

Ethical and Sustainable Website Checklist

A practical checklist to design and build websites that minimize environmental impact while delivering a fast, inclusive user experience.

Purpose

Help teams create sustainable, efficient, and ethical websites aligned with environmental and social responsibility.

When to use

At project kickoff, during audits, redesigns, and ongoing optimizations.

How to use

Follow the checklist step-by-step—benchmark your current site, eliminate waste, optimize essentials, offset remaining impact, and set clear energy/impact goals.

Fundamental Principles

- Keep content and code simple and only include what is needed
- Make sure everything runs fast and uses fewer resources
- Design the website so it works well for all people and devices
- Be clear about what you do and protect user privacy
- Build the site to reduce harm and help the environment recover
- Be truthful in design and in how you handle data

Workflow Steps

- Measure the current site and compare it to the old version and competitors. Look at performance, energy use, and user experience.
- Remove any features, content, or code that are not needed.
- Improve all the important parts like code, images, and user experience.
- Calculate the remaining carbon footprint and offset it responsibly.
- Set clear energy and impact goals, including stretch targets.

Server-Side (Hosting & Backend)

- Choose data centers geographically close to users to reduce latency and energy
- Use a Content Delivery Network (CDN)
- Select hosts with low PUE (Power Usage Effectiveness)
- Prefer green hosting providers (renewable energy powered)
- Enable server caching to reduce repeated processing
- Use scalable cloud resources, matching demand precisely
- Write clean, efficient backend code (avoid wasteful loops/queries)
- Block unnecessary bots and scrapers with robots.txt or firewall
- Design a well-architected database
- Optimize database queries and indexes
- Consider serverless architecture to reduce idle server usage
- Choose energy-efficient server types and instances (on-demand, spot)
- Use appropriate storage classes (e.g., cold storage for infrequently accessed data)
- Explore edge computing to reduce server load and latency

- Minimize 3rd party integrations (each adds overhead)
- Prefer programming languages known for efficiency (e.g., Go, Rust)

Client-Side (Front-End Optimization)

- Enable browser caching
- Apply code compression and minification (JS, CSS, HTML)
- Compress and optimize images (lossless/lossy as appropriate)
- Serve responsive image sizes based on device
- Use modern image formats (WebP, AVIF)
- Implement lazy loading to defer images and content until needed
- Minimize or remove 3rd party embeds/widgets
- Reduce tracking scripts and analytics bloat
- Use system or standard fonts, limit font variations
- Use efficient font loading (e.g., font-display: swap) to avoid invisible text
- Limit use of heavy media: images, videos, animations
- Optimize videos with adaptive streaming, efficient codecs; avoid autoplay
- Use AMP (Accelerated Mobile Pages) where appropriate
- Build Progressive Web Apps (PWA) to improve efficiency and offline use
- Prefer static sites or static site generators (Hugo, Jekyll, Eleventy)
- Remove unused JS and CSS
- Avoid heavy frameworks unless absolutely needed
- Write clean, minimal front-end code
- Minimize page reloads and redirects

- Reduce HTTP requests by combining files (CSS sprites, JS bundling)
- Use Prefetch & Preconnect to speed up essential resource loading
- Offer dark mode option to reduce OLED screen energy consumption

UX & Content

- Design with good information architecture (IA)
- Improve SEO with semantic markup and optimized metadata
- Streamline content: focus on essentials, remove fluff
- Optimize for accessibility (WCAG compliance, keyboard navigation, alt text)
- Test on older and low-spec devices to ensure inclusivity
- Follow Privacy by Design: limit data collection and processing to reduce energy

Useful Resources

Website Carbon Calculator — <https://www.websitecarbon.com/> - estimate the carbon footprint of any website

Green Web Foundation — <https://www.thegreenwebfoundation.org/> - check if your web host uses renewable energy

Lighthouse (Google Chrome DevTools) — <https://developers.google.com/web/tools/lighthouse/> - audit website performance and efficiency

WebPageTest — <https://www.webpagetest.org/> - analyze website speed and load efficiency

W3C Sustainability Guidelines — <https://www.w3.org/TR/sustainability/> - official guidelines for sustainable web design and development