

LionAuction

Term Project (Phase 2)

CMPSC431W - Database Management Systems

Author : Anurag Pendyala Email : app5997@psu.edu

Folder Structure

```
+-- data
|   +-- Address.csv
|   +-- Bidders.csv
|   +-- Categories.csv
|   +-- Helpdesk.csv
|   +-- Ratings.csv
|   +-- Sellers.csv
|   +-- Users.csv
|   +-- Auction_Listings.csv
|   +-- Bids.csv
|   +-- Credit_Cards.csv
|   +-- Local_Vendors.csv
|   +-- Requests.csv
|   +-- Transactions.csv
|   +-- Zipcode_Info.csv
+-- static
|   +-- css
|   |   +-- style.css
|   +-- images
|   |   +-- logo.png
+-- templates
|   +-- base.html
|   +-- index.html
|   +-- pay.html
|   +-- profile.html
|   +-- bidder_dashboard.html
|   +-- login.html
|   +-- pay_handle.html
|   +-- seller_dashboard.html
+-- app.py
+-- database.db
+-- Readme.md
+-- Readme.pdf
+-- utils.py
```

Requirements for the Website

Make sure the following are installed:

- Python3
- SQLite3
- Flask
- Web Browser
- hashlib

Running the Website

1. Go to the folder where `app.py` and `populate_data.py` are located. Make sure you have the `.csv` data files in `data/` folder. Run the following command to load the data to a database.

```
python3 populate_data.py
```

This command can be run every time a reset to the database is required.

2. To access the website, run the following command.

```
python3 app.py
```

By default, the website can be accessed at `http://127.0.0.1:5000/`. However, this might differ based on the port availability. This link can be accessed from the output of the above command. Go to the link. Make sure the program is still running in the background.

Navigating the Website

Upon opening the link, you are welcomed with a Login page where you are requested to give an email ID, password, and a role with which you want to login. If you are both a Bidder and a Seller, you can change it later in your Profile section. After successfully logging in, you will go to your role specific dashboard.

Profile

This button on the navbar navigates to the profile page of the user. If the user is a bidder, the page shows the personal details of the bidder. If the user is a seller, the page just displays the role to be seller. If a user is both a bidder and a seller, they will have an option to change the role so they can perform their specific duties.

Logout

This button on the navbar just logs the user out of the website and will be redirected to the login page.

Bidder Dashboard

You can choose the category of products that you wish to browse through. By default, items belonging to all categories are presented. The drop downs can help you narrow down to the category of interest. According to the category, respective sub categories can also be chosen to filter the results further.

A few items are not eligible for auction yet. For those, the product card will have `Unavailable`. Else, you can bid for the product. You cannot bid for the product only when the last bid is also yours. Also, if the item reaches maximum bids, the bidder is notified and can go ahead and pay for the item. Eligible bids can be placed by entering a price which is atleast \$1 greater than the current price. After selecting the price, the bidder can click on bid and the bid will be placed.

Payment Procedure

Once the bidder has secured the product, they will be notified and can go ahead and pay upon clicking a button saying Pay. The bidder will be redirected to a payment page which summarizes the purchase and asks for payment confirmation with the credit card shown. After the payment is done, a page displaying the Transaction ID is shown. Bidder can note this TID for future references if needed.

(Extra) The bidder can also review the seller after the purchase. This is completely optional but will help the seller by looking at their feedback.

Note: The number of bids for a particular product is maintained using another SQL relation with Item ID and Number of Bids. This helps in easily cross checking the Maximum Bids and Current number of Bids for a particular item.

Seller Dashboard

The seller has 2 buttons that can be clicked for the following two actions:

Viewing owned items

The seller can view all the items that belong to them. Few items are unavailable, few are available for auction, and a few are already sold. These sold items are not displayed at all. For the remaining 2 kinds of items, the seller has an option to change the status for available to unavailable by just clicking the button `Toggle Status` on each of the product cards. Once the product is sold, it vanishes from the list. However, it is still stored in the database and can be accessed by the admin.

Add new item

The seller can click on this button to add a new item that goes for auction. Appropriate details can be filled in the redirected form and submitted. By default, the new item is unavailable for auction. However, again, it can be toggled with just a button click.

Folders and Files Description

- `data/` - Directory that contains all the `.csv` files with data of the users, items, creditcards, addresses, bids, etc.
- `static/css/style.css` - Is the stylesheet used to customize various divisions in the HTML code for the website.
- `static/images/logo.png` - Is the logo of Lion Auction created for the project. This can be seen in the Navbar spanning across all the various pages of the website.
- `templates/` contains the HTML website codes for various pages. `base.html` sets the base for all the pages. It has code to display the navbar and various pages that can be accessed from every page like Dashboard, Profile, and Logout.
 - `bidder_dashboard.html` and `seller_dashboard.html` contains html code to display the dashboards for both the roles.
 - `login.html` Displays the login page.
 - `pay.html` and `pay_handle.html` has the html code to confirm and make payment for the bidder to purchase an item.
 - `profile.html` shows the profile of the user.
- `app.py` is the main file that needs to be run to access the website. Each function routes a particular page in the website. Each function is explained in the `app.py` file itself.

- `populate_data.py` is the first file that needs to be run to load the data into `database.db` SQL-Based database.
- `utils.py` is the file that contains hashing code required to store the passwords for the users since direct text cannot be stored in the database.

References

1. [Article](#) for SHA256 Hashing for Passwords
2. [Youtube Tutorial on Flask](#): Python Website Full Tutorial - Flask, Authentication, Databases & More.
3. HTML, MySQL, Flask Tutorials - Media Gallery of CMPSC431W Spring 2023.