**1. Task Description**

Create a **Django-based web application** that allows users to **filter product listings dynamically** using:

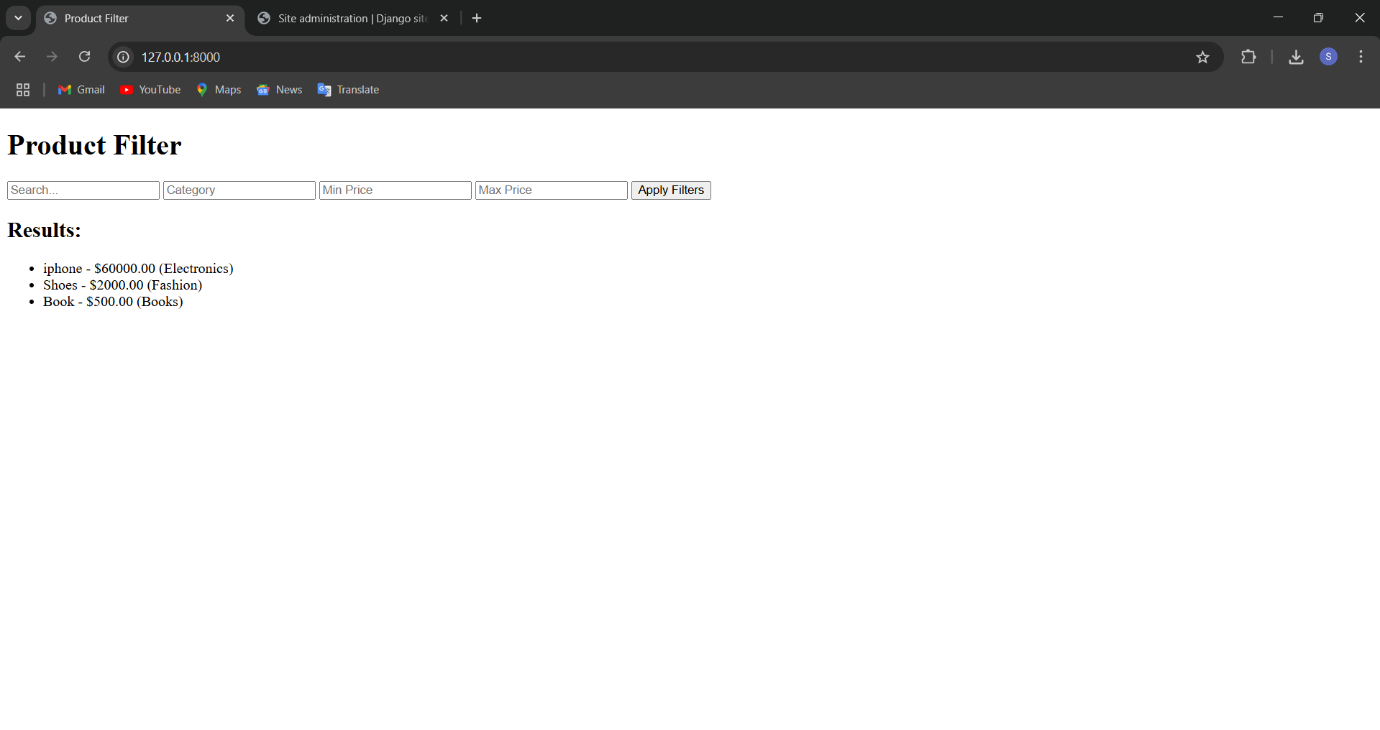
* Keyword search (Q object for name/description/category)
* Price range (minimum and maximum)
* Category matching

The filter is implemented using **Django’s ORM with Q objects** for flexible and dynamic querying. The product data is displayed on the homepage. Users can also add products via the admin panel or shell. The project is fully functional with filtering logic applied in real-time.

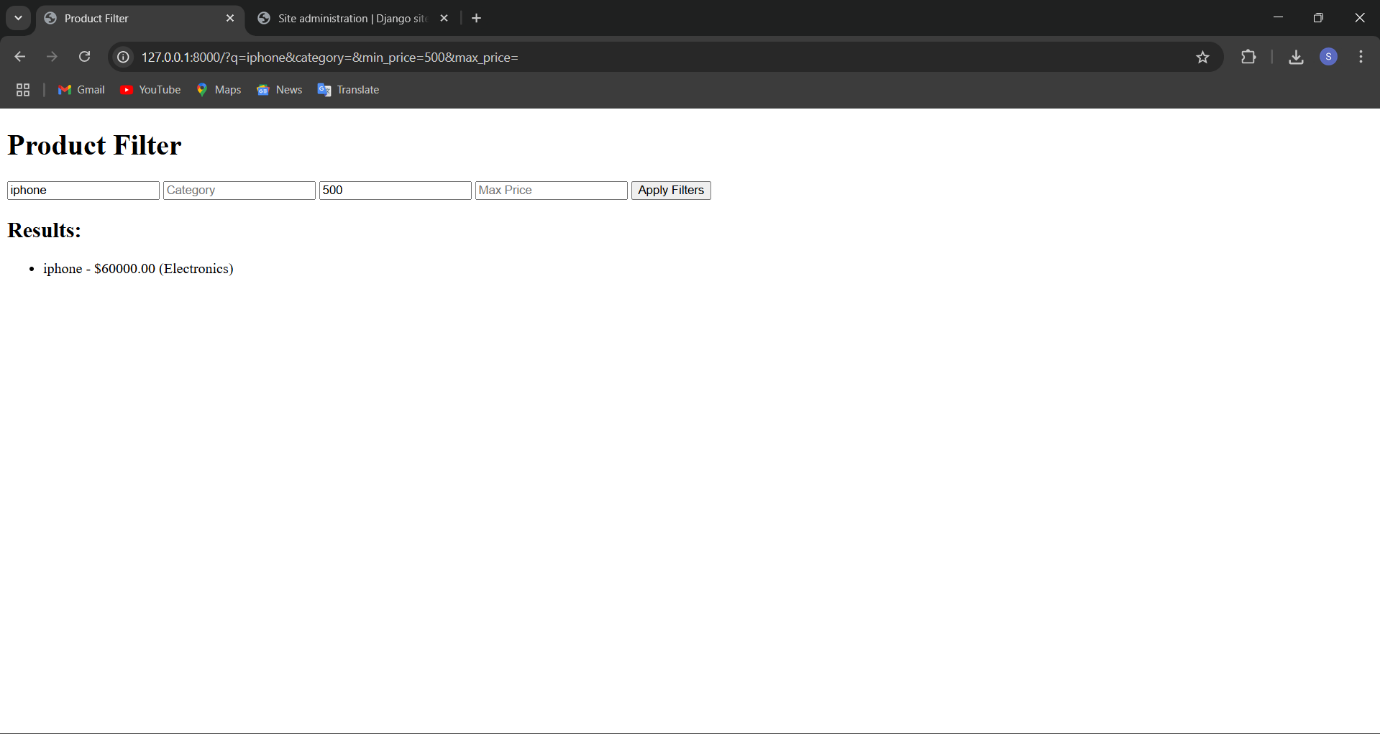
This is a live Django web application that starts on http://127.0.0.1:8000/.

**2. Task Output Screenshot**

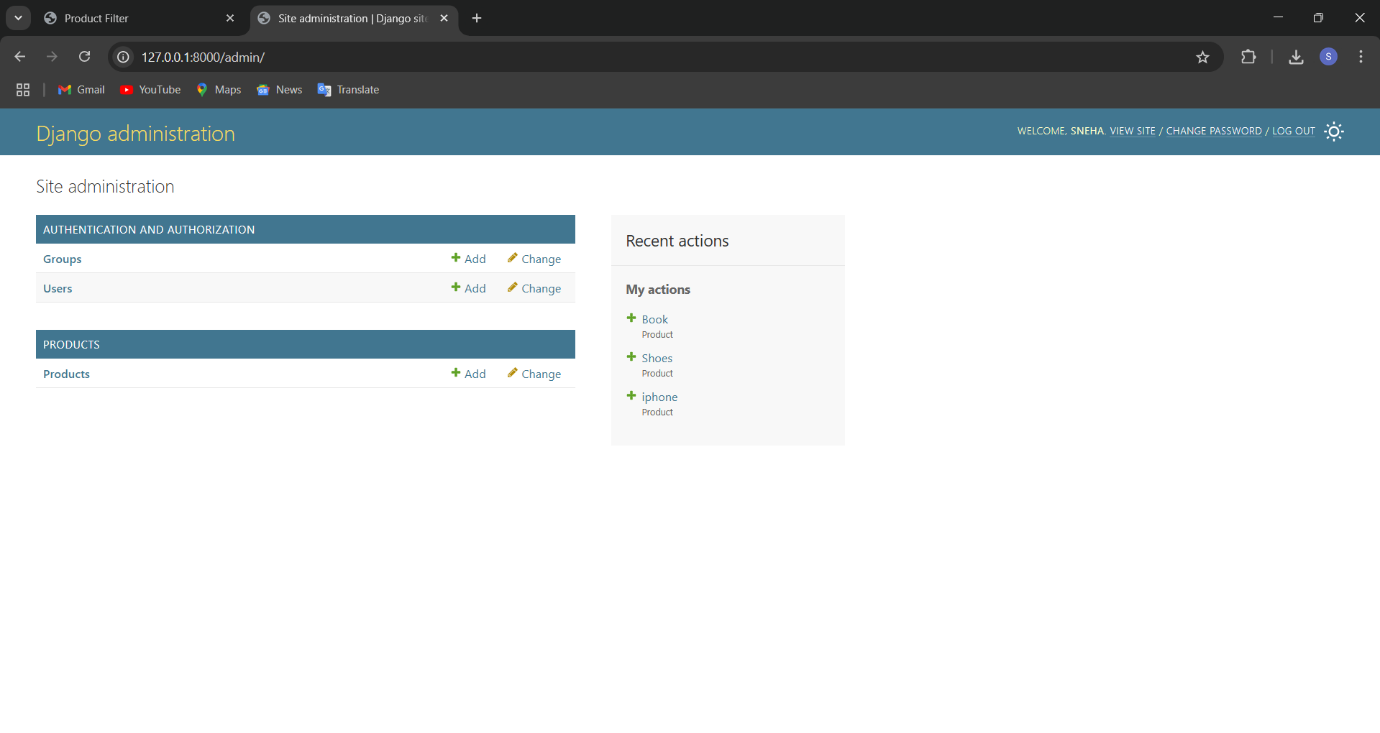
* **Screenshot 1:** Homepage with all products listed



* **Screenshot 2:** Page after filtering by category



* **Screenshot 3:** Django Admin panel showing product entries



**3. Widget/Algorithm Used in Task**

* **Q Object (django.db.models.Q)** – Enables dynamic filtering across multiple fields with OR/AND logic.
* **Price Range Filtering** – Handled using form inputs and Django ORM (price\_\_gte, price\_\_lte).
* **HTML Form** – Used for capturing search text and numeric range.
* **Bootstrap** – For frontend styling .
* **Django Admin** – Used to easily add/manage product entries.
* **SQLite** – Default Django database to store product data.