

## React Beyond the Basics

**durata: 3gg.**

Questo corso avanzato è pensato per sviluppatori full stack che hanno già una solida conoscenza di TypeScript e OOP e desiderano approfondire le proprie competenze in React. L'obiettivo è fornire gli strumenti necessari per sviluppare applicazioni moderne e scalabili utilizzando le best practices di React, con particolare attenzione alle performance e alla gestione dello stato.

Il corso esplorerà i concetti avanzati di React e come applicarli per migliorare la gestione della UI, ottimizzare le performance e implementare pattern architetturali avanzati come HOC (Higher-Order Components), Render Props e Componenti Composti. Inoltre, verranno trattati approfondimenti su Next.js, uno dei framework più utilizzati in combinazione con React, per ottimizzare il rendering, la gestione delle API e la configurazione di progetti complessi.

Al termine del corso, sarai in grado di costruire applicazioni React robuste, efficienti e facilmente manutenibili, applicando tecniche avanzate di gestione dello stato, ottimizzazione delle performance e design pattern, con un occhio di riguardo alle best practices in ambiente React e Next.js.

### CONTENUTI

#### Modulo 1 - Typescript avanzato:

- Introduzione a Typescript
- La programmazione Funzionale
- Uso avanzato dei Type Manipulation
- Generics
- Modellazione in Classi
- Mixins

#### Modulo 2 - React e Next.js: Fondamenti Avanzati

- Introduzione a React e le sue evoluzioni/versioni
- Introduzione a Next.js
- Differenze tra l'approccio React e l'approccio Next.js e vantaggi per applicazioni avanzate
- Considerazioni sull'impianto e la configurazione dei progetti
- Static Generation (SSG) e Server-Side Rendering (SSR): differenze, casi d'uso e configurazione
- Brevi cenni a JSX
- `getStaticProps`, `getServerSideProps`, `getStaticPaths`: come funzionano e quando utilizzarli
- Ottimizzazione delle performance con il prefetching automatico e caching delle API
- Routing
- Introduzione alle API Routes di Next.js per creare API back-end interne
- Data Fetching e caching
- Introduzione a React Context API per la gestione dello stato globale
- Panoramica di Redux Toolkit: quando e perché usarlo in Next.js
- Strategie di ottimizzazione delle performance in React: memoization, `useCallback`, `useMemo`

### Modulo 3 - Next.js Avanzato e Best Practices per Applicazioni React

- Ottimizzazione della UI con il code splitting e il dynamic import in Next.js
- Gestione dello stato lato server con SWR o React Query
- Autenticazione in Next.js: utilizzo di JWT e integrazione con NextAuth.js
- Collegamento a un database (es. MongoDB o PostgreSQL) con ORM
- Rendering ibrido: combinare SSR e SSG con rendering on-demand e ISR (Incremental Static Regeneration)
- BFF (Backend For Frontend): pattern per gestire API complesse con Next.js
- Architettura a componenti e design pattern avanzati: HOC (Higher-Order Components), Render Props e Compound Components
- Pattern avanzati di gestione dello stato: State Machines e Statecharts con XState
- Ottimizzazione delle immagini con il componente next/image
- Gestione del caricamento dei file statici e delle risorse
- Internazionalizzazione (i18n) in Next.js
- Analisi delle performance e strumenti di debugging

**Commentato [Id1]:** E' possibile aggiungere un su GraphQL/relay e uno su React Query per il fetching?

**Commentato [Id2]:** Da togliere, al momento eviterei cose server

**Commentato [Id3]:** Da togliere, al momento eviterei cose server

**Commentato [Id4]:** Da togliere, al momento eviterei cose server

**Commentato [Id5]:** Questo punto da approfondire bene soprattutto la parte di debugging sia di react, browser e vscode