

# *eCommerce Api Project*

## Introduction to My Django E-commerce Project

In the fast-paced digital era, e-commerce has become an indispensable part of our lives. It's not just a means of buying and selling; it's a gateway to a world of products and services available at our fingertips. In this context, I am thrilled to introduce my Django e-commerce project, a dynamic and robust online platform designed to cater to the modern shopping needs of today's consumers.

### *Project Overview:*

My Django e-commerce project is a culmination of my passion for web development and my desire to create an efficient, user-friendly, and feature-rich online shopping experience. Leveraging the power of Django, a high-level Python web framework, I have crafted a sophisticated e-commerce solution that embodies the following key principles:

**Product Catalog:** The platform boasts an extensive product catalog, categorizing items for easy navigation. Each product is presented with detailed descriptions, high-quality images, and pricing information.

**Shopping Cart and Checkout:** Customers can add items to their shopping carts, review their selections, and proceed to a secure and straightforward checkout process.

**Admin Panel:** As an administrator, you have full control over the platform. You can manage products, inventory, customer orders, and monitor website analytics through a user-friendly admin panel.

**Search and Filters:** An advanced search functionality and product filters make it easy for customers to find the products they're looking for quickly.

**Scalability:** The architecture is designed to be scalable, allowing for easy expansion as your business grows.

## **Admin Features:**

- **Add Product**

- Update the product
- Delete The Product
- Product List View

### User Features:

- Product List
- Add to Cart
- Place to Order
- Filter the product price and category

### Post Man Screenshots

- User Registration:



Postman interface showing a GET request to `http://localhost:8000/api/products/`. The response body is displayed in JSON format:

```
{
  "id": 2,
  "name": "Update product iphone 11",
  "description": "update description of product iphone 11",
  "price": "100.00",
  "quantity": 12,
  "categories": [
    1
  ],
  "image": "/static/product_images/Screenshot_2023-07-29_114423.png"
},
{
  "id": 3,
  "name": "Iphone 11",
  "description": "description Iphone 11",
  "price": "1000000.00",
  "quantity": 5
}
```

The status is 200 OK, Time: 95 ms, Size: 1.04 KB.

## • Add To Cart:

Postman interface showing a POST request to `http://localhost:8000/cart/`. The request body is displayed in JSON format:

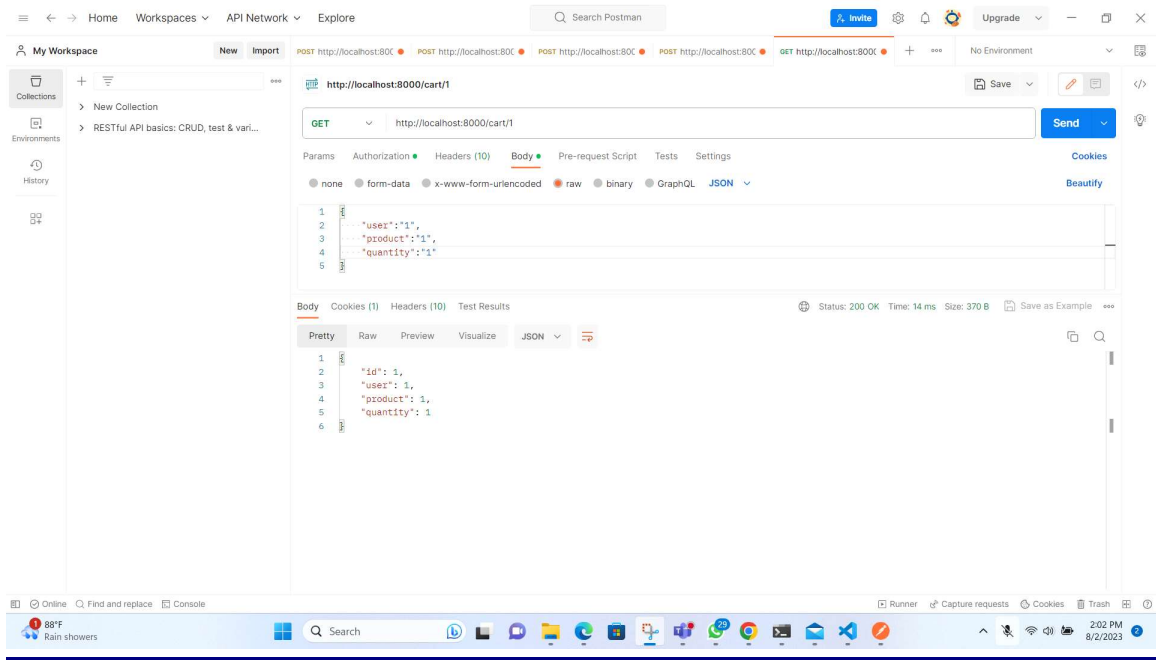
```
{
  "user": "1",
  "product": "1",
  "quantity": "1"
}
```

The response body is displayed in JSON format:

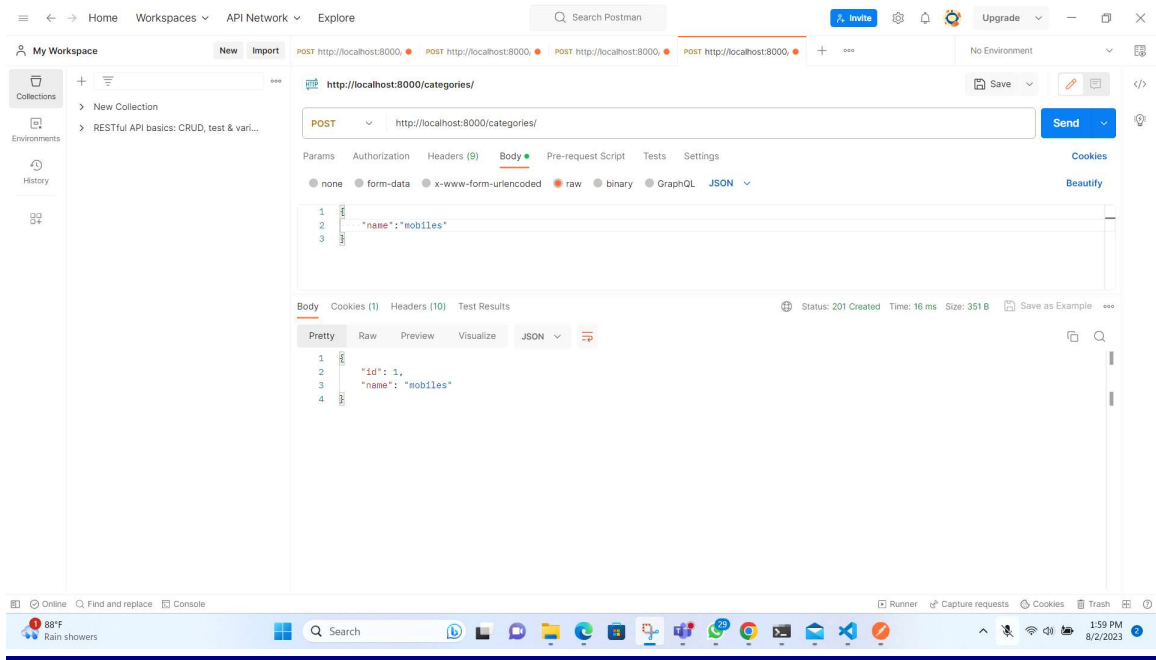
```
{
  "id": 1,
  "user": 1,
  "product": 1,
  "quantity": 1
}
```

The status is 201 Created, Time: 25 ms, Size: 368 B.

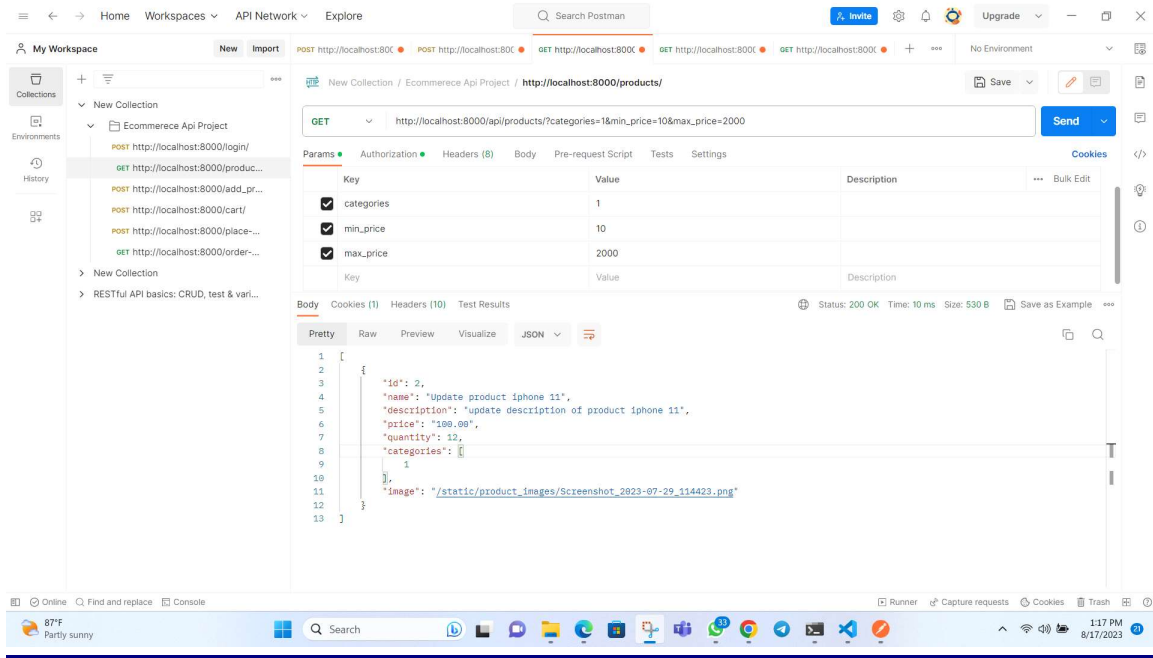
## • Cart Detail View:



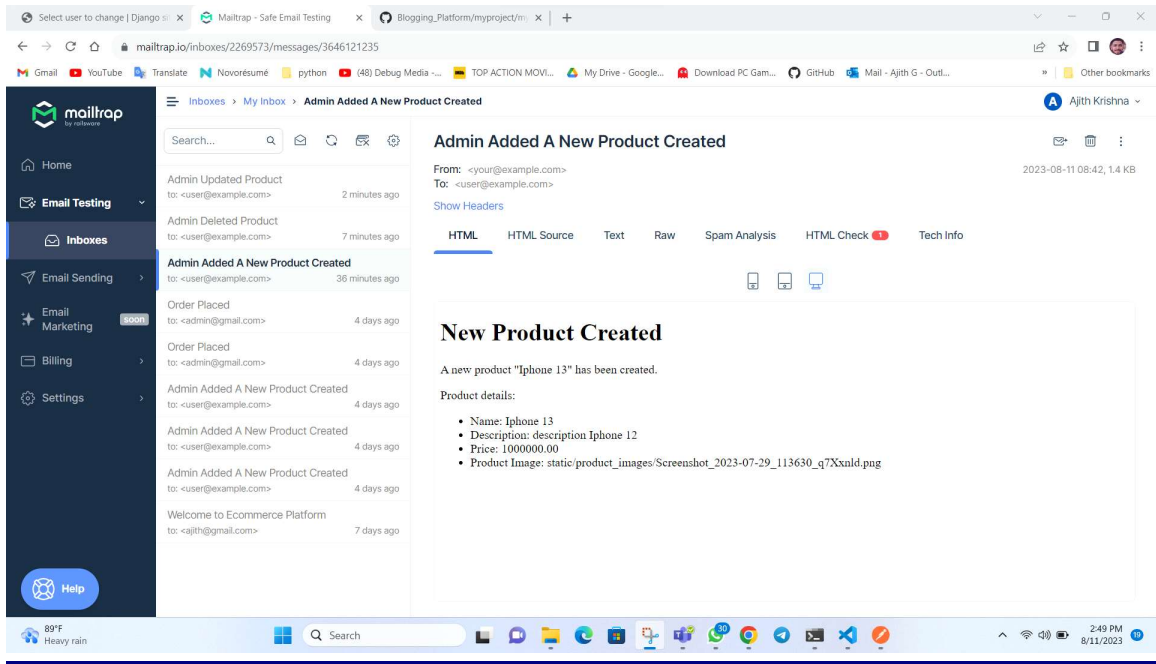
## • Create Category:



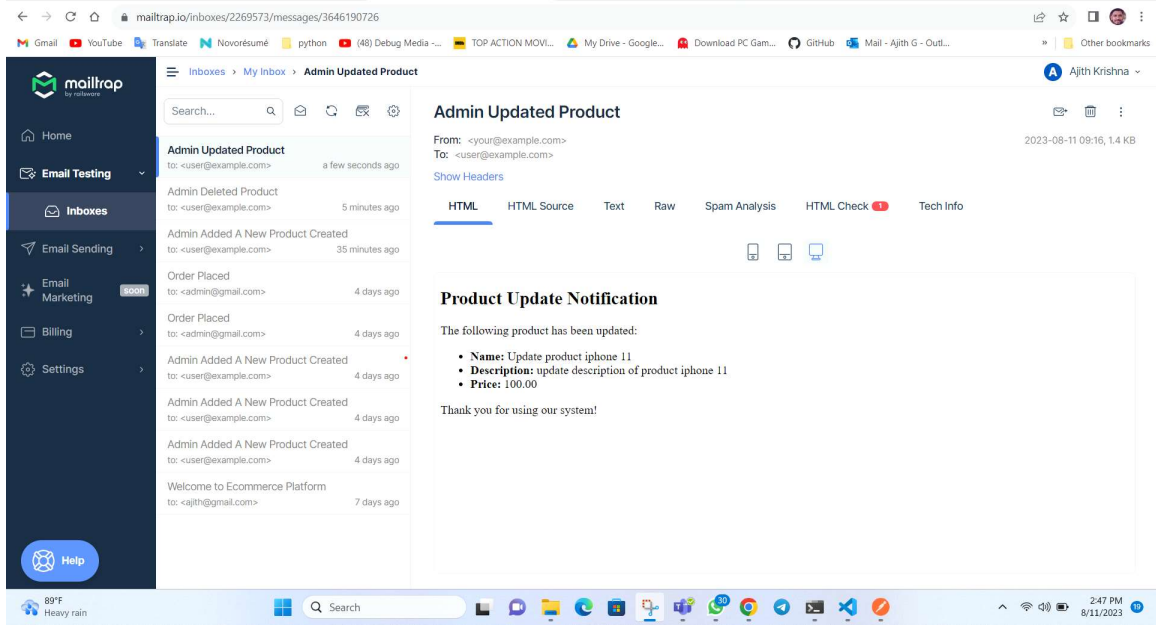
- Filter The Product by Price range and category:



- Admin Add New Product Mailtrap View:

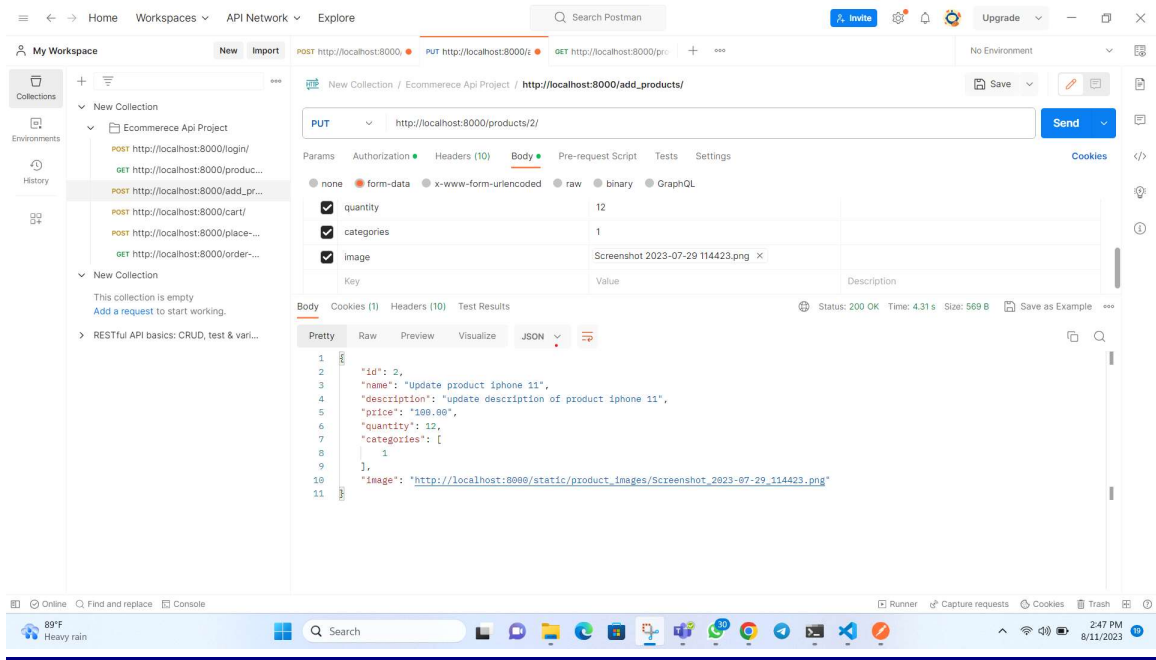


## • Admin Update Product Mailtrap View:

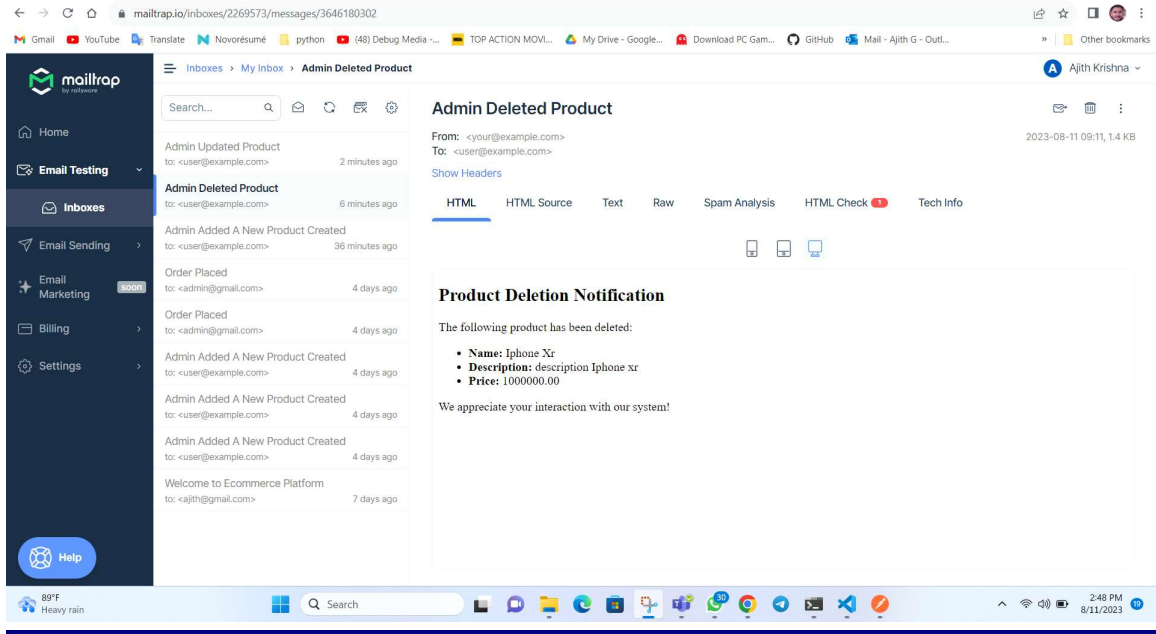




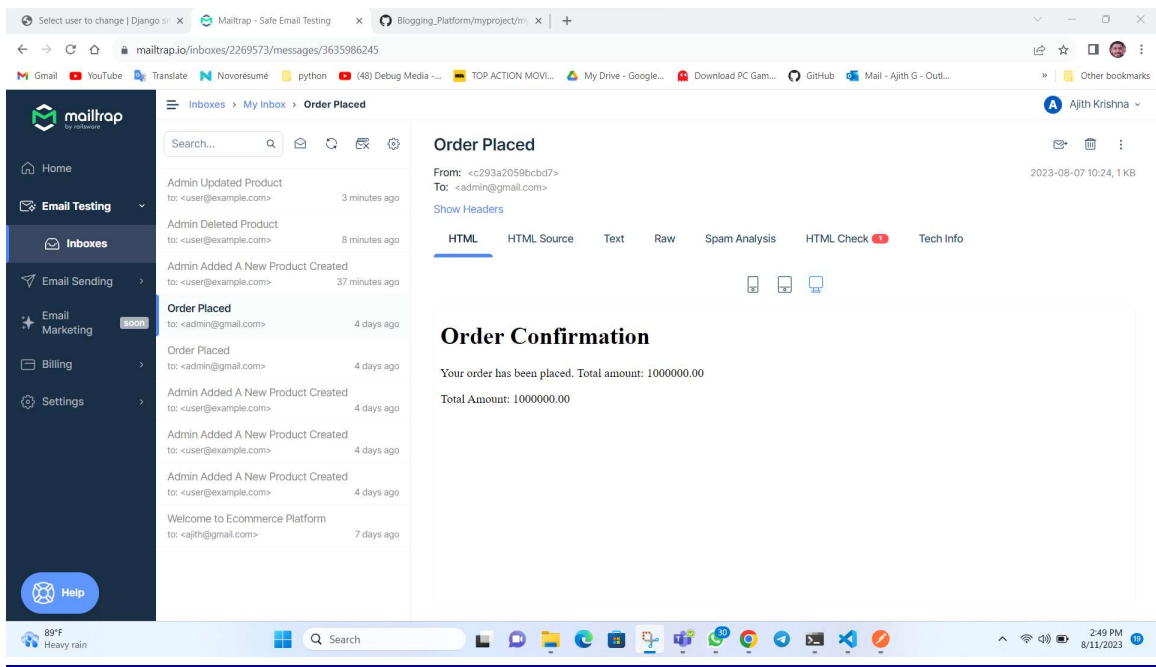
- Admin Update Product Postman View:



- Admin Delete The product Mailtrap View:

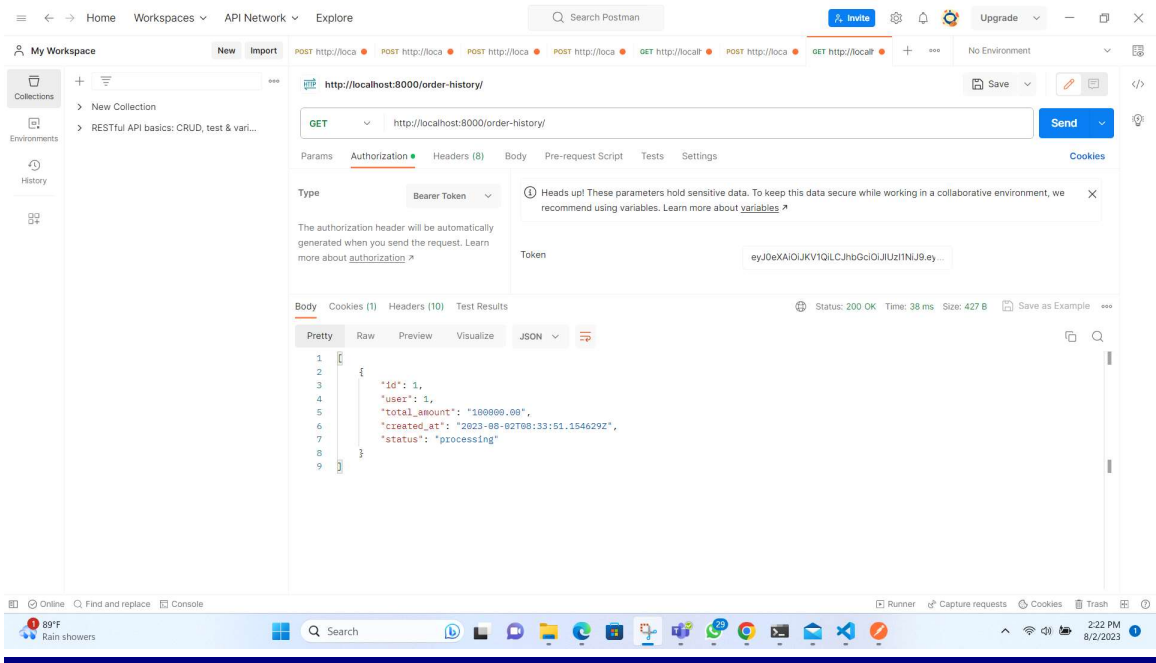


## • Order Placed Mailtrap View:



## • User Order History Postman

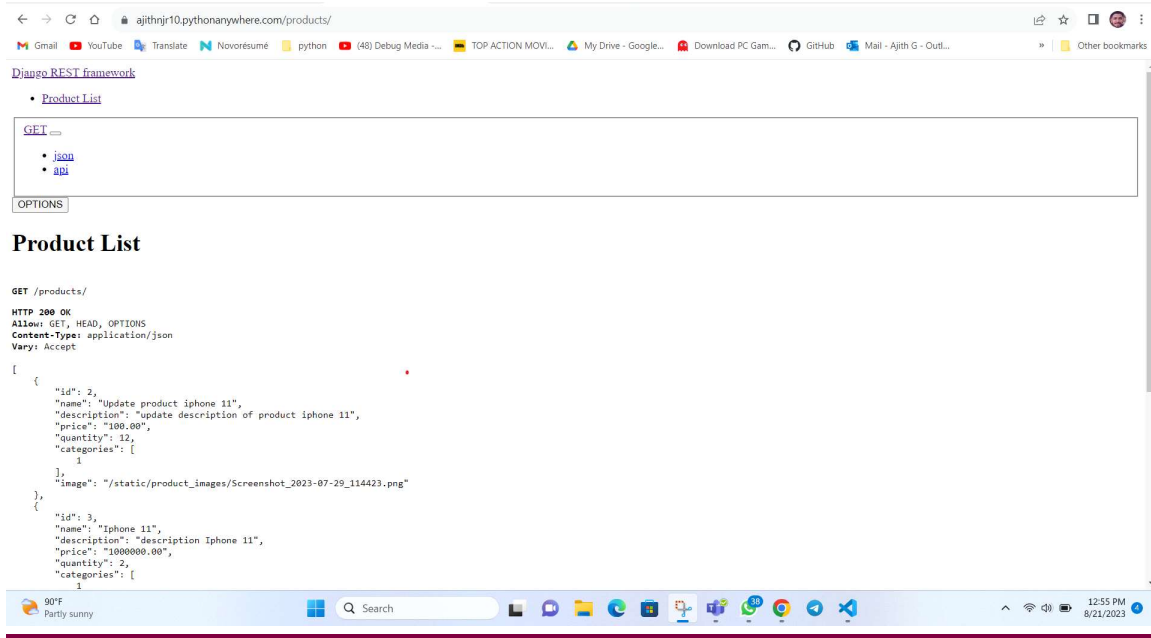
## View:



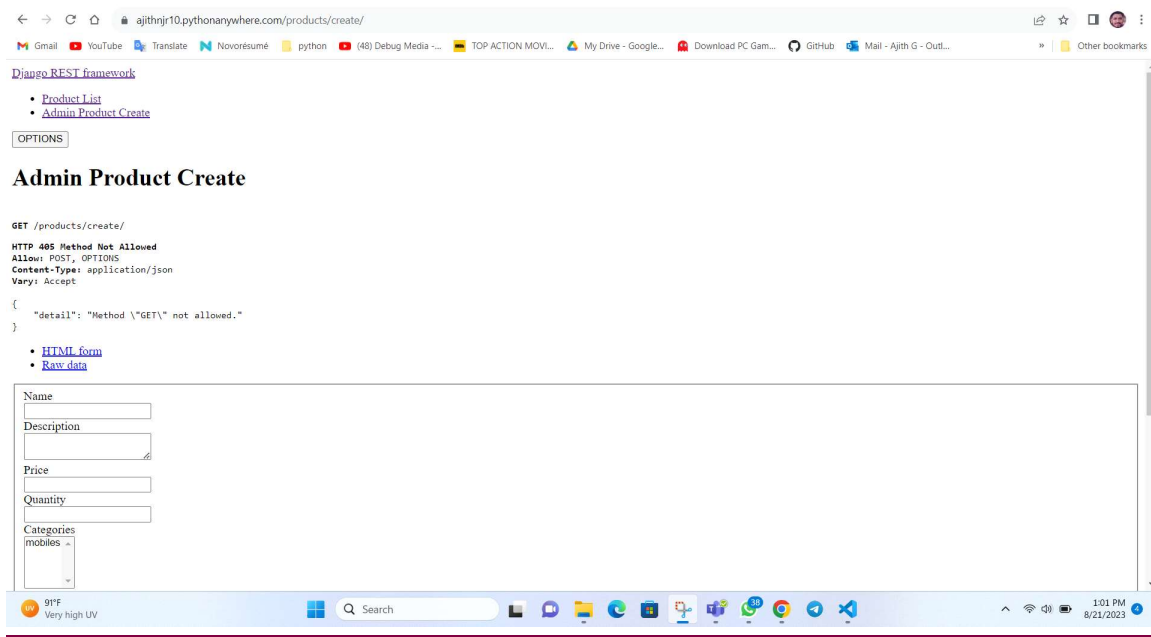
## PythonAnywhere screenshots

- Register Page:





## • Admin Add New product Page:



## • Category List Page:

ajithnjr10.pythonanywhere.com/categories/

GET

- json
- api

OPTIONS

## Category List

GET /categories/  
HTTP 200 OK  
Allow: GET, POST, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

```
[  
  {  
    "id": 1,  
    "name": "mobiles"  
  }  
]
```

Media type:  
application/json

Content:

POST

91°F Partly sunny 12:59 PM 8/21/2023







