CS4753 Class Project – Milestone 2 (of 5)

Assigned: Wednesday, February 15, 2017
Due Date: Submit on collab by 11:59 pm on Wednesday, March 1, 2017
CAUTION: Collab closes promptly at 11:59 pm. Don't wait until the last minute to submit.
It is better to submit multiple times than to risk no submission.
Weight: 6% of final course grade

(Note: Milestone 3 will be worth 8% of the final course grade and will be due March 22. Milestone 4 will be worth 8% of your final course grade and will be due April 5. Milestone 5 is worth 4% of your final grade and will be due April 19. Another 5% of your final grade will depend upon your group's evaluation of your personal effort and contribution to your team.)

For the second milestone of the class project, you will be adding *back end* functionality to the web application that you created for milestone 1. Back end work is related to data management. This will include data collection, transfer, storage, and retrieval. While your application needs to look neat and professional, data provides the real functionality behind your business (assuming you settled on a valid eCommerce business concept). For this milestone, it is **highly encouraged** that you use PHP as your server-side language and mySQL as your database. While you are permitted to program in the language of your choice, the TAs will only support PHP/mySQL implementations. However you do it, ***you*** are responsible for knowing/learning PHP/mySQL. PHP/mySQL questions will appear on the exams.

Good PHP/mySQL resources can be found at the following links:

- http://www.tutorialspoint.com/php/
- http://php.net/manual/en/faq.general.php
- https://yashi.com/blog/best-resources-learning-php
- http://www.w3schools.com/php/default.asp
- http://www.w3schools.com/sql/default.asp

Grading Criteria

CRITERIA	POINTS
Create an additional page within your web application (front end) which includes a form that will allow users (customers) to enter their sign-up information. The form must be neat and cleanly designed. If a sign-up page looks too complicated or unappealing, a potential customer may abandon the process.	20
Install XAMPP on your machine (https://www.apachefriends.org/index.html). Within myPhpAdmin, create an appropriately named table (e.g., siteUsers, currentCustomers) that includes attributes that will be used to store your customer information. At a minimum, the table must have appropriate columns to store a user's name, email, address, city, state, and 5-digit zip code.	10

Implement front end form validation for all fields in the form on your Sign Up page. You must be checking for multiple eventualities, for example (1) that there is actually input in each of the text fields, (2) that the input includes the proper form of content for that specific text field (a name must not contain numbers, an email must contain the '@' symbol and at least one '.' symbol). These two checks are only examples; the TA's will be testing all inputs for correctness. Regular expressions and comparisons can be useful for these checks. If no input or if incorrect input has been identified, you must not submit the form to your backend PHP processing file. In the event of error, you must alert the user of the incorrect input. This could involve turning the borders of the text boxes containing incorrect input red, adding a new red error message to the sign-up screen, and/or having an alert box pop up detailing which fields contained incorrect information. Other methods are also acceptable.	25
Create a 'Sign Up' button at the bottom of your form that when clicked, calls some routine or program that will be used to insert the data in the form into your database table.	5
Connect to your myPhpAdmin database using server-side code (highly suggested to use PHP) to access the information that is sent to the server. For information about database access with PHP, refer to http://www.w3schools.com/php/php_mysql_connect.asp	5
Pass data inside the form fields on your web application page to your server-side processing code, passing the data as POST variables.	10
Insert the data passed by the user to the server into your database table that you created within myPhpAdmin. This will involve constructing an insertion query which you will use to query and operate on the database. If a user tries to sign up with a username that is already in the database, the information submitted by the user must not be put into the database, and the user should be alerted that he/she has already signed up for your service.	20
Update your previous pages' navigation bar to account for the addition of the new 'Sign Up' page. **Every** page within your application must be accessible from any other page. The 'Sign Up' page must have the identical navigation bar as your first two pages.	5
TOTAL **(MISTAKES/ASKED OF CORRECTIONS FROM MILESTONE 1 MUST BE FIXED OTHERWISE POINTS WILL BE DEDUCTED FROM OVERALL MILESTONE 2 SCORE)**	100

Class Policies:

Late Policy: You will incur a 10-point late penalty for each day late that you submit your project. After 5 late days, no submissions will be accepted. CAUTION: Collab closes promptly at 11:59 pm. Don't wait until the last minute to submit. It is better to submit multiple times than to risk no submission.

Submission Policy: You will be submitting a .zip file of your project code to Collab under the correct assignment page. Groups will be required to sign up for in-person demos for each assignment, in which they will display their current progress to the TAs. On your honor, you agree to demo the exact code you submitted on collab.

Demo Policy: If you miss your scheduled demo appointment (disregarding emergency situations and extreme circumstances) you will incur a 5-point penalty on your milestone grade.