



# HOME WORK HOTTIES

## A BUDDING BUSINESS

### THAT FITS YOUR STUDENT

### NEEDS

[By Akane Simpson & Michael Hendrix](#)

#### About the Project

This project was planned originally to help a work colleague, good friend, an excellent student and student entrepreneur. As a budding business she was able to see the need for students who have difficulties in completing assignments on time while managing their social and work life. She has experience in running a team and excels at

## Table of Contents

<b>SECTION 1 - INTRODUCTION</b> .....	3
<b>Background Information</b> .....	3
<b>Problem</b> .....	4
<b>A Possible Solution</b> .....	4
<b>Project Timeline – Phase 1</b> .....	5
<b>SECTION 2 – EXPLORING THE PROJECT</b> .....	6
<b>Requirements</b> .....	6
<b>Table 1 – System Requirements for Website</b> .....	6
<b>Table 2 – Client Requirements for Website</b> .....	7
<b>Design</b> .....	8
<b>DOM of Website Web Page</b> .....	8
<b>Customer Website Site Map</b> .....	9
<b>Basic Webpage Model</b> .....	10
<b>Normal User-System Interaction</b> .....	11
<b>Flow and Processing of Data in the Project</b> .....	12
<b>SQL Database Schema Design</b> .....	13
<b>Gathering the Tools</b> .....	14
<b>Major Components proposed to be used in Project</b> .....	14
<b>SECTION 3 – CREATING THE PROJECT</b> .....	15
<b>Analysis</b> .....	15
<b>Comparing Discovered Tools</b> .....	15
<b>Major Tools</b> .....	15
<b>Minor Tools</b> .....	17
<b>Software Testing &amp; Implementation</b> .....	18
<b>November 24, 2016</b> .....	18
<b>November 25, 2016</b> .....	19
<b>November 26, 2016</b> .....	19
<b>November 27, 2016</b> .....	19
<b>References</b> .....	20

## SECTION 1 - INTRODUCTION

### Background Information

Home Work Hotties is a new student oriented business recently developed by a Grambling State University student. Their vision is being able to help students to complete their various assignments and tests on time which leaves them time to build themselves up and give more free time for the busiest and studious student who just needs a break.

The company was established in the summer of 2016

Home Work Hotties has become a payed service where students are billed depending on various factors such as:

- The subject area
- The type of questions such essay type, short answer or discussion type questions
- the number of questions to be completed
- The difficulty of these questions such that if they fall within freshman, sophomore, junior or senior level work
- Tutoring on most available subject
- Resume building
- And several more factors.

At the moment the company can only complete assignments from courses such as English, Sociology, Mathematics and Computer Information Systems. There are future plans of expanding the range of subjects that the business covers, which may include more subject areas being covered. The range of clients so far includes Louisiana to Texas and Florida to name a few.

Clients currently submit their work orders through text messages and screenshots of their assignments. After an evaluation of the assignments a price is generated by hand and an invoice is manually created and sent to the client's email.

When a work order has been created either the CEO or the CFO will have to post the order in a *GroupMe* chat as an event that any of the employees can 'attend' which in short gives the employee the responsibility to complete the task.

Once the employee has accepted the task they will be contacted by the CFO and given the required details to complete the assignment. Once completed a screenshot is required to be taken and sent to either CEO or CFO as a confirmation of completion.

For those who are interested in joining this business one would need to contact an associative of the company and asked to receive an application. After completing the application one would send this to the CEO and await approval.

## Problem

The current system of having the client directly messaging either the CEO or the CFO is inefficient. Along with having to manually create an invoice and sending that invoice to the client both CEO and CFO would need to instruct the client on how to pay for the service through downloading the *SquareCash* app. Using this method as well as a possibility of an influx of work orders can cause their current system to break.

This does not include the relatively small work force that are both on and off campus that will have to fulfill these orders on time. If this were to happen then it may as well sink the future and credibility of this business before it can even blossom.

Other issues that can be foreseen is the lack of a security system that forces the CEO to interact directly with a client. Having the head of a business take customer orders is not necessarily a bad thing, however, the interactions between client and employee has various different outcomes that may contribute to a poor customer experience.

Another issue that arises is the lack of a portal for employees to access available work orders and easily get the information to complete the task. As an employee one needs to be in constant contact with their boss and the use of a messaging app such as *GroupMe* is not the best methods of fulfilling this action.

In summary, most of these issues arise from the lack of a proper platform that can adequately take all incoming and outgoing traffic and request. Since the business owner in question does not have any real experience in online operations or building an online storefront. This absence of an online platform can stifle the growth of the company and result in the loss of possible income.

## A Possible Solution

Originally, the project was meant to be focused on creating a web application to allow meteorologists to enter weather data into a form. Instead, this project will be focused on web development through the use of various tools to accomplish this task. With this change in development the project's development team has been tasked with developing for a small business.

A solution to this is to develop a website that would act as their storefront for new possible customers. The website should also have an employee only site where employees are able to carry out their assignments easily. This website main goal should be to replace the need for the CEO or the CFO to directly interact with customers every day and allow a system to:

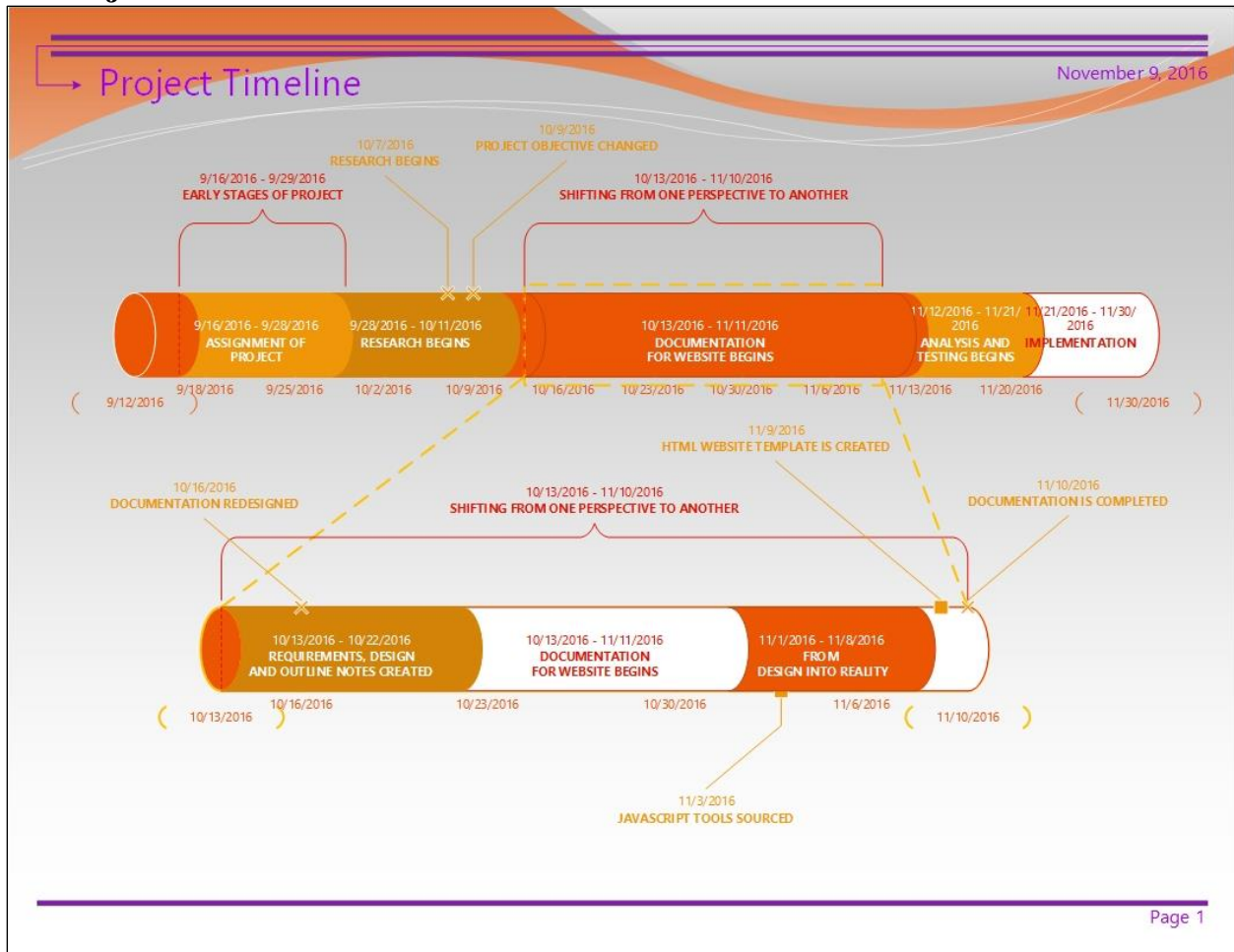
- host a semi-interactive website for users,
- allow a customer to create an account for them to store transaction details and invoices,
- generate transaction details and invoices for both ends of the website,
- accept requests generated by customer orders and create work orders for employees,
- store these requests on a database,
- send employees these work orders on an employee only site,
- have administration have access to all employee work progress,

- having employees able to complete and send confirmations that the order was received,
- have users/customers schedule dates for tutoring (if available or requested for),
- and secure customer information.

However, due to a predetermined time limit the project will not be able to be completed based on the scope of the development. Therefore, the project will be broken up into several phases, namely three.

The first phase will be based on the development of the website and eventually implementing it. The development cycle should be focused on making a basic, low functional website that can later be edited for visual effects. The second phase of the project will be to improve on the scripts used for the website. Phase 2 aim is to also implement style sheets to the project to improve user interface so as to create a longer lasting and attractive store front for customers. Phase 3 is proposed to be a long term phase where the development team will occasionally review the source code of the project and make the necessary edits and fine tuning as well as help in troubleshooting.

### Project Timeline – Phase 1



This project timeline is divided in five parts. The first phase was to gather the research material web development, namely, displaying a form for a user to input their reading into a database. The

second phase then researched the ways in which meteorologists recorded their information and how it can be displayed later. However, the members of the team did not enjoy the idea of making a simple project such as the original assignment. Therefore, one of the members, Akane Simpson, received an offer to create a website for the business, Homework Hotties.

This change in project aim forced the team to restart their project with an even more limited time. After gathering the necessary requirements, the third phase was based on a more detailed documentation period while preparing for the fourth phase. The fourth phase should focus on the testing of the methods created and making adjustments as they go. After testing the project should be able to develop a working alpha for demonstration purposes. If the alpha is successful then it is then hosted on a third party server and can be accessed from the public domain.

If the project is successful then Phase 2 would begin where each function that is implemented is tested to go with the yet to be conceived new format. Phase 2 also aims to upgrade the proposed website to improve user interface. No date has been set for phase 2 as yet. Phase 3 focuses on system maintenance.

## SECTION 2 – EXPLORING THE PROJECT

### Requirements

As a developing website there is a need for certain requirements to allow the full and complete experience for both possible customers and the client. In such a case some of these requirements may be expanded upon for later development.

Due to the previously mentioned time constraints the goal is to complete the basic structure of the website so as to later improve it. During this time the website should be able to carry out these system requirements with several functioning requirements to be wary of. The user requirements will have secondary priority as they can be reached at a later time during the projects.

These system requirements are split into two sections, one section for front end the other for back-end. Some of these system requirements are as follows:

**Table 1 – System Requirements for Website**

<b>Functional Requirements</b>
1. Saving a monthly backup of the website on a separate database
2. Sorting completed work orders every second week for the month and archiving them for accounting reference
3. Protection of the user's information from corruption and security breaches during process of saving data to the database through the server.
4. Updating search parameters for every addition of another webpage to the website
5. Have JavaScript enabled on browser to enable interactive features of website

6. An online form that can be used to input information that acts as customer's work assignment. This will be linked to their account.
7. Search engine can be used to find specific webpages from directory without searching manually. If there are several webpages with the same name, then a list is displayed that shows those with the keyword used in the search.
<b>Non-Functional Requirements</b>
1. No more than one user can be logged on a single system at the same time to prevent multiple work orders being made at the same time.
2. Website layout should be adaptive to different systems such as computer, tablet or smart phone
3. Administration will have access over both customer (with permission) and employee accounts if there are problems that requires direct maintenance
4. Browser should support latest version of JavaScript and be updated when a new version is released (up to a certain version that can support the functions used on the website)
5. System should be able to track progress of assignments as well as track time before assignment due date
6. Monthly server maintenance and database backup done by a technician
7. Removal of expired or broken webpages

The client requirements are more straightforward and have less priority but are still intended to be fulfilled at a later point in the form of an expansion. These requirements can be used as a guideline on designing an easy to understand website to use. These requirements are listed below

**Table 2 – Client Requirements for Website**

<b>Functional Requirements</b>
1. Be idiot proof
2. Allow tracking of work order progress
3. Provide information on how submitting works
4. Contact information available on home page
5. Home page should be easy to understand and state what the website is about
6. Separate portals/links for customer log in and employee log in
7. User can access website on smart phone
8. General users cannot access employee portal without identification
9. User name and password required for services

At this time there are no non-functional requirements. There is room for improvements this section can be updated at a later time.<sup>1</sup>

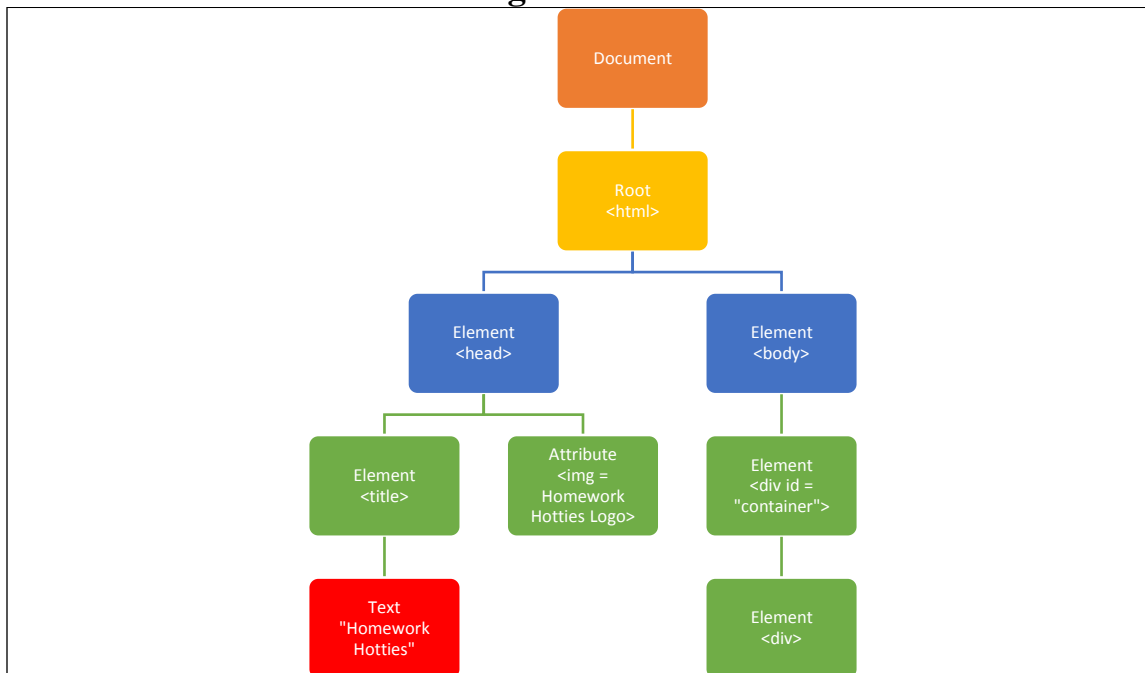
<sup>1</sup> Requirements gathered are from meetings performed on October 17, 20 and 23.

## Design

The website design for this project will use **HTML** and **JavaScript** to create a basic interface for users. One of the main features of this website is to create a template that can be used to quickly display information without the need to designing repeated copies of the same webpage by hand. Instead, using the principal of object oriented programming, the development team uses premade objects to create a continuous amount of webpages when new information is generated and needs to be displayed. With the inclusion of so many components there is a need to create a schema that maps how the website will be structured and the various webpages.

The website is planned to be constantly expanding its size so as to accommodate the addition of new webpages the database will also have to be able to expand so as to fit the size of the data being transferred. This project requires several tools and customization that will take time to implement effectively. The goal of creating a functional alpha needs to take priority by removing the 'fluff' and focusing on the basic requirements of the project.

### DOM of Website Web Page



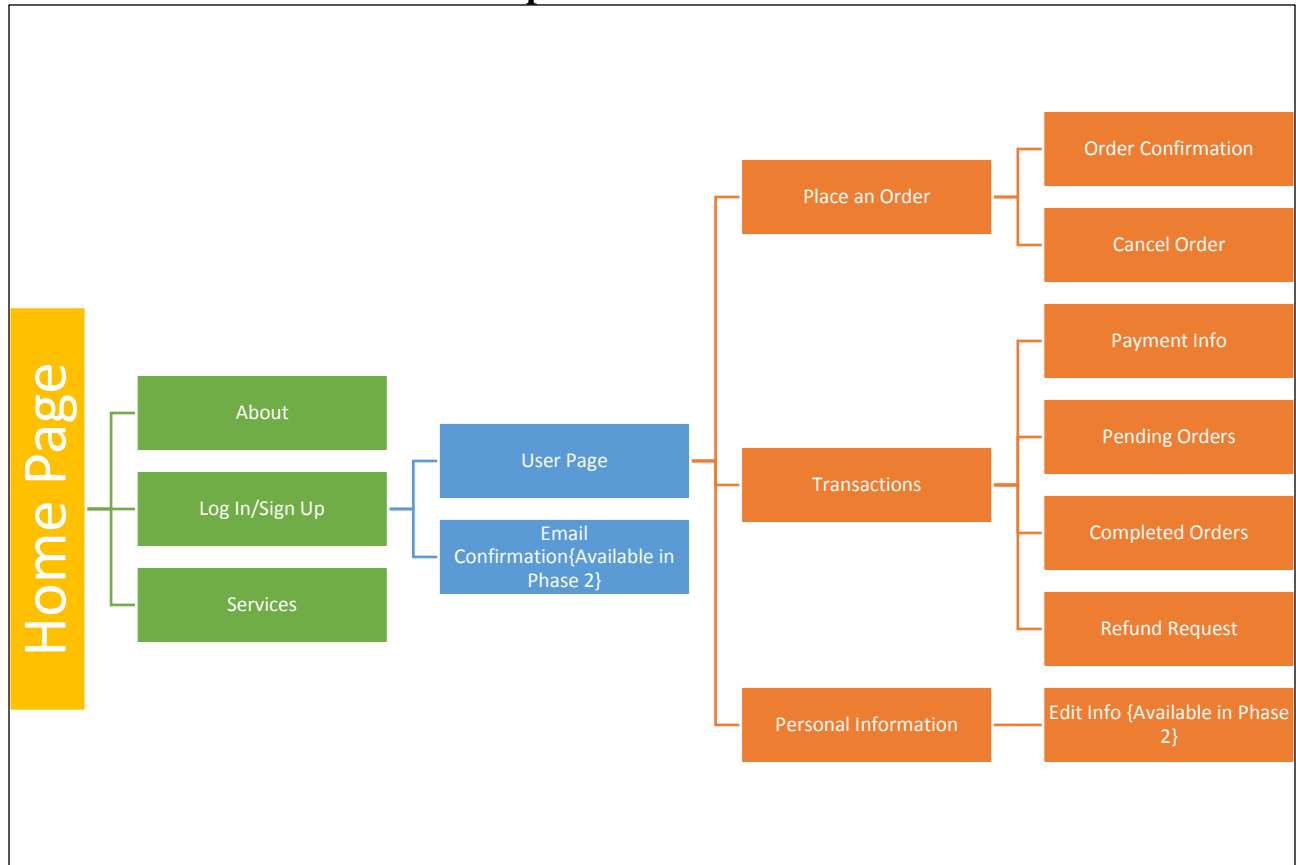
The 'skeleton' or **Document Object Model (DOM)** design of a website is similar to a tree diagram with parent nodes containing several child nodes. The **DOM** consists of the elements/tags in **HTML** that structure each webpage. The **DOM** also affects the structure of the JavaScript functions that are included in the project with some of these **JS** files being made externally. This concept can be applied to the project through the use of a 'skeleton'. This 'skeleton' is the basic look of a webpage and has no outstanding feature to it. Therefore, this 'skeleton' can be used to create as many pages as needed.



The 'skeleton' will only have the basic functions in the forms of buttons or tiles that can link to other pages when created. In later phases the skeleton will be updated to create a better looking interface.

Due to the size of websites it is common for developers to include or create a search engine so as users can easily search for specific information on a topic. When a page is created the page link should be added to a directory for the search engine to explore. This search engine could also be used for the employee side of the website when it is retrieving SQL table values and displaying them in a webpage.

### Customer Website Site Map



The Customer website site shows the linking between the various webpages that will be included in the customer section of the project. In addition, some of these webpages will be created in Phase 2 and the focus will be fulfilling most of the requirements. The Home Page has three options that you can choose from. The About option lets the user learn vital information about the website. The services option lets the user see what the website offers and what it can do for them. It also includes a Log In/Sign Up option which brings the user to the User Page and the Email Confirmation which is available in Phase 2. The User Page then asks the user to place an order. Once the order is placed it gives the user a order confirmation and also gives them the option to cancel the order. The User Page also lets the user see their transactions. This shows the payment info, pending order, completed orders, and the option to request a refund. The last part of the User page is the personal information. It allows the user to edit their personal information.

## Basic Webpage Model

# Homework Hotties

Header [Contains Logo and Text]

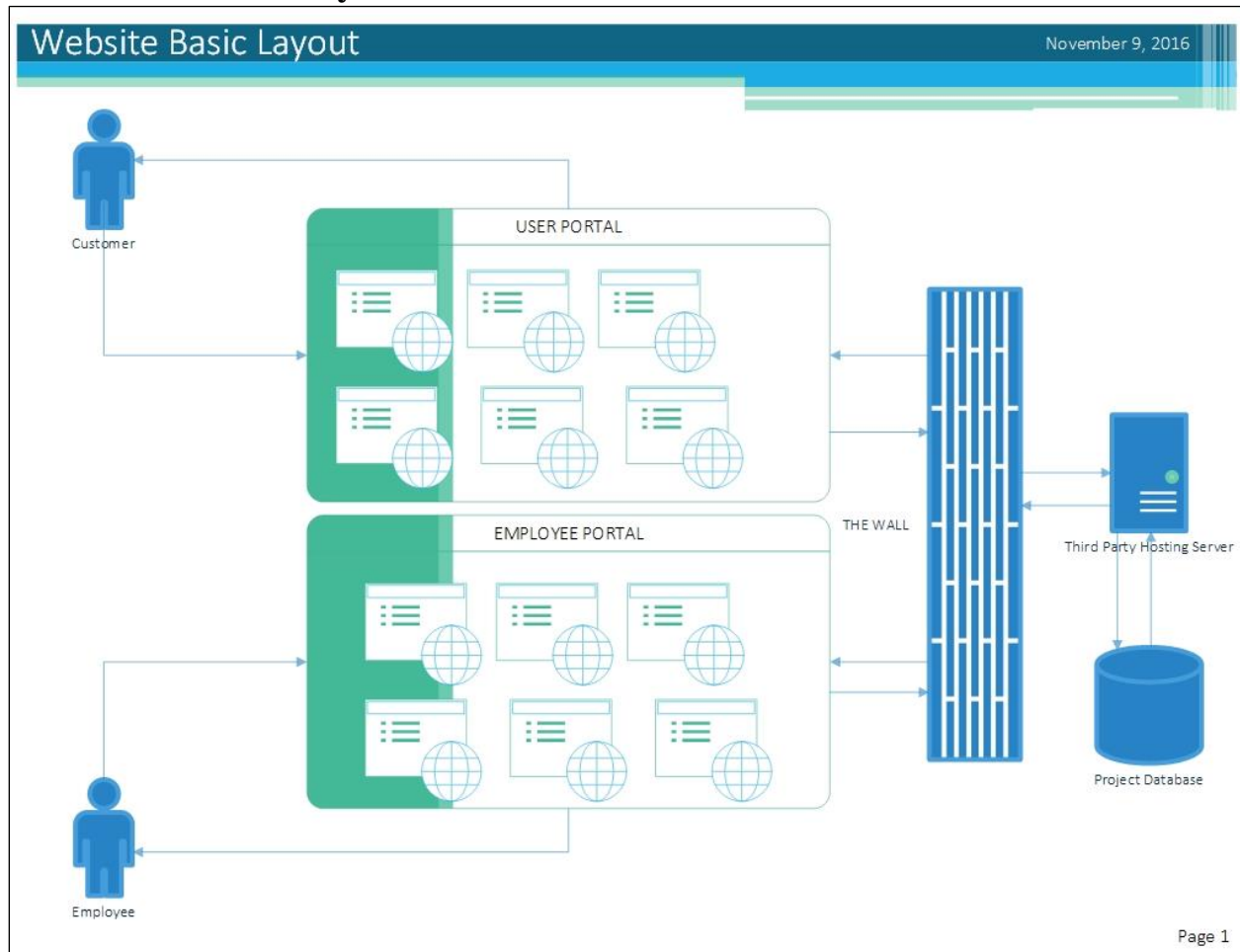
Navigation [Contains Home, About, Services and Log In/Sign Up Buttons]

Section [Section is used when referring to text and possible images]

Div [Div is used to contain the various elements and attributes used in the website body]

Footer [Contains Contact, Copyright, and Terms of Use Links]

## Normal User-System Interaction

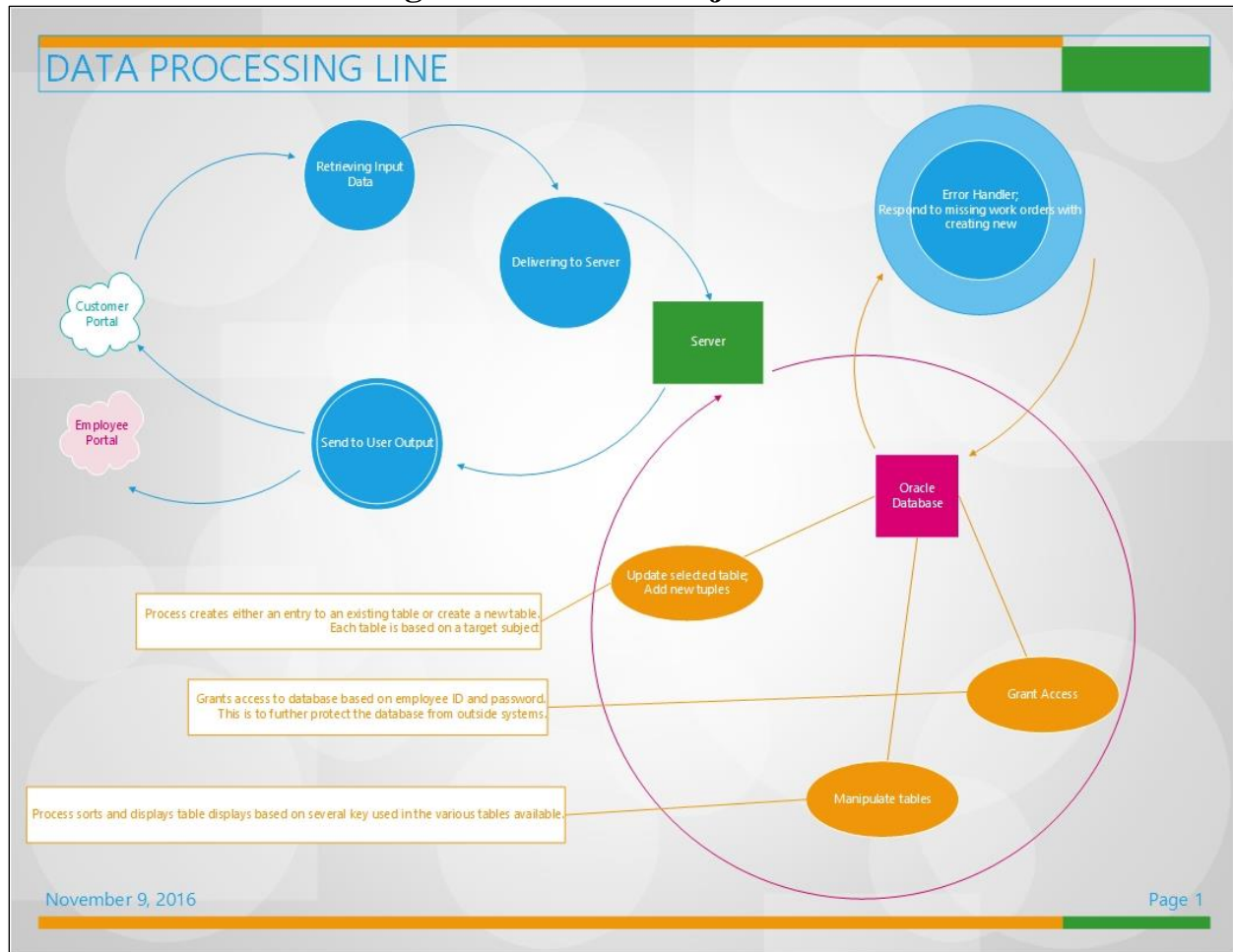


As shown in the above diagram the website will contain two portals, one for customer services while the other would be the employee portal that contains secured information. The customer portion will contain their basic webpages that will create and send these work orders to a remote server that is hosted by a third party vendor. These work orders are created from a form for submission of assignments.

This remote server will contain the project's database that can be accessed through administrative or authorized user rights. The next step is to return requested work orders to the employee portal where these results can be manipulated based on their subject and type of assignment. The employee portal will feature an interface a search feature to access the database so as to view the various tables that contain the customers' assignments. The assignment should be displayed in the body of the website with all the information needed for an employee to make a decision whether to accept an assignment or not.

A more detailed diagram can be used to show how the flow of data is suggested to be moved from process to process. These processes are based on the proposed work order submission where customers will fill a form stating the nature of their assignment. Using keywords from the form completion the assignment is placed in a table on the server that fits the description.

## Flow and Processing of Data in the Project

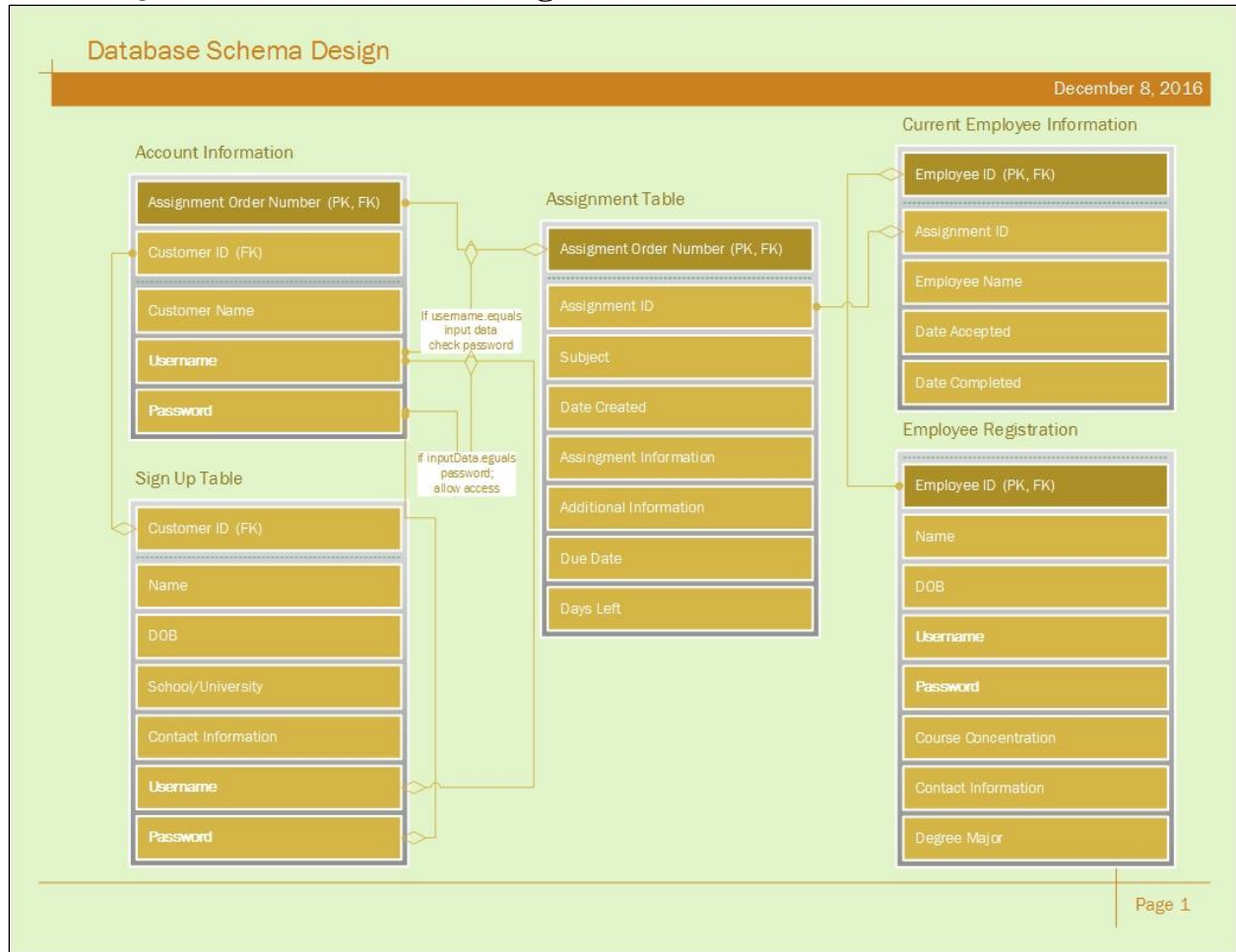


The diagram expands upon the Diagram 2 and lists key points for the processes behind mainly the storing and retrieving of data from the database. The three listed processes as it relates to the server process cycle is key to the distribution of orders to employees.

The process to the top right of the diagram called the Error Handler, is to focus on checking for error in both the information sent and received as well as the format this data is stored. When information is sent to the database through the server the 'Error Handler' process checks to see the legitimacy of the data from the previous process. If there are errors in the data or there is not a match, then 'Error Handler' will send the required signals to the database to either store the data in a different table than the intended one or to create a new table to store the data.

The other two processes should be used by the employee side of the website. For example, the 'Manipulate tables' process is used by the user to search for keywords or attributes in a selected table. The selected table is based on the employee's preference and key areas of study. The other process, 'Grant Access' gives permission to the user to access the database. This access is restricted and can only be issued to those who log on to the employee portal. The last process, Update selected/Add new table' updates a selected table by adding or removing rows from the table. Add new table creates a table which will follow the database schema.

## SQL Database Schema Design



The diagram above is a detailed version to what is expected of the database. For example, the Sign Up Table is used during the Log In/Sign Up process to create an account that will be used for authentication of a user. If the account information provided at sign up is different than the information in the account information, then that instance of user log in is denied. For security purposes the website will only allow a user to try logging in only four times before they are shut out of the system for a certain amount of time.

The Assignment Table is the main part of the database as it will contain the work order/assignment information of a user. This table can then be accessed by an administrator or an employee. The table can also be searched using its primary key. The primary key, 'Assignment Order Number' is a generated numerical sequence of 10 numbers that is created when a user submits a work order. This work order must fill certain fields in the correct format and includes the 'Assignment Order Number'.

The Employee Registration table is similar to the Sign Up table where a user will be prompted to fill out a form to complete the registration process. The form contains some basic fields where

some require the user input, such as username and password, to allow the user to submit their data to the website. This data will then be sent to a database and stored in the appropriate fields.

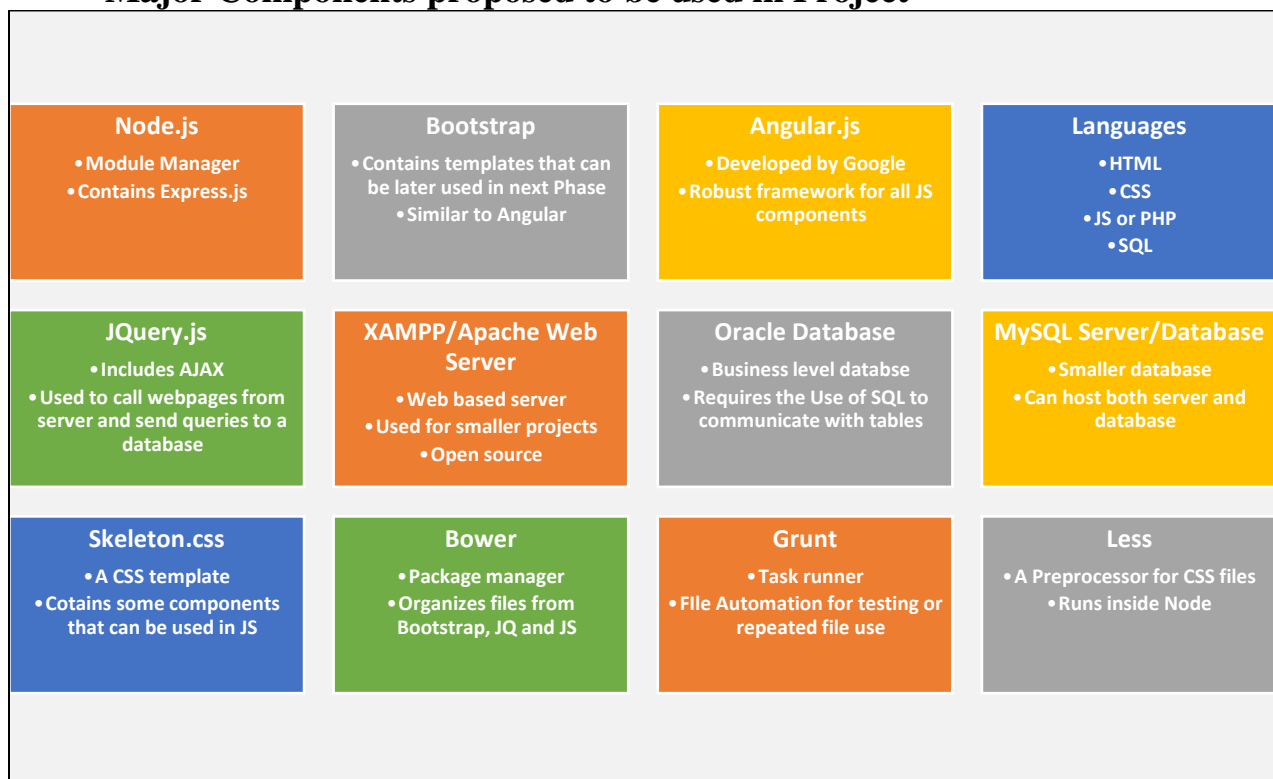
The sign up table generates a id from the account information. The table prompts the user to input their name, date of birth, school/university, and contact information. The user will choose what their username and password will be.

The Account Information table will be a significant part of the database. It generates an assignment number and customer name for the user. It also uses the same customer id from the sign up table. The user will then input the username and password that they chose.

## Gathering the Tools

The project requires various tools and resources that can be found on the internet. Some of these tools include:

### Major Components proposed to be used in Project



As previously stated **HTML** and **JavaScript** are used to create the actual website for both users and employees. **Sequence Query Language (SQL)** is used to send queries to a database and retrieve information from the table.

Most of these additional components/tools such as **Node.js**, **Grunt.js**, **Less.js** are thought to be used on Phase 2 of the project centered around improving the user interface (UI). **Node.js** is a component that is used to load certain design files for a website. The **Node.js** operates by installing modules that can be used in improving the interface of a system while providing supporting during the process of creating code. Note, **Grunt** and **Bower** are similar in nature and both are used to compile **Cascading Style Sheets(CSS)** files that are used in the project.

The **Glassfish Web Server** is used to host the website during the testing and implementation process. The **Glassfish Server** is an available feature of the **NetBeans IDE** as an add-on and will allow the user to load a project on to this server for anyone to access it. This is an important for future testing as it acts as a simulator before launching the project. The project also has the option to use two databases, **Oracle Database** and **MySQL** database.

**Skeleton.js** is a CSS framework/template that can be used for a website. A more robust framework such as **Angular.js** can also be used in later stages which can work in conjunction with **Bootstrap.js** if there are changes to the project.

## SECTION 3 – CREATING THE PROJECT

### Analysis

This project is modelled after the waterfall method where each step must be completed before moving on to the next step. This is necessary for this project as it gives the developers more incentive to complete each section as quickly as possible while providing as much information as possible.

### Comparing Discovered Tools

#### Major Tools

In this project the tools gathered will be separated into two categories, major and minor tools. Major tools are those that are a necessity in the development process and cannot be excluded. These tools are:

- **HTML**
- **JavaScript (jQuery, AJAX)**
- **PHP**
- **CSS**
- **A Web Server (Glassfish Web Server vs MySQL vs XAMPP)**
- **A Database (Oracle Database vs MySQL Database)**

## HTML

The **HTML** language is used to create webpage that are read by a browser interpreter. **HTML** is the industry standard when creating webpages on the internet. **HTML** documents contain elements that contain various attributes that can range from simple text to images, audio, video and

## JavaScript

**JavaScript** is a scripting tool used to give **HTML** documents and their elements function and create an interactive experience for users. This is important to the survivability of a website as websites that only display information have poor traffic

## JQUERY

**JQuery** is a **JavaScript** that can be used to fetch and send files to and from a server. **JQuery** also makes **JavaScript** easier to manage. **JQuery** selects **HTML** elements and performs actions on these elements without the need to This is done through the addition of **Asynchronous JavaScript and XML (AJAX)** in its library. **AJAX** is used as a client side scripting language that follows **JavaScript** and allows the scripting language to be more expansive and to communicate with the server.

**JQuery** also has the ability to send queries to databases that are found within the client or that have a link to an outside server. With this it allows websites to submit and retrieve information from a specified table. This component is a major key that will be used in this project so as to complete one of the requirements listed.

## PHP

**PHP** is a server side scripting language. It is most often used to retrieve web pages and sending it to the browser so as to be processed. **JavaScript** and **PHP** behave very differently as they operate on different ends of the connection process. **PHP** is helpful to calling pages from a server while **JavaScript** requires the addition of **JQuery** to give it this functionality. It should be noted, however, that **JavaScript** is more friendly to front end development than **PHP**. Both languages have the ability to be used parallel to each other with minimal intersects in code.

## CSS

**CSS** is used to create and stylized web pages and alter the way elements are displayed to the user. **CSS** also removes some of the coding needed when including it in an **HTML** document. **CSS** styles can be applied to any webpage or having several **CSS** styles being applied to multiple webpages.

## Server Choice

The two servers that are available are the **Glassfish Server** that is available as an add-on to the **NetBeans IDE** and there is the **MySQL Server**. The **Glassfish Server** is built for use of enterprise level applications and is more difficult to use. **MySQL** database is a simpler database which supports community development and leans more towards small teams or individuals than **Glassfish**. This version of **MySQL** allows for one to host their database on a native server. However, there is a possibility to change server choices to an open source software provided by **XAMPP** which host various servers such as the **Apache Web Server**, **FileZilla** and **Tomcat**.



**XAMPP** also provides additional components such as the **WordPress** module that allows publishing and website hosting.

### Database Choice

The two databases that have been found are the **Oracle Database** and the **MySQL Database**. **Oracle Database** is able to be linked to the **NetBeans IDE**, however, the reason that **MySQL** was chosen for this project is for two reasons. The connection that is normally used in the **Oracle Database** is unavailable and the **MySQL** Installer is able to host a server and a database in the same server at the same time. This eliminates the need for a developer to work only using **NetBeans IDE** and instead can directly upload the website to the server where it can easily interact with the database.

### Minor Tools

The minor tools are tools that have no significant impact and can be later applied to the project to help with its look and functionality. The minor tools can range from additional components or to the changing of the visual aspect of the website. These tools are:

- **Node**
- **Bootstrap**
- **Grunt**
- **Bower**
- **Less**
- **Google Search Engine**

### Node.js

**Node**'s primary function is to implement some of the frameworks and additional modules to **HTML** documents and **JavaScript** scripts. These nodes help with automating some of the frequent tasks. Some of the nodes that are implemented by the module are **Bower**, **Grunt** and **Less** along with **Angular** is later on in this project.

### Bootstrap

**Bootstrap** should be used in the second phase of the project. However, it is important to discuss some of the capabilities that **Bootstrap** offers when compared to the other components and tools that were sourced for this project. **Bootstrap** is an all-around framework that supports **HTML**, **CSS** and **JavaScript**.

### Grunt

**Grunt** is an automated script tester that tests the scripts that are created to verify if they are correct. This is an important possible component that can be used in the project if the website increases in size and requires more functionality.

### Bower

**Bower** is

## Google Search Engine

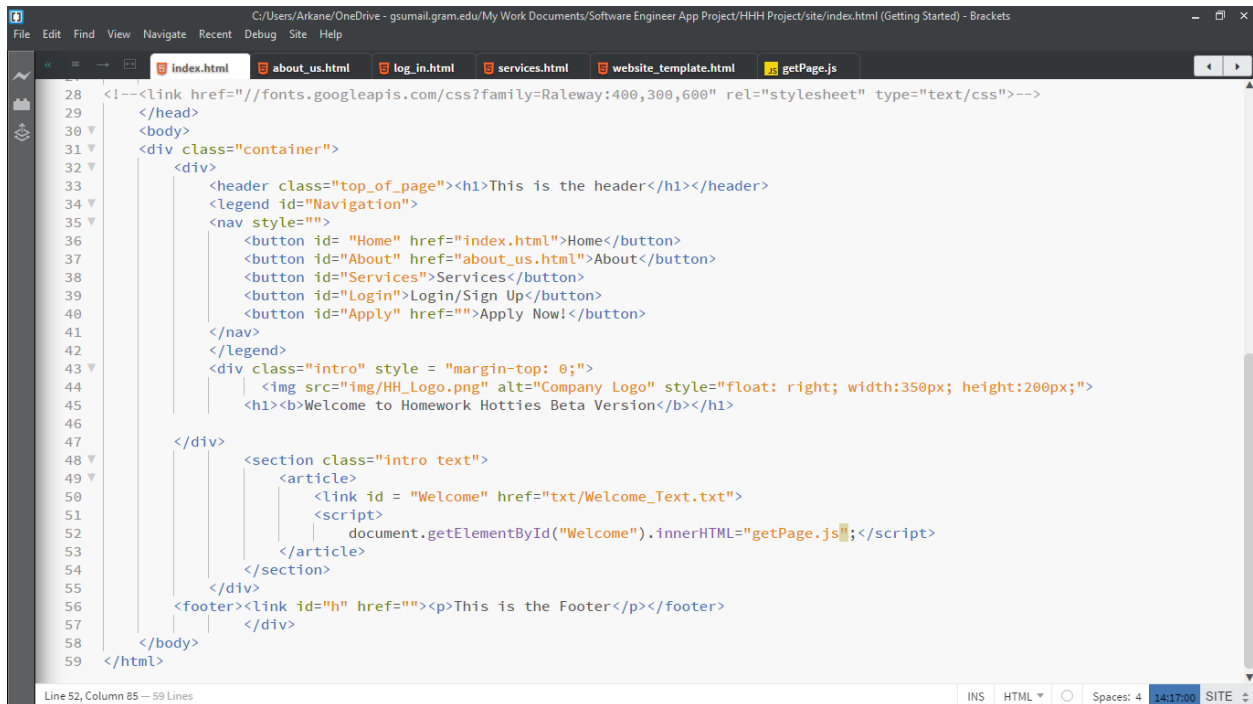
A search engine needs to also be implemented that finds webpages by using user input. The ideal search engine to be used is the **Google Search Engine**. The search engine will use various tags and keywords to find a list of webpages that match the parameters in the search engine. The search engine should also be able to be used in searching for general documents such as site, user account,

Maximizing the remaining time is a priority for this project. Therefore, the website will have minimum components to them and be focused on fulfilling the requirements as previously stated.

## Software Testing & Implementation

**November 24, 2016**

The first step of the project was to create basic HTML files that follow the DOM design. This design of the HTML was used to create a template that could be reused to create new websites and would cut down on the amount off coding needed for each webpage.



```

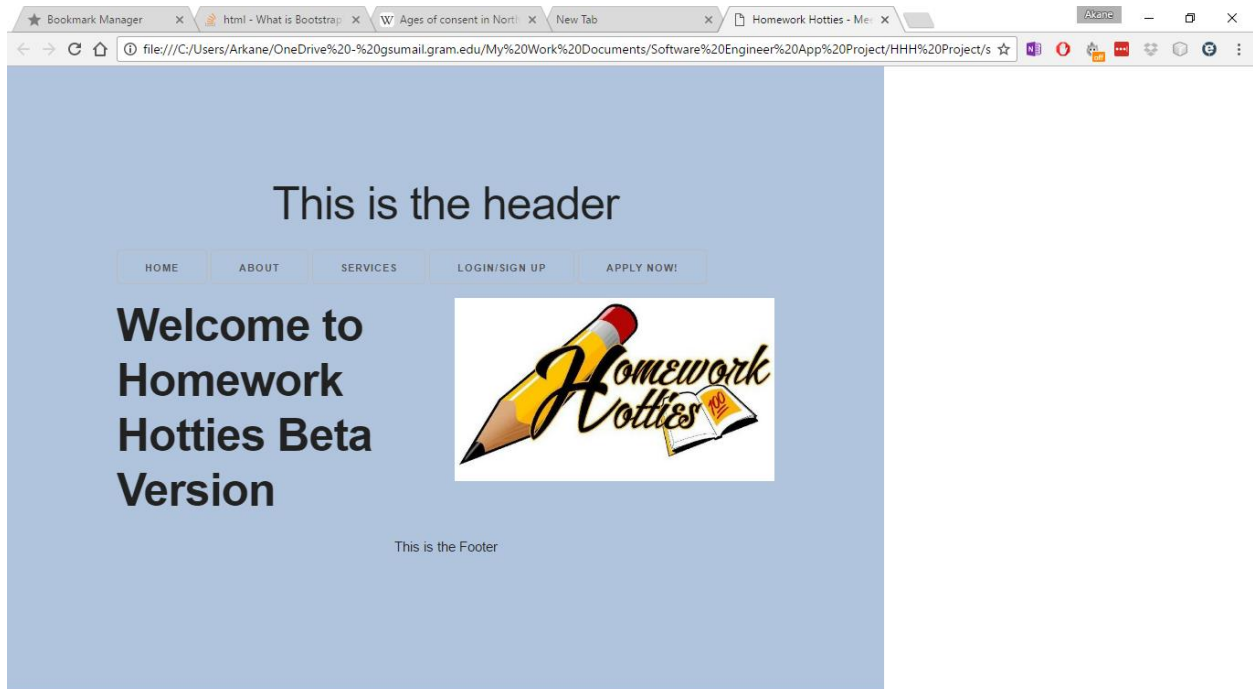
28 <!--<link href="//fonts.googleapis.com/css?family=Raleway:400,300,600" rel="stylesheet" type="text/css"-->
29 </head>
30 <body>
31 <div class="container">
32 <div>
33 <header class="top_of_page"><h1>This is the header</h1></header>
34 <legend id="Navigation">
35 <nav style="">
36 <button id="Home" href="index.html">Home</button>
37 <button id="About" href="about_us.html">About</button>
38 <button id="Services">Services</button>
39 <button id="Login">Login/Sign Up</button>
40 <button id="Apply" href="">Apply Now!</button>
41 </nav>
42 </legend>
43 <div class="intro" style="margin-top: 0;">
44 
45 <h1><b>Welcome to Homework Hotties Beta Version</b></h1>
46
47 </div>
48 <section class="intro text">
49 <article>
50 <link id="Welcome" href="txt/Welcome_Text.txt">
51 <script>
52 document.getElementById("Welcome").innerHTML="getPage.js";</script>
53 </article>
54 </section>
55 </div>
56 <footer><link id="h" href=""><p>This is the Footer</p></footer>
57 </div>
58 </body>
59 </html>

```

Above shows a screenshot of the current HTML document that displays the current template for the project.

One of tools previously mentioned, **Skeleton.css**, is used to give a basic clean look to the website, primarily on the font of the text and buttons.

Within the document



The webpage is basic and none of the links displayed are functional. The log used in this project also needs to be changed to a transparent background to prevent

### **November 25, 2016**

Testing began on and focused on the viability of the scripts made especially when calling pages from the site folder. The scripts created would be based on either JavaScript and PHP. PHP would be used when calling pages from the server while JavaScript would change the elements of the web page with every interaction.

So far several pages have been created such as the About and Contact page but no substantial progress have been made.

### **November 26, 2016**

The **MySQL Database Server** was setup under the default name localhost with port connection being 3306. After the creation of this server a database dubbed "Project Test" was created to hold the schema and eventual tables to be formed.

### **December 8, 2016**

Another server was found and installed on the development system.

## References

- Duckett, Jon. *HTML & CSS : Design and Build Websites*. Indianapolis: John Wiley & Sons, Inc., 2011. Book.
- . *JavaScript & JQuery: Interactive Front-End Web Development*. Indianapolis: John Wiley & Sons, Inc., 2014. Book.
- Hassani, Hossein. "Understanding XML: A Software Development Approach." 2015. *Bookboon.com*. PDF. October 2016.
- Lang, Curt and Jeff Chow. *Database Publishing On The Web & Intranets*. Scottsdale, Arizona: The Coriolis, 1996. Book.
- Shah, Nilesh. *Database Systems Using Oracle, A Simplified Guide to SQL and PL/SQL*. Upper Saddle River, New Jersey: Prentice-Hall, 2002. Book.