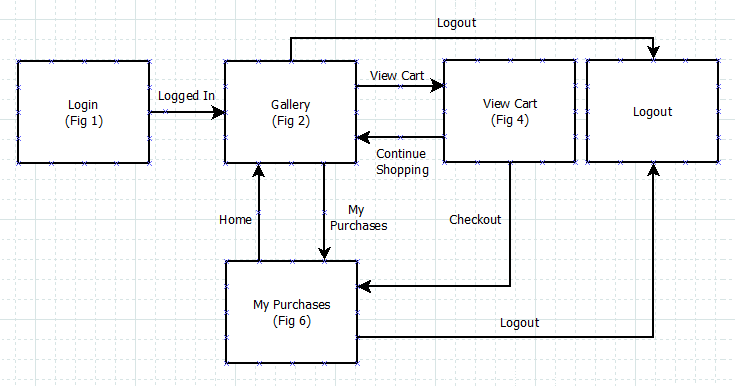


Figure 7:Workflow with Login

There would be occasions where users are directed directly to the Gallery page (e.g. via a bookmark). We want to allow these users to place items on the cart but before they can checkout, we need to get them to login (again, we assume that the user is already one of our customers). Once login is successful, we would complete the purchase.

The program flow for users who did not log in before placing items into the cart is depicted in Figure 8. Again, your program must follow the flow.

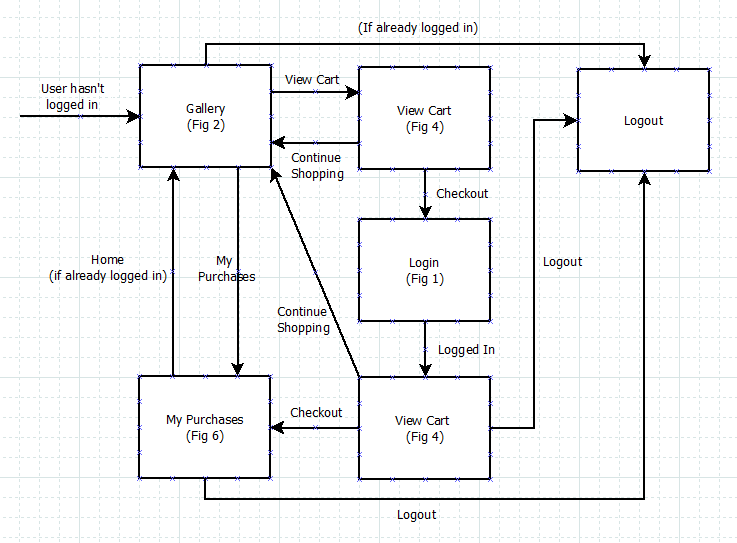


Figure 8: Workflow without Login

The basic functionalities are:

1. Login
2. List Products
3. Search
4. Add to Cart
5. View Cart
6. Price Calculation
7. Manage Activation Codes
8. View Purchase History
9. Logout

The basic functionalities add up to **20 marks**. An additional **5 marks** are allocated for extra credits. Please feel free to implement anything that your team think is interesting and useful.

Here are some ideas for extra credits:

1. List of Product Reviews (e.g. 3 stars, add/display comments)
2. Product Recommendation (e.g. show related products)

Other Requirements

1. Please work in the final-project team that you have been assigned to.
2. The use of Entity Framework, for database operations, is highly encouraged.
3. Use of third-party libraries is discouraged (but allowed). Writing your own code to experiment and learn is much preferred.
4. You will be assessed based on the following criteria:
   1. Number of features completed
   2. Clean code and sensible design
   3. Functional user-interface
   4. Video Demo of your app
5. The deliverable for this CA is a **single zipped file** that consists of:

a) Project source-code

b) Video presentation of your work (**try to keep it within 20 mins**)

1. The submission deadline is Wednesday (14 Oct), 9pm.
2. A LumiNUS submission folder will be created with an appropriate name like CA Submissions. There should only be 1 submission per team. Please name your files using your team name. For example, if you are Team 1, then your zipped file should be Team1.zip.