|  |
| --- |
| Université du Québec à Rimouski |
| Travail Pratique 2 |
| 8INF957 - Programmation objet avancée  Professeur – Yacine Yaddaden, Ph. D. |

|  |
| --- |
| ⭢Tidjet Zaid  ⭢ Benjamin Lapointe pinel |

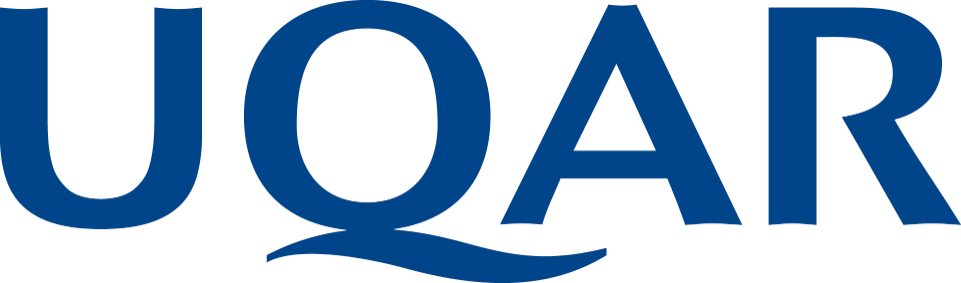


Table des matières

[Introduction 2](#_Toc102165464)

[Aperçu de l’interface web 2](#_Toc102165465)

[API REST 5](#_Toc102165466)

[consommation DE L’API rest avec le wpf 5](#_Toc102165467)

[Conclusion 5](#_Toc102165468)

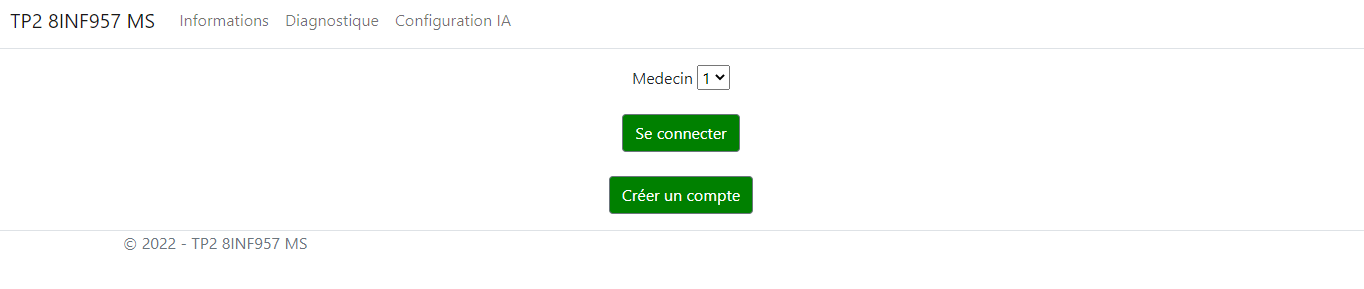
[Références 5](#_Toc102165469)

# Introduction

Le but de ce travail est de faire une application web dynamique pour la gestion des médecins qui permet d’interagir rapidement et avec fluidité avec l’utilisateur en utilisant la programmation d’interface web, création d’une API REST et intégration avec une application de bureau, un ORM et une base de données comme concepts.

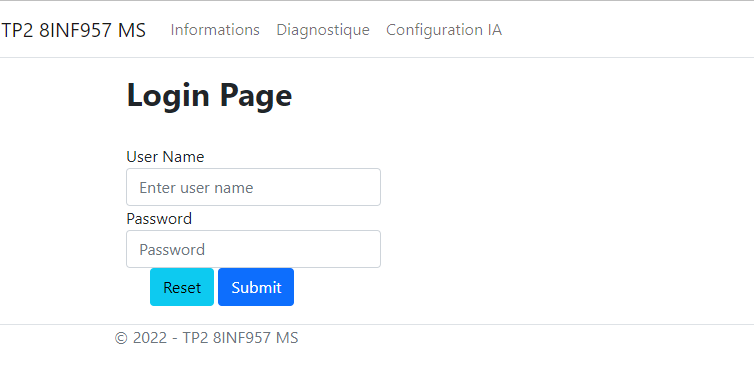
# Aperçu de l’interface web

Dans cette première partie du travail nous avons créé un projet asp.net core vide qui s’appelle « 8INF957\_MS-TP2 » et on a fait dessus l’interface web (vue razors) avec toutes les fonctionnalités qui permet à un médecin de se connecter ou de créer un compte, d’ajouter un patient, de faire un diagnostique et d’avoir toutes les informations en utilisant le principe des modèles vue et Controller.

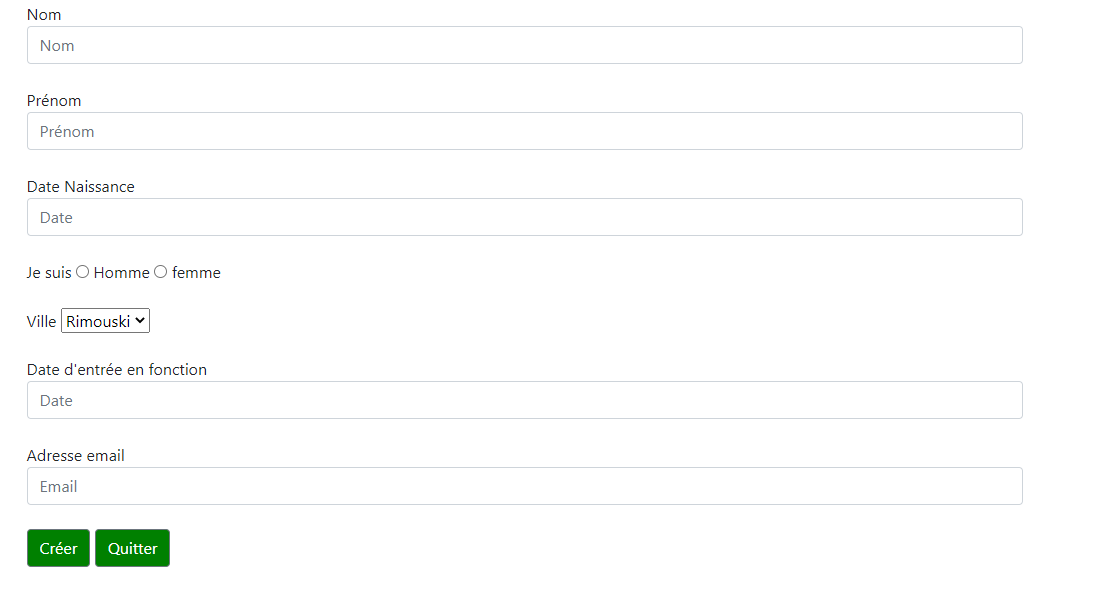
1. **Connexion :**
2. **Se connecter :**

L’authentification par session a été utilisé pour déterminer l’identité d’un

Utilisateur. L’autorisation est le processus qui consiste à déterminer si un l’utilisateur a accès à une ressource en fournissant sont UserName et son password.

****

1. **Créer un compte :**

****

1. **Ajouter un patient :**

**Une image contenant texte

Description générée automatiquement**

1. **Configuration IA :**

****

1. **Diagnostique :**

**Une image contenant texte

Description générée automatiquement**

1. **Informations :**

Une image contenant texte

Description générée automatiquement

# API REST

Dans cette partie un autre projet webAPI ‘RESTApi’ qui est utilisé par le projet ASP.NET et l’outil swagger a été installé et configuré qui nous permet de faciliter l’utilisation de **L’API REST** et de manipuler les requêtes Http pour les médecins et de cela on peut :

* Recevoir la liste des patients du médecin connecté.
* Récupérer les informations d’un patient en particulier avec l’ID ou le supprimer et en cas de suppression d’un patient, l’ensemble de ses diagnostiques sont supprimés.
* Récupérer la liste des diagnostiques appartenant au patient avec l’ID défini.
* Récupérer un diagnostique particulier avec l’ID ou le supprimer.
* Effectuer le diagnostique d’un patient avec les informations fournies.
* Récupérer ou mettre à jour les informations du médecin connecté.

L’authentification forte par jeton JWT s’appuie sur un protocole qui permet à un utilisateur de recevoir un [jeton d’accès](https://www.okta.com/identity-101/access-token/) unique après avoir confirmé son identité et dans notre cas ça nous permet de fournir l’identifiant du médecin ainsi que son mot de passe pour recevoir le jeton.

# consommation DE L’API rest avec le wpf

Dans cette partie la consommation de L’API REST est utilisée au niveau de services/RestApi.cs à travers les requêtes http avec la classe HttpClient. Avec les requêtes on peut :

* Récupérer la liste des clients du médecin connecté dans une liste déroulante
* Effectuer le diagnostique en cliquant sur un Button.
* Afficher le résultat en bas après traitement côté serveur.

# Conclusion

# Références

* Cours
* [appeler une API Web à partir d’un Client .net (C#)-ASP.NET 4. x | Microsoft Docs](https://docs.microsoft.com/fr-fr/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client)
* Labat, L. (2013). ASP.NET MVC 4 - Développement d'applications Web en C# - Concepts et bonnes pratique. Éditions ENI.
* [vue d’ensemble de l’authentification ASP.NET Core | Microsoft Docs](https://docs.microsoft.com/fr-fr/aspnet/core/security/authentication/?view=aspnetcore-6.0)
* D’autres références sont faites dans le code source des projets