

# Rapport de projet de fin de module NoSql



**Fait par:**

-> **SOUFIANI Assia**

-> **MECHHEDAN Insafe**

**Encadré par:**

-> **Prof . EL ACHAK Lotfi**

## I. Introduction:

Le fait d'utiliser des images afin de comprendre des données existe depuis des siècles, allant des cartes et diagrammes au 17<sup>e</sup> siècle jusqu'à l'invention du graphique circulaire au début du 19<sup>e</sup> siècle. Plusieurs décennies plus tard, Charles Minard marqua l'histoire des graphiques statistiques lorsqu'il cartographia l'invasion de la Russie par Napoléon. Sa carte représentait la taille de l'armée et le chemin emprunté par Napoléon pour se retirer de Moscou, ainsi qu'une échelle des températures accompagnées de leur date, pour une compréhension plus immersive des événements.

Pourtant c'est la technologie qui a véritablement propulsé en avant la visualisation de données. Les ordinateurs ont rendu possible le traitement de grandes quantités de données à une vitesse éclair. Aujourd'hui, la visualisation de données (ou Dataviz) est un savant mélange de science et d'art, qui évolue rapidement et va assurément bouleverser le paysage des entreprises dans les années à venir.

La visualisation de données est la présentation de données sous forme graphique. Elle permet de présenter les données analytiques de manière visuelle, afin de mieux comprendre des concepts complexes ou d'identifier de nouvelles tendances. Grâce à la visualisation interactive, il est possible de pousser le concept encore plus loin en utilisant cette technologie pour explorer en profondeur les tableaux et graphiques, en modifiant interactivement les données que vous voyez et la façon dont elles sont traitées.\

## II. L'objectif du projet:

L'objectif principal de notre projet est la réalisation d'une plateforme web de type single page application, pour les opérations d'un ETL " Extract, Transform and Load " et Data Visualisation / Analysis depuis une base de données NoSql (Not Only SQL).

## III. La description du projet:

Dans notre projet nous avons tout d'abord collecté les données d'après les deux sites Amazon et Flipkart.

L'étape suivante c'est le pre-processing et pre-traitement des données .

Puis nous avons commencé à travailler sur le backend en reliant django,mongodb,elasticsearch..

Et en fin l'étape de visualisation de données.

#### IV. Les outils utilisés:



**Langage de programmation utilise dans le backend.**

# django

**Django est un framework Python de haut niveau, permettant un développement rapide de sites internet, sécurisés, et maintenables**



**Angular est une plateforme de développement, construite sur TypeScript**



**permet de créer un objet Python représentant un document HTML. Une manière de créer un tel objet est de passer simplement le code HTML.**



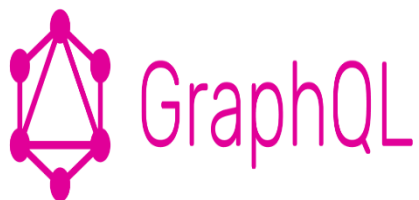
**permet de créer un objet Python représentant un document HTML. Une manière de créer un tel objet est de passer simplement le code HTML.**



**MongoDB est une base de données NoSQL  
orientée documents.le code HTML.**



**Elasticsearch est un moteur de recherche  
et d'analyse RESTful distribué, conçu pour  
répondre à une multitude de cas  
d'utilisation**



**GraphQL est un langage de requête et un  
environnement d'exécution côté serveur  
pour les interfaces de programmation  
d'application (API).**

## V. Scraping des données:

Nous avons pu scraper les données à l'aide de BeautifulSoup.

### Scraping Flipkart:

```
in.py x dashProject.py x scraper_data.csv x scraping.py x produits.csv x avocado.csv x dashbord.py x
import requests
import pandas as pd
from bs4 import BeautifulSoup
import csv
from itertools import zip_longest

print("type what you want to search")
subject = input()
links_title = []
description_title = []
Prix = []
Links = []
Products = []
num_page = 1
while True:
    resultat = requests.get(
        f"https://www.flipkart.com/search?q={subject}&otracker=search&otracker1=search&marketplace=FLIPKART&as-sh
    )
    resultats = requests.get(
        f"https://www.flipkart.com/search?q={subject}&otracker=search&otracker1=search&marketplace=FLIPKART&as-sh
    )

    src = resultat.content
    srcs = resultats.content
    soup = BeautifulSoup(src, "lxml")
```

```
Manufacturer;Model Name;Category;Screen Size;Screen;CPU;RAM;Storage;GPU;Operating System;;Weight;Price (Euros)
Apple;MacBook Pro;Ultrabook;"13.3"";IPS Panel Retina Display 2560x1600;Intel Core i5 2.3GHz;8GB;128GB SSD;Intel Iris Pl
Apple;Macbook Air;Ultrabook;"13.3"";1440x900;Intel Core i5 1.8GHz;8GB;128GB Flash Storage;Intel HD Graphics 6000;macOS;;
HP;250 G6;Notebook;"15.6"";Full HD 1920x1080;Intel Core i5 7200U 2.5GHz;8GB;256GB SSD;Intel HD Graphics 620;No OS;;1.86k
Apple;MacBook Pro;Ultrabook;"15.4"";IPS Panel Retina Display 2880x1800;Intel Core i7 2.7GHz;16GB;512GB SSD;AMD Radeon Pr
Apple;MacBook Pro;Ultrabook;"13.3"";IPS Panel Retina Display 2560x1600;Intel Core i5 3.1GHz;8GB;256GB SSD;Intel Iris Pl
Acer;Aspire 3;Notebook;"15.6"";1366x768;AMD A9-Series 9420 3GHz;4GB;500GB HDD;AMD Radeon R5;Windows;;2.1kg;400
Apple;MacBook Pro;Ultrabook;"15.4"";IPS Panel Retina Display 2880x1800;Intel Core i7 2.2GHz;16GB;256GB Flash Storage;Int
Apple;Macbook Air;Ultrabook;"13.3"";1440x900;Intel Core i5 1.8GHz;8GB;256GB Flash Storage;Intel HD Graphics 6000;macOS;;
Asus;ZenBook UX430UN;Ultrabook;"14.0"";Full HD 1920x1080;Intel Core i7 8550U 1.8GHz;16GB;512GB SSD;Nvidia GeForce MX150;
Acer;Swift 3;Ultrabook;"14.0"";IPS Panel Full HD 1920x1080;Intel Core i5 8250U 1.6GHz;8GB;256GB SSD;Intel UHD Graphics 6
HP;250 G6;Notebook;"15.6"";1366x768;Intel Core i5 7200U 2.5GHz;4GB;500GB HDD;Intel HD Graphics 620;No OS;;1.86kg;393,9
HP;250 G6;Notebook;"15.6"";Full HD 1920x1080;Intel Core i3 6006U 2GHz;4GB;500GB HDD;Intel HD Graphics 520;No OS;;1.86kg;
Apple;MacBook Pro;Ultrabook;"15.4"";IPS Panel Retina Display 2880x1800;Intel Core i7 2.8GHz;16GB;256GB SSD;AMD Radeon Pr
Dell;Inspiron 3567;Notebook;"15.6"";Full HD 1920x1080;Intel Core i3 6006U 2GHz;4GB;256GB SSD;AMD Radeon R5 M430;Windows;
Apple;"MacBook 12"";Ultrabook;"12.0"";IPS Panel Retina Display 2304x1440;Intel Core M m3 1.2GHz;8GB;256GB SSD;Intel HD
Apple;MacBook Pro;Ultrabook;"13.3"";IPS Panel Retina Display 2560x1600;Intel Core i5 2.3GHz;8GB;256GB SSD;Intel Iris Pl
Dell;Inspiron 3567;Notebook;"15.6"";Full HD 1920x1080;Intel Core i7 7500U 2.7GHz;8GB;256GB SSD;AMD Radeon R5 M430;Wind
Apple;MacBook Pro;Ultrabook;"15.4"";IPS Panel Retina Display 2880x1800;Intel Core i7 2.9GHz;16GB;512GB SSD;AMD Radeon Pr
Lenovo;IdeaPad 320-15IKB;Notebook;"15.6"";Full HD 1920x1080;Intel Core i3 7100U 2.4GHz;8GB;1TB HDD;Nvidia GeForce 940MX;
Dell;XPS 13;Ultrabook;"13.3"";IPS Panel Full HD / Touchscreen 1920x1080;Intel Core i5 8250U 1.6GHz;8GB;128GB SSD;Intel U
Asus;Vivobook E200HA;Netbook;"11.6"";1366x768;Intel Atom x5-28350 1.44GHz;2GB;32GB Flash Storage;Intel HD Graphics 400;V
Lenovo;Legion Y520-15IKBN;Gaming;"15.6"";IPS Panel Full HD 1920x1080;Intel Core i5 7300HQ 2.5GHz;8GB;128GB SSD + 1TB HDD
```

```
jupyter amazon_scraping Dernière Sauvegarde : Hier à 10:23 (auto-sauvegardé)

File Edit View Insert Cell Kernel Widgets Help

[+] [-] [↺] [↻] [↱] [↲] [↵] [⏮] [⏭] [⏪] [⏩] [Code] [Run]

Entrée [23]: import csv
from pymongo import MongoClient
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from msedge.selenium_tools import Edge, EdgeOptions

Entrée [24]: driver = webdriver.Chrome('C:/chromedriver.exe')

Entrée [25]: def get_url(search_text):
    """Generate a url from search text"""
    template = 'https://www.amazon.com/s?k={}&ref=nb_sb_noss_1'
    search_term = search_text.replace(' ', '+')

    # add term query to url
    url = template.format(search_term)

    # add page query placeholder
    url += '&page={}'

    return url

Entrée [26]: #amazon
def extract_record(item):
    """Extract and return data from a single record"""

28]: # run program
main('laptop')

('SGIN Laptop 15.6 Inch 12GB DDR4 512GB SSD, Windows 11 Laptops with Intel Celeron N4500, FHD 1920x1080, Dual Band WiFi, 2xUSB 3.0, Up to 2.8ghz, Bluetooth 4.2, Supports 512GB TF Card Expansion, Gray', '$419.99', '', '', 'https://www.amazon.com/gp/sllredirect/picassoRedirect.html/ref=pa_sp_atf_aps_sr_pg1_1?ie=UTF8&adId=A04805432ZB92100YNI1&url=%2FSGIN-1920x1080-Bluetooth-Supports-Expansion%2Fdp%2FB09XQZM564%2Fref%3Dsr_1_1_sspa%3Fkeywords%3DLaptop%26qid%3D1654334600%26sr%3D8-1-spons%26psc%3D1&qualifier=1654334600&id=1404998964440953&widgetName=sp_atf')
('Acer Nitro 5 AN515-57-79TD Gaming Laptop | Intel Core i7-11800H | NVIDIA GeForce RTX 3050 Ti Laptop GPU | 15.6" FHD 144Hz IPS Display | 8GB DDR4 | 512GB NVMe SSD | Killer Wi-Fi 6 | Backlit Keyboard', '$998.82', '', '', 'https://www.amazon.com/gp/sllredirect/picassoRedirect.html/ref=pa_sp_atf_aps_sr_pg1_1?ie=UTF8&adId=A05951653GBLL6ZRENUL4&url=%2FAcer-AN515-57-79TD-i7-11800H-GeForce-Keyboard%2Fdp%2FB09R65RN43%2Fref%3Dsr_1_2_sspa%3Fkeywords%3DLaptop%26qid%3D1654334600%26sr%3D8-2-spons%26psc%3D1&qualifier=1654334600&id=1404998964440953&widgetName=sp_atf')
('ASUS ProArt StudioBook Pro 16 OLED Laptop, 16" 3840x2400 OLED Display, Intel Xeon W-11955M, 64GB DDR4, 4TB PCIe SSD, Nvidia RTX A5000, Windows 11 Pro, W7600H5A-XH99, Star Black', '$4,999.99', '', '', 'https://www.amazon.com/ASUS-StudioBook-3840x2400-W-11955M-W7600H5A-XH99/dp/B09NWFSDVN/ref=sr_1_3?keywords=laptop&qid=1654334600&sr=8-3')
('2022 Newest HP 15.6" HD Laptop Computer, Intel Celeron Quad-Core N4120(up to 2.6GHz), 8GB DDR4 RAM, 128GB SSD, HDMI, Bluetooth, Webcam, USB-C, RJ45 Ethernet, Windows 11S, Silver, JVK Mousepad', '$349.00', '', '', 'https://www.amazon.com/HP-Laptop-Computer-Quad-Core-Bluetooth/dp/B09YCS55K4/ref=sr_1_4?keywords=laptop&qid=1654334600&sr=8-4')
('MSI Raider GE66 15.6" UHD 4K 120Hz Gaming Laptop Intel Core i9-12900HK RTX3080TI 32GBDDR5 1TB NVMe SSD Win11', '$3,799.00', '', '', 'https://www.amazon.com/MSI-Raider-GE66-i9-12900HK-RTX3080TI/dp/B09RB91BBT/ref=sr_1_5?keywords=laptop&qid=1654334600&sr=8-5')
('MSI Raider Forge 15.6" QHD 165Hz Gaming Laptop Intel Core i9-12900HK RTX3080TI 32GBDDR5 1TB NVMe SSD Win11', '$3,799.00', '', '', 'https://www.amazon.com/MSI-Raider-Forge-i9-12900HK-RTX3080TI/dp/B09RB91BBT/ref=sr_1_6?keywords=laptop&qid=1654334600&sr=8-6')
```



## VI. Preprocessing des données:



Dans notre cas nous avons fait:

Nettoyage des données : trouvez les « Null », les valeurs manquantes et les données dupliquées. Il faut remplacer les « Null » et les valeurs manquantes par d'autres valeurs (ou les supprimer) .

Décomposition des données : les colonnes de texte contiennent parfois plus d'une information ; nous devons donc les diviser en autant de colonnes dédiées que nécessaire. Si certaines colonnes représentent des catégories, convertissez-les en colonnes de type catégorie.

Exemples:

```
jupyter preprocessing Dernière Sauvegarde : il y a 2 heures (auto-sauvegardé)
File Edit View Insert Cell Kernel Widgets Help
+ - < > Exécuter Code
Entrée [31]: import pandas as pd
Entrée [32]: df = pd.read_csv("C:/Users/SOUFIANI Assia/Desktop/resultats.csv")
Entrée [33]: df.tail()
Out[33]:
  Description Price Rating ReviewCount Url
455 USB C Hub-YOCOWOCO Dongle USB C Adapter for Ma... $16.99 NaN NaN https://www.amazon.com/gp/siredirect/picassoRe...
456 SAMSUNG 15.6" Galaxy Book2 Pro Laptop Computer... $1,349.99 NaN NaN https://www.amazon.com/gp/siredirect/picassoRe...
457 Vanja 5 in 1 SD Card Reader USB Hub 3.0, USB 3... $10.99 NaN NaN https://www.amazon.com/gp/siredirect/picassoRe...
458 Packard Bell CloudBook 10.1 inch Windows 2 in ... $169.99 NaN NaN https://www.amazon.com/gp/siredirect/picassoRe...
459 2021 Newest Lenovo IdeaPad Laptop, 15.6" HD To... $569.00 NaN NaN https://www.amazon.com/gp/siredirect/picassoRe...
Entrée [34]: df.shape
Out[34]: (460, 5)
Entrée [35]: df.isnull().sum()
Out[35]: Description    0
         Price        0
         Rating    460
         ReviewCount  460
         Url          0
```



jupyter preprocessing Dernière Sauvegarde : il y a 2 heures (auto-sauvegardé)

File Edit View Insert Cell Kernel Widgets Help

Entrée [36]: `del df['Rating']`

Entrée [37]: `del df['ReviewCount']`

Entrée [38]: `df`

Out[38]:

|     | Description                                       | Price      | Uri  |
|-----|---|------------|--|
| 0   | SGIN Laptop 15.6 Inch, 12GB DDR4 512GB SSD Win... | \$499.99   | https://www.amazon.com/gp/sllredirect/picassoRe... |
| 1   | SGIN Laptop 15.6 Inch 12GB DDR4 512GB SSD, Win... | \$419.99   | https://www.amazon.com/gp/sllredirect/picassoRe... |
| 2   | ASUS ProArt StudioBook Pro 16 OLED Laptop, 16"    | \$4,999.99 | https://www.amazon.com/ASUS-StudioBook-3840x24...  |
| 3   | 2022 Newest HP 15.6" HD Laptop Computer, Intel... | \$349.00   | https://www.amazon.com/HP-Laptop-Computer-Quad...  |
| 4   | MSI Raider GE66 15.6" UHD 4K 120Hz Gaming Lapt... | \$3,799.00 | https://www.amazon.com/MSI-Raider-GE66-i9-1290...  |
| ... | ...   | ...        | ...  |
| 455 | USB C Hub-YOCOWOCO Dongle USB C Adapter for Ma... | \$16.99    | https://www.amazon.com/gp/sllredirect/picassoRe... |
| 456 | SAMSUNG 15.6" Galaxy Book2 Pro Laptop Computer... | \$1,349.99 | https://www.amazon.com/gp/sllredirect/picassoRe... |
| 457 | Vanja 5 in 1 SD Card Reader USB Hub 3.0, USB 3... | \$10.99    | https://www.amazon.com/gp/sllredirect/picassoRe... |
| 458 | Packard Bell CloudBook 10.1 inch Windows 2 in ... | \$169.99   | https://www.amazon.com/gp/sllredirect/picassoRe... |
| 459 | 2021 Newest Lenovo IdeaPad Laptop, 15.6" HD To... | \$569.00   | https://www.amazon.com/gp/sllredirect/picassoRe... |

460 rows x 3 columns

jupyter preprocessing Dernière Sauvegarde : il y a 2 heures (auto-sauvegardé)

File Edit View Insert Cell Kernel Widgets Help

Entrée [30]: `df.isnull().sum()`

Out[30]:

|     | Description | Price | Uri |
|-----|-------------|-------|-----|
| 0   |             |       |     |
| 1   |             |       |     |
| 2   |             |       |     |
| 3   |             |       |     |
| 4   |             |       |     |
| ... |             |       |     |
| 455 |             |       |     |
| 456 |             |       |     |
| 457 |             |       |     |
| 458 |             |       |     |
| 459 |             |       |     |

dtype: int64

Entrée [39]: `df['Description']`

Out[39]:

|     |   |
|-----|---|
| 0   | SGIN Laptop 15.6 Inch, 12GB DDR4 512GB SSD Win... |
| 1   | SGIN Laptop 15.6 Inch 12GB DDR4 512GB SSD, Win... |
| 2   | ASUS ProArt StudioBook Pro 16 OLED Laptop, 16"    |
| 3   | 2022 Newest HP 15.6" HD Laptop Computer, Intel... |
| 4   | MSI Raider GE66 15.6" UHD 4K 120Hz Gaming Lapt... |
| ... | ...   |
| 455 | USB C Hub-YOCOWOCO Dongle USB C Adapter for Ma... |
| 456 | SAMSUNG 15.6" Galaxy Book2 Pro Laptop Computer... |
| 457 | Vanja 5 in 1 SD Card Reader USB Hub 3.0, USB 3... |
| 458 | Packard Bell CloudBook 10.1 inch Windows 2 in ... |
| 459 | 2021 Newest Lenovo IdeaPad Laptop, 15.6" HD To... |

Name: Description, Length: 460, dtype: object

## VII. Structure de notre base de données

### Mise en place de la migration :

Pour se faire, on exécute les commandes suivantes :

- `python manage.py makemigrations`
- `python manage.py migrate`

The screenshot shows the MongoDB Compass interface for the database 'localhost:27017/back'. The left sidebar displays the database structure with 8 DBS and 30 COLLECTIONS. The main panel shows the 'Collections' tab with a table of collections:

| Collection Name                     | Storage size | Documents | Avg. document size | Indexes | Total index size |
|-------------------------------------|--------------|-----------|--------------------|---------|------------------|
| <code>_schema__</code>              | 20.48 kB     | 12        | 295.00 B           | 3       | 77.82 kB         |
| <code>auth_group</code>             | 4.10 kB      | 0         | 0 B                | 3       | 12.29 kB         |
| <code>auth_group_permissions</code> | 4.10 kB      | 0         | 0 B                | 5       | 20.48 kB         |
| <code>auth_permission</code>        | 20.48 kB     | 32        | 109.00 B           | 4       | 81.92 kB         |

Voilà notre collection back qui contient les données scrapeés

The screenshot shows the MongoDB Compass interface for the database 'localhost:27017/back.products\_laptopop'. The left sidebar displays the database structure with 8 DBS and 30 COLLECTIONS. The main panel shows the 'Documents' tab for the collection 'back.products\_laptopop'. The collection has 1.3k documents and 1 index. The 'Documents' tab is selected, showing a list of documents. The first document is displayed in a JSON format:

```
{
  "_id": ObjectId('629c80f60d5ad88c2050bb55'),
  "Manufacturer": "Apple",
  "Model Name": "MacBook Pro",
  "Category": "Ultrabook",
  "Screen Size": "13.3\"",
  "Screen": "IPS Panel Retina Display 2560x1600",
  "CPU": "Intel Core i5 2.3GHz",
  "RAM": "8GB",
  "Storage": "128GB SSD",
  "GPU": "Intel Iris Plus Graphics 640",
  "Operating System": "macOS",
  "Weight": "1.37kg",
  "Price (Euros)": "1339,69"
}
```

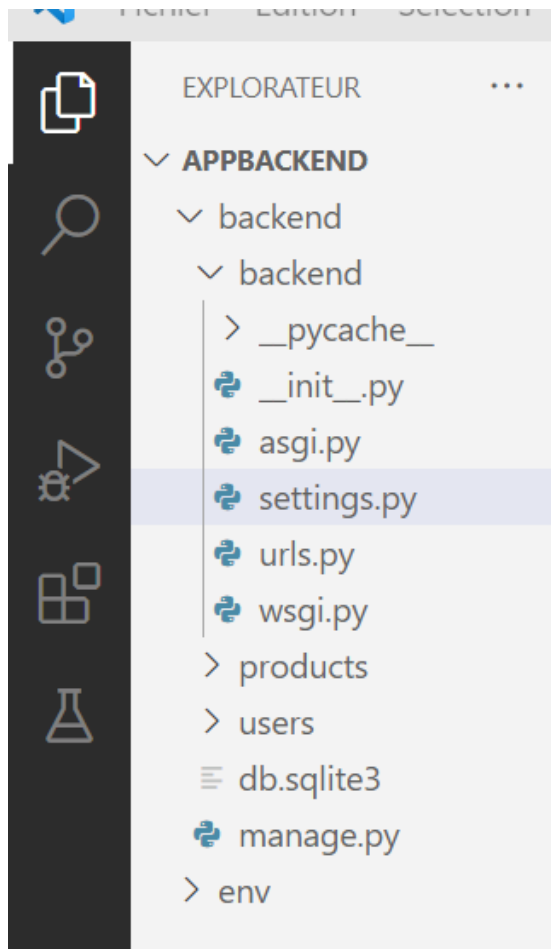
## VIII. Structure de backend

- La première chose que nous avons fait, c'est de créer et activer un environnement virtuel avec les commandes `virtualenv env` et `source env/script/activate`.

- Ensuite on installe django avec `pip install django` et on crée un projet avec : `django-admin startproject backend`

- Et maintenant on crée notre application avec : `python manage.py startapp product`.

Et dans settings -> installed app on ajoute 'products'



```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'products',  
    'users',  
    'graphene_django',  
    'django_elasticsearch_dsl',  
]
```

## IX. GraphQL

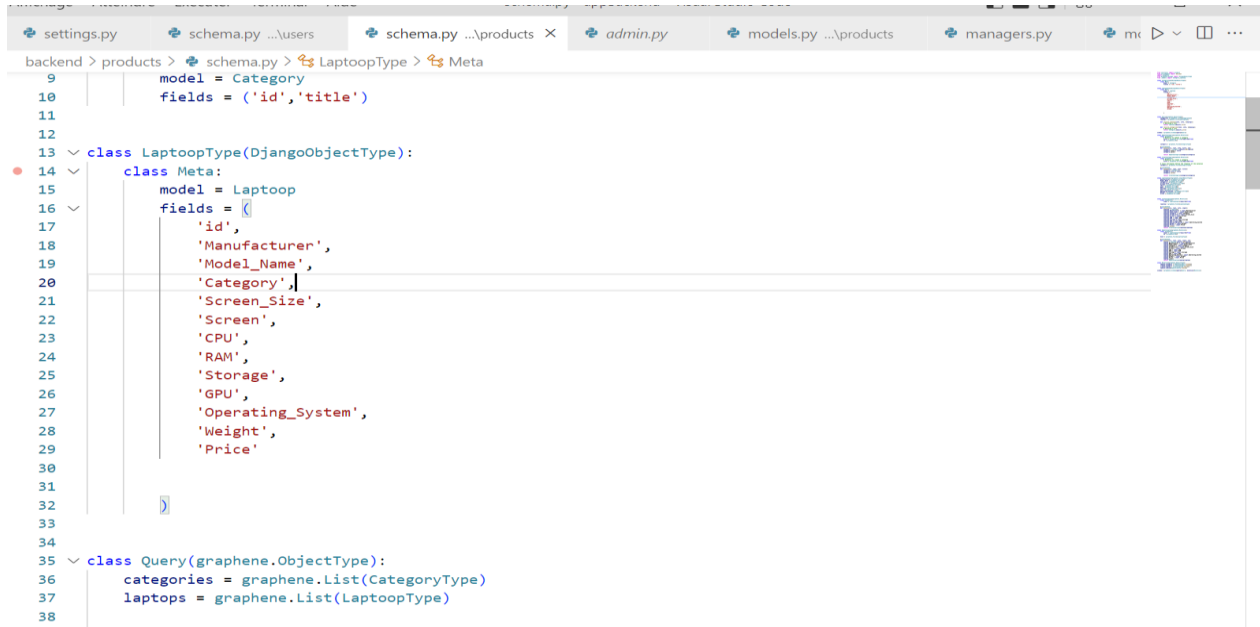
- Installation de graphql :

On met tous les 'requirements' qu'on aura besoin dans un

fichier.txt et on execute la commande :

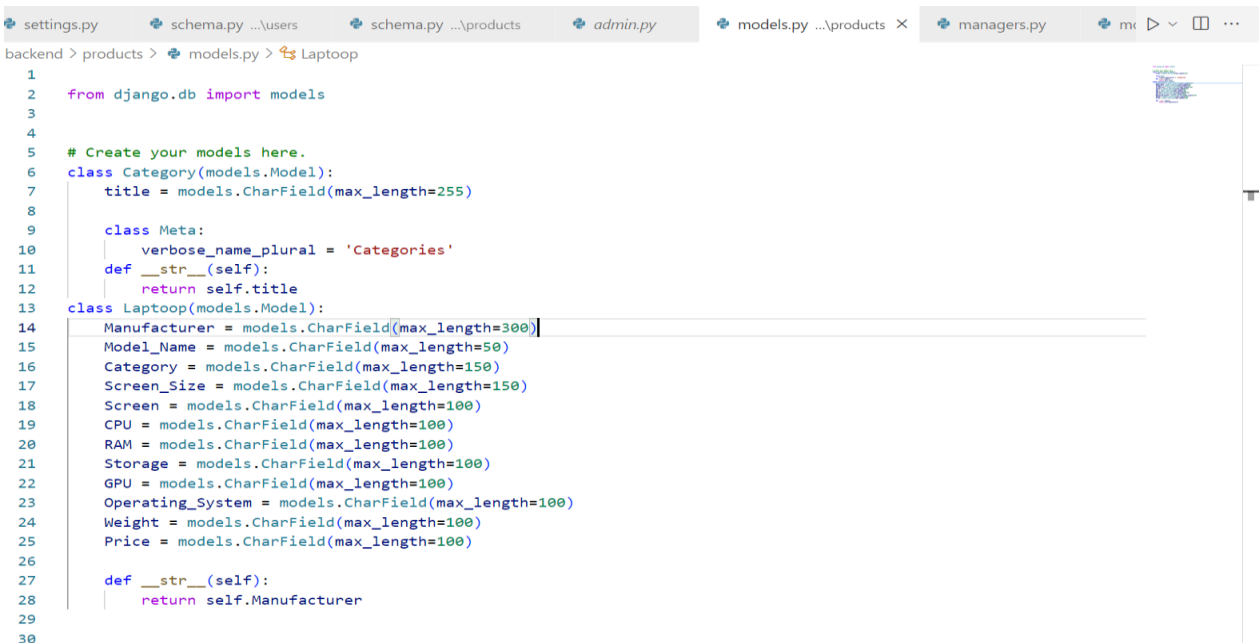
Pip install -r requirements.txt

- Maintenant on crée un schéma :



```
settings.py  schema.py ...Users  schema.py ...\products  admin.py  models.py ...\products  managers.py  mc  ▾  □  ...
backend > products > schema.py > LaptopType > Meta
9      model = Category
10      fields = ('id','title')
11
12
13  class LaptopType(DjangoObjectType):
14      class Meta:
15          model = Laptop
16          fields = (
17              'id',
18              'Manufacturer',
19              'Model_Name',
20              'Category',
21              'Screen_Size',
22              'Screen',
23              'CPU',
24              'RAM',
25              'Storage',
26              'GPU',
27              'Operating_System',
28              'Weight',
29              'Price'
30          )
31
32  )
33
34
35  class Query(graphene.ObjectType):
36      categories = graphene.List(CategoryType)
37      laptops = graphene.List(LaptopType)
38
```

- Et voici le modèle :



```
settings.py  schema.py ...Users  schema.py ...\products  admin.py  models.py ...\products  managers.py  mc  ▾  □  ...
backend > products > models.py > Laptop
1
2  from django.db import models
3
4
5  # Create your models here.
6  class Category(models.Model):
7      title = models.CharField(max_length=255)
8
9      class Meta:
10          verbose_name_plural = 'Categories'
11      def __str__(self):
12          return self.title
13  class Laptop(models.Model):
14      Manufacturer = models.CharField(max_length=300)
15      Model_Name = models.CharField(max_length=50)
16      Category = models.CharField(max_length=150)
17      Screen_Size = models.CharField(max_length=150)
18      Screen = models.CharField(max_length=100)
19      CPU = models.CharField(max_length=100)
20      RAM = models.CharField(max_length=100)
21      Storage = models.CharField(max_length=100)
22      GPU = models.CharField(max_length=100)
23      Operating_System = models.CharField(max_length=100)
24      Weight = models.CharField(max_length=100)
25      Price = models.CharField(max_length=100)
26
27      def __str__(self):
28          return self.Manufacturer
29
30
```

- Setup d'URL de GraphQL :

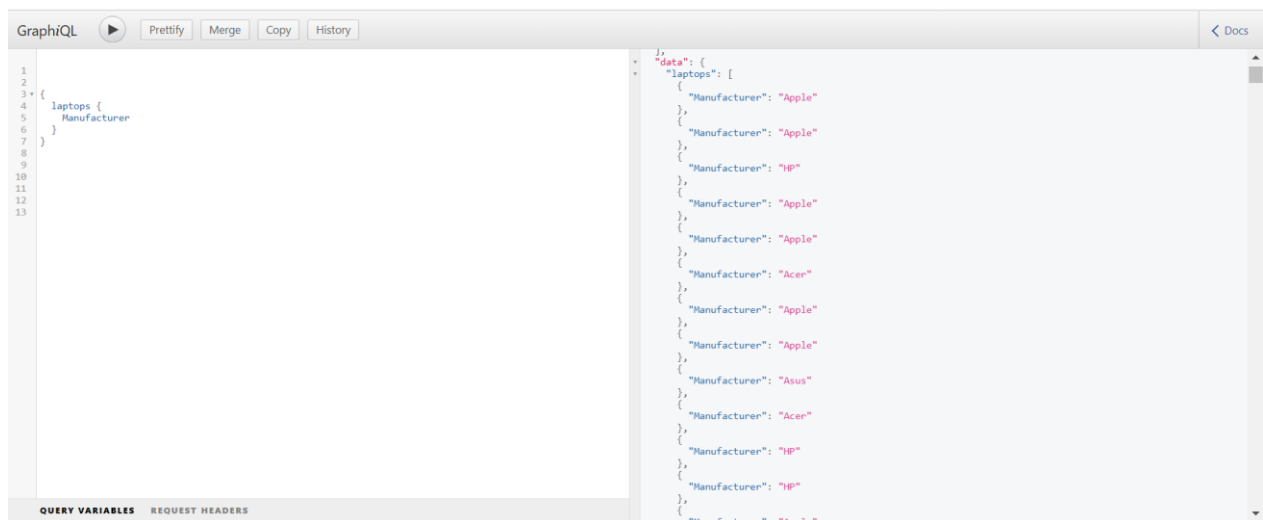
On navigue vers :

src->products -> urls.py

```
from django.contrib import admin
from django.urls import path
from graphene_django.views import GraphQLView
from products.schema import schema

urlpatterns = [
    path('admin/', admin.site.urls),
    path("graphql", GraphQLView.as_view(graphiql=True, schema=schema)),
]
```

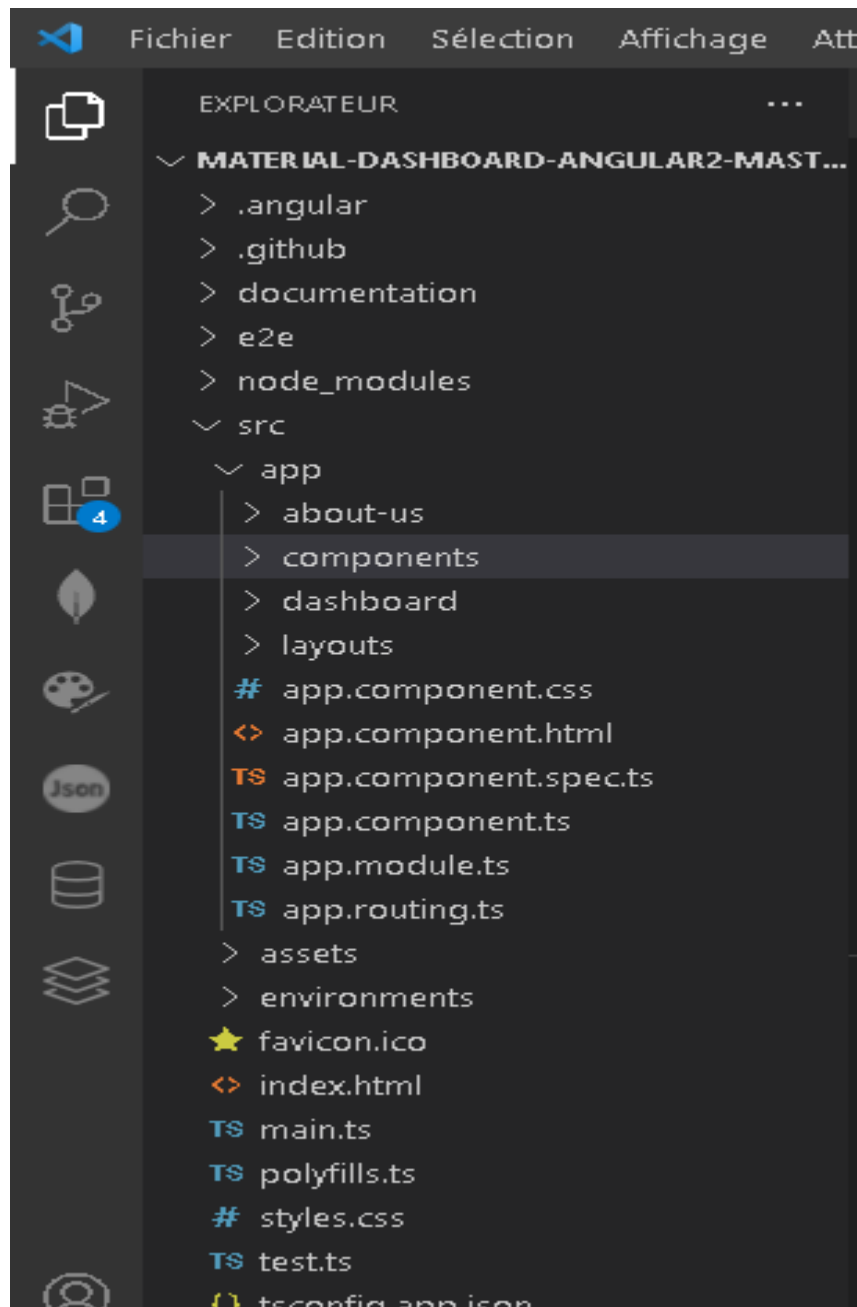
Et on lance <http://localhost:8000/graphql>:



## X. Structure de Frontend:

Pour le frontend nous avons travaille avec angular 13

La premiere chose a faire c'est d'installer NodeJS et AngularCli,puis de créer un projet Angular,et la creation des composants qu'on a besoin.



## XI. Visualisation des données:

