



École Nationale Supérieure d'Informatique et d'Analyse des Systèmes - RABAT

# Rapport de Projet de fin d'année : APPLICATION WEB DE COMPARAISON DE PRIX

*Réalisé par :*

Adel Dahani
Oussama Alami

*Encadré par :*

Pr. Mohamed Naoum

Année académique 2020/2021

# *Remerciements :*

C'est parce que nous avons beaucoup estimé tous ceux qui nous ont écoutés, conseillés, critiqués et encadrés que nous tenons à leur faire part de toutes nos gratitudes, et plus particulièrement, nous tenons à remercier à travers ces courtes lignes :

**Monsieur Mohamed NAOUM**, qui a acceptée d'examiner et d'encadrer ce travail pour le valoriser malgré ses nombreuses occupations. Nos remerciements vont également à nos professeurs des modules de « *Méthodes de résolution de programmation mathématiques* » : **Monsieur Mohamed LAZAAR** pour son entière disponibilité, « *Optimisation Non Linéaire* » : **Monsieur Abdellatif EL AFIA** pour le savoir qu'il nous a transmis. Enfin, nos remerciements s'adressent aux professeurs et au corps administratif de l'**Ecole Nationale Supérieure de l'Informatique et d'Analyse des Systèmes**.

# Table des matières

# Table des figures

# Introduction générale

Dernièrement, beaucoup de gens reviennent aux outils informatiques afin de faciliter leurs vies quotidiennes. Ces outils peuvent être utilisés dans tous les secteurs, à savoir : le secteur professionnel, **secteur éducatif** et d'autres secteurs.

La plupart des acheteurs en ligne perdent beaucoup de temps à trouver les produits dont ils ont besoin à un prix abordable. Le rôle de l'informatique est de résoudre ce problème en développant une application Web de **comparaison de prix**.

Ce rapport retrace, à travers les différentes étapes que nous avons suivies pour atteindre cet objectif, ces étapes peuvent être résumées en deux chapitres :

⇒ *Le premier chapitre est consacré à la présentation de la problématique et le cadre générale du projet, ainsi que les outils de la résolution du problème.*
⇒ *Le dernier chapitre aborde la phase développement et mise en oeuvre, avec une présentation de l'application, ainsi qu'une description des difficultés rencontrées.*

# Chapitre 1

# Analyse et conception

## 1.1 Problématique

Il s'agit de réaliser une application Web qui permet de scraper des produits de sites marketing de technologie afin de les utiliser pour afficher les produits par ordre croissant de prix.

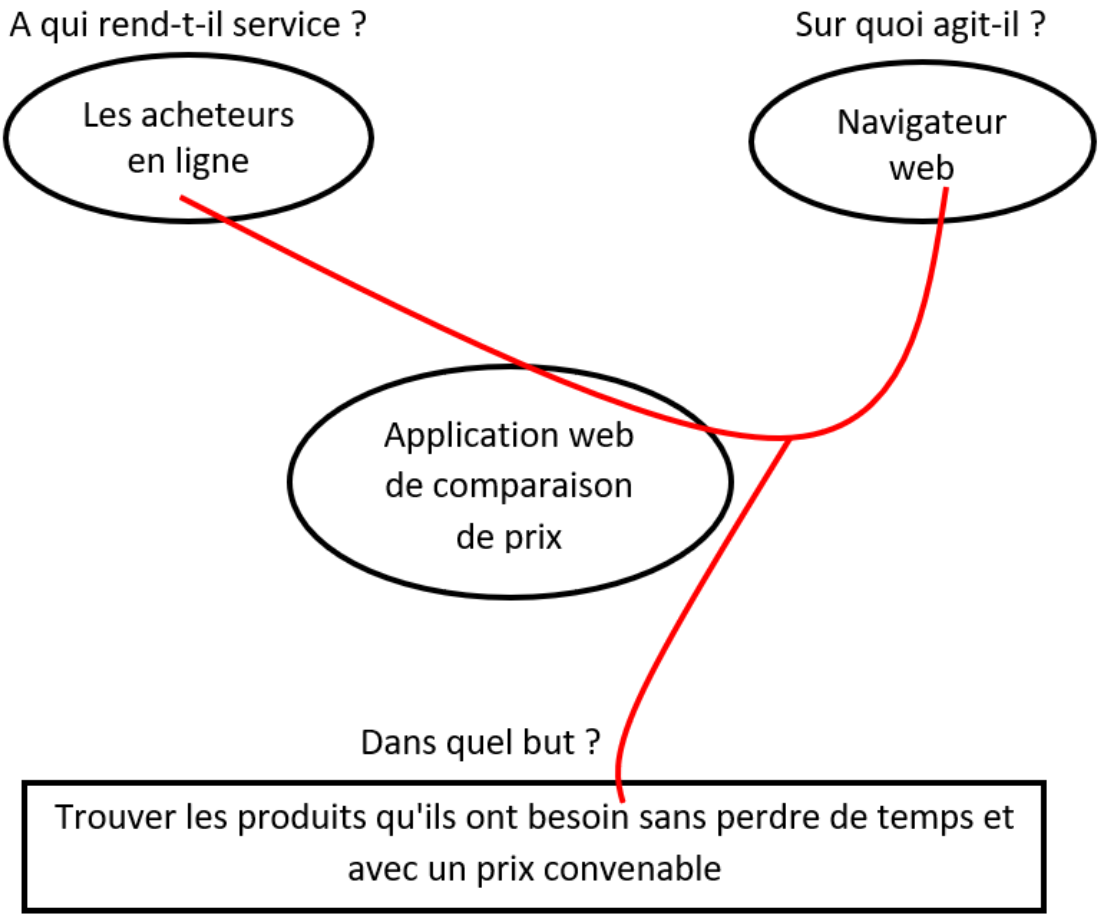


FIGURE 1.1 – Diagramme bête à corne de l'application

### Web Scraping et Web Crawling :

- Le **Web Scraping** fait référence à l'extraction de données à partir d'un site Web ou d'une page Web.
- Le **Web Crawling** fait référence au processus d'utilisation de robots (spiders) pour lire et stocker tout le contenu d'un site Web à des fins d'archivage ou d'indexation.

## 1.2 Cahier des charges

### 1.2.1 Contexte et objectifs

De nos jour , les gens ont obligés de perdre le temps pour trouver les produits qu'ils ont besoin avec un prix convenable. Or il y a des outils informatique pour faciliter la tache et gagner du temps. on est obligé de finaliser ce projet qui consiste à assurer une source de défis et de développement de logique. Notre objectif à travers ce travail est de construire une application Web qui répond aux besoins de l'utilisateur.

### 1.2.2 Les besoins fonctionnels

Après une étude détaillée du système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces exigences sont :

Un utilisateur peut faire :

- **Recherche générale :** Notre site Web propose une recherche générale à partir des sites Web les plus populaires comme amazon.
- **Recherche par catégorie :** Pour optimiser les résultats, les utilisateurs peuvent effectuer une recherche par catégorie, ce qui leur permet d'obtenir un résultat des meilleurs sites Web pour cette catégorie.
- **Changer de devise :** L'utilisateur peut également modifier la devise du résultat avec une grande précision (les devises sont en mises à jour toutes les heures).

### 1.2.3 Les besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et en ce qui concerne notre application, nous avons dégagé les besoins suivants :

- **La disponibilité :** L'application Web doit être disponible pour être utilisé par n'importe quel utilisateur.
- **La performance :** L'application Web doit réagir dans un temps convenable, quel que soit l'action de l'utilisateur.
- **La synchronisation :** Plusieurs ulisateurs peuvent chercher dans l'application Web sans perdre de performance.

### 1.2.4 Caractéristiques de l'application

- Design simple et conviviale.
- Langue utilisée : anglaise.

## 1.3 Les étapes de la résolution du problème

Avant tout, la gestion du programme s'articule essentiellement sur la gestion des bases de données pour stocker les données dont nous avons besoin.

### 1.3.1 Base de données des catégories

Le but de cette base de données est de stocker les sites que nous voulons scraper dans chaque catégorie ; elle contient deux attributs :

— **category** : La catégorie du site web
— **website** : Le nom du site web

### 1.3.2 Base de données des produits

Cette base de données a pour objectif stocker les résultats de recherche des utilisateurs (les résultats des crawlers) ; elle contient les attributs suivants :

— **search_id** : Un identificateur unique pour distinguer chaque recherche
— **source** : La source de ce produit
— **name** : Le nom du produit
— **price** : Prix du produit
— **url** : Le lien du produit
— **img** : L'image du produit

### 1.3.3 Base de données des devises

Le but de cette base de données est de stocker la valeur de chaque devise par rapport au dirham marocain ; elle contient deux attributs :

— **symbol** : Le symbole de devise
— **rate** : La valeur de devise

# Conclusion

Après avoir dégagé les besoins fonctionnels et opérationnels et toute les critères qu'on doit prendre en considération, et afin de modéliser les besoins attendus de notre application et que les objectifs soient atteints, on va suivre la démarche du processus unifié qu'on va le détailler dans la prochaine partie.

# Chapitre 2

# Mise en œuvre et réalisation

## 2.1 Les outils utilisés

◦ **Les outils de scraping :**


FIGURE 2.1 – Scrapy

**Scrapy** est un framework open-source permettant la création de robots d'indexation. Développé en Python, il dispose d'une forte communauté, offrant de nombreux modules supplémentaires.

**Scrapyd** est une application pour déployer et exécuter les crawler *(spider)* de Scrapy. Il vous permet de déployer (télécharger) vos projets et de contrôler leurs crawler à l'aide d'une API JSON.

**Schedule** Schedule est une bibliothèque Python qui permet la planification des scripts pour les humains, et les exécutez périodiquement (ou tout autre appelable) en utilisant une syntaxe conviviale.

◦ **Les outils de frontend :**


FIGURE 2.2 – HTML


FIGURE 2.3 – CSS


FIGURE 2.4 – Sass


FIGURE 2.5 – JavaScript

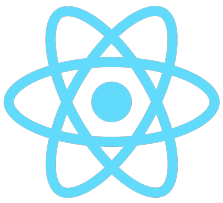

FIGURE 2.6 – React

◦ **Les outils de backend :**


FIGURE 2.7 – Django


FIGURE 2.8 – Django REST framework

### 2.1.1 Les caracteristiques des outils utilisés

- **Scrapy :** Plus rapide et plus performant que les autres outils.
- **Scrapyd :** Resoudre le probleme de synchronisation.
- **React :** Rapide en term de chargement des composantes.
- **Django rest framework :** Facilite la communication entre frontend et backend.

## 2.2 Présentation de l'application

### 2.2.1 Page principale

Cette barre représente la barre de recherche des produits avec catégories.

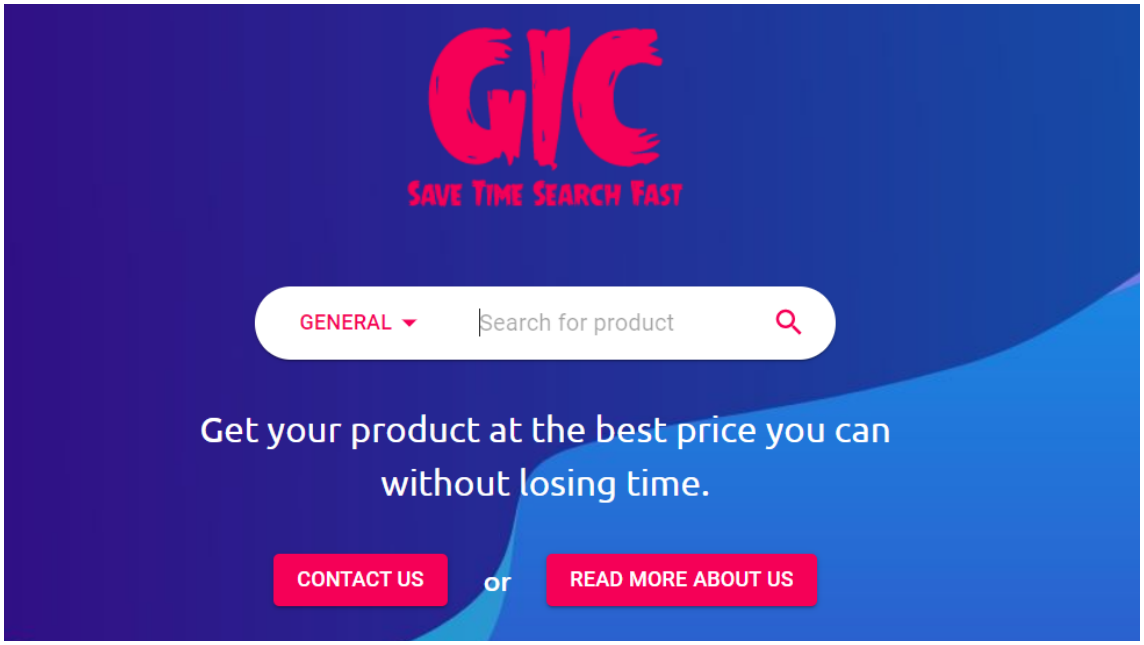


FIGURE 2.9 – Barre de recherche

### 2.2.2 Barre de navigation

Cette barre est statique , existe dans tous les pages, elle contient un boutton pour la page pricipale , d'autres pour "about us" et "contact us".

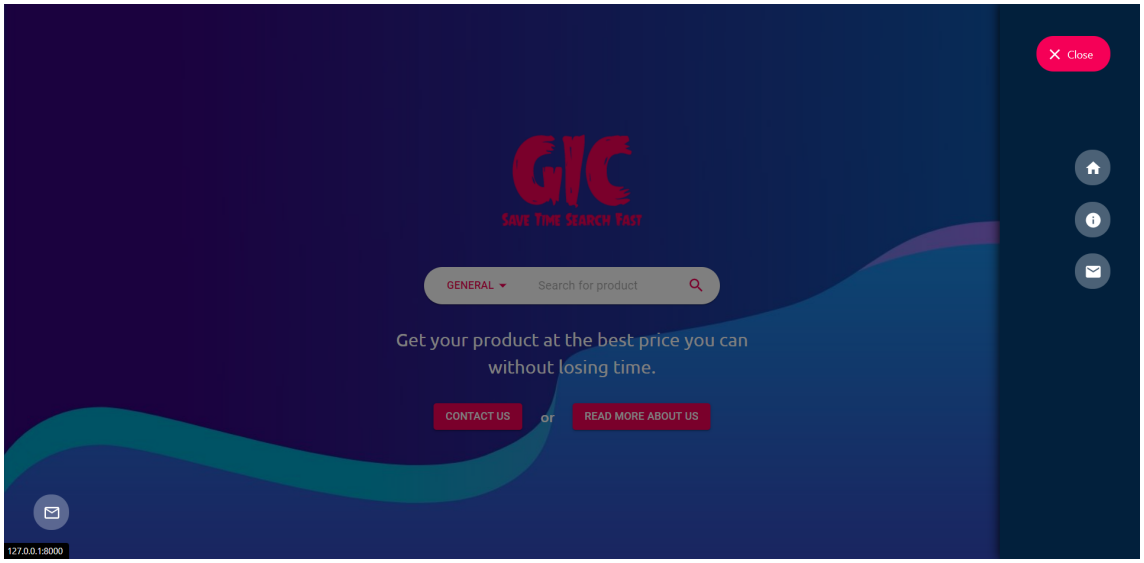


FIGURE 2.10 – Barre de navigation

### 2.2.3 Menu des catégories

Avant de faire la recherche du produit, l'utilisateur peut choisir une catégorie precise ou bien laisser la recherche general.

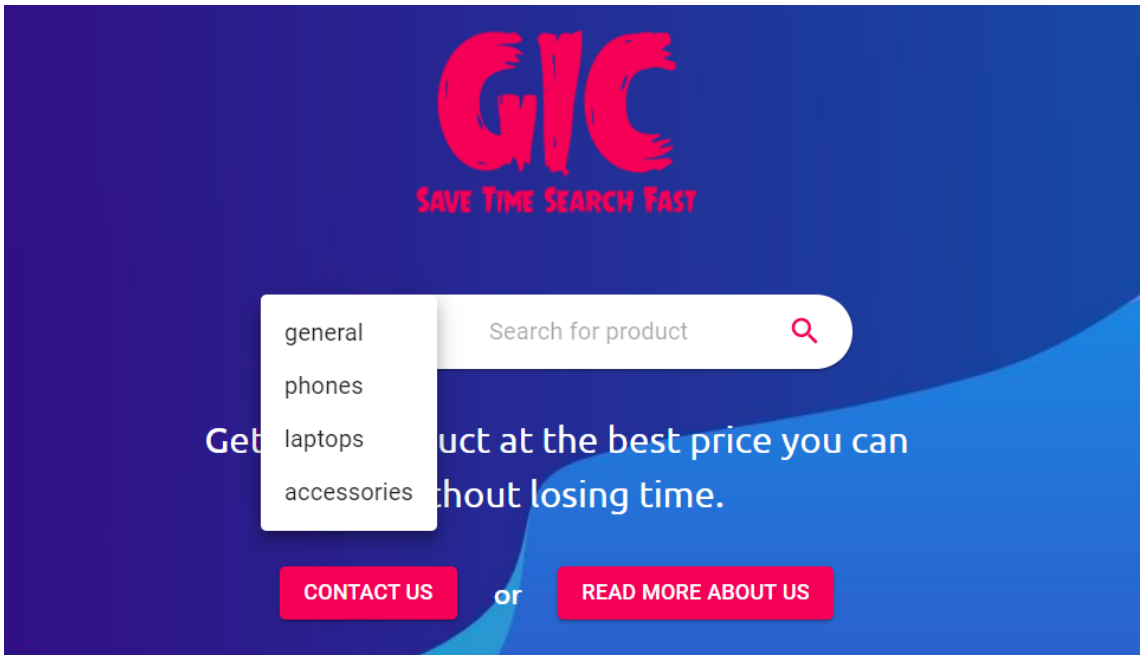


FIGURE 2.11 – Menu des catégories

### 2.2.4 Page de chargement

Cette page est mis en œuvre lorsque la recherche de données est en cours d'exécution.

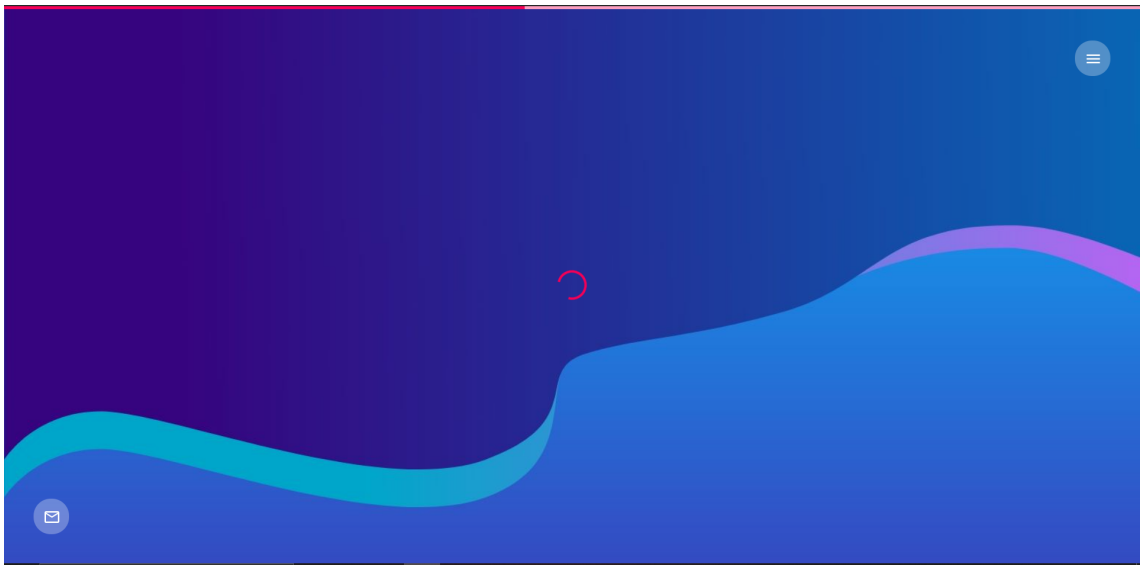

FIGURE 2.12 – Page de chargement

### 2.2.5 Page des résultats

Cette page affiche avec les résultats désirés.

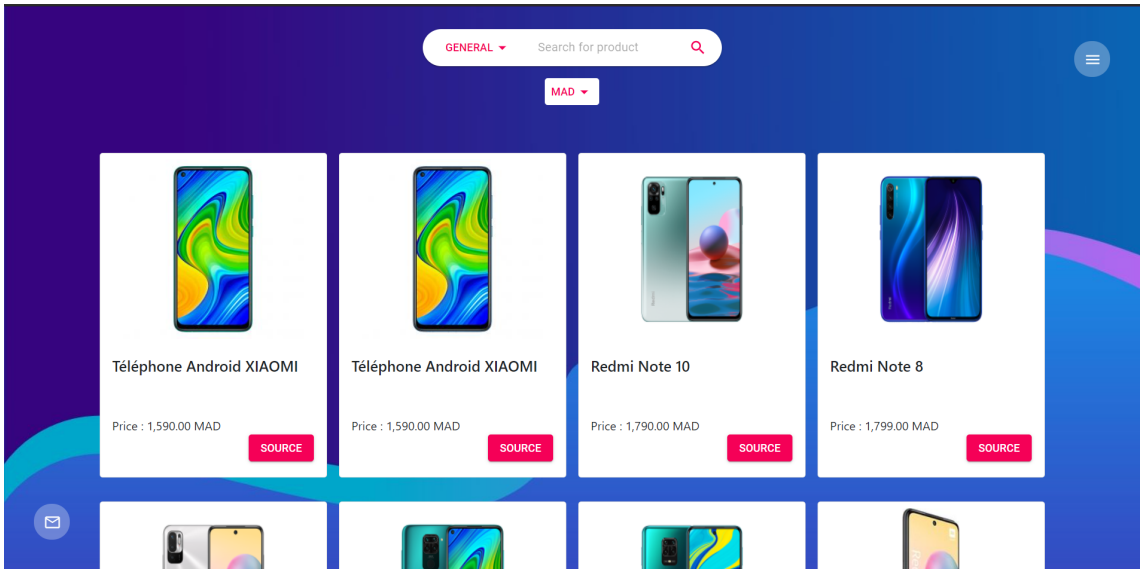


FIGURE 2.13 – Page des résultats

### 2.2.6 Table des devises

Cette table contient les devises des prix désirés, elle est par défaut en MAD .

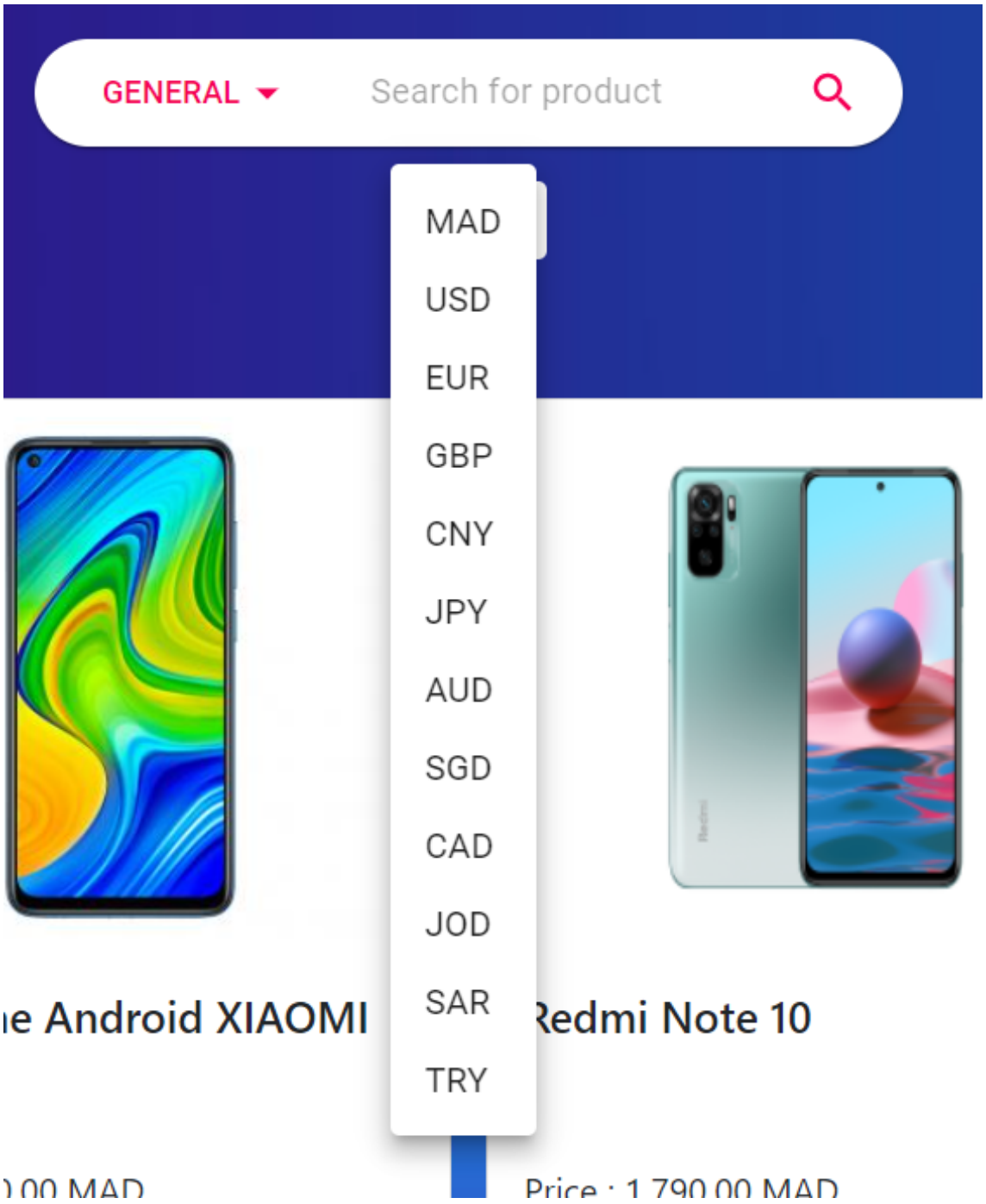


FIGURE 2.14 – Table des devises

## 2.3 Les difficultés rencontrées

### 2.3.1 Connection entre Django et Scrapy

Pour connecter Django et Scrapy il faut automatiser le script de scraping à partir du produit écrit par l'utilisatuer et envoyer les résultats de script. Alors on a besoin de stocker le résultat des spiders dans la base de donnée de Django pour l'utiliser à notre interface (Frontend).
Le problème a été resolu par l'utilisation d'une bibliothèque **Django Item** qui fait la liaison entre le model de Django (notre table de Django) et notre spider.

### 2.3.2 La synchronisation

On a rencontré un probleme de syncronisation des recherches, mises en place à la fois par deux utilisateurs (au plus). Le serveur django se bloque dans ce cas. Le problème a été resolu par **scrapyd**, car ce dernier exécute le scripte de recherche (crawler) en un autre serveur isolé.
D'où plusieurs utilisateurs peuvent rechercher en même temps.

# Conclusion générale

Notre projet a consisté à la réalisation d'une application web ''comparison de prix''. Ce projet nous a permis d'approfondir nos connaissances théoriques, acquises tous le long de notre formation, par la pratique des nouvelles technologies. Cette expérience nous a permis de maîtriser les technologies de web ainsi que le frame-work scrapy.
Ce rapport explicite une première version du projet. L'ensemble des fonctionnalités demandées dans le cahier de charges ont été réalisées et implémentées dans le temps et fonctionnent correctement.
L'amélioration qu'on peut voir dans ce projet est le fait d'ajouter d'autre manière de sorte des données : par rapport aux qualité, popularité et feedback des utillisateurs.

# Bibliographie


[1] <https://docs.djangoproject.com/en/3.2//>. [Documentation de Django].

[2] <https://www.django-rest-framework.org/>. [Documentation de Django REST framework].

[3] <https://scrapy.org/doc/>. [Documentation de Scrapy].

[4] <https://alioguzhan.medium.com/how-to-use-scrapy-with-django-application-c16fabd0e62e>. [Connection entre Django et Scrapy].

[5] <https://schedule.readthedocs.io/en/stable/>. [Documentation de Schedule].