### **TP Component Life Cycle**

### **Auto-sauvegarde des données d'un formulaire avec useEffect**

Dans cet exercice, vous allez créer un formulaire simple qui sauvegarde automatiquement les données saisies par l'utilisateur dans le **localStorage** à chaque modification. Vous apprendrez à restaurer ces données lors du chargement du composant et à comprendre comment utiliser efficacement useEffect sans avoir besoin d'une fonction de nettoyage.

#### **Objectifs :**

* Sauvegarder les données du formulaire dans **localStorage** à chaque modification.
* Restaurer les données sauvegardées lors du montage du composant.
* Utiliser useEffect pour automatiser la gestion de la persistance des données.

#### **Instructions :**

1. Créez un composant fonctionnel nommé AutoSaveForm.
2. Utilisez la fonction   
   localStorage.setItem("formData", JSON.stringify(formData)); pour sauvegarder les données du formulaire.
3. Utilisez la fonction

const savedFormData = localStorage.getItem("formData");

pour récupérer les données sauvegardées lors du montage du composant.

#### **Étapes détaillées :**

1. Au montage du composant (lorsqu'il est chargé dans la page), vous allez utiliser useEffect pour vérifier si des données précédemment saisies sont déjà sauvegardées dans **localStorage** avec localStorage.getItem("formData");.   
   Si des données existent, elles seront restaurées dans le formulaire.
2. À chaque modification des champs du formulaire, vous utiliserez useState pour mettre à jour l'état du formulaire et localStorage.setItem("formData", JSON.stringify(formData)); pour sauvegarder automatiquement ces nouvelles données dans **localStorage**.