Cours 420-5C6-LI **Applications web II** **Automne 2025** **Cégep Limoilou** **Département d'informatique** **Professeur : Martin Simoneau**	TP 1 v1 **SPA sans backend**

Objectifs

- Créer un frontend SPA en utilisant les composants *React* et le *js*

Remise:

- Le travail sera remis sur Léa à la date indiquée.

Contexte : (vous devez vous assurer de bien maîtriser les éléments suivants):

1) Ce TP est le premier de 3 TPs qui **impliqueront le même projet**.
   a) TP1 frontend *SPA* sans *backend*
   b) TP2 backend avec un seul serveur
   c) TP3 backend en microservices - déploiement *Docker-compose* + *message queue*
2) Le projet consiste à faire un site de nouvelles. Chaque équipe doit choisir un sujet de nouvelle (sport, actualité, sciences, musique, technologie, automobile…).
3) Les types se font en équipe de 2 ou 3 déterminées par l'enseignant. Certaines équipes pourront être refaites s'il y a des écarts appréciables dans la qualité des travaux.
4) IMPORTANT :
   a) La présence au laboratoire est essentielle pendant les périodes de travail sur les tps. Si un étudiant ne se présente pas à une séance de travail sur un tp sans une justification valable, il sera retiré de l'équipe et devra faire le travail seul.
   b) Vous devez utiliser Jira/Github et chaque personne doit commiter son propre code. Le crédit sera automatiquement donné à celui qui fait le commit.
   c) Vous devez maîtriser tout le code que vous allez remettre. Dans le doute, le professeur pourra vous convoquer en entrevue pour le valider. Vous devrez alors démontrer une compétence complète sur toutes les technologies utilisées dans votre code qui n'ont pas été vues durant le cours.

À faire

1) Fonctionnalités (les scripts doivent emmagasiner toutes les données dans le ***local Storage***)
   a) Nouvelle :
      i) **Création** : Une nouvelle doit avoir **au moins** :
         (1) Une date
         (2) Un titre
         (3) Un numéro universel de référence
         (4) Une image
         (5) Un texte
         (6) Un résumé
      ii) **Effacement** : on doit pouvoir retirer une nouvelle. Cela doit également effacer tous les critères de recherche qui lui sont associés.
      iii) **Modification** : On doit pouvoir modifier les informations d'une nouvelle (sauf le numéro universel!)
      iv) Votre site doit contenir au moins 10 nouvelles en partant. Vous pouvez faire générer des nouvelles par un IA.
   b) Utilisateur :
      i) Pour les équipes de 3 personnes :
         (1) On doit pouvoir saisir l'utilisateur (nom, mot de passe, date d'inscription, date de naissance)

(2) Au minimum, il doit y avoir un utilisateur invité et un utilisateur journaliste. Pour obtenir un meilleur résultat (85+), vous devez également gérer un admin et un client.
(3) On doit pouvoir modifier et effacer un utilisateur à partir de votre UI.

ii) Pour les équipes de 2 personnes :
(1) Les utilisateurs sont hardcodés.
(2) Il y a un admin et 3 journalistes.
(3) Aucun mot de passe n'est requis. Un simple menu permet de changer l'utilisateur en cours.

c) **Critère de Recherche** : Un critère de recherche est un élément qui permet de sélectionner une ou plusieurs nouvelles. Vous devez décider de sa complexité. Plus votre critère de recherche sera intuitif et complet, meilleure sera votre note.
i) **Création** :
(1) Le numéro de référence de la personne (pour faire le lien avec la personne)
(2) La date de saisie du critère
(3) Les éléments que vous jugez importants
(4) La valeur de la critique (moyenne des sous-critiques : entre 0 et 100);
ii) **Effacement** : Le critère de recherche peut être effacé.
iii) **Modification** : le critère de recherche ne peut pas être modifié, il faut l'effacer et en refaire une nouvelle.
iv) Tous les critères saisis demeurent dans le SPA, et à chaque changement dans les nouvelles (saisie, effacement ou modification) le critère doit être réévalué.
v) Résultats : afficher en tout temps tous les résumés et titres de nouvelle qui respectent chaque critère emmagasiné.

2) UI *React*
a) Le site doit présenter :
i) Section de nouvelles (permettant toutes les fonctionnalités demandées)
ii) Section de critères de recherche (permettant toutes les fonctionnalités demandées)
iii) Section pour la gestion des utilisateurs (équipes de 3 personnes)
iv) Page de statistiques
(1) Le nombre de nouvelles.
(2) La taille de la nouvelle la plus longue et la plus courte.
(3) La taille moyenne des nouvelles (en nombre de caractères).
(4) La date de la plus ancienne nouvelle et celle de la plus récente.
(5) Le nombre de nouvelles correspondant à chaque critère de recherche.

3) Formatage CSS
a) Formatage consistant dans l'ensemble du site
b) Utilisation de classes de style par programmation
4) Chaque personne doit ajouter une fonctionnalité de son cru. (important pour ceux qui veulent se démarquer (80+))

Exigences

5) Le travail est fait en équipe sur *Jira* ou *Github*
6) Chaque tâche doit être accompagnée d'une échéance claire. Si l'échéance n'est pas respectée, un autre membre de l'équipe peut s'octroyer la tâche et la faire à la place de celui qui est en faute.
7) Vous devez maîtriser tout le code que vous allez remettre. Dans le doute, le professeur pourra vous convoquer en entrevue pour le valider. Vous devrez alors démontrer une compétence complète sur toutes les technologies utilisées dans votre code qui n'ont pas été vues durant le cours.
8) Toutes les fonctionnalités de l'application (qui serait normalement dans le backend) doivent être faites dans des scripts en pure js (sans *React*) et placées dans un dossier ***scripts***.
9) Les composants doivent être pensés pour être réutilisés. Ce sera utile/nécessaire lorsque vous fusionnez les projets dans le TP2.
10) Les scripts js sont bien différentiés des scripts jsx des composants React.

11) **Vous devez utiliser Jira/Github et chaque personne doit commiter son propre code. Le crédit sera automatiquement donné à celui qui fait le commit.**
12) Chaque commit doit être associé à un *numéro de ticklet*/issue sur *Github* ou *Jira*.
13) Exigences minimales par personne :
    a) 3 composants dynamiques/interactifs *React* (avec au moins 2 états)
    b) La gestion soit des nouvelles, soit des utilisateurs, soit des critères de recherche dans le script js
    c) Gestion de *CSS*
    d) Établissement et respect des échéanciers.
    e) Ne pas faire le travail d'un collègue s'il n'y a pas de bris d'échéance.
14) Exigences pour l'équipe.
    a) Utilisation d'un *context React*.
    b) Il y a un style général pour l'ensemble du site.
    c) Les fonctionnalités sont bien intégrées.
    d) Le travail a été réparti convenablement.
    e) Établissement et respect des échéanciers.
15) Rapport contenant :
    a) Description des fonctionnalités par personne
    b)

Critères d'évaluation

16) Pour chaque critère :
    a) **90+** l'étudiant maîtrise tous les concepts avancés du cours et ne fait pas d'erreurs. L'UI est fait avec MUI.
        i) L'équipe travaille de façon continue et maîtrise les outils de gestion de projet (git/github)
        ii) Le travail fait preuve d'une grande créativité et d'une certaine originalité.
        iii) Pas de message d'erreur évitable en console pendant l'exécution.
        iv) Le code est parfaitement découpé et facilement réutilisable.
        v) Le site est original. Ce n'est pas une simple adaptation des formatifs.
        vi) L'étudiant démontre une maîtrise de tous les concepts avancés
            (1) partage de références.
            (2) Positionnement, partage et choix des états impeccable
            (3) Composition de composant (décoration).
            (4) Utilisation pertinente et efficace de Context .
        vii) L'apparence du site est impeccable.
            (1) L'espace est utilisé intelligemment lors du redimensionnement.
            (2) La palette de couleur choisie est évidente et bien respectée (3 couleurs : primaire, secondaire et accent).
            (3) L'utilisation du site est évidente et naturelle.
                (a) Les éléments avec lesquels on peut interagir sont mis en évidence subtilement (ex : hover).
                (b) Les informations sont bien regroupées et claires à consulter.
    b) 80+
        i) Très peu de messages en console pendant l'exécution.
        ii) Le code est clairement découpé et facilement réutilisable.
        iii) Le travail fait preuve d'une bonne créativité.
        iv) L'étudiant démontre une bonne compréhension de la majeure partie des concepts vus en classe :
            (1) Utilisation appropriée des références.
            (2) Les états (positionnement, utilisation, propagation et choix) sont fonctionnels.
            (3) Utilisation pertinente et efficace de *Context*.
            (4) Composants enfants
        v) L'apparence du site est très bonne.
            (1) L'espace est utilisé intelligemment lors du redimensionnement.

(2) La palette de couleur choisie est évidente et bien respectée (3 couleurs : primaire, secondaire et accent).
(3) L'utilisation du site est évidente et naturelle.
(a) Les éléments avec lesquels on peut interagir sont mis en évidence subtilement (ex : *hover*)
(b) Les informations sont bien regroupées et claires à consulter

c) 70 L'étudiant conçoit une application qui fonctionne correctement, mais qui n'utilise surtout les éléments de base ou maladroitement les éléments plus avancés. Il peut y avoir des bogues mineurs ou des imperfections dans l'interface.
i) Aucun message d'erreur important en console.
ii) Le travail est assez générique et convenu.

d) 60+ Les majorités des fonctionnalités sont là et elles sont utilisables via le SPA.

FIN