

OBJECTIFS

- ✦ Créer une application Web (sans BD).
- ✦ Planifier le travail à faire et identifier les principales fonctions, objets et interactions à développer.
- ✦ Programmer la logique applicative côté client.
- ✦ Programmer les interfaces côté client en exploitant **Bootstrap et JavaScript (jQuery)**.
- ✦ Programmer la logique applicative côté serveur avec le Framework **ASP.NET Core MVC**.
- ✦ Programmer en orienté-objet avec C#.
- ✦ Respecter le modèle MVC.
- ✦ Valider le fonctionnement de l'application.

AVANT DE COMMENCER CE TP

Avant de commencer à réalisation de ce Tp, il serait important de mettre à jour le code du Tp 1 selon les conseils et commentaires reçus de votre professeur lors de la démo. Dans la plupart des cas, il s'agissait de renforcer la robustesse de vos classes par de meilleures contraintes sur vos propriétés et d'ajuster les tests unitaires en conséquence. Dans le cas présent, il serait aussi intéressant, si c'est le cas, de sortir la liste de favoris de la classe utilisateurs et de concevoir les classes nécessaires pour en faire la gestion.

DESCRIPTION DU PRÉSENT TP

Créer une application Web faisant la gestion de produits multimédias choisis dans le Tp 1 (par exemple : film, vidéo, jeux, etc.). Votre application ne comporte aucune donnée provenant d'une base de données. Elle devra disposer des modèles C# (développés dans le Tp 1) et utilisera des données à partir de fichiers JSON pour peupler sa liste de médias, sa liste d'utilisateurs et les favoris des utilisateurs. Les fichiers JSON seront aussi là pour conserver les modifications faites à ces différentes listes par l'entremise de votre application. Les fichiers JSON assurent la persistance des données de votre application.

Votre application Web permettra entre autres :

- ✦ De s'inscrire et de se connecter comme **utilisateur** ou comme **administrateur**.
- ✦ D'afficher les médias disponibles.
- ✦ De choisir un média et d'obtenir de l'information plus complète sur ce média.
- ✦ D'ajouter ou retirer (avec confirmation) un média de la liste des favoris d'un utilisateur.
- ✦ D'afficher la liste de favoris.
- ✦ D'afficher la liste des utilisateurs (administrateur)

INSTRUCTIONS

Voici les instructions qu'il est obligatoire de respecter, vous avez la liberté pour l'apparence et pour certaines fonctions. Certaines actions sont présentées, mais seront développées dans le Tp3.

Données de l'application

Vous devez utiliser vos modèles développés dans le TP1 (il est fortement conseillé de les mettre à jour avant de commencer) pour enregistrer votre catalogue soit toutes les informations sur vos médias (nom, description...), les informations sur les utilisateurs, les favoris et les évaluations. Vos modèles auront besoin de méthodes pour enregistrer et lire les données. Comme écrit précédemment, vos fichiers JSON sont la persistance des données de votre application.

Médias, utilisateurs et favoris existants au premier démarrage de l'application Web

Au démarrage de votre application Web, il faut avoir dans les fichiers JSON (c'est vos données initiales):

- Au moins 10 médias complets (données et fichiers images...).
- ~~Un administrateur;~~
- Au moins deux utilisateurs et un administrateur [un administrateur n'est pas un utilisateur](#)
- Quelques médias en favoris pour chacun des 2 utilisateurs.

Utilisateurs et favoris existants après le premier démarrage de l'application

Tout utilisateur créé à l'exécution de l'application doit être ajouté au fichier JSON concerné.

Toute modification aux favoris d'un utilisateur doit affecter le fichier JSON concerné.

(**IMPORTANT** : gardez une copie propre de vos données initiales pour réinitialiser votre base de données lors de vos tests et lors de la remise pour la démonstration).

APPLICATION WEB À FAIRE

Page «Accueil» (non connecté)

- Fournir de l'information à propos de votre site :
 - Contexte de développement. [site en construction/ cadre du cours](#)
 - Choix du média.
 - Services disponibles pour le moment dans votre application Web.
- Placer une image en arrière-plan (utilisez des sites comme [Pixabay](#) et respectez les licences d'utilisation des images).
- Permettre à un utilisateur existant de se connecter. [et aussi admin](#)
- Permettre à un nouvel utilisateur de créer un compte (s'inscrire). [choix a l'inscription admin ou utilisateur](#)
[case a cocher si admin](#)

Page «Accueil» (connecté comme utilisateur)

Une fois connecté comme « utilisateur », vous arrivez dans la page d'accueil des utilisateurs. Cette page d'accueil offre une barre de navigation commune aux utilisateurs qui contient le « nom de l'application et un logo cliquable » permettant de revenir à l'accueil (connecté comme utilisateur), « Tous les médias » (accueil par défaut), « Mes favoris » et un bouton « Se déconnecter » portant le nom de l'utilisateur présentement connecté.

Page «Accueil» (connecté comme administrateur)

Une fois connecté comme « administrateur », vous arrivez dans la page d'accueil des administrateurs. Cette page d'accueil offre une barre de navigation qui contient le nom de l'application et un logo cliquable permettant de revenir à l'accueil (connecté comme admin), « Les utilisateurs » (accueil par défaut), [media non développé](#) « Les médias » et un bouton « Se déconnecter » portant le nom de l'administrateur présentement connecté avec une icône indiquant qu'il est admin).

Page «Tous les médias»

Affiche un aperçu de l'ensemble des médias disponibles (attention à la présentation). En cliquant sur un média en particulier, il est possible d'accéder à une page offrant [la fiche complète](#) du média cliqué.

Page « Mes favoris »

Affiche les médias favoris de l'utilisateur sous forme de fiche résumée. En cliquant sur un des médias en favoris, il doit être possible de consulter sa fiche complète.

Page « Les utilisateurs » (admin)

Affiche un aperçu de l'ensemble des utilisateurs et administrateurs enregistrés (attention à la présentation). Il est important d'indiquer si c'est un utilisateur ou un administrateur. Dans cette page il est seulement possible de supprimer un utilisateur ou un administrateur avec confirmation.

Page « Les médias » (admin)

Affiche seulement le texte suivant « Affichage de liste de tous les médias ici... ».

À LA RECHERCHE D'UNE SOLUTION 😊 pas de points, mais essay quand même, car intéressant +++ pour le futur À faire en dernier

Normalement il ne doit pas être possible d'accéder à certaines pages si je ne suis pas connecté ou selon les droits que j'ai. Un utilisateur n'est pas un administrateur et un administrateur n'est pas un utilisateur comme tel, pour le moment. Dans « .NET Core MVC », il existe une solution pour avoir le contrôle sur l'accès et la sécurité des utilisateurs, soit « d'ASP.NET Core Identity », mais un peu complexe et demande l'utilisation d'une BD. Mis à part cette solution que je ne veux pas que vous utilisiez (implémentiez), pouvez-vous me proposer une solution simple qui utilise nos données dans des fichiers JSON et qui pourrait au moins contrevenir aux accès non voulus des pages ou des sections du site Web?

AUTRES CONSIDÉRATIONS

- ♦ Votre application Web doit comporter :
 - Des dispositions "layout" différentes entre le mode non connecté, connecté utilisateur et connecté administrateur.
 - Au moins une vue partielle.
 - Au moins trois contrôleurs.
 - Un routage efficace selon le modèle MVC utilisé
 - Une mise en page qui exploite les styles de bases de Bootstrap.
 - Au moins un traitement en JavaScript et/ou jQuery.
 - Une présentation soignée et ergonomique est importante.
 - Pour aller plus loin sans être nécessaire, je vous invite à lire le document « Standard sur l'accessibilité des sites Web » du gouvernement du Québec.
- ♦ Portez attention :
 - À un affichage adaptable aux résolutions.
 - À la propreté de tout votre code.
 - À la bonne factorisation de votre code (duplication et réutilisation).
 - À la documentation de tout votre code.
- ♦ Vos modèles doivent être testés unitairement de manière automatisée et bien documentés.
Encore des tests...

ANALYSE

NOTE : Vous devez faire approuver votre document de conception avant de commencer la programmation.

- ♦ Une page titre standard.
- ♦ Une table des matières.
- ♦ Des titres et sous-titres pour les parties demandées.

- ✦ Une introduction (brève description de ce que vous devez faire dans ce travail (tp 2)).
- ✦ Une maquette de chacune des pages demandées avec leur route comme titre (en en-tête). Pour chacune des pages, faire un dessin propre qui représente approximativement ce que votre page finale aura l'air. Attention au niveau de détails. Je ne veux pas une sortie programmée. La programmation vient après l'analyse. Si sur papier, prendre une photo du dessin de la page pour mettre dans le document.

À lire « Créer les maquettes de son site web » : <https://www.lafabriquedunet.fr/creer-site-internet/creer-maquettes-site-web/>

- ✦ Un copier-coller du contenu des fichiers JSON de vos données de médias, d'utilisateurs de départ (au premier démarrage de l'application Web). Formatez les données pour quelles soient bien présentées.
- ✦ Une description des traitements ou effets prévus en JavaScript ou JQuery avant de les programmer. Pas de code Validation
- ✦ Donnez deux avantages et désavantages d'inclure Bootstrap dans le projet.

ÉQUIPE ET REMISE

- Le travail se réalise en équipe de 2 ou 3 sur environ 4 semaines.
- Le dépôt de **chacune des** parties de ce travail se fait sur le réseau dans les dossiers **indiqués ci-après**.
- Partie 1 (Document Word d'analyse selon la demande)
 - **Le 26 et 27 mars** pendant votre période de laboratoire.
 - Dossier : **Q:\!Depot\JGoulet\420-4B6-H24\Tp 1 - Partie1 (Document d'analyse)\Gr 1 ou 2**
 - Nom du fichier à remettre : **Noms_des_équipiers-Tp2-H24-Analyse.docx**
- Partie 2 (Code et démo)
 - **Le 12 avril** selon votre période de laboratoire.
 - Le dossier de solution seulement (si vous le zippez, faites-le seulement avec l'outil Windows)
 - Dossier : **Q:\!Depot\JGoulet\420-4B6-H24\Tp 2 - Partie2 (Code final)\Gr 1 ou 2**
 - Nom du fichier ou dossier à remettre : **Noms_Des_Équipiers -Tp2-H24-Code**

NOTE : Au moment de la démonstration de votre travail, vous aurez à présenter le bon fonctionnement de toutes les pages demandées et à démontrer la robustesse de votre code, à répondre aux questions de l'enseignant à propos de votre code et de sa concordance avec votre document d'analyse et la demande, à vos données JSON.

Il est aussi possible que l'on revienne sur le code du Tp 1, les classes du modèle et les tests unitaires.

Tous les membres de l'équipe doivent être là, préparez-vous en conséquence s.v.p.