INFSCI 2710 Database Management Group Project

Database System for E-Commerce

Group members:

Yunpeng Guo, Lin Yang, Renjie Xu

1. Readme

Before running server to test our system, please link the django project to your local mysql database by modifying ‘settings.py’. Also, a sql file ‘project.sql’ has been derived for you to create the testing relations, which includes 1 superuser object (system manager) ‘anton’, 1 customer object “Anton_test”, and 3 product objects (project1, project2 and project3). These three product objects are necessary for the system’s working normally, and you may create more superuser objects and customer objects to test the system if you like.

2. Updated content

Based on what have been shown during the demo, our group adds some additional functions to the system.

Firstly, the purchase mechanism has been realized. Three products are available to be bought. After login, the customer can click the product name (e.g., Product1), or the red button ‘add to card’ to process the purchase (shown by figure 2.1). Then the webpage will be directed to the corresponding product information page (shown by figure 2.2). Product name, price, and current inventory number will be provided. The customer is prompted to input the number of the product he/she wants to buy. A javascript function is bounded with the input widget to check whether the customer inputs a number greater than the current inventory amount. If so, the submit button will be disabled and an error message will be shown.

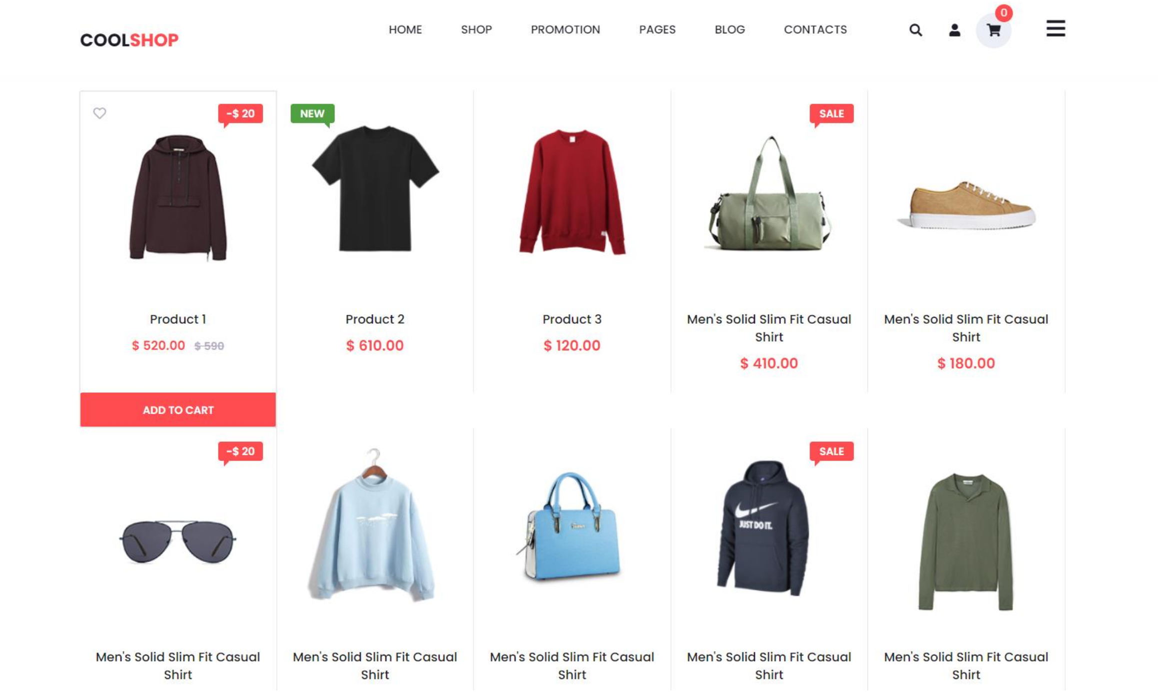
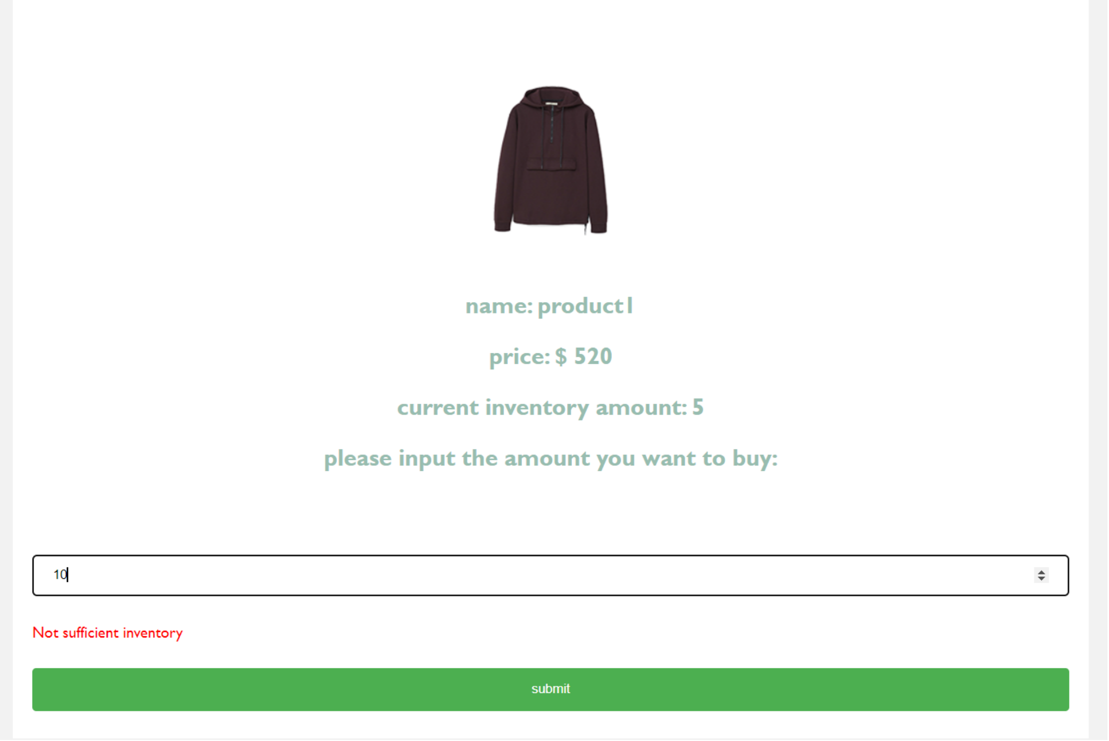

COOLSHOP
HOME
SHOP
PROMOTION
PAGES
BLOG
CONTACTS
0
-$ 20
NEW
SALE
Product 1
$ 520.00 $ 590
ADD TO CART
Product 2
$ 610.00
Product 3
$ 120.00
Men's Solid Slim Fit Casual Shirt
$ 410.00
Men's Solid Slim Fit Casual Shirt
$ 180.00
-$ 20
SALE
Men's Solid Slim Fit Casual Shirt
Men's Solid Slim Fit Casual Shirt
Men's Solid Slim Fit Casual Shirt
Men's Solid Slim Fit Casual Shirt
Men's Solid Slim Fit Casual Shirt

Figure 2.1 Purchase



name: product1

price: $ 520

current inventory amount: 5

please input the amount you want to buy:

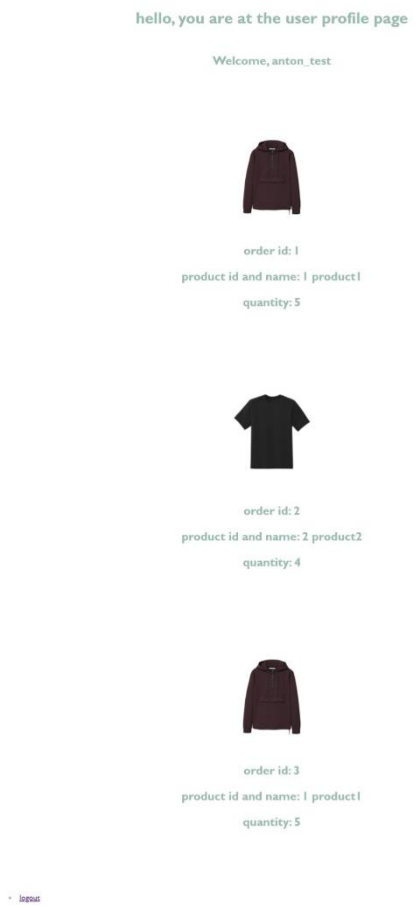
10

Not sufficient inventory

Figure 2.2 Product information page

Therefore, after a product has been sold out, i.e., the ‘current inventory amount’ is 0, the customer will be unable to submit the purchase query.

After successfully purchasing a product, the page will be redirected to the main page and the customer can continue to shop. Also, the customer can track the history order in the user profile (Figure 2.3). The order id, product information and product quantity will be shown. To realize this function, the Transaction model is updated to have foreign key relationship with the User, Customer and Product models.



hello, you are at the user profile page

Welcome, anton_test

order id: 1

product id and name: 1 product1

quantity: 5

order id: 2

product id and name: 2 product2

quantity: 4

order id: 3

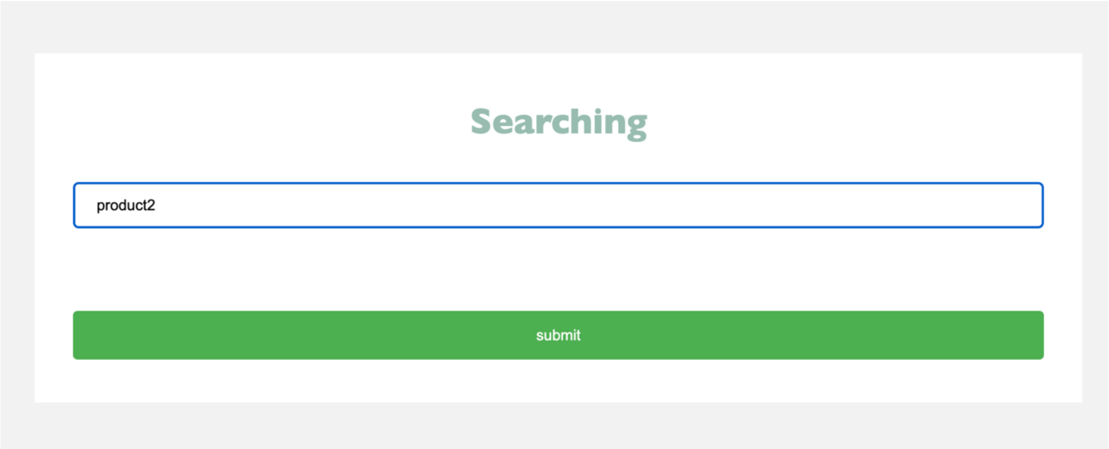
product id and name: 1 product1

quantity: 5

- logout

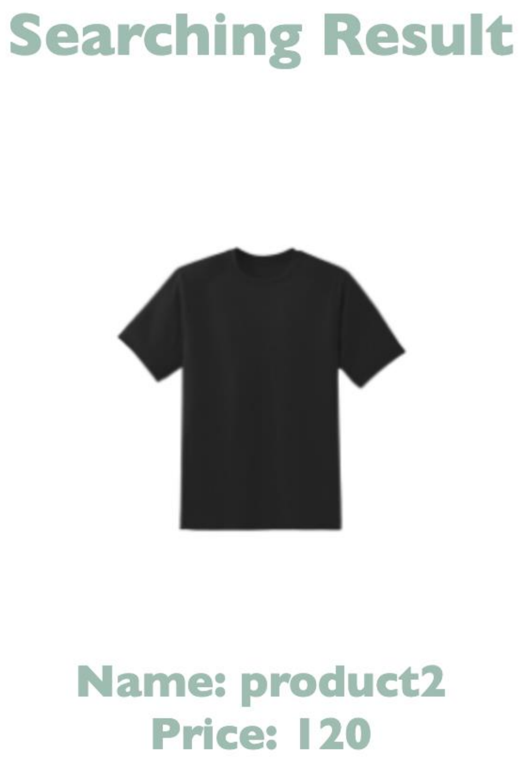
Figure 2.3 Updated user profile

At the same time, searching mechanism is enabled. When you click on the search button on the top right of the page, the searching page will appear. You just need to input the product name you want to search into the text field. The function is realized by matching user's input text with product's name in views.py and returns the searched product page (Figure 2.5) when input correctly (Figure 2.4) and the black page (Figure 2.7) when input incorrectly (Figure 2.6).


Searching

product2

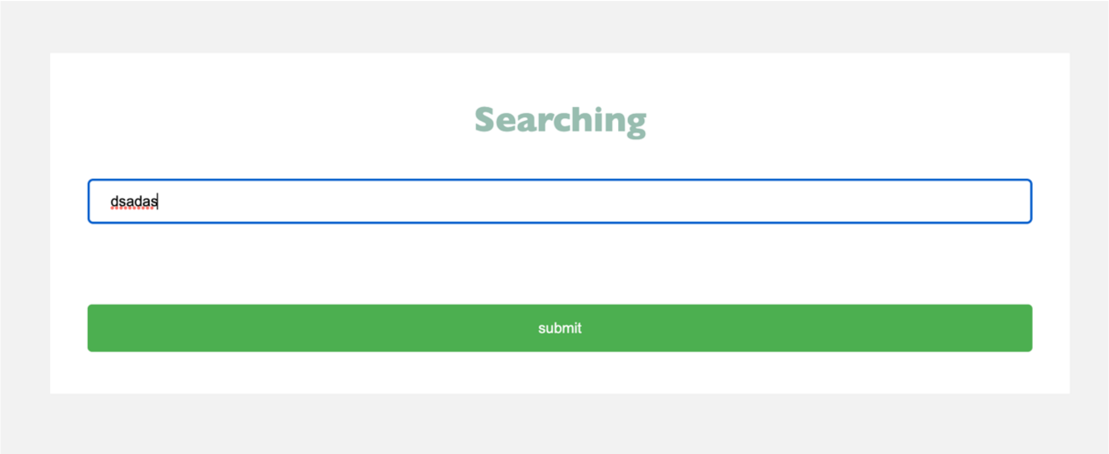
Figure 2.4 Input correctly



Searching Result

Name: product2
Price: 120

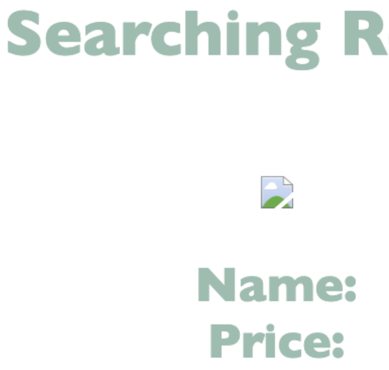
Figure 2.5 Correct result



Searching

dsadas

Figure 2.6 Input incorrectly

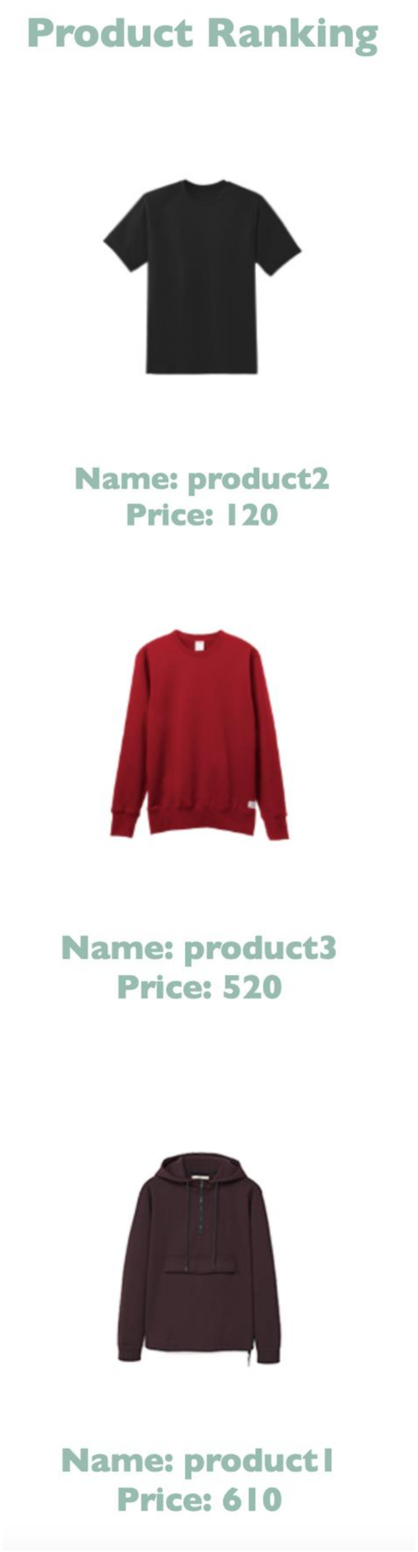


Searching Result

Name:
Price:

Figure 2.7 Incorrect result

Furthermore, ranking products based on price is realized, when you click on shop now button in the home page, it will directly lead you to the ranking page. The page contains products ranked by their price. For example, if we have three products named product 1 (price: 610 USD), product 2 (price: 120 USD) and product 3 (price: 520 USD), their ranking is shown in Figure 2.7. The function is achieved by using sort functions in the views.py to get an ascending order of products and display them in order in the html file. Also, as shown in Figure 2.8, an aggregation function is used to show the highest price of the products.



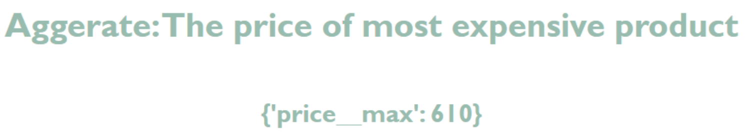
Product Ranking

Name: product2
Price: 120

Name: product3
Price: 520

Name: product1
Price: 610

Figure 2.7



Aggerate:The price of most expensive product

```
{'price__max': 610}
```

Figure 2.8

3. Additional note

After discussion, all the group members agree that we request to deduct Renjie Xu’s final mark of this project to 15 percent lower than the overall group score of this project, and Yunpeng Guo’s and Lin Yang’s scores should be increased to 15 percent higher than the overall group score. This decision is made based on the actual workloads of each group member.