

# Exercice React - Hooks de base

Dans cet exercice, vous allez créer une application CRUD simple dont voici un exemple à l'adresse suivante : <https://hooks-react-project.vercel.app/>. Vous êtes libre sur le style de l'application mais, vous devrez forcément pouvoir ajouter, mettre à jour ou supprimer des utilisateurs. Vous n'utiliserez aucune classe React, et à la place, vous utiliserez des Hooks d'état et d'effet, **useState** et **useEffect** sur des composants fonctionnels. Voici un petit guide pour démarrer :

Tout d'abord, vous commencez par installer le projet avec [ViteJS](#) :

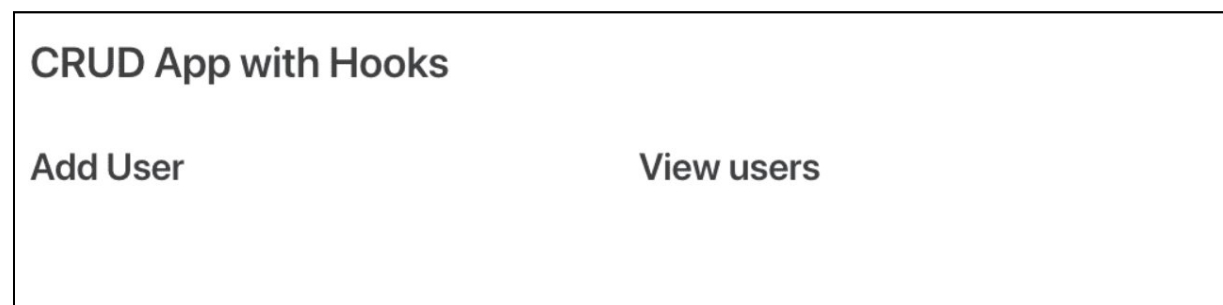
```
npm create vite@latest hooks-react  
Ou  
yarn create vite hooks-react  
Ou  
pnpm create vite hooks-react
```

Choisissez React sans TypeScript pour le moment

```
cd hooks-react
```

puis : `npm start`

Ensuite, vous pouvez configurer le projet en effaçant tous les fichiers du boilerplate dont vous n'avez pas besoin. Et tentez de modifier **App.js** pour que votre application ressemble à ceci :



Après, vous pouvez vous occuper de configurer la vue avec des exemples de données et une table pour les afficher, vous pouvez créer un nouveau dossier dans **/src** nommé **/tables** et un fichier **UserTable.js** par exemple. Créez le squelette de votre table, voici à quoi peu ressembler le projet à ce stade :

## CRUD App with Hooks

### Add user

### View users

Name	Username	Actions	
Name info	Username info	Edit	Delete

Maintenant, vous allez pouvoir retourner dans **App.js** pour créer des fausses données et utiliser le Hook **useState**.

Pour ce qui est du mappage des données utilisateurs et l'envoi ainsi que l'affichage des données de chaque utilisateur vous pourrez utiliser les Props comme vu précédemment.

A ce stage, les boutons Edit et Delete ne fonctionnent pas encore, mais logiquement, vous pouvez mettre en place la fonctionnalité d'ajout d'un nouvel utilisateur.

Vous êtes libre pour la suite de l'exercice, le but étant d'obtenir une app qui ressemble à la démo (d'un point de vue fonctionnelle, pas nécessairement visuelle) vous publierez votre résultat sur GitHub.

### Conseils / indices :

- Vous aurez besoin de créer un formulaire pour ajouter un nouvel user.
- Si vous ne comprenez pas ce qui se passe, utilisez des `console.log()`.
- Pensez à vérifier les changements éventuels d'événements (`onChange`).
- N'oubliez pas de permettre d'implémenter la modification et la suppression des utilisateurs → [CRUD](#)
- Vérifiez que votre app fonctionne bien, notamment lorsque vous modifiez un utilisateur, essayez d'en modifier un autre après, si rien ne se passe c'est peut-être que le composant est déjà « ouvert », pensez au `useEffect` (pas forcément nécessaire à la réussite du projet mais peut être une solution) !