React Performance Optimization – Executable Guide

This document demonstrates route-level code splitting, lazy loading of heavy components, and skeleton loaders in a React application.

1. App.js – Route-level Code Splitting

```
import React, { Suspense, lazy } from "react";
import { BrowserRouter, Routes, Route } from "react-router-dom";

const Home = lazy(() => import("./pages/Home"));
const Courses = lazy(() => import("./pages/Courses"));
const Gallery = lazy(() => import("./pages/Gallery"));

function App() {
  return (
    <BrowserRouter>
      <Suspense fallback={<div>Loading...</div>}>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/courses" element={<Courses />} />
          <Route path="/gallery" element={<Gallery />} />
        </Routes>
      </Suspense>
    </BrowserRouter>
  );
}

export default App;
```

2. Lazy-loaded Heavy Component

```
import React, { lazy, Suspense } from "react";

const GalleryGrid = lazy(() => import("../components/GalleryGrid"));

export default function Gallery() {
  return (
    <Suspense fallback={<div>Loading gallery...</div>}>
      <GalleryGrid />
    </Suspense>
  );
}
```

3. Skeleton Loader Component

```
import "./skeleton.css";

export default function SkeletonCard() {
  return (
    <div className="skeleton-card">
      <div className="skeleton-image"></div>
      <div className="skeleton-text"></div>
      <div className="skeleton-text short"></div>
    </div>
  );
}
```

Using these techniques improves both actual load time and perceived performance.