To start the application run the following command: *python app.py* in the environment where Flask and required libraries are installed.

Requirements:

- **Framework:**

  - Flask *(pip install Flask)*

  - Flask Forms *(pip install Flask-WTF)*
- **SQL:**

We used Flask-MySQLdb *(pip install flask-mysqldb).*

There are two possible ways to connect with the database *auction*:

1. First way is to install My SQL Server, next do this https://stackoverflow.com/questions/22447651/copying-mysql-databases-from-one-computer-to-another and then write your root password in the following line in file dbconnect.py:

   conn = MySQLdb.connect(host="localhost", user = "root",
     passwd = ***"Your password"****, db = "auction")*

2. Second way is to install **XAMPP** and use it to connect with the database. Then the line in file dbconnect.py looks like this:

   conn = MySQLdb.connect(host="localhost", user = "root",
     passwd = **""***, db = "auction")*

Functionalities:

Sign up: registering new users. Sending email with login and password.

Log in: you should log in to see content of the other pages (available products, new offer and archives are reachable only for logged users)

After logging in button *Log in* is changing to *Log out* and button *Sign up* is changing to button with username and dropdown menu which contains:

- **Settings:** you can change password here.
- **Purchased:** you can check products that you have bought here.
- **My offers:** this page contains offers created by logged in user divided into sold and current offers.

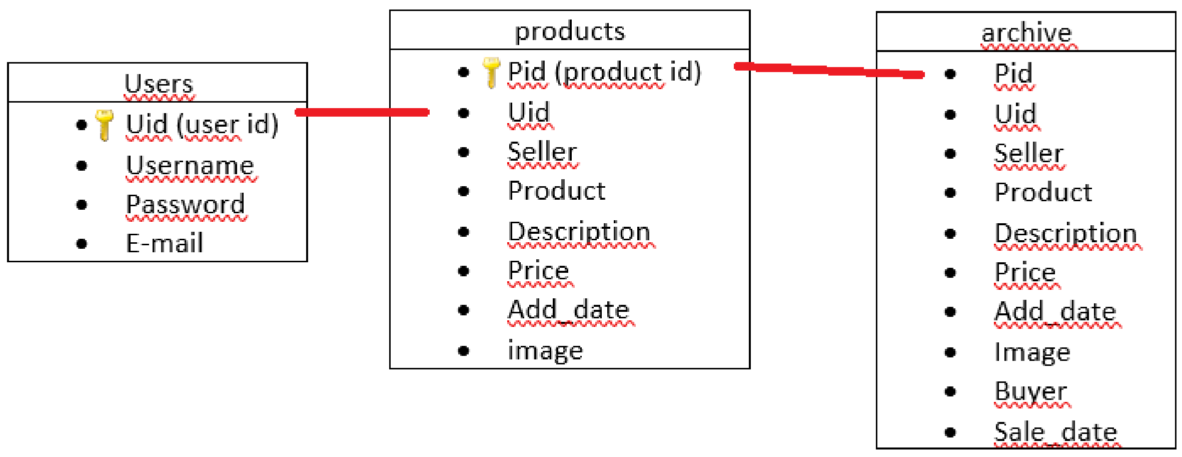
Available products: page contains current users offers.

New offer: form for adding new offers.

Archives: sold products.

<u>How it works:</u>

Auction.sql - database scheme



Main folder

app.py - Defines:

- **function index** - returns home page.
- **function register_page** - based on the registration form. In case of correctly filled fields, it adds the values entered in the form to the database and sends an e-mail to the address provided in the form with the login and password. If the data is entered incorrectly or there is a user with the same name or email, it returns a message.
- **function login** - based on the login form. If the data is correct, it logs the user in and creates a session.
- **function logout** - Defines a logout button to log the user out when clicked and displays a logout message.
- **function products** - Fetching in a list of all inserted products from the products table. The data is then displayed on the subpage “product.html”.
- **function archive** - Fetching in a list all bought products from the archive table. The data is then displayed on the subpage “archive.html”.
- **function buy_product** - Defines the button displayed under offers on the subpage. After clicking the *buy* button, the offer is inserted in the archive table with the added

information about the purchase date and the buyer. Then the observation is removed from the products table.

- **function settings_page** - After correctly entering the old and new password, it will update the password column for that user in the users table. In case of not entering all fields or wrong password, a message will be displayed
- **function purchased** - Collects data from the archive table concerning logged user and then displays it on the "purchased" subpage.
- **function my_offers** - Collects data from the archive and products tables concerning logged user and then displays it on the "my offers" subpage
- **function new_offer** - Adds the data entered in the form to the products table and attaches the selected photo. If all data is not entered, it returns a message

dbconnect.py - Defines a database connection
Forms.py - forms are defined here.