

<p align="center"> <b>Cours 420-5C6-LI</b>  <b>Applications web II</b>  <b>Automne 2024</b>  <b>Cégep Limoilou</b>  <b>Département d'informatique</b> </p> <p> <b>Professeur :            Martin Simoneau</b> </p>	<p align="center"> <b>TP 1</b>   <b>SPA sans backend</b> </p>
--	---

## 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 TP qui impliqueront le même projet.
  - a) TP1 frontend *SPA sans backend*
  - b) TP2 plusieurs *web services* coordonnés avec *Eureka*
  - c) TP3 déploiement *Docker* + utilisation des *websocket / message queue*
- 2) Le projet consiste à faire un site de critique (review) de produits. Chaque équipe doit choisir un produit.
- 3) Le TP1 se fait en équipe de 2, mais pour les TP 2 et 3 les équipes seront combinées pour faire des équipes de 4. Les équipes seront faites par le professeur en fonction des TP1 et des résultats généraux. Certaines équipes pourront être refaites s'il y a des écarts appréciables dans la qualité des travaux.

## À faire

- 1) Fonctionnalités (scripts du backend qui tournent dans le frontend utilise le localStorage)
  - a) Produit
    - i) **Création** : le produit doit avoir au moins
      - (1) Une date de création/entrée en service/mise en marché
      - (2) Un nom de produit
      - (3) Un fabricant [Région d'origine](#)
      - (4) Un numéro universel de référence
      - (5) Une sous-catégorie
      - (6) Un prix suggéré.
    - ii) **Effacement** : on doit pouvoir retirer un produit. Cela doit également effacer toutes les critiques qui lui sont associées.
    - iii) **Modification** : On doit pouvoir modifier les informations du produit
  - b) Critique
    - i) **Création** :
      - (1) Le numéro de référence du produit (pour faire le lien avec le produit)
      - (2) La date de la critique
      - (3) Au moins 3 sous-critères d'évaluation
      - (4) La valeur de la critique (moyenne des sous-critiques : entre 0 et 100);
    - ii) **Effacement** : La critique peut être effacée, cela n'a pas d'incidence sur le produit critiqué
    - iii) **Modification** : la critique ne peut pas être modifiée, il faut l'effacer et en refaire une nouvelle.
- 2) UI React
  - a) Le site doit présenter :
    - i) Section de critique (permettant toutes les fonctionnalités demandées)
    - ii) Section de produit (permettant toutes les fonctionnalités demandées)
    - iii) Page de statistique
      - (1) Tous les produits dans l'ordre d'appréciation
      - (2) La cote moyenne pour chaque catégorie

- iv)
- 3) Formatage CSS
  - a) Formatage consistant dans l'ensemble du site
  - b) Utilisation de class 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) Vous pouvez utiliser les IA, mais vous devez maîtriser tout le code que vous allez remettre. Dans le doute, le professeur pourra vous convoquer en entrevue pour le valider.
- 7) 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
- 8) Les composants doivent être pensés pour être réutilisés. Ce sera utile/nécessaire lorsque vous fusionnez les projets dans le TP2.
- 9) Les script js sont bien différenciés des script jsx des composants React.
- 10) Vous devez utiliser Jira/Github et chaque personne doit commiter son propre code. Le crédit sera automatiquement donné à celui qui fait le commit**
- 11) Chaque commit doit être associé à un *issue* sur Github.()
- 12) Exigences minimales par personne :
  - a) Une référence utile
  - b) 3 composants React (avec au moins 2 états)
  - c) La gestion soit du produit soit de la critique dans le script js
  - d) Gestion de CSS
  - e) Utilisation de context ou redux.
- 13) Rapport contenant :
  - a) Description des fonctionnalités
  - b) Faire le lien entre la matière du cours (exigences minimales par personne) et le travail remis

## Critères d'évaluation

- 14) Pour chaque critère :
  - a) **90+** l'étudiant maîtrise tous les concepts avancés du cours et ne fait pas d'erreurs.
    - 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 en console pendant l'exécution
    - iv) Le code est parfaitement découpé et facilement réutilisable
    - v) L'étudiant démontre une maîtrise de tous les concepts avancés
      - (1) partage de références
      - (2) Positionnement et choix des états
      - (3) Composition de composant (décoration)
      - (4) Utilisation pertinente et efficace de Context ou Redux
    - vi) 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 message 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 *d'imperativeHandle* pour les références
- (2) Positionnement et choix des états
- (3) Utilisation pertinente et efficace de Context ou Redux
- (4) Composants enfants
- v) 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
- 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