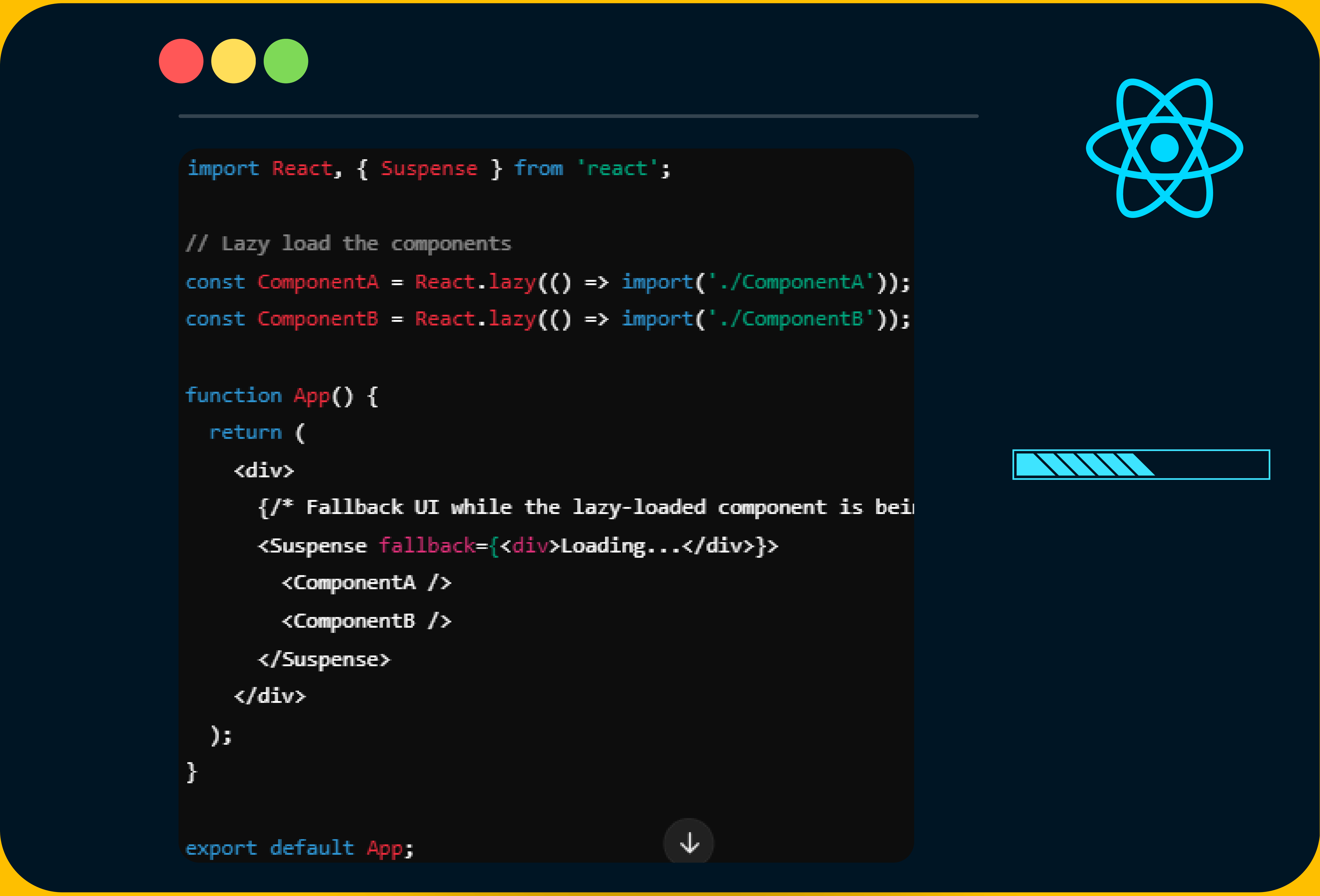
ReactJs

Reduced React App Load Time by 30%

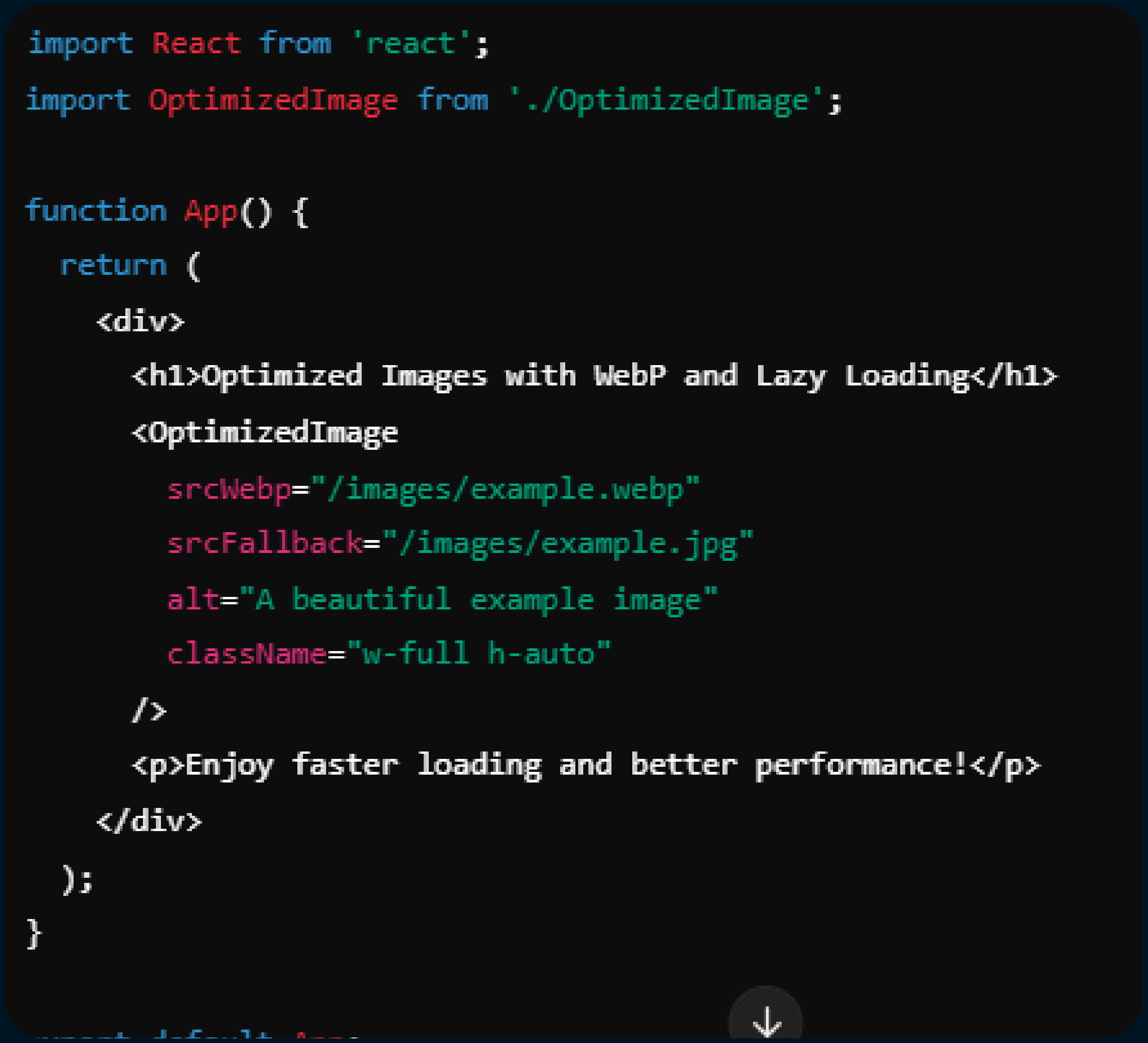
1 Split the code using React's lazy and Suspense.



```
import React, { Suspense } from 'react';

// Lazy load the components
const ComponentA = React.lazy(() => import('./ComponentA'));
const ComponentB = React.lazy(() => import('./ComponentB'));

function App() {
  return (
    <div>
      {/* Fallback UI while the lazy-loaded component is bei
      <Suspense fallback={<div>Loading...</div>}>
        <ComponentA />
        <ComponentB />
      </Suspense>
    </div>
  );
}

export default App;
```

NEXT

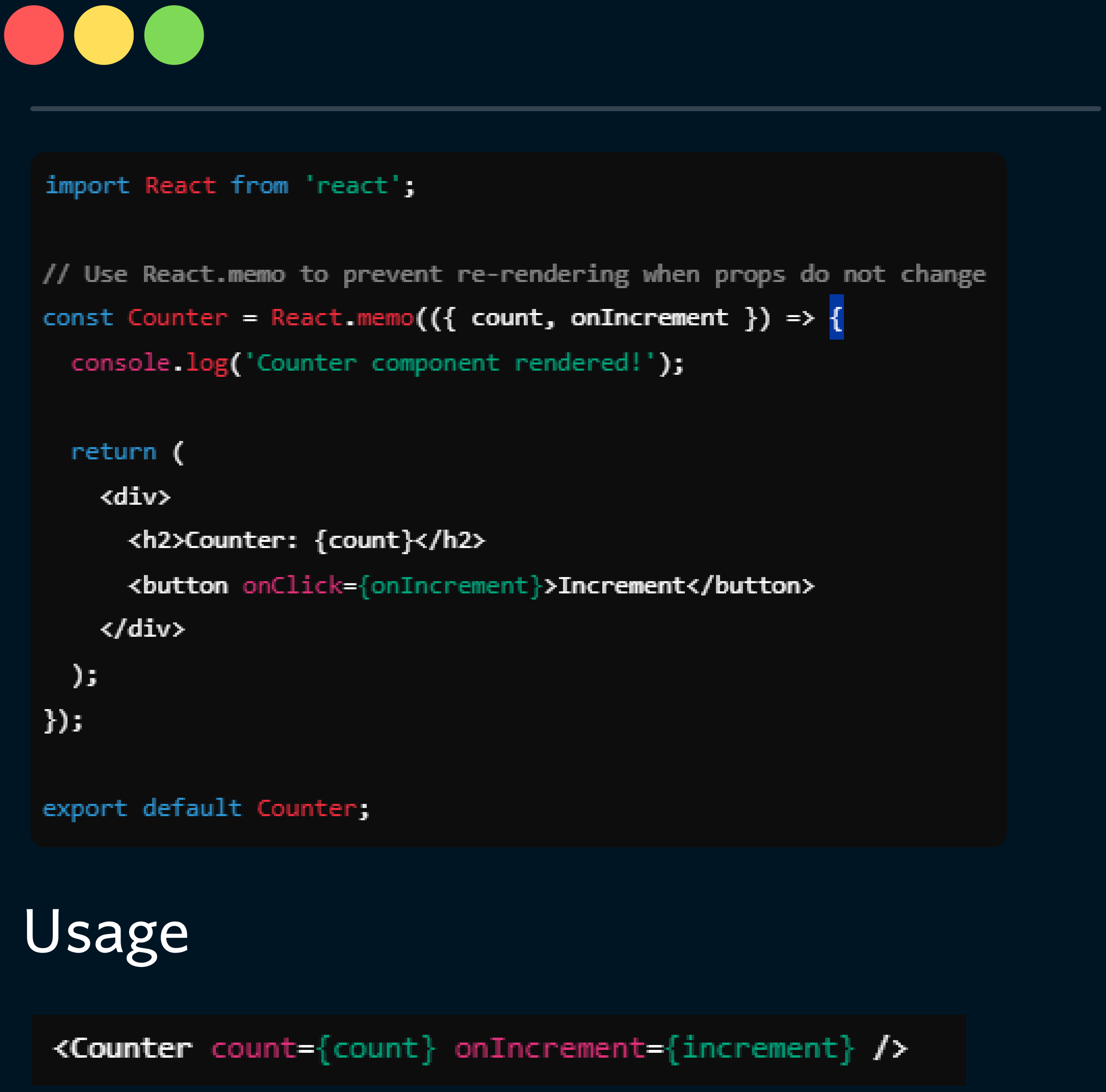
2 Optimized images with WebP and lazy loading.



```
import React from 'react';
import OptimizedImage from './OptimizedImage';

function App() {
  return (
    <div>
      <h1>Optimized Images with WebP and Lazy Loading</h1>
      <OptimizedImage
        srcWebp="/images/example.webp"
        srcFallback="/images/example.jpg"
        alt="A beautiful example image"
        className="w-full h-auto"
      />
      <p>Enjoy faster loading and better performance!</p>
    </div>
  );
}

export default App;
```

NEXT

3 Used React.memo and useCallback to prevent unnecessary re-renders.



```
import React from 'react';

// Use React.memo to prevent re-rendering when props do not change
const Counter = React.memo(({ count, onIncrement }) => {
  console.log('Counter component rendered!');

  return (
    <div>
      <h2>Counter: {count}</h2>
      <button onClick={onIncrement}>Increment</button>
    </div>
  );
});

export default Counter;
```

Usage

```
<Counter count={count} onIncrement={increment} />
```

NEXT

Thank You

Md Hafijur Rahman