Michael Anderson
COMP 296
Prof. Mattos
Final Submission Readme

ONLINE AUCTION SYSTEM - README

Table of Contents:

Project objective:

The Online Auction System permits users to solicit bids for various services through an online interface and for other users to bid competitively to provide the services. Through an online interface, the consumer-user creates an auction listing with a description of the job to be done and sets a maximum bid for the prospective job. The auction listing is posted to the Online Auction System website for review by other users. Users can then place bids - which represent the cost to be charged to the consumer - through the website during the period of time that the auction is ongoing. At the end of the auction process, the system determines which service provider has placed the lowest bid. At the conclusion of the auction, the user that created the auction and the low bidder are provided with one another's contact information and are given use of an interface that allows them to update job status - e.g. complete, incomplete or in dispute.

Implementation information:

The system has been coded principally in PHP. In addition, Javascript is used for client-side data validation, and Java is used for the server-side "Auction Crawler" module, which queries the database every 60 seconds and updates auction listings that have expired. HTML and CSS are used for webpages.

File organization:

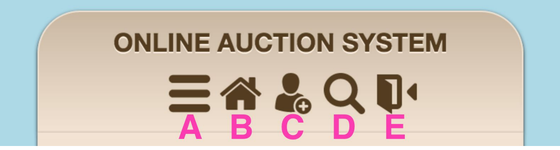
AuctionSystem

- Classes - Contains PHP classes that can be instantiated and used in PHP scripts
- formHandlers - Contains PHP scripts that process form data and connect with database
- publicHTML - Contains PHP files with HTML components
- databaseSchemas - Contains database schemas in .SQL format
- images - Contains Public Domain/Creative Commons License images for use as icons

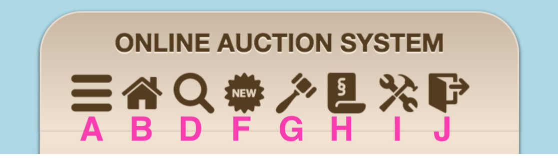
Note: The main CSS file (style1.css), config.php, and index.php are contained in the AuctionSystem directory itself.

GUI Navigation:

While logged out:



While logged in:



A: Dropdown navigation menu (same options as navigation icons to right)
B: Home screen
C: Create new account
D: Search actions
E: Log in
F: Create new auction
G: User's auctions
H: User's bids
I: User's jobs
J: Log out

Note: What is displayed in on a given screen will vary based on whether a user is logged in; whether there are active auction listings or not; active bids or not; active jobs or not; etc. The list above does not include various ancillary screens, but is only intended to be a guide to the main navigation menu.

Time spent on project and thoughts on use of time:

Though I did not take rigorous notes as to time spent on the project, I can say with confidence that over 100 hours were spent on the project, inclusive of learning and researching PHP. At least 15 hours were spent on each of the last five weekends on the project, in addition to smaller time investments on weekday nights.

My use of time was not always the most efficient - on several occasions, I had to scrap and rewrite code that had become unwieldy or difficult to integrate with other parts of the program. Learning PHP on the fly contributed to the amount of time required to build the system. Ultimately, the experience was extremely rewarding, but learning on-the-go necessarily led to some subpar code that had to be replaced, which proved time consuming. However, towards the end of the project, the amount that I could code/accomplish in a given amount of time was many times more than when I first began.

Set Up:

The auction system consists of three components: a website hosted on an Apache webserver with scripts written in PHP; a MySQL database; and a server-side “auction crawler” program written in Java that updates the database.

The folder AuctionSystem should be placed in the HTDOCS folder that serves as the root server directory for Apache web server. Apache should be running locally. The Apache port should be set as 80.

A MySQL server should be running locally. The database schema and data can be loaded by running the SQL script named mysql_script_05_01_16.sql. The database password should be set as root and the database user should be set as root. The database name should be auction_system. The MySQL server port should be set as 8889.

The Auction Crawler source code is in AuctionSystem/databaseCrawler/AuctionSystemCrawler V2/src. The program should be compiled and run in an IDE such as Eclipse or exported as a JAR and run. The JDBC/MySQL connection driver, Mysql-connector-java-5.1.38-bin.jar, (which is saved in directory AuctionSystem/databaseCrawler and is also available at https://dev.mysql.com/downloads/connector/j/) must be included in the filepath of the program. The Auction Crawler redirects System.out to a text file to serve as a log; before compiling the source code running the Crawler, the constant variable OUTPUTLOGFILEPATH must be changed to the filepath of the directory you want the log file to be written to. Alternatively, the code that redirects System.out to a file can be commented out with the log messages printed to the console. Once started, the Crawler will run in an infinite loop until stopped through the Crawler’s GUI, which consists of a single button that shuts down the program.

The system should now be running locally. The website may be navigated to in a browser by going to localhost/auctionsystem.

Please note that the CSS file uses a media query to determine the CSS layout to use; the site will be optimized for mobile devices if the browser window is less than 650 pixels wide; otherwise, it will display in wider format for desktop/tablet.

References:

Codular - MySQLi for Beginners
http://codular.com/php-mysqli

Devcurry.com - Validate US zip code using Javascript
http://www.devcurry.com/2010/07/validate-us-zip-code-using-javascript.html
Note: Code example used as reference for zip code validating script

HTML Code Tutorial - Restricting a Field to Numbers Only
http://www.htmlcodetutorial.com/forms/index_famsupp_158.html
Script used with slight modifications for validating HTML fields containing money values

Icons DB
http://www.iconsdb.com/
Free-to-use images and icons

Liang, Y. Daniel. Introduction to Java Programming: Comprehensive Version. Boston: Pearson, 2013. Print.
Note: Used specifically with respect to Chapter 34 on Java database programming and JDBC in coding the Auction Crawler

Live Tools - Form Builder
http://livetools.uiparade.com/form-builder.html#
Note: Generated initial CSS for forms

Mozilla Developer Network - Using media queries
https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries
Note: Explanation of CSS media queries, which permitted different CSS designs to be used based on the size of the screen viewing the page

PHP.net - mysqli_insert_id
http://php.net/manual/en/mysqli.insert-id.php
Note: PHP documentation for MySQLi - Useful for returning the primary key of a newly inserted record

Quigley, Ellie, and Marko Gargenta. PHP and MySQL by Example. Upper Saddle River, NJ: Prentice Hall, 2007. Print.

Sitepoint.com - Hashing Passwords with the PHP 5.5 Password Hashing API
http://www.sitepoint.com/hashing-passwords-php-5-5-password-hashing-api/

Stack Overflow - Creating a PHP header/footer http://stackoverflow.com/questions/8054638/creating-a-php-header-footer

Stack Overflow - Email match with confirm email address in contact form
http://stackoverflow.com/questions/18223467/email-match-with-confirm-email-address-in-contact-form
Note: Used code in part for email validation script (Javascript)

Stack Overflow - How to fix a div to top of page with CSS only
http://stackoverflow.com/questions/6786511/how-to-fix-a-div-to-top-of-page-with-css-only

Stack Overflow - Java: How to save console output to a text file
http://stackoverflow.com/questions/21476423/java-how-to-save-console-output-to-a-text-file
Note: Used for Auction Crawler to redirect System.Out to .txt file

Stack Overflow - Regex: Validate email address in Javascript
http://stackoverflow.com/questions/46155/validate-email-address-in-javascript
Note: On using regular expressions to validate email addresses, and why it might not be a good idea

W3Schools.com - CSS Dropdown
http://www.w3schools.com/css/css_dropdowns.asp
Note: Code snippets used in coding dropdown menu

W3Schools.com - Javascript form validation
http://www.w3schools.com/js/js_validation.asp

W3Schools.com - PHP Prepared Statements
http://www.w3schools.com/php/php_mysql_prepared_statements.asp
Note: Useful for learning how to format prepared statements with MySQLi

W3Schools.com - Using CSS with HTML forms
http://www.w3schools.com/css/css_form.asp