# TECHNICAL SEO CHECKLIST FOR SAAS

SEO Curly

## SITE STRUCTURE

- ○ Use subdirectories instead of subdomains (/blog vs blog.yoursite.com)
- ○ Implement logical URL hierarchy (/features/category/specific-feature)
- ○ Ensure every page is reachable within 3 clicks from the homepage
- ○ Create hub pages that link to related content
- ○ Set up proper internal linking between marketing and product pages

## URL MANAGEMENT

- ○ Use descriptive, keyword-rich URLs
- ○ Implement 301 redirects for changed URLs
- ○ Avoid dynamic parameters in important URLs
- ○ Create clean URLs for onboarding steps
- ○ Document URL changes for team reference

## LOADING SPEED (LCP)

- ○ Preload critical images and resources
- ○ Implement next-gen image formats (WebP)
- ○ Optimize CSS delivery and eliminate render-blocking resources
- ○ Use CDN for static assets
- ○ Compress images and optimize file sizes

## INTERACTIVITY (FID)

- ○ Split JavaScript bundles and lazy load non-critical code
- ○ Use web workers for heavy computations
- ○ Minimize third-party script impact
- ○ Optimize interactive elements (demos, calculators, forms)
- ○ Test all interactive features for responsiveness

## VISUAL STABILITY (CLS)

- ○ Reserve space for dynamic content
- ○ Set dimensions for all images and videos
- ○ Avoid inserting content above existing elements
- ○ Test layout stability across different screen sizes
- ○ Monitor for layout shifts after deployments

SEO Curly

## RENDERING OPTIMIZATION

- ○ Implement server-side rendering (SSR) for public pages
- ○ Use progressive enhancement for interactive features
- ○ Ensure core content loads without JavaScript
- ○ Test pages with JavaScript disabled
- ○ Monitor JavaScript errors that could affect SEO

## SPA & FRAMEWORK SETUP

- ○ Implement proper meta tag management for route changes
- ○ Set up canonical tags for dynamic routes
- ○ Create XML sitemaps for JavaScript-generated content
- ○ Test URL changes with Google's URL Inspection tool
- ○ Monitor crawl errors in Search Console

## PUBLIC VS PRIVATE CONTENT

- ○ Clearly define what content should be publicly accessible
- ○ Block user dashboards and private data from search engines
- ○ Allow crawling of marketing pages, docs, and help content
- ○ Implement progressive disclosure for sensitive features
- ○ Regular audit for accidentally exposed private content

## ROBOTS.TXT CONFIGURATION

- ○ Block /dashboard/, /admin/, /user/ directories
- ○ Allow /docs/, /help/, /api/docs/ directories
- ○ Block sensitive URL parameters (?token=, ?session=)
- ○ Test robots.txt with Google Search Console
- ○ Update robots.txt when adding new features

## CANONICAL TAGS

- ○ Point authenticated pages to public canonical versions
- ○ Implement canonical tags for similar content across user states
- ○ Handle pagination with rel="next" and rel="prev"
- ○ Test canonical implementation across different user journeys
- ○ Monitor for canonical conflicts

SEO Curly

## HREFLANG IMPLEMENTATION

- ○ Set up hreflang tags for all language/region variations
- ○ Include x-default for fallback targeting
- ○ Test hreflang with Google's International Targeting report
- ○ Handle regional pricing and feature differences
- ○ Monitor international organic traffic performance

## LOCALIZATION TECHNICAL SETUP

- ○ Use subdirectories for international versions (/de/, /fr/)
- ○ Configure CDN for regional performance optimization
- ○ Implement currency and pricing localization
- ○ Handle GDPR compliance without blocking search engines
- ○ Set up geo-targeting in Google Search Console

## DOCUMENTATION STRUCTURE

- ○ Organize API docs by use case, not just endpoints
- ○ Create SEO-friendly URLs for documentation sections
- ○ Include complete, runnable code examples
- ○ Target developer-specific search queries
- ○ Link between related API methods and guides

## CODE EXAMPLE OPTIMIZATION

- ○ Make all code examples copy-pasteable
- ○ Include explanatory comments in code snippets
- ○ Provide examples in multiple programming languages
- ○ Create dedicated integration guides for popular frameworks
- ○ Test all code examples for accuracy

## TRIAL PAGES

- ○ Optimize trial signup pages for conversion keywords
- ○ Implement structured data for software applications
- ○ Create unique URLs for different onboarding steps
- ○ Include clear value propositions and feature lists
- ○ A/B test trial page elements while maintaining SEO

## ONBOARDING FLOW SEO

- ○ Give each onboarding step a descriptive URL
- ○ Create content that serves both users and search engines
- ○ Link onboarding content to relevant feature pages
- ○ Track conversion rates from organic traffic through onboarding
- ○ Monitor for drop-offs in onboarding

## DATA ISOLATION

- ○ Use clear URL patterns to separate tenant data (/app/tenant-id/)
- ○ Block all tenant-specific URLs from search engines
- ○ Implement database queries that exclude tenant data from sitemaps
- ○ Test for accidental exposure of customer data
- ○ Regular security audits for SEO-accessible content

## PLATFORM CONTENT STRATEGY

- ○ Focus SEO efforts on platform-level features and capabilities
- ○ Create generic examples that don't reveal customer data
- ○ Link tenant areas to relevant public documentation
- ○ Monitor for tenant data appearing in search results
- ○ Maintain a clear separation between public and private content

SEO Curly

## PERFORMANCE MONITORING

- ○ Set up Core Web Vitals monitoring by page type
- ○ Monitor crawl errors in Google Search Console weekly
- ○ Track technical SEO metrics alongside business KPIs
- ○ Set up alerts for performance regressions after deployments
- ○ Monitor JavaScript errors that could impact SEO

## REGULAR AUDITS

- ○ Monthly technical SEO health checks
- ○ Quarterly comprehensive technical audits
- ○ Post-deployment SEO regression testing
- ○ Regular robots.txt and sitemap reviews
- ○ Monitor competitor technical SEO changes

## BUSINESS IMPACT TRACKING

- ○ Track organic traffic to trial conversion rates
- ○ Monitor customer acquisition cost from organic channels