# Objectif

**L’objectif** de ce travail est de **valider l’attente des sous-objectifs** dans le cadre du cours :

1. Effectuer le développement d’applications Web transactionnelles. (Compétence 00SU)
   * 6 - Programmer la logique applicative du côté client.
   * 7 - Contrôler la qualité de l’application.
   * 9 - Rédiger la documentation.

# Mise en situation

Vous êtes **programmeur Web** et avez à faire la programmation d’un **gestionnaire de produit** à l’aide de **JavaScript** pour en maximiser l’interactivité.

Vous devez **réaliser la programmation** du gestionnaire au complet selon les spécifications des **récits utilisateurs** et en respectant **les consignes.**

Vous devez choisir un **produit** par **vous-même**. Demeurez dans le *« politically correct* » de niveau **collégial**.

## Détails supplémentaires

* Respecter les [bonnes pratiques de code](http://appweb.progwmj.ca/documentations/bonnes-pratiques/code)
* Respecter les [bonnes pratiques d’interface utilisateur](https://appweb.progwmj.ca/documentations/bonnes-pratiques/interface-utilisateur)
* Respecter les [bonnes pratiques de couleur UI](https://appweb.progwmj.ca/documentations/bonnes-pratiques/couleurs-ui)
* Respecter les [normes et standards](https://appweb.progwmj.ca/documentations/normes)
* Vous devez respecter les **récits utilisateurs fournis**
* Vous devez respecter les **consignes énoncées** plus bas.
* N’oubliez pas de remplir la **grille d’auto-évaluation Excel**!

# Récits utilisateurs

|  |  |
| --- | --- |
| En tant qu’utilisateur... | Précisions |
| 1. Je peux ajouter un nouveau produit. | Remplir un formulaire et cliquer sur "Ajouter". Validation des champs obligatoires. |
| 1. Je peux modifier un produit existant. | Bouton "Modifier" qui préremplit un nouveau formulaire avec les données du produit sélectionné. |
| 1. Je peux dupliquer un produit existant pour en créer un nouveau rapidement. | Bouton "Dupliquer" qui préremplit un nouveau formulaire avec les données du produit sélectionné. |
| 1. Je peux supprimer un produit. | Confirmer la suppression après avoir cliqué sur le bouton "Supprimer". |
| 1. Je peux consulter la liste des produits. | Affichage sous forme de tableau avec affichage conditionnel |
| 1. Je peux rechercher un produit par son nom. | Champ de recherche dynamique avec mise à jour instantanée des résultats. |
| 1. Je peux voir les détails d’un produit. | Cliquer sur un produit affiche ses détails (nom, description, prix, stock). |
| 1. Je peux voir les quantités disponibles d’un produit. | Affichage en temps réel des quantités avec indicateur visuel (couleurs). |
| 1. Je peux être averti lorsqu'un produit est en rupture de stock. | Notification (message) lorsqu'un produit atteint un stock critique => 0. |
| 1. Je peux exporter la liste des produits en CSV | Bouton "Exporter" permettant de télécharger la liste dans le format CSV. |

# 

# Méthodologie

1. Prendre **connaissance** de tous les **récits utilisateurs**.
2. Faites l’installation complète d’un **nouveau projet Vue**
   * **IMPORTANT :** *Ne prenez pas un ancien exercice ou du code partagé avec un autre étudiant.*
3. Faites l’installation de **Bootstrap** ou de **Tailwind** de la **manière vue** en classe dans une **application Vue.**
   * **IMPORTANT** : *Ne prenez pas un framework autre ou des composants du style* ***Vuetify****.*
4. Vous devez faire le découpage de votre application en **composants**, en lien avec les fonctionnalités et l’interface.
   * **IMPORTANT** : Il devrait au **minimum** autant que de **parties fonctionnelles** (*recherche = composant)*
5. Vous devez faire l’inclusion d’au moins une image qui représente votre entreprise (logo, image IA, etc.) dans votre application. Ex. : *image pour votre application au complet.*
6. Vous devez développer chaque **récit utilisateur** et vous assurer de son **bon fonctionnement** ET de son **apparence**.
7. Dans un fichier **README.md,** vous devez identifier :
   * Description du projet
   * Toute mention et description d’utilisation d’une IA.
   * URL du repo GitHub
   * URL du site déployé
8. Vous devez faire un repo sur **GitHub** de votre projet
9. Vous devez déployer votre application sur **GitHub pages**
   * **IMPORTANT:** le répertoire doit être nommé « **appweb-trpr01 ». Ex. :** [**https://user.github.io/appweb-trpr01/**](https://user.github.io/appweb-trpr01/)
10. Remettre votre **projet complet** sur **Léa.**

# Consignes

* Travail pratique **individuel (20% de la note)**
* Réalisé en-**dehors** **du** **cours et temps en classe**
* Durée approximative de **16-24 heures**.
* Le fichier de remise est un fichier compressé nommé : **appweb\_trpr01\_votrenom.zip.**
  + Les **fichiers complets** de votre site
  + **IMPORTANT :** supprimer le dossier **node\_modules**
  + La **grille d’évaluation** au format **Excel et le README.md** sontà la **racine du site**
* Remis sur **Léa**à la date mentionnée

## Plagiat

Prendre connaissance du [document officiel du Cégep concernant le plagiat.](https://www.csfoy.ca/fileadmin/documents/notre_cegep/publications_officielles/PlagiatFraudeTricherie.pdf)

* **Reproduire partiellement ou totalement un travail sans citer la source.**
* **Remettre un travail qui ne provient pas de soi, en tout ou en partie. => travail INDIVIDUEL**
* **Générer du code sans citer la source** **est considérée comme du plagiat.**

**En cas de doute, je peux vous convoquer à mon bureau pour une entrevue de vérification de votre code.**

### ***Conséquences en cas de plagiat ou de tricherie***

**En plus des conséquences normales du Cégep, je me réserve le droit d’empêcher l’étudiant de faire les TP02 et TP03 en équipe.**

# Critères d’évaluation (100 %)

## Qualité du code (30 %)

* **Utilisation correcte de Vue.js :** Application des concepts clés de Vue.js tels que les composants, les directives, et le système réactif. **IMPORTANT :** Vous ne devez pas utiliser de concepts qui ne sont pas encore vus en classe (exemple les stores). => Vérifiez avec moi au cas.
* **Utilisation correcte de TypeScript :** Types bien définis pour les props, les états des composants, et les objets. Utilisation des avantages de TypeScript pour augmenter la sécurité et la maintenabilité du code.
* **Bonnes pratiques de programmation et code propre.** Exemples: noms significatifs, évitez la répétition, clarté et simplicité, respect des normes, etc.
* **Structure du projet :** Organisation logique des fichiers et des dossiers, utilisation appropriée des modules et des composants.
* **Gestion des états :** bonne utilisation des états locaux du composant et réactivité.

# **Fonctionnalités (50%)**

* **Implémentation des fonctionnalités requises** : L'application doit implémenter toutes les fonctionnalités demandées dans les récits utilisateur. (10 récits \* 4 = 40%)
* **Bonnes pratiques UI**: L'application affiche correctement les interfaces utilisateurs, par exemple, affichage d’erreurs, validation, couleurs. (10 récits \* 1 = 10%)

## Contrôle de qualité (10 %)

* **Bogues et performances :** Aucun bogue, aucune erreur d'exécution et aucun problème de performance.
* **Qualité de l'interface** : Aucune erreur d’affichage. Intégration de Bootstrap ou Tailwind, utilisation de bons composants Bootstrap ou Tailwind.

# Déploiement et documentation (10%)

* **Repo correctement créé sur GitHub :** 
  + Le fichier README.md est correctement rempli avec les informations demandées.
  + Le repo est créé avec le dernier commit de votre projet le plus récent.
* **Application déployée correctement sur GitHub Pages :**
  + L’application est correctement déployée (build)
  + La bonne URL demandée est utilisée (appweb-trpr02)

## Respect des consignes (-10 %)

* Les consignes du travail sont respectées

***La grille d’évaluation est fournie sur Léa.***