



## Master spécialisé en Data science pour l'économie et la finance



# Projet de fin de module : Langue de programmation.

E-epicier.

2023

**Sous la direction :**

Prof : EL AACHAL Lotfi

**Réalisé par :**

BELLAHSINI Anass

LICHARI Abd el monim

SARHI Salma

github : <https://github.com/bellahsinianass/e-epicier->

github :

github :



## ***remerciement :***

Nous tenons à vous remercier du fond du cœur pour votre dévouement et votre passion dans l'enseignement de la langue de programmation. Vos compétences techniques, votre patience et votre engagement envers nos apprentissages ont été une source d'inspiration et de motivation tout au long de ce cours.

Grâce à vos enseignements, nous avons acquis une solide compréhension de la langue de programmation, ce qui nous a non seulement permis de réussir ce cours, mais également de nous ouvrir à de nombreuses opportunités professionnelles dans le domaine de la programmation.

Nous sommes reconnaissants pour les défis que vous nous avez lancés, les conseils précieux que vous nous avez prodigués et les ressources que vous nous avez partagées. Votre enseignement a été une expérience inoubliable, qui nous a aidé à développer nos compétences et à élargir nos horizons.

Encore une fois, nous vous remercions de tout cœur pour votre engagement envers notre réussite. Vos contributions ont été inestimables et ont eu un impact positif sur nos vies professionnelles.



## ***Plan :***

### **Contexte général**

#### **Chapitre 1 : les framework adopté au mise en place du projet.**

1. Figma
2. MySQL
3. Le framework Bootstrap
4. Django

#### **Chapitre 2 : La phase de conception du projet.**

1. Le diagramme Uses Cases.
2. Le diagramme de classe.

#### **Chapitre 2 : les différentes étapes de la mise en place du projet.**

1. Authentification par un login et mot de passe.
2. Gestion des clients.
3. Gestion des produits/articles.
4. Gestion des crédits par clients.



## **Contexte général :**

L'industrie de l'épicerie a connu une transformation numérique significative ces dernières années avec l'avènement des applications web pour la gestion des entreprises. Aujourd'hui, les épiciers cherchent à automatiser davantage leurs opérations pour améliorer l'efficacité de leurs activités quotidiennes.

Dans cette optique, la création d'une application web E-épiciier peut être une solution très intéressante pour les épiciers souhaitant améliorer leur gestion des produits et les crédits de client. Cette application permettrait de centraliser toutes les informations relatives aux crédits clients, de manière à faciliter la gestion des remboursements et à minimiser les risques d'erreurs humaines.

L'application web E-épiciier serait conçue pour permettre aux épiciers de suivre les dépenses de leurs clients en temps réel, ainsi que de vérifier les soldes de crédit. Les clients pourraient également consulter leur propre solde de crédit et effectuer des paiements en ligne directement depuis l'application. Les épiciers pourraient également envoyer des notifications de rappel de paiement à leurs clients pour éviter les retards de paiement.

Cette application web E-épiciier permettrait également aux épiciers de générer des rapports sur les crédits clients, offrant ainsi une visibilité accrue sur la situation financière de leur entreprise. Les rapports générés pourraient aider les épiciers à prendre des décisions plus éclairées en matière de gestion de crédit, ce qui pourrait avoir un impact positif sur la rentabilité de leur entreprise.

En somme, la création d'une application web E-épiciier permettrait aux épiciers de simplifier la gestion de leurs crédits clients, d'améliorer leur efficacité opérationnelle, et de prendre des décisions plus éclairées en matière de gestion financière. Cette solution numérique pourrait ainsi offrir un avantage compétitif à ceux qui l'adoptent en leur permettant de mieux se concentrer sur leur cœur de métier.



## **Chapitre 1 : les framework adopté au mise en place du projet.**

### **1. Django :**

Django est un framework web open-source écrit en Python qui permet de développer rapidement des applications web robustes et évolutives. Il suit le principe de conception Model-View-Controller (MVC), également connu sous le nom de Modèle-Vue-Contrôleur en français.

Le modèle de données dans Django est défini en utilisant des classes Python, et les vues et les templates sont créés à l'aide de langages de modèle tels que HTML, CSS et JavaScript. Django fournit également une interface d'administration prête à l'emploi pour gérer facilement les données du site web.

Les fonctionnalités clés de Django incluent l'ORM (Object-Relational Mapping) qui permet de manipuler des données de manière abstraite, ce qui facilite la communication avec la base de données. Django prend également en charge la création automatique de formulaires pour l'entrée utilisateur et la validation des données.

Django est modulaire, ce qui permet aux développeurs d'ajouter facilement de nouvelles fonctionnalités ou de personnaliser les fonctionnalités existantes. Les extensions Django sont appelées "applications" et peuvent être utilisées dans d'autres projets. De plus, Django est livré avec de nombreuses applications préconstruites, telles que l'authentification utilisateur et l'interface d'administration.

Django est un choix populaire pour les sites web de toutes tailles, des petites applications web aux sites web à trafic élevé. Il est utilisé par de nombreuses grandes entreprises et organisations telles que Instagram, Pinterest et Mozilla.

Dans l'ensemble, Django est un framework web puissant et polyvalent qui facilite la création de sites web robustes et évolutifs.

### **2. MySQL :**

MySQL est un système de gestion de base de données relationnelle (SGBDR) open-source qui utilise le SQL (Structured Query Language) pour gérer et manipuler des données. Il est un choix populaire pour les applications web et est largement utilisé par des entreprises et des organisations de toutes tailles.

MySQL peut être utilisé pour stocker et récupérer des données à partir de tables, qui sont des collections de données connexes. SQL est utilisé pour créer et modifier des tables, insérer, mettre à jour et supprimer des données, et récupérer des données en fonction de certains critères. Certaines commandes SQL courantes utilisées dans MySQL comprennent SELECT, INSERT, UPDATE, DELETE et JOIN.

MySQL inclut également des fonctionnalités telles que la prise en charge de transactions, de déclencheurs, de procédures stockées et de vues. Les transactions permettent de réaliser plusieurs modifications à la base de données en une seule opération, garantissant que toutes les modifications sont effectuées ou aucune. Les déclencheurs sont des procédures qui sont automatiquement exécutées en réponse à certains événements, tels qu'une insertion ou une mise à jour d'une table. Les procédures stockées sont des routines pré-écrites qui peuvent être appelées à partir d'une application ou d'un autre code, et les vues sont des tables virtuelles qui peuvent être utilisées pour simplifier les requêtes complexes.

MySQL peut être utilisé avec une variété de langages de programmation, y compris PHP, Java et Python, et il peut être exécuté sur une variété de systèmes d'exploitation, y compris Windows, Linux et macOS. Il prend également en charge la réplication, ce qui permet de copier des données entre plusieurs serveurs pour une performance et une fiabilité accrues.

Dans l'ensemble, MySQL est un SGBDR puissant et polyvalent qui convient bien à une large gamme d'applications et de cas d'utilisation.

### **3. Le framework Bootstrap:**

Bootstrap est un framework de développement web open-source, qui permet de créer rapidement et facilement des sites web et des applications web responsives.

Bootstrap fournit un ensemble de composants HTML, CSS et JavaScript préconçus, tels que des boutons, des formulaires, des tableaux, des grilles, des menus de navigation, des alertes et bien plus encore, qui facilitent le développement de pages

web réactives et attrayantes. Ces composants peuvent être personnalisés en fonction des besoins du projet.

Le framework Bootstrap est conçu pour être facilement personnalisable et extensible, et peut être utilisé avec d'autres frameworks et bibliothèques de développement web. Il est également compatible avec les derniers navigateurs web et les appareils mobiles.

Bootstrap est un choix populaire pour les développeurs web de tous niveaux, car il permet de gagner du temps dans le développement de sites web et d'applications web responsives, tout en offrant une grande flexibilité et une grande adaptabilité.

En résumé, Bootstrap est un framework de développement web puissant, flexible et facile à utiliser, qui permet de créer rapidement des sites web et des applications web responsives et attrayants.

#### **4. Figma:**

Figma est un outil de conception d'interface utilisateur basé sur le web utilisé pour créer des interfaces utilisateur, des prototypes et des projets de conception collaboratifs. Il permet aux concepteurs de créer des graphiques vectoriels et des prototypes interactifs, et de collaborer en temps réel avec d'autres membres de l'équipe sur un projet de conception partagé.

Figma est populaire pour sa capacité à permettre à plusieurs utilisateurs de collaborer en temps réel, ce qui en fait un outil idéal pour les équipes de conception qui travaillent à distance ou dans des endroits différents. L'outil comprend des fonctionnalités telles que les commentaires, le contrôle des versions et le partage, qui permettent aux équipes de travailler ensemble de manière transparente et efficace.

Les outils de conception de Figma comprennent l'édition vectorielle, les opérations booléennes, l'édition de texte et une vaste bibliothèque d'icônes et de graphiques. L'outil comprend également une large gamme de plugins et d'intégrations, permettant aux utilisateurs d'étendre ses fonctionnalités et de s'intégrer à d'autres outils et services.

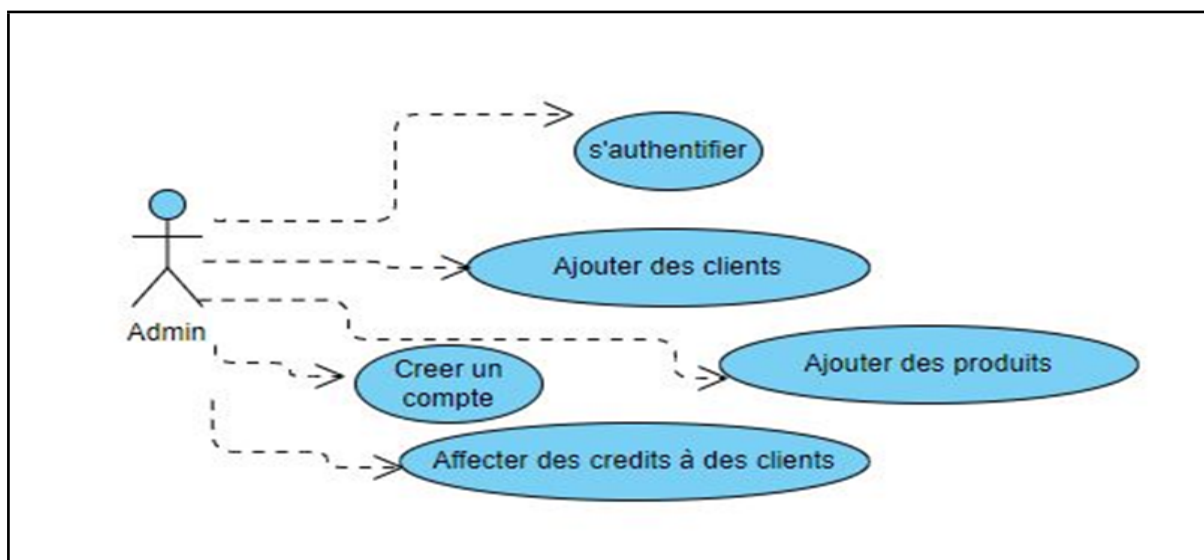
Figma est utilisé par un large éventail de concepteurs, des concepteurs UI / UX aux graphistes, et est largement reconnu pour son interface conviviale, sa polyvalence et ses capacités collaboratives.

## Chapitre 2 : La phase de conception du projet.

La phase de la conception c'est la phase la plus importante avant de mettre la main sur le développement, c'est la phase d'où on définit nos spécifications et les fonctionnalités de notre application web.

### 1. Le diagramme Uses Cases:

Notre acteur c'est l'admin, un admin peut s'authentifier d'abord avant chaque action. il peut créer un compte avant si n'a pas déjà un. De plus, un admin après l'authentification peut ajouter des clients ajouter des produits et affecter des crédits à des clients.



### 2. Le diagramme de classe:

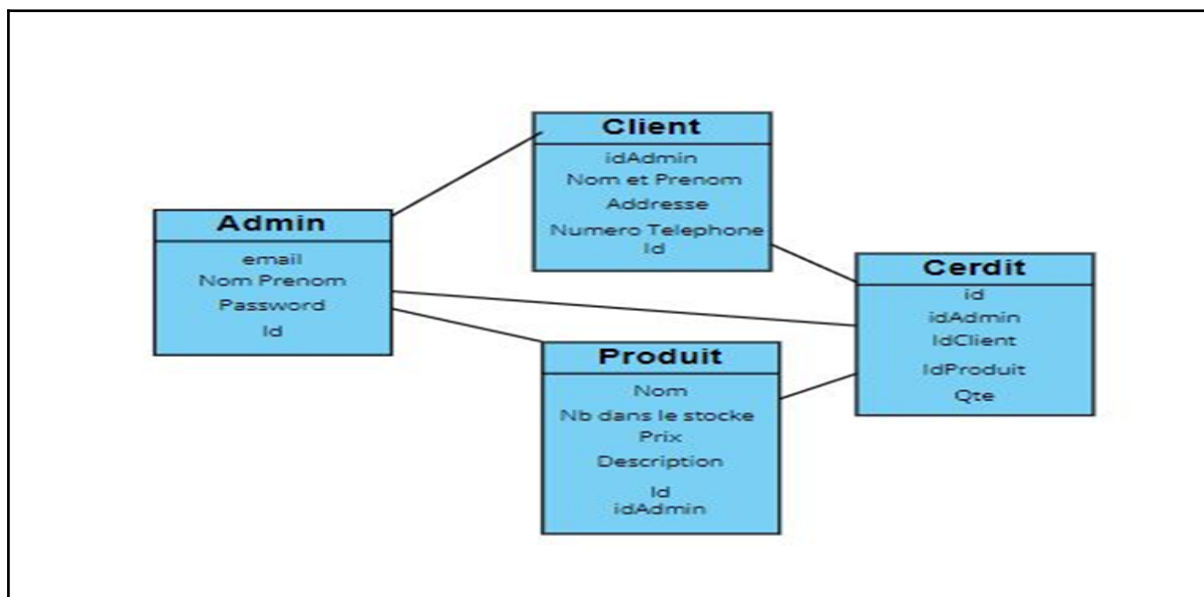
Concernant notre Diagramme de classe, on a vu notre système autant que 3 classes:

Client : c'est la classe qui présente les clients dans notre système, il a des attributs à savoir Le nom et le prénom, Adresse, téléphone.

Produit : c'est la classe qui présente les produits dans notre système, il a des attributs à savoir Le nom, Prix, Description, Quantité.



Crédit : c'est la classe qui présente les Crédits dans notre système, il contient des informations sur produit, client et la Quantité.



### Chapitre 3 : les différentes étapes de la mise en place du projet.

#### 1. Authentification par un login et mot de passe:

- Back-End:

```
4 class Admin(models.Model):
5     nom = models.CharField(max_length=50)
6     prenom = models.CharField(max_length=50)
7     password = models.CharField(max_length=50)
8     username = models.CharField(max_length=20)
9     created_on = models.DateTimeField(auto_now_add=
0
```

- **Front-End:**

Afin de faire un prototype de notre application, on a utilisé L'outil Figma pour réaliser les maquettes. Notre application web contient les pages à savoir :

**Page Login :** C'est la page d'où l'admin doit s'authentifier pour accéder aux autres pages à l'aide de leur email et mot de passe. C'est la page d'où l'admin peut s'authentifier pour accéder à notre application à l'aide de leur email valide et leur mot de passe, sinon il doit créer un compte à l'aide de cliquer sur le mot créé au-dessous de la page.

**Page Register :** C'est la page d'où l'admin peut créer un compte login. Après qu'un utilisateur de notre application web à cliquer sur le mot Créé, il va trouver notre deuxième c'est la page d'où il peut créer un nouveau compte, il faut tout simplement remplir la formule par leurs informations (nom, email, mot de passe choisie).

The image displays two wireframe designs for the Epecier application, presented as screenshots within a larger frame. Both wireframes feature a light yellow background and a dark blue square on the left containing the word "Epecier" in white serif font.

The top wireframe represents the "Register" page. On the right side, there is a white rectangular area containing the title "Register" in blue. Below the title are four input fields with light yellow backgrounds and thin blue borders, labeled "Nom", "Prenom", "Username", and "Password". Underneath these fields is a dark blue button with the text "S'inscrire" in white. At the bottom of the white area, there is a link that reads "Vous avez un compte Connectez Vous".

The bottom wireframe represents the "Login" page. It follows the same layout as the Register page. The title "Login" is in blue. It has two input fields labeled "Username" and "Password". Below them is a dark blue button with the text "Se Connecter" in white. At the bottom, the link reads "Vous avez un compte S'inscrire ICI".

Après la validation de leur compte, l'admin peut accéder à l'application, il va trouver un menu qui contient trois axe :

· Clients · Produits · Crédits

Dans chaque axe on a deux pages, qu'on va les voir en détail par la suite.

## 2. Gestion des clients:

- **Back -End:**

```
11 class Client(models.Model):
12     nom = models.CharField(max_length=50)
13     prenom = models.CharField(max_length=50)
14     adresse = models.CharField(max_length=255)
15     phoneNumber = models.CharField(max_length=20)
16     admin=models.ForeignKey(Admin, on_delete=models.CASCADE)
17     total=models.FloatField()
18     created_on = models.DateTimeField(auto_now_add=True)
19
```

- **Front-End: Pour les clients, on a deux pages :**

**Nouveau Client :** C'est la page d'où on peut ajouter des nouveaux clients.

Il suffit de remplir les champs suivants pour qu'un admin peut ajouter un nouveau client: leur nom, leur adresse et leur numéro du Téléphone, et cliquer sur ajouter pour enregistrer les informations saisies.

**Gestion Clients :** Notre deuxième page est la page de gestion des clients quand on a déjà ajouté : c'est la page d'où on affiche tous les clients enregistrés (leur nom, leur adresse et leur numéro du Téléphone), de plus la possibilité d'effectuer des opérations consiste à éditer un client ou bien le supprimer.

Pour faciliter la gestion, on a réalisé un champ Chercher un client pour trouver le client que l'admin cherche rapidement.

Clients

Produits

Credits

M.Admin

LOGOUT

Créer Gérer

Ajouter un nouveau Client à votre carnet

Nom

Prenom

Adresse

Telephone

Ajouter

Clients

Produits

Credits

M.Admin

LOGOUT

Créer Gérer

Consultez vos Clients

Trouvez Clients

Search

Nom	Prenom	Adresse	Telephone	Credits	Actions
anass	bellahsini	adresse11	062255488	50.0	<div>Editer</div> <div>Payer</div> <div>Credits</div>
client1	prenom1	adresse1	05555448	0.0	<div>Editer</div> <div>Payer</div> <div>Credits</div>
test	ahmed	adresse3	554899611	0.0	<div>Editer</div> <div>Payer</div> <div>Credits</div>

### 3. Gestion des produits/articles:

- Back -End:

```
# Create your models here.
class Produit(models.Model):
    nom = models.CharField(max_length=50)
    image = models.ImageField(upload_to='images')
    commentaire = models.TextField()
    prix=models.FloatField()
    admin=models.ForeignKey(Admin, on_delete=models.CASCADE)
    created_on = models.DateTimeField(auto_now_add=True)
```

- **Front-End: Pour les produits, on a deux pages aussi :**

**Nouveau Produit :** C'est la page d'où on peut ajouter des nouveaux produits. Il suffit de remplir les champs suivants pour qu'un admin peut ajouter des nouveaux produits: le nom du produit, leur prix, la quantité et la description, et cliquer sur ajouter pour enregistrer les informations du nouveau produit.

**Gestion Produits :** Notre deuxième page est la page de gestion produits : c'est la page d'où on affiche les produits enregistrés (le nom du produit, quantité, prix et leur description), de plus la possibilité d'effectuer des opérations éditer un produit ou bien le supprimer. Pour faciliter la gestion, on a réalisé un champ Chercher un produit pour trouver le produit que l'admin cherche rapidement.

The image displays two screenshots of a web application interface for product management. Both screenshots feature a sidebar on the left with navigation links: Clients, Produits (highlighted), Credits, and a user profile section for M.Admin with a LOGOUT button.

**Top Screenshot: Ajouter un nouveau Produit**

The main content area shows a form titled "Ajouter un nouveau Produit à votre carnet". The form includes the following fields and buttons:

- Buttons: Créer, Gérer
- Form Title: Ajouter un nouveau Produit à votre carnet
- Fields:
  - Nom
  - Prix
  - Commentaire
  - Choose file (No file chosen)
- Action Button: Ajouter

**Bottom Screenshot: Consultez vos Produits**

The main content area shows a table titled "Consultez vos Produits". The table has a search bar "Trouvez Produits" with a Search button. The table columns are Image, Nom, Commentaire, Prix, and Actions.

Image	Nom	Commentaire	Prix	Actions
	produit1 u	comment1	10.0	Editer Supprimer
	produit2	commen2	12.0	Editer Supprimer

#### 4. Gestion des crédits par clients:

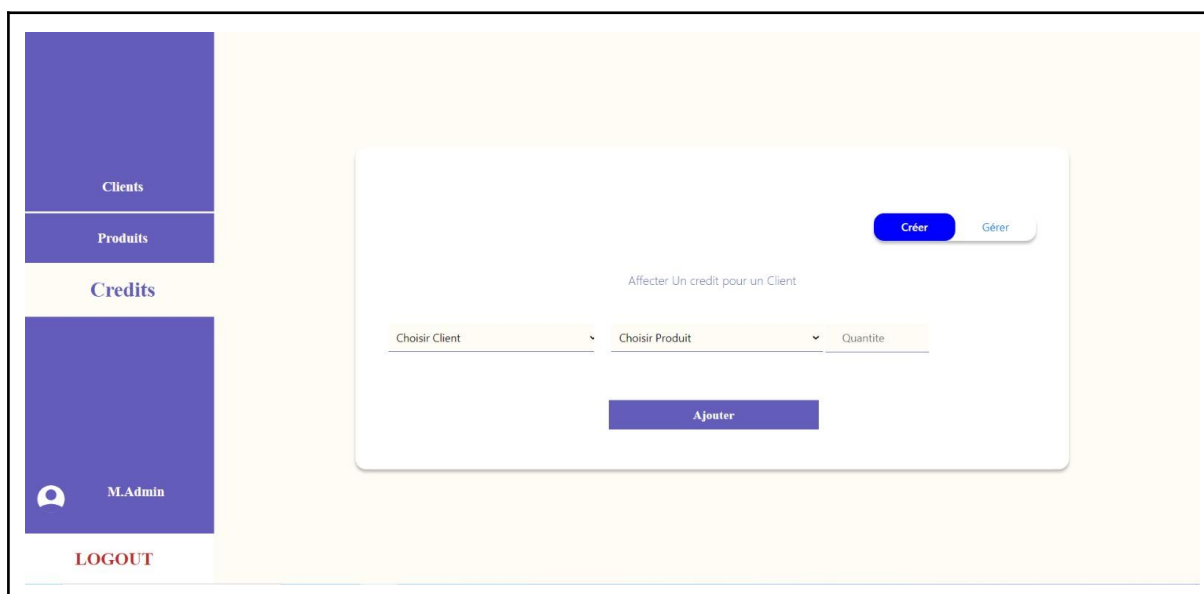
- **Back -End:**

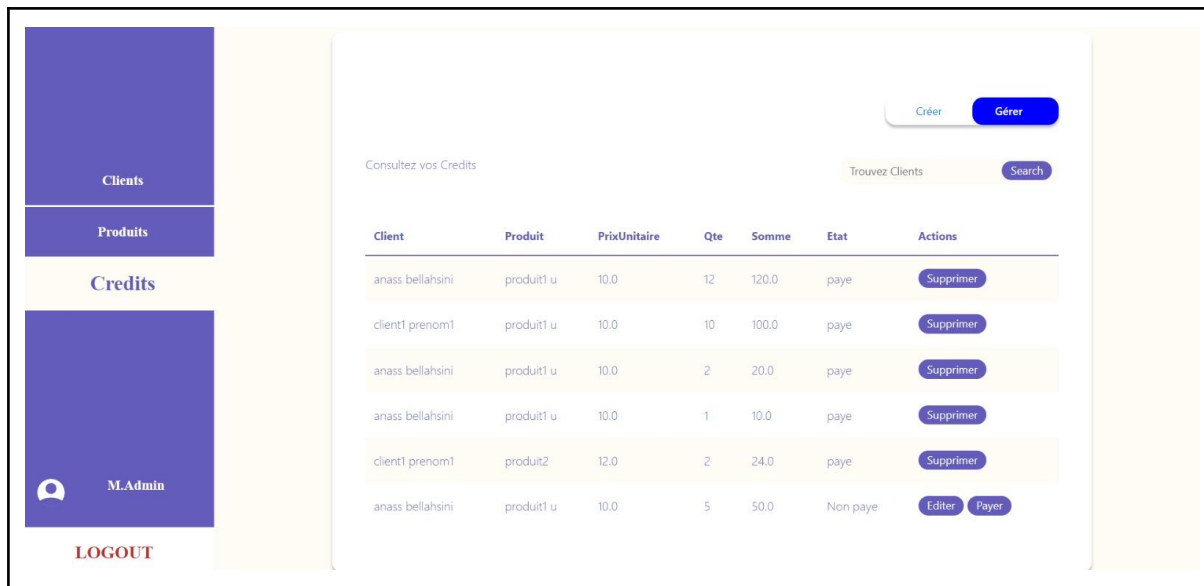
```
class Credit(models.Model):
    produit=models.ForeignKey(Produit, on_delete=models.CASCADE)
    client=models.ForeignKey(Client, on_delete=models.CASCADE)
    quantite=models.IntegerField()
    price=models.FloatField()
    total=models.FloatField()
    admin=models.ForeignKey(Admin, on_delete=models.CASCADE)
    etat=models.BooleanField(default=False)
```

- **Front-End: Pour notre dernier axe Crédit, on a deux pages aussi :**

**Nouveau Crédit :** C'est la page d'où on peut affecter des nouveaux crédits à des clients. Il suffit de choisir tout simplement le client voulu, la quantité et à quels produits, et cliquer sur ajouter pour enregistrer les informations du nouveau crédit.

**Gestion Crédits :** Notre deuxième page est la page de gestion des crédits : c'est la page d'où on affiche les crédits enregistrés (le nom du client, produit, prix, quantité, total et l'état du crédit c'est-à-dire payé ou non payé), de plus la possibilité d'effectuer des opérations éditer un crédit, payer un crédit ou bien le supprimer. Pour faciliter la gestion, on a réalisé un champ Chercher un client pour trouver le client et les informations sur leur crédit que l'admin cherche rapidement.





## Conclusion :

Le développement d'une application web E-épiciier présente de nombreux avantages pour les épiciers. Tout d'abord, la gestion des crédits clients devient plus facile et plus efficace. Les épiciers peuvent facilement suivre les paiements de leurs clients et accéder à des données en temps réel sur les soldes de leurs clients, ce qui peut aider à éviter les retards de paiement et à réduire les risques de créances douteuses.

En outre, l'utilisation de cette application web peut également améliorer l'efficacité opérationnelle des épiciers. Les tâches administratives telles que la facturation, la gestion des stocks et la tenue des registres peuvent être effectuées plus rapidement et avec moins d'erreurs grâce à l'automatisation offerte par l'application. Cela peut permettre aux épiciers de gagner du temps, d'économiser de l'argent et de se concentrer sur leur cœur de métier.

Enfin, la prise de décision financière est facilitée grâce à l'accès à des données en temps réel sur les ventes, les stocks et les crédits clients. Les épiciers peuvent ainsi prendre des décisions plus éclairées en matière de gestion financière, comme l'achat de stocks en fonction de la demande ou la décision d'accorder ou non du crédit à un client.

Dans l'ensemble, le développement d'une application web E-épiciier offre un avantage concurrentiel à ceux qui l'adoptent. Les épiciers qui utilisent cette solution numérique peuvent gagner du temps, améliorer leur efficacité opérationnelle et prendre des décisions financières plus éclairées, ce qui leur permet de se concentrer davantage sur leur activité principale et de mieux répondre aux besoins de leurs clients.