Nittany Market Demonstration (10-Minute-Reading)

Designed by *Jun Kang*

Backgrounds

NittanyMarket is a new platform under planning which aims to provide members of Nittany State University (NSU) to sell and buy used or unused goods online. The original objective of the platform is to allow departments and units of NSU to handle the removal of surplus equipment owned by NSU in environmentally responsible ways by reselling them to students, faculty, and staff in discounted prices. As a student of Dr. Lee, I implemented an online market place as Nittany Market.

Goals

To build the platform comprehensively, need to accomplish the following tasks:

- Populate all data into the database
- Design HTML5 pages
- Implement function logics
- User LogIn
- Checking Info
- Category Hierarchy
- Publilsh Product Listing
- Place Order
- Review & Rating
- Shopping Cart
- Searching

Technologies Used

Tech	URLs
PyCharm	https://www.jetbrains.com/pycharm/
Flask (Python)	https://flask.palletsprojects.com/en/2.0.x/
HTML5 (CSS & JS)	https://www.w3schools.com/html/
MySQL	https://www.mysql.com
jQuery	https://jquery.com/
BootStrap	https://getbootstrap.com/

Pages

- index.html
- index1.html
- index2.html
- index3.html

- login.html
- info.html
- info1.html
- sellerinfo.html
- list.html
- emptylist.html
- detail.html
- deny.html

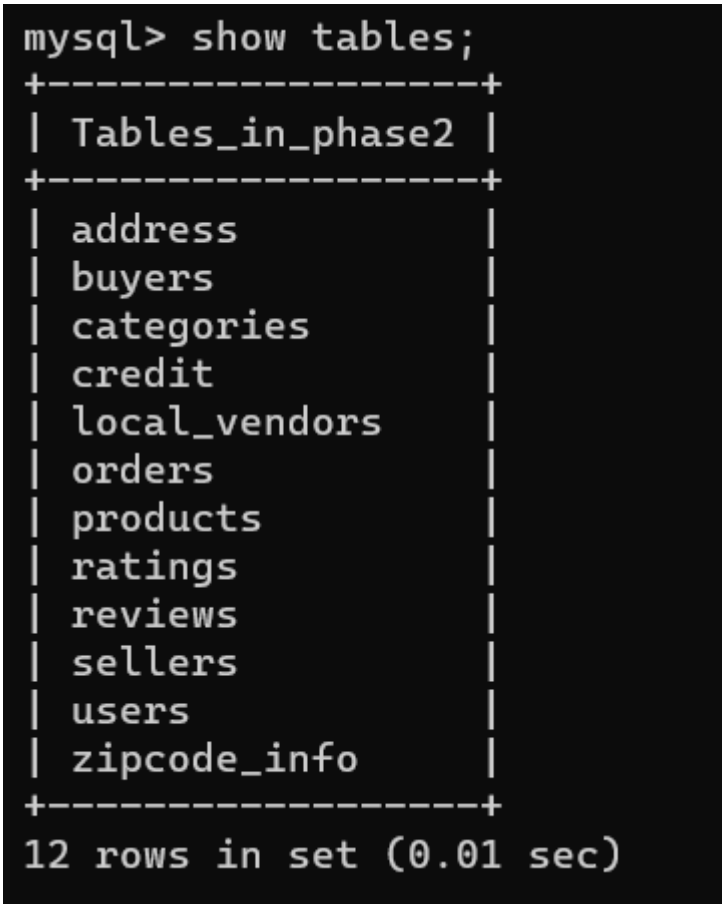
Populating Data

Based on the schemas and table records provided by the instructors, I created 12 tables in total by using MySQL, and load all data files into corresponding tables. The following are some most common codes used in MySQL.

```
# Create database
Create Database [DatabaseName];
# Create table
Create Table [TableName](FieldName1 FieldType1, FieldName2...);
# Checking the column info
Desc [Tablename];
# load data into table from external files
LOAD DATA INFILE [dstn] INTO TABLE [TableName] FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' ESCAPED BY '"'
# add any constraints or modify any specifications
Alter Table [TableName] add ...
```

Besides, I added primary key `PK` and foreign key `FK` appropriately in order to link data together in a better way.

Now, my database looks like this:



Update:

Added v_product and period attribute to the product table to indicate whether a product is listed or not, and when it's unlisted.

Added another table Cart to help implementing shopping cart function. Buyers who add products to shopping cart, their email, and the product info being added will be stored as an entity into this table. | Field | Type | Null | Key | Default | Extra | | ------ | ------ | ------ | ------ | ------ | ------ | | cid | int | NO | PRI | NULL | auto_increment | | buyer_email | varchar(50) | NO | | NULL | | | listing_id | int | NO | | NULL | | | quantity | int | NO | | NULL | | | payment | bigint | NO | | NULL | | | product_name | varchar(150) | NO | | NULL | |

User Log In

Before starting to design the webpage, I linked my flask program with the database I just designed by using pymysql pckg. By entering the info of specific database, it would connect to the program so that I could query any info might be used in this platform.

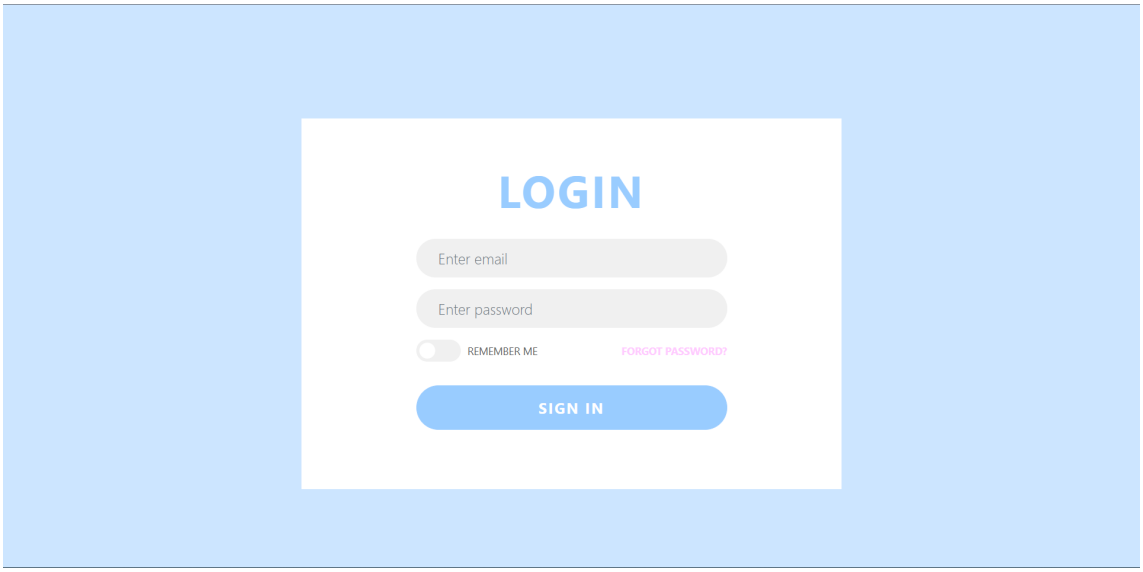
I designed the Login page by using HTML5 and CSS. I imported `bootstrap.min.css` to directly use some styles. The main element here would be the login form, I used post method to collect the email and password being entered, then use MySQL to query out all possible users, once the collected one matched with any user email and password,

then they will be redirected to the home page, otherwise, an error msg will pop out saying they entered wrong email or password.

While entering password, it will be hidden so only the program will check the correctness, others will not be able to see. Furthurmore, I used MD5 coding to encode all passwords, it's way much safer.

After the user successfully logged in, he/she will be directed to the proper index page according to the role he/she is. If the user is only buyer, he/she could do anything except publishing new products. If the user is both buyer and seller, he/she could do pretty much anything. Else, if the user is only seller or only local vendors, he/she could not purchase / add items to shopping cart, or review any orders. Once the platform comfirm the user's identity, pages and functions will be adjusted to fit his/her role automatically.

The Login page would look like this:



Cheking Info

All users could check their personal information. They can click their name on the top right corner at the index page to view their information.

In the information page, they could view their personal information (including email address, name, gender, addresses, etc), products being listed (both valid and invalid) if they are sellers, orders if they are buyers, shopping carts if they are buyers. They are also able to change their password. Once they changed their password, they will be redirected to the login page to log in again. Their new passwords will be automatically stored into the User table in MySQL database. If they are both sellers and buyers, except the buyer information, their ratings and balance will also be displayed in the info page; however, if they are only sellers or local vendors, only ratings and balance will be displayed when browsing personal info.

Besides, for buyers, once they added some products to their shopping cart, those products will be displayed inside the info page in `Shopping Cart` module. They could also remove any products from the shopping cart. Buyers can also view their past

orders, and be able to write comments and rate, but the same product bought by the same buyer can be only reviewed once.

Sellers could find their listed products, even they are not being listed anymore. Those information will be displayed under `My Products` module. They could also unlist products in one click, inside the database, the product will set to be invalid but the record will be kept.

All above features are automically linked to the database, so the user's information will be fetched, and if any updates being made, the altered field will be displayed in database and reflected updated field on the info page.

Category Hierarchy

All products available at Nittany Market are categorized into different categories. Categories include parent and sub categories, mainly depend on how sellers would classify them. All parent and sub categories are shown at index pages, by hovering on the parent categories, users are able to see all sub categories under the parent category. Users need to login first to click on any of the parent/sub category to view products under that. If there is no products listed under that category, the page will tell you there's no product. Otherwise, all filtered products will be displayed.

Once the users see all products listed under the specific category, they can click the product to see the product detail. Detail includes its `picture(If the product has one), title, name, listing_id, description, total quantity, unit price`. While the user is clicking the `Comments & Reviews` button, a window would popped out and show all part reviews regarding the product wrote by past customers.

Important: Users whose roles are seller or local vendors only are not permitted to view product details. But they can still be able to browse through all categories.
#

Publish Product Listing

Users who are sellers or local vendors are able to list new products. There is a orange button `List New Products` at the index page. This button is not used when user is only buyer. Once the user click the button, a list form pops out and they could fill in all blanks. Once they submit the form, the data is being collected and updates are happened to the database. If the seller gives a new category or sub category, database will start to try to insert that in to category table, if failed, error msg will be printed. Then the product information will also be placed into the database and linked to its category.

Once their products are published, they could see it while traversing through the category hierarchy. Sellers can also view they listed products at their info page. They can also choose to unlist the product. If they unlist any products, the product will be invalid inside the database, and the `period` attribute will be changed to the day it became unavailable.

Place Order

Buyers can place order at the product detail page. They could select how many they want to buy, and the total payment will be adjusted automatically. They cannot select negative quantity or any number above total quantity. Once they are ready, they could

click the `Buy` button. A window will be popped out and they need to select which credit card to make the payment, they can also add a new card, and enter the card number, expiration date and cvv. The new card will be inserted into the credit card table. Buyers should also select their delivering address, but they cannot add a new one for now. Once they are ready, click the submit button and the order will be processed. The order detail will be added to the orders table, and buyers can see the order at the info page.

Once purchased, they will stay at the product detail page, and the total quantity will be decremented by how many the buyer bought. If there are 0 left, there will be an alert saying `This product is currently unavailable.` This product will also be marked as invalid, as unlisted, and the period will be modified.

Review & Rating

Buyers at thier info page can see all their past orders at `My Orders` module. They can review those orders, once they clicked the review button, a form pops out and they need to fill out the rating for the seller, and the descriptions to the product.

Worth Attention: The same product bought by the same buyer can only be reviewed once, if there's error message printed out after reviewing, it means the buyer already reviewed this product before.

If no error message comes out, buyers can also see their comments and review at that product's detail page. If there's no review of a product, it will display no records.

Ratings to the seller are also being recorded. Rating scale 1 to 5, if it's 1, it will add `Bad` to the description. If it's 2 or 3, then the description will be `Not Bad`. Else, the description will changed to `Awesome`. Meanwhile, the seller's overall rating will also be changed.

Shopping Cart

In the product detail page, except for buying, buyers can also add the product to shopping cart. It's important for them to select desired quantity before doing so. Once succeeded, they can view results in their shopping cart at the info page under `Shopping Cart` nodule. The quantity of the product will not change, the status maintained. Buyers can also remove the item from shopping cart by cliking remove at info page. The record will be completely deleted from the database so it will not be able to traced back. If the product is sold out or manually unlisted, then it will be marked as invalid inside the database, it will not be displayed in that user's shopping cart.

Searching

For searching, I implemented fuzzy search. If the user has would like to search for an item, once they clicked search, the database started to query out all products in products table which has title or product_name or description like `%{{search input}}%`. If matched, then those items will be displayed in list page; otherwise, empty list page will be displayed.

Test

After running the program, you could either type `127.0.0.1:5000` in any search engines or click the urls in terminal to test.

Conclusion

There are several micro bugs when using the Nittany Market. But the above functions and pages are all implemented and worked. The database is being fixed and ready to use.

References

https://www.w3schools.com/html/default.asp https://www.w3schools.com/css/default.asp
https://www.apple.com https://www.iconfont.cn/
https://www.feednavigator.com/Suppliers/Information-not-available bootstrap.min.css