Energy Effeciency Center Website: Winter Progress Report

Garrett Amidon, SR Kanna, James O'Neal

February 17, 2017

Abstract

OSUs Energy Efficiency Center help manufacturing and industrial companies increase their productivity and reduce their energy footprint by producing reports for energy and productivity recommendations. These reports, projects and funds are maintained in their website. The website has been developed and maintained by several programmers. As a result, the website has become disorganized, difficult to update and use. 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 while on the job site. With enough time, we would like to create a secure mobile app for the client which they are able to remotely access.

Contents

1 Purpose and Goals

OSUs 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 cant access the website while on the job site. We aim to create a website that is mobile accessible and friendly for increased accessibility.

2 Kanna

2.1 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 thirty 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 to a degree but are still adding to them for increased functionality and usability. The features took us the first several weeks to implement, particularly because we didnt 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 doesnt 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 datas 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, because 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 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.

2.2 Still To Do

We still need to make sure our website is mobile friendly as we continue to program. Furthermore, we can add more javascript, ajax and php to create more continuity. We need to continue testing security through common website attacks. We have set up our login and database, so we need to be careful to avoid sql injections. Our Gantt chart is able to create tasks and projects, but we need to be able to test it with client data and get our clients approval on our new model. We are currently hosting our database on one of our team members mysql database, but this cannot be a permanent solution. Although our client has given us permission to use our own database (we could not get access to theirs), we need to make sure our the transfer or information is relatively easy for the client.

Although log in is a subset of security, Im still working testing the security of the website. All software should be designed with security in mind. Common security risks include malicious SQL injections, forgetting to update software, XSS, error messages, server side validation, passwords, file uploads and SSL. There are some web security tools available to test a website for security. All of these need to be common attacks need to be tested against.

2.3 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 dont 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 dont 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.

3 GARRETT

3.1 Current Progress

The sections I was officially assigned to at the beginning were database management, employee records and clock in and out. Throughout this term though, it has been proven vital that we each help each other out where we can because are sections rely on each other. For instance, without the database, the login would be able to work correctly. At the start of this term, we all sat down and discussed what tables were needed and what indexes each table would have. This gave me a good starting point and made it so I only needed to make small changes throughout the development process. So as far as the database aspect is coming along, I feel as if this section is complete, other than a few changes here and there to make things work smoother. As stated above, helping each other out, where we could, made the development process go quicker and smoother so I spent a lot of my time helping out with CSS and making queries where needed. At first, we didnt really know what style to go with but after a bit of searching, we finally settled on a sidebar style that looks nice on mobile devices and is easy to navigate from one feature to another.

In regards to employee records, we have made some progress. We have the employee profile page and registration page done. The employee profile page took a few days to finish, but ultimately came out looking well. This page pulls user information from the employee information table in our database based on which user is logged in. Also, we made a registration page that only logged in users can access to maintain security. This registration page makes it so users can be added to both the login table and employee information table and maintain the same user id for future use. With this feature being added, it will make it much easier to pull information from the database when the rest of the employee records features are implemented.

3.2 Still To Do

Because I spent a lot of time helping in other areas of the website, I wasnt able to start the clock in and out feature. This is something I have spent a lot of time thinking about and designing, so I have a general idea of how I am going to implement it. Over the next week, I hope to have this feature complete or almost complete and be in the process of stress testing it. Once this feature is done, I will be able to add a recently clocked hours portion to the profile page, so user can see how they have allocated their time over the past week.

For the employee records, I need to add a page that allows users to look at other users profile pages. Again, this is something I have thought a lot about and have been designing. I should be able to implement this easily and should have it done by the end of this coming week. To do this, I will have a page that has a scroll bar and search bar and lets users search for a specific employee, querying results along the way. When an employee is clicked on, it will store their user id and will then send the current user to a new page similar to the profile page, but instead will list the chosen employees information. A few things I plan on adding to the profile page is the ability to upload a new profile photo and if a user has not uploaded a photo, provide a default profile picture until they do. Also, I plan on adding an edit profile option so users can change contact information where needed.

3.3 Roadblocks

The biggest roadblock we hit was trying to access the EECs server and databases. We asked multiple times and were not given access just until recently. Because of this, we discussed with Kevin and them that we will be developing on our own servers and using our own database. We will be making the appropriate changes to our requirements document to reflect this. Another issue we ran into was when we would have an error in our php or html and the web page would just say Server 500 error. This was extremely annoying because we would have to go through our code and comment things out until we would have a working page and slowly add new lines of code to try and find where the error was. Over time though, our coding abilities enhanced and being able to find small errors such as a missing semicolon or parenthesis has become easier to spot and has proven to be a great asset. Another small roadblock that we hit every now and again is when were all developing and 2 people make a small change to the same file and try to push to github. This produces a merge conflict, and at first, we did not know how to fix this and ultimately someone would lose their changes. After time and repeated exposure, we learned how to remedy this problem and no longer is that great of an issue for us.

3.4 Interesting Code

```
<?php
        if(isset($_SESSION['uid']) && !empty($_SESSION['uid'])){
echo 'Home';}
        else{
                echo 'Login';}?></a>
        </li><?php
        if(isset($_SESSION['uid']) && !empty($_SESSION['uid'])){
                echo '
                <li>
                        <a href="profile.php">Profile</a>
                </li>
                <li>
                        <a href="registration.php">Add New Employee</a>
                </li>
                <li>
                        <a href="gantt.php">Gantt</a>
                </li>
                <!--
                <li>
                        <a href="#">Projects</a>
                </li>
                <li>
                        <a href="#">Tasks</a>
                </li>
                <li>
                        <a href="#">My Hours</a>
                </li>
                <li>
                        <a href="#">Contacts</a>
                </li>
                -->' ;}
        else {
                echo "Please Log in";}?>
```

This code snippet above is particularly interesting because it upholds our sidebar security and checks to see if the user has signed in by checking the stored user id in the session. If the user id in the session is empty, the sidebar will only display a link that says Login and all the other sidebar options will not appear. If the user id is not empty, the Login button will change to a Home button and the rest of the sidebar options will appear.

```
if($result = $db->query("select * from employee_information where user_id = '$id'")){
                while($obj = $result->fetch_object()){
                        $first = htmlspecialchars($obj->first_name);
                        $last = htmlspecialchars($obj->last_name);
                        $user = htmlspecialchars($obj->user_name);
                        $phone = htmlspecialchars($obj->phone_number);
                        $email = htmlspecialchars($obj->email);
                        $major = htmlspecialchars($obj->major);
                        $current = htmlspecialchars($obj->current);
                        $hire_date = date_create(htmlspecialchars($obj->hire_date));
                        $end_date = date_create(htmlspecialchars($obj->end_date));
                        $week_hours = htmlspecialchars($obj->week_hours);
                        $image_data = $obj->pic;
                }
                $result->close();
} ?>
```

Above is another interesting code snippet from the profile page. This code queries through our database and grabs the appropriate user information by using the logged in users id. From there we use a function called htmlspecialchars to escape any potentially dangerous values that could have been inserted into the database.

4 James

4.1 Current Progress

Starting out this term my main focus was to get the Gantt chart feature operational. This feature is essential to the operation of the engineering efficiency center. It allows employees to get an overview of specific projects and tasks. It also allows a visual interface for updating data about projects and tasks such as the overall progress of each.

The Gantt chart had several requirements in order for it to be functional in a way the efficiency center could utilize it. First, it had to be able to push and pull data from a database without reloading the entire page. It also had to be easy to use and have a flexible design. Initially in the technical document I had selected google Gantt for these purposes. After realizing it was not suited for the job I looked at another one of my technologies. DHTMLX fit all the criteria for a Gantt chart. Currently we have a working prototype of the Gantt chart. It has customized time scales and updates changes directly to the database.

The Gantt chart is fully operational but it requires fine tuning. To accomplish this, I will be meeting with the client to show them a demo. I have done this once and updated the chart accordingly for the alpha release. The second meeting should reveal more usability items that they would like. One such item may be an undo and redo functionality. These items can be tweaked and added over time. Through documentation it will also be possible for future efficiency center employees to change and update this chart.

4.2 Still To Do

A feature that still needs to be fully implemented is a way to track and manage employee hours. Currently it is unclear how exactly the efficiency center tracks this. Furthermore, it is unclear how they would like the new system to work. A table of some sort will be implemented which can pull recorded employee hours from the database. This will allow managers to see who has been working on specific tasks at different times. This will be my main focus throughout the remaining weeks.

Another issue is the creation of tasks and projects. While it is possible and simple to create new tasks and projects in the Gantt chart, it may not be the primary way they are created. We will need to meet with the client again to figure out exactly what information each project and task has associated with it. This information can then be used to create a submission form. Once the project or task information has been submitted to the database it can be viewed and updated on a relevant page. One way this may be accomplished is by using the Gantt charts task ID field. This can be used to pull up relevant data about certain tasks and link the Gantt chart to the more in depth information.

Documentation and software coding standards are a large part of this project. One of the biggest issues with the centers current site is the lack of clear instructions. It will be my job to make sure all the info and instructions regarding the sites pages is present and accessible. I will be working on this throughout the project.

4.3 Roadblocks

One of the largest roadblocks we have encountered is the layout of the clients database. Using our own database has worked well for creating a working site. However, we are unsure about all the data fields which are contained in the efficiency centers database. We have snapshots of some of the database tables but we would like to gather more information. Ultimately the user has told us that they would like to keep the data private so that it is not visible at an expo demo.

DHTMLX uses a specific table set up inside the database. There are SQL commands for inserting this setup but it is unclear how this will be integrated to fit the necessary data fields in the clients database. It is possible that the Gantt chart will only be used to view overall project information and not include the specific details of each task. This would mean that the Gantt chart would give a view of project time tables but more complex data would be viewed elsewhere.

4.4 Interesting Code

In order to improve the Gantt chart it was necessary to edit the time scales. Originally the time scale was a fixed feature using only days. It was apparent that there would need to be more flexibility when viewing the different projects and tasks. The time scale determines how close of a view you have when viewing a task. The updated implementation allows for a year, month, week, and day view. This allows users the ability to select the appropriate scale for what they are doing.

```
<script type="text/javascript">
function setScaleConfig(value){
        switch (value) {
        case "1":
                gantt.config.scale_unit = "day";
                gantt.config.step = 1;
                gantt.config.date_scale = "%d";
                gantt.config.subscales = [
                        {unit:"day", step:1, date:"%D" }
                ];
                gantt.config.scale_height = 50;
                gantt.templates.date_scale = null;
                break;
        case "2":
                var weekScaleTemplate = function(date){
                var dateToStr = gantt.date.date_to_str("%d");
                var endDate = gantt.date.add(gantt.date.add(date, 1, "week"), -1, "day");
                return dateToStr(date) + " - " + dateToStr(endDate);
                };

                gantt.config.scale_unit = "week";
                gantt.config.step = 1;
                gantt.templates.date_scale = weekScaleTemplate;
                gantt.config.subscales = [
                        {unit:"month", step:1, date:"%F %y" }
                ];
                gantt.config.scale_height = 50;
                break;
        case "3":
                gantt.config.scale_unit = "month";
                //gantt.config.step = 1;
                gantt.config.date_scale = "%F %y";
                gantt.config.scale_height = 50;
                gantt.templates.date_scale = null;
                break;
        case "4":
                gantt.config.scale_unit = "year";
                gantt.config.step = 1;
                gantt.config.date_scale = "%Y";
                gantt.config.min_column_width = 50;

                gantt.config.scale_height = 50;
                gantt.templates.date_scale = null;
                break;
        }
}
```

The code above uses radio buttons to determine which time scale to use. In each case there is a primary scale and also a subscale. The exception to this is in the year view. In this view there is no subscale because this view is used to see the largest picture possible. When adding in a subscale all values in that scale must fit below the desired year. This effectively zooms in making it hard to see a full year in one screen. The other views each have a subscale to help define the view. For example, in the week view, the months are also visible.

Something interesting I learned about this code came about when a subscale was not defined. After switching from an option that had a subscale defined to one that did not, the old subscale was still in use. To remedy this problem, an empty set subscale had to be defined. Another interesting part of this code is found in case number two, the week scale. A function had to be created to define what a week was. It takes every seven days and breaks it up into a scale. The variable holding this scale is then set as the primary time scale.