

Fiche du cours

65 h

Titre :

Web220 - React Frontend (Débutant → Intermédiaire)

Description :

Ce cours intensif vous apprendra à utiliser React 18+, la bibliothèque JavaScript frontend la plus populaire, pour construire des Applications Monopages (SPA) modernes, performantes et maintenables. Nous partir des concepts fondamentaux (le modèle mental de React, JSX, les composants) pour aller vers des sujets intermédiaires essentiels : les Hooks, le routage, la gestion des données asynchrones, les formulaires, le styling, l'accessibilité et les tests.

Objectifs :

- Comprendre le modèle de composants (composition, props, clés, rendu conditionnel).*
- Maîtriser state & hooks de base : useState, événements, levée d'état, useEffect pour l'asynchrone.*
- Gérer formulaires (contrôlés), erreurs/chargements, routing (React Router).*
- Structurer l'état global simple (Context + reducer) et organiser les appels API.*
- Styliser (CSS Modules ou Tailwind), appliquer des notions d'accessibilité (ARIA, focus).*
- Écrire des tests (Jest/React Testing Library) et builder/déployer le projet.

Chapitres :

1. Setup & fondamentaux : Vite, scripts, JSX, composants, React DevTools*
2. Composition & props : listes/keys, rendu conditionnel, patterns simples*
3. State & événements : useState, levée d'état, état dérivé (bonnes pratiques)*
4. Effets & async : useEffect, nettoyage, pièges du tableau de dépendances, fetch de base*
5. Formulaires : contrôlés vs non contrôlés, validation simple, messages d'erreur*
6. Routing : React Router (routes, params, navigation, erreurs/404)*
7. Données & état global : conventions d'appels API, gestion chargement/erreur, Context + reducer (intro)*
8. UI & styles : CSS Modules / Tailwind, responsive, composants réutilisables, accessibilité*
9. Qualité : linter/formatter, tests (Jest/RTL), debug, préparation build*
10. Mini-projet : petite SPA liste/détail + formulaire + appels API, README & build de démo

À la fin :

Vous saurez réaliser une SPA complète en React : composants clairs et réutilisables, hooks (useState, useEffect), routing, formulaires contrôlés et appels API robustes — le tout stylé (CSS Modules/Tailwind), testé (Jest/RTL) et outillé (Vite + linter), prêt pour le portfolio.

Suite recommandée (avancé) : performance React (memo, useMemo, useCallback), gestion de données (TanStack Query / Redux Toolkit), découpage de code, lazy loading et Suspense.