

IFT3225 – Technologie de l'internet

Axel Seguin

Projet de session : Application web en tuile.

Mise en contexte

Dans ce projet, vous travaillerez en équipe de deux personnes. Toutes vos pages devront pouvoir passer dans un validateur HTML 5 sans renvoyer d'erreur.

Il s'agit de développer une application web présentant des tuiles sur la page d'accueil. Chaque tuile représente un élément de l'application. Votre application peut être un système de gestion de tâches, de recettes de cuisines, de voyages, personnages historiques, jeux vidéo, bibliothèque, vidéothèque ou autre. Ce système de gestion aux utilisateurs de créer, modifier et supprimer des tuiles (grâce à une API REST), et de les partager en temps réel avec d'autres utilisateurs connectés. Les fonctionnalités principales du projet sont décrites ci-dessous.

Description du projet

Vous devrez utiliser **obligatoirement** les technologies suivantes dans votre travail :

- Backend : PHP ou NodeJS
- MySQL ou MongoDB
- Requêtes asynchrone : fetch avec des promesses
- HTML5
- CSS3
- REST

Vos pages web devront également être liées à des feuilles de style. Nous vous donnons la possibilité de choisir toutes les bibliothèques et *frameworks* CSS que vous désirez, même celles que nous n'aurions pas abordées durant les séances théoriques et pratiques.

Si jamais vous vous inspirez de morceaux de code disponibles en ligne, n'oubliez pas d'indiquer sa source dans l'entête de vos fichiers pour vous protéger du plagiat.

Vous pouvez travailler en local sur vos ordinateurs personnels pour la réalisation de votre projet, comme nous vous l'avons présenté en classe. Pensez à l'utilisation de phpMyAdmin (pour mysql) afin de faciliter la conception ainsi que la gestion de vos bases de données. N'oubliez pas d'effectuer des sauvegardes et copies de sécurité de votre travail !

Composition du projet (21 points : 3.5 points par fonction)

Interface utilisateur : Créez une interface utilisateur attrayante et conviviale où les utilisateurs pourront voir les différentes tuiles, en ajouter, les modifier et les supprimer.

Un maximum de 15 tâches sera affiché par page, pensez à créer un système de pagination s'il y a plus de 15 tâches.

Vous pouvez vous inspirer des applications suivantes si vous manquez d'inspiration : <https://trello.com/fr> et <https://www.atlassian.com/fr/software/jira>

Affichage dynamique en fonction de la taille de l'écran (CSS): Le nombre de tuiles affiché sur la largeur de l'écran ne doit jamais obliger l'utilisateur à se promener de droite à gauche, le nombre de tâches affiché en largeur doit s'adapter à la largeur de l'écran.

Authentification utilisateur : Mettez en place un système d'authentification pour permettre aux utilisateurs de s'inscrire, de se connecter et de gérer leurs tuiles personnelles. Pensez à faire en sorte que les mots de passe n'apparaissent pas en brut dans vos bases de données ! Vous pouvez utiliser une méthode de hash. Pour le système d'inscription, assurez-vous que l'adresse électronique saisie soit correcte uniquement syntaxiquement. Pas besoin de vérifier si cette dernière existe véritablement. Il peut y avoir plusieurs comptes administrateurs.

Gestion des tâches (REST): Permettez aux utilisateurs de lister les tuiles, d'en afficher une ou d'en supprimer une. Vous utiliserez JSON pour la communication entre le serveur et le client pour l'API REST.

Partage en temps réel (requêtes asynchrones): Utilisez des requêtes asynchrones pour permettre la mise à jour en temps réel de la liste des tuiles. Lorsqu'un utilisateur ajoute, modifie ou supprime une tâche, ces changements doivent être reflétés instantanément pour tous les utilisateurs connectés.

Recherche et filtrage : Ajoutez des fonctionnalités de recherche et de filtrage pour permettre aux utilisateurs de trouver rapidement des tuiles spécifiques en fonction de différents critères (par exemple, par utilisateur, par catégorie, par date, par description, titre, etc.).

Stockage de données : Utilisez une base de données pour stocker les informations sur les tuiles, les utilisateurs et les catégories.

Sauvegarde et gestion des versions : l'utilisation de gitHub est vivement recommandée pour sauvegarder votre code et les différentes versions de votre code. Utilisez un dépôt privé. (pas de points pour cette section).

Résumé :

À l'aide d'un formulaire de création de tuiles, l'utilisateur peut ajouter une tuile, le formulaire fera les validations nécessaires.

Une tuile doit contenir un titre, une date, éventuellement une catégorie ainsi qu'une description.

L'administrateur peut tout modifier de toutes les tuiles. Certains champs du formulaire de création de tâches doivent être obligatoires et il doit y avoir une validation des erreurs.

La page principale contient les tuiles (maximum 15 par page). Chacune des tuiles doit contenir un bouton de modification et un bouton de suppression. Le bouton de modification ouvre un formulaire pour modifier la tuile. En cliquant sur valider, la modification est envoyée au serveur à l'aide d'une requête asynchrone. Le bouton de suppression entraîne une demande de confirmation.

Une API REST permet de lister toutes les tuiles au format JSON, d'en lister une seule ou d'en supprimer une.

À remettre :

Vous remettrez tous les fichiers nécessaires au fonctionnement de votre application : fichiers html, php, javascript ainsi que les fichiers nécessaires à la création de votre base de données. Remettez aussi une documentation expliquant comment déployer votre application. Il vous est vivement recommandé de tester cette documentation car s'il n'est pas possible d'installer votre application, 35% des points seront perdus. La documentation expliquera aussi les routes de l'API REST ainsi que la répartition du travail entre les deux membres de l'équipe.

Si un admin est créé pour votre application et qu'il a un mot de passe prédéfini dans l'application, utilisez ift3225 comme mot de passe.

ATTENTION :

Comme pour le projet 1, aucun retard n'est accepté pour aucune raison. Aucune réponse à des questions à partir de 72h avant la date de remise du projet.