

L'INSTITUT NATIONAL DES POSTES ET TÉLÉCOMMUNICATIONS - RABAT

Rapport de Projet WEB/J2EE : TechGadgets

Réalisé par :

EL BATOURI BADR-EDDINE
ELGAHBI TAHA

Encadré par :

Mr. EL HAMLAOUI MAHMOUD





Remerciements :

Nous voudrions tout d'abord adresser toute notre gratitude à notre encadrant Mr. EL HAMLAOUI Mahmoud, pour sa confiance, sa disponibilité et surtout cette opportunité pour acquérir des nouvelles compétences et découvrir des nouveaux langages.

Nous désirons aussi remercier tous ceux qui ont contribué à la réussite de ce projet par leur partage de connaissances et de conseils ainsi que leur disponibilité à tout temps.



Résumé

Notre projet de Programmation WEB/JEE entre dans le cadre d'E-Shopping, il consiste à développer une application web qui permet aux personnes, qui n'ont pas assez de connaissances autour de la technologie et le matériel informatique, d'avoir une idée sur les gadgets qui sont convenables avec leurs besoins. La plateforme web offrera 2 services :

- Recommandation générale pour les invités.
- Recommandation plus personnalisée pour les abonnés.

Lors d'une visite, l'utilisateur peut choisir de "Guest mode" qui offre une recommandation base sur un paragraphe descriptif donné par la personne, ou choisir le mode d'authentification soit l'inscription sur notre plateforme ou bien l'identification par le compte Google.

La plateforme Tech-Gadgets facilite donc le Shopping en recommandant aux utilisateurs des appareils convenables avec leurs besoins.

Mots clés : E-Shopping, Recommandation, Web, authentification.

Abstract

Our WEB/JEE project is envisioned to have an impact on e-shopping , it consists on developing a web platform that allows shoppers to get recommendations for tech gadgets based on the user's description. It contains principally two types of services :

- General Recommendation for guest.
- Personalized Recommendations for subscribers.

During a visit, users will have the option to choose Guest mode that offers a recommendation based on a description paragraph given by the user, or choose Authentication mode that can either be done through signing up to the platform or with a Google account

The Tech-gadgets platform facilitate therefore the process of shopping by recommending electronic devices to users according to their needs.

Keywords : E-Shopping, Recommendation, Web, authentication

Table des figures

2.1	Diagramme des cas d'utilisation	12
2.2	Microservices	13
2.3	Opérateurs qui extraient les données nécessaires sur les produits	14
2.4	ORM Entity of Authentication Microservice	22
2.5	Test unitaires	23
2.6	Dataset Sample	24
2.7	Cleaned Dataset	24
2.8	Example of Normalization	25
2.9	Stemming vs Lemmatizationt	25
2.10	Models Testing	26
2.11	Aspect Based Sentiment Analysis	27
3.1	Page d'accueil/ propriétés TechGadgets	33
3.2	Page d'accueil/ barre recherche	34
3.3	Page d'accueil/ Login et Signup	35
3.4	Page d'accueil/ barre recherche	35
3.5	Les produits recommandés	36
3.6	Les détails specifics aux abonnées	36

Table des matières

Résumé	3
Abstract	4
Table des figures	5
Table des matières	7
Introduction générale	8
1 Contexte général du projet	9
1.1 Présentation du sujet	9
1.2 Besoins fonctionnels	10
1.3 Besoins non fonctionnels	10
2 Analyse et conception	11
2.1 Acteurs :	11
2.2 Diagramme des cas d'utilisation ;	11
2.3 Conception et analyse d'architecture générale de la solution :	12
2.3.1 différents éléments de l'architecture	13
2.4 Architecture générale et workflow :	17
2.5 Design patterns :	20
2.5.1 Inversion of control and dependecny injection :	20
2.5.2 MVC modele :	20
2.5.3 ORM	21
2.5.4 Unit testing : Test driven devloppment	22
2.6 Machine Learning	23

2.6.1	Préparation des données	23
2.6.2	Prétraitement de texte	24
2.6.3	Features Extraction Sentiment Analysis . . .	26
2.7	Conclusion	27
3	Mise en oeuvre	28
3.1	outils utilisés :	28
3.1.1	HTML et CSS :	28
3.1.2	Bootstrap 5 :	29
3.1.3	Angular :	29
3.1.4	Spring Cloud :	30
3.1.5	Python :	31
3.1.6	Pandas :	32
3.1.7	NLTK :	32
3.2	Réalisation :	33
3.2.1	Guest :	33
	Conclusion générale	37

Introduction générale

Nous avons choisi le domaine d' E-Shopping du matériel informatique vu l'importance de ce domaine dans un monde qui est en route vers la digitalisation. Nous avons tous, sans doute, cherché pendant des heures et des heures dans l'internet pour savoir l'appareil qu'on doit acheter.

Notre projet consiste à proposer une solution à ce problème en recommandant aux utilisateurs des appareils convenables avec leurs besoins, la solution sera une plateforme web et un model de Machine Learning qui fonctionnent suivant une architecture microservices en utilisant Angular, Spring Boot, Python et ses frameworks.

Dans ce rapport on va parler dans un premier plan du contexte général , traiter la partie de l'analyse et la conception puis passer à la réalisation en vous montrant les outils et l'apparence finale de notre application.

Dans le premier chapitre nous allons parler du contexte général du projet pour vous mettre en situation, après nous allons passer à l'analyse et la conception dans le deuxième chapitre. Ce dernier va contenir une description des cases d'utilisation et les besoins de chacun d'eux en terme de technologies et d'architecture. Et finalement dans le 3ème chapitre nous allons citer les outils utilisés et insérer des captures d'écran de l'application.

Chapitre 1

Contexte général du projet

Cette partie nous permet de présenter les circonstances de notre projet d'une manière générale, vous allez découvrir le sujet, les besoins et la planification du projet.

1.1 Présentation du sujet

L' E-Shopping est devenue un secteur très riche en innovation et en créativité c'est ce que nous a attirer vers ce domaine puisqu'il lie deux grands volés différents qui sont l'informatique et le shopping, ce mélange se présente dans notre projet qui est la mise en place d'une platforme web qui facilite le shopping en recommandant aux utilisateurs des appareils convenables avec leurs besoins.

La platforme TechGadgets cible le problème de passer plusieurs heures et même jours en cherchant dans l'internet un gadget qui satisfait nos besoins et résout également la nécessité d'avoir beaucoup de connaissances dans le domaine de technologie et matériel informatique. L'utilisateur aura le choix d'inscrire ou bien choisir le "guest mode" pour accéder à notre service qui sera une recommandation des gadgets en se basant sur une description d'utilisateur.

L'application n'arrête pas ici, mais elle dépasse tout ça pour offrir l'option d'obtenir des recommandations plus personnalisées pour les abonnés en utilisant l'histoire des transactions ainsi que la description.

1.2 Besoins fonctionnels

Coté Guest :

- Description d'un Tech-Gadget
- Obtenir une liste des produits

Coté abonné :

- Authentification
- Description d'un Tech-Gadget
- Obtenir une liste des produits
- Évaluer Les resultats

1.3 Besoins non fonctionnels

Une fois traiter le coté fonctionnel , nous devons considérer aussi les besoins non fonctionnels , par exemple :

- User experience
- Actualisation des informations
- Maximum de sécurité
- Adaptation à l'utilisateur

Chapitre 2

Analyse et conception

Ce chapitre présente l'analyse théorique de notre projet.

2.1 Acteurs :

Les acteurs de notre application sont principalement comme suit :

Le développeur : Il gère les comptes des utilisateurs, il peut aussi ajouter des services à l'application(Ajouter la recherche du produit dans Jumia par exemple, améliorer le model NLP)

L'utilisateur : Chercher un produit sur Internet à l'aide d'un texte descriptif, en précisant les sites ciblés dans son recherche

2.2 Diagramme des cas d'utilisation ;

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

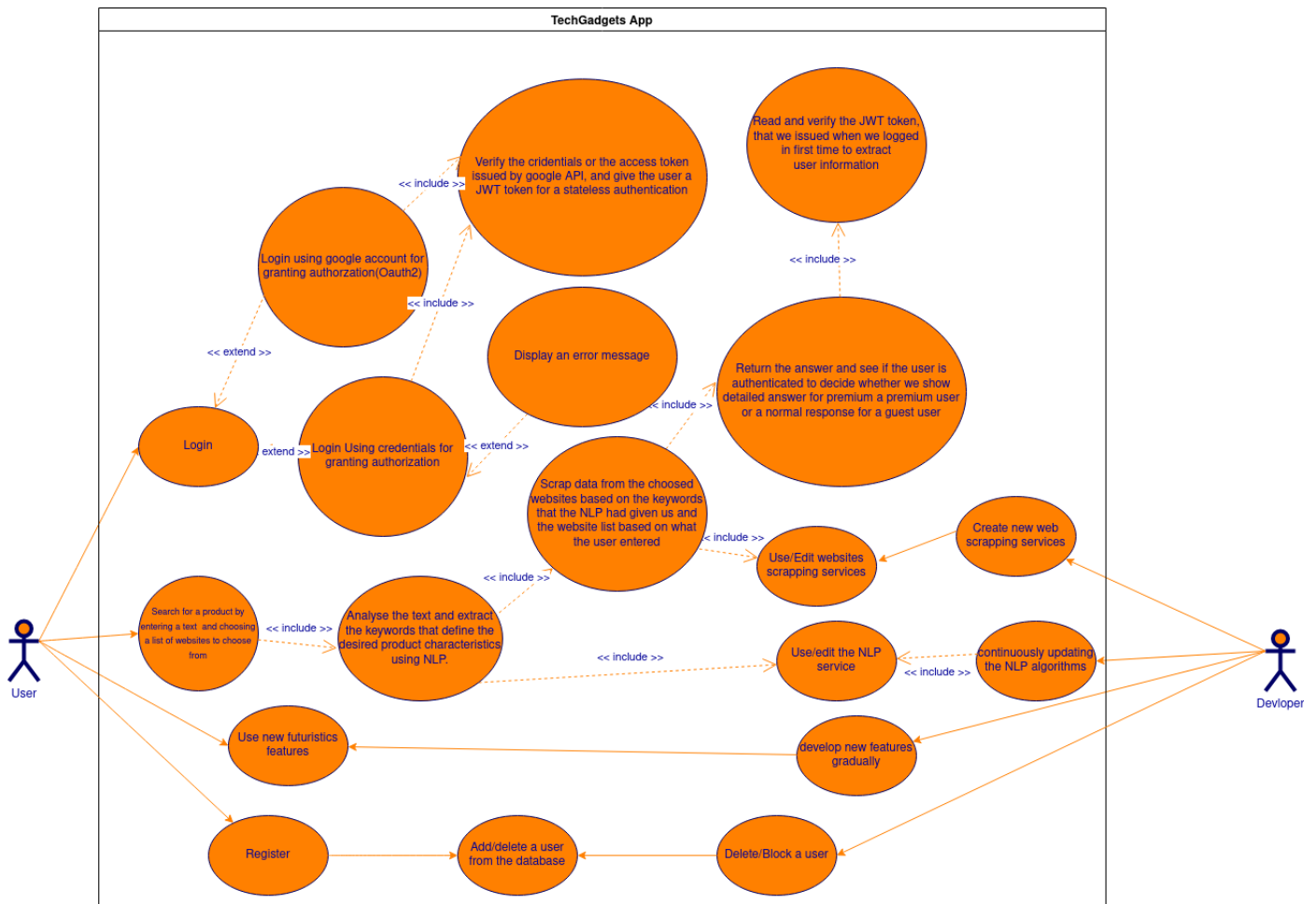


FIGURE 2.1 – Diagramme des cas d'utilisation

2.3 Conception et analyse d'architecture générale de la solution :

Après avoir présenté en quoi consiste l'application. Dans cette partie du chapitre, nous allons fournir l'architecture générale de notre application, y compris le workflow de l'interaction de l'utilisateur avec l'application, et comment le serveur gère la requête HTTP de l'utilisateur. Et les problèmes que l'architecture en microservice a résolu et vue générale sur le design pattern de nos microservices de l'application à l'aide de diagrammes pertinents car notre architecture est une architecture basée sur des microservices et chaque microservice est une petite application elle-même et on repete la meme design pattern pour chaque petite application.

2.3.1 différents éléments de l'architecture

Un microservice : un microservice est une petite application qui représente une unité logique qui effectue une tâche logique dans l'ensemble de notre application, ces petites applications peuvent être hébergées sur différentes machines. lorsque nous avons besoin d'un microservice pour effectuer une opération, nous lui fournissons simplement l'entrée nécessaire et nous attendons la sortie. les flux d'entrée/sortie sont basés sur un protocole REST. Tous nos microservices seront basés sur une application Spring Boot ou une application Flask nous pouvons donc considérer nos microservices comme des ensembles des petites API. Mais ce qui en a fait des microservices, c'est le "Registry service" (Spring cloud registry service) qui fournit les informations nécessaires pour connecter ces microservices entre eux.

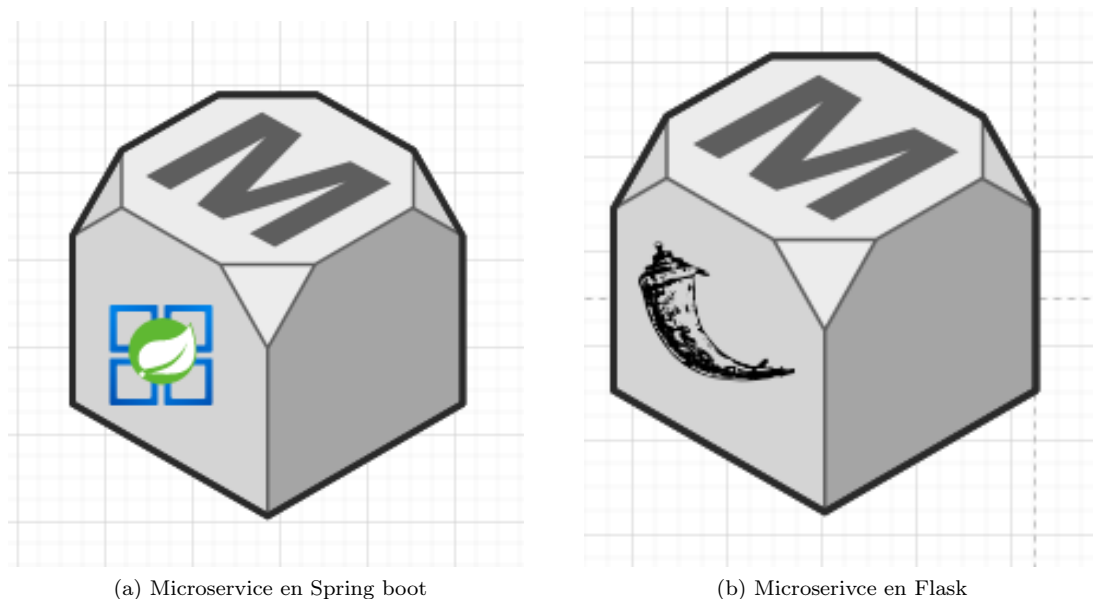


FIGURE 2.2 – Microservices

Opérateurs : un opérateur est un ensemble des fonctions qui exécutent certaines tâches en Java ou en python sans avoir besoin d'appels externes pour d'autres microservices, pour le moment nous avons 3 types d'opérateurs l'opérateur NLP est un service qui analysera le texte de description que l'utilisateur émettra et le service "scrapping" qui extraira les données des sites Web ciblés ou le service de rechange qui utilisera des API externes pour extraire les données

nécessaires dans le cas où le service de "Scrapping" est hors-ligne ou il renvoie une erreur.

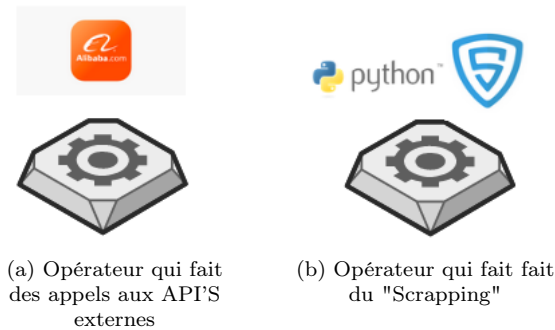
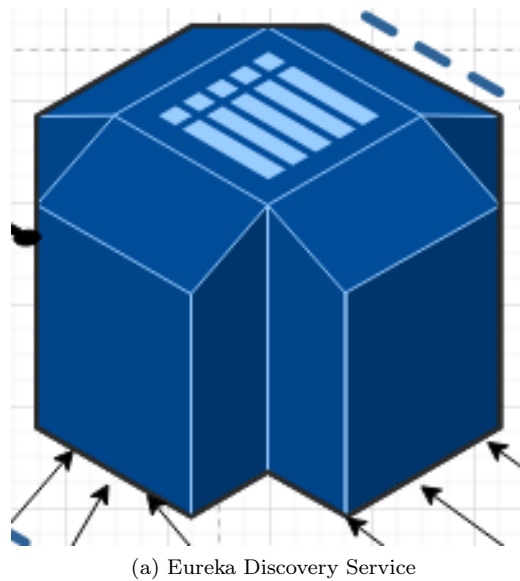


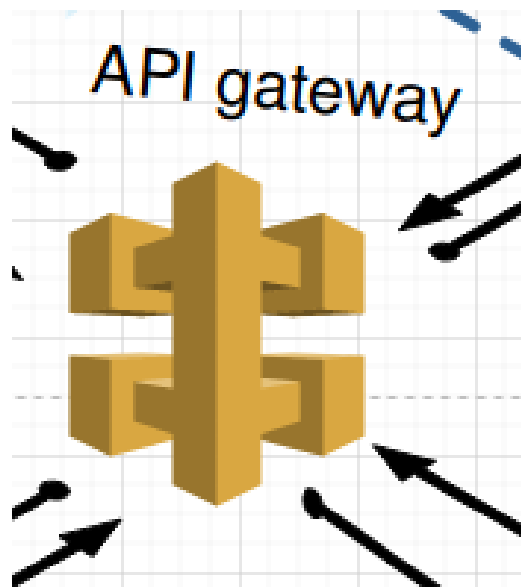
FIGURE 2.3 – Opérateurs qui extraient les données nécessaires sur les produits

Registry Service(Eureka) : c'est un microservice qui offre des informations sur tous les autres microservices impliqués dans notre application. afin que nous puissions continuer à encadrer leurs activités, leur santé et leurs adresses dans le cloud. de cette façon si nous voulons les faire communiquer entre eux, nous pouvons simplement renvoyer leur nom sur le "Registry Service Eureka" et il nous fournira les informations nécessaires (adresse IP, port et instances disponibles, notez que chaque microservice peut avoir plusieurs instances) pour qu'on puisse appeler. Il est à noter que chaque microservice injecte ses informations sur le "Registry Service"



Api gateway : comme nous l'avons vu, chaque microservice peut être hébergé sur une machine différente donc une adresse IP différente. Le rôle de cette API Gateway est d'être la passerelle c'est à

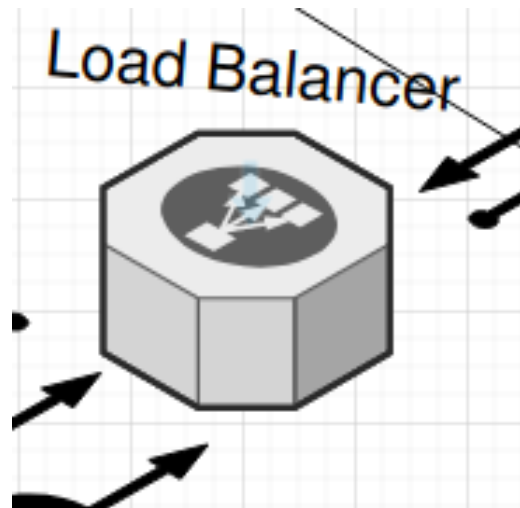
dire ; centraliser toutes les requêtes vers nos microservices. Au lieu d'appeler directement le microservice, nous utilisons une route de passerelle et nous mappons cette route au microservice à l'aide du Eureka. afin que le client envoie la requête à la même adresse IP et au même itinéraire même si le microservice souhaité change son adresse IP ou son port fréquemment, Eureka fournira toujours ces informations car le rôle d'eureka est de mapper le nom d'un microservice à son adresse même s'il est hébergé sur le cloud et son adresse IP varie fréquemment, puis le "API GateWay" utilise ces données pour faire le routage. La centralisation des requêtes n'est pas la seule chose que fait la passerelle, nous pouvons aussi associer des règles à chaque route, telles que la journalisation des adresses IP des clients qui accèdent à une route spécifique ou sécuriser des routes et les rendons accessibles uniquement aux utilisateurs autorisés à y accéder. Nous centralisons donc la sécurité et la journalisation à un seul endroit afin qu'elles soient gérables ce qu'on appelle "Cross concern centralisation whit an Aspect oriented approach".



(a) API Gateway

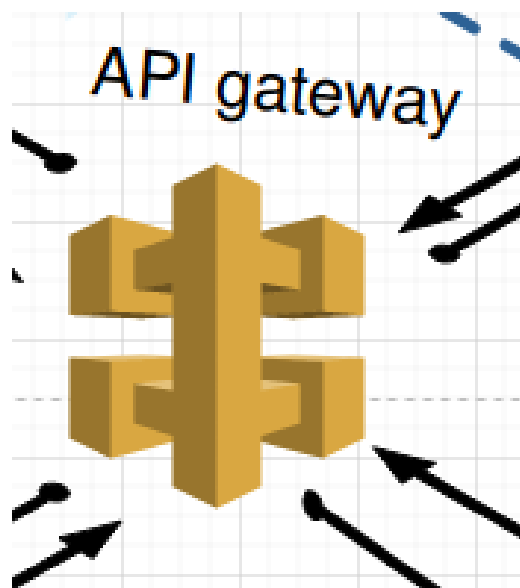
Load balancer : Comme nous l'avons vu, chaque microservice peut avoir plusieurs instances, donc le "load balancer" utilisera des algorithmes d'équilibrage pour répartir équitablement les requêtes entrantes sur les instances disponibles.

Database : la base de données est utilisée à des fins de stockage.



(a) Load balancer

Ici nous utilisons des bases de données MySQL pour la production et des bases de données H2 pour les tests, il est à noter que la seule microservice qui utilise la base de données c'est le microservice d'authentification en utilisant hibernate pour faire la communication (ORM).



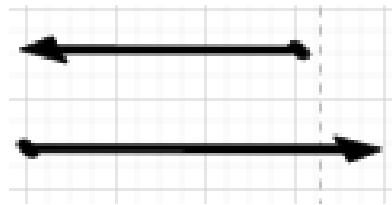
(a) Database

Les relations entre les éléments : Il existe 3 types de relations entre notre composant :

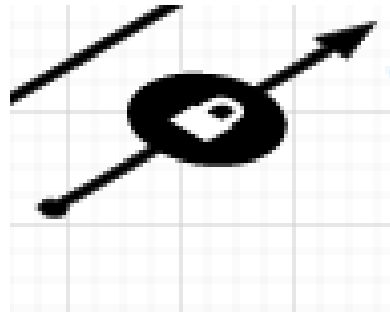
- 2 Flèche représentant un protocole REST pour la communication entre 2 composants.

- 1 Flèche signifie injection de dépendance, nos opérateurs seront injectés dans les microservices.

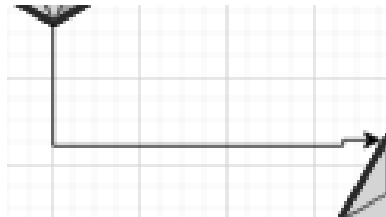
- Flèche avec un symbole de verrouillage représente un itinéraire sécurisé afin que seuls les utilisateurs autorisés obtiendront la réponse



(a) 2 Flèche



(b) 1 Flèche



(c) Flèche avec un symbole de verrouillage

2.4 Architecture générale et workflow :

Maintenant que tous les éléments sont connus pour le lecteur, nous pouvons plonger dans l'architecture générale et voir comment chaque composant de notre application interagit avec d'autres composants. ainsi nous établirons les connaissances nécessaires pour voir pourquoi l'architecture des microservices est si puissante. et peut maintenant discuter quel type de problèmes elle résout pour notre cas d'utilisation, et aussi quel type de problème elle génère et comment nous allons gérer ces problèmes avec les technologies fournies par NET-FLIX offert dans le framework Spring Cloud.

lien vers l'architecture : (Voir fichier ci-joint, Architecture.html)

Flux de travail idéal :

Le client envoie une requête HTTP comprenant le numéro de page, le texte de description ou les mots-clés du produit et ses informations d'identification (nom d'utilisateur et mot de passe ou jeton d'accès google) le microservice d'authentification validera les informations d'identification et émettra un jeton JWT au client contenant ses autorisations et ses informations personnelles à chaque fois qu'il envoie une requête au serveur il envoie ce jeton JWT, la passerelle verra si le jeton JWT est valide (Chiffrement et signature RSA). après cela, il déléguera la requête au microservice de l'utilisateur final qui déléguera le texte de description au microservice NLP pour extraire les mots-clés qui caractérisent le produit souhaité par le client, puis déléguera ces mots-clés au microservice qui récupère les données du produit à partir des sites Web externes comme amazon et ali-express, ce microservice vérifiera si le service de "Scrapping" fonctionne. si cela fonctionne, il récupérera les données en utilisant le "Scrapping", sinon il utilisera l'API externe (nous l'avons fait cette algorithmes parce que l'API externe est un service payant). Ensuite, il enveloppera la réponse et la déléguera à l'utilisateur si l'utilisateur est authentifié, il lui fournira plus d'informations sur le produit (est ce que produit est "Best selling product", est-il en promotion, etc.), sinon il donnera un résultat normal.

Maintenant, avec le flux de l'application à l'esprit, nous allons parler des problèmes que l'architecture de microservices crée pour nous, et comment nous parvenons par la suite à résoudre ces problèmes.

La sécurité par le modèle de session : En partant de l'idée que chaque microservice peut vivre dans une machine différente (Cloud), la solution typique de "session" pour l'autorisation et le suivi des utilisateurs authentifiés n'est plus valide et nous devons rechercher une solution "Stateless", c'est-à-dire ne pas conserver les informations des utilisateurs en mémoire, mais plutôt les conserver chez un autre service ou l'utilisateur lui-même. pour résoudre ce problème, nous avons utilisé le jeton JWT et les protocoles OAuth2 (on a utilisé, ce

du google).

Orchestration des microservices : Chaque microservice est une application autonome, pour garder une trace de toutes ces applications, nous avons utilisé la solution que Netflix a développé "Spring cloud". qui fournissent la "Spring cloud API Gateway" pour centraliser les requetes et le serveur de découverte Eureka en tant "Service Registry", et "Health Actuator" pour pour tracer l'état de santé des microservices.

Microservices applications and Bootstraping : chaque microservice est une petite application donc a chque fois on veut creer un microservice il faut bcp des configuration, on a resoudre ca en utilisant spring boot et flask parce que ces deux framework minimiser la configuration necessaire pour demarer une application. meme ils generent les dependences qu'on aura besoin.

Error Handling and circuits Breaker : Imaginez si un microservice génère une erreur, cette erreur sera transmise au microservice appelant en tant que réponse HTTP avec le statut 500 en tant qu'erreur interne, et comme cela, nous ne pouvons pas savoir où nous obtenons cette erreur ou quelle est la cause de l'erreur, pour résoudre ce problème nous avons utilisé "Spring Cloud cricuit breaker" basé sur la "résillance", il coupera le circuit et donnera une réponse directement au client en cassant le modèle d'appelant appelé.

Pourquoi on a utliser les microservices :

Résilience : Les microservices nous fournissent un grand nombre de fonctionnalités, l'une est la résilience si nous voulons ajouter une nouvelle fonctionnalité ou déboguer un problème, nous n'avons pas besoin de déconnecter notre application mais uniquement le service concerné, et comme vous savez notre application dependent largement des services externes.

"Cloud native architecture" : "Cloud native application, or Distrbuted System native application" nous aide à augmenter notre base d'utilisateurs, de sorte que notre application sera en mesure de prendre en charge une énorme base d'utilisateurs, et même pour le

"scraping", il faut qu'on utilise plusieurs IP adresse pour éviter le blocage.

"Ready production features : " Concentrez-vous sur la production, chaque membre de l'équipe utilisera les technologies qu'il aime utiliser. Composants réutilisables, chaque microservice sera dédié à une seule tâche logique afin que nous puissions le réutiliser dans d'autres projets.

2.5 Design patterns :

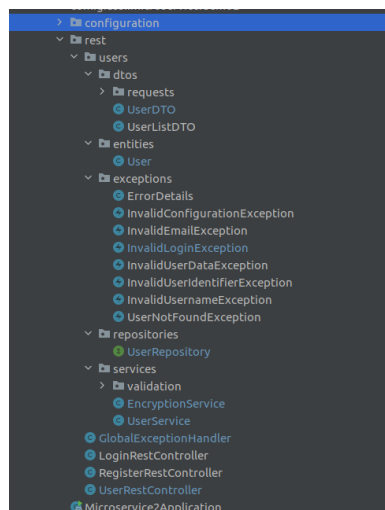
dans cette section, nous parlerons des modèles de conception que nous avons utilisés pour développer nos micro services, on va étudier une seule micro service(Celui d'authentification) et les autres doivent suivi le même design patter

2.5.1 Inversion of control and dependecny injection :

Le framework spring est base sur ces deux concepts qui offrent une "Designe pattern" pret pour la production

2.5.2 MVC modele :

Nous avons utilisé le célèbre modèle MVC(avec des petites customisation parce qu'on a pas du "VIEW" ici)avec une gestion d'erreur personnalisée pour intégrer « Ciciut Breaker »



(a) MVC :application

"DTOS package classes" : Des classes qui sont dédiées à la serialisation des objets JSON que le client envoie à des objets "POJO"

"ENTITES and REPOSITORIES package classes" : On peut donc utiliser n'importe quelle base de données, car c'est Hibernate qui crée les requêtes sql en transmettant votre code java basé sur la spécification JPA.

"exceptions package classes" : Des classes qui sont dédiées à la manipulation des erreurs et déclenchement du "Circuit Breaker".

"SERVICES package classes and Controller" : Ces classes représentent le contrôleur qu'il délègue la requête aux routes propres et les services sont ceux qui contiennent notre couche métier.

2.5.3 ORM

Un mapping objet-relationnel (en anglais object-relational mapping ou ORM) est un type de programme informatique qui se place en interface entre un programme applicatif et une base de données relationnelle pour simuler une base de données orientée objet. Ce programme définit des correspondances entre les schémas de la base de données et les classes du programme applicatif. On pourrait le désigner par là, « comme une couche d'abstraction entre le monde objet et monde relationnel ». Du fait de sa fonction, on retrouve ce type de programme dans un grand nombre de frameworks sous la forme de composant ORM qui a été soit développé, soit intégré depuis une solution externe.

```

@Entity
@Table(name="T_users")
@Data
public class User {

    @Id
    @GeneratedValue(strategy= GenerationType.AUTO, generator="native")
    @GenericGenerator(name = "native", strategy = "native")

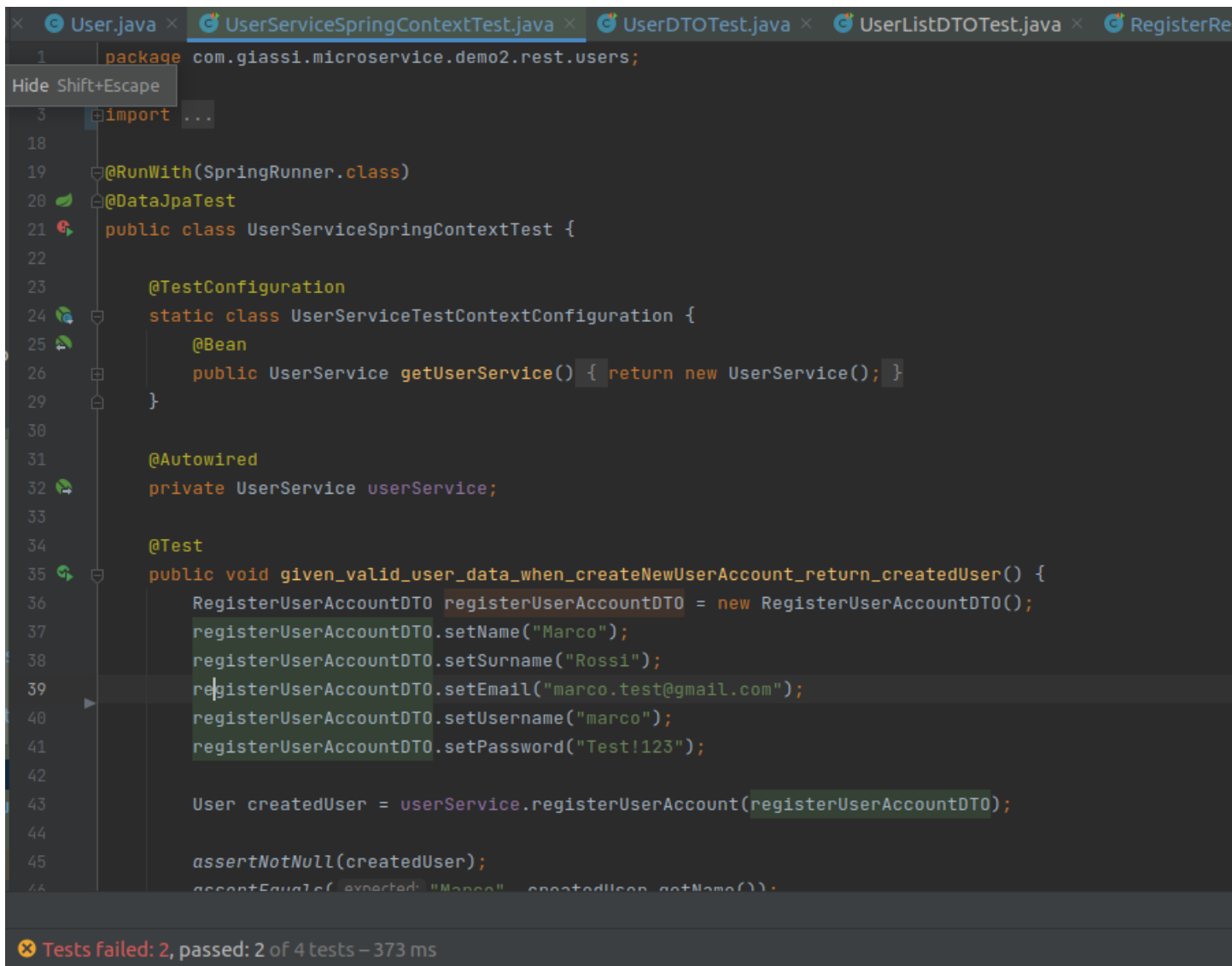
    @Column(name="id")
    private Long id;
    @Column(name="username", nullable = false)
    private String username;
    @Column(name="email", nullable = false)
    private String email;
    @Column(name="password", nullable = false)
    private String password;
    @Column(name="name", nullable = false)
    private String name;
    @Column(name="surname", nullable = false)
    private String surname;
    @Column(name = "birth_date")
    private java.time.LocalDate birthDate;
    @Column(name="enabled")
    private boolean enabled;
    @Column(name="uid", nullable = true)
    private Integer uid;
    @CreationTimestamp
    private java.time.LocalDateTime creationDt;
    @Basic
    private java.time.LocalDateTime updatedDt;
    @Basic
    private java.time.LocalDateTime loginDt;
}

```

FIGURE 2.4 – ORM Entity of Authentication Microservice

2.5.4 Unit testing : Test driven developpment

Pour uni testing on a utilisé JUNIT et Mockito, la seule micro service qui a ces unis test totalement faites est celui de l'authentification, lors de l'écriture de ce microservice on a suivi "Test Driven Devloppement Approach"



```
1 package com.glass1.microservice.demo2.rest.users;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 @RunWith(SpringRunner.class)
20 @DataJpaTest
21 public class UserServiceSpringContextTest {
22
23     @TestConfiguration
24     static class UserServiceTestContextConfiguration {
25
26         @Bean
27         public UserService getUserService() { return new UserService(); }
28     }
29
30
31     @Autowired
32     private UserService userService;
33
34
35     @Test
36     public void given_valid_user_data_when_createNewUserAccount_return_createdUser() {
37         RegisterUserAccountDTO registerUserAccountDTO = new RegisterUserAccountDTO();
38         registerUserAccountDTO.setName("Marco");
39         registerUserAccountDTO.setSurname("Rossi");
40         registerUserAccountDTO.setEmail("marco.test@gmail.com");
41         registerUserAccountDTO.setUsername("marco");
42         registerUserAccountDTO.setPassword("Test!123");
43
44         User createdUser = userService.registerUserAccount(registerUserAccountDTO);
45
46         assertNotNull(createdUser);
47         assertEquals("expected: 'Marco', createdUser.getName()");
48     }
49 }
```

Tests failed: 2, passed: 2 of 4 tests - 373 ms

FIGURE 2.5 – Test unitaires

2.6 Machine Learning

Notre plateforme doit être capable de recommander les appareils électroniques d'après une description d'utilisateur. Bien entendu le "Natural Langage Processing" sera la technologie qu'on utilise pour résoudre cette problématique, cependant on doit passer par plusieurs étapes pour réaliser un modèle capable de réaliser cette tâche.

2.6.1 Préparation des données

Pour ce projet, l'ensemble de données électroniques se compose de commentaires et d'informations sur les produits d'amazon Cet ensemble de données comprend des évaluations (cotes, texte, votes sur

l'utilité) et des produits. les métadonnées (descriptions, renseignements sur la catégorie, prix, marque et caractéristiques de l'image)..

Sample from metadata Dataset

```
"category": [
  "Cell Phones & Accessories",
  "Cases, Holsters & Sleeves"
],
"tech1": "",
"description": [
  "",
  "Elegani Butterfly Case 3D Pattern Back Cover Case Skin For Apple iPhone 5 5G",
  "Ultra light weight case",
  "Elegantly designed with 3D cutout effect",
  "Super rich rubber coating"
],
"fit": "",
"title": "MinisDesign 3d Bling Crystal Bow Transparent Case for Apple Iphone 4 and 4s-Pink",
"tech2": "",
"brand": "ELEGANI",
"feature": [
  "Fashionable with unique 3D butterfly design for the iPhone 5 from others",
  "Precise cut design with access to all ports and features of the iPhone 5",
  "Dirt-resistant Butterfly Case for iPhone 5 with 3D PC Cover",
  "Design: Simple hard back case, Luxury and classical butterfly",
  "3D art design. Creative 3D butterfly design"
],
"rank": [
  ">#228,544 in Cell Phones & Accessories (See Top 100 in Cell Phones & Accessories)",
  ">#114,920 in Cell Phones & Accessories > Cases, Holsters & Clips"
],
"main_cat": "Cell Phones & Accessories",
"similar_item": "",
"date": "",
"price": "",
"asin": "7508492919",
"imageURL": [
  "https://images-na.ssl-images-amazon.com/images/I/41H010yPcXL._SX38_SY50_CR,0,0,38,50_.jpg"
],
```

Dataset source

Sample from review Dataset

```
"overall":5.0,
"verified":true,
"reviewTime":"08 4, 2014",
"reviewerID":"A24E3SXTG62LJI",
"asin":"7508492919",
"style":{
  "Color":" Bling"
},
"reviewerName":"Claudia Valdivia",
"reviewText":"Looks even better in person. Be careful
to not drop your phone so often because the rhineston
es will fall off (duh). More of a decorative case tha
n it is protective, but I will say that it fits perfe
ctly and securely on my phone. Overall, very pleased
with this purchase.",
"summary":"Can't stop won't stop looking at it",
"unixReviewTime":1407110400
```

FIGURE 2.6 – Dataset Sample

Dataset Cleaning

asin	category	description	title	brand	feature	rank	details	similar_item	price	imageURLHighRes	review_text	helpfulness
7508492919	['Cell Phones & Accessories', 'Cases, Holsters...	['Elegani Butterfly Case 3D Pattern Back C...	MinisDesign 3d Bling Crystal Bow Transparent C...	ELEGANI	['Fashionable with unique 3D butterfly design ...	['>#228,544 in Cell Phones & Accessories (See ...	{}	NaN	NaN	['https://images-na.ssl-images-amazon.com/imag...	good I guess I've got what I paid for, at \$2.4...	0.02
8288853439	['Cell Phones & Accessories', 'Accessories', '...	['Samsung Official OEM Travel Wall Charger for...	Samsung Official OEM Travel Wall Charger for y...	BlackBerry	['Safely charge your phone from your car using...	['>#494,589 in Cell Phones & Accessories (See ...	{}	NaN	NaN	['https://images-na.ssl-images-amazon.com/imag...	Galaxy S4 Charger This Charger is awesome! Fir...	0.04
8288878881	['Cell Phones & Accessories', 'Accessories', '...	['Safely charge your phone using the original ...	Samsung Galaxy S2 Phone OEM Official Travel US...	Samsung	['Safely charge your phone using the original ...	['>#245,551 in Cell Phones & Accessories (See ...	{}	NaN	NaN	[]	Quick and perfect! Received it promptly and it...	0.08
8288878881	['Cell Phones & Accessories', 'Accessories', '...	['Safely charge your phone using the original ...	Samsung Galaxy S2 Phone OEM Official Travel US...	Samsung	['Safely charge your phone using the original ...	['>#245,551 in Cell Phones & Accessories (See ...	{}	NaN	NaN	[]	It charges, but it doesn't sync with my laptop...	0.03
8288878881	['Cell Phones & Accessories', 'Accessories', '...	['Safely charge your phone using the original ...	Samsung Galaxy S2 Phone OEM Official Travel US...	Samsung	['Safely charge your phone using the original ...	['>#245,551 in Cell Phones & Accessories (See ...	{}	NaN	NaN	[]	Not as nice as I thought it would be in a nuts...	0.02

FIGURE 2.7 – Cleaned Dataset

2.6.2 Prétraitement de texte

Puisque le texte est la forme la plus peu structurée de toutes les données disponibles, divers types de bruit sont présent dans celui-ci

et les données ne sont pas facilement analysables sans aucun prétraitement. L'ensemble du processus de nettoyage et de normalisation du texte, le rendant sans bruit et prêt pour l'analyse est connu comme prétraitement du texte. Dans cette section, le prétraitement du texte suivant a été appliqué.

```
In [200]: cln_phone_reviews['clean_text'] = cln_phone_reviews['review_text'].map(lambda text: normalize_and_lemmatize(text))
          print(cln_phone_reviews['review_text'][0])
          cln_phone_reviews['clean_text'][0]

good I guess I've got what I paid for, at $2.49. It (the phone charger) arrived undamaged, but it doesn't work at all. I'd give
the product NO stars cuz A phone charger that isn't functioning is worthless u gotta gimme my money back.

Out[200]: 'good guess get pay phone charger arrive undamaged doesn't work would give product no star cuz phone charger not function worthl
ess u get give money back'
```

```
In [198]: print (cln_phone_reviews.head(19))
```

	review_text	clean_text
0	good I guess I've got what I paid for, at \$2.49...	good guess get pay phone charger arrive undama...
1	Galaxy S4 Charger This Charger is awesome! Fir...	galaxy charger charger awesome first wasn't exp...
2	Quick and perfect! Received it promptly and it...	quick perfect receive promptly work perfectly ...
3	It charges, but it doesn't sync with my laptop...	charge doesn't sync laptop although charge albe...
4	Not as nice as I thought it would be In a nuts...	not nice think would nutshell not oem charger ...
5	Works as expected I just wanted a charger like...	work expect want charger like original one gal...
6	it works and it looks cool too it works and th...	work look cool work cool blue light neat help ...
7	seems good. sorry i'm not super exited, it is ...	seem good sorry im not super exit phone charge...
8	Nice case, suits my Mom's needs perfectly. I b...	nice case suit moms need perfectly buy case mo...
9	I love them because the "I had a dead battery"...	love thehem dead battery excuse buy go childre...
10	OK Not as good as my old Mophie ... Sorry Powe...	ok not good old mophie sorry powerbear nice pr...
11	Excellent charger This device works well on my...	excellent charger device work well htc high qu...
12	PowerBear iPhone 5c Great product for the mone...	powerbear iphone c great product money really ...
13	Works great! I am a real estate agent and when...	work great real estate agent work great real e...
14	So far, so good... I bought this for my 15 yea...	far good buy year old daughter take school tri...
15	Five Stars Awesome charging case!! I've referr...	five star awesome charge case refer lot friend...
16	Does not work I read the comments on this item...	not work read comment item negative smartphone...
17	Good for the car A good inexpensive cable that...	good car good inexpensive cable work like orig...
18	CHEAP and works perfectly I bought this recent...	cheap work perfectly buy recently not expect m...

FIGURE 2.8 – Example of Normalization

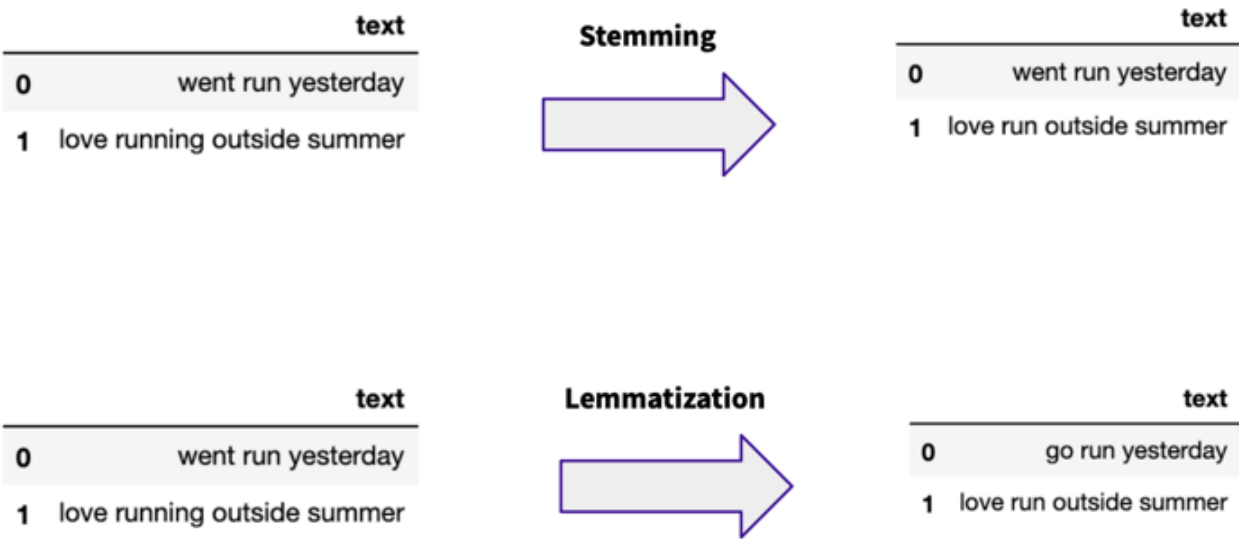


FIGURE 2.9 – Stemming vs Lemmatization

2.6.3 Features Extraction Sentiment Analysis

Les modèles d'apprentissage automatique prennent des valeurs numériques comme entrée. Les révisions sont faites de phrases, donc afin d'extraire des modèles à partir des données ; nous devons trouver un moyen de les représenter d'une manière que l'algorithme d'apprentissage automatique peut comprendre, i.e. comme une liste de nombres.

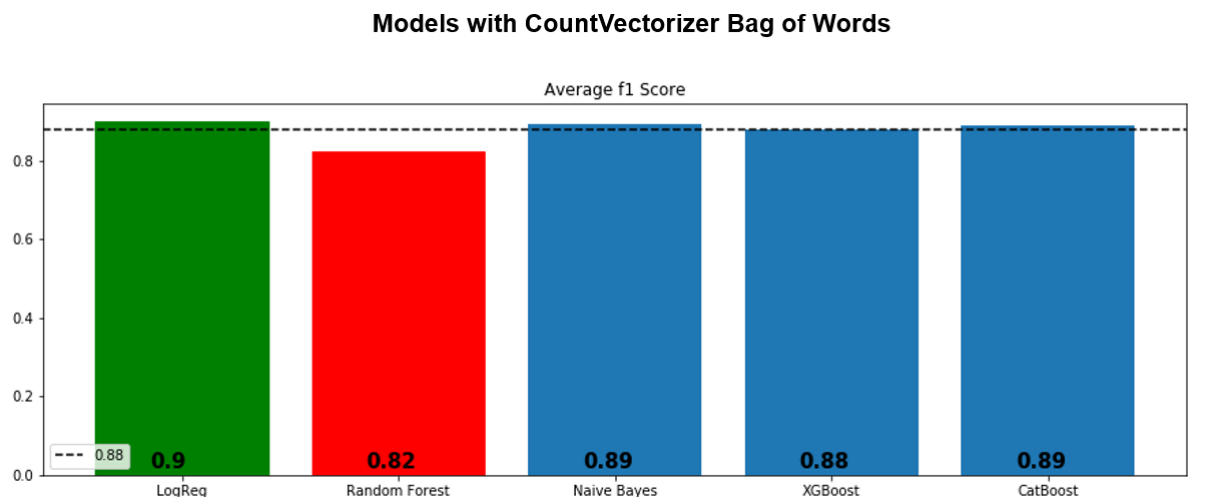


FIGURE 2.10 – Models Testing

Feature engineering consiste à utiliser la connaissance du domaine des données pour créer des fonctions qui : rendre les algorithmes d'apprentissage automatique fonctionnable. Les "Features" sont généralement de nature numérique et peuvent être valeurs numériques absolues ou caractéristiques catégoriques qui peuvent être encodées en tant que caractéristiques binaires pour chaque dans la liste à l'aide d'un processus appelé encodage one-hot. Le processus d'extraction et la sélection des caractéristiques est à la fois l'art et la science, et ce processus s'appelle l'extraction de caractéristiques ou de caractéristiques ingénierie.

```

In [64]: user_description="I want a pink phone. that takes beautiful pictures. A nice big screen. And I want the battery to be really good
print("the review text is: ",user_description)
print("these are keywords:", keywords_extraction(user_description))
Classify_and_stats(user_description)

the review text is: I want a pink phone. that takes beautiful pictures. A nice big screen. And I want the battery to be really good
these are keywords: ['pink', 'phone', 'beautiful', 'pictures', 'big', 'screen', 'good', 'battery']

Out[64]: [[0: Sentence("I want a pink phone."),
  'aspects': 'phone',
  'description': 'pink',
  'stats': Sentiment(polarity=-0.1, subjectivity=0.3),
  'classify': 'positive'},
  [1: Sentence("that takes beautiful pictures."),
  'aspects': 'pictures',
  'description': 'beautiful',
  'stats': Sentiment(polarity=0.85, subjectivity=1.0),
  'classify': 'positive'},
  [2: Sentence("A nice big screen."),
  'aspects': 'screen',
  'description': 'big',
  'stats': Sentiment(polarity=0.3, subjectivity=0.55),
  'classify': 'positive'},
  [3: Sentence("And I want the battery to be really good"),
  'aspects': 'battery',
  'description': 'really good',
  'stats': Sentiment(polarity=0.7, subjectivity=0.6000000000000001),
  'classify': 'positive']]

```

FIGURE 2.11 – Aspect Based Sentiment Analysis

Dans le cadre de ce projet, Bag of Words model, TF-IDF, Hashing Vectorizer, Word2Vec et ajout des mots les plus courants dans la liste des mots clés, SMOTE, PCA et SVD tronqué techniques en modèles de classification dans les sections suivantes dans le cadre de l'ingénierie des caractéristiques et la sélection.

2.7 Conclusion

Dans ce chapitre nous avons présenté de manière globale les principales étapes de l'analyse et la conception de notre application suivant les différents schémas, afin de concilier la phase de mise en œuvre. Le prochain chapitre sera consacré à la phase de développement de notre application.

Chapitre 3

Mise en oeuvre

Après avoir réalisé une conception répondant au mieux aux besoins de notre application, nous commençons la partie implémentation de l'application que nous avons développée, en exposant les différents outils de développement et langages utilisés lors de la production de nos application ainsi que les résultats obtenu.

3.1 outils utilisés :

3.1.1 HTML et CSS :



HTML, HyperText Markup Language, donne une structure et un sens au contenu en définissant ce contenu par principe de balisage comme, par exemple, des titres, des paragraphes ou des images.. etc

CSS, ou Feuilles de style en cascade, est un langage de présentation créé pour styliser l'apparence du contenu, en utilisant, par exemple, des polices ou des couleurs.

3.1.2 Bootstrap 5 :



Bootstrap est un Front-End Framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce Framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement.

Il est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore.

3.1.3 Angular :



Angular s'inscrit définitivement dans une approche moderne de la création d'applications web « One page » et d'interfaces utilisateurs. Orienté objet, ce Framework MVC exploite le langage Typescript (langage compilé qui va générer du Javascript). Avec Angular, la

logique applicative est insérée directement dans le HTML par l'intermédiaire d'éléments ou d'attributs (principe du data-binding).

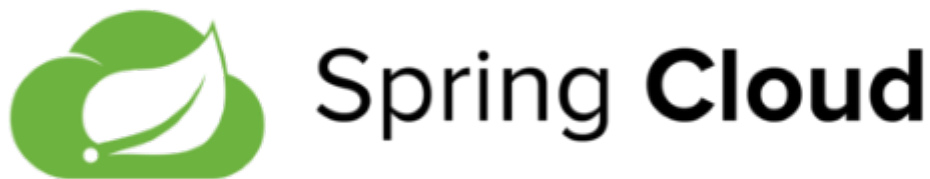
=> TypeScript :

TypeScript est un langage open-source pris en charge par Microsoft, qui s'appuie sur JavaScript en ajoutant une fonction de type statique facultative. Les types fournissent un moyen de structurer et de valider son code avant de l'exécuter.

=> gapi (google API) pour une authentification oauth2)

Le contrôle d'accès pour les API Google Cloud comprend l'authentification, l'autorisation et l'audit. L'authentification établit qui vous êtes, l'autorisation détermine ce que vous pouvez faire et les audits enregistrent ce que vous avez fait.

3.1.4 Spring Cloud :



Spring Boot est un framework de développement JAVA. C'est une déclinaison du framework classique de Spring qui permet essentiellement de réaliser des micro services (ce sont la majeure partie du temps des services web qui sont regroupés en API).

=> Spring cloud circuit breaker(Resilience4J)

Spring Cloud Circuit breaker provides an abstraction across different circuit breaker implementations. It provides a consistent API to use in your applications allowing you the developer to choose the circuit breaker implementation that best fits your needs for your app.

=> Eureka discovery service

C'est un service développé en Java par Netflix, dont le rôle est de gérer l'enregistrement et la localisation de services à des fins de load balancing et de failover. Sa très bonne intégration avec Spring Boot fait de lui un très bon choix de Service Registry dans le monde des micro-services Java.

3.1.5 Python :



Python est particulièrement populaire pour l'analyse de données et la Machine Learning, mais aussi pour le développement web backend. Ce langage est aussi utilisé pour scraper des données du web et développer des outils de productivité.

=>json

Python est livré avec un paquet intégré appelé json pour encoder et décoder les données JSON. Le processus d'encodage de JSON est généralement appelé sérialisation. Ce terme désigne la transformation de données en une série d'octets (donc en série) à stocker ou à transmettre sur un réseau.

3.1.6 Pandas :



Pandas est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

3.1.7 NLTK :



NLTK est une plate-forme pour manipuler les données du langage humain. NLTK guide le lecteur à travers les principes fondamentaux de l'écriture de programmes Python, le travail avec les corpus, la catégorisation du texte, l'analyse de la structure linguistique, et plus encore..

3.2 Réalisation :

Pour bien comprendre toutes les fonctionnalités de l'application on suit deux scénarios selon les acteurs :

3.2.1 Guest :

à l'ouverture de l'application le patient voit la page d'accueil suivante où il peut faire sa recherche, Pour faire sa recherche , le Guest peut écrire juste une description generale du produit et (Figure 3.2) :

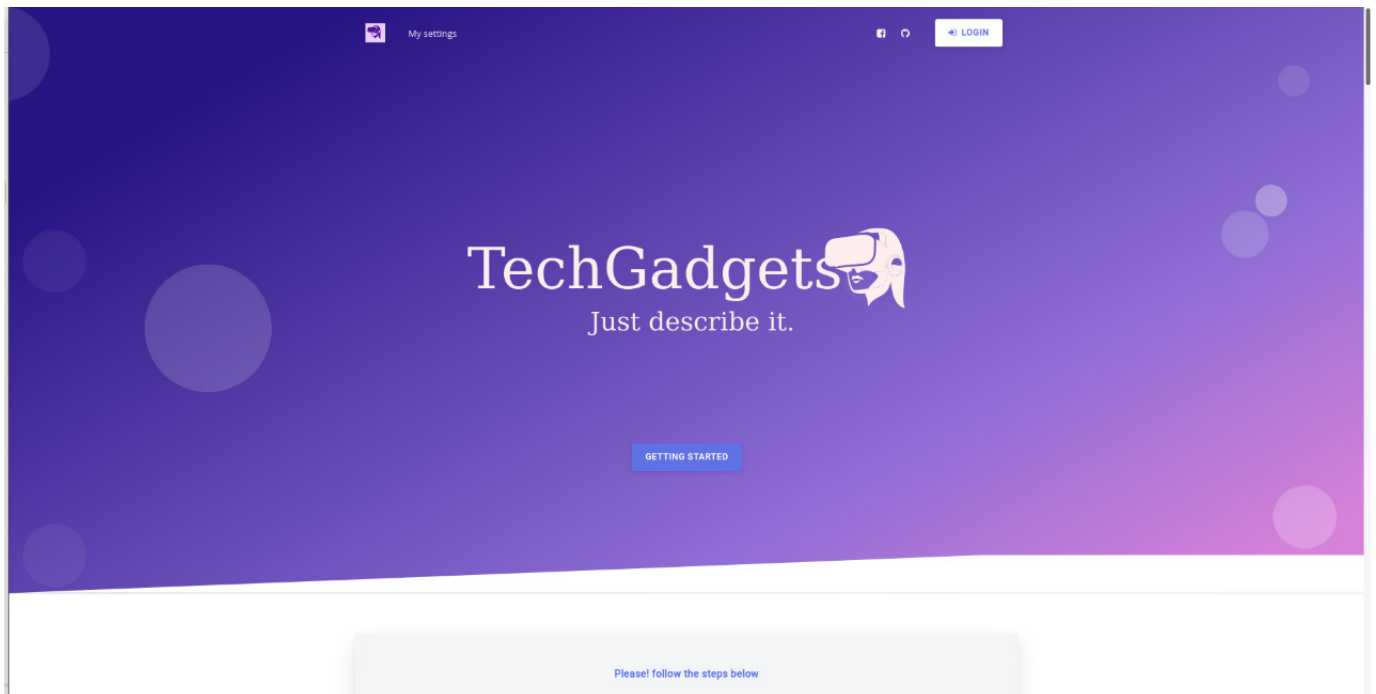


FIGURE 3.1 – Page d'accueil/ propriétés TechGadgets

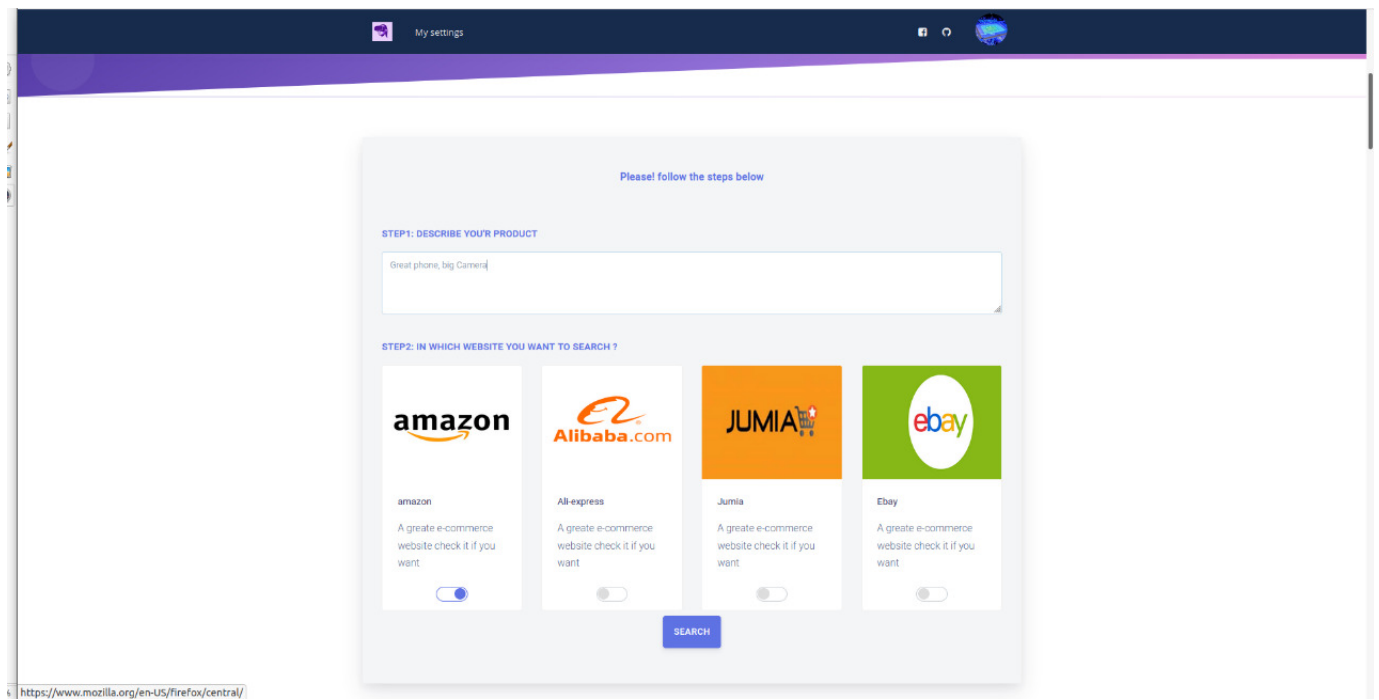


FIGURE 3.2 – Page d'accueil/ barre recherche

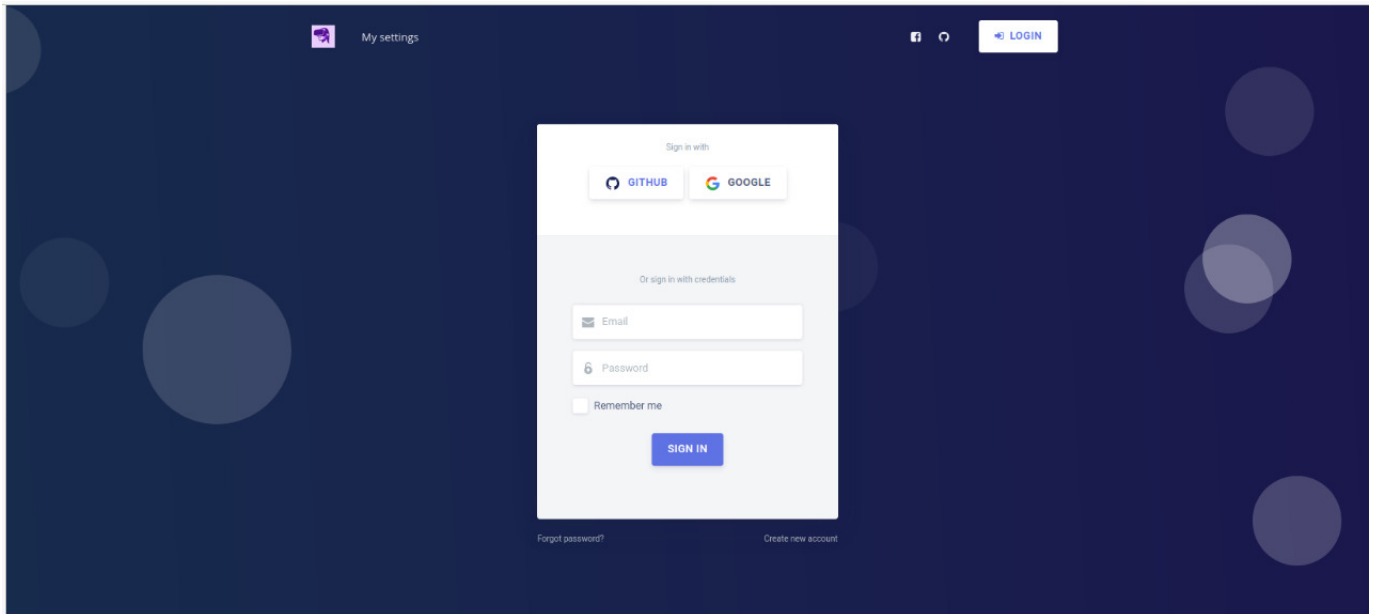


FIGURE 3.3 – Page d'accueil/ Login et Signup

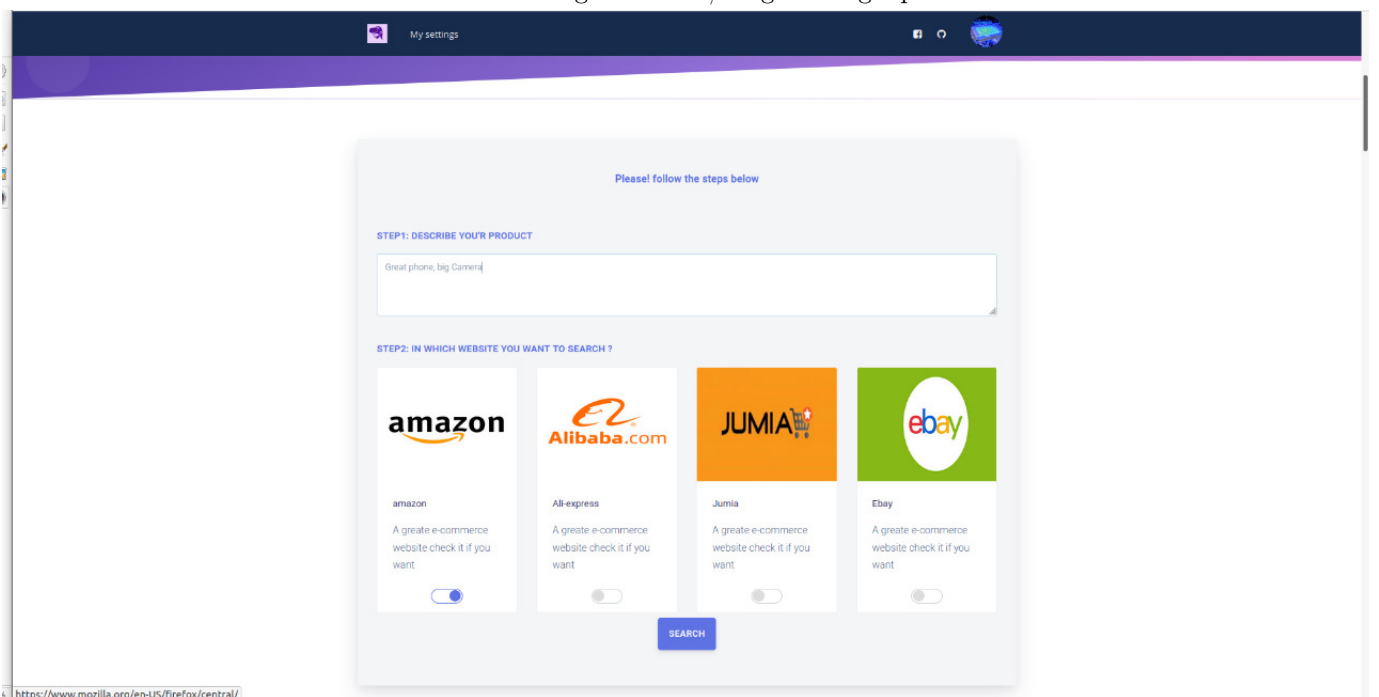


FIGURE 3.4 – Page d'accueil/ barre recherche

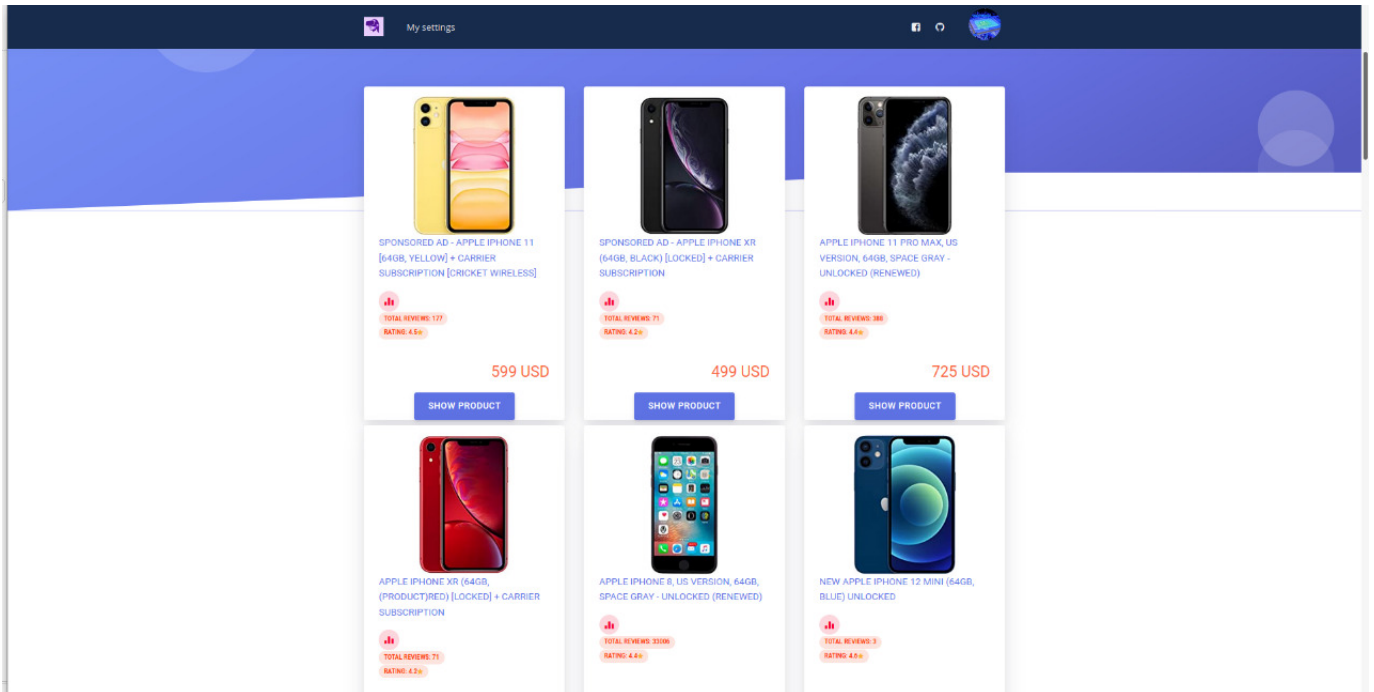


FIGURE 3.5 – Les produits recommandés

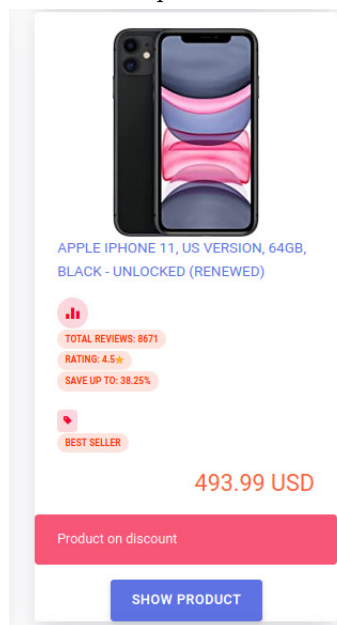


FIGURE 3.6 – Les détails spécifiques aux abonnés

Conclusion générale

Le travail que nous avons fait a consisté sur la réalisation d'une application web de recommandation . Nous avons fait une petite analyse aux situations des utilisateur. nous avons fourni aussi l'architecture générale de notre application, y compris le workflow de l'interaction de l'utilisateur avec l'application et les détails de notre modèle de Machine Learning.

Ce projet nous a permis d'acquérir une expérience personnelle et professionnelle. Il nous a été bénéfique car nous avons eu la chance d'améliorer nos connaissances en développement web, Architecture Microservices, Machine Learning et aussi de développer une capacité de résolution des problèmes vu qu'on a eu une difficulté dans le choix du framework le plus convenable pour notre application ainsi dans la recherche des solutions des erreurs qui , évidemment, apparaissent partout dans le code.

Arrivant à ce point là, nous restons encore ambitieux pour faire des améliorations qu'on n'a pas eu le temps pour les réaliser dans la période du module à savoir :

- L'authentification d'utilisateur par Amazon Account
- L'intégration des plateformes des avis techniques
- la possibilité aux utilisateurs de contribuer à notre Modèle

Bibliographie

- [1] <https://angular.io/docs>. Site officiel du Angular.
- [2] <https://spring.io/projects/spring-bootoverview>. Spring Boot.
- [3] <https://spring.io/projects/spring-cloud>. Site officiel de Spring CCloud.
- [4] <https://www.nltk.org/>. NLTK documentation.
- [5] <http://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>. Article sur NLP.