The Ultimate SharePoint Performance Tuning Checklist

Phase 1: Analysis & Monitoring

- **Establish a Baseline:** Use the "Page Diagnostics for SharePoint" tool on key pages to get an initial score.
- **Run a Lighthouse Audit:** Generate a performance report in your browser's DevTools to identify specific bottlenecks.
- **Analyze the Network Tab:** Check for large files (>100KB) or slow API calls (>500ms).
- **Identify Key Metrics:** Note your current Largest Contentful Paint (LCP) and Time to Interactive (TTI). Your goal is to improve these.

Phase 2: Image & Media Optimization

- **Audit All Images:** Identify images larger than 100KB on the page.
- **Choose the Right Format:** Convert PNGs to JPEGs or WebP for photos. Use SVG for logos.
- **Resize to Scale:** Resize images to the exact dimensions they will be displayed. Do not rely on CSS to shrink large images.
- **Compress Images:** Use a tool like TinyPNG or an image editor to compress images before uploading.
- **Implement Lazy Loading:** Ensure images and media-heavy web parts below the fold are set to lazy load.

Phase 3: Web Part & Custom Code Review

- **Analyze SPFx Bundle Size:** Use `gulp bundle --ship --analyze` to check for large dependencies.
- **Replace Heavy Libraries:** Swap out large libraries (e.g., Moment.js) with lighter alternatives (e.g., date-fns).
- **Minimize API Calls:** Combine multiple API calls into a single one where possible. Cache results that don't change often.
- **Audit Third-Party Scripts:** Question the necessity of every external script or tracker. Remove any that are not business-critical.

Phase 4: Caching & Network Strategy

- **Enable Office 365 CDN:** Ensure the public CDN is enabled in your tenant for faster global asset delivery.
- **Review Cache Headers:** For custom APIs, ensure correct cache-control headers are being sent to allow browser caching.
- **Leverage Browser Caching:** Confirm that static assets (CSS, JS) have long cache expiration dates.