C235 - Project: Process  
Project Summary Module Objectives  
Practicing Product Development Design a database with multiple tablesAs you worked through the learning activities for this week, you learned about the process that can be used to develop an application for a client. Now it is your turn to put the process to work. Imagine the Vice President of your organization has asked you to create a web application. She would like you to present to the Board next week showing how you plan to solve the particular problem and show a simple demo. Where will you start?

This project should be completed as five distinct steps:• Using the development process, write an app that meets the client’s needs.  
• Interview the client/instructor for a needs assessment of the client’s application.  
• Establish a clear and consistent problem-solving process.  
• Research the best solutions that are necessary to meet the client’s needs.  
• Write the application using the prototype process.  
1. Do a needs assessment.  
2. Do a problem analysis.  
3. Write proof-of-concept code.  
4. Prototype the Ul.  
5. Write the demo app.

This is an individual assignment and the work must be your own.Beware of folks who "need help" and might possibly copy and submit your code as their own...

File Management  
Inside the www folder of your AMPPS application, make a new folder named prjProcess. It will contain the following files when the project is complete:  
• A web page named index.php This will be the page the users will see.  
• A web page named presentation.php This will be your presentation to the Board of Directors outlining your solution.  
• A page with test code named proofOfConcept.php will demonstrate how you will solve specific software problems.  
• A web page named edit.php will be used by the administrators, managers, instructor to edit the data.  
• A program named dbfCreate.php or a text file named data.txt that will allow the instructor to run your program on a computer using localhost (without having to rekey in all the data....)  
• A sub-folder named graphic to hold all any images you use.  
• An external CSS file named style.css with your CSS code.

Specifications (50 points total possible)

Step 0. Think of a project  
Think of a simple app that will solve a problem similar to the hyperlink problem presented in this module. Here are some examples to act as a catalyst for your own app.  
1. A restaurant menu. The manager needs to be able to edit the data to change the menu items and prices whenever it is necessary.  
2. A university schedule. The administration office needs to be able to edit the data to change the schedule showing the course, time, place, and instructor for a set of classrooms.  
3. A work schedule that allows employees to click their names revealing the times they are scheduled to work that week. The manager needs to be able to edit the data whenever the schedule changes.  
4. A best seller list. The store manager needs to be able to update the data each week showing the best selling books from the previous week.  
5. A grocery coupon app The store manager needs to be able to update the products and savings on a weekly basis. If an item is on the list, the discount is automatically applied.  
You will be developing a display page, that the end-user will see, as well as an editing page that only the manager, director, administration, or your instructor will use to update the data.

Step 1. Needs Assessment (10 points)  
Create a web page named presentation.php containing at least five questions you would ask a client in order to complete this project. Then, acting as the client, answer these questions. As a guide, you can start with the five W’s (Who, What, When, Where, and Why). Display an HTML table showing at least four records of data to use for testing the app.  
• Design this page as a presentation for your client. (Your instructor is your client in this case.)  
• Keep UX (User experience) and readability in mind using headings, bullet points, bold and regular, and other HTML/CSS techniques. For example, display the questions using bold text and the answers as regular text.  
• Include your name, the title you are giving your project, and the current date at the top of the page. This information should display. (This information should also be included as part of the standard comment block at the top of your code.)

Step 2. Problem Analysis (10 points)  
At the bottom of presentation.php describe at least two ways you could approach this application as the developer. For example, you might suggest using a plain text file to store the data as one approach and a database as a second approach. Decide which is the best approach for your client and write a short paragraph highlighting your recommendation and why it is the best option for your client.  
You may find it helpful to answer the following questions:• How will the data be handled? What are some alternative formats for the data?  
• What will the Ul be like, from when the user clicks on a link to when information displays? Visualize this using a simple sketch or flowchart. (A sketch is worth a thousand words to every client!)  
• What can you do to make the information appealing and readable for your client? (Imagine that you are submitting a competing bid to get the job.) Include a menu on presentation.php that will have links to the following pages:  
• Presentation linking to this presentation.php file  
• Home Displaying the information for the user (index.php)  
• Edit The editing or management page (edit.php). Normally this would not be visible to the user, but this will make grading go smoother for the instructor.

Step 3. Proof-Of-Concept Code (10 points)  
Using your problem analysis, write proof-of-concept code demonstrating the code that will solve specific problems needed for the application to work. Name this file proofOfConcept.php.This code is to help you noodle around with very little overhead. The output from this code does not have to be pretty!  
• Determine where the sticky-wicket(s) (specific problem areas) are in the project specs. What techniques will you need to figure out to make sure these parts of the program work? For example, how will you read a text file? How will you write to a text file? Or, how would you read data from a database table? How would you write data to a table? The 'sticky wickets' will be based on the solution you proposed.  
• Keep the code as simple as possible. If you can solve the problem in a single line of code, that's excellent.  
• Do not spend time writing user input code. Simply set up a variable with test data and use it in your example code. Later, it will be easy to replace that variable with data actually collected from the user.  
• Focus on a specific problem without worrying about the other challenges. Being able to zoom in and focus on a single programming issue is key to being an effective programmer.  
As a programmer, you want to write several small, test programs each solving a specific problem. Once you have a piece working, add it to the finished application program. This will allow you to focus on one thing at a time, building the final app, one piece at a time. (Psychologically, this is very hard for new  
programmers to do. But, once you get in the practice of writing "side" programs, you will never go back to the old way of only working one large and forever expanding page of code.)

Step 4. Prototype the Ul - User Interface (5 points)  
Draw a prototype screen (wireframe) of the pages. You will have at least two screen layouts. One for the information being presented to the users, and one showing the edit screen allowing the manager or administration to update the data. You do not have to write the actual code at this point. Just show a visual picture of what the screens might look like.

• Use any drawing software you with to create this wireframe such as Google Docs Draw or draw.io. Or, use pencil and paper and scan in the image.  
• Include this image at the bottom of presentation.php• If you scan the image verify that you have formatted the image appropriately for display on the web page for size and clarity. Tiny, unreadable images or fuzzy images are not acceptable. On the other end, HUGE images are anti-UX and should be avoided. As a rule of thumb, images should be between  
500px and 1000px wide, with a proportionate height.

Step 5. Write the App (10 points)  
Write the final app. It should include the following:  
• A web page that the users will see, named index.php.• A web page used to edit the data named edit.php with a user-friendly interface that allows an office manager or administrator to update the content. This screen must include editable fields, a submit button, and a display of the data that automatically updates the display as changes are made.  
• All pages must have a clean Ul, using the same CSS code for all pages.  
• If you use a text file to store the changing data you need to include a copy named data.txt containing sample data.  
• If you use a database, you need to create a file named dbfCreate.php that the instructor can use to build and populate the database tables on a computer using localhost.  
• If you use a database, include the "server sniffer" code that will automatically use the correct userlD/password based on which server running the code.  
(See the FAQ for this project.)

5 points - Publish to the Server  
Publish your working code and database to your hosting service, uploading all the necessary files to the server.

Submitting your project  
1. Verify that you have all the files listed in the File Management section above.  
2. Zip up the prjProcess folder from your localhost and submit it to Blackboard.  
3. In the comment section include the URL to your site out on the server. (Please double check to make sure the URL opens the correct page before submitting your work.)  
4. This URL is for the instructor's review. You may also be posting the URL as part of your discussion posting which will allow the other students to view your website.  
Note: Publishing your work out on the server is not enough. You must submit your code and a valid URL to your remote site on Blackboard in order to receive a grade for this project.  
The instructor must be able to run your code on a computer running localhost.

Other Notes  
Points will be deducted for the following:  
• Missing a comment block at the top of each page of code.  
• Not following the naming conventions (code, file and folder names, form elements, data field names)  
• Using filenames other than those specified.  
• Spelling or grammar errors. Proof-read each other work. Use Grammarly.com  
• Inadequate commenting.  
• Poorly formatted code. Use tools such as Free Formatter or Dan's Tools - FITML formatter

Commenting  
Comments are required for each block of code. Add comments as notations to the instructor showing your knowledge and understanding of the code in  
your own words. Just having the code is not acceptable.  
Every file created by you must have a comment header block at the top of the page including the following:

• Name of the file and its purpose.  
• Course name and project name  
• Your name and email.  
• The date the file was written/created.  
• A place to document future revisions, (date, person, what was done)

Naming Conventions  
The following naming conventions must be followed when naming all folder, filename, id, and class names:  
• Always start a name using a lower case letter.  
• Do not use spaces or underscores  
• useCamelCaseForReadability. Acronyms can remain all caps, using a hyphen only when absolutely necessary.  
• Use singular names. For example, graphic not graphics.  
• Keep all FITML elements and filename extensions lower case (Watch out for photo editing software that tries to save the filename with the extension  
set to all caps.)  
Revised: 12-13-16