# Next.js Interview Q&As (Part 3)

# (Performance, SEO & Deployment)

## Section 1 : Performance Optimization

**Q1. How does Next.js improve performance over React?**

Answer: Built-in SSR, SSG, ISR reduce load times.
Image & script optimization by default.
Automatic code splitting reduces bundle size.
💡 Tip: Mention Core Web Vitals improvements.

**Q2. What is Automatic Code Splitting?**

Answer: Each page loads only the JS it needs.
Reduces initial bundle size.
Improves page speed.
💡 Tip: Contrast with React SPA loading everything.

**Q3. How does Next.js handle lazy loading?**

Answer: Dynamic imports with next/dynamic.
Load components only when needed.
Supports SSR fallback.
💡 Tip: Mention for charts, modals, heavy components.

**Q4. How does Next.js optimize images?**

Answer: next/image resizes, compresses, and lazy-loads images.
Supports WebP & responsive sizes.
Improves LCP & Core Web Vitals.
💡 Tip: Big SEO advantage over <img>.

**Q5. How does Next.js optimize fonts?**

Answer: next/font loads Google/local fonts efficiently.
Removes layout shifts (CLS).
Improves performance scores.
💡 Tip: New feature in Next.js 13+.

**Q6. How does Next.js handle caching?**
Answer: Uses Cache-Control headers for static/SSR pages.
ISR uses background regeneration.
API routes can set custom cache headers.
💡 Tip: Helps balance speed & freshness.

**Q7. How to optimize large lists in Next.js?**
Answer: Use pagination, infinite scroll, or virtualization.
Avoid fetching huge data sets at once.
💡 Tip: Combine with ISR for performance.

**Q8. What is Script Optimization in Next.js?**
Answer: <Script> component controls script loading.
Options: beforeInteractive, afterInteractive, lazyOnload.
💡 Tip: Use for third-party scripts like analytics.

**Q9. How does Next.js handle bundle analysis?**
Answer: Enable @next/bundle-analyzer.
Shows size of JS bundles.
Helps optimize imports.
💡 Tip: Essential for large apps.

**Q10. How do you reduce Next.js build size?**
Answer: Tree-shaking, dynamic imports, image compression.
Remove unused npm packages.
Enable ProGuard-like minification.
💡 Tip: Deploy on Vercel for automatic optimizations.

## Section 2 : SEO Features

**Q11. How does Next.js improve SEO vs React?**
Answer: SSR/SSG → HTML ready for crawlers.
Better performance = higher rankings.
Built-in Head management.
💡 Tip: Mention React = CSR → weaker SEO.

**Q12. What is the next/head component?**

Answer: Controls <title>, meta tags, and favicon.
Works per page.
Improves SEO & social sharing.
💡 Tip: Compare with React Helmet.

**Q13. How do you add dynamic meta tags in Next.js?**

Answer: Use next/head inside dynamic pages.
Pass props from getServerSideProps or getStaticProps.
💡 Tip: Essential for blogs, products, profiles.

**Q14. How does Next.js handle sitemap generation?**

Answer: Generate with plugins like next-sitemap.
Creates sitemap.xml and robots.txt.
Improves crawlability.
💡 Tip: Mention for SEO-focused apps.

**Q15. How do you implement Open Graph & Twitter meta tags?**

Answer: Add <meta property="og:title" ... /> inside <Head>.
Dynamic values come from props.
💡 Tip: Important for link previews.

**Q16. What is Structured Data in Next.js?**

Answer: JSON-LD added inside <script type="application/ld+json">.
Helps search engines understand content.
💡 Tip: Used for rich snippets in Google.

**Q17. How do Core Web Vitals affect SEO in Next.js apps?**

Answer: Metrics: LCP, FID/INP, CLS.
Optimized with Next.js image, font, and caching.
💡 Tip: Vercel Analytics can monitor vitals.

**Q18. How to handle canonical URLs in Next.js?**

Answer: Add <link rel="canonical" href="..." /> in <Head>.
Prevents duplicate content issues.
💡 Tip: Interviewers love SEO-focused details.

**Q19. How does ISR impact SEO?**
Answer: Keeps static pages fresh without rebuilds.
Search engines always see updated content.
💡 Tip: Best for blogs & e-commerce.

**Q20. How do you handle localization (i18n) in Next.js?**
Answer: Configure i18n in next.config.js.
Supports multiple locales, domain routing.
💡 Tip: Mention SEO benefits for global apps.

## Section 3 : Deployment & CI/CD

**Q21. How do you deploy a Next.js app?**
Answer: Easiest with Vercel.
Also works on Netlify, AWS, or custom Node server.
💡 Tip: Vercel auto-optimizes builds.

**Q22. What is the difference between static export and SSR deploy?**
Answer: Static export → next export, no SSR, pure static.
SSR deploy → needs Node.js server or Vercel.
💡 Tip: Use static export for blogs/docs.

**Q23. How do you generate a production build?**
Answer: Run next build → optimized .next folder.
Serve with next start.
💡 Tip: Always test with production build.

**Q24. How do you configure custom domains in Next.js?**
Answer: On Vercel → add custom domain.
Or set up reverse proxy on custom servers.
💡 Tip: HTTPS auto-enabled on Vercel.

**Q25. How do you add environment variables in deployment?**
Answer: Add in .env.local or in hosting platform’s dashboard.
Access via process.env.
💡 Tip: Use NEXT_PUBLIC_ for client variables.

**Q26. How do you set up CI/CD for Next.js?**
Answer: Use GitHub Actions, GitLab CI, or Vercel auto-deploy.
Builds triggered on push/merge.
💡 Tip: Vercel = zero-config CI/CD.

**Q27. What is Next.js Image Loader configuration?**
Answer: In next.config.js, define external image domains.
Ensures optimized images load from outside hosts.
💡 Tip: Needed for CMS integrations.

**Q28. How do you deploy Next.js with Docker?**
Answer: Create Dockerfile → build app → run in Node container.
Good for custom cloud setups.
💡 Tip: Use multi-stage builds for smaller images.

**Q29. What is the role of next.config.js in deployment?**
Answer: Configures images, redirects, rewrites, headers.
Runs at build and runtime.
💡 Tip: Key file for production apps.

**Q30. How do you monitor performance in production?**
Answer: Use Vercel Analytics, Google Lighthouse, or New Relic.
Track Core Web Vitals, errors, API latency.
💡 Tip: Monitoring = real-world readiness.