420-4W6 - TP4 - Post Hub (20%)

Date de remise : 8 décembre 23h59

# Consignes

* Le projet sera fait en équipe de deux.
* Le projet client sera fait avec le Framework Angular. (Projet de départ est fourni)
* Le projet serveur sera fait avec le Framework ASP.Net Core. (Projet de départ fourni)
* Le projet doit être remis sur Léa / Teams et sur Git.

# Mise en contexte

Mise en garde : ce TP sera nettement plus complexe que le TP3. Toutes les notions ne sont pas forcément dans les notes de cours. (Presque toutes le sont) Vous aurez parfois à chercher un peu sur internet. Dans tous les cas, n’hésitez pas à poser des questions à l’enseignant, comme toujours, pour avoir des pistes pour orienter vos recherches. Ce TP vous préparera au cours suivant, Programmation Web avancée, pour lequel les TPs seront complexes et se feront à 4.

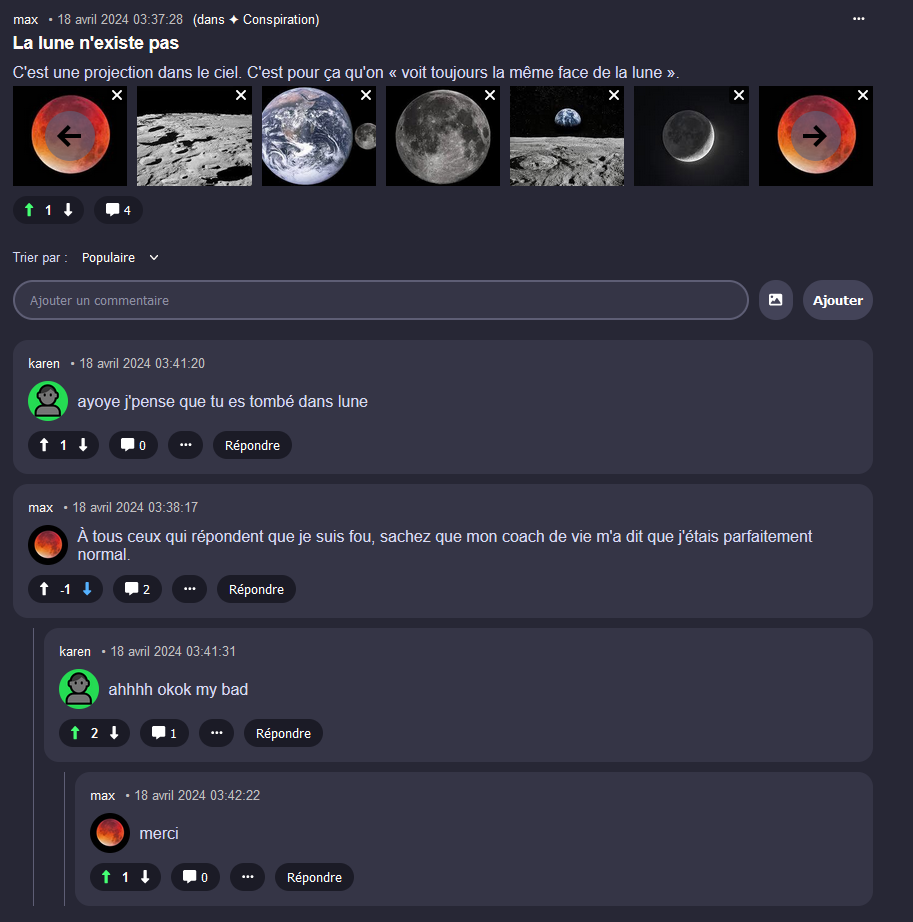
Une application qui sert de multi-forums de discussion vous est fournie. Vous devrez ajouter des fonctionnalités à l’application, principalement pour la gestion d’images et de rôles.

En résumé, l’application contient des **hubs**, des **posts** et des **commentaires**.

* Un **hub** est un rassemblement de **posts** autour d’un même thème. Par exemple, le hub « ✦Conspiration » contient plusieurs discussions (**posts**) conspirationnistes.
* Un **post** est un titre et du texte (un « **commentaire** principal ») suivi de plusieurs commentaires. Chaque **commentaire** peut avoir des sous-commentaires.
* Un **commentaire** peut être le message principal d’un **post** ou encore être un des nombreux commentaires qui visent à répondre à un **post**. (Ou à un autre **commentaire**)

Dans la prochaine page, on peut voir un **post**. (Qui a été créé dans le **hub** « ✦Conspiration ».

* Le **commentaire** principal du **post** est situé en haut, après le titre.
* Plus bas, il y a 2 **commentaires** associés au post. (Ainsi que deux autre sous-commentaires associés à des commentaires)



# Liste des fonctionnalités à ajouter

Chaque membre de votre équipe a une liste de fonctionnalités à implémenter. Vous ne pouvez pas partager les fonctionnalités librement, car elles sont séparées de manière à vous préparer pour l’examen. Rien ne vous empêche de vous entraider ponctuellement !

Il y a plus de détails sur les fonctionnalités dans les documents **yin.docx** et **yang.docx**.

|  |  |
| --- | --- |
| Membre yin | Membre yang |
| Étape A -  : Vous entendre sur quel membre de l’équipe est **yin** et quel membre de l’équipe est **yang**. Si vous êtes en classe et que votre partenaire est absent(e), vous **devez** choisir sans lui / elle. Pas le choix pour pouvoir commencer le travail. | |
| Étape B -  :   * Créer un repository Git qui contient les deux projets de départ. * Ajouter l’autre membre de l’équipe et l’enseignant(e) en collaborateurs * Créer une branche dev. Le projet ne sera merge dans la branche main qu’à la toute fin du projet pour la remise.(À chaque fois que vous travaillez sur une fonctionnalité, créez une nouvelle branche à partir de dev. Quand elle est terminée, vous pouvez merge cette branche dans dev, puis recommencer une nouvelle branche pour la prochaine fonctionnalité…) | |
| Étape C -  :   * Lorsqu’on crée un commentaire, on doit être capable d’y joindre une ou plusieurs images. | Étape C -  :   * Lorsqu’on crée un post, on doit être capable d’y joindre une ou plusieurs images. |
| **Halte !**   * Au plus tard le **27 novembre**, vous devez, à deux, vous présenter en classe et prendre 5 minutes pour **me montrer que vous avez complété les étapes A à C**. Si vous ne le faites pas, il y a une **pénalité de 25%** dans la note du / des équipiers absents. * Si vous faites la présentation seul(e), seul(e) votre partenaire reçoit la pénalité. * Si vous faites la présentation en retard, il y a une pénalité, mais plus petite. * Si les étapes A à C ne sont pas toutes complétées, la présentation ne peut pas être faite. * Pourquoi cette halte agaçante ? C’est un gros projet, en équipe en plus ! Je veux m’assurer que vous ayez commencé tôt. Rassurez-vous : les étapes A à C ne sont pas si longues… * Il n’y a rien à préparer pour la présentation … à part votre TP ! | |

|  |  |
| --- | --- |
| Étape D -  :   * Les utilisateurs doivent pouvoir choisir un avatar personnalisé. Il est affiché à côté de leurs commentaires. (Mais pas dans le message principal d’un post) * L’avatar peut être changé à tout moment. | Étape D -  :   * Si un post (commentaire principal d’un post) contient plus de 4 images, les images sont affichées avec un carrousel glidejs plutôt qu’en simple rangée. * Les commentaires d’un post n’ont jamais de carrousel. |
| Étape E -  :   * Cliquer sur une image dans un post (commentaire principal d’un post) doit permettre de l’afficher en pleine taille, dans un autre onglet. | Étape E - :   * Lorsqu’on modifie un commentaire ou un post, on doit pouvoir ajouter une ou plusieurs images supplémentaires en même temps de modifier le texte. (Les anciennes images du commentaire, s’il y en avait, sont préservées) |
| Étape F -  :   * On doit pouvoir supprimer les images d’un commentaire / post, individuellement. | Étape F -  :   * Lorsqu’on supprime un commentaire ou un post, toutes ses images doivent être supprimées. |
| Étape G -  :   * Les utilisateurs peuvent signaler (Report) les commentaires / posts des autres utilisateurs. | Étape G -  :   * Les utilisateurs doivent pouvoir se connecter en utilisant leur nom d’utilisateur OU leur adresse courriel. (Plutôt que seulement leur nom d’utilisateur) * Les utilisateurs doivent pouvoir changer leur mot de passe. |
| Étape H -  :   * Un rôle modérateur existe. * Les modérateurs peuvent voir la liste des commentaires signalés. Ils peuvent supprimer les commentaires de leur choix via cette liste. * Un utilisateur avec le rôle modérateur est ajouté dans le seed. | Étape H -  :   * Un rôle administrateur existe. * Les administrateurs peuvent ajouter le rôle modérateur à des utilisateurs. * Un utilisateur avec le rôle administrateur est ajouté dans le seed. |
| Étape I -  :   * S’assurer que les suppléments agaçants (Voir grille de correction) sont respectés. * S’assurer que tout a été merge dans dev, puis dans main. * S’assurer que tout fonctionne. | |

# Correction

|  |  |  |
| --- | --- | --- |
| 10 pts | Étape C | |
|  | 5 pts | On peut ajouter des images lorsqu’on crée un post et les voir ensuite. |
| 5 pts | On peut ajouter des images lorsqu’on crée un commentaire et les voir ensuite. |
| 6 pts | Étape D | |
|  | 3 pts | Les utilisateurs peuvent choisir un avatar et le modifier à volonté. |
| 3 pts | Les images d’un post sont affichées dans un carrousel à 5 images ou plus. |
| 6 pts | Étape E | |
|  | 2 pts | Toute image (sauf les avatars) peut être cliquée pour être affichée en grand dans un nouvel onglet. |
| 4 pts | La modification de post / commentaire permet d’ajouter des images. |
| 6 pts | Étape F | |
|  | 4 pts | Les images des posts et des commentaires peuvent être supprimées individuellement par l’auteur. |
| 2 pts | Supprimer un post ou un commentaire supprime ses images. |
| 4 pts | Étape G | |
|  | 2 pts | Les posts / commentaires des autres utilisateurs peuvent être signalés si on est connecté. |
| 1 pt | La connexion fonctionne avec le nom d’utilisateur et le courriel. |
| 1 pt | On peut modifier son mot de passe à volonté. |
| 8 pts | Étape H | |
|  | 4 pts | Le rôle de modérateur existe, permet de supprimer des commentaires signalés et présente un utilisateur avec le rôle dans le seed. |
| 4 pts | Le rôle d’administrateur existe, permet de nommer des modérateurs et présente un utilisateur avec le rôle dans le seed. |
| 20 pts | Suppléments agaçants | |
|  | 6 pts | Sécurité sur le serveur (Aucune action ne peut être exécutée par un utilisateur qui n’en a pas le droit) Ceci touche les objets qui ne peuvent être modifiés que par leur auteur et les opérations qui ne peuvent être utilisées que par les modérateurs / administrateurs. Cela touche aussi à des choses plus subtiles comme ne pas pouvoir signaler son propre commentaire. |
| 3 pts | Fausse sécurité sur le client Angular. (Aucun bouton permettant de faire une action qu’on n’a pas le droit de faire ne doit être VISIBLE dans les pages Web.) La manière la plus simple de cacher les boutons / options / sections auxquelles on ne devrait pas avoir accès est d’envoyer des données supplémentaires avec le token lorsqu’on se connecte. (Ces données, qui seraient stockées dans le localStorage, permettront d’indiquer si l’utilisateur connecté est un administrateur et / ou un modérateur, par exemple) |
| 3 pts | L’architecture du projet serveur respecte les lignes directrices vues dans les notes de cours. (Les contrôleurs n’ont pas accès au DbContext, usage de Models et de DTOs au besoin, etc.) De plus, sur le serveur, évitez la répétition de code dans la mesure du raisonnable. |
| 1 pt | Toute image supprimée / remplacée dans la BD doit être supprimée du disque également. |
| 2 pts | L’architecture du projet client respecte les lignes directrices du cours. (Pas de requête avec HttpClient dans un composant, usage de Models, services, composants, etc.) |

|  |  |  |
| --- | --- | --- |
|  | 5 pts | Git a été utilisé de manière responsable, c’est-à-dire :   * Une branche dev a été créée et elle a seulement été merge dans main à la toute fin du TP. * Une branche a été créée à partir de la branche dev pour chaque étape / fonctionnalité pour limiter les conflits avec son coéquipier. * Le **français** est évalué dans les messages des commits.\* (Voir plan de cours) * Les titres et les descriptions des commits sont appropriés et clairs. |
| -10 pts | Pénalité possible si des fonctionnalités qui étaient déjà implémentées se sont mises à ne plus fonctionner. |
| -5 pts | Pénalité possible si l’interface est dangereusement déformée ou altérée. |
| -15 pts | Pénalité possible pour la présentation des étapes A, B et C. |
| 60 pts | Total | |

# Annexe

Liste des requêtes disponibles sur le serveur

La plupart de ces requêtes existent déjà !

Une image contenant texte, capture d’écran, nombre, Parallèle

Description générée automatiquement

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

À quoi correspond chaque composant Angular ?

* HubComponent

Dès qu’une série de posts sont affichés, (onglets Accueil, Découvrir ou encore un hub précis) on est dans le composant HubComponent.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

* NewHubComponent

Accessible en créant un nouveau hub.

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* PostThumbnailComponent

Dès qu’on est dans le composant HubComponent, les multiples posts sont chacun affichés à l’aide du composant PostThumbnailComponent.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

* PostComponent

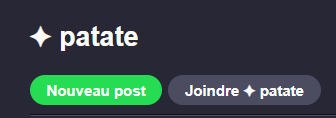
Si on clique sur un post pour l’afficher en entier, on arrivera dans PostComponent.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

* EditPostComponent

Lorsqu’on est dans un hub et qu’on clique sur « Nouveau post », on atterit dans le composant EditPostComponent pour créer un nouveau post.

Une image contenant capture d’écran, Logiciel multimédia, logiciel

Description générée automatiquement

* CommentComponent

Le composant CommentComponent est utilisé dès qu’on commentaire qui n’est pas le commentaire principal d’un post est affiché. Ce composant peut d’ailleurs lui-même contenir un autre CommentComponent si un commentaire possède des sous-commentaires.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

* ModerateCommentsComponent

Bien que ce composant existe déjà et qu’il est associé à la route /reports, il n’est pas encore accessible. Il vous servira lorsque vous implémenterez la gestion des commentaires signalés. Il permet d’afficher plusieurs commentaires car il peut contenir plusieurs CommentComponent.