|  |  |  |
| --- | --- | --- |
|  |  |  |



**TECHNIQUES DE DÉVELOPPEMENT**

**INFORMATIQUE**

***(TDI*)**

*Niveau Technicien spécialisé*

|  |
| --- |
| **Rapport de projet de fin de formation**  **Conception et réalisation d’une application de gestion de stock** |

### Réalisé par :

### ▪ Ayoub elmarhraoui

### ▪ IDRISS AIT SI EL ARABI

**ISTA NTIC SYBA [2020/2021]**

# Glossaire

**ISTA** : Institut Spécialisé de Technologie Appliquée

Tri par ordre alphabétique

*(Texte Times new Roman 12 Interligne 1.5)*

# List des figures

[Figure 1 : modèle de conception de données MCD 12](#_Toc73283583)

[Figure 2 : Page authentification utilisateur 14](#_Toc73283584)

[Figure 3 : page accueil 15](#_Toc73283585)

[Figure 5 : Page ajoute Maintenance 16](#_Toc73283587)

[Figure 6 : Page Rechercher des Maintenances 17](#_Toc73283588)

[Figure 7 : Page ajoute de produit 17](#_Toc73283589)

[Figure 8 : page recherche des produits 18](#_Toc73283590)

[Figure 9 : page modifier de produit 18](#_Toc73283591)

[Figure 10 : Page ajoute de vente 19](#_Toc73283592)

[Figure 11 : Page recherche des ventes 19](#_Toc73283593)

[Figure 12 : Page modification de vente 20](#_Toc73283594)

[Figure 13 : page gestion des Fournisseurs 20](#_Toc73283595)

[Figure 14 : page modification du fournisseur 21](#_Toc73283596)

[Figure 15 : page gestion des utilisateurs 21](#_Toc73283597)

[Figure 16 : page modification des utilisateurs 22](#_Toc73283598)

[Figure 17 : page gestion des catégories 22](#_Toc73283599)

[Figure 18 : Page modification du catégorie 23](#_Toc73283600)

[Figure 19 : page gestion des commandes 23](#_Toc73283601)

[Figure 20 : page modification du commande 24](#_Toc73283602)

[Figure 21 : page gestion des produits retournes 24](#_Toc73283603)

[Figure 22 : page modification du produit retourne 25](#_Toc73283604)

[Figure 23 : forme de connexion à l'application de bureau 26](#_Toc73283605)

[Figure 24 : forme d’accueil à l'application de bureau 27](#_Toc73283606)

[Figure 25 : forme gestion de vente à l'application de bureau 27](#_Toc73283607)

[Figure 26 : forme Rechercher des ventes à l'application de bureau 28](#_Toc73283608)

[Figure 27 : forme gestion de maintenance à l'application de bureau 28](#_Toc73283609)

[Figure 28 : forme recherché des maintenances à l'application de bureau 29](#_Toc73283610)

[Figure 29 : forme gestion de produit à l'application de bureau 29](#_Toc73283611)

[Figure 30 : forme recherché des produits à l'application de bureau 30](#_Toc73283612)

[Figure 31 : forme gestion de produit retourné à l'application de bureau 30](#_Toc73283613)

[Figure 32 : forme gestion des fournisseurs 31](#_Toc73283614)

[Figure 33 : forme gestion des utilisateurs 31](#_Toc73283615)

Figure 34 : Logo de Langage Csharp………………………………………………………………………………………………………………………………………….33

Figure 35 : Logo de Langage Type Script……………………………………………………………………………………………………………………………………33

Figure 36 : Logo ASP.NET Core………………………………………………………………………………………………………………………………………………….34

Figure 37: Logo Angular……………………………………………………………………………………………………………………………………………………………34

Figure 38 : Logo HTML………………………………………………………………………………………………………………………………………………………………34

Figure 39 : Logo CSS………………………………………………………………………………………………………………………………………………………………….34

# Liste des tableaux

[Tableau 1 : les opérations de les rôles de les d'utilisateurs 10](#_Toc73222019)

[Tableau 2 : modèle dictionnaire de données 10](#_Toc73222020)

# Table des matières

*(Texte Times new Roman 12 Interligne 1.5)*

Introduction …………………………………………

**Chapitre 1. Titre du chapitre**

**Chapitre 2. Titre du chapitre**

2.2 Titre du paragraphe

2.2.1 Sous paragraphe

2.2.2

2.3

2.3.2

**…………………………………….**

**Conclusion**

**Références**

**Annexes**

# Introduction

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science étudiant les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de l'entreprise et d'autres établissements.

Avant l'invention de l'ordinateur, nous enregistrions toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières. Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention à permis d'informatiser les systèmes de gestion de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui. les Magasins que vente et réparation des produit font partie des entreprises que l'informatique pourra beaucoup aider En effet, Cela nécessite la saisie et l'analyse des données Gestion rationnelle, efficace et rapide, or et jusqu'à ce jour, la gestion manuelle est encore la plus dominante. Nous remarquons ainsi la mauvaise organisation du travail dans la Magasins lors de la recherche d'une information ou lors de la création des statistiques, l'information n'est pas toujours précise et disponible d'où la nécessité d'introduire l'informatique dans les Magasins.

Vu cet état de fait, notre projet de fin de cycle a pour objectif de concevoir et mettre en œuvre une application web et desktop interactive, fiable, conviviale et facile à intégrer dans l'environnement de travail des Magasins.

Notre mémoire est organisé en trois chapitres principaux :

Le premier chapitre porte sur Contexte général du projet. La spécification des besoins nous permettra de délimiter notre problématique et les objectifs visés.

Le deuxième chapitre porte sur la conception et Analyse, il regroupe toutes les étapes de notre processus de développement en utilisant le langage de modélisation MERISE.

Le troisième et dernier chapitre est consacré à la réalisation où nous allons définir tout les outils qui nous ont permis de concevoir notre application web et desktop, quelques interfaces y seront présentées. Notre travail s’achèvera par une conclusion générale.

**CHAPITRE 1 :**

**CONTEXTE GENERAL DU PROJET**

## Contexte général du projet :

Comme dans le cas de la plupart des entreprises particulièrement qui vendent et réparent du matériel électronique comme les téléphones et les ordinateurs et leurs composants, elles doivent consigner les données telles que les produits vendus, matériels réparé, les produits en stocks et d’autres opérations. La méthode traditionnelle utilisée est d’enregistrer les données généralement dans des carnets ou des fichiers Excel, ce qui conduit à la combinaison des difficultés dans la gestion de ces données.  
 Dans la recherche des solutions pratiques qui facilitent notre type de processus, nous avons proposé une solution technique à travers notre projet de fin d’étude, qui est deux modèles d’application informatique qui rendent le même processus de gestion des données facilement en comparaison à la méthode traditionnelle: une application web et une application de bureau.

### 1.1 Présentation de l’existant et problématique :

En raison de l’importance des données, il ressort clairement d’après l’étude que la méthode traditionnelle (carnets, fichiers Excel, …) de gestion des ventes, des réparations et de stock ainsi que les informations personnelle des fournisseurs est extrêmement difficile en raison de grande quantité des données qu'il faut registre.

Parmi les problématiques rencontrées :

* Personnaliser un carnet ou un fichier pour chaque catégorie particulier, et à mesure que les catégories augmentent, cela devient plus difficile.
* Perte de temps.
* Difficulté de modifier ou supprimer une donnée particulière.
* Difficulté de chercher, de filtrer et classer les données de façon différentes.
* Difficulté de faire des statistique tel que les produit les plus vendus, ou le jour, mois et l'année dans les quelle les ventes sont augmentées ou l'inverse.
* Niveau de sécurité des données faible.
* Difficulté de faire des rapports de manière automatique.
* Découvrir l'état de stock d'une manière difficile.

Pour les figures :

Les figures doivent être nettes avec un titre contenant le numéro du chapitre (en gras et centré), par exemple.

**Figure x.2. Titre de la figure**

(X : numéro du chapitre).

**Tableau x.1. Titre du Tableau**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 1.2 Solution proposée et objectifs du projet :

On propose notre projet de fin d'étude qui deux applications (web et bureau) comme une solution informatique adapter au genre de problématiques cités dans la partie précédente et qui met à la disposition du client un outil qui satisfaire ses besoins dont les objectifs et les avantages sont:

* Un contrôle centralisé de l’entreprise.
* Gagner le temps.
* L’optimisation des processus de gestion.
* La flexibilité de navigation entre les composantes de projet.
* Une possibilité de recherche et de filtrage de données avec des manières différentes.
* L’ajout, la modification et suppression des données facilement.
* Assurer la sécurité de ses informations.
* Accès aux données limité selon le rôle de l’utilisateur de l’application.
* Suivre l’avancement des ventes et des réparations des produits par mois et par l’année en utilisant des graphes.
* Suivre l’état de stock facilement.
* Sauvegarde uniforme des données et évite sa redondance.
* Réduire la consommation des papiers.
* La cohérence et homogénéité des informations.
* Une aide à la productivité.
* Optimiser la traçabilité des produits les plus vendus.

**CHAPITRE 2 :**

**ANALYSE ET CONCEPTION DU PROJET**

## 2. Analyse et conception du projet

### 2.1 Les besoins fonctionnels :

**Gestion des ventes et réparations**:

- Créer, modifier et supprimer des ventes et réparations.

- Recherche des ventes et réparations par jour, mois, jour, mois ou entre deux dates.

**Gestion de stock des produits :**

- Créer, modifier et supprimer des produits.

- Créer, modifier et supprimer des catégories.

- Recherche des produits par catégorie.

**Gestion des fournisseurs :**

- Créer, modifier et supprimer des fournisseurs.

- Créer, modifier et supprimer des commandes.

- Créer, modifier et supprimer de détails commande.

**Gestion d’utilisateur :**

- Créer, modifier et supprimer des utilisateurs.

**Gestion des produits retournés aux fournisseurs :**

- Créer, modifier et supprimer des produits retournés.

**Statistique des ventes et réparations :**

- le montant total de tous mois par cette année.

- Le montant total de tous les jours de cette mois.

**Authentification :**

- Login par nom d'utilisateur et mot de passe.

|  |  |  |
| --- | --- | --- |
| administrateur | Vendeur | gérer le stock |
| Gestion des ventes et réparations  Gestion de stock des produits  Gestion des fournisseurs  Gestion d’utilisateur  Gestion des produits retournés aux fournisseurs | Gestion des ventes et réparations  Gestion de stock des produits | Gestion de stock des produits  Gestion des fournisseurs  Gestion des produits retournés aux fournisseurs |

Tableau 1 : les opérations de les rôles de les d'utilisateurs

### 2.2 Choix de la méthode de conception :

Merise est une méthode d'analyse, de conception et de gestion de projet complètement intégrée, ce qui en constitue le principal atout. Elle a fourni un cadre méthodologique et un langage commun et rigoureux à une génération d'informaticiens français.

Issue de l'analyse systémique, la méthode Merise est née dans les années 1970, à la demande du ministère de l'industrie, et a surtout été utilisée en France, par les SSII de ses membres fondateurs (Sema Metra, ainsi que par la CGI Informatique) et principalement pour les projets d'envergure, notamment des grandes administrations publiques ou privées.

La conception du système d'information se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance entre données et traitements afin de vérifier que toutes les données nécessaires aux traitements sont présentes et qu'il n'y a pas de données superflues.

### 2.3 Modèle 1 :

|  |  |  |
| --- | --- | --- |
| Propriétés | Description | Type |
| NumFournisseur | Numéro du Fournisseur | Entier |
| TelFournisseur | Téléphone du Fournisseur | Chaine de caractères |
| AdresseFournisseur | Adresse du Fournisseur | Chaine de caractères |
| NomFournisseur | Nom du Fournisseur | Chaine de caractères |
| NumUtilisateur | Numéro d’utilisateur | Entier |
| Nom | Nom d’utilisateur | Chaine de caractères |
| Prénom | Prénom d’utilisateur | Chaine de caractères |
| Tel | Téléphone d’utilisateur | Chaine de caractères |
| Adresse | Adresse d’utilisateur | Chaine de caractères |
| Rôle | Rôle d’utilisateur | Chaine de caractères |
| Login | Login d’utilisateur | Chaine de caractères |
| PassWord | Mot de passe d’utilisateur | Chaine de caractères |
| NumCategorie | Numéro de la catégorie | Entier |
| NomCategorie | Nom de la catégorie | Chaine de caractères |
| NumReparation | Numéro de la réparation | Entier |
| TitreReparation | Titre de la réparation | Chaine de caractères |
| PrixReparation | Prix de la réparation | Réel |
| DateReparation | Date de la réparation | Date |
| Commentaire | Commentaire sur la réparation | Chaine de caractères |
| NumProduit | Numéro de Produit | Entier |
| NomProduit | Nom de Produit | Chaine de caractères |
| QuantiteStockee | Quantité stockée de Produit | Entier |
| Prix | Prix de Produit | Réel |
| NumCommande | Numéro de Commande | Entier |
| DateCommande | Date De la Commande | Date |
| QuantiteCommandee | Quantité de produit commandé | Entier |
| DateVente | Date du Vente | Date |
| PrixVente | Prix du Vente | réel |
| QuantiteVendu | Quantité Vendu | Entier |
| PrixUnitaireDeGros | Prix Unitaire de Produit De Gros | Réel |
| NumProduitRetourne | Numéro de Produit retourne | Entier |
| DateRetoure | Date de Produit retourne | Date |
| CommentaireProduitRetourne | Commentaire de Produit retourne | Chaine de caractères |
| QuantitéProduitRetourne | Quantité de Produit retourne | Entier |

Tableau 2 : modèle dictionnaire de données

### 2.4 Modèle 2 :

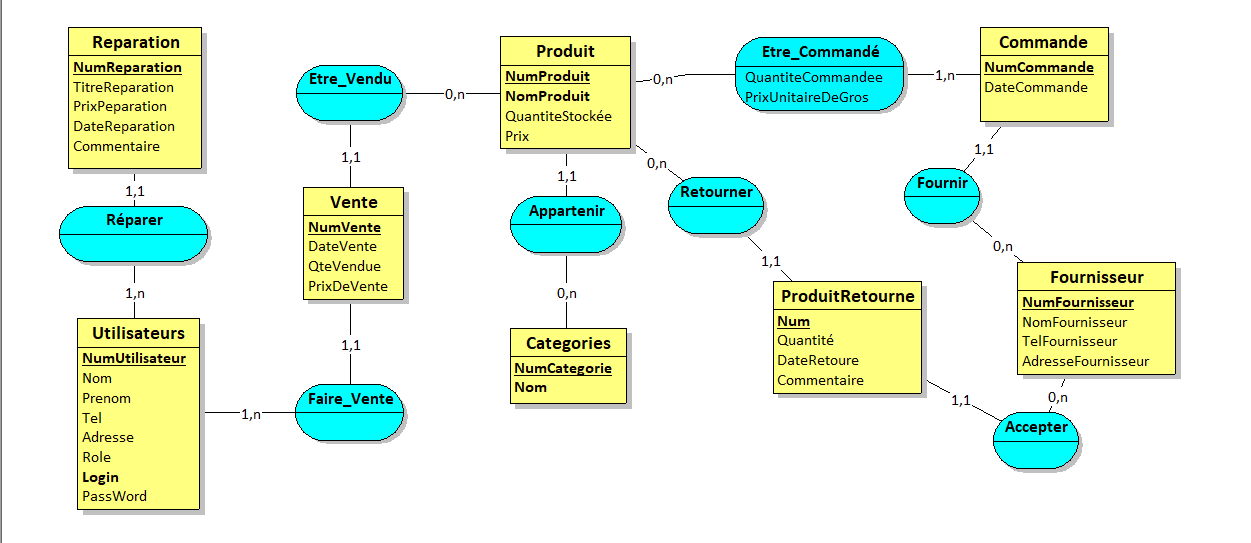


Figure 1 : modèle de conception de données MCD

### 2.5 Modèle 3 :

### Fournisseur = (NumFournisseur, NomFournisseur, TelFournisseur, AdresseFournisseur);

* **Utilisateurs** = (NumUtilisateur, Nom, Prenom, Tel, Adresse, Role, Login, PassWord);
* **Categories** = (NumCategorie, Nom);
* **Reparation** = (NumReparation, TitreReparation, PrixPeparation, DateReparation, Commentaire, #NumUtilisateur);
* **Produit** = (NumProduit, NomProduit, QuantiteStockée, Prix, #NumCategorie);
* **Commande** = (NumCommande, DateCommande, #NumFournisseur);
* **Vendre** = (#NumProduit, #NumUtilisateur, DateVente, PrixVente, QuantitéVendu);
* **EtreCommandé** = (#NumCommande , #NumProduit, QuantiteCommandee, PrixUnitaireDeGros);
* **ProduitRetourne** = (NumProduitRetourne, QuantitéProduitRetourne, DateRetoure, CommentaireProduitRetourne*, #NumFournisseur, #NumProduit*);

|  |
| --- |
| **CHAPITRE 3 :**  **REALISATION ET MISE EN ŒUVRE**  *(Times new Roman 24 gras centré)* |

## 3. Réalisation et mise en œuvre : *(Times new Roman 16 gras)*

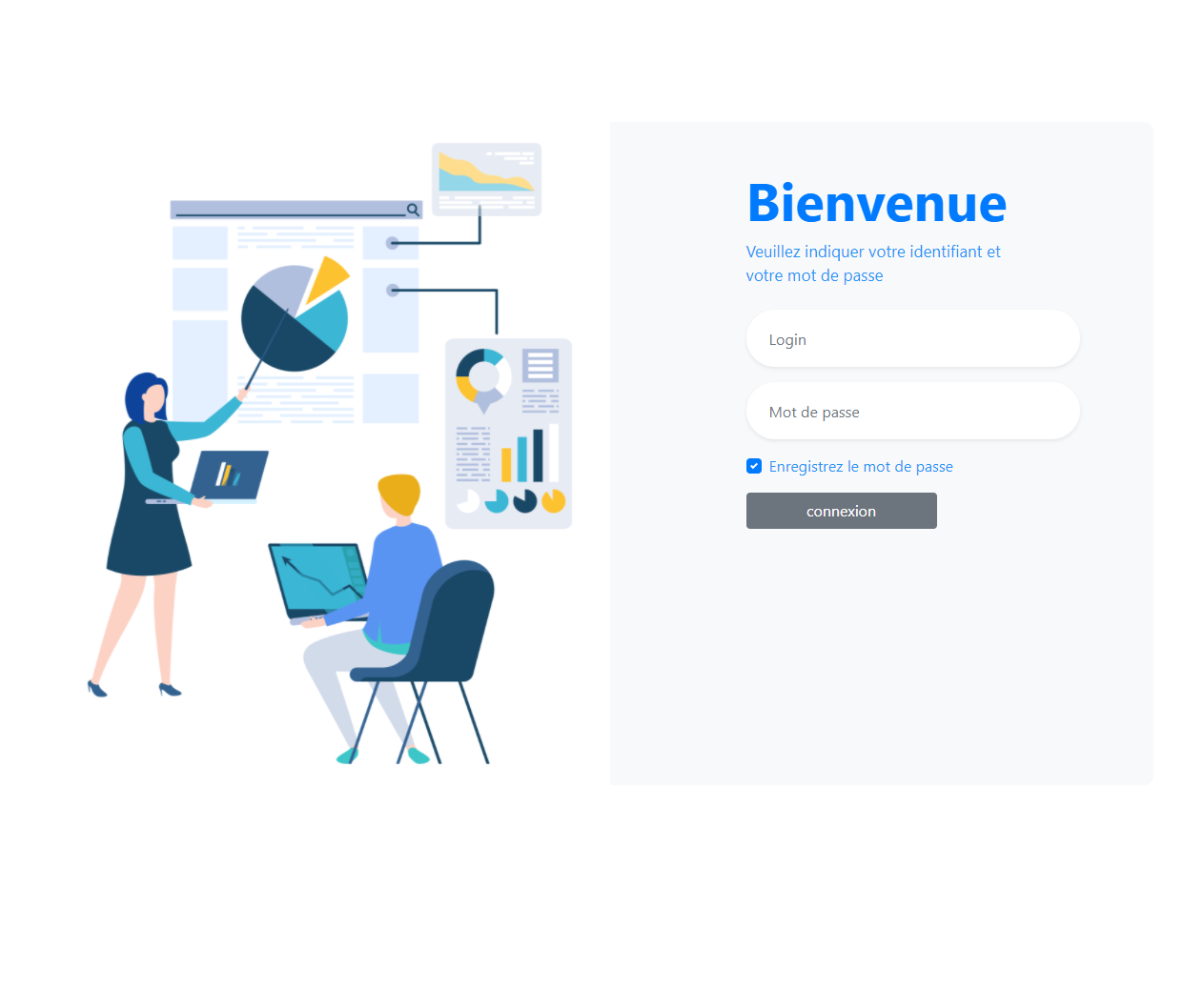


Figure 2 : Page authentification utilisateur

**

Figure 3 : page accueil

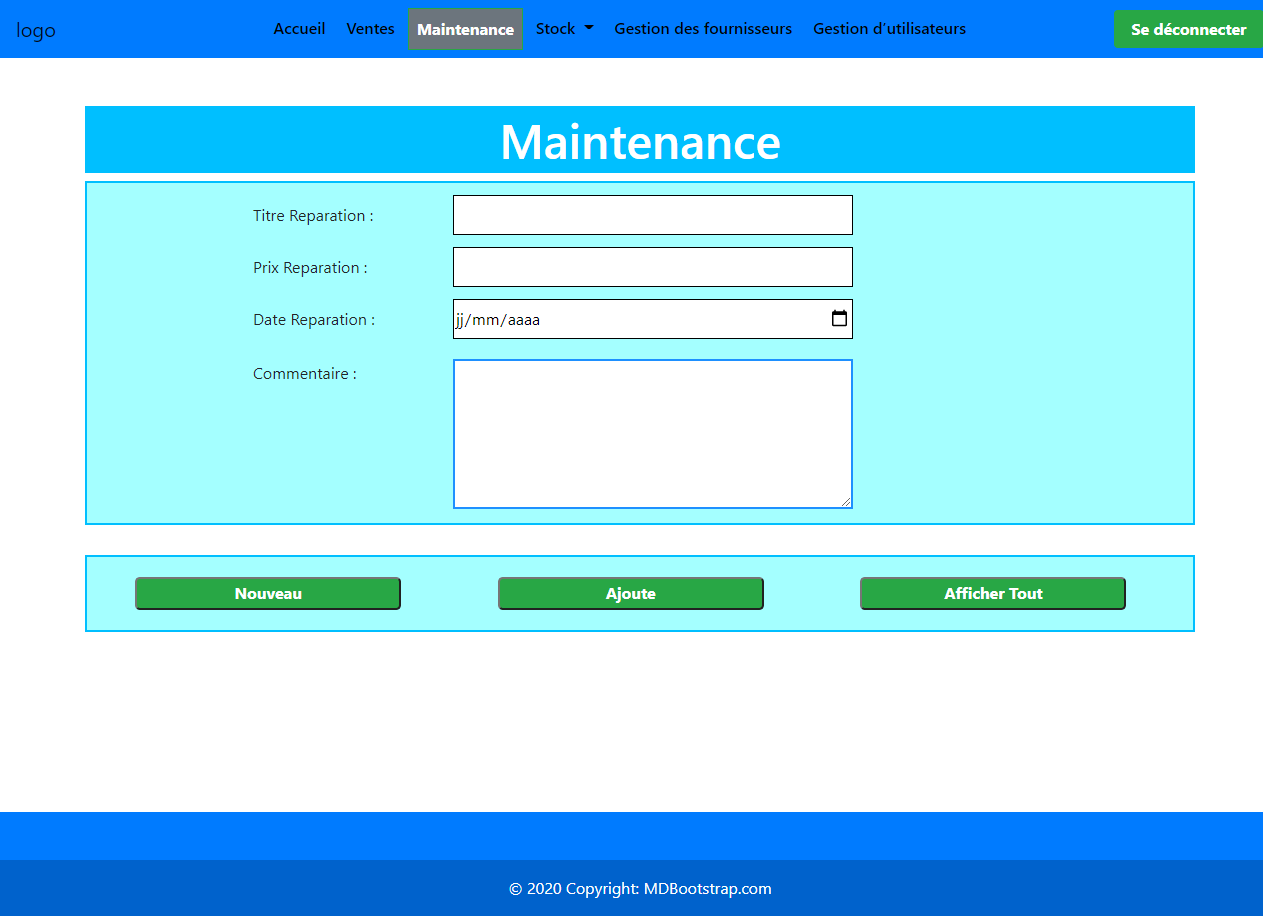
****

Figure 5 : Page ajoute Maintenance

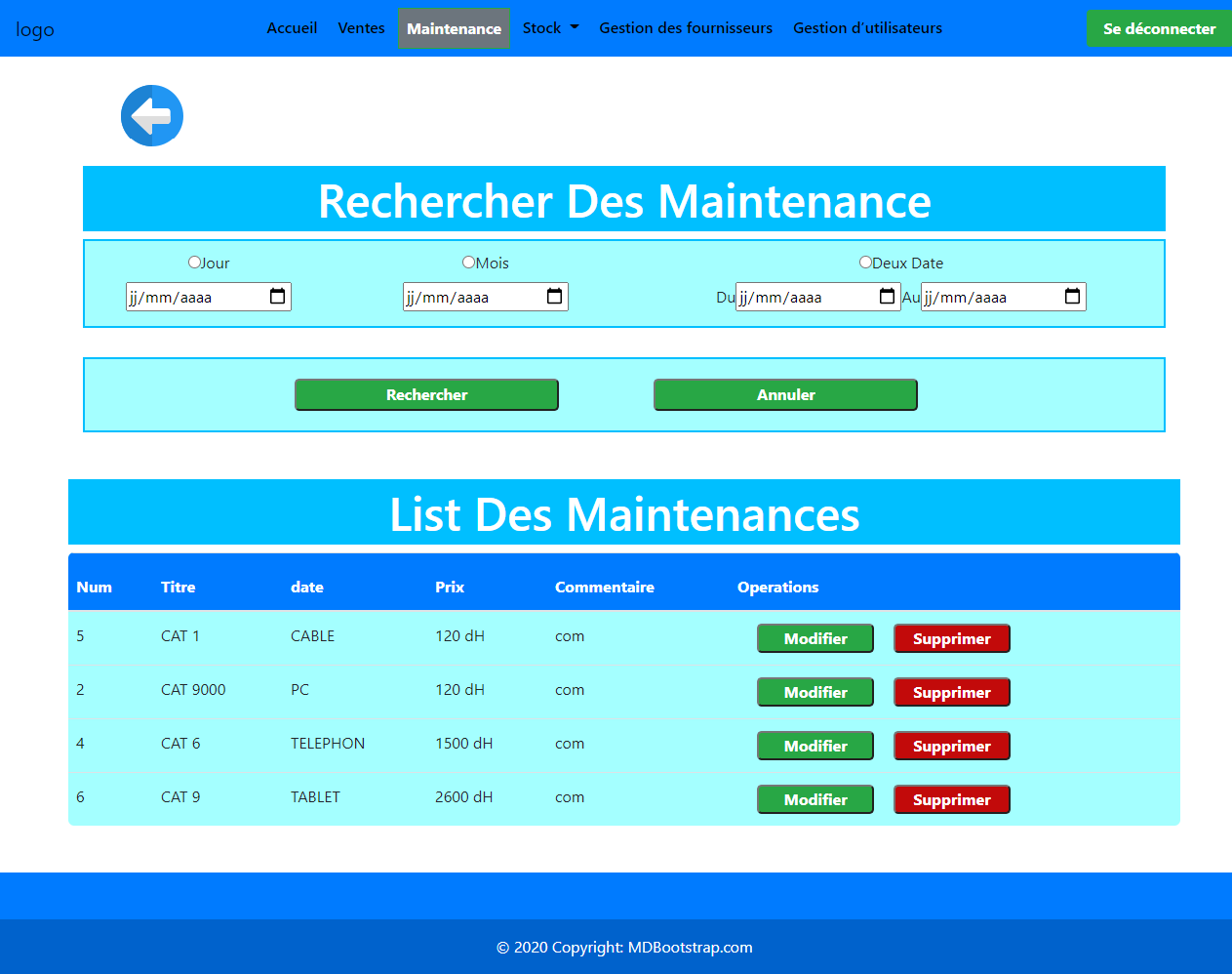


Figure 6 : Page Rechercher des Maintenances

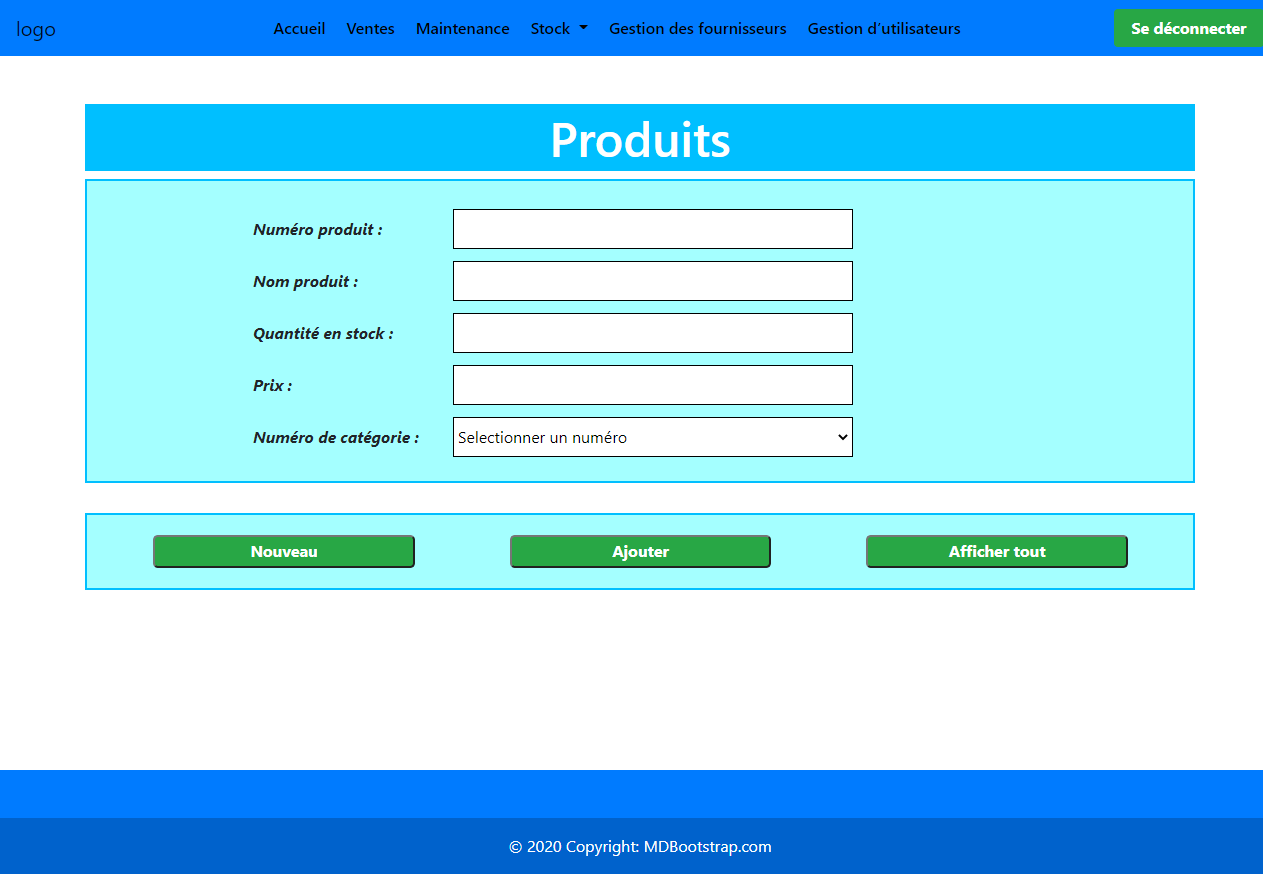


Figure 7 : Page ajoute de produit

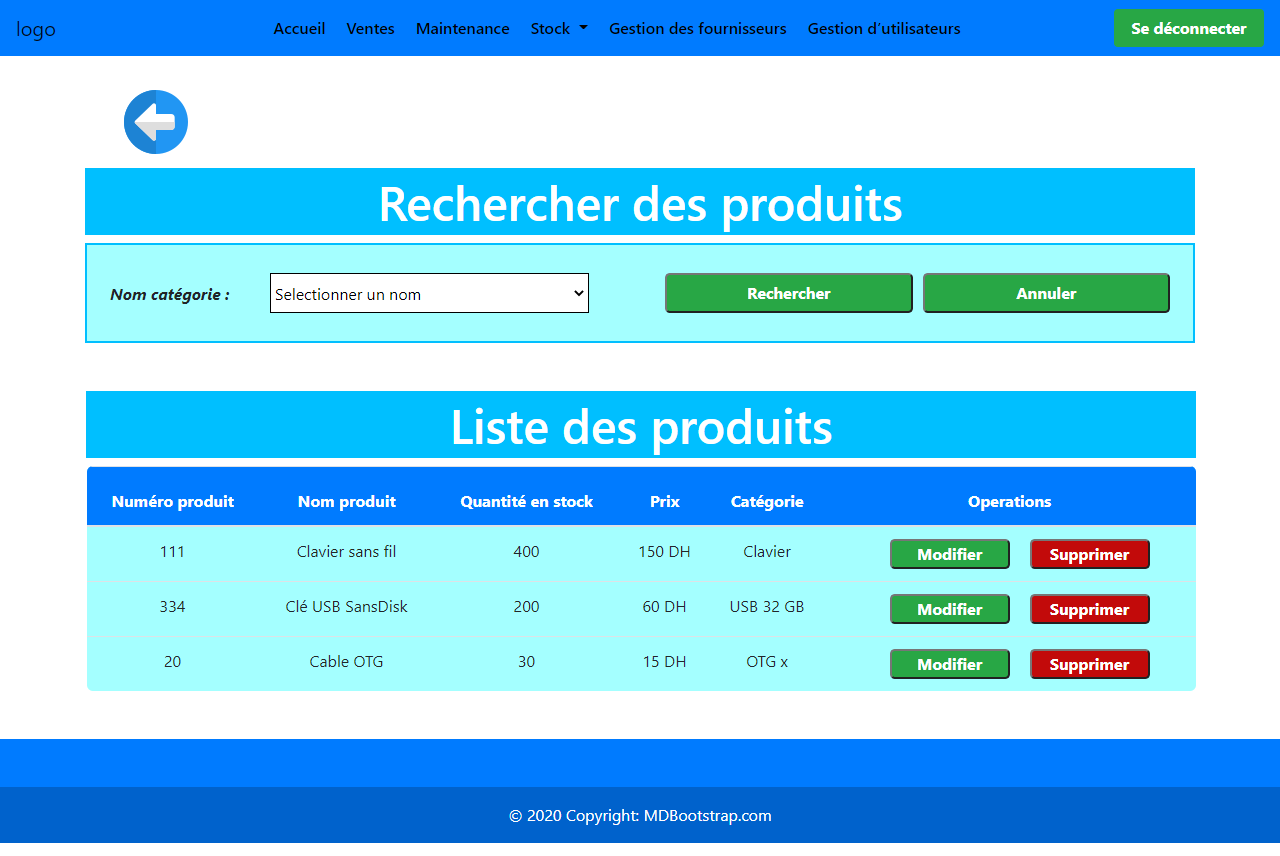


Figure 8 : page recherche des produits

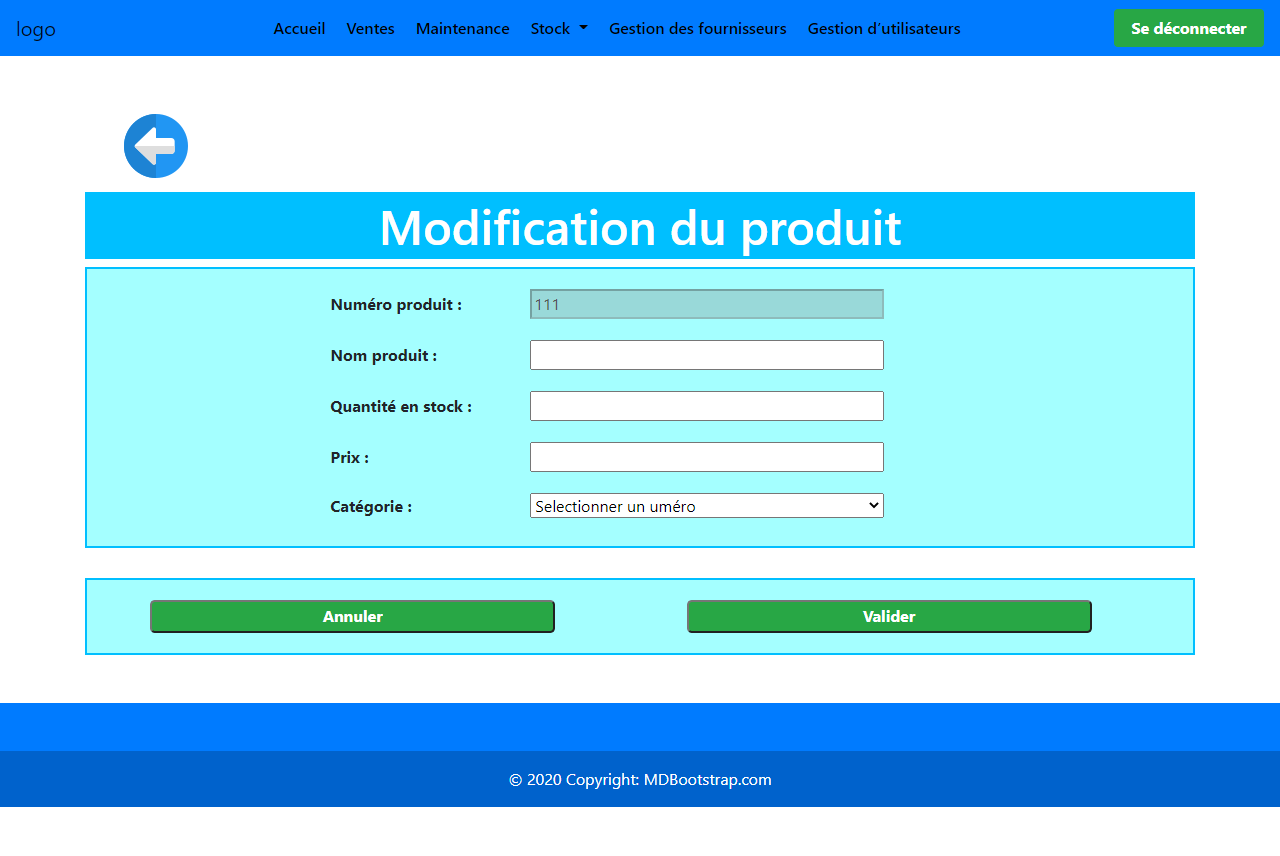


Figure 9 : page modifier de produit

### C:\Users\ayoub\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Gestion Ventes.png

Figure 10 : Page ajoute de vente

### screencapture-localhost-4200-Gestion-Ventes-Rechercher-2021-05-29-21_00_06

Figure 11 : Page recherche des ventes



Figure 12 : Page modification de vente

### Gestion des Fournisseurs

Figure 13 : page gestion des Fournisseurs



Figure 14 : page modification du fournisseur

### Gestion Utilisateurs

Figure 15 : page gestion des utilisateurs

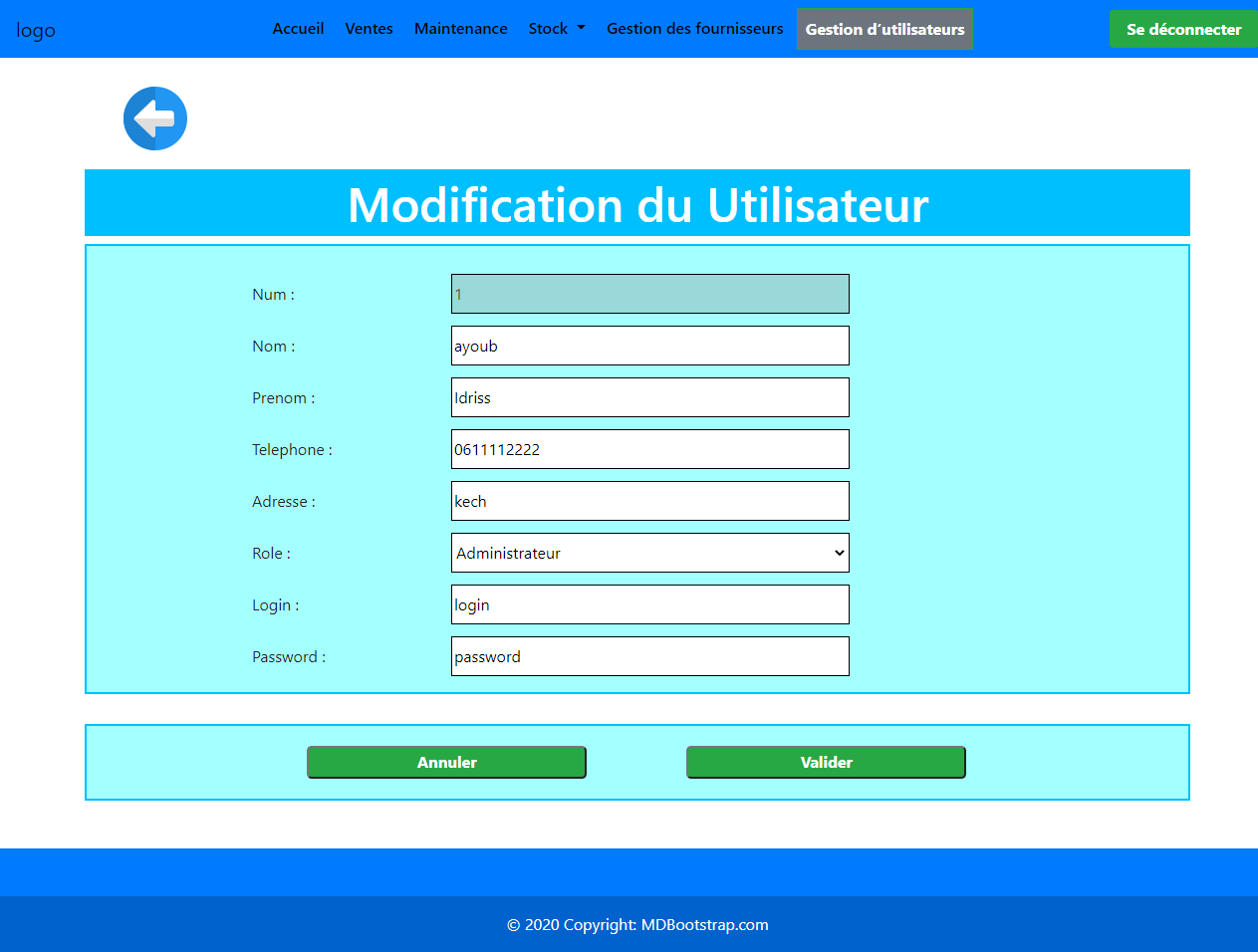


Figure 16 : page modification des utilisateurs

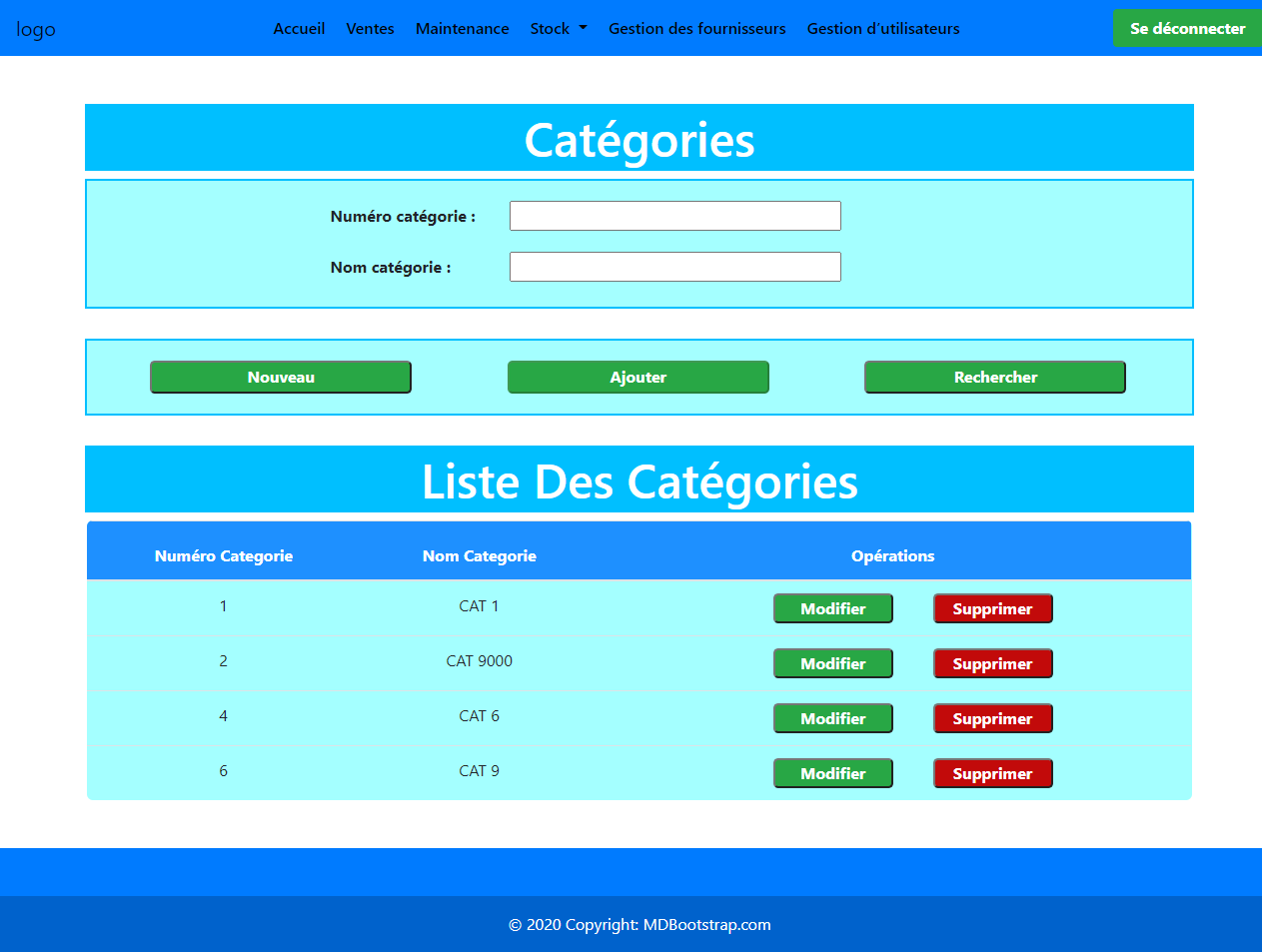


Figure 17 : page gestion des catégories

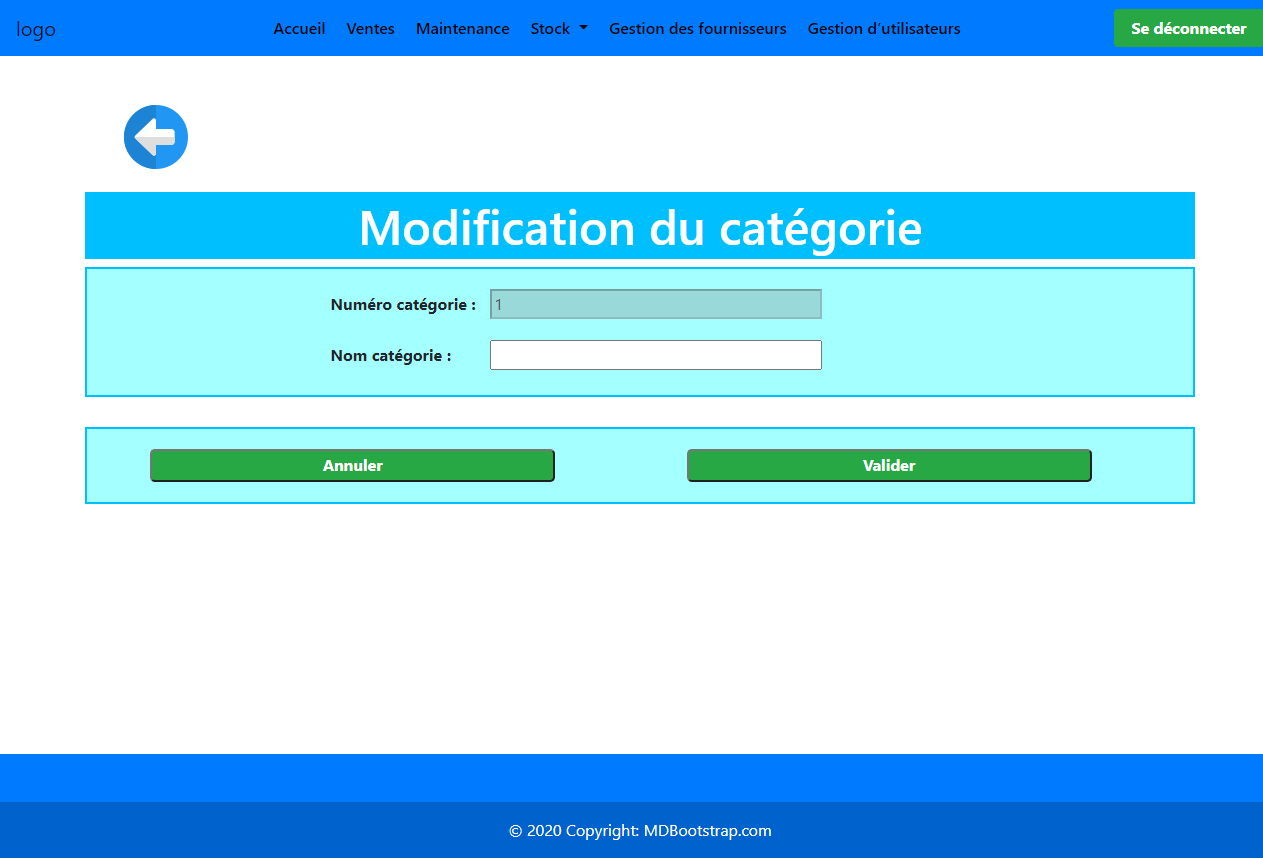


Figure 18 : Page modification du catégorie

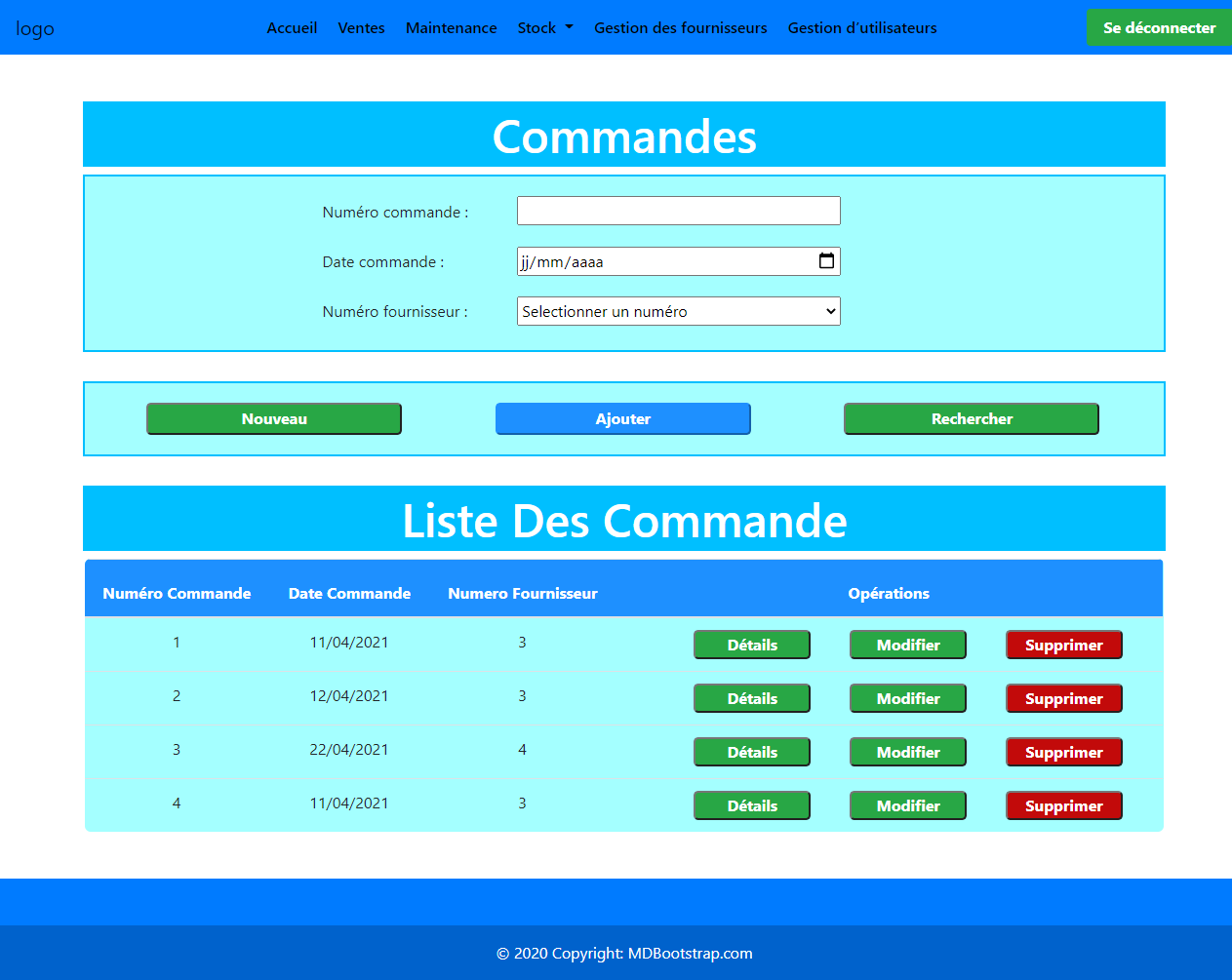


Figure 19 : page gestion des commandes

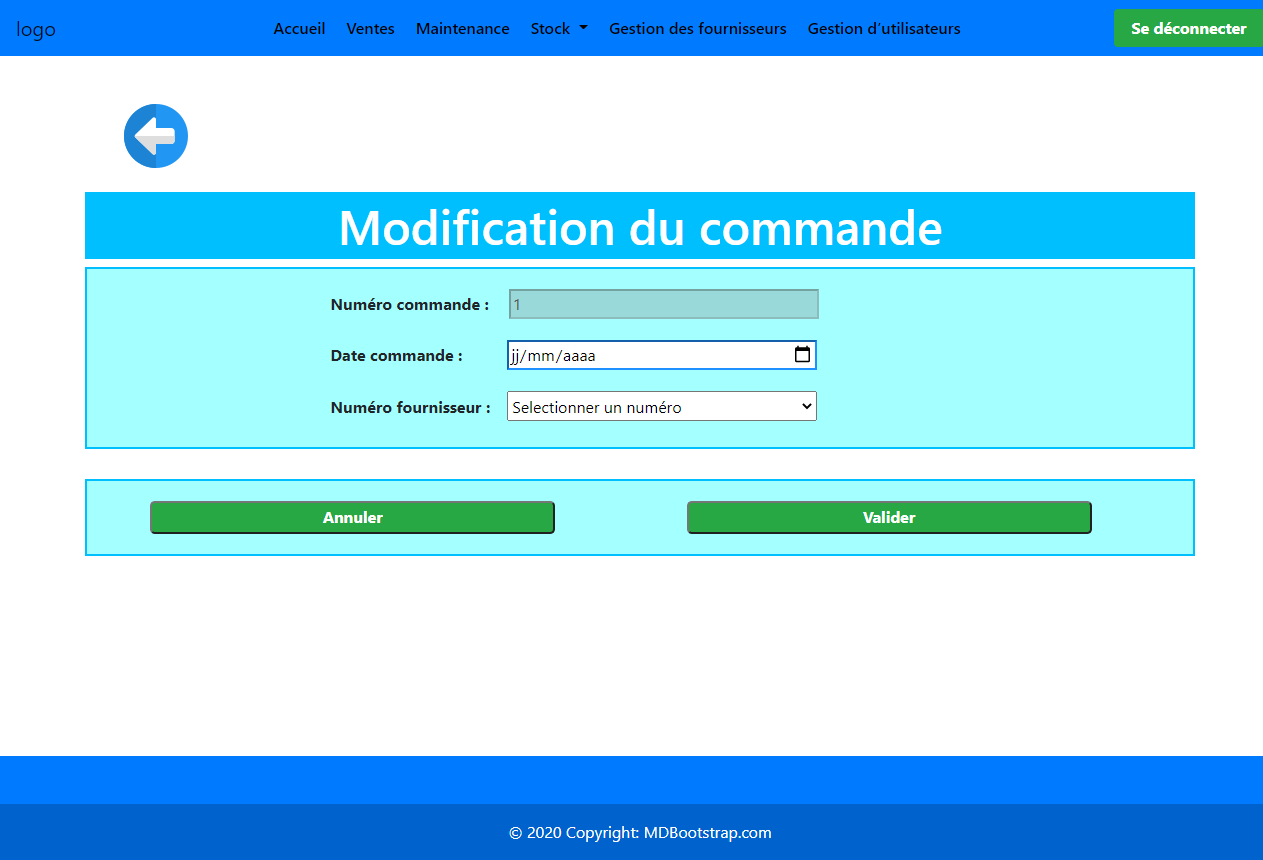


Figure 20 : page modification du commande

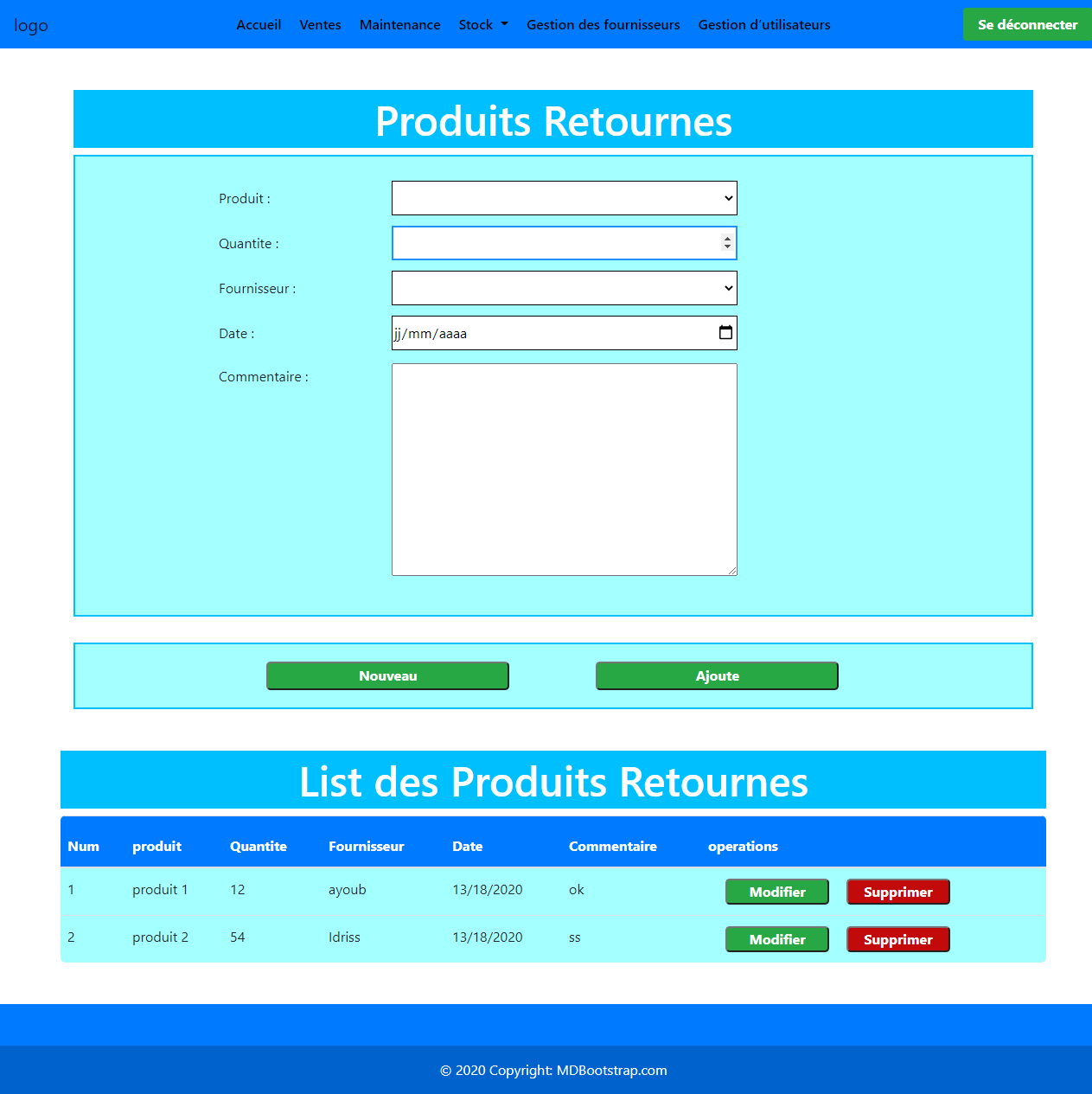


Figure 21 : page gestion des produits retournes

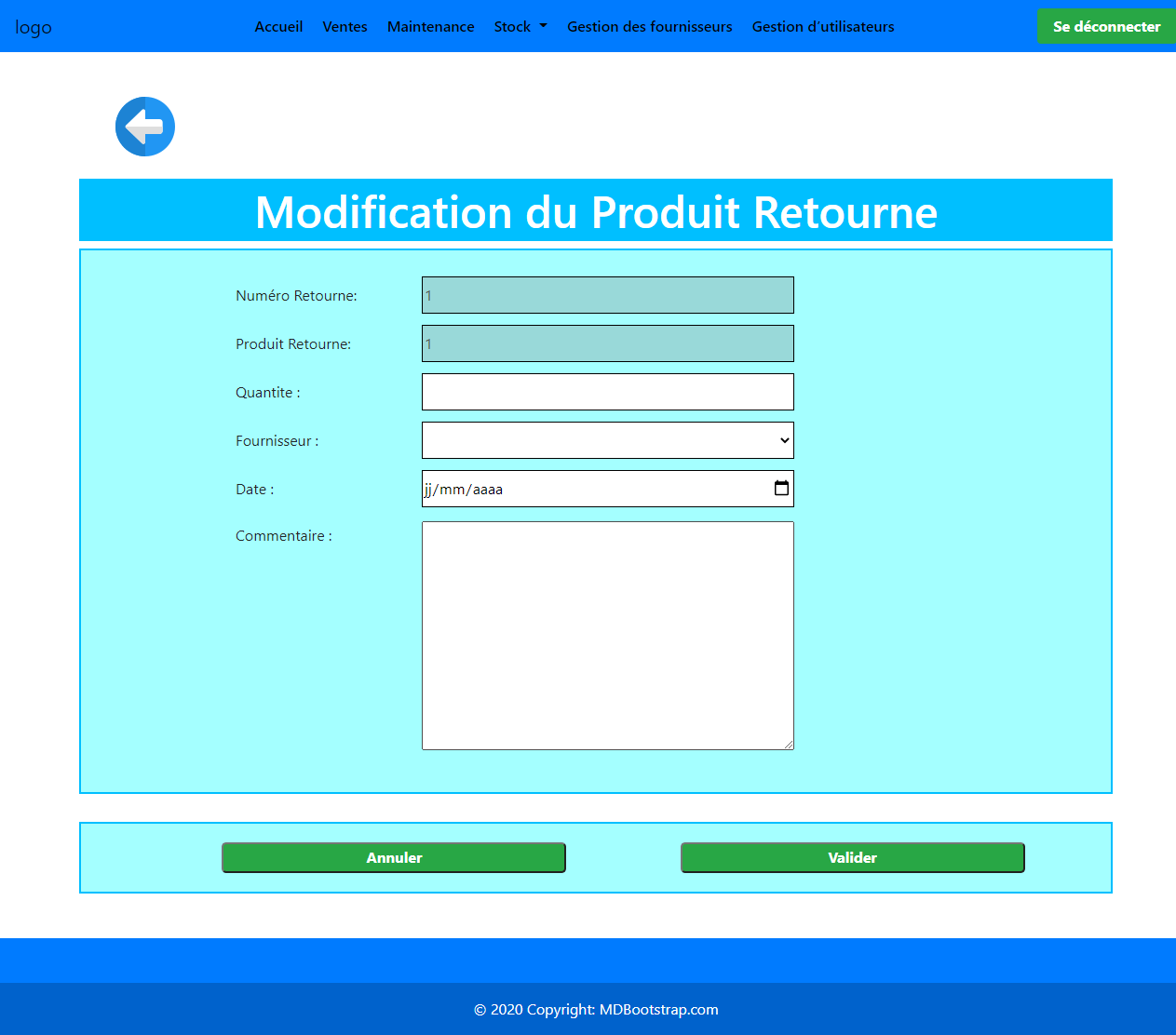


Figure 22 : page modification du produit retourne

# Application desktop :

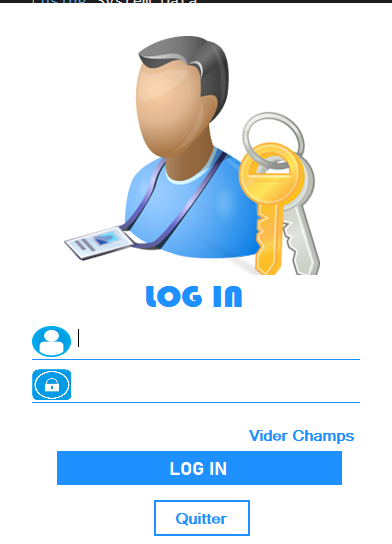


Figure 23 : forme de connexion à l'application de bureau

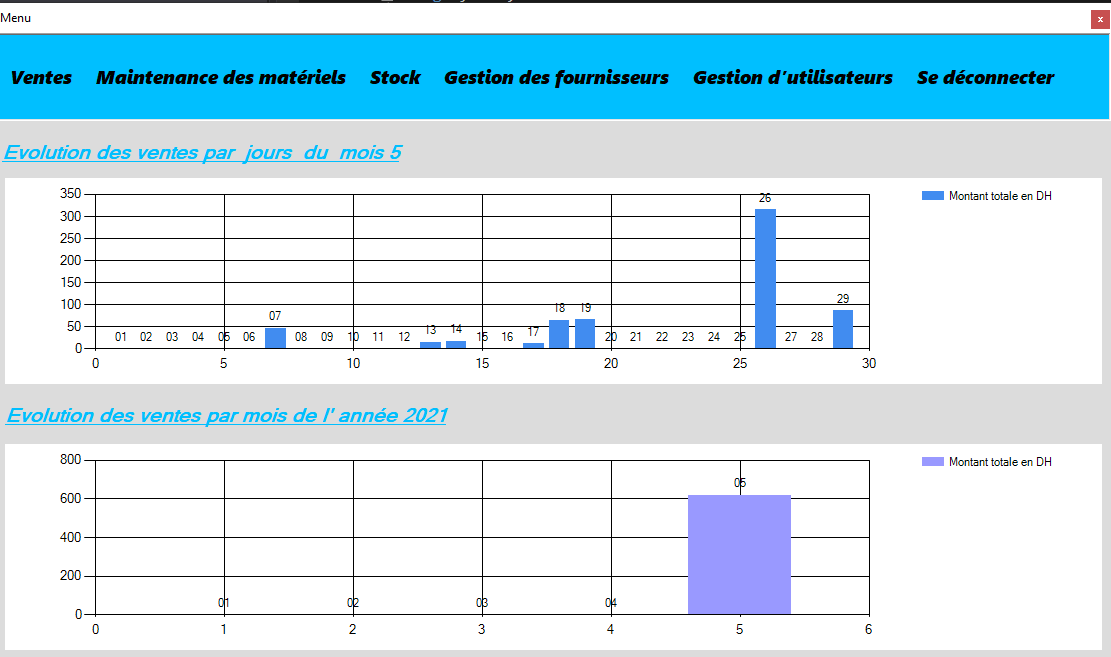


Figure 24 : forme d’accueil à l'application de bureau



Figure 25 : forme gestion de vente à l'application de bureau



Figure 26 : forme Rechercher des ventes à l'application de bureau

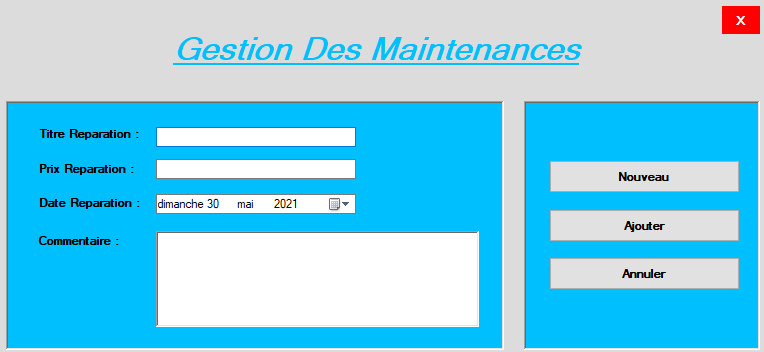


Figure 27 : forme gestion de maintenance à l'application de bureau

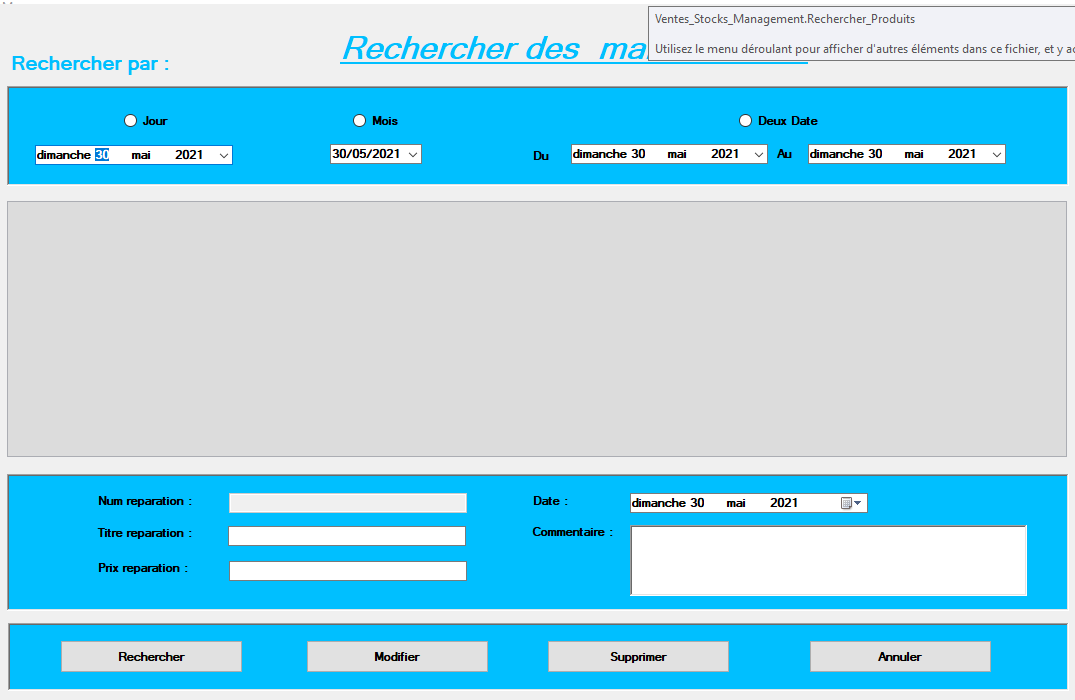


Figure 28 : forme recherché des maintenances à l'application de bureau



Figure 29 : forme gestion de produit à l'application de bureau

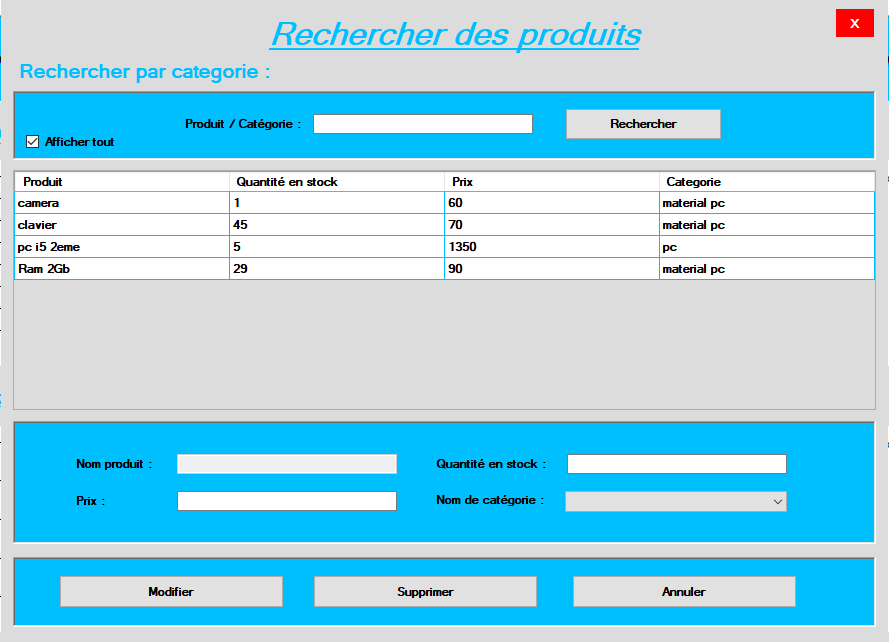


Figure 30 : forme recherché des produits à l'application de bureau



Figure 31 : forme gestion de produit retourné à l'application de bureau

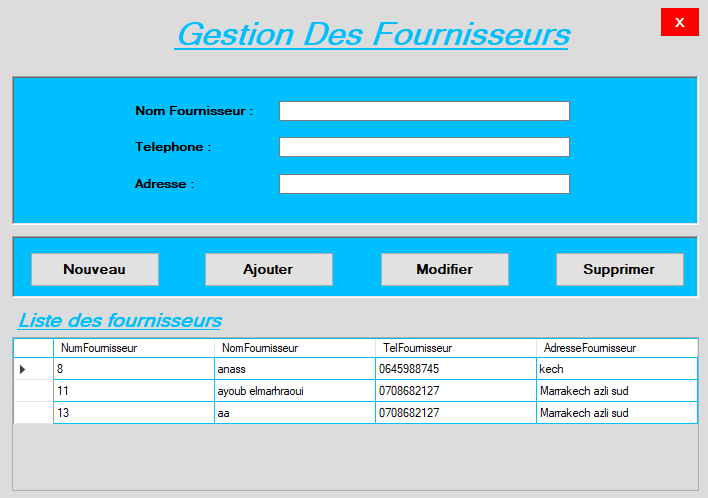


Figure 32 : forme gestion des fournisseurs

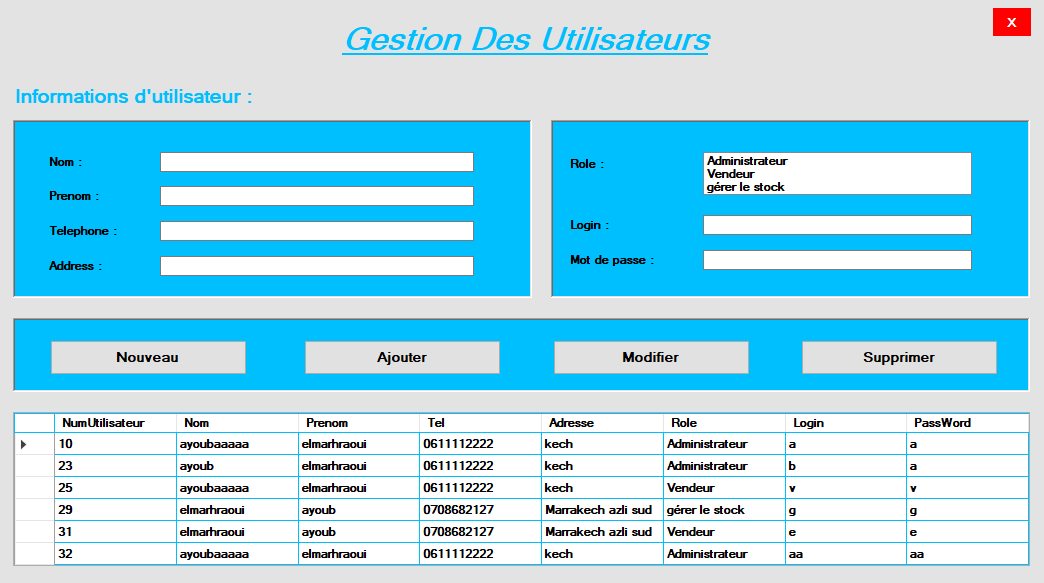


Figure 33 : forme gestion des utilisateurs

### 3.1 Les outils de travail :

**3.1.1 Langages**

**a. Csharp (C#) :**

Nous avons utilisé C# pour développer le coté back-end de l’application.

C# est un langage de programmation orientée objet, fortement typé, dérivé de C et de C++, ressemblant au langage Java. Il est utilisé pour développer des applications web, ainsi que des applications de bureau, des services web, des commandes, des bibliothèques de classes.



Figure 33 : Logo de Langage Csharp

**b. TypeScript :**

Nous avons utilisé TypeScript pour développer le coté front-end et le service de l’application.

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript). Le code TypeScript est « transpilé » en JavaScript, pouvant ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript. TypeScript permet un typage statique optionnel des variables et des fonctions, la création de classes et d'interfaces, l'import de modules, tout en conservant l'approche non-contraignante de JavaScript.



Figure 34: Logo de Langage Type Script

**3.1.2 Frameworks et librairies**

**a. ASP.NET Core (.NET 5) :**

**ASP.NET Core** est une infrastructure multiplateforme, à hautes performances et open source développé par Microsoft pour la création d’applications modernes, basées sur le Cloud et connectées à Internet. Avec **ASP.NET Core**, vous pouvez : Créez des applications et services **Web**, des applications d’Internet des objets (IOT) et des serveurs principaux mobiles.



Figure 35 : Logo ASP.NET Core

**b. Angular** :

Angular est une plate-forme et un Framework JavaScript et open source développé par Google pour la construction d’applications clientes d’une seule page en HTML et TypeScript. Angular est écrit en TypeScript. Il implémente les fonctionnalités de base et facultatives sous la forme d’un ensemble de bibliothèques TypeScript que vous importez dans vos applications.



Figure 36 : Logo Angular

**c. HTML**

Nous avons utilisé HTML pour développer les interfaces web avec Angular.

L’HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. Permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.



Figure 37 : Logo HTML

**d. CSS**

Nous avons utilisé CSS pour ajouter de style à nous composantes, ce qui nous permet d’avoir une excellente expérience utilisateur.

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.



Figure 38 : Logo CSS

### 3.2 les principales interfaces :

**Set web :**

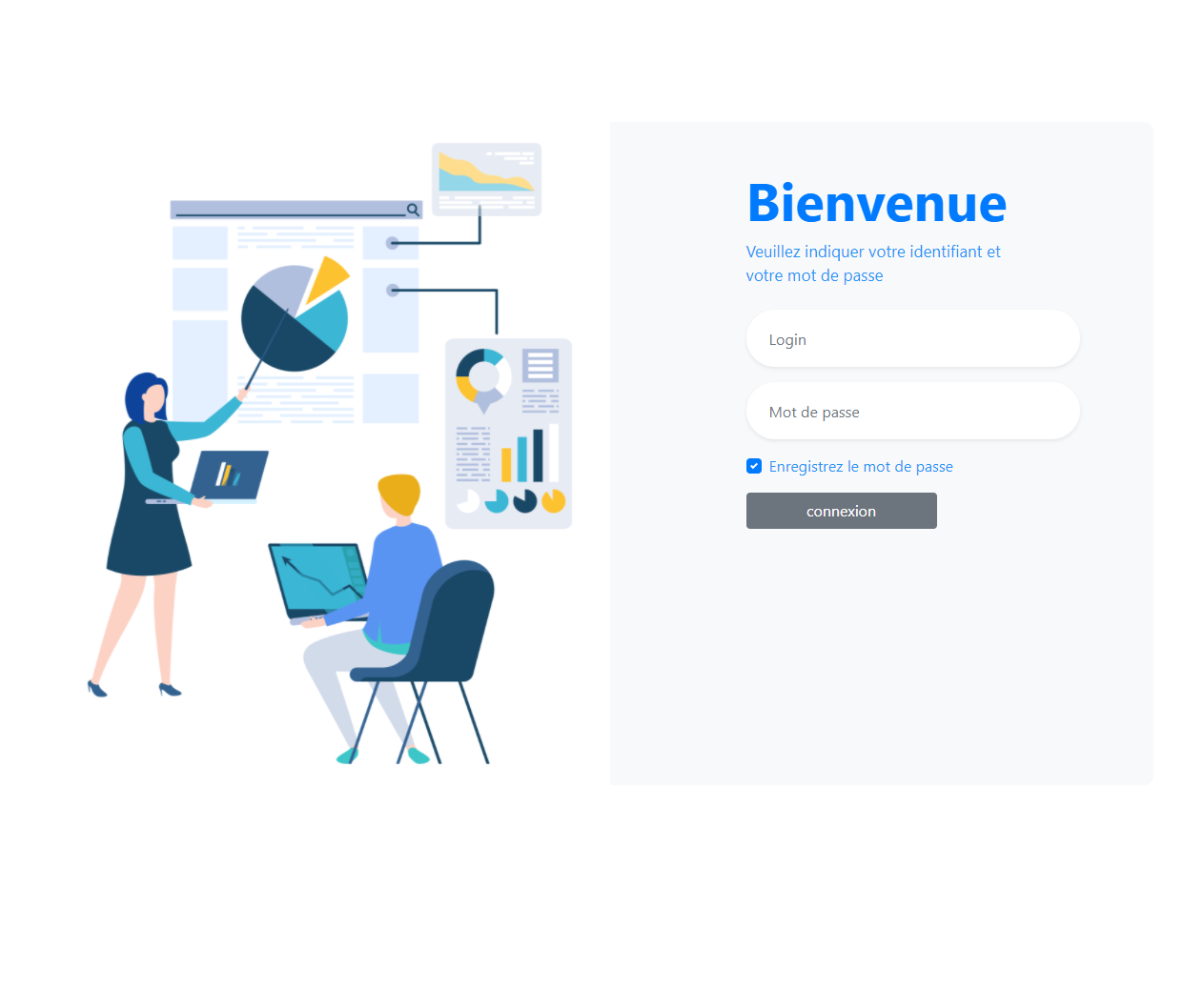
****

Figure 3 : page accueil

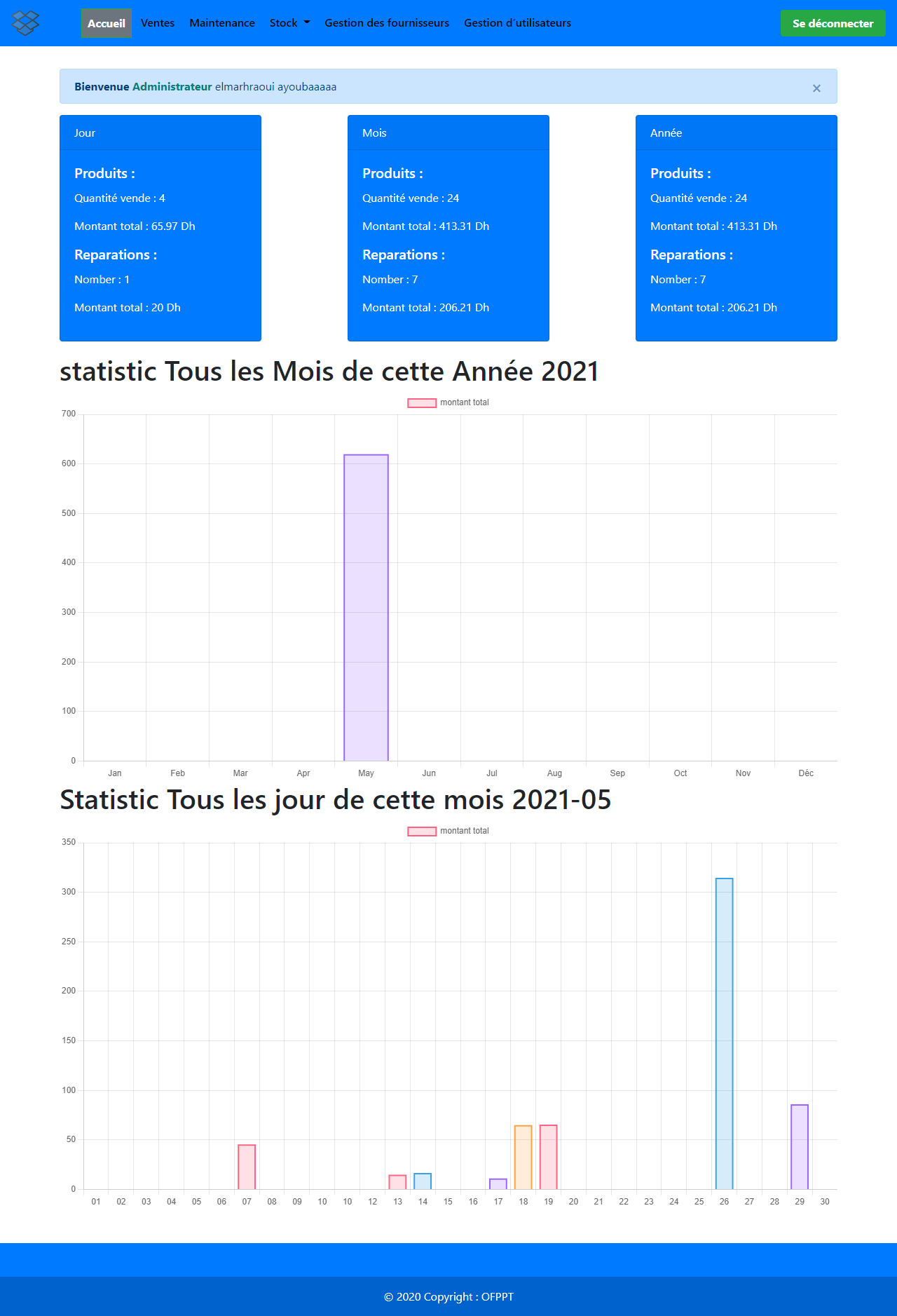
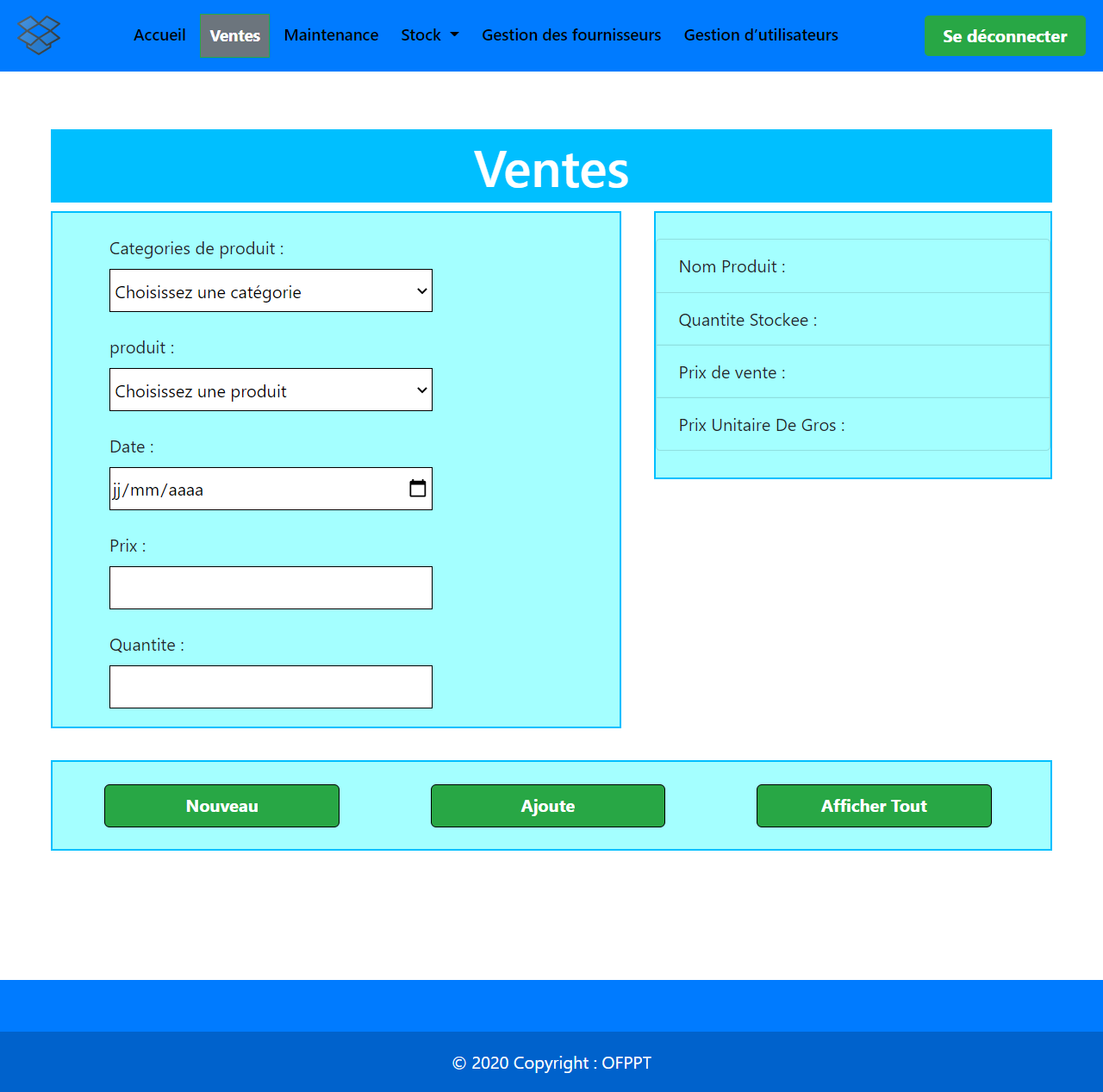


Figure 3 : page accueil



**Figure 10 : Page ajoute de vente**

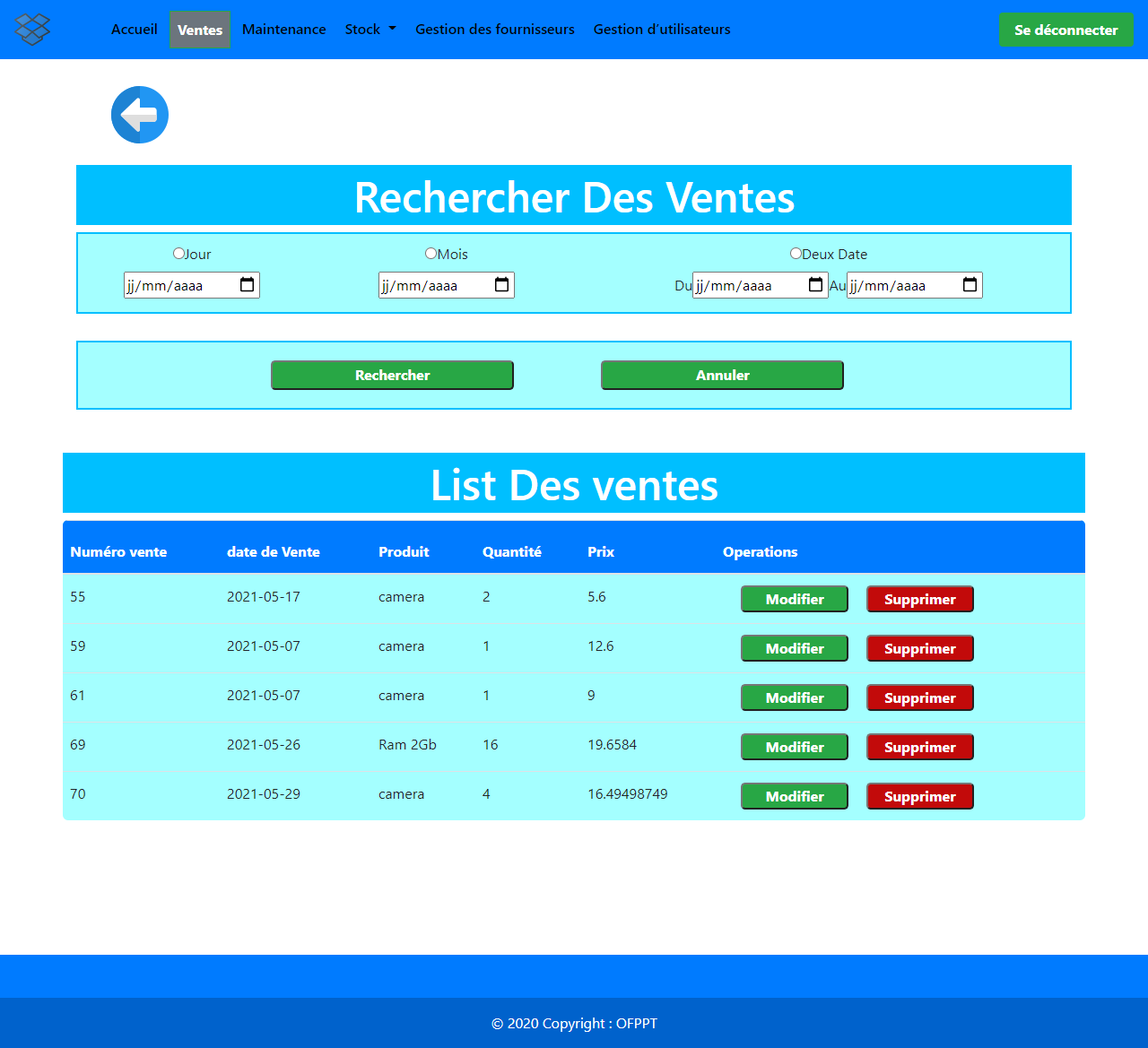


Figure 11 : Page recherche des ventes



Figure 15 : page gestion des utilisateurs

# Application desktop :

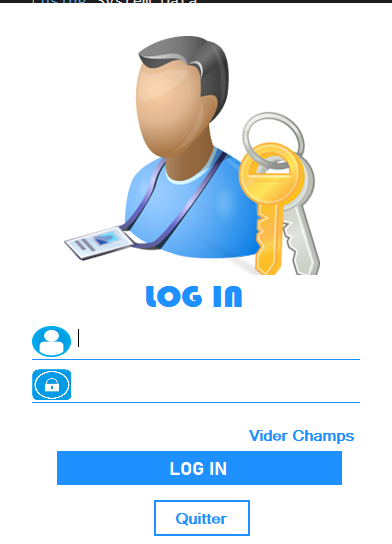
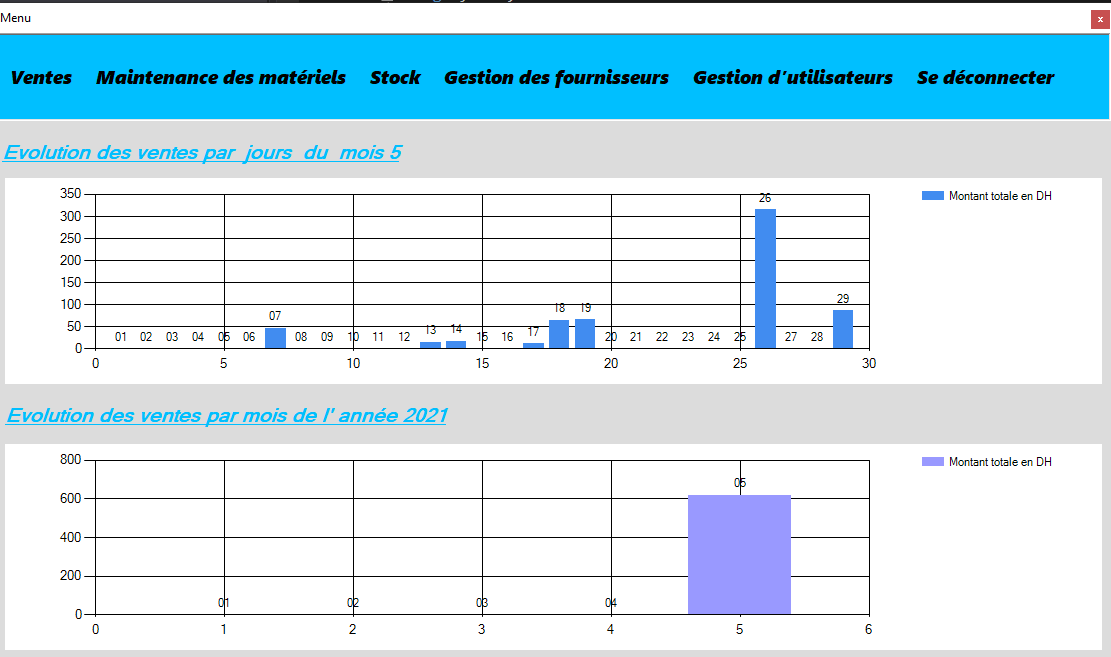


Figure 23 : forme de connexion à l'application de bureau



# Figure 24 : forme d’accueil à l'application de bureau

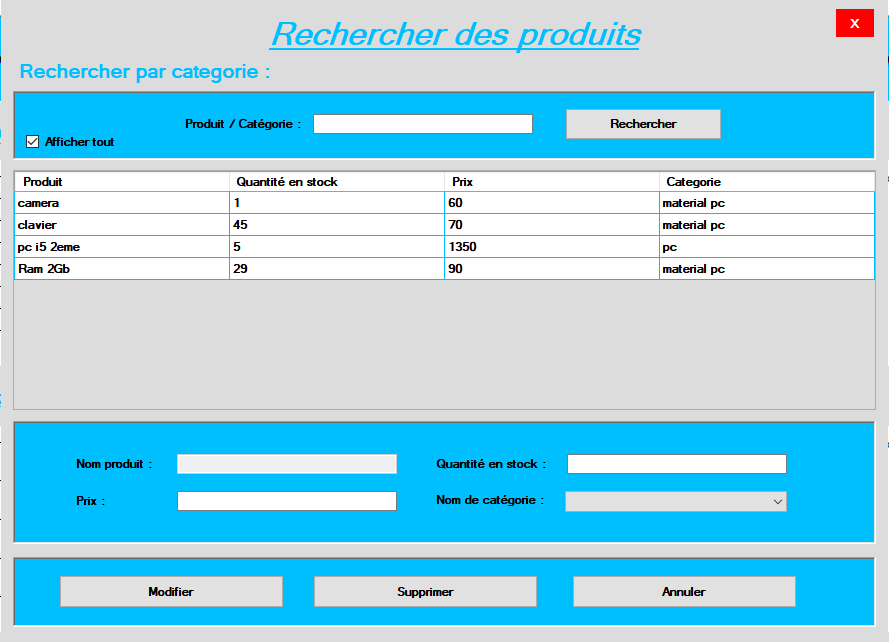


Figure 30 : forme recherché des produits à l'application de bureau



Figure 25 : forme gestion de vente à l'application de bureau

# Conclusion

Ce projet avait pour but de développer une application (web et bureau) de gestion des ventes, des réparations et de stock. Pour mener à bien ce projet, nous avons définis une démarche de travail sur laquelle nous avons commencé par identifier le contexte général en spécifiant les objectifs à atteindre, en introduisant la problématique à laquelle le projet doit répondre. Après nous avons effectués une étude de l’existant dans le but de relever les points négatifs et les limitations de l’ancienne méthode de gestion.

Comme suite de notre projet nous propos d’ajouter la possibilité de rédaction des rapports et des factures automatiquement, et d’ajouter des images aux produits vendus, et d’améliorer le design des interfaces utilisateurs pour avoir une excellente expérience utilisateur.

Ce projet de fin formation nous a permis de mettre en pratique nous connaissances théoriques acquises durant notre formation pendant ces deux années, et aussi de découvrir de nouvelles compétences techniques et de gestion à savoir : l’utilisation avancé de l’outil de versioning GIT, Angular, TypeScript, ASP.NET Core WebAPI la méthodologie de travail notamment l’écriture du code propre on utilisant quelques les principes de solide (Solide Principales) , les modèles de conception (Design Patterns) tel que Repository et MVC (Model, View, Controller) et nous offrant ainsi une bonne préparation pour le monde professionnelle.

# Références

1. **Formation sur Angular,** [Complète Formation sur Angular Framework | Bright Coding with Mohamed (teachable.com)](https://brightcoding.teachable.com/p/complete-formation-sur-angular-framework)**, consulté le 01/05/2021.**
2. **.NET Core C#8**, [(6) 1.Introduction in Asp.Net Core 5 Web Api in DotNet 5 and Angular 11 in Darija - YouTube](https://www.youtube.com/watch?v=IDfzvF1lXMY&list=PLR9SFNeqskaGhjFgyyKoShcdxtkF_GGhQ), **consulté le 15/05/201.**
3. **LINQ,** [(6) Introduction of LINQ(Darija) - YouTube](https://www.youtube.com/watch?v=W6-wIhWdN0E&list=PLR9SFNeqskaFmt7amE3EU0GDXEztgJypx), **consulté le 05/05/201.**
4. **Angular 7 and ASP.NET Core 2,** <https://www.youtube.com/watch?v=jXaEcT1-fGE&list=PLwj1YcMhLRN0AFQmc-vxwcguRy3TFER8m>**, consulté le 05/05/2021.**
5. **ASP.NET Core Fundamentals,** <https://www.youtube.com/watch?v=TWcW9-rYCvw&list=PLwj1YcMhLRN1zbN2olMzvIBXP06FIwoes>**, consulté le 05/05/2021.**
6. **Définition C#,** [C# — Wikipédia (wikipedia.org)](https://fr.wikipedia.org/wiki/C_sharp#:~:text=C%23%20est%20un%20langage%20de%20programmation%20orient%C3%A9e%20objet%2C,commandes%2C%20des%20widgets%20ou%20des%20biblioth%C3%A8ques%20de%20classes.), **consulté le 25/05/2021.**
7. **Logo C#,** <https://www.bing.com/images/search?view=detailV2&ccid=VNdbOFNY&id=B7C0403B6D1F4CF59DA9BA2A7C2BEED2CB66C>, **consulté le 25/05/2021.**
8. **ASP.NET Core,** [ASP.NET documentation | Microsoft Docs](https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0), **consulté le 25/05/2021.**
9. **Logo ASP.NET Core,** [ASP.NET Core Web API - Bing images](https://www.bing.com/images/search?view=detailV2&ccid=luZhB2nr&id=69790DF5F902099C3FBE7AE7A66F5733784EF0E2&thid=OIP.luZhB2nrtxjLDI6m8h-USwHaDY&mediaurl=https%3a%2f%2fwww.thecodebuzz.com%2fwp-content%2fuploads%2f2020%2f09%2fAPI-Versioning-in-ASP.NET-Core-with-Examples.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR96e6610769ebb718cb0c8ea6f21f944b%3frik%3d4vBOeDNXb6bneg%26pid%3dImgRaw&exph=806&expw=1765&q=ASP.NET+Core+Web+API&simid=608034113560125354&ck=E0BC2E67848292E9C1DF2BB8F3307944&selectedIndex=26&FORM=IRPRST&ajaxhist=0&ajaxserp=0), **consulté le 25/05/2021.**
10. **TypeScript,** <https://fr.wikipedia.org/wiki/TypeScript>**, consulté le 30/05/2021.**
11. **Logo TypeScript,** <https://worldvectorlogo.com/fr/logo/typescript>, **consulté le 30/05/2021.**
12. **Angular,** [Introduction au component Angular - apcpedagogie](https://apcpedagogie.com/introduction-au-component-angular/), **consulté le 30/05/2021.**
13. **Logo Angular,** [angular logo latest version - Bing images](https://www.bing.com/images/search?view=detailV2&ccid=7DsZgLiz&id=8B8D50A8216FBA0C787AF703B90B5E540943C40B&thid=OIP.7DsZgLizBt_wk8ndOAPzQAHaGs&mediaurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fRec3b1980b8b306dff093c9dd3803f340%3frik%3dC8RDCVReC7kD9w%26riu%3dhttp%253a%252f%252fzigterman.com%252fwp-content%252fuploads%252f2017%252f11%252fangular-logo.jpg%26ehk%3dTkGLYlyO%252btWSXy3FPA1goRdvdHFEjrdPl7hpLTnRLZE%253d%26risl%3d%26pid%3dImgRaw&exph=638&expw=705&q=angular+logo+latest+version&simid=608043884612707698&ck=194B43CC86C97CF135C54E59DE94E1E9&selectedIndex=2&FORM=IRPRST&ajaxhist=0&ajaxserp=0)**, consulté le 30/05/2021.**
14. **HTML,** <https://fr.wikipedia.org/wiki/Hypertext_Markup_Language>,  **consulté le 28/05/2021.**
15. **Logo HTML,** [logo html 5 - Bing images](https://www.bing.com/images/search?view=detailV2&ccid=4dQkxLm%2b&id=983ADCE82A6DF84D2DB0567436E67C9C3B0D9F11&thid=OIP.4dQkxLm-cAndV-9OfVjjQwHaE8&mediaurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fRe1d424c4b9be7009dd57ef4e7d58e343%3frik%3dEZ8NO5x85jZ0Vg%26riu%3dhttp%253a%252f%252f1.bp.blogspot.com%252f-NGHwBncyA68%252fUiMm_8b2ZUI%252fAAAAAAAAAnA%252f17OGXCKI4zE%252fs1600%252fLogo%252bHTML5.JPG%26ehk%3drnVe3RUksYQ4LMnsZ6Xxyf3F5lkj3Br1Eu6vOVCkYh0%253d%26risl%3d%26pid%3dImgRaw&exph=1067&expw=1600&q=+logo+html+5&simid=607994093058804541&ck=867644E40894B96561E3C068261CA2AB&selectedIndex=0&FORM=IRPRST&ajaxhist=0&ajaxserp=0), **consulté le 28/05/2021.**
16. **CSS,** <https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade>, ,  **consuléle 28/05/2021.**
17. **Logo CSS,** [logo CSS 3 - Bing images](https://www.bing.com/images/search?view=detailV2&ccid=Hj9OC1Gf&id=A5ACCC09DD7722797F2924F3C6BC284B8BAFD5BD&thid=OIP.Hj9OC1GfUFd5Xxhr1CJPuAHaKX&mediaurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR1e3f4e0b519f5057795f186bd4224fb8%3frik%3dvdWvi0sovMbzJA%26riu%3dhttp%253a%252f%252flogonoid.com%252fimages%252fcss3-logo.png%26ehk%3d1834Yvb9mgURw5ej9KtjAZvXXTxCWs8bA08xVL%252ftEBU%253d%26risl%3d%26pid%3dImgRaw&exph=1024&expw=731&q=+logo+CSS+3&simid=608056116678117562&ck=1BFD53EB768C392BDA8D12D5F0967682&selectedIndex=1&FORM=IRPRST&ajaxhist=0&ajaxserp=0),  **consuléle 28/05/2021.**

# Annexes

(*Times new roman 24 gras centré*)

**Annexe 1 :**

namespace ventes\_stocks\_management.Services

{

public class GenericService<TEntity> : IGenericServise<TEntity> where TEntity : class

{

private readonly Gestion\_Ventes\_Produits\_ReparationsContext \_dbContext;

private readonly DbSet<TEntity> \_table = null;

public GenericService(Gestion\_Ventes\_Produits\_ReparationsContext dbContext)

{

\_dbContext = dbContext;

\_table = \_dbContext.Set<TEntity>();

}

public void Add(TEntity entity)

{

try

{

\_table.Add(entity);

\_dbContext.SaveChanges();

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

public void Delete(int id)

{

try

{

var entity = Get(id);

if(entity != null)

{

\_table.Remove(entity);

\_dbContext.SaveChanges();

}

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

public TEntity Get(object id)

{

return \_table.Find(id);

}

public IEnumerable<TEntity> GetAll()

{

return \_table.ToList();

}

public void Update(TEntity entity)

{

try

{

\_dbContext.Entry(entity).State = EntityState.Modified;

\_dbContext.SaveChanges();

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

}

**Annexe 2 :**

export class UtilisateurService {

ULR:string="https://localhost:44322/Utilisateur";

constructor(private http:HttpClient) { }

getAllUsers()

{

return this.http.get(this.ULR);

}

getUser(id:number)

{

return this.http.get<Utilisateurs>((this.ULR+"/"+id.toString()).toString());

}

rechercherLogin(login:string)

{

return this.http.get(this.ULR+"/"+login);

}

postUser(user:Utilisateurs)

{

return this.http.post(this.ULR,user)

}

DeleteUser(id)

{

return this.http.delete((this.ULR+"/"+id.toString()).toString());

}

updateUser(user:Utilisateurs)

{

return this.http.put(this.ULR,user);

}

}