

Phase 5 of your Database Application Develop a PHP or Java Application

Due to the amount of operations and data held within the database and code itself, I have decided not to use either PHP or JAVA as a front-end and instead when for Python using Django Framework for both front-end and back-end work. It has built in security features to prevents Cross site scripting and SQL injections within fields. It also minimizes boilerplate code and overall code. The entire project runs on a server and is accessible via URL.

Available Items for sale
Individual Orders

site product id: 1								
Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
40	6	Seller_name_6	2	Manufacturer_name_18	Product_name_40	40	98.99	Product_description_40
45	9	Seller_name_9	3	Manufacturer_name_16	Product_name_45	45	98.99	Product_description_45
48	9	Seller_name_9	3	Manufacturer_name_17	Product_name_48	48	12999.99	Product_description_48

site product id: 2								
Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
25	6	Seller_name_6	2	Manufacturer_name_19	Product_name_25	25	12999.99	Product_description_25
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	12999.99	Product_description_34
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46
50	6	Seller_name_6	2	Manufacturer_name_16	Product_name_50	50	98.99	Product_description_50

site product id: 3								
Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
24	8	Seller_name_8	3	Manufacturer_name_19	Product_name_24	24	98.99	Product_description_24
30	10	Seller_name_10	4	Manufacturer_name_20	Product_name_30	30	399.98	Product_description_30
31	7	Seller_name_7	4	Manufacturer_name_20	Product_name_31	31	98.99	Product_description_31
49	9	Seller_name_9	3	Manufacturer_name_20	Product_name_49	49	12999.99	Product_description_49

This is the home page, it runs the MySQL procedure

'cis_363_project.get seller product listing remaining' via python code.

```
for i in list_site_product_ids:
    # Calling procedure get_seller_product_listing_remaining in MySQL
    with connection.cursor() as cursor:
        cursor.callproc('cis_363_project.get_seller_product_listing_remaining', [i.site_product_id])
        seller_product_listing_remaining = cursor.fetchall()

        list_dict_seller_product_listing_remaining.append({i.site_product_id: seller_product_listing_remaining})
```

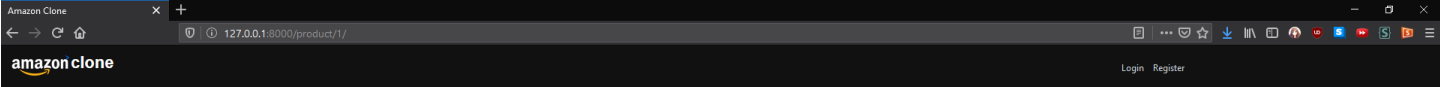
The MySQL procedure

```
DROP PROCEDURE IF EXISTS cis_363_project.get_seller_product_listing_remaining;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_seller_product_listing_remaining(IN given_site_product_id INT)
BEGIN
    SELECT seller_product_listing.Seller_product_listing_ID, seller.id as Seller_ID, seller.Seller_name as Seller_name, seller.Seller_rating as Seller_rating ,
    Manufacturer_name, Product_name, product.Product_ID,
    Product_pricing, Product_description
    FROM seller_product_listing
    LEFT JOIN product ON seller_product_listing.Product_ID = product.Product_ID
    LEFT JOIN item_order ON seller_product_listing.Seller_product_listing_ID = item_order.Seller_product_listing_ID
    JOIN seller ON seller_product_listing.id = seller.id
    JOIN manufacturer ON product.id = manufacturer.id
    WHERE seller_product_listing.Site_product_ID = given_site_product_id AND Customer_order_ID IS null;
END//
DELIMITER ;
```

The python code loops through the existing rows in seller_product_listing in the DB. In total there are 50 products on the website, however not all of them are shown on the home page because some of these items are in orders by a Customers. Here are some of the products on the website.

Seller_product_listing_ID	id	Product_ID	Site_product_ID	Product_pricing	Product_description	Product_image
29	8	29	2	98.99	Product_description_29	NULL
30	10	30	3	399.98	Product_description_30	NULL
31	7	31	3	98.99	Product_description_31	NULL
32	9	32	4	12999.99	Product_description_32	NULL
33	6	33	5	98.99	Product_description_33	NULL
34	7	34	2	12999.99	Product_description_34	NULL
35	10	35	4	98.99	Product_description_35	NULL
36	9	36	4	399.98	Product_description_36	NULL
37	10	37	5	399.98	Product_description_37	NULL
38	7	38	5	12999.99	Product_description_38	NULL
39	10	39	4	399.98	Product_description_39	NULL
40	6	40	1	98.99	Product_description_40	NULL
41	10	41	2	399.98	Product_description_41	NULL
42	8	42	2	399.98	Product_description_42	NULL
43	8	43	5	98.99	Product_description_43	NULL
44	7	44	5	12999.99	Product_description_44	NULL
45	9	45	1	98.99	Product_description_45	NULL
46	8	46	2	399.98	Product_description_46	NULL
47	6	47	5	98.99	Product_description_47	NULL
48	9	48	1	12999.99	Product_description_48	NULL
49	9	49	3	12999.99	Product_description_49	NULL
50	6	50	2	98.99	Product_description_50	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

On the home page you can see a site_product_id above the groups of items, that is the product page id for the those given items, they are also links to their corresponding product page. Here are some example product pages (notice that the URL has changed.)



PRODUCT PAGE FOR site_product_id: 1

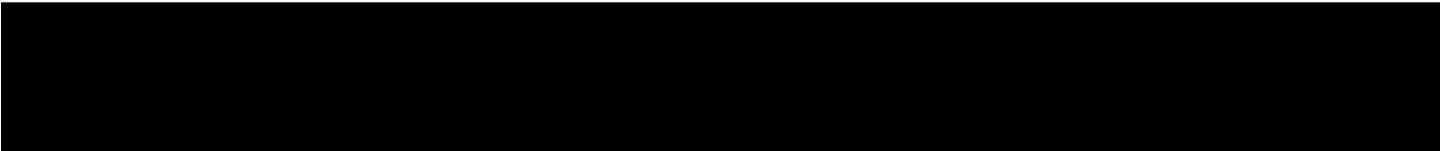
Product average star rating: 0
Amount of products left: 3

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
40	6	Seller_name_6	2	Manufacturer_name_18	Product_name_40	40	98.99	Product_description_40
45	9	Seller_name_9	3	Manufacturer_name_16	Product_name_45	45	98.99	Product_description_45
48	9	Seller_name_9	3	Manufacturer_name_17	Product_name_48	48	12999.99	Product_description_48

COMMENTS

PRODUCT TAGS

Tag_Title_1



PRODUCT PAGE FOR site_product_id: 2

Product average star rating: 3
Amount of products left: 7

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
25	6	Seller_name_6	2	Manufacturer_name_19	Product_name_25	25	12999.99	Product_description_25
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	12999.99	Product_description_34
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46
50	6	Seller_name_6	2	Manufacturer_name_16	Product_name_50	50	98.99	Product_description_50

COMMENTS

Username_6
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 4

Username_7
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 2

PRODUCT TAGS

Tag_Title_2

PRODUCT PAGE FOR site_product_id: 5

Product average star rating: 2
Amount of products left: 10

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
21	10	Seller_name_10	4	Manufacturer_name_19	Product_name_21	21	399.98	Product_description_21
22	8	Seller_name_8	3	Manufacturer_name_19	Product_name_22	22	12999.99	Product_description_22
23	10	Seller_name_10	4	Manufacturer_name_20	Product_name_23	23	98.99	Product_description_23
27	10	Seller_name_10	4	Manufacturer_name_16	Product_name_27	27	12999.99	Product_description_27
33	6	Seller_name_6	2	Manufacturer_name_20	Product_name_33	33	98.99	Product_description_33
37	10	Seller_name_10	4	Manufacturer_name_20	Product_name_37	37	399.98	Product_description_37
38	7	Seller_name_7	4	Manufacturer_name_16	Product_name_38	38	12999.99	Product_description_38
43	8	Seller_name_8	3	Manufacturer_name_19	Product_name_43	43	98.99	Product_description_43
44	7	Seller_name_7	4	Manufacturer_name_16	Product_name_44	44	12999.99	Product_description_44
47	6	Seller_name_6	2	Manufacturer_name_19	Product_name_47	47	98.99	Product_description_47

COMMENTS

Username_1
May 21, 2020, 9 a.m.
Comment Very cool!
Rating: 1

Username_2
May 21, 2020, 9 a.m.
Comment Very cool!
Rating: 3

Username_5
May 21, 2020, 9 a.m.
Comment Very cool!

On each of these pages they call 2 queries automated by Django and 3 functions in python that call the procedures in MySQL. The 2 automated queries query from the tables `cis_363_project.site_product`; and `cis_363_project.site_product_comment`; with the condition that the `site_product_id` (which is in the URL) be in the rows. The information from these queries is then sent to a `site_product` template html file and then modify the line “Product average star rating” and the “COMMENTS” section.

```
row_SiteProduct = get_object_or_404(SiteProduct, site_product_id=site_product_id)
row_SiteProductComment = SiteProductComment.objects.filter(site_product_id=site_product_id)
```

The 3 `cursor.callproc()` functions called here in python correspond to calling the procedures in MySQL that get the `site_product_id`'s tags, the amount of items available that can order, and the products that you can order, respectively.

```
# Calling procedure get_site_product_tags in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.get_site_product_tags', [row_SiteProduct.site_product_id])
    row_ProductTagPair = cursor.fetchone()

# Calling procedure get_site_product_amount_products in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.get_site_product_amount_products', [row_SiteProduct.site_product_id])
    amount_products_site_product = cursor.fetchone()

# Calling procedure get_seller_product_listing_remaining in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.get_seller_product_listing_remaining', [row_SiteProduct.site_product_id])
    seller_product_listing_remaining = cursor.fetchall()
```

Here are the procedures in MySQL

```
DROP PROCEDURE IF EXISTS cis_363_project.get_site_product_amount_products;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_site_product_amount_products(IN given_site_product_id INT)
BEGIN
    SELECT count(seller_product_listing.Seller_product_listing_ID) - count(Customer_order_ID) as amount_left
    FROM seller_product_listing
    LEFT JOIN item_order ON seller_product_listing.seller_product_listing_id = item_order.seller_product_listing_id
    where site_product_id = given_site_product_id;
END//
DELIMITER ;

DROP PROCEDURE IF EXISTS cis_363_project.get_site_product_tags;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_site_product_tags(IN given_site_product_id INT)
BEGIN
    SELECT tag_information.Tag_Title as tag_title FROM product_tag_pair
    JOIN site_product ON site_product.Site_product_ID = product_tag_pair.Site_product_ID
    JOIN tag_information ON tag_information.Product_tag_ID = product_tag_pair.Product_tag_ID
    WHERE product_tag_pair.Site_product_ID = given_site_product_id;
END//
DELIMITER ;
```

**Note that the procedure `cis_363_project.get_seller_product_listing_remaining`; is already displayed above.*

Notice that we are not logged in. Let's login as one of the generated usernames. Unfortunately, Django has a system set up to handle the user logins already built in, so my DB had to be modified to support those changes (which was basically a new table User table called auth_user in the DB with a one to one relationship to my existing user table) and thus I had to generate a part of the users from a python script while the rest was generated in the original SQL script. A neat benefit was that the passwords are hashed.

The python script

```
list_given = [['Username_1', 'Password', 'first_name_1', 'last_name_1', 'Email_1@email.com'],
['Username_2', 'Password', 'first_name_2', 'last_name_2', 'Email_2@email.com'],
['Username_3', 'Password', 'first_name_3', 'last_name_3', 'Email_3@email.com'],
['Username_4', 'Password', 'first_name_4', 'last_name_4', 'Email_4@email.com'],
['Username_5', 'Password', 'first_name_5', 'last_name_5', 'Email_5@email.com'],
['Username_6', 'Password', 'first_name_6', 'last_name_6', 'Email_6@email.com'],
['Username_7', 'Password', 'first_name_7', 'last_name_7', 'Email_7@email.com'],
['Username_8', 'Password', 'first_name_8', 'last_name_8', 'Email_8@email.com'],
['Username_9', 'Password', 'first_name_9', 'last_name_9', 'Email_9@email.com'],
['Username_10', 'Password', 'first_name_10', 'last_name_10', 'Email_10@email.com'],
['Username_11', 'Password', 'first_name_11', 'last_name_11', 'Email_11@email.com'],
['Username_12', 'Password', 'first_name_12', 'last_name_12', 'Email_12@email.com'],
['Username_13', 'Password', 'first_name_13', 'last_name_13', 'Email_13@email.com'],
['Username_14', 'Password', 'first_name_14', 'last_name_14', 'Email_14@email.com'],
['Username_15', 'Password', 'first_name_15', 'last_name_15', 'Email_15@email.com'],
['Username_16', 'Password', 'first_name_16', 'last_name_16', 'Email_16@email.com'],
['Username_17', 'Password', 'first_name_17', 'last_name_17', 'Email_17@email.com'],
['Username_18', 'Password', 'first_name_18', 'last_name_18', 'Email_18@email.com'],
['Username_19', 'Password', 'first_name_19', 'last_name_19', 'Email_19@email.com'],
['Username_20', 'Password', 'first_name_20', 'last_name_20', 'Email_20@email.com']]

for i in list_given:
    x = User.objects.create_user(username=i[0], password=i[1], first_name=i[2], last_name=i[3], email=i[4])
    x.save()
    print("Ok")
```

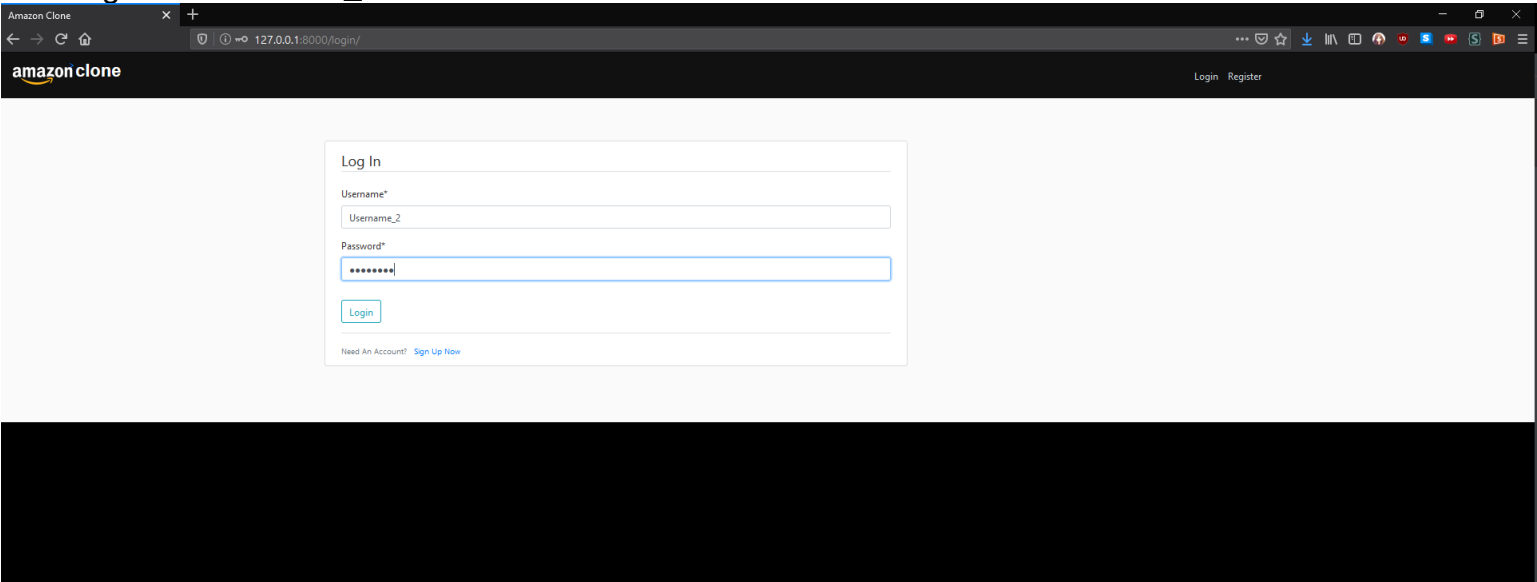
The Table auth_user in MySQL (I also manually set Username_1 as an Admin for later)

id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined
1	pbkdf2_sha256\$180000\$jr4330amfzD\$OPTED...	NULL	1	Username_1	first_name_1	last_name_1	Email_1@email.com	1	1	2020-05-21 22:59:58.768823
2	pbkdf2_sha256\$180000\$QJTRCb5dK3\$RfmnQ/...	NULL	0	Username_2	first_name_2	last_name_2	Email_2@email.com	0	1	2020-05-21 22:59:58.876916
3	pbkdf2_sha256\$180000\$3YqAZrtpIqS\$Aw5fjS...	NULL	0	Username_3	first_name_3	last_name_3	Email_3@email.com	0	1	2020-05-21 22:59:58.963989
4	pbkdf2_sha256\$180000\$1r00VgltB8\$YQRPi...	NULL	0	Username_4	first_name_4	last_name_4	Email_4@email.com	0	1	2020-05-21 22:59:59.051063
5	pbkdf2_sha256\$180000\$g\$SoThuwNhgT\$ikq0...	NULL	0	Username_5	first_name_5	last_name_5	Email_5@email.com	0	1	2020-05-21 22:59:59.137136
6	pbkdf2_sha256\$180000\$TbkmOO882q50\$SmR...	NULL	0	Username_6	first_name_6	last_name_6	Email_6@email.com	0	1	2020-05-21 22:59:59.223209
7	pbkdf2_sha256\$180000\$zhR93Aog149dSqDWI...	NULL	0	Username_7	first_name_7	last_name_7	Email_7@email.com	0	1	2020-05-21 22:59:59.313286
8	pbkdf2_sha256\$180000\$Nefz7EPLuWk\$DuC2v...	NULL	0	Username_8	first_name_8	last_name_8	Email_8@email.com	0	1	2020-05-21 22:59:59.398359
9	pbkdf2_sha256\$180000\$M9PondXMTJ\$D\$8MW...	NULL	0	Username_9	first_name_9	last_name_9	Email_9@email.com	0	1	2020-05-21 22:59:59.483431
10	pbkdf2_sha256\$180000\$PG68peVqScXv\$VNE...	NULL	0	Username_10	first_name_10	last_name_10	Email_10@email.com	0	1	2020-05-21 22:59:59.569505
11	pbkdf2_sha256\$180000\$nxod7P\$ecpAB\$N5Zt...	NULL	0	Username_11	first_name_11	last_name_11	Email_11@email.com	0	1	2020-05-21 22:59:59.656579
12	pbkdf2_sha256\$180000\$7X7H\$EW6AS2df\$R09...	NULL	0	Username_12	first_name_12	last_name_12	Email_12@email.com	0	1	2020-05-21 22:59:59.742652
13	pbkdf2_sha256\$180000\$z2Vvm5UPWCD\$E+br...	NULL	0	Username_13	first_name_13	last_name_13	Email_13@email.com	0	1	2020-05-21 22:59:59.828726
14	pbkdf2_sha256\$180000\$0TQLXagakpyV\$8TBID...	NULL	0	Username_14	first_name_14	last_name_14	Email_14@email.com	0	1	2020-05-21 22:59:59.917802
15	pbkdf2_sha256\$180000\$nhH4MCo8G6p\$Tnyk...	NULL	0	Username_15	first_name_15	last_name_15	Email_15@email.com	0	1	2020-05-21 23:00:00.004876
16	pbkdf2_sha256\$180000\$JQTrm8SLya5\$98ZT...	NULL	0	Username_16	first_name_16	last_name_16	Email_16@email.com	0	1	2020-05-21 23:00:00.090949
17	pbkdf2_sha256\$180000\$wvSX98weGbq\$0v08...	NULL	0	Username_17	first_name_17	last_name_17	Email_17@email.com	0	1	2020-05-21 23:00:00.179025
18	pbkdf2_sha256\$180000\$YXZ5bKaxJTU\$Ndlajh...	NULL	0	Username_18	first_name_18	last_name_18	Email_18@email.com	0	1	2020-05-21 23:00:00.266100
19	pbkdf2_sha256\$180000\$NknMealN2EQu8bNgI...	NULL	0	Username_19	first_name_19	last_name_19	Email_19@email.com	0	1	2020-05-21 23:00:00.353173
20	pbkdf2_sha256\$180000\$FgBEnTw3SCBx\$frfY...	NULL	0	Username_20	first_name_20	last_name_20	Email_20@email.com	0	1	2020-05-21 23:00:00.438245
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

My modified user Table in MySQL to reflect the one to one relationship

id	Address_1	Address_2	City	State_Province	Zip	Phone_1	Phone_2
1	Address_1_1	NULL	City_1	State_Province_1	19489	7401026360	NULL
2	Address_1_2	NULL	City_2	State_Province_2	96575	8382456836	NULL
3	Address_1_3	NULL	City_3	State_Province_3	65454	9853833303	NULL
4	Address_1_4	NULL	City_4	State_Province_4	39430	9242000634	NULL
5	Address_1_5	NULL	City_5	State_Province_5	98885	1651531962	NULL
6	Address_1_6	NULL	City_6	State_Province_6	72537	9977513021	NULL
7	Address_1_7	NULL	City_7	State_Province_7	32090	6951986730	NULL
8	Address_1_8	NULL	City_8	State_Province_8	27484	8750035058	NULL
9	Address_1_9	NULL	City_9	State_Province_9	67164	5870271011	NULL
10	Address_1_10	NULL	City_10	State_Province_10	89909	9086317834	NULL
11	Address_1_11	NULL	City_11	State_Province_11	59573	5614407036	NULL
12	Address_1_12	NULL	City_12	State_Province_12	20235	7192322326	NULL
13	Address_1_13	NULL	City_13	State_Province_13	77272	2382990994	NULL
14	Address_1_14	NULL	City_14	State_Province_14	72949	2472659543	NULL
15	Address_1_15	NULL	City_15	State_Province_15	73062	4972143439	NULL
16	Address_1_16	NULL	City_16	State_Province_16	45984	8984420365	NULL
17	Address_1_17	NULL	City_17	State_Province_17	48104	5501173776	NULL
18	Address_1_18	NULL	City_18	State_Province_18	50206	4745292483	NULL
19	Address_1_19	NULL	City_19	State_Province_19	13755	3700465468	NULL
20	Address_1_20	NULL	City_20	State_Province_20	48620	4767453301	NULL

Let's login as *Username_2*



Notice that we are redirected to the home page and we have a “Welcome Username_2” at the top right and a **Profile** button and **Logout** button.

Amazon Clone

127.0.0.1:8000

Welcome Username_2 Profile Logout

Available Items for sale

Individual Orders

site product id: 1

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
40	6	Seller_name_6	2	Manufacturer_name_18	Product_name_40	40	98.99	Product_description_40
45	9	Seller_name_9	3	Manufacturer_name_16	Product_name_45	45	98.99	Product_description_45
48	9	Seller_name_9	3	Manufacturer_name_17	Product_name_48	48	12999.99	Product_description_48

site product id: 2

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
25	6	Seller_name_6	2	Manufacturer_name_19	Product_name_25	25	12999.99	Product_description_25
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	12999.99	Product_description_34
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46
50	6	Seller_name_6	2	Manufacturer_name_16	Product_name_50	50	98.99	Product_description_50

site product id: 3

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
24	8	Seller_name_8	3	Manufacturer_name_19	Product_name_24	24	98.99	Product_description_24
30	10	Seller_name_10	4	Manufacturer_name_20	Product_name_30	30	399.98	Product_description_30
31	7	Seller_name_7	4	Manufacturer_name_20	Product_name_31	31	98.99	Product_description_31
49	9	Seller_name_9	3	Manufacturer_name_20	Product_name_49	49	12999.99	Product_description_49

Now let's go to site_product_id: 2

Amazon Clone

127.0.0.1:8000/product/2/

80%

Welcome Username_2 Profile Logout

PRODUCT PAGE FOR site_product_id: 2

Product average star rating: 3
Amount of products left: 7

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
25	6	Seller_name_5	2	Manufacturer_name_19	Product_name_25	25	1299.99	Product_description_25	Order
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	88.99	Product_description_29	Order
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	1299.99	Product_description_34	Order
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41	Order
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42	Order
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46	Order
50	6	Seller_name_5	2	Manufacturer_name_16	Product_name_50	50	88.99	Product_description_50	Order

COMMENTS

Username_6
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 4

Username_7
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 2

Make Comment

Comments

Product rating

1

Post

Since we are logged in, we can comment and order products. I will comment and give a rating of 1, this will be added to the page via Django's `Model object` that will automatically insert into the table `cis_363_project.site_product_comment`; and "Product average star rating: 3" should change to "Product average star rating: 2" because of the MySQL trigger that I set up when I created the DB.

Make Comment

Comments

THIS PRODUCT SUCKS!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Product rating

1

Post

The MySQL trigger

```
DROP TRIGGER IF EXISTS cis_363_project.trigger_site_product_rating_update;
# Update site_product.Product_rating_avg when Site_product_comment row has been inserted
DELIMITER //
CREATE TRIGGER cis_363_project.trigger_site_product_rating_update
AFTER INSERT ON cis_363_project.Site_product_comment
FOR EACH ROW
BEGIN
    SET @average_rating = (SELECT AVG(Product_rating) FROM Site_product_comment WHERE Site_product_comment.Site_product_ID = NEW.Site_product_ID);

    UPDATE site_product
    SET site_product.Product_rating_avg = @average_rating
    WHERE site_product.Site_product_ID = NEW.Site_product_ID;
END //
DELIMITER ;
```

The result of commenting (rating changed, and new comment added)

Amazon Clone

127.0.0.1:8000/product/2/

Welcome Username_2 Profile Logout

PRODUCT PAGE FOR site_product_id: 2

Product average star rating: 2

Amount of products left: 7

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
25	8	Seller_name_5	2	Manufacturer_name_19	Product_name_25	25	12999.99	Product_description_25	Order
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	88.99	Product_description_29	Order
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	12999.99	Product_description_34	Order
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41	Order
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42	Order
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46	Order
50	8	Seller_name_8	2	Manufacturer_name_16	Product_name_50	50	88.99	Product_description_50	Order

COMMENTS

Username_6
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 4

Username_7
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 2

Username_2
May 21, 2020, 5:19 p.m.
Comment THIS PRODUCT SUCKS!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Rating: 1

Make Comment

Comments

Here is what's in the table cis_363_project.site_product_comment;

	Site_product_comment_ID	id	Site_product_ID	Datetime_created	Datetime_modified	Comments	Images	Product_rating
▶	1	1	5	2020-05-21 16:00:12	NULL	Very cool!	NULL	1
	2	2	5	2020-05-21 16:00:12	NULL	Very cool!	NULL	3
	3	3	3	2020-05-21 16:00:12	NULL	Very cool!	NULL	3
	4	4	4	2020-05-21 16:00:12	NULL	Very cool!	NULL	4
	5	5	5	2020-05-21 16:00:12	NULL	Very cool!	NULL	1
	6	6	2	2020-05-21 16:00:12	NULL	Nice!	NULL	4
	7	7	2	2020-05-21 16:00:12	NULL	Nice!	NULL	2
	8	8	5	2020-05-21 16:00:12	NULL	Very cool!	NULL	2
	9	9	4	2020-05-21 16:00:12	NULL	Looks Good!	NULL	2
	10	10	4	2020-05-21 16:00:12	NULL	Looks Good!	NULL	5
	11	11	3	2020-05-21 16:00:12	NULL	Nice!	NULL	3
	12	2	2	2020-05-22 00:19:52	NULL	THIS PRODUCT SUCKS!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!	NULL	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The new comment added to the table.

Now, lets order Seller Product Listing IDs 25, 34, and 50 from the page.

Amazon Clone

127.0.0.1:8000/product/2/

Welcome Username_2

Profile

 Logout

PRODUCT PAGE FOR site_product_id: 2

Product average star rating: 2
Amount of products left: 4

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29	<div>Order</div>
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41	<div>Order</div>
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42	<div>Order</div>
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46	<div>Order</div>

COMMENTS

Username_6
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 4

Username_7
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 2

Username_2
May 21, 2020, 5:19 p.m.
Comment THIS PRODUCT SUCKS!!!!!!!!!!!!!!!!!!!!!!
Rating: 1

Make Comment

The site automatically reflects the changes and now we can see our orders in our profile.

Clicking on the profile button

Amazon Clone

127.0.0.1:8000/profile/

Welcome Username_2

Profile

 Logout

Username_2
Email_2@email.com
Prime Member: True
Address 1: Address_1_2
Address 2: None
City: City_2
State/Province: State_Province_2
Zip: 96575
Phone 1: 8382456836
Phone 2: None

Create new Order

Your orders

Order ID	Amount of items in order	Order Total
2	4	26197.96

Individual Orders

Order ID: 2

Delete

Seller product listing ID	Name	Price	Description	Seller	Manufacturer	
2	Product_name_2	98.99	Product_description_2	Seller_name_10	Manufacturer_name_17	<div>Remove</div>
25	Product_name_25	12999.99	Product_description_25	Seller_name_6	Manufacturer_name_19	<div>Remove</div>
34	Product_name_34	12999.99	Product_description_34	Seller_name_7	Manufacturer_name_16	<div>Remove</div>
50	Product_name_50	98.99	Product_description_50	Seller_name_6	Manufacturer_name_16	<div>Remove</div>

Here is the function for the **Order** button in python that calls its corresponding procedure in MySQL.

```
# Calling procedure add_item_to_order_latest in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.add_item_to_order_latest', [request.user.id, seller_product_listing_id])

return redirect('website:site_product', site_product_id)

DROP PROCEDURE IF EXISTS cis_363_project.add_item_to_order_latest;
DELIMITER //

CREATE PROCEDURE cis_363_project.add_item_to_order_latest(IN id_given INT, IN Seller_product_listing_ID_given INT)
BEGIN
    IF (SELECT count(id) FROM customer_order WHERE customer_order.id = id_given) = 0 THEN
        CALL cis_363_project.create_customer_order(id_given);
    END IF;

    INSERT INTO item_order(Customer_order_ID, Seller_product_listing_ID) VALUES
    ((SELECT max(Customer_order_ID) FROM customer_order WHERE customer_order.id = id_given), Seller_product_listing_ID_given);
END//
DELIMITER ;
```

On the link profile, we see information about the user and the user's orders. The information on this page is ONLY accessible to the user logged in even though the URL is the same for every user, we will see this later on for other users and for special users such as Sellers who get a different layout (currently we are a Customer).

On the profile page, it calls 3 procedures and 1 Django automated query. The automated query is on the line with `User.objects.get(pk=user_current.id)` where `user_current.id` is based on the logged in user. This is used to get the information about the user such as their address and phone number.

```
# Calling procedure is_prime in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.is_prime', [user_current.id])
    prime_member = cursor.fetchone()

# Calling procedure get_user_all_orders in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.get_user_orders_all', [user_current.id])
    user_all_orders = cursor.fetchall()

list_dict_user_orders_all = []

for order_id, customer_order_id, sum_total in user_all_orders:
    # Calling procedure get_items_in_customer_order_id in MySQL
    with connection.cursor() as cursor:
        cursor.callproc('cis_363_project.get_items_in_customer_order_id', [order_id])
        order = cursor.fetchall()

        list_dict_user_orders_all.append({order_id: order})

context = {
    'is_customer': True,
    'user_all_orders': user_all_orders,
    'user_all_orders_headers': ("Order ID", "Amount of items in order", "Order Total"),
    'list_dict_user_orders_all': list_dict_user_orders_all,
    'list_user_orders_all_headers': ("Seller product listing ID", "Name", "Price", "Description",
                                     "Seller", "Manufacturer"),
    'prime_member': True if prime_member else False,
    'user_profile': User.objects.get(pk=user_current.id)
}
```

Here are the corresponding procedures in MySQL for the profile page.

```
# Get an Order's items
DROP PROCEDURE IF EXISTS cis_363_project.get_items_in_customer_order_id;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_items_in_customer_order_id(IN given_customer_order_id INT)
BEGIN
    SELECT seller_product_listing.Seller_product_listing_ID, product.Product_name, seller_product_listing.Product_pricing, seller_product_listing.Product_description,
    seller.Seller_name, manufacturer.Manufacturer_name
    FROM customer_order
    JOIN item_order ON item_order.Customer_order_ID = customer_order.Customer_order_ID
    JOIN seller_product_listing ON seller_product_listing.Seller_product_listing_ID = item_order.Seller_product_listing_ID
    JOIN seller ON seller.id = seller_product_listing.id
    JOIN product ON product.product_ID = seller_product_listing.product_ID
    JOIN manufacturer ON manufacturer.id = product.id
    WHERE customer_order.Customer_order_ID = given_customer_order_id;
END //
DELIMITER ;

# Get a user's orders (Meant for the user to use)
DROP PROCEDURE IF EXISTS cis_363_project.get_user_orders_all;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_user_orders_all(IN given_id INT)
BEGIN
    SELECT
    customer_order.Customer_order_id as "Order ID",
    count(customer_order.Customer_order_id) as "Amount of items in order",
    sum(seller_product_listing.product_pricing) as "Order Total"
    FROM customer
    INNER JOIN customer_order ON customer.id = customer_order.id
    INNER JOIN item_order ON customer_order.Customer_order_id = item_order.Customer_order_id
    INNER JOIN seller_product_listing on seller_product_listing.seller_product_listing_ID = item_order.seller_product_listing_ID
    WHERE customer.id = given_id
    GROUP BY customer_order.Customer_order_id;
END //
DELIMITER ;

DROP PROCEDURE IF EXISTS cis_363_project.is_prime;
DELIMITER //
CREATE PROCEDURE cis_363_project.is_prime(IN id INT)
BEGIN
    SELECT Prime_member_ID FROM prime_member
    WHERE prime_member.id = id;
END//
DELIMITER ;
```

Let's click on the **Remove** button on the items with the Seller Product Listing IDs 25 and 50 and go back to their product page site_product_id: 2

Amazon Clone

127.0.0.1:8000/profile/

Welcome Username_2 Profile Logout

You have deleted Seller product listing ID from your Order: 50

Username_2

Email_2@email.com

Prime Member: True

Address 1: Address_1_2

Address 2: None

City: City_2

State/Province: State_Province_2

Zip: 96575

Phone 1: 8382456836

Phone 2: None

Create new Order

Your orders

Order ID	Amount of items in order	Order Total
2	2	13098.98

Individual Orders

Order ID: 2

Delete

Seller product listing ID	Name	Price	Description	Seller	Manufacturer	
2	Product_name_2	98.99	Product_description_2	Seller_name_10	Manufacturer_name_17	<div>Remove</div>
34	Product_name_34	12999.99	Product_description_34	Seller_name_7	Manufacturer_name_16	<div>Remove</div>

Amazon Clone

127.0.0.1:8000/product/2/

Welcome Username_2 Profile Logout

PRODUCT PAGE FOR site_product_id: 2

Product average star rating: 2

Amount of products left: 6

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
25	6	Seller_name_6	2	Manufacturer_name_19	Product_name_25	25	12999.99	Product_description_25	<div>Order</div>
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29	<div>Order</div>
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41	<div>Order</div>
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42	<div>Order</div>
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46	<div>Order</div>
50	6	Seller_name_6	2	Manufacturer_name_16	Product_name_50	50	98.99	Product_description_50	<div>Order</div>

COMMENTS

Username_6

May 21, 2020, 9 a.m.

Comment Nice!

Rating: 4

Username_7

May 21, 2020, 9 a.m.

Comment Nice!

Rating: 2

Username_2

May 21, 2020, 5:19 p.m.

Comment THIS PRODUCT SUCKS!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Rating: 1

Here is the corresponding code in python and code in MySQL for the procedure call when the **Remove** button is pressed.

```
def site_remove_from_order(request, customer_order_id, seller_product_listing_id):
    # Calling procedure remove_from_item_order in MySQL
    with connection.cursor() as cursor:
        cursor.callproc('cis_363_project.remove_from_item_order', [customer_order_id, seller_product_listing_id])

    messages.warning(request, f'You have deleted Seller product listing ID from your Order: {seller_product_listing_id}')
    return redirect('website:profile')
```

The procedure in MySQL

```
DROP PROCEDURE IF EXISTS cis_363_project.remove_from_item_order;
DELIMITER //
CREATE PROCEDURE cis_363_project.remove_from_item_order(IN Customer_order_ID_given INT, IN Seller_product_listing_ID_given INT)
BEGIN
    DELETE FROM item_order
    WHERE item_order.customer_order_id = Customer_order_ID_given AND item_order.Seller_product_listing_ID = Seller_product_listing_ID_given;
END//
DELIMITER ;
```

Now back on the profile page, click on **Create new Order** button and go to site_product_id: 4 and order Seller Product Listing IDs 28 and 39.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/profile/`. The page header includes the 'amazon clone' logo and a welcome message for 'Username_2'. A green notification bar at the top states 'You have created a new Order ID'. The main content area is divided into sections: 'Username_2' with user details, a 'Create new Order' button, 'Your orders' with a table showing one order, and 'Individual Orders' with a table listing items in the order and 'Remove' buttons for each.

Username_2
Email: email_2@email.com
Prime Member: True
Address 1: Address_1_2
Address 2: None
City: City_2
State/Province: State_Province_2
Zip: 90575
Phone 1: 8382456836
Phone 2: None
[Create new Order](#)

Your orders

Order ID	Amount of items in order	Order Total
2	2	13098.98

Individual Orders

Order ID: 2 [Delete](#)

Seller product listing ID	Name	Price	Description	Seller	Manufacturer	
2	Product_name_2	98.99	Product_description_2	Seller_name_10	Manufacturer_name_17	Remove
34	Product_name_34	12999.99	Product_description_34	Seller_name_7	Manufacturer_name_16	Remove

Product page for site_product_id: 4

Amazon Clone

127.0.0.1:8000/product/4/

Welcome Username_2 Profile Logout

PRODUCT PAGE FOR site_product_id: 4

Product average star rating: 4
Amount of products left: 6

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
26	10	Seller_name_10	4	Manufacturer_name_17	Product_name_26	26	399.98	Product_description_26	Order
28	7	Seller_name_7	4	Manufacturer_name_19	Product_name_28	28	399.98	Product_description_28	Order
32	9	Seller_name_9	3	Manufacturer_name_18	Product_name_32	32	12999.99	Product_description_32	Order
35	10	Seller_name_10	4	Manufacturer_name_20	Product_name_35	35	98.99	Product_description_35	Order
36	9	Seller_name_9	3	Manufacturer_name_17	Product_name_36	36	399.98	Product_description_36	Order
39	10	Seller_name_10	4	Manufacturer_name_16	Product_name_39	39	399.98	Product_description_39	Order

COMMENTS

Username_4
May 21, 2020, 9 a.m.
Comment Very cool!
Rating: 4

Username_9
May 21, 2020, 9 a.m.
Comment Looks Good!
Rating: 2

Username_10
May 21, 2020, 9 a.m.
Comment Looks Good!
Rating: 5

Ordering Seller Product Listing IDs 28 and 39

Amazon Clone

127.0.0.1:8000/product/4/

Welcome Username_2 Profile Logout

PRODUCT PAGE FOR site_product_id: 4

Product average star rating: 4
Amount of products left: 4

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
26	10	Seller_name_10	4	Manufacturer_name_17	Product_name_26	26	399.98	Product_description_26	Order
32	9	Seller_name_9	3	Manufacturer_name_18	Product_name_32	32	12999.99	Product_description_32	Order
35	10	Seller_name_10	4	Manufacturer_name_20	Product_name_35	35	98.99	Product_description_35	Order
36	9	Seller_name_9	3	Manufacturer_name_17	Product_name_36	36	399.98	Product_description_36	Order

COMMENTS

Username_4
May 21, 2020, 9 a.m.
Comment Very cool!
Rating: 4

Username_9
May 21, 2020, 9 a.m.
Comment Looks Good!
Rating: 2

Username_10
May 21, 2020, 9 a.m.
Comment Looks Good!
Rating: 5

Make Comment

Comments

The results of our actions back in our profile (Notice the changes in our *Order Total* column for each order)

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/profile/'. The page header includes the 'amazon clone' logo and a welcome message for 'Username_2' with links to 'Profile' and 'Logout'.

The profile section for 'Username_2' (Email: email_2@email.com) lists user details: Prime Member (True), Address 1 (Address_1_2), Address 2 (None), City (City_2), State/Province (State_Province_2), Zip (96575), Phone 1 (8382456836), and Phone 2 (None). A 'Create new Order' button is present.

The 'Your orders' section contains a table with 3 columns: Order ID, Amount of items in order, and Order Total.

Order ID	Amount of items in order	Order Total
2	2	13098.98
11	2	799.96

The 'Individual Orders' section shows details for Order ID: 2 (with a 'Delete' button) and Order ID: 11 (with a 'Delete' button'). Each order is represented by a table with columns: Seller product listing ID, Name, Price, Description, Seller, Manufacturer, and a 'Remove' button.

Order ID: 2 details:

Seller product listing ID	Name	Price	Description	Seller	Manufacturer	
2	Product_name_2	98.99	Product_description_2	Seller_name_10	Manufacturer_name_17	Remove
34	Product_name_34	12999.99	Product_description_34	Seller_name_7	Manufacturer_name_16	Remove

Order ID: 11 details:

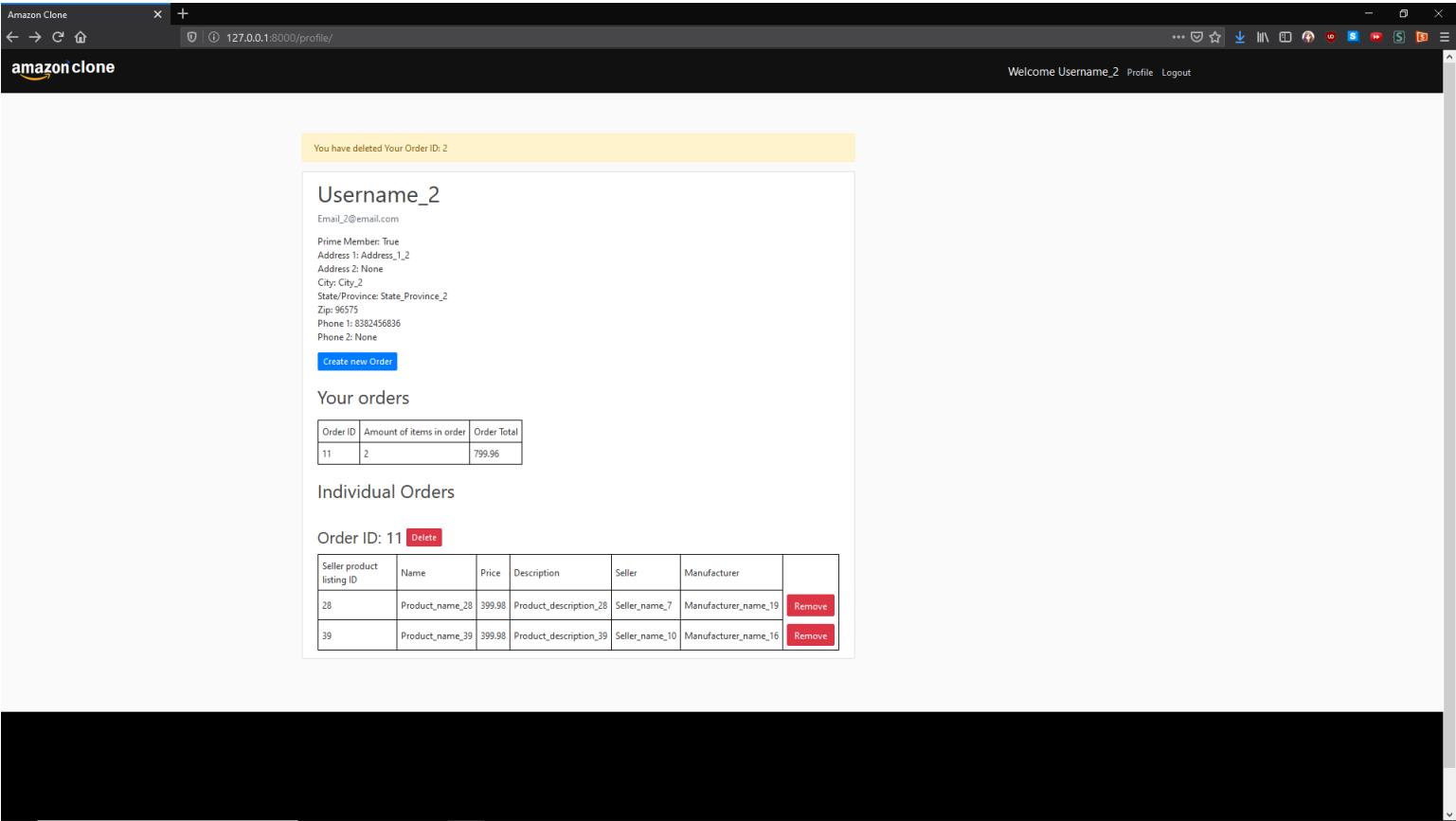
Seller product listing ID	Name	Price	Description	Seller	Manufacturer	
28	Product_name_28	399.98	Product_description_28	Seller_name_7	Manufacturer_name_19	Remove
39	Product_name_39	399.98	Product_description_39	Seller_name_10	Manufacturer_name_16	Remove

Here are the procedures and python code used in these operations which is just the **Create new Order** button as the rest was already covered.

```
def site_create_order(request):  
    # Calling procedure create_customer_order in MySQL  
    with connection.cursor() as cursor:  
        cursor.callproc('cis_363_project.create_customer_order', [request.user.id])  
  
    messages.success(request, f'You have created a new Order ID')  
    return redirect('website:profile')
```

```
DROP PROCEDURE IF EXISTS cis_363_project.create_customer_order;  
DELIMITER //  
CREATE PROCEDURE cis_363_project.create_customer_order(IN id_given INT)  
BEGIN  
    INSERT INTO customer_order(id) VALUES (id_given);  
END//  
DELIMITER ;
```

Now let's **Delete** Order ID: 2 and look back to their corresponding site_product_id 2 for Seller product listing ID 34 and site_product_id 4 for Seller product listing ID 2



Here is the code for this operation.

```
def site_delete_order(request, customer_order_id):  
    # Calling procedure delete_order in MySQL  
    with connection.cursor() as cursor:  
        cursor.callproc('cis_363_project.delete_order', [customer_order_id])  
  
    messages.warning(request, f'You have deleted Your Order ID: {customer_order_id}')  
    return redirect('website:profile')
```

```
DROP PROCEDURE IF EXISTS cis_363_project.delete_order;  
DELIMITER //  
CREATE PROCEDURE cis_363_project.delete_order(IN Customer_order_ID_given INT)  
BEGIN  
    DELETE FROM item_order  
    WHERE item_order.customer_order_id = Customer_order_ID_given;  
  
    DELETE FROM customer_order  
    WHERE customer_order.customer_order_id = Customer_order_ID_given;  
END//  
DELIMITER ;
```


Seller Product Listing ID 34 is back here

Amazon Clone

127.0.0.1:8000/product/2/

Welcome Username_2 Profile Logout

PRODUCT PAGE FOR site_product_id: 2

Product average star rating: 2
Amount of products left: 7

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
25	6	Seller_name_6	2	Manufacturer_name_19	Product_name_25	25	12999.99	Product_description_25	Order
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29	Order
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	12999.99	Product_description_34	Order
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41	Order
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42	Order
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46	Order
50	6	Seller_name_6	2	Manufacturer_name_16	Product_name_50	50	98.99	Product_description_50	Order

COMMENTS

Username_6
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 4

Username_7
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 2

Username_2
May 21, 2020, 5:19 p.m.
Comment THIS PRODUCT

Seller Product Listing ID 2 is back here

Amazon Clone

127.0.0.1:8000/product/4/

Welcome Username_2 Profile Logout

PRODUCT PAGE FOR site_product_id: 4

Product average star rating: 4
Amount of products left: 5

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
2	10	Seller_name_10	4	Manufacturer_name_17	Product_name_2	2	98.99	Product_description_2	Order
26	10	Seller_name_10	4	Manufacturer_name_17	Product_name_26	26	399.98	Product_description_26	Order
32	9	Seller_name_9	3	Manufacturer_name_18	Product_name_32	32	12999.99	Product_description_32	Order
35	10	Seller_name_10	4	Manufacturer_name_20	Product_name_35	35	98.99	Product_description_35	Order
36	9	Seller_name_9	3	Manufacturer_name_17	Product_name_36	36	399.98	Product_description_36	Order

COMMENTS

Username_4
May 21, 2020, 9 a.m.
Comment Very cool!
Rating: 4

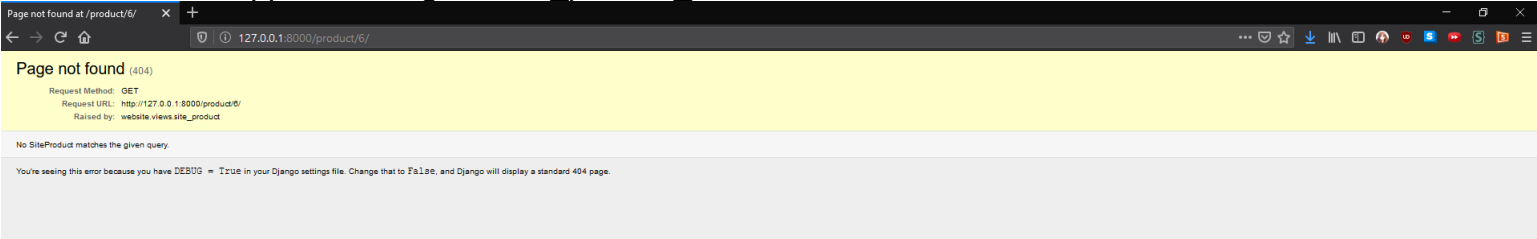
Username_9
May 21, 2020, 9 a.m.
Comment Looks Good!
Rating: 2

Username_10
May 21, 2020, 9 a.m.
Comment Looks Good!
Rating: 5

Currently there are 5 site_product_ID pages.

Site_product_ID	Product_rating_avg
1	0
2	3
3	3
4	4
5	2
NULL	NULL

Let’s see what happens if we go to site_product id 6 which does not exist



We get a 404 error,

Now let’s login as a Username_1 who is an Admin on Django’s built in Admin page and add that page. Once the page exists in the DB, Django can automatically generate that page.



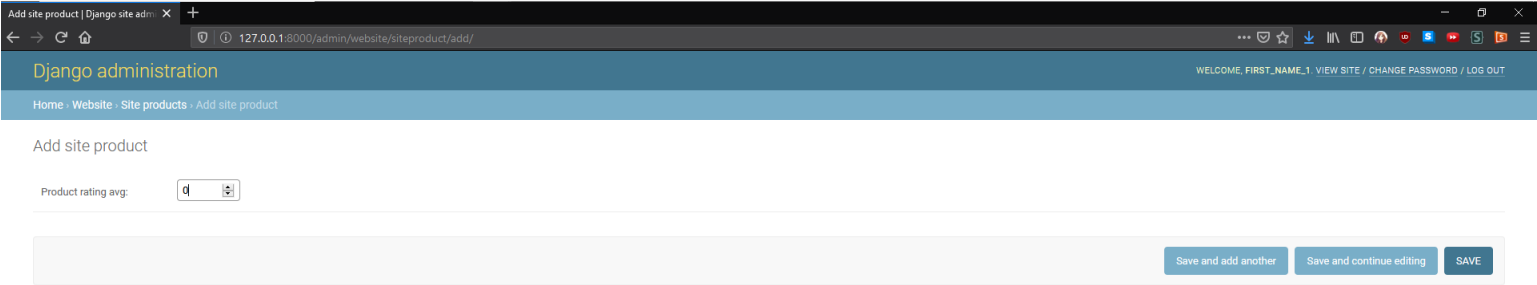
Django administration

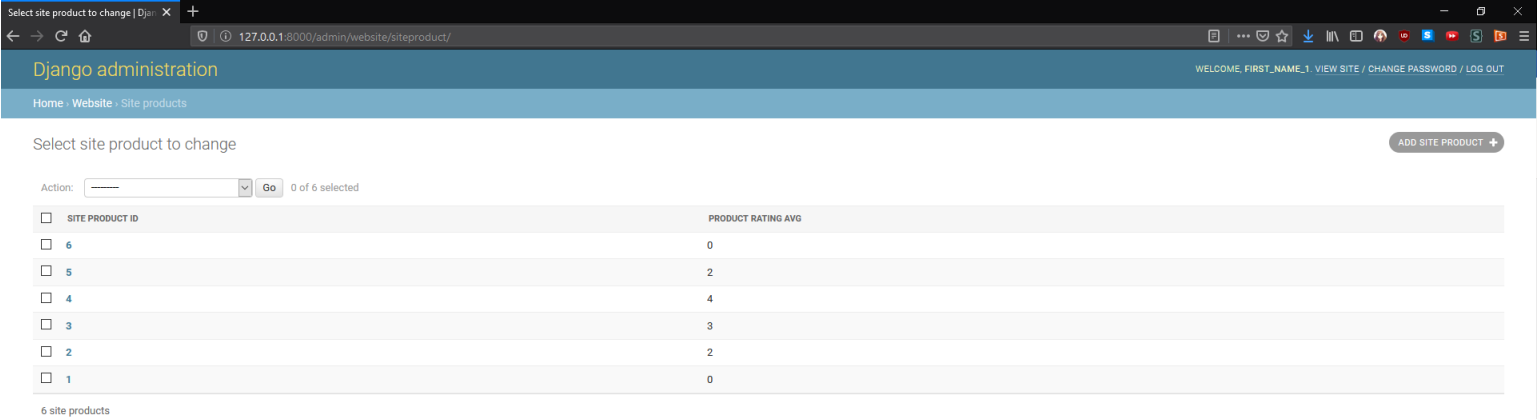
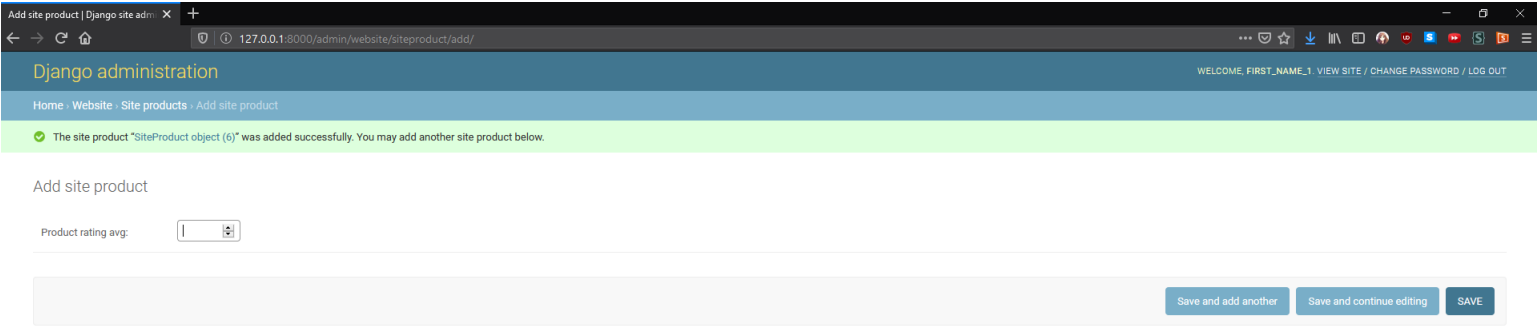
You are authenticated as Username_2, but are not authorized to access this page. Would you like to login to a different account?

Username:

Password:

Log in





Now let's go to <http://127.0.0.1:8000/product/6/> and we see the page can be generated now.



PRODUCT PAGE FOR site_product_id: 6

Product average star rating: 0

Amount of products left: 0

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
---------------------------	-----------	-------------	---------------	-------------------	--------------	------------	-----------------	---------------------

COMMENTS

Make Comment

Comments

Product rating

1

Post

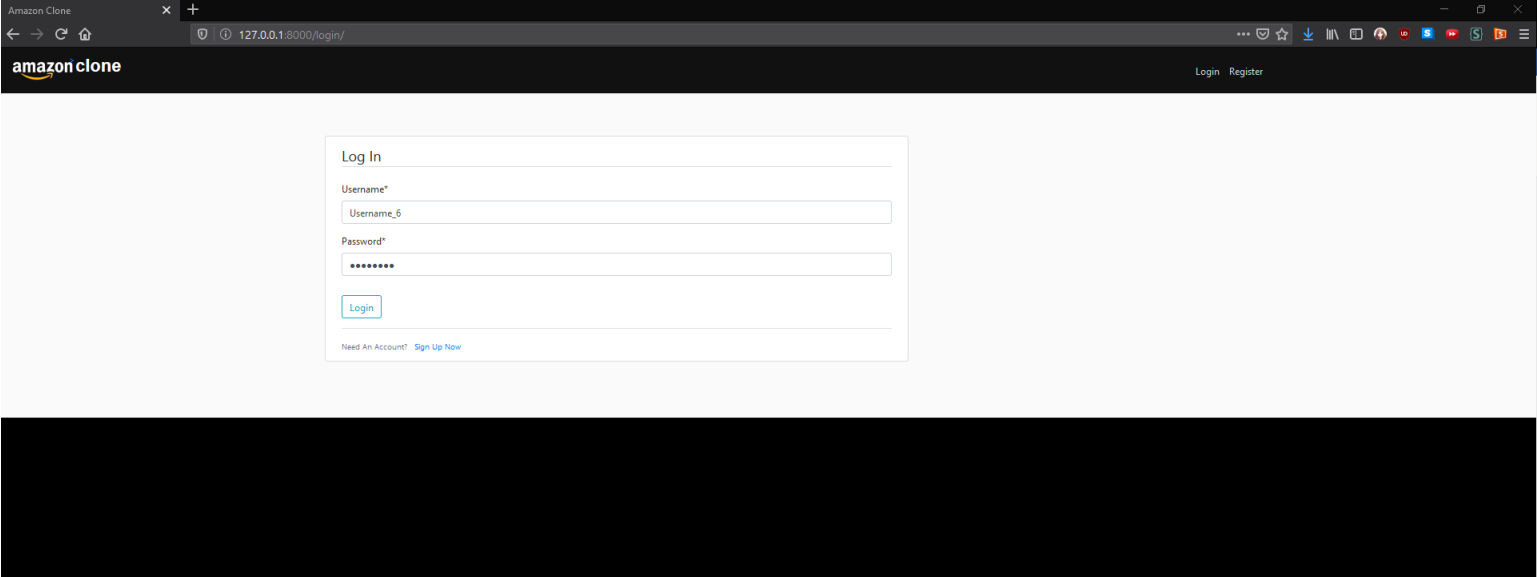
PRODUCT TAGS



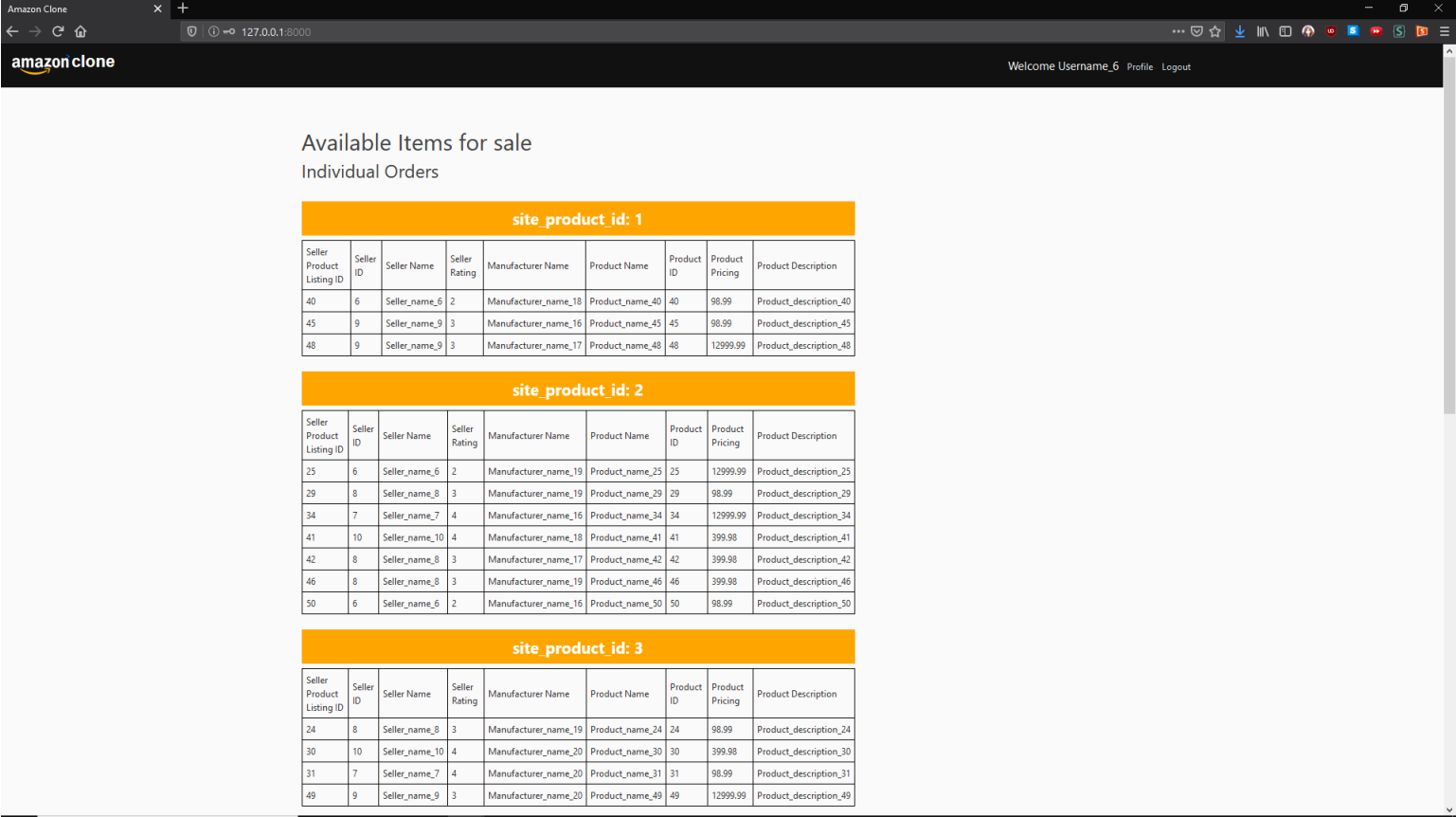
Let's login as a Seller. In the DB's seller table, sellers are assigned to specific users and their ids have a one to one relationship to the ids in the user table and thus auth_user table.

id	Credit_card_ID	Seller_name	Seller_rating
6	6	Seller_name_6	2
7	7	Seller_name_7	4
8	8	Seller_name_8	3
9	9	Seller_name_9	3
10	10	Seller_name_10	4
NULL	NULL	NULL	NULL

Standard login



Home page



Note: We can see that at the bottom of the home page site_product_id: 6 is added dynamically via code.

Amazon Clone									
Welcome Username_6 Profile Logout									
Listing ID									
2	10	Seller_name_10	4	Manufacturer_name_17	Product_name_2	2	98.99	Product_description_2	
26	10	Seller_name_10	4	Manufacturer_name_17	Product_name_26	26	399.98	Product_description_26	
32	9	Seller_name_9	3	Manufacturer_name_18	Product_name_32	32	12999.99	Product_description_32	
35	10	Seller_name_10	4	Manufacturer_name_20	Product_name_35	35	98.99	Product_description_35	
36	9	Seller_name_9	3	Manufacturer_name_17	Product_name_36	36	399.98	Product_description_36	
site_product_id: 5									
Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
21	10	Seller_name_10	4	Manufacturer_name_19	Product_name_21	21	399.98	Product_description_21	
22	8	Seller_name_8	3	Manufacturer_name_19	Product_name_22	22	12999.99	Product_description_22	
23	10	Seller_name_10	4	Manufacturer_name_20	Product_name_23	23	98.99	Product_description_23	
27	10	Seller_name_10	4	Manufacturer_name_16	Product_name_27	27	12999.99	Product_description_27	
33	6	Seller_name_6	2	Manufacturer_name_20	Product_name_33	33	98.99	Product_description_33	
37	10	Seller_name_10	4	Manufacturer_name_20	Product_name_37	37	399.98	Product_description_37	
38	7	Seller_name_7	4	Manufacturer_name_16	Product_name_38	38	12999.99	Product_description_38	
43	8	Seller_name_8	3	Manufacturer_name_19	Product_name_43	43	98.99	Product_description_43	
44	7	Seller_name_7	4	Manufacturer_name_16	Product_name_44	44	12999.99	Product_description_44	
47	6	Seller_name_6	2	Manufacturer_name_19	Product_name_47	47	98.99	Product_description_47	
site_product_id: 6									
Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	

Now let's go to the Profile page and see the differences it has compared to a Customer. As opposed to a customer, sellers have these features:

- List a new Product
 - Assign Manufacturer ID based on existing Manufacturers in the Manufacturer table in the DB (This is enforced by the code to prevent unwanted behavior)
 - Set product name
 - Assign where the product will go on the site (e.g. site_product_id 6) (Also enforced by code to prevent unwanted behavior)
 - Set the price
 - Set the description
- View products they own in Customer's orders
- View products they own listed
- Delete products they have listed (not in a Customer's order)
- View revenue
- They are also not Prime members

Notice that the URL for the profile page is the same as the customer, I set up code to distinguished between the types of users and display the appropriate information for both.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/profile/'. The page header features the 'amazon clone' logo on the left and a user profile section on the right with the text 'Welcome Username_6 Profile Logout'.

--	--	--	--	--	--

Seller product listing ID	Manufacturer	Name	Product ID	Product pricing	Sold to User
---------------------------	--------------	------	------------	-----------------	--------------

Here is the python code that calls 4 procedures and 1 Django query to the User table for user information for the profile

```
# Calling procedure is_prime in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.seller_all_site_product_id', [user_current.id])
    seller_all_site_product_id = cursor.fetchall()

print(seller_all_site_product_id)
list_dict_seller_product_listing_sold = []

for i in seller_all_site_product_id:
    # Calling procedure get_seller_product_listing_sold in MySQL
    with connection.cursor() as cursor:
        cursor.callproc('cis_363_project.get_seller_product_listing_sold', [i[0], user_current.id])
        seller_product_listing_sold = cursor.fetchall()
        list_dict_seller_product_listing_sold.append({i[0]: seller_product_listing_sold})

# Calling procedure get_seller_revenue in MySQL
with connection.cursor() as cursor:
    cursor.callproc('cis_363_project.get_seller_revenue', [user_current.id])
    seller_revenue = cursor.fetchone()

list_dict_seller_product_listing_not_sold = []

for i in seller_all_site_product_id:
    # Calling procedure get_seller_items_not_sold in MySQL
    with connection.cursor() as cursor:
        cursor.callproc('cis_363_project.get_seller_product_listing_not_sold', [i[0], user_current.id])
        seller_items_not_sold = cursor.fetchall()
        list_dict_seller_product_listing_not_sold.append({i[0]: seller_items_not_sold})

context = {
    'is_customer': False,
    "list_dict_seller_product_listing_sold": list_dict_seller_product_listing_sold,
    "seller_product_listing_sold_headers": ("Seller product listing ID", "Manufacturer", "Name",
                                           "Product ID", "Product pricing", "Sold to User"),
    "seller_revenue": seller_revenue,
    "list_dict_seller_product_listing_not_sold": list_dict_seller_product_listing_not_sold,
    "seller_product_listing_not_sold_headers": ("Seller product listing ID",
                                                "Product ID", "Product pricing",
                                                "Name", "Manufacturer"),
    'user_profile': User.objects.get(pk=user_current.id)
}
```

The procedures here:

- Get all site_product_id where the seller has products in
- Get all products sold by the seller
- Calculate revenue made by the seller
- Get all products not sold by the seller

Here are the procedures in MySQL

```
DROP PROCEDURE IF EXISTS cis_363_project.seller_all_site_product_id;
DELIMITER //
CREATE PROCEDURE cis_363_project.seller_all_site_product_id(IN given_id INT)
BEGIN
    SELECT site_product_id FROM seller_product_listing
    WHERE seller_product_listing.id = given_id
    GROUP BY site_product_id;
END//
DELIMITER ;

DROP PROCEDURE IF EXISTS cis_363_project.get_seller_product_listing_sold;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_seller_product_listing_sold(IN given_site_product_id INT, IN given_seller_id INT)
BEGIN
    SELECT seller_product_listing.Seller_product_listing_ID,
    Manufacturer_name, Product_name, product.Product_ID, Product_pricing,
    auth_user.username as "Sold to User"
    FROM seller_product_listing
    JOIN product ON seller_product_listing.Product_ID = product.Product_ID
    JOIN item_order ON seller_product_listing.Seller_product_listing_ID = item_order.Seller_product_listing_ID
    JOIN seller ON seller_product_listing.id = seller.id
    JOIN manufacturer ON product.id = manufacturer.id
    JOIN customer_order ON customer_order.customer_order_id = item_order.customer_order_id
    JOIN auth_user ON auth_user.id = customer_order.id
    WHERE seller_product_listing.Site_product_ID = given_site_product_id AND seller.id = given_seller_id;
END//
DELIMITER ;

DROP PROCEDURE IF EXISTS cis_363_project.get_seller_product_listing_not_sold;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_seller_product_listing_not_sold(IN given_site_product_id INT, IN given_seller_id INT)
BEGIN
    SELECT seller_product_listing.seller_product_listing_id, product.product_ID, product_pricing, product_name, manufacturer_name
    FROM seller_product_listing
    JOIN product ON seller_product_listing.Product_ID = product.Product_ID
    JOIN manufacturer ON product.id = manufacturer.id
    LEFT JOIN item_order ON seller_product_listing.seller_product_listing_id = item_order.seller_product_listing_id
    WHERE seller_product_listing.id = given_seller_id
    AND site_product_id = given_site_product_id
    AND item_order.customer_order_id is NULL;
END//
DELIMITER ;

DROP PROCEDURE IF EXISTS cis_363_project.get_seller_revenue;
DELIMITER //
CREATE PROCEDURE cis_363_project.get_seller_revenue(IN given_seller_id INT)
BEGIN
    SELECT sum(product_pricing) as "Revenue"
    FROM seller_product_listing
    JOIN item_order on seller_product_listing.seller_product_listing_ID = item_order.seller_product_listing_ID
    WHERE seller_product_listing.id = given_seller_id;
END//
DELIMITER ;
```


Let's Delete Seller product listing ID 25 from site_product_id: 2 and see it's result

Amazon Clone

127.0.0.1:8000/profile/

Welcome Username_6ProfileLogout

Username_6

Email_6@email.com

Address 1: Address_1_6
Address 2: None
City: City_6
State/Province: State_Province_6
Zip: 72537
Phone 1: 9077513021
Phone 2: None

Revenue: 26098.97

List new Product

Your items Not Sold

site_product_id: 2

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
50	50	98.99	Product_name_50	Manufacturer_name_16	Delete

site_product_id: 1

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
40	40	98.99	Product_name_40	Manufacturer_name_18	Delete

site_product_id: 5

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
33	33	98.99	Product_name_33	Manufacturer_name_20	Delete
47	47	98.99	Product_name_47	Manufacturer_name_19	Delete

Your Items Sold

site_product_id: 2

Seller product listing ID	Manufacturer	Name	Product ID	Product pricing	Sold to User
---------------------------	--------------	------	------------	-----------------	--------------

Here is the home page

Amazon Clone

127.0.0.1:8000

Welcome Username_6ProfileLogout

Available Items for sale

Individual Orders

site_product_id: 1

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
40	6	Seller_name_6	2	Manufacturer_name_18	Product_name_40	40	98.99	Product_description_40
45	9	Seller_name_9	3	Manufacturer_name_16	Product_name_45	45	98.99	Product_description_45
48	9	Seller_name_9	3	Manufacturer_name_17	Product_name_48	48	12999.99	Product_description_48

site_product_id: 2

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	12999.99	Product_description_34
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46
50	6	Seller_name_6	2	Manufacturer_name_16	Product_name_50	50	98.99	Product_description_50

site_product_id: 3

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
24	8	Seller_name_8	3	Manufacturer_name_19	Product_name_24	24	98.99	Product_description_24
30	10	Seller_name_10	4	Manufacturer_name_20	Product_name_30	30	399.98	Product_description_30
31	7	Seller_name_7	4	Manufacturer_name_20	Product_name_31	31	98.99	Product_description_31
49	9	Seller_name_9	3	Manufacturer_name_20	Product_name_49	49	12999.99	Product_description_49

Here is the page for site_product_id: 2

Amazon Clone

127.0.0.1:8000/product/2/

Welcome Username_6 Profile Logout

PRODUCT PAGE FOR site_product_id: 2

Product average star rating: 2
Amount of products left: 6

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
29	8	Seller_name_8	3	Manufacturer_name_19	Product_name_29	29	98.99	Product_description_29	Order
34	7	Seller_name_7	4	Manufacturer_name_16	Product_name_34	34	12999.99	Product_description_34	Order
41	10	Seller_name_10	4	Manufacturer_name_18	Product_name_41	41	399.98	Product_description_41	Order
42	8	Seller_name_8	3	Manufacturer_name_17	Product_name_42	42	399.98	Product_description_42	Order
46	8	Seller_name_8	3	Manufacturer_name_19	Product_name_46	46	399.98	Product_description_46	Order
50	6	Seller_name_6	2	Manufacturer_name_16	Product_name_50	50	98.99	Product_description_50	Order

COMMENTS

Username_6
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 4

Username_7
May 21, 2020, 9 a.m.
Comment Nice!
Rating: 2

Username_2
May 21, 2020, 5:19 p.m.
Comment THIS PRODUCT SUCKS!!!!!!!!!!!!!!!!!!!!!!
Rating: 1

Now let's create a new product and list it on site_product_id: 6

Amazon Clone

127.0.0.1:8000/createSellerProductListing/

Welcome Username_6 Profile Logout

Crate a new Product and list it on the website

Manufacturer ID*
19

Product name*
"I CAN'T THINK OF A NAME FOR A PRODUCT"

Site product ID*
6

Product pricing*
777.77

Product description*
A Cool product made by Joseph

List Product

Back to our profile

Amazon Clone

127.0.0.1:8000/profile/

Welcome Username_6ProfileLogout

You have created a new Seller product listing ID: Seller object (6) in site_product_id: 6

Username_6

Email_6@email.com

Address 1: Address_1_6

Address 2: None

City: City_5

State/Province: State_Province_6

Zip: 72537

Phone 1: 9977513021

Phone 2: None

Revenue: 26098.97

List new Product

Your items Not Sold

site_product_id: 2

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
50	50	98.99	Product_name_50	Manufacturer_name_16	Delete

site_product_id: 1

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
40	40	98.99	Product_name_40	Manufacturer_name_18	Delete

site_product_id: 5

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
33	33	98.99	Product_name_33	Manufacturer_name_20	Delete
47	47	98.99	Product_name_47	Manufacturer_name_19	Delete

site_product_id: 6

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
---------------------------	------------	-----------------	------	--------------	--

The new product is created, and it is at site_prodouct_id: 6

Amazon Clone

127.0.0.1:8000/profile/

Welcome Username_6ProfileLogout

Your items Not Sold

site_product_id: 2

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
50	50	98.99	Product_name_50	Manufacturer_name_16	Delete

site_product_id: 1

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
40	40	98.99	Product_name_40	Manufacturer_name_18	Delete

site_product_id: 5

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
33	33	98.99	Product_name_33	Manufacturer_name_20	Delete
47	47	98.99	Product_name_47	Manufacturer_name_19	Delete

site_product_id: 6

Seller product listing ID	Product ID	Product pricing	Name	Manufacturer	
51	51	777.77	"I CAN'T THINK OF A NAME FOR A PRODUCT"	Manufacturer_name_19	Delete

Your Items Sold

site_product_id: 2

Seller product listing ID	Manufacturer	Name	Product ID	Product pricing	Sold to User
3	Manufacturer_name_17	Product_name_3	3	98.99	Username_4
12	Manufacturer_name_16	Product_name_12	12	12999.99	Username_4

site_product_id: 1

Seller product listing ID	Manufacturer	Name	Product ID	Product pricing	Sold to User
20	Manufacturer_name_18	Product_name_20	20	12999.99	Username_5

site_product_id: 5

Here is the new product on the home page

Amazon Clone

127.0.0.1:8000

Welcome Username_6 Profile Logout

32	9	Seller_name_9	3	Manufacturer_name_18	Product_name_32	32	12999.99	Product_description_32
35	10	Seller_name_10	4	Manufacturer_name_20	Product_name_35	35	98.99	Product_description_35
36	9	Seller_name_9	3	Manufacturer_name_17	Product_name_36	36	399.98	Product_description_36

site product id: 5

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
21	10	Seller_name_10	4	Manufacturer_name_19	Product_name_21	21	399.98	Product_description_21
22	8	Seller_name_8	3	Manufacturer_name_19	Product_name_22	22	12999.99	Product_description_22
23	10	Seller_name_10	4	Manufacturer_name_20	Product_name_23	23	98.99	Product_description_23
27	10	Seller_name_10	4	Manufacturer_name_16	Product_name_27	27	12999.99	Product_description_27
33	6	Seller_name_6	2	Manufacturer_name_20	Product_name_33	33	98.99	Product_description_33
37	10	Seller_name_10	4	Manufacturer_name_20	Product_name_37	37	399.98	Product_description_37
38	7	Seller_name_7	4	Manufacturer_name_16	Product_name_38	38	12999.99	Product_description_38
43	8	Seller_name_8	3	Manufacturer_name_19	Product_name_43	43	98.99	Product_description_43
44	7	Seller_name_7	4	Manufacturer_name_16	Product_name_44	44	12999.99	Product_description_44
47	6	Seller_name_6	2	Manufacturer_name_19	Product_name_47	47	98.99	Product_description_47

site product id: 6

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
51	6	Seller_name_6	2	Manufacturer_name_19	"I CAN'T THINK OF A NAME FOR A PRODUCT"	51	777.77	A Cool product made by Joseph

Here is the new product on site_product_id: 6

Amazon Clone

127.0.0.1:8000/product/6/

Welcome Username_6 Profile Logout

PRODUCT PAGE FOR site_product_id: 6

Product average star rating: 0
Amount of products left: 1

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
51	6	Seller_name_6	2	Manufacturer_name_19	"I CAN'T THINK OF A NAME FOR A PRODUCT"	51	777.77	A Cool product made by Joseph	Order

COMMENTS

Make Comment

Comments

Product rating

1

Post

PRODUCT TAGS

Here is the code

```
def site_delete_seller_product_listing(request, seller_product_listing_id):
    # Calling procedure delete_seller_product_listing in MySQL
    with connection.cursor() as cursor:
        cursor.callproc('cis_363_project.delete_seller_product_listing', [seller_product_listing_id])
    return redirect('website:profile')

DROP PROCEDURE IF EXISTS cis_363_project.delete_seller_product_listing;
DELIMITER //
CREATE PROCEDURE cis_363_project.delete_seller_product_listing(IN seller_product_listing_id_given INT)
BEGIN
    DELETE FROM seller_product_listing
    WHERE seller_product_listing.seller_product_listing_id = seller_product_listing_id_given;
END//
DELIMITER ;
```

Now let's create a New user on the site and then order that product.

Amazon Clone

127.0.0.1:8000/register

amazon clone

Login Register

Create Account

Username*

Joseph_Edradan

Required: 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name*

Joseph

Last name*

Edradan

Email*

fakeemail@gmail.com

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

Address 1*

Fake address 1

Address 2

Fake address 2

City*

Fake city

State province*

CA

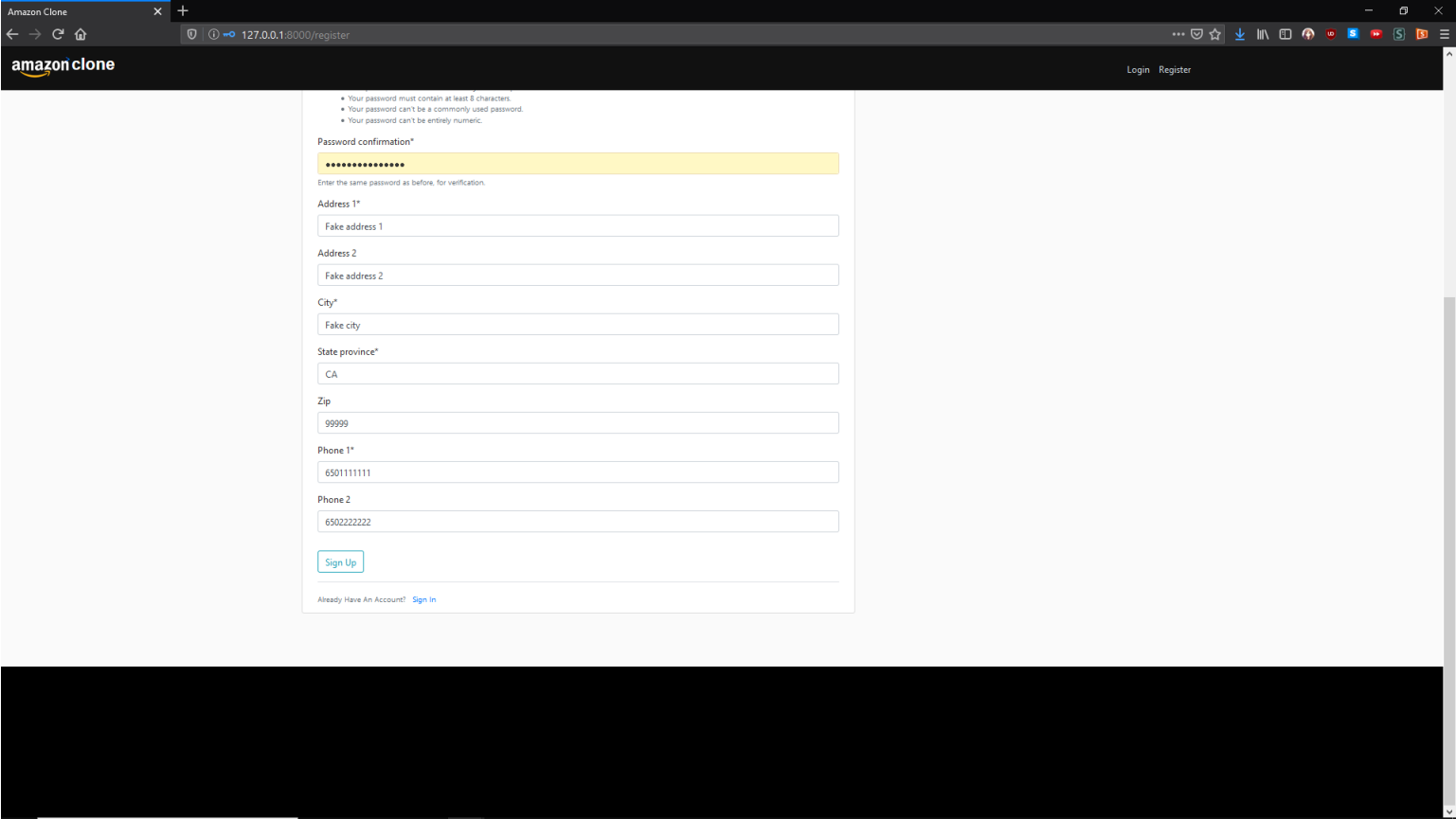
Zip

99999

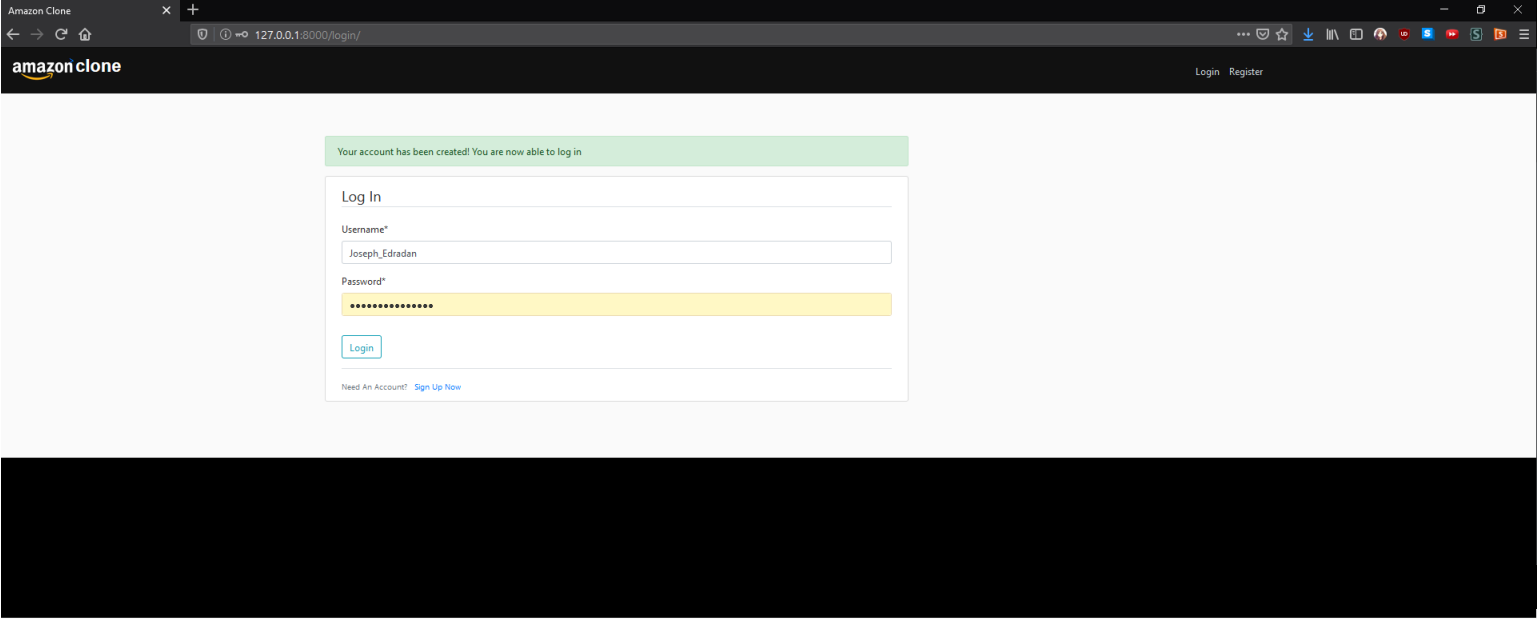
Phone 1*

555-555-5555

Createing the account



Logging in the account



Going to the site_product_id: 6

Amazon Clone

127.0.0.1:8000/product/6/

Welcome Joseph_Edradan Profile Logout

PRODUCT PAGE FOR site_product_id: 6

Product average star rating: 0
Amount of products left: 1

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description	
51	6	Seller_name_6	2	Manufacturer_name_19	"I CAN'T THINK OF A NAME FOR A PRODUCT"	51	777.77	A Cool product made by Joseph	Order

COMMENTS

Make Comment

Comments

Product rating

1

Post

PRODUCT TAGS

Ordering the product

Amazon Clone

127.0.0.1:8000/product/6/

Welcome Joseph_Edradan Profile Logout

PRODUCT PAGE FOR site_product_id: 6

Product average star rating: 0
Amount of products left: 0

Seller Product Listing ID	Seller ID	Seller Name	Seller Rating	Manufacturer Name	Product Name	Product ID	Product Pricing	Product Description
---------------------------	-----------	-------------	---------------	-------------------	--------------	------------	-----------------	---------------------

COMMENTS

Make Comment

Comments

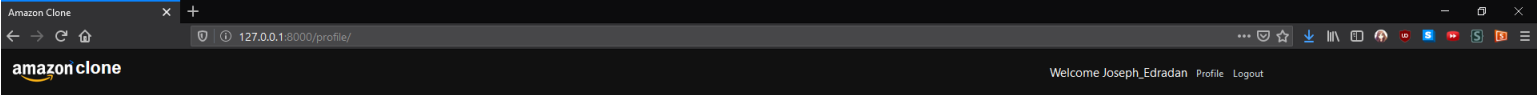
Product rating

1

Post

PRODUCT TAGS

Now the product is in our Order. When the account was created the account didn't initially have an order; however, if you look back at the procedure `cis_363_project.add_item_to_order_latest`; an order was created automatically.



Joseph_Edradan

fakeemail@gmail.com

Prime Member: False
Address 1: Fake address 1
Address 2: Fake address 2
City: Fake city
State/Province: CA
Zip: 99999
Phone 1: 6501111111
Phone 2: 6502222222

Create new Order

Your orders

Order ID	Amount of Items in order	Order Total
12	1	777.77

Individual Orders

Order ID: 12 [Delete](#)

Seller product listing ID	Name	Price	Description	Seller	Manufacturer	
51	?! CAN'T THINK OF A NAME FOR A PRODUCT	777.77	A Cool product made by Joseph	Seller_name_6	Manufacturer_name_19	Remove

The code involved for this process does not call any procedures from MySQL; instead, it uses Django's `UserCreationForm` and `forms.ModelForm` to call the models the appropriate models corresponding to the tables in the DB. The code creates an `auth_user` row, `user` row, and a `customer` row.

```
def site_register(request):
    if request.method == 'POST':
        form_default = UserRegisterFormExtended(request.POST)
        form_user_profile = UserRegisterFromCustom(request.POST)

        if form_default.is_valid() and form_user_profile.is_valid():
            form_default_saved = form_default.save() # id comes from form_default.save()

            form_default_saved_id = form_default_saved.id

            print("-" * 200)
            print("form_default.save() ID: {}".format(form_default_saved_id))
            print("form_default.save() type: {}".format(type(form_default_saved))) #
            print("-" * 200)

            user_profile = form_user_profile.save(commit=False)

            auth_user = AuthUser.objects.get(pk=form_default_saved_id)

            user_profile.id = auth_user # GET ID FROM THE default_save AND SET IT FOR user_profile

            user_profile.save() # Returns None

            # type(user_profile) is of type User

            Customer.objects.create(id=user_profile).save()

            messages.success(request, 'Your account has been created! You are now able to log in')
            return redirect('website:login')
        else:
            form_default = UserRegisterFormExtended()
            form_user_profile = UserRegisterFromCustom()

            context = {'form_default': form_default,
                      'form_user_profile': form_user_profile,
                      }

            return render(request, 'users/register.html', context)
```

Here are the tables in their respective order.

cis_363_project.auth_user;

id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined
8	pbkdf2_sha256\$180000\$Nefz7EPLuWK\$DuCZv...	NULL	0	Username_8	first_name_8	last_name_8	Email_8@email.com	0	1	2020-05-21 22:59:59.398359
9	pbkdf2_sha256\$180000\$M9PDndXMI9D\$8MW...	NULL	0	Username_9	first_name_9	last_name_9	Email_9@email.com	0	1	2020-05-21 22:59:59.483431
10	pbkdf2_sha256\$180000\$PG68peVqScXV\$vvNE...	NULL	0	Username_10	first_name_10	last_name_10	Email_10@email.com	0	1	2020-05-21 22:59:59.569505
11	pbkdf2_sha256\$180000\$nxod7PYecpAB\$N5Zt...	NULL	0	Username_11	first_name_11	last_name_11	Email_11@email.com	0	1	2020-05-21 22:59:59.656579
12	pbkdf2_sha256\$180000\$X7HEW6ASZd\$R09...	NULL	0	Username_12	first_name_12	last_name_12	Email_12@email.com	0	1	2020-05-21 22:59:59.742652
13	pbkdf2_sha256\$180000\$z2Wvm5UPWcID\$E+...	NULL	0	Username_13	first_name_13	last_name_13	Email_13@email.com	0	1	2020-05-21 22:59:59.828726
14	pbkdf2_sha256\$180000\$TQLXagakpyV\$8IB1D...	NULL	0	Username_14	first_name_14	last_name_14	Email_14@email.com	0	1	2020-05-21 22:59:59.917802
15	pbkdf2_sha256\$180000\$nhH4McP8G6p\$TnyK...	NULL	0	Username_15	first_name_15	last_name_15	Email_15@email.com	0	1	2020-05-21 23:00:00.004876
16	pbkdf2_sha256\$180000\$JQTmrB5Ly5\$Q9BZT...	NULL	0	Username_16	first_name_16	last_name_16	Email_16@email.com	0	1	2020-05-21 23:00:00.090949
17	pbkdf2_sha256\$180000\$wxSIX98weGbq\$Ov0B...	NULL	0	Username_17	first_name_17	last_name_17	Email_17@email.com	0	1	2020-05-21 23:00:00.179025
18	pbkdf2_sha256\$180000\$YXZSbKaxJTU\$NdlajN...	NULL	0	Username_18	first_name_18	last_name_18	Email_18@email.com	0	1	2020-05-21 23:00:00.266100
19	pbkdf2_sha256\$180000\$NkrMeailNZEQu\$b8NgI...	NULL	0	Username_19	first_name_19	last_name_19	Email_19@email.com	0	1	2020-05-21 23:00:00.353173
20	pbkdf2_sha256\$180000\$FgBEnTw3SCBx\$frIYY...	NULL	0	Username_20	first_name_20	last_name_20	Email_20@email.com	0	1	2020-05-21 23:00:00.438245
25	pbkdf2_sha256\$180000\$AcXIf6QcLmP\$0W2ID...	2020-05-22 03:09:47.565332	0	Joseph_Edra...	Joseph	Edradan	fakeemail@gmail.com	0	1	2020-05-22 03:09:14.773011
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

cis_363_project.user;

id	Address_1	Address_2	City	State_Province	Zip	Phone_1	Phone_2
7	Address_1_7	NULL	City_7	State_Province_7	32090	6951986730	NULL
8	Address_1_8	NULL	City_8	State_Province_8	27484	8750035058	NULL
9	Address_1_9	NULL	City_9	State_Province_9	67164	5870271011	NULL
10	Address_1_10	NULL	City_10	State_Province_10	89909	9086317834	NULL
11	Address_1_11	NULL	City_11	State_Province_11	59573	5614407036	NULL
12	Address_1_12	NULL	City_12	State_Province_12	20235	7192322326	NULL
13	Address_1_13	NULL	City_13	State_Province_13	77272	2382990994	NULL
14	Address_1_14	NULL	City_14	State_Province_14	72949	2472659543	NULL
15	Address_1_15	NULL	City_15	State_Province_15	73062	4972143439	NULL
16	Address_1_16	NULL	City_16	State_Province_16	45984	8984420365	NULL
17	Address_1_17	NULL	City_17	State_Province_17	48104	5501173776	NULL
18	Address_1_18	NULL	City_18	State_Province_18	50206	4745292483	NULL
19	Address_1_19	NULL	City_19	State_Province_19	13755	3700465468	NULL
20	Address_1_20	NULL	City_20	State_Province_20	48620	4767453301	NULL
25	Fake address 1	Fake addr...	Fake city	CA	99999	6501111111	6502222222

cis_363_project.customer;

id	Credit_card_ID
25	NULL
1	1
2	2
3	3
4	4
5	5
NULL	NULL

Purpose:

The purpose of this project is to make a crude copy of Amazon, it works for the most part, but does not handle advance features such as monetary transactions because that is too advanced.

Design:

The DB and the code could have been designed better to support more features for users by separating useful and less useful information into more tables at the cost of speed.

Implementation:

The code dynamically creates pages and uses the html templates as a basis for similar pages. Back end code also handles the appropriate information for each html page based on the URL. It also handles GET and POST requests and the Django framework also minimizes over all code. The project is still big though.