

## SÉRIE D'EXERCICES 7

### GESTION DE PRODUITS + ACCÈS AUX DONNÉES (ADO.NET)

---

#### Pour commencer

1. Créez un projet Web MVC en sélectionnant le modèle de projet application Web ASP.NET (.NET Framework) et nommez le projet : **ProduitApp**.
2. Vous êtes maintenant prêt.

#### Partie 1 :

On souhaite créer une application qui permet de gérer des produits. Chaque produit est défini par son identifiant, sa désignation et son prix.

L'application doit permettre de :

- ✓ Afficher tous les produits.
- ✓ Afficher les détails d'un produit.
- ✓ Saisir et ajouter un nouveau produit.
- ✓ Éditer et modifier un produit.
- ✓ Supprimer un produit.

Les produits sont stockés dans une collection de produits de type *Dictionary* dont la clé de chaque produit est représentée par son identifiant.

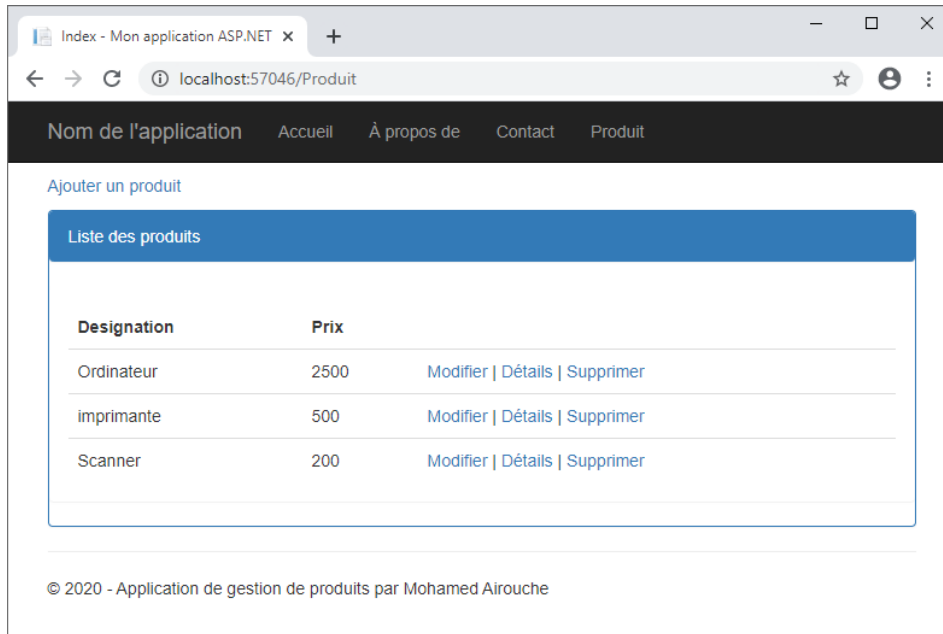
L'application se compose de deux couches :

- Une couche Service qui contient :
  - ✓ L'entité **Produit**.
  - ✓ L'interface **IProduitService**.
  - ✓ Une implémentation de cette interface.
- Une couche Web basée sur ASP.NET MVC.

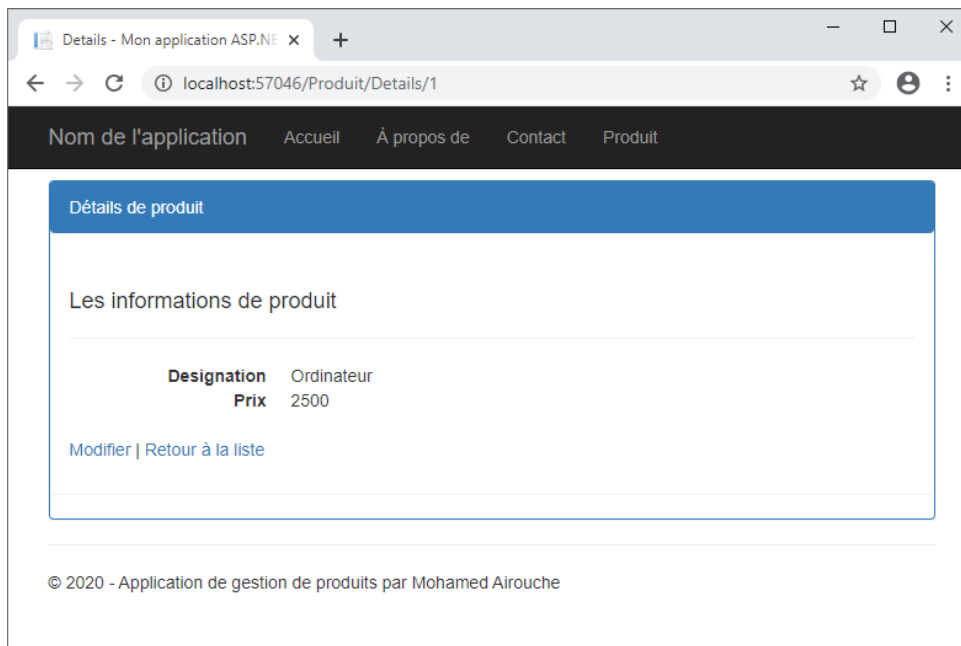
1. Ajoutez un nouveau dossier au projet. Nommez-le **Service**.
2. Ajoutez un modèle à l'application dans le dossier Service. Nommez votre classe **Produit.cs**. Cette classe permet de définir l'entité Produit qui est définie par son identifiant, sa désignation et son prix. Toutes les propriétés sont obligatoires. Le prix doit être un double dans l'intervalle [50, 1000000] et la désignation doit être une chaîne de caractères dont la longueur est dans l'intervalle [5, 25].
3. Ajoutez un contrôleur **ProduitController** dans le dossier Controllers du projet. Vous pouvez utiliser le modèle Contrôleur MVC 5 avec des actions de lecture/écriture.
4. Ajoutez du code dans les différentes méthodes d'action afin de réaliser les différentes fonctionnalités de l'application.
5. Ajoutez les différentes vues afin de réaliser les différentes fonctionnalités de l'application.

Pour avoir une idée du travail à effectuer, voici quelques captures d'écrans des pages qu'affiche l'application :

- ✓ La page qui permet d'afficher tous les produits



- ✓ La page qui permet d'afficher les détails d'un produit



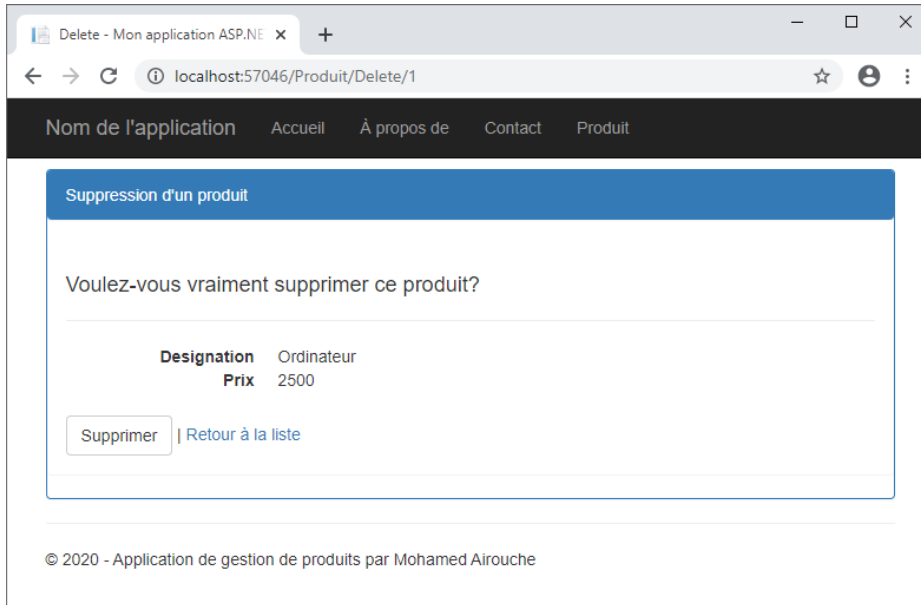
- ✓ La page qui permet de saisir et d'ajouter un nouveau produit

The screenshot shows a web browser window with the address bar displaying 'localhost:57046/Produit/AjoutProduit'. The page has a dark navigation bar with links: 'Nom de l'application', 'Accueil', 'À propos de', 'Contact', and 'Produit'. Below this is a blue header for 'Ajout d'un produit'. The main content area contains the text 'Veillez remplir les informations de produit à ajouter'. There are two input fields: 'Designation' and 'Prix', with the 'Prix' field containing the value '0'. Below the inputs is an 'Ajouter' button and a link 'Retour à la liste'. The footer shows '© 2020 - Application de gestion de produits par Mohamed Airouche'.

- ✓ La page qui permet d'éditer et de modifier un produit

The screenshot shows a web browser window with the address bar displaying 'localhost:57046/Produit/Edit/1'. The page has a dark navigation bar with links: 'Nom de l'application', 'Accueil', 'À propos de', 'Contact', and 'Produit'. Below this is a blue header for 'modification d'un produit'. The main content area contains the text 'Editer les champs à modifier pour le produit'. There are two input fields: 'Designation' containing 'Ordinateur' and 'Prix' containing '2500'. Below the inputs is a 'Modifier' button and a link 'Retour à la liste'. The footer shows '© 2020 - Application de gestion de produits par Mohamed Airouche'.

- ✓ La page qui permet de supprimer un produit



6. Ajoutez une interface au dossier Service. Nommez-le : **IProduitService.cs**. Cette interface contiendra les opérations de la logique métier pour réaliser les différentes fonctionnalités de l'application.
7. Ajoutez une classe **ProduitServiceImp.cs** au dossier Service. Cette classe implémente l'interface et permet de réaliser les différentes fonctionnalités de l'application avec une liste de produits qui sont stockés dans une collection de produits de type *Dictionary* dont la clé de chaque produit est représentée par son identifiant.

## Partie 2 :

### Gestion d'une liste de produits se trouvant dans une base de données

Dans cette deuxième partie, on demande d'ajouter une deuxième implémentation **ProduitServiceImpBD** de l'interface **IProduitService** pour gérer les produits qui sont stockés dans une base de données.

1. Créez une base de données **ProduitAppBD** sur le serveur de base de données locale.
2. Ajoutez une table **tbProduit** dans cette base de données qui permet de stocker des produits avec les mêmes propriétés définies dans la partie 1.
3. Ajoutez du code pour l'implémentation **ProduitServiceImpBD** pour gérer les produits qui sont stockés dans la base de données en ayant les mêmes fonctionnalités de l'application de la partie 1.

## Remise :

- À suivre... (ne pas remettre tout de suite).