

Module .NET

UFR Sciences et Techniques

Université de Rouen

Master 2 GIL

FTEL

TP1

Durée 3h



Installation de Visual Studio Community

1. Téléchargez Visual Studio Community - <https://visualstudio.microsoft.com/fr/downloads/>
2. Installez Visual Studio Community
 - a. En cochant « Développement web et ASP.NET »
3. Lancez Visual Studio

Création du projet

1. Créer une application MVC
 - a. Fichier
 - b. Nouveau projet
 - c. Visual C# / Web / Application web ASP.NET (.NET Framework)
 - d. Nommez le « M2Link »
 - e. Cochez « MVC » afin d'avoir le template initial
 - i. Laissez Décoché : Web Forms, API Web, Docker, Tests unitaires
 - ii. Laissez l'Authentification : Aucune authentification

Création de compte

Définition de votre premier modèle

1. Créez une classe « RegisterModel » avec les propriétés suivantes :
 - a. Nom
 - b. Prénom
 - c. Email
 - d. Pseudo
 - e. Mot de passe
 - f. Vérification du mot de passe

Définition de votre premier formulaire

1. Créez le controller « Register »
2. Créez l'action « Form » (Méthode en Get)
3. Créez la vue « Form » correspondante
 - a. Pensez à mettre le model en paramètre de la vue
4. Créez le formulaire dans la vue
 - a. Pensez à utiliser
 - i. Html.BeginForm
 - ii. Html.EndForm
 - iii. LabelFor
 - iv. TextBoxFor
 - v. ValidationMessageFor
5. Créez l'action « Form » (Méthode en Post) pour récupérer les données envoyées du formulaire
 - a. Pensez à vérifier que vous récupérez bien les données en utilisant le debugger
6. Ajouter les vérifications nécessaires
 - a. Champs obligatoires
 - b. Mot de passe de 8 caractères + mélange lettres, majuscules, nombres, caractères spéciaux
 - c. Pensez à utiliser le ModelState.IsValid
 - d. Si le modèle est valide, redirigez vers la page d'accueil
 - e. Sinon, réaffichez le formulaire avec les messages d'erreurs

Sauvegarde (temporaire) des données

En attendant la sauvegarde des données dans une base, nous allons créer une classe simulant la table des utilisateurs.

1. Créez votre entité « User »
2. Créez une classe « Database »
3. Ajoutez un attribut public static « Users », liste de « User »

Nous stockerons les utilisateurs dans cet attribut.

Attention, les données en static sont perdues à chaque redémarrage / compilation de l'application.

4. Lors de la création de compte, avant de rediriger l'utilisateur vers la page d'accueil, créez et sauvegardez l'utilisateur dans cette liste
 - a. Pensez à vérifier que les données se stockent bien avec le debugger

Gestion de l'authentification

1. Créez une page de connexion
 - a. Avec vérification du pseudo / mot de passe
 - b. Si valide, redirigez vers l'accueil
 - c. Sinon, réaffichez la page avec les messages d'erreur
2. Ajouter l'authentification avant de rediriger vers l'accueil
 - a. Pensez à utiliser « FormsAuthentication.SetAuthCookie »
3. Empêchez l'accès à la page d'accueil sans être authentifié
 - a. Pensez à utiliser « [Authorize] »
 - b. Pensez à renseigner le mode d'authentification et l'url de la page de login dans le web.config

```
<system.web>  
  <authentication mode="Forms">  
    <forms loginUrl="~/Login/Form" />  
  </authentication>  
</system.web>
```

4. Ajouter un bouton de déconnexion dans le menu

Edition de son profil

1. Créez une page permettant d'éditer son profil
 - a. Créez un modèle
 - b. Affichez les anciennes valeurs
 - c. Sauvegarder les données

Affichage des utilisateurs

1. Créer une page pour afficher l'ensemble des utilisateurs inscrits sur votre M2Link
 - a. Il faut être authentifié pour y accéder
2. Créer une page de profil pour afficher un profil d'utilisateur
 - a. Il faut être authentifié pour y accéder

Pour aller plus loin

1. Ajoutez une photo aux profils des utilisateurs
2. Ajoutez une description aux profils des utilisateurs
3. Chiffrez les mots de passe
4. Mettez en forme les pages
 - a. Pensez à utiliser le Layout