

# Cahier des charges - Projet SCRIPTS CLIENTS

## Contact Manager - VUE CLI

### Contexte

Vous allez développer une application de gestion de contacts en utilisant **Vue CLI**, **Pinia** pour la gestion des états, et le **Vue Router** pour la navigation. L'objectif est de créer une application interactive et modulaire permettant de gérer des contacts. Le projet est découpé en étapes (*issues* GitLab), certaines étant obligatoires pour valider le projet (atteindre 10/20), et d'autres optionnelles pour obtenir des points supplémentaires. Il me semble plus pratique de suivre ces issues via un système de versionning mais ce n'est pas obligatoire.

### 1. Obligatoire (Seuil de réussite - 10/20)

Vous devez implémenter les fonctionnalités suivantes :

#### 1.1 Initialisation du projet (Issue: Initialize Vue CLI Project)

- Créer une application Vue CLI.
- Configurer **Pinia**.
- Configurer **Vue Router** avec deux routes principales :
  - Dashboard (liste des contacts).
  - AddContact (formulaire d'ajout et modification).

#### 1.2 Récupération et affichage des contacts (Issue: Display contacts from localStorage)

- Stocker des contacts fictifs dans le **localStorage** au démarrage de l'application.
- Utiliser **Pinia** pour gérer l'état des contacts.
- Afficher les contacts sous forme de liste dans la page Dashboard.

#### 1.3 Ajout d'un contact (Issue: Add a new contact)

- Implémenter un formulaire sur la page AddContact pour ajouter un contact.
- Les champs requis sont : name, email, phone.
- Utiliser **Pinia** pour ajouter le contact et mettre à jour l'état global.

#### 1.4 Watcher profond sur le localStorage (Issue: Implement deep watcher for localStorage)

- Mettre en place un watcher profond dans **Pinia** pour que toute modification de l'état des contacts soit synchronisée avec le **localStorage**.

#### 1.5 Modification d'un contact (Issue: Edit a contact)

- Permettre la modification d'un contact via le formulaire AddContact.
- Pré-remplir le formulaire avec les données du contact sélectionné.

- Rediriger vers la liste quand on valide le formulaire

### **1.6 Suppression d'un contact (Issue: Delete a contact)**

- Activer un bouton pour supprimer un contact sur la page Dashboard.
- Utiliser **Pinia** pour supprimer le contact et mettre à jour l'état global.

## **2. Optionnel (Pour améliorer la note - au-delà de 10/20)**

### **2.1 Calcul dynamique du nombre de contacts (Issue: Display computed contact count)**

- Ajouter un compteur sur la page Dashboard qui affiche dynamiquement le nombre de contacts présents dans la liste.

### **2.2 Filtrage des contacts (Issue: Implement search/filter functionality)**

- Filtrer les contacts affichés en fonction de l'entrée texte (par exemple, sur le nom ou l'email).

### **2.3 Notifications utilisateur (Issue: Add toastify for user feedback)**

- Implémenter des notifications via **vue3-toastify** pour informer l'utilisateur des actions comme :
  - Contact ajouté avec succès.
  - Contact supprimé.
  - Erreur de validation.

### **2.4 Validation du formulaire (Issue: Implement form validation)**

- Ajouter une validation des champs du formulaire :
  - Le champ name ne doit pas être vide.
  - Le champ email doit contenir un email valide.
  - Le champ phone doit respecter un format numérique.

## **3. Critères d'évaluation: Voir grille sur le campus**