Capstone Final Writeup

SR Kanna

March 2017

1 Purpose

OSU's Energy Efficiency Center helps manufacturing and industrial companies reduce their energy footprint. This is accomplished by producing reports about energy trends and making recommendations based on them. The reports, projects, and other data are maintained in their internal website. The website has been developed by different programmers and as a result has become disorganized and difficult to update. In order to remedy these issues we will design a secure, user friendly website with good code practices for the Energy Efficiency Center. Furthermore, the website is not accessible from mobile devices which decreases productivity of workers who can't access the website while on the job site. We aim to create a website that is mobile accessible and friendly for increased accessibility.

2 Current Progress

Our intention is to host a user-friendly website which implements several key features and is connected to a database. We are about ten percent away from completely accomplishing our goal. We divided our portions among the features and the requests of the client. In particular, my job was to implement the login, security, and CSS. Even if we divided up our tasks for the purposes for the requirements document, we have all contributed to the all tasks for the project.

Features which the client has requested us to implement include log-in, homepage, managing and creating projects/task, and the employee page. We have implemented all of these pages. The features took us the first several weeks to implement, particularly because we did not finalize our database setup until a few weeks ago. Although the database was not not part of our explicit requirements, we needed to implement one in order to interact with the features. We have approximately five tables which can be efficiently queried by our features. There is a log in, employee information, gantt tasks, gantt projects, and task table.

The login took a few weeks to implement. I used an old html file as an outline, but connecting to our temporary database proved to be extremely difficult and time consuming. One source of confusion is that our original html used a config

file which had a different set-up than our current header file. Being unable to use the header file prevented sessions from storing and updating the database. Ultimately we found the php error, but since php doesn't report back errors, this took a long time to debug. In particular, we had to be careful because a subset of security is logging into the website. The website should only allow users with clearance to access sensitive data. This prevents malicious users without authorization from accessing the information and maintains the data's integrity. We achieved this goal by only allowing authorized users to create new users under the new employees tab. I was particularly concerned from a security perspective, becauswe login in websites should prevent against common attacks such as sql injections, session hijacking, network eavesdropping, cross site scripting, brute force attacks, and converting time channel attacks.

The vast majority of websites need to be protected against cross-site scripting. By eliminating XSS vulnerabilities, we have excluded around 50 percent of potential attacks to websites. Cross-site scripting executes javascript on the client's browser which can lead to malicious activity. Often the hacker tries to gain administrative access by stealing cookies or other stored personal information. Luckily, preventing cross-site scripting is fairly simple. We are still adding php sanitization in our input and output. ScanMyServer produces an extensive report and check against SQL injections, Cross Site Scripting, PHP code injections, Source disclosure, HTTP Header injection, Blind SQL injection etc. By adding an authorization script at the end of our index page, ScanMyServer sent us the report via email. The findings reported that there were no vulnerabilities. Sucuri is a malware and security scanner which tests for malware, website blacklisting, injected SPAM and defacements. The tool even provides preventative measures. It's compatible with WordPress, Joomla, Magento, Drupal, phpPP etc. The findings didn't find any traces of malware, website blacklisting, injected SPAM or defacements. It did detect a weak firewall which has a medium risk factor. SSL labs is used to scan SSL web servers. It analyzes expiry days, rating, cipher, ssl/tls, handshake simulation, protocols, BEAST etc. Open SSL rated our website an A-. Although most factors such as certificates and protocol support were exploratory, key exchange and cipher strengths were a bit weaker. However the risks are minimal. Quttera identifies malware and vulnerabilities by checking for malicious files, phishTanks, safe browsing and malware domain lists. Web inspector provides an analysis of potentional Blacklists, phising, malware, worms, backdoors, Trojans, suspicious frames and connections etc. Similar to the other tools, web inspector found zero issues.

The last feature I was in charge of was good human computer interaction or user interactions. Aesthetics and ease of use are essential parts of user interactions. I ended up choosing between a few CSS tools which allow for a responsive website (accessible and dynamic on all platforms). We spent the first few weeks debating until we realized the website had been maintained by several developers and lacked usability within the website for clients and programmers and the best tool for the task was Bootstrap. For example, the tools to maintain projects, tasks, and employees were not attractive, meeting the client's needs nor does it have continuity between them. By using the same design technology

to recreate the tools, we increased the continuity, attractiveness and ease of usability for the clients.

We have used a bootstrap theme which uses neutral colors such as grays and blacks for easy readability. The CSS ultimately affects how the user navigates and interacts with the website. For example, on the landing page, the user is asked to log in with their pre-existing credentials and is redirected to the homepage. On the homepage there is a side-navigation bar which allows the user to view their profile, add new employees, overview, events, about, services, contact etc. This navigation bar is collapsible for increased screen size on mobile devices.

3 Roadblocks

We initially spent several weeks asking our clients for database access and server files. Our client is in charge of the database for the EEC, but for unknown reasons was unable to send us a config file or ssh server username/password. This had a severe impact on our progress because we were not able to implement the php portion of our project until a week and half ago when we hosted our own database. Although unpractical, the client insists we don't use or model their database and use our own. We transparently told the clients that our php files and database may not interact with theirs, but they did not want us using their database. The compromise we reached since we needed a database for the alpha release was to implement our own and have the client transfer data if they liked our website model.

We also asked access to their server and hosting site, but were unable to get it. Since our client was not able to tell us what the ssh login was, we are hosting on our public html. This means we are not able to connect to their other features since we don't have access to their files, but the client was ok with this comprise. We initially had a lot of back and forth between us and the clients and tried multiple avenues to get access to their database and server. Our client is technical, but is younger than us, so their understanding of our needs may not have been fully understood. We will anticipated this and better communicated our needs and the consequences of not meeting them in the future.

4 Team Dynamic

We have had an excellent group dynamic, which means we have made an excellent development team. In particular, my job was to implement the login, security, and CSS. Even if we divided up our tasks for the purposes for the requirements document, we have all contributed to the all tasks for the project. We may not all work at the same time, but there is understanding the work will get done. Garrett has focused his attention on implementing the database, which took tremendous effort. James spent most of his time on the gannt chart. Most of my time was spent working on CSS, security, code readability, writing

etc. We have all had mostly equal work distributions, although I am need to finish implementing my part regarding security. We have all worked in different ways to help contribute to the project. It is sometimes hard to switch from one developer to another, so whomever implemented the php features was in charge of reusing the code for the rest of the features, because it was code they were familiar with.

5 Retrospective

Positives	Deltas	Actions
group dynamic		
complete features	polish features	work spring term
	database tables	use our database
	access to files	host in public html
	host in public html	client understands
		we don't have files
		difficulty to transfer data