

# Lion Auction ReadMe

Liam Sullivan  
CMPSC 431w  
4/23/2023

## Functions

**Generate Address ID** - This function simply generates a unique address id to be stored in the database

**Contact** - Event handler for contact button pressed in the auction listing web page.

**My Bids** - Allows the user to view the auctions that they have placed a bid on.

**Rating\_Left** - Event handling for updating the database after a rating is made.

**Leave Rating** - Prepares a webpage to be rendered for users to view all of their completed auctions and leave ratings.

**Go Home** - Rerouts back to the homepage and resets some state values.

**Verify Listing** - Creates and verifies an auction listing being created

**Choose** - Allows a seller to choose if they would like to view active or inactive listings.

**View Listing** - Filters out listings by their status

**Update** - Event handling for the update of a product listing

**Cancel** - Cancels an auction listing and resets status.

**Reasoning** - Handles the event of a seller taking down a product and leaving a reasoning for doing so. Updates the database with this reason.

**Create Listing** - Renders out the webpage for a user to create a listing.

**Go Back** - Exclusively used for the categorical hierarchy to make it easier to move up and down the tree.

**Product** - renders out a product listing.

**Seller Create** - Updates a state variable that re-renders the create user page by listing the required fields for a seller.

**Create Account** - Handles the rendering and database work of a user creating an account. Accommodates sellers and bidders.

**Update Usertype** - Handles the verification and state change of a a user deciding to switch roles without logging out.

**Bidder Home** - Provides the necessary information for the rendering of a bidders home page

**Seller Home** - Provides the necessary information for the rendering of a sellers home page.

**Search Results** - prepares the listings that are to be seen inside of the search bar on the home page.

**Get Bids** - Conducts the necessary database operation to find the bids to be listed on the product page of a specific listing.

**Select Product** - This is the most important function I implemented. The product page is highly interactive. This function handles the rendering of a popup window that allows users to place a bid, various messages for errors in placing a bid, and the verification process of a valid bid placement. Additionally, this function handles a winning bid scenario by verifying the bid, and redirecting the user to a separate page to complete the transaction.

**Generate Transaction ID** - Generates a transaction id based for a given transaction

**Transaction** - Event handling for the winning of an auction. Transaction table populated.

**Get Subcategories** - Finds the subcategories of a given category

**Categorical Hierarchy** - Assists in the rendering and presentation of the categorical hierarchy traversal

**Update Profile** - Receives input by the user and updates a given profile.

**Authenticate** - Authenticates a user to login to their web page.

**User Type** - Another step used in the authentication process for seeing if a user is registered as a given user type.

**Logout** - Resets state and redirects user to the login page.

**Login** - Sets state and redirects users to their corresponding login page.

**Search.py** - Various functions using character matching to display matching products when typed into the search box.

**Setup.py** - Generates a csv file that stores the securely hashed passwords to be stored in a database.

## Organization

This project's functionality is mostly defined in app.py. Here lies almost 1,000 lines of code controlling the backend of this site. Aside from app.py, I defined search.py to handle the character matching of the search box, and setup.py to handle the secure hashing. The front end of this project was designed with 15 separate html files. Some of these files, especially product\_listings.html, served multiple purposes and were rendered with other files to produce more complex pages.

## Features/Operations

- Login
- Change user type
- Logout
- *Create new Account*
- Categorical Hierarchy
- *Search Bar*

- View Auction listings that user is involved in (for bidders and sellers)
- Take down / update auction listings
- *Update profile*
- *Leave rating*
- View Products
- Place bid
- Leave reasoning for taking down a bid.
- Input credit card info and populate transaction tables.
- View product info and average rating
- View Sellers contact info
- Return to the home page from any given page.
- Move forwards and backwards in categorical hierarchy.
- View Lists of products with all their information. Products listed are defined using specific search criteria.

## **Instructions**

Correct database creation and population is done manually initially from setup.py, the query console, and from the pycharm database functions. Setup.py will correctly hash the password values into a new csv file, which can then manually be inserted into the users database; Additionally, a few functions must be conducted in the query console. For example, the auction listing table needs to be updated with remaining bids set to max bids, and a reasoning column initially set to null. The app can be easily run through pycharm, and no additional instructions should be needed to run it. Query console will

