Inventory API – Django

In this challenge, you have to create an inventory API in Django Rest Framework.

Step - 1:

To install all the required packages, open Terminal, go to **wings-t4-miniproject-inventory-api** folder, type "**python3 -m pip install -r requirements.txt**" (or) "**sh install.sh**".

Step - 2:

Models- Create models with appropriate fields as mentioned below.

Field Name	**Type**	**Primary key**	**Comments**
id	integer	yes	autoincrement
name	string	no	
category	string	no	
price	integer	no	
quantity	integer	no	
barcode	integer	no	Unique=true

The following API needs to be implemented:

1. adding new item- POST request should be created to add a new item. The API endpoint would be **/inventory/items/**. The request body contains the details of the item. The HTTP response should be 201.

2. Deleting any item by id-DELETE request to endpoint **inventory/items/**. The request body would contain the id of the item that needs to be deleted. The HTTP response should be 204. If item is deleted, return an empty array.

3. Editing the item -PUT request to endpoint **/inventory/items/**. The route would contain the id of the item that needs to be updated and the request body contains the information that needs to be updated. The HTTP response should be 200 in case the item to be edited does not exist, return 400.

4. Getting all items-GET request to endpoint **/inventory/items/** should return all the items in the system. The HTTP response code should be 200.

5. Getting all items in the category-GET request to endpoint **/items/query/{category}** should return the entire list of items in this category. The HTTP response code should be 200. If no items exist return in this category, return an empty array.
6. Getting all items sorted as per the price in descending order-GET request to endpoint **/items/sort** should return the entire list of items in descending price order. The HTTP response code should be 200.

Example:

```
{
    "name": "Shirt",
    "category": "Top wear",
    "price": 500,
    "discount":50,
    "quantity": 10,
    "barcode": 12345678
}
```

Step - 3:

- To run the application: Open terminal, go **to wings-t4-miniproject-inventory-api** folder, type as "**sh run.sh**" or "**python3 manage.py makemigrations** "**python3 manage.py migrate-run- syncdb**" "**python3 manage.py runserver**".
- **To test the application:** Open terminal go to **wings-t4-miniproject-inventory-api** folder, type as "**sh test.sh**".
- Check the application for syntax error before you click **submit**.
- **To preview the application:** Open the browser and type http://localhost:8000/in address bar and hit enter.

Click **Submit** to complete the test.