



# **Optimize** **Performance** **in React**

Make your applications blazing fast.



Utilize React.memo and PureComponent to prevent unnecessary re-renders of components.

# Minimize re-renders



# Virtualize large lists

Implement techniques like windowing or infinite scrolling to render only visible items for improved rendering speed.



# Memoize expensive computations

Use memoization libraries like memoize-one or useMemo to cache expensive computations and avoid redundant calculations.



# Optimize renders with **shouldComponentUpdate**

Fine-tune component updates by implementing `shouldComponentUpdate` to avoid rendering when not required.



# Leverage the power of **React.lazy**

Load components lazily using React.lazy and display fallback UI with Suspense to improve initial load times.



# Use production builds

Ensure that your application is running in production mode to take advantage of optimizations like minification and dead code elimination.



# Profile performance with **React DevTools**

Identify performance bottlenecks using the React DevTools Profiler and make targeted optimizations.



# Optimize image loading

Compress and optimize images, use lazy loading techniques, or consider using responsive image libraries to reduce load times.



# Optimize data fetching

Implement efficient data fetching strategies such as pagination, caching, or using GraphQL to minimize network requests.



# Split code with code splitting

Split your application's code into smaller chunks using tools like webpack's code splitting or React.lazy to load only what is necessary for each route or feature.



# Hi, I'm Rahul

I'll help you get  
better at **React**,  
**Follow along.**

