

Name: Adam Hemmann
Pawprint: ashgx6
Student ID: 10235141
Date: 12-10-2021

CS 2830 – Web Application Development I

Final Project Supplemental Documentation

This document is intended to provide a basic walkthrough of both the functionality of my web application as well as the constituent code. I will provide a section for each page as well as any php action scripts used on that page with relevant references to line numbers in the code where appropriate. The web address for my application is:

<http://34.66.157.22/2830FinalProject/>

Foreword

I work as a warehouse manager for an online printing company, based in Columbia and Pittsburg, KS; called Miller's, Inc. Our two primary brands are Mpix and Miller's Professional Imaging. For my final project I created a web-based system for viewing and updating our paper inventory. The inventory data I used in my database was pulled directly from our internal inventory system.

header.php

This file is included with every page contained within the app. Most importantly, `header.php` calls `session_start()` (see line 3) and will redirect the user to the appropriate error page if `session_start()` is unsuccessful. Additionally, `header.php` provides the user with a set a navigation links which change both based on the user's admin status and whether or not they are logged in (see lines 30-69).

index.php

When navigating to the app, the user is greeted with a simple login/splash page. The nav bar at the top displays two links to the web sites for both Miller's Professional Imaging and Mpix. The login form accepts inputs for a username and a password. Below the form is an embedded YouTube video from our channel, detailing the production process at our Columbia facility.

The login form is processed via `php_action/doLogin.php` through a jQuery event listener. AJAX is used here to simultaneously submit the form data to the server for processing and to provide the user with feedback if they enter input which is invalid. See `js/app.js`, lines 48-68.

For demonstrative purposes, I have created three users in the database:

- Username: adamh | Password: password | Admin: **yes**
- Username: test | Password: pass | Admin: **yes**
- Username: samg | Password: password | Admin: **no**

showInventory.php

Upon a successful login, the user is redirected to `showInventory.php`. This page is meant to serve as a dashboard for a user who wants to view current inventory data. The page contains two

Name: Adam Hemmann
Pawprint: ashgx6
Student ID: 10235141
Date: 12-10-2021

snippets of php: One to grab the total number of items being tracked in the database (see lines 28-39), and the other to generate an HTML table to display current information for each item in the inventory (see lines 60-84).

editInventory.php

The first of three pages which can only be access by users designated as admin, `editInventory.php` displays a single `<select>` object which is populated as soon as the page loads with the name for each item in the database (see lines 33-48). Upon selecting an item, jQuery is used to render a form object which contains data specific to the selected option (see `js/app.js`, lines 4-18).

addUsers.php

Similar to `editInventory.php`, this page can also only be access by users designated as admins. Here, the user can input data to create a new user for the app. The ability to set a user's admin status is included in the form. Once again, jQuery and AJAX are used to handle the form submission (see `js/app.js`, lines 81-105). The form is processed via `php_action/doCreateUser.php`.

removeUsers.php

This page works very similar to `editInventory.php`. A select object is populated via php (see lines 30-45). Importantly, the user which is currently logged in will **not** be displayed as an option for deletion. Once an option has been selected jQuery is used to submit the form data to the server via `php_action/doRemoveUser.php`. See also `js/app.js`, lines 107-132.