**Examen formatif**

* Durée : 2 heures.
* Toute documentation permise. Recherche Web permise.
* Communication via Teams, Discord, Messenger, etc. interdite. L’examen se fait de manière individuelle !
* Si vous n’arrivez pas à compléter une question, laissez au moins vos tentatives dans le code en //commentaires !

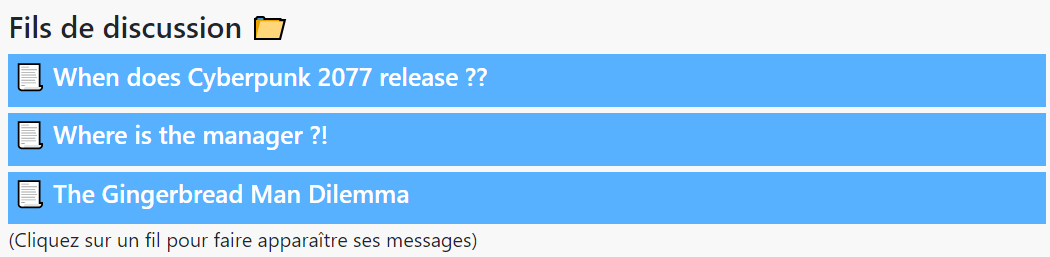
**Attention** : Le projet Web API utilise la base de données *sqllocaldb*. Avant de démarrer le projet API, utilisez la commande dotnet ef database update pour générer la base de données et son seed.

**(2 pts) Authentification par courriel et sécuriser une action**

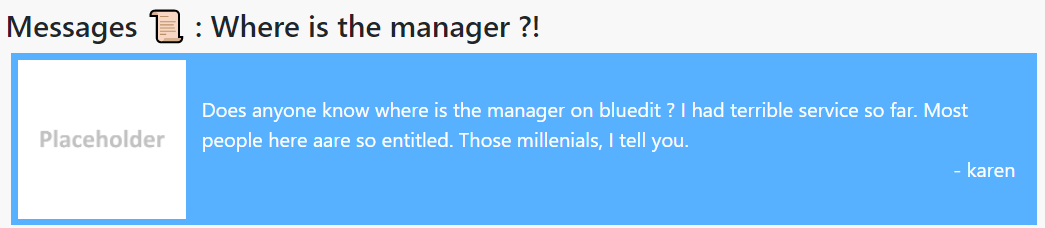
* Les utilisateurs doivent pouvoir se connecter au serveur de deux manières (au choix) :
  + Nom d’utilisateur + Mot de passe
  + Courriel + Mot de passe (On écrit le courriel dans le champ « Nom d’utilisateur » à ce moment)

* Pour le moment, même sans se connecter, nous avons accès aux messages d’un fil de discussion en cliquant dessus. Faites les modifications nécessaires pour qu’il soit seulement possible de consulter les messages d’un fil si on est authentifié avec un token valide.
  + On doit pouvoir continuer de voir ceci sans être authentifié :

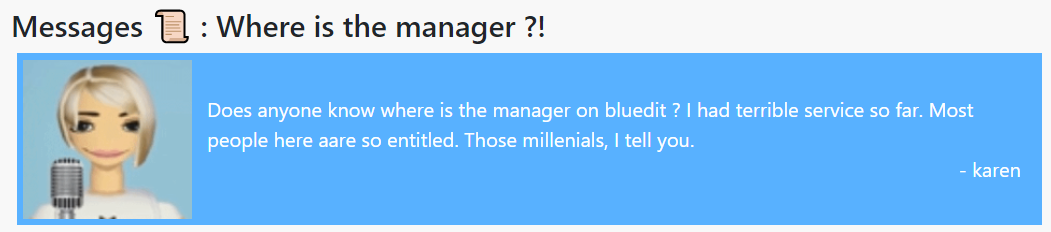


* + On ne doit pas pouvoir voir ceci à moins d’être authentifié :



**(2 pts) Afficher les avatars (images) des utilisateurs**

* Actuellement, pour chaque utilisateur du seed, une image temporaire est utilisée comme avatar. En vérifiant l’action GetPictures dans UsersController, faites la modification nécessaire au projet Angular pour afficher les avatars de ces utilisateurs.



**(2 pts) Ajouter un message au seed**

* Modifiez le seed pour lui ajouter un message supplémentaire. (Si possible un message scandaleux pour épicer la correction) Il doit être lié à l’utilisateur de votre choix et au fil de discussion de votre choix, tant qu’ils sont déjà dans le seed.
  + Mettez à jour les migrations pour vous assurer que nous puissions voir le nouveau message lorsque la base de données sera mise à jour.

**(4 pts) Chercher les messages d’un utilisateur**

* Faites les modifications nécessaires pour qu’on puisse obtenir les messages d’un utilisateur spécifique à l’aide du formulaire ci-dessous. (Notez que le formulaire fonctionnera seulement si on est authentifié et c’est le comportement souhaité)
  + Il faut modifier une action sur l’application serveur et modifier une fonction sur l’application cliente.
  + L’affichage dans le HTML est déjà géré, il reste à remplir la variable *this.messagesRecherche* dans le composant app.
  + Si le nom d’utilisateur recherché n’existe pas, on *return BadRequest()* dans l’action.
  + Il n’est pas nécessaire de gérer le cas où un utilisateur n’aurait posté aucun message.



**(5 pts) Créer un nouveau message dans un fil existant**

* Faites les modifications nécessaires pour qu’on puisse ajouter un nouveau message dans un fil de discussion lorsqu’on est connecté.
  + Le message créé doit être lié à l’utilisateur l’ayant posté et au fil de discussion concerné.
  + Il y a des modifications à faire côté client et côté serveur.
  + Ce n’est pas grave si le nouveau message n’apparait pas immédiatement, tant qu’il est présent si on rafraîchit le fil de discussion.

