*Technical Manual*

*Contents*

*Introduction* 1

*Purpose* 1

*Software Requirements* 1

*System Overview* 2

*Hosting Information* 2

*HTML Forms* 2

*Additional Software Plugins Used* 2

# *Introduction*

The Athens State Senior class has worked with the Limestone County 911 Center in the restructuring of their website and enhancing it with additional functionality. The project consisted of re-designing the job application form and address request form. Users will also have the ability to submit both forms online, and the forms will then be sent via email to the 911 facility for review. A “Smart 911” tab was also added that includes 3 videos and a link to the Smart 911 facility. Users will also be able to view pictures that will cycle though a carousel on the homepage. The GIS map will now open up in a separate browser that will give the user a better sense of visibility. All files used to create the site can be located on GitHub <https://github.com/bmbost/cs452-911/pull/3#event-642338825> .

# *Purpose*

The purpose of this document is to assist future developers in expanding the functionality of the Limestone County 911 Center website. It will give a step-by-step guide on how the database was setup, how the forms were created, and what software is needed for development.

# *Software Requirements*

The following packages will need to be downloaded in order to update and test website during development phase. All software is free.

1. The Apache HTTP Server Project.
2. phpMyadmin.
3. MySQL
4. XAMPP (**Note**: XAMPP includes all 3 above)
5. Notepad++ or some other IDE

**For Windows operating systems:**

* <https://httpd.apache.org/download.cgi>
* <https://www.phpmyadmin.net>
* <https://www.mysql.com/downloads/>
* <https://www.apachefriends.org/download.html>

**For Mac operating systems:**

* <https://directory.apache.org/studio/download/download-macosx.html>
* <https://www.phpmyadmin>
* <https://dev.mysql.com/downloads,mysql>
* <http://m.en.softonic.com/app/xampp/mac>

# *System Overview*

## *Hosting Information*

Athens-Limestone County 911 Center has purchased website hosting services through GoDaddy. All updated website files must be placed on the GoDaddy host when going "live." This will allow the forms to be interacted with on the ALC website by the applicant. The database and tables will have to be exported and the connect file in the PHP will need to be updated to contain the appropriate login information.

## *HTML Forms*

The job form and address request form were both constructed using a combination of HTML, CSS, and Javascript. HTML comprises the form elements and static web page display. CSS code controls the format of HTML elements on the web page. JavaScript functions were incorporated to add form interactivity such as button clicks, next page, and browser side validation.

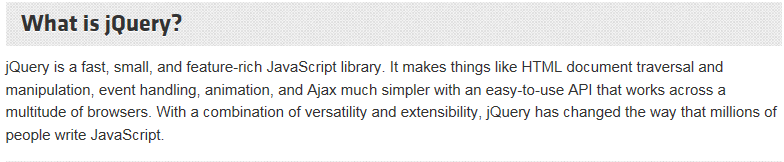
### *Additional Software Plugins Used*

Plugins were used on both forms. The purpose of using plugins is to aid in the development process. Many of the plugins already provided functionality such as validation or datepicker. So rather than reinvent the wheel and code things like this from scratch, we simply added the following plugins.

#### jQuery

The jQuery library was used throughout the form to assist with coding in Javascript. See Figure 2 below for a description of jQuery. The jQuery library can be downloaded from <https://jquery.com/>.

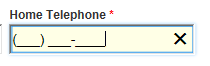
Figure 1



#### Masked Input

The Masked Input Plugin was used to create input masks for fields on the job application form such as phone numbers and zip codes. The Masked Input Plugin shows the user what will be expected when a given field is in focus and only allows characters that meet the requirements set for the field. Figure 3 below is an example of an input mask for the Home Telephone number on the job application form. The Masked Input plugin was implemented in the jobappform.js file.

Figure



<http://digitalbush.com/projects/masked-input-plugin/>

#### Tooltipster and jQuery Validate

Two plugins, Tooltipster and jQuery Validate, were used synchronously for browser side validation. jQuery Validate is a fully configurable tool that allows you to set predefined (or custom) validation rules and error messages for form fields. Tooltipster is a plugin that displays tooltips for form fields. On both the job application form and the address form, these two plugins work together in providing the user an “error” message that display in tooltips beneath the fields when filling out the forms. The error messages occur immediately on keyup (typing in a form field) or if the form is submitted. jQuery Validate and Tooltipster is configured within the custom Javascript files that go along with each form (in the Document Ready function). The plugins can be downloaded from the following sites:

<http://iamceege.github.io/tooltipster/>

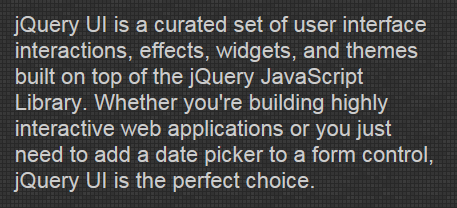
<https://jqueryvalidation.org/>

#### jQuery UI (User Interface)

jQuery UI was only used in the job application form. Please see Figure 4 below for description. This plugin offered a couple of interface interactions that we needed for the form, and it was incredibly easy to implement. jQuery UI was used to add a date picker (calendar pop up that allows user to choose a date) to all date fields on the job form. This was accomplished by calling date picker in the Document Ready function of the custom Javascript file. All form fields with the class “datepicker” were affected.

Additionally, jQuery UI was used to create collapsible employers on Part 4. Employment History of the job application form. The jQuery UI Accordion function was used to accomplish this task. The Accordion was implemented in both the structure of the HTML and by calling the accordion function on the Document Ready function.

Figure



The PHP that was written will interact with the HTML job application form. This is done by setting the HTML form to send to the PHP form POST on submit. The PHP form then takes the information sent POST from the HTML variables and sets them to the PHP variables. From there, PHP can interact with MySQL because it knows the language. It is capable of querying information and inserting it into the database. It can also pull information out of the database in the same way. The PHP form includes a database connect section of code that connects to the database using the username and password set up for the database.

Once the user submits one or both forms they are then sent to a representative’s email address. The PHP mail function is used to create this feature. HTML creates the table and PHP grabs the variable POST. An email then gets sent to the address that is created inside PHP mail function.

For the emails, PEAR PHP’s mail client was used. This is an add on for PHP that is more robust than the default mail client built into PHP. The biggest advantages of this add-on are the abilities to set up the account details for the server and set up the account to be used to send out the e-mails from within our code. Captcha code was also added to prevent spam e-mails from being sent to clients.