

Python Assignment

Submitted By:

Yugaank Sharma

Assignment Objective: Design and code a simple sales RESTful web application

To complete this assignment, following steps were taken:

1. Started a Django project called sales. Sales project can have multiple apps but one of the apps that I developed is called Apollo. This app allows us to add customer and product into the SQLite database.
2. The app not only allows us to add customer and products but also get the whole customer list, the products that they have in their cart, delete a customer or product and add/delete a product into/from the customer's cart.
3. The admin panel allows administrator to do the same functions. Username for admin panel is "admin" and password is "apollo2204"
4. The testing for the functions were done mostly using POSTMAN which is a chrome application and the demonstration can be seen in the video that I submitted along with this file.
5. POSTMAN allows us to add, delete or put in case of addition of a product into the customer's cart into the database. The POST or PUT data is sent as JSON.
6. There are two models, Product and Customer. The product table has three columns, primary key field which is auto generated, product name field and product description field. The customer table has four columns, primary key field which is auto generated, customer name field, customer last name initial and products field which is a many to many field of products table. The products field in the customer table is basically the customer's cart. We can add multiple products into the customer cart's.
7. Since, I didn't include the quantity field of each product, we are assuming that each product has a large quantity. If I was to include the quantity, I would have had to decrement the quantity field of each product anytime a customer adds that product. I would also allow a feature for customers to add multiple quantity of a product like, buy 3 cellphones of the same brand and model.
8. Since the products field (cart field) in the customer table is many to many field, I have not allowed cascading. Say, if a customer is removed, I don't want the products to be removed from the product table. But the many to many field allows me to cascade the products which are currently in the customer's cart to be removed in case a product is deleted from the product's table which is how it should happen.
9. A more better approach to do this assignment would be to use Django REST Framework which has a better usability because of its web browsable API. It's authentication policies includes the OAuth1a and OAuth2, has extensive documentation and great community support.
10. The Apollo welcome page has certain options to click on, like list all customers and their cart items or add a product/customer (functionality incomplete). In future, we can add those functionality plus other functions that could be done. A better UI is a possibility.
11. HTML forms send POST form data as encoded URL format and that would be against the guidelines of this assignment. So, I did plan to send the form data as an ajax request but did not

fully complete it. It had some issues which I will fix in the future. As ajax allows us to specify the content type.

12. Finally, to run the app, python manage.py runserver, and go to localhost:8000

In the zip archive, there is a README.PDF (current) that contain details about endpoints, model structure and few points that were considered to complete the assignment. Addition to README.pdf, there is a video that demonstrates the working of the functions like adding/delete customer/product etc. There is another zip archive which is the source code of my application.