CSC 317 Project Documentation
Fall 2019

Written by Bradley Justice

Bradley Justice, JingTao Zhang, XiaoQian Yang, Aimee Pavlik
CSC41301
https://github.com/pambalos/csc317-webapp

Write-Up

Project Overview

This project involved creating a fully hosted and crafted website using the LAMP/MAMP - *AMP stack, specifically Apache, MySSQL, and PHP. Our group decided to create a 'Gold For Code' website which would sell code. In the end, we successfully created a website that could offer these services. This project was a learning process all throughout, but it was a very useful and informative foray into the PHP language, and the easy with which it can be integrated with markup languages such as html.

Technical Overview

We managed to work through each of the different stages fully:

HTML:

At the HTML stage, we were really only able to get a very static and sort of older looking website going, and found ourselves very limited by the inbuilt HTML functions and capabilities. We did what we could, including mostly making the layout of the website, but it was very limited.

We were quite impressed with many of the features and styles that we saw demonstrated during that first round - some of them looked surprisingly responsive for fully html pages.

HTML & CSS:

During this stage, we were able to do a lot more to our website in terms of getting it to where we wanted it to be in regards to its aesthetics. With CSS we were able to actually specify all of the different properties we wanted to, in order to make it look exactly how we wanted it to.

In addition to the powerful layout and aesthetic features of incorporating CSS into our project, we were also able to actually simulate some responsiveness as well, by using the CSS state changes, such as Hover before and after, and some of the transform and webkit animations available.

HTML & CSS & JS:

This stage allowed us to make our website very actively dynamic and responsive, in a way that we simply could not quite do with the CSS alone. In addition to allowing us to make our individual pages responsive and dynamic, we were also able to *simulate a back end* by creating global variables that we treated like they were returned from a server.

Implementing Javascript really brought our pages to life, making them pop and making them respond to user input.

HTML & CSS & JS & PHP: & MYSQL:

This final stage was arguably the most difficult part. What we quickly came to realize is the fact that a lot of the data that we simulated via javascript in the previous stage was not a simple switch. Instead, we had to really do a lot in order to make the PHP work with the SQL. There was a significant amount of configuration and testing and sweat and tears that went into making that connection click, but we managed to figure it out early enough to really develop the majority of our services using php and sql.

One thing I will say though is that we were not fully able to transition all of our services into the php architecture, and would have definitely been a priority if we had actually had enough time to finish a good website, along with the final project.

Development Environment

I downloaded and installed the MAMP stack from Bitnami at https://bitnami.com/stack/mamp This allowed me to install and run a full Apache, MySQL and PHP stack outside of the virtual image. Periodically I would pull the changes from the git repository into the virtual image and test, and it seemed to work, so long as the settings under ‘common.php’ are set to whatever you have set up.

How to Build/Import your Project

As mentioned before, this project is built to run PHP on an Apache Web server. So long as you have php 5.4 or later, it should compile and run fine. You can simply clone the Github folder from https://github.com/pambalos/csc317-webapp into the php server, and be able to access the website.

Assumption Made

We made certain assumptions; largely in regards to security. We had very minimal security procedures and processes in place; things like Okta for login, salt pepper and hashing for passwords, input cleaning for protection against SQL injection - all of that was ignored for the sake of prudence and getting an a website done. We also assumed that this was the best stack for our project, and did not have a design review stage where we considered different technology stacks.

Implementation Discussion

Once we got used to PHP, we were surprised at the capabilities it demonstrated and the ease with which it could be made to work with html and javascript. If we had more time, I would have very much enjoyed to fully develop all of our services using PHP, as well as clean up the code a lot so there isnt so much repeated code.

Project Conclusion/Results

This project was fully completed in terms of understanding and completing some part of the project at each stage fully. Although we did not have time to debug every little thing, and fully rewrite every service we wanted to using php, we were still able to fully get a website that utilizes the entire stack.

Overall this was a very enjoyable project that really made us practice and get used to working with web technologies.