Eric Zorn

2/27/2018

ICT 4561

Professor Reyes

Week/Module 10

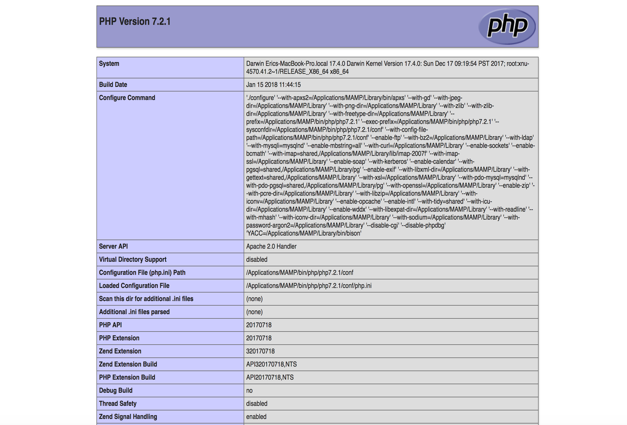
**Week/Module 10 Final Papers: Course Retrospective**

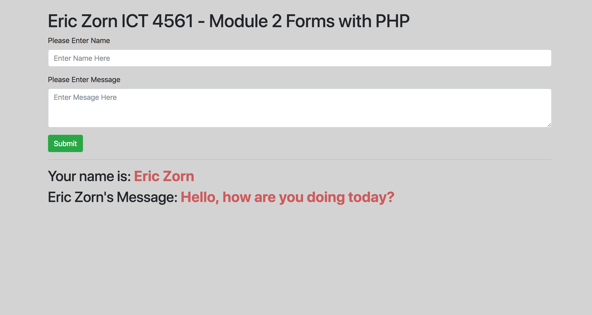
Throughout this course, ICT 4561, we have learned a lot about the PHP programming language. As many of you know at this point throughout the course, PHP stands for Hypertext Preprocessor and is a server-side programming language. Before taking this course, I had studied HTML/CSS and JavaScript. All three of these languages are known as client-side languages and are used to render information on the browser through the client’s machine and choice of internet browser. HTML or Hypertext Markup Language and CSS or Cascading Stylesheets, are really not known to be programming languages. Instead, as defined by their names, are markup languages and stylizing languages. As we know, HTML is used to define the structure of a web page, with its plethora of different elements and attributes. Examples of these would be different headings (h1, h2, etc.), paragraphs (p), forms (form), break tags and horizontal rules (br and hr), and many more. Once these elements are defined in the document, the designer and or developer would use CSS to define the styles of each of these web pages and their contained HTML elements. The stylizing may consist of colors, positioning, animation, widths and heights, media queries (used for responsive design), image galleries, and many more. CSS styling can also be used many times for different elements and web pages when the document is linked to an external directory. Therefore, elements across the entire site can inherit styling from that one document. JavaScript, which can also be linked to an external document and reused, is often referred to as a programing language. Any JavaScript is placed within the script elements in the HTML and or linked through the source attribute. JavaScript is also a cross-platform language, which means that it is compatible with most modern web browsers on any operating system. JavaScript, as mentioned above, is used as a client-side programming language primarily, with the exclusion of frameworks like Node.js. JavaScript is written to interact with the HTML elements in the DOM (Document Object Model). It can control pages to become more dynamic and respond to user events, such as any clicks, hovers, input, and navigation. JavaScript is also often used for making AJAX calls (Asynchronous JavaScript and XML/HTML) to the server. It allows JavaScript to access different web servers with HTTP protocol requests, such as GET and POST. Therefore, information from an API would be able to be utilized and rendered on the client’s browser. An example of this may be known as a weather app that is constantly refreshing data from an API and displaying the updated information in the application.

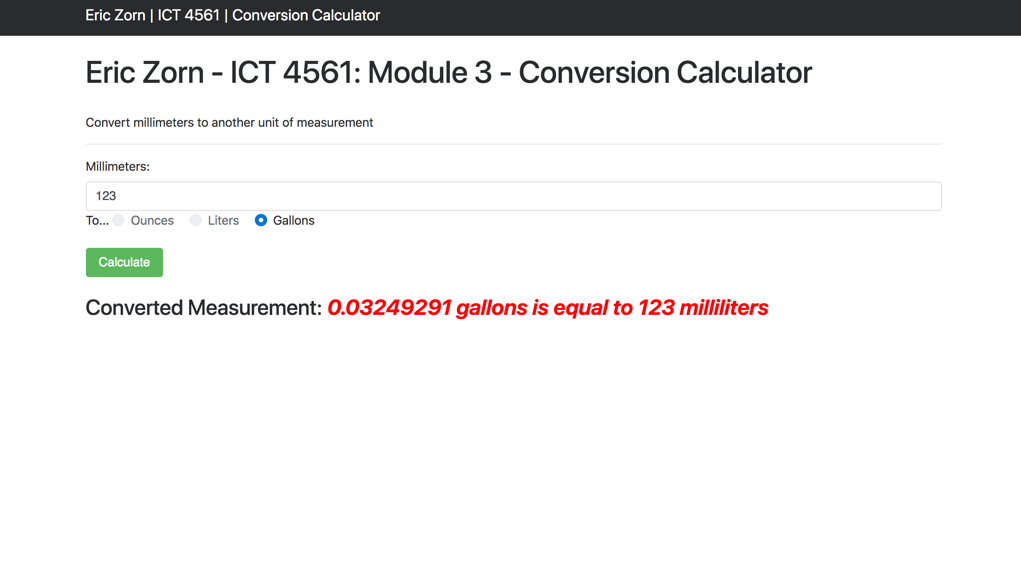
Now, with this basic knowledge of how client-side scripting and markup works, it is time to shift our focus to server-side scripting with PHP. PHP, much like JavaScript is a programming language and can manipulate the DOM with different elements and user input or processing. PHP on the contrary, will take in all of this information from the user and will process it on the server. None of the PHP code is going to be visible to the user on the client-side. PHP also is known as a cross-platform programming language which can be written and executed on most any modern-day operating system. PHP is also open-sourced, allowing for developers around the world to contribute to the language and update it over time. PHP can generate dynamic page content (HTML), create, open, read, write, delete, and close files on the server, collect and process form data, send and receive cookies. PHP can also add, delete, or modify existing data in your database, commonly MySQL. Lastly, PHP is often used to control user-access to the application and encrypt data that is being sent over the server.

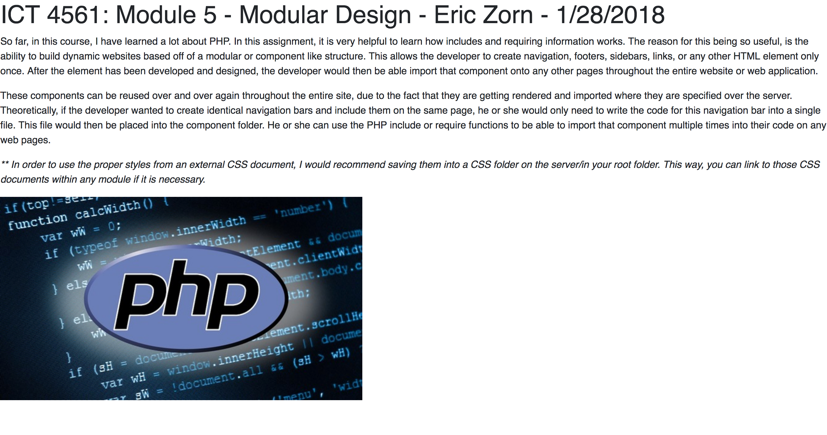
Throughout this course, we have learned and have used many, if not all of these features from developing our form messages, conversion calculator, and image uploading applications. We have learned how to submit and interact with form data that is being sent with the HTTP method of POST and controlling the action or location of which page will retrieve that data. We have learned how to process conditional logic (if/else/else if) and perform loops to iterate over a dataset (for and while). We also have learned how to create an includes folder to enable modular designed applications. This allows us as developers to require or include that component on many web pages within our document (commonly a navigation bar and footer). We also have dived into using Cookies and Sessions to be able to store information about the user’s input or browsers. This information can remain permanent or can expire and update after a specified time by the developer. We have also touched upon the concepts of functions and how to create reusable code blocks that will perform logic on the incoming parameters to those functions. Lastly, we have learned how to allow the user to upload files to a specific directory and process that information on the server. Once the files have been uploaded, we have learned how to store their information into our MySQL database and display those files based on the information that is queried from the database. Once the user has clicked the delete button, those files and their information in the database will be removed as well.

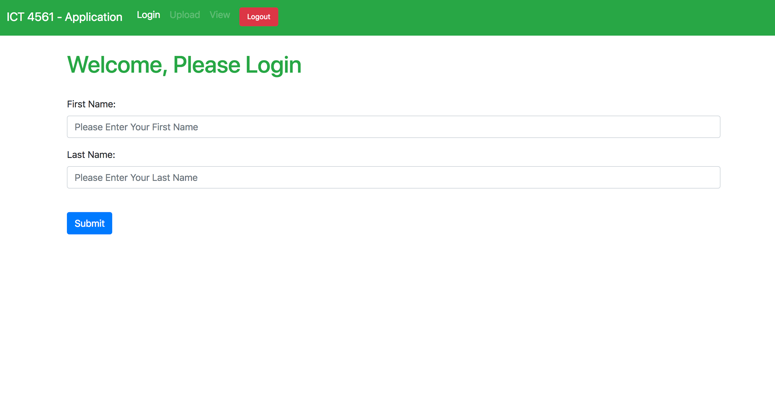
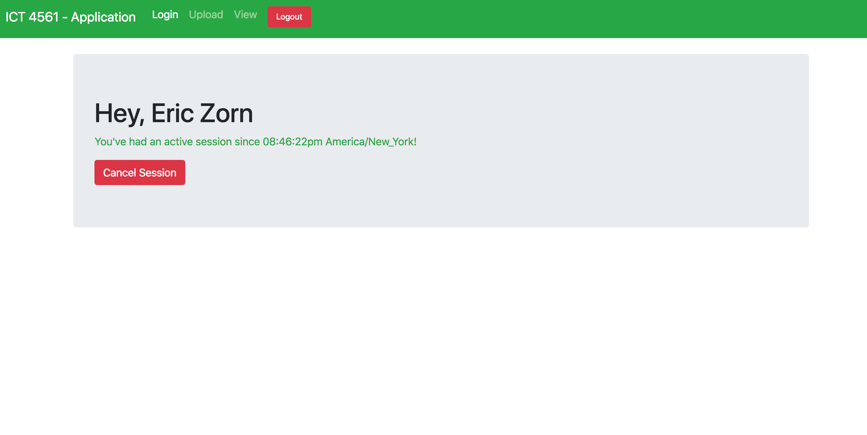
Each of these concepts is very powerful and has shown how we as a developer can change using only static web pages to become more dynamic and render content that is being processed on the server and database. The part of the logic that is being performed behind the scene, on the server, is known as the back-end. This consists of any processing logic and database manipulation through the server. When you combine creating a front-end (User Interface) and a back-end (Server and Database Logic), we are essentially becoming full-stack developers. This is where client-side languages like JavaScript can fall short. It is much harder for languages like JavaScript to be implemented to interact with a database and server without using a framework like Node.js.

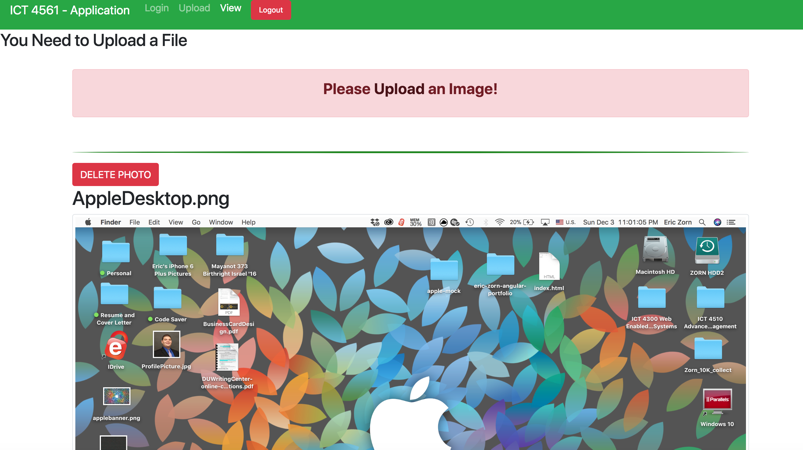
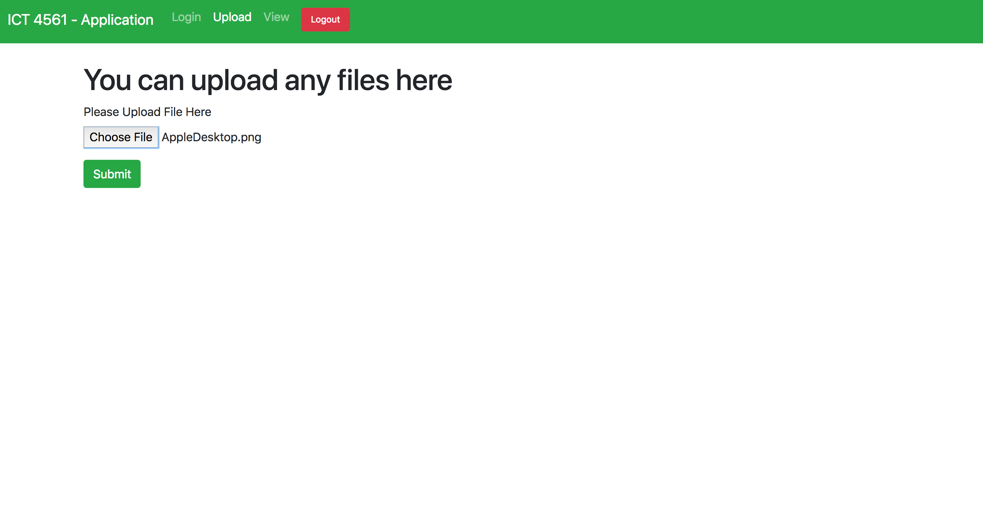
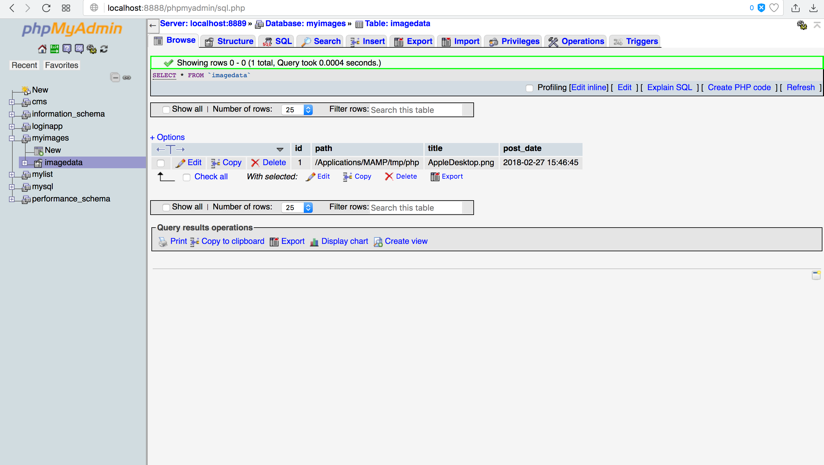
 The first project that we had worked on was outputting the information from the PHP configuration that was on our local system and server setup, through MAMP or XAMPP on our computer. This was a simple application/webpage that used the echo statement and the phpinfo() function that automatically generated the web page of all of the PHP language and configuration settings.

The second project that we have worked on in this course is sending data to the server, using an HTML form and outputting that form as a concatonated and stylized string to the browser. In this project, I had used the Bootstrap CSS framework for basic form and text area stylization. I also had made the form fields required in order for the form to submit to the browser and the server. When the form fields are filled and the user clicks the submit button, PHP will check to see if both form fields are set and then, if so, will export the user’s inputted name and message to the browser window as a concatonated message.

The third project that we have completed in this course is, a conversion calculator for different liquid measurements, such as ounces, liters, and gallons. Bootstrap has also been used in this project for basic user interface styles. The first user input in this assignment is the milliliters that the user would like to convert. Next, there are three different radio buttons that the user will have the choice of clicking to convert the milliliters into ounces, liters, or gallons. Once the user has filled out each of the form fields and has pressed submit, the information will be sent and processed over the server with PHP. PHP will capture this information and will perform different logical conditions to determine which conversion is desired by the user. This is done by checking with the isset() function for which radio button has been selected. At the beginning of the document, I had defined different functions that would take in the proper milliliters as the paramter. Depending on the met radio button condition, the proper function will be called to calculate the conversion and return the value. The milliliter amount will be passed as a function argument from the PHP superglobal array of $\_POST data. The server will then export a concatonated string within an em HTML element for the user to see the final conversion. If the conversion was susccesful, the output will be stylized as red. If there was not user input, the server and PHP will export that there is no measurement inputted.

The second to last project that we have completd is on how to properly use the includes folder and create files to export modular code that will be resuable. This assignment was fairly simple, but is very powerful and covers one of the most useful aspects of the PHP programming languages. We have disected the page heading and opening body tag into one module, as well as the footer and closing body tags into another module. Each of those modules were separate PHP files that consisted of the HTML code. We they used the include or require statements to import this information to the index.php file. Between those two includes, we have included basic paragraphs and a single image. All of this text and the image will be automatically generated and outputted between the opening and closing body tags, as long as they had been included and or required in the proper locations throughout the index.php document. When loading the webpage. PHP compiles and exports all of this information out as one single webpage that consits of the included modules.

The last two assignments combined into one another. The first part of the assignment was to create a Bootstrap stylized loginform and information output. When the user had inputted their first and last name to the login form and have pressed submit, a new HTTP session had been created. The HTTP session will store the user information like the user’s first and last name to the browser. The browser will then redirect the user to a different webpage that will display a greeting message to the user with their first and last name that has been retrieved from the web browser. Below the greeting is an output of the user’s session start time and time zone that they are currently in. Below the greeting and session begin information, there will be a logout button that the user can click. If the user clicks the logout button, the user will be redirected to the index/login page and the session/session data will be terminated.

The last and largest assignment that we have had in this course, is to create a full-stack application that has built upon the last project of the user logging in and storing their information. In this assignment, we have learned how to creata a database and database tables within the PHPMyAdmin web application. The database that has been created is called myimages and the table is named imagedata. Within the imagedata SQL table, there are four columns. The irst column is the ID in which will be the primary key and is set to auto increment with each new image data inputted. The second column is the path with a datatype of varchar (SQL datatype) and has a maximum of 255 characters. The third column is the title column, which is also a varchar datatype with 255charachters. The path column stands for the filepath of the image that is being uploaded and the title stands for filename of the image, excluding the path. The fourth and final column is the date posted column which will automatically populate when data is inputted with a timestamp of the data and time. This field had not been required, but I found this would be potentially useful for future querying within the database. I have included further styling with a navigation bar and disabled links with JavaScript, unless the user has logged in from the last assignment. Once logged in, the user will be directed to the same greeting page as the previous project, however, all navigation links will be accesible. The user can then click the upload link, which will display a Bootstrap form that will allow the user to upload an image. In order for a form to be sent and processed over the server, the form submission must have the encrypted type attribute. It will read enctype=”multipart/form-data”. When the form has been submitted with the file, a custom PHP function will check the filetype/extension, such as jpg, png, and gif. If the file is validated as an image, it will upload that file to the uploads directory with the proper file name from the $\_FILES superglobal and basename. If there is no file uploaded, both PHP and JavaScript will alert the user to upload a file before the form is submitted to the server. Once validated, the user will be redirected to the View URL which will display all uploaded images and delete buttons per image. As the view loads, the path and title of each image are stripped from the full file path and are uploaded into the database. The view will display the image based on a database query for the filename and file path and customized Bootstrap stylizing per image uploaded. If the user clicks the delete button, the database will update and will delete the uploaded image information, as well as the view will remove the image from the UI. Also, if there is no image being uploaded, a link and alert will inform the user to navigate back to the uploads section to properly upload an image. Now, this application is considered to be full-stack and has a fully dynamic UI with database integration and server logic from PHP.

For me personally, I had not had much experience with PHP prior to this course. In one of my past undergraduate courses, we had touched upon creating a local web-server to host our project and how to create include statements for modular code. However, we had not touched upon many of the built in PHP functions, conditional statements and or database integration that the PHP language supports. Therefore, this course has been very helpful and has shown how to take a simple website and make it a scalable and interactive web application. Of course, with a small-time frame of ten weeks, it is impossible to cover every aspect of the programming language and any frameworks. However, if I were to be able to add some new content to the course, I would definitely include using OOP (Object-Oriented Programming) and PHP. This would cover classes, methods, attributes, inheritance, and constructors. A class is a reusable block of code/object that when assigned to a variable, will create a new instance of that class. That variable will inherit all of the methods (functions) and attributes (variables) from that class that can be simply called continuously throughout the program. The developer would give the class a name, not a reserved word in PHP, and then would define all variables and methods within that class. The \_\_construct method or constructor, which is used in many different programming languages, is the function that will be called as soon as each instance of the class has been executed. The class instance may look something like $me = new Person(). The variable titled me, will create an instance of the person class and will inherit each of those available methods and attributes.

Many frameworks in PHP are based off of this methodology of being object-oriented. Therefore, if we had some more time, I would definitely have liked to go over the most popular framework in PHP. This is known as the Laravel framework. Laravel is designed to build scalable, full-stack web applications with speed and clean code. It follows the MVC or Model View Controller design pattern, in which the model controls database interaction, the View controls UI interaction and data being displayed, while the Controller will handle all incoming and outgoing business logic between the View and the Model. This is a common design pattern that is also being utilized by many other web frameworks in other programming languages as well. Some of these may be known as Angular and React JS.

I really have enjoyed taking this course this semester and learning about PHP. Over the past few years, as I have been learning about web design and development, I have become mostly comfortable with the front-end and have really wanted to understand full-stack. Throughout this course and the other course that I have taken this semester, Programming with Python, I have learned a lot about the back-end and the server. Through the Django framework in Python and PHP, I have learned about the vast amount of HTTP protocol and database integration, as well as URL routing and session/cookie storage that turn a website into a dynamic web application. This class has definitely opened my eyes to the language and an interest in pursuing a career as a PHP developer. I plan to learn much more about PHP and the different frameworks that are being used for creating applications on the market. I appreciate all of the discussion posts, research, readings, and projects that we have done in order to learn the fundamentals of the PHP language.

**Works Cited**

"HTML5 Elements." W3 Schools. Accessed February 27, 2018. <https://www.w3schools.com/html/html_elements.asp>.

"CSS Styling - Navbar." W3 Schools. Accessed February 27, 2018. https://www.w3schools.com/css/css\_navbar.asp.

Chapman, Stephen. “This Is What JavaScript Is Used For.” *ThoughtCo*, ThoughtCo, 24 July 2017, [www.thoughtco.com/what-is-javascript-used-for-2037679](http://www.thoughtco.com/what-is-javascript-used-for-2037679).

Mozilla Developer Network. “Introduction - JavaScript.” *MDN Web Docs*, MDN, [developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction](http://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction)

W3 Schools. “PHP5 Introduction.” *W3 Schools*, W3 Schools, [www.w3schools.com/php/php\_intro.asp](http://www.w3schools.com/php/php_intro.asp).

Code Academy. “Object-Oriented PHP.” *Codecademy*, Code Academy, [www.codecademy.com/en/courses/web-beginner-en-bH5s3/0/2](http://www.codecademy.com/en/courses/web-beginner-en-bH5s3/0/2).

Otwell, Taylor. “Love beautiful code? We do too.” *Laravel - The PHP Framework For Web Artisans*, Laravel, [laravel.com/.](http://laravel.com/)