A Practical Framework for Sustainable Website Design and Development

Tiberiu-Septimiu Ungurianu, Sahar Al-Sudani, and Ramzi Djemai
School of Computing and Digital Media
London Metropolitan University, London, UK
corresponding author: s.alsudani@londonmet.ac.uk

***Abstract*—Sustainable web design is a fundamental aspect of Green Computing that aims to reduce carbon emissions and energy consumption. This paper explores sustainable website development by assessing best practices, comparing design approaches, such as minimalist design, code minimisation and clean code, efficient image compression, reduced reliance on heavy JavaScript frameworks and Content Delivery Network (CND) services, optimised caching strategies and examining methodologies contributing to energy efficiency.**

This paper presents a practical framework for sustainable website design. It incorporates renewable energy hosting, lightweight coding principles and evidence-based sustainability interventions. The framework was implemented and tested on The London Library website, yielding a remarkable 99% reduction in carbon emissions per visit. The results prove how crucial it is to track environmental metrics and offer applicable steps for creating websites that are not only eco-friendly but also high-performing.

The London Library website was selected for this project due to its consistent, high-traffic usage, particularly among academic audiences. Although such sites may appear to have a limited environmental impact, our findings reveal that substantial reductions in carbon emissions are achievable without compromising performance or user experience. This transformation highlights the untapped potential of sustainable optimisation even for seemingly efficient websites.

Keywords — ***Sustainable web design, Carbon footprint reduction, Green computing***

I. INTRODUCTION

The rapid growth of the Internet has significantly increased global electricity consumption. If there can be made an imagination exercise and the internet was a country, it would rank among the top energy consumers worldwide. Websites and web applications contribute substantially to this footprint, necessitating a shift toward sustainable web design. This paper proposes a framework based on empirical research, guiding developers toward a greener digital infrastructure.

Most companies first interact with customers through their websites, where they showcase products, services, and sustainability commitments. Some companies actively pursue carbon neutrality, which improves their image and influences customer decisions. Beyond physical product sustainability, the digital world also has a significant environmental impact. Although websites are typically designed to be reliable and upgradable, future standards will need to address both technological advancement and environmental concerns.

Although not widely admitted, websites leave a significant environmental footprint. Running a site requires servers and physical machines that consume constant electricity. Most electricity production still relies on polluting processes: fossil fuels emit CO_2, nuclear power generates radioactive waste and hydropower can destroy ecosystems and release methane. Therefore, powering servers with solar and wind energy remains the most sustainable solution.

This work aims to go beyond using green hosting servers by developing a prototype for a fully functional website with minimal resources while delivering a user experience comparable to a typical website. In addition, this paper proposes a Green Framework intended to assist website designers in developing environmentally sustainable websites. The framework integrates several core methodologies, including minimalist design to simplify interfaces and reduce page complexity, code minimisation and clean coding practices to eliminate unnecessary resource consumption and efficient image compression techniques, such as adopting modern formats and lazy loading, to optimise media delivery. It further supports a reduction in reliance on heavy JavaScript frameworks and third-party Content Delivery Network (CDN) services to decrease site bloat and associated energy demands and implement optimised caching strategies to enhance load efficiency for returning users. Along with these recommendations regarding renewable energy hosting, lightweight technology adoption and the provision of structured support, guidance and resources, this framework offers a foundation for creating environmentally responsible and high-performing websites. So, it is crucial to understand how important and actual this subject is. It is crucial to minimise and eliminate any negative environmental impact of the technologies that are used. To further advance, the digital space needs to be restructured in a cleaner and less resource-consuming manner than it is now, and this paper will demonstrate how a website today can be implemented while meeting the green criteria of standards

II. RELATED WORK

Halleberg et al (2023) in their paper examine how much web developers know about sustainable web design and how much they actually implement. While many acknowledge the importance of sustainability, their actual practices lag behind. A survey of 77 developers in Norway revealed that most do not optimise file formats, subset fonts, or use simple methods like dark mode to reduce energy consumption. There is also no clear industry standard, leading developers to stick with familiar methods, even when they are inefficient. To make real progress, the industry needs better guidelines, improved education, and incentives that encourage developers to apply more sustainable practices.

According to Willis et al. (2020) in their paper, older websites consume excessive energy due to bloated code, redundant scripts, and animations that add no real value. Many of these sites still run on high-carbon servers, even though more sustainable hosting solutions exist. This study outlines a path toward a low-power web by cutting down unnecessary JavaScript, switching to more efficient media formats, and relying on server-side rendering rather than making users' devices handle all computations. A more efficient approach to web development can significantly reduce unnecessary energy consumption and improve site performance.

Arora et al. (2024) highlight that many websites claiming to be sustainable fail to deliver meaningful energy savings in practice. Some claim to be low-carbon while still loading unnecessary assets in the background, resulting in minimal real reductions in power consumption. A common misconception is that lowering data transfer alone will automatically lower energy use, but in reality, factors such as network efficiency, cloud services, and device-level processing all contribute significantly. This research highlights the need for better measurement tools to assess sustainability efforts and argues that web sustainability must be built on real data rather than assumptions. González-Sordé (2024) argues that simplifying website text can reduce energy consumption by lowering the amount of data that needs to be transferred and processed. This study compares nine websites before and after implementing Easy Language and finds a 42% reduction in CO_2 emissions. Less complex text results in smaller page sizes, faster load times, and lower power usage. In addition to sustainability benefits, this approach improves accessibility, making content more readable for users with cognitive impairments. However, oversimplification could reduce engagement for some users, so there is a need to balance clarity with content depth. According to Dornauer and Felderer (2023) with mobile devices accounting for over half of web traffic, mobile web apps are a major factor in energy consumption. JavaScript-heavy applications, inefficient rendering, and excessive background processes contribute to high power usage. This study reviews ten years of research on optimising mobile web energy efficiency. It recommends switching to Progressive Web Apps, optimising image and video formats using WebP and AV1, and minimising unnecessary JavaScript execution to reduce CPU load. Reducing mobile web energy consumption by just 5% would save enough power to offset the output of a nuclear reactor, highlighting the scale of potential improvements.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Černý and Donahoo (2016) in their study examine different UI frameworks impact energy consumption. Client-side rendering reduces server load but increases the energy demands on users' devices, while server-side rendering shifts the processing burden to data centres, which may be more efficient at scale but require robust infrastructure. A balanced approach, distributing processing between the client and the server, offers the best potential for energy efficiency. Caching and resource prioritisation also play an important role in lowering power usage, preventing devices from reloading the same resources unnecessarily. Choosing a UI framework is not just about speed and usability; it also affects sustainability.

III. SUSTAINABLE WEBSITE TOOLS AND TECHNIQUES

Implementing a green website requires more than efficient code design. While coding is a critical component, it is equally essential to host the website on servers powered by renewable energy. Furthermore, code should be clean and maintainable by default to support future iterations and scalability. The system architecture should be modular to facilitate upgrades, modifications, and the seamless scaling of applications.

The choice of development tools—including programming languages, backend frameworks, and databases—should prioritise low energy consumption and efficient memory usage by default. . The user interface should keep delivering both ease of use and a smooth experience to the users so that the providers can fully adopt the switch to green methods without any postponement, which explains the late adoption of these features.

Lighthouse is an open-source, automated tool integrated into Chrome DevTools that evaluates a website's performance, accessibility, SEO, adherence to best practices, and compliance with progressive web app standards. The evaluation is made during an audit process on a scale from 0 to 100, where it is highlighted both passing elements as well as the ones that need improvement, providing some actionable recommendations with links to resources where the problems are explained as well as the solutions. Together, these features make Lighthouse a valuable tool for web developers aiming to optimise and enhance their websites, as it enables systematic tracking of the impact of each modification. In this study, Lighthouse was employed to assess the sustainability-focused website, providing empirical benchmarks for performance, accessibility, and best practice compliance.

For this work, the front-end part was developed using React.js supported by Vite with HTML and CSS, which form the foundation of the web interface. These front-end technologies were carefully selected and implemented with efficiency in mind, reducing file sizes and optimising rendering performance. React's Virtual DOM significantly improves energy efficiency by minimising DOM manipulations and using an efficient diffing algorithm to identify only necessary UI changes. Together with Vite's ability to deliver fast hot module replacement and optimised production builds, this front-end tech stack directly supports the project's goal of being green.

Spring Boot and Apache Maven are used to develop the backend server for local development because they manage resource allocation efficiently. They adapt to user requests dynamically, using more resources only when required. This makes them well-suited for sustainable backend development while offering robust tools for dependency management, testing, and configuration. Spring has caching capabilities and optimised requests that reduce the computational overhead and every computation during operational times. These technologies are an efficient foundation for developing scalable and modular server-side components.

The website is hosted on Cloudflare servers, a platform recognised by The Green Web Foundation for its green credentials. Deployment was handled via Wrangler, Cloudflare's command-line interface tool that streamlines the publishing process. The setup includes continuous integration and continuous deployment (CI/CD) pipelines to automate updates, compress static files, and apply HTTPS encryption. All assets are served locally through Cloudflare's energy-efficient infrastructure, ensuring minimal reliance on third-party services and improving both performance and environmental sustainability through reduced data transfer and optimised resource utilisation.

There are many ready-to-use analysis tools for green websites. The ones worth mentioning are: Ecograder, EcoIndex, Website Carbon, Beacon and Siti Green. Each offers a plethora of indicators that assess the green attributes sought after to satisfy the requirements that make a website green as depicted in Table-1.

Indicator / Tool	Ecograder	EcoIndex	Website Carbon	Beacon	Siti Green
CO_2 per pageview	X	X	X	X	X
Green hosting check	X		X	X	X
Page weight (KB/MB)	X	X	X	X	X
Image optimisation	X	X			
Number of server requests		X			
DOM complexity (number of elements)		X		X	
JavaScript/CSS optimisation	X	X		X	
Core Web Vitals (e.g. CLS, FID, LCP)				X	
UX/design impact on emissions	X				
Hosting energy source (green/offset)	X		X	X	X
Estimated annual emissions / impact	X	X	X	X	X
Water consumption estimate		X			
Emissions equivalence (trees, cars, etc.)	X		X	X	X

Table 1-Analysis Tools for Green Websites

IV. GREEN WEBSITE DESIGN AND DEVELOPMENT

The website selected for this work is a library website, which, after being assessed with tools provided by The Green Web Foundation, showed an emission of 21 grams of CO2 per visit. Converting the existing website into a green website involved several optimisation techniques.

The investigative model follows minimalist design principles. The menu has straightforward navigation with no secondary menus. The number of visual elements is kept low and only the most relevant ones are shown. This contributes to a minimalist design and reduces the computational resources required for rendering to a minimum. Unnecessary animations were replaced with alternatives that significantly lower energy consumption by reducing loading time and improving performance.

Data loading operations during the page loading process were minimised with particular attention to optimising image loading. This was achieved through file compression and implementation of lazy-loading techniques throughout the website guaranteeing that non-critical images are loaded only when needed. These techniques also apply to video content, which was optimised by turning off autoplay and preventing background loading. Videos load only when explicitly requested by the user, ensuring minimal bandwidth usage and reducing unnecessary resource consumption.

The development process integrated both local automation scripts and web-based compression tools to optimise image assets. Initially, images were converted from JPEG and PNG formats to WEBP and then compressed using services such as TinyJPG, which significantly reduced file size while maintaining acceptable visual quality. Image quality was reduced from 100% to 75%, a change imperceptible to users but significantly reducing file sizes. Although formats like AVIF or JPEG XL offer better compression, WEBP was preferred due to its wide browser support.

To further optimise delivery, local plugins were integrated within the project build system. Terminal commands like 'cwebp' were utilised to automate the compression. This way multiple responsive versions of each image were generated at 250px, 400px, 600px and 1200px resolutions, providing optimal rendering across various devices. These versions are dynamically served based on the user's device or screen resolution, ensuring that only the smallest appropriate image is loaded, significantly reducing data usage and improving loading times. These combined optimisations significantly reduced the image footprint, enhancing performance without compromising visual quality. , resulting in improved performance and contributing to attaining a consistent 100/100 performance score in the Lighthouse reports.

Code minimisation was achieved using a global CSS file, which helps reduce redundancy and keep the codebase cleaner. Vite plugins, such as 'vite-plugin-compression', were used to automatically compress and optimise all code and text files during the build and deployment phases, improving loading speed and reducing energy consumption.

The website does not rely on heavy JavaScript frameworks or external Content Delivery Networks (CDNs). It utilises a built-in font stored locally in the

WOFF2 format, which is lightweight and offers the best compression. No heavy front-end frameworks were employed beyond React and Vite, without additional third-party frameworks. All necessary components such as fonts, JavaScript, CSS and images, were bundled and served locally, thereby reducing external HTTP requests, improving load performance, and contributing to a lower energy consumption.
Effective caching techniques were implemented across the site to ensure rapid response times. These include setting appropriate HTTP cache headers, using service workers for offline functionality and leveraging browser storage mechanisms. By serving preloaded resources to returning users, the system conserves energy that would otherwise be used for repeated data fetching and rendering. This optimisation enhances user experience with quicker load times and minimises repeated server requests, thus lowering server load and associated energy usage.

Minimising HTTP requests was achieved by bundling assets and reducing to zero unnecessary external calls, like importing fonts or images and optimising all static resources. Backend caching was configured using Spring's @Cacheable annotation, enabling rapid delivery of frequently accessed content and reducing redundant computations. An H2 in-memory database was used to simulate backend data delivery, supporting efficient front-end rendering with minimal resource overhead. Large text content was restructured to be retrieved directly from JSON files. It makes the data ready for immediate use without additional rendering or processing. This method speeds up the system and uses less space to run it.

Deploying the prototype on green energy-running servers, recognised by the Green Web Foundation, guaranteed green hosting, and, along with all the aforementioned applied strategies, resulted in a website that barely has an environmental impact.

Figure-1 and Figure 2 depicts a screenshot of the original website and the developed green website home page:

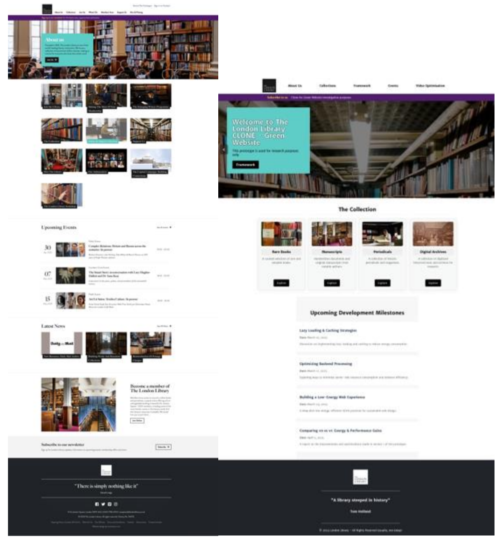

Figure 1-Original website home page; Figure 2-Developed website home page

V. FINDINGS AND DISCUSSION OF RESULTS

Results of the comparison are available in the Figure-3 and Table-2 . The London Library's website was compared with the prototype designed for sustainability. The developed model massively outperforms the library's website in area such as: speed, efficiency, performance , overall resource consumption and memory..

The comparison criteria are derived directly from Google Lighthouse metrics, offering a comprehensive and quantifiable assessment of a web platform's user experience, technical performance, and adherence to best practices.

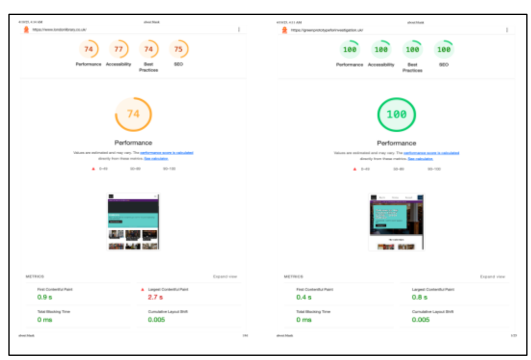

Figure 3-Lighthouse results for the original website and developed website

Criteria Evaluated	London Library's Website	Investigation Prototype
Performance Score	74 (Moderate)	100 (Excellent)
Accessibility Score	77 (Average)	100 (Perfect)
Best Practices Score	74 (Needs improvement)	100 (Best practice)
SEO	75 (Average)	100 (Excellent)
First Contentful Paint	0.9 s (Acceptable)	0.4 s (Very fast)
Largest Contentful Paint	2.7 s (Slow)	0.8 s (Optimised)
Total Blocking Time	0 ms (Good)	0 ms (Excellent)
Cumulative Layout Shift	0.005 (Stable)	0.005 (Stable)
Speed Index	5.9 s (Laggy)	0.6 s (Excellent)
Image Optimisation	Poor (Potential savings: ~55,019 KiB)	Optimised (Savings: ~30 KiB)
Unused JavaScript	High (Potential savings: ~196 KiB)	Minimal (Savings: ~97 KiB)
JavaScript Minification	Not minified (Performance impacted)	Minified (Efficient)
CSS Minification	Not minified (Wasteful)	Minified (Clean)
DOM Size	Large (Excessive - 627 elements)	Small (Efficient - 82 elements)
Security (CSP, HSTS, COOP)	Mixed: lacks HTTP/2 on many assets	Secure (uses HTTP/2)
Server Response Time (TTFB)	1,150 ms (Slow)	100 ms (Very fast)
Network Payload	~100,000 KiB (Very High)	302 KiB (Optimised)
Third-party Impact	High (Many 3rd-party assets)	Minimal 3rd-party use
Critical Request Chains	Long chains (High latency)	Short chains (Low latency)
Largest Image Load Time	Several >10 MB	~40 KiB
Running Environment	Online (https://www.londonlibrary.co.uk/)	Online (https://greenprototypeforinvestigation.uk/)

Table 2-Lighthouse comparison

As shown in the table, First Contentful Paint (FCP) dropped from 0.9s to 0.4s, that means users see content faster. The Largest Contentful Paint (LCP) was cut from 2.7s to just 0.8s, making the site feel way more responsive. The Speed Index improved from 5.9s to 0.6s, proving the site loads noticeably faster.

From the resources perspective. the developed website fixed major inefficiencies:

- Image optimisation was in the original (~55,019 KiB wasted), but in the prototype, it reduced to ~30 KiB, which is a drastic improvement.
- JavaScript bloat was tackled, cutting unused scripts from ~196 KiB to ~97 KiB.
- CSS files who were previously unminified and bloated are now clean and compressed.
- DOM size went from 627 elements to 82 elements, making rendering smoother and less power-hungry.

The biggest impact can be seen on the server response time (TTFB) which dropped from a sluggish 1150ms to 100ms. However, security partial improvements, due to some key protections (COOP, HSTS) that are not implemented in the developed website. Furthermore, the critical request chains went from long chains down to short chains, cutting load delays drastically.

The network payload shrank from ~100,000 KiB to 302 KiB, this indicates that the prototype loads with a fraction of the bandwidth, which is a big step for the sustainability purposes of the research.

Largest image load time dropped from several >10 MB to just ~40 KiB, this is a result of lazy loading, smarter compression strategies and using new advanced formats. By applying lazy loading, optimised media formats, caching strategies and script minimisation, we can create websites that look the same but perform significantly better, with far less energy waste.

Table-3 shows comparative environmental Impact report using the green analysis tools.

Indicator / Tool	Investigation Prototype	Original website
Website Carbon Calculator		
CO2 per pageview	0.06g	21.93g
Green hosting check	Yes	Yes
Estimated annual emissions (10,000 views)	6.69 kg CO_2	219.3 kg CO_2
Emissions equivalence	1 tree / 112 km electric car	~10 trees / high electricity use
EcoIndex.fr		
Score / Rating	87 / A	33 / E
Page weight	0.623 MB	107.6 MB
Number of server requests	14	120
DOM complexity (elements)	107	669
CO2 (monthly estimate)	1.26 kg CO_2	2.34 kg CO_2
Water consumption estimate	18.9 litres	35.1 litres
Ecograder		
CO_2 per pageview	0.09g	30.68g
Page weight	255.5 KB	90.95 MB
Image optimisation	100/100 (Excellent)	44/100 (Poor)
JavaScript/CSS optimisation	75–100/100 (Very good)	25–50/100 (Needs work)
UX/design impact on emissions	89/100 (Excellent)	51/100 (Weak)
Green hosting check	Yes	Yes
Caching efficiency	Partial: 50/100 (return = 0.027g)	Inefficient: 50/100 (return = 32.386g)
Beacon		
CO2 per pageview (new visit)	0.079g	32.603g
CO2 per pageview (return visit)	0.027g	32.386g
Page weight (new visit)	255.68 KB	102.61 MB
DOM complexity	95 elements	~700 elements
Core Web Vitals – CLS	0.006	0.007
Core Web Vitals – FID	30ms	110ms
Caching effectiveness	~66% CO_2 reduction	<1% difference
Emissions equivalence	1 hr Netflix, 0.82 mi Tesla, 1 tree/year	127 hrs Netflix, 558 mi, 16 trees/year
SitiGreen.it		
CO2 per page view	0.06g	25.01g
Page weight	255.77 KB	102.61 MB
Green hosting check	Yes	Yes

Table 3 - Comparative Environmental Impact Report

This evaluation assessed the environmental sustainability of the two websites by using five widely recognised tools. These tools analyse various aspects of a website's design, performance and infrastructure to estimate their environmental impact in terms of carbon emissions, energy use and resource efficiency.

The following key areas were assessed:

1. **Carbon Dioxide (CO_2)** Emissions per Pageview. Each tool provided an estimate of the amount of CO2 emitted every time a user loads the webpage. This figure is calculated based on page weight, data transfer energy and the power of the hosting server.
2. **Green Hosting.** The tools checked whether the website is hosted on servers powered by renewable energy or carbon offset infrastructure.
3. **Page Weight and File Size**. The web page size—including all media, scripts, and stylesheets—was measured. Heavier pages require more data transfer and therefore more energy to load.
4. **Number of Server Requests.** This refers to how many times the browser needs to fetch external resources (e.g., images, fonts, scripts). Fewer requests generally indicate better optimisation and lower energy usage.
5. **DOM Complexity (Number of HTML Elements).** Many DOM elements suggest a complex layout, which increases processing requirements and negatively impacts performance and energy efficiency.
6. **JavaScript and CSS Optimisation**. The degree to which code has been minified, consolidated, and efficiently used was assessed, affecting both speed and environmental impact.
7. **Core Web Vitals (CLS-Cumulative Layout Shift, FID-First Input Delay).** These metrics were evaluated to reflect user experience quality and the page's rendering efficiency.
8. **UX and Design Efficiency**. Tools examined how effectively the site helps users achieve tasks without unnecessary loading or content, reducing the carbon footprint of user devices.
9. **Caching Efficiency.** This shows how well the website uses browser caching to reduce

repeat download costs, thus lowering emissions for returning visitors.

10. Estimated Annual Impact. Based on assumed traffic, tools provided a projected annual carbon footprint, expressed in kilograms of CO_2 and in relatable equivalents (e.g., car journeys, trees).
11. **Water Consumption** (provided only by Ecoindex). Some tools estimated the amount of water used indirectly to power the infrastructure behind each website visit.
12. **Emissions Equivalence**. The results were contextualised to make them easier to understand by comparing the site's emissions to everyday activities such as driving a car, charging a smartphone or brewing tea.

Each tool offered slightly different methodologies and strengths, some focused more on frontend optimisation and user experience (e.g., Ecograder, Beacon). In contrast, others prioritised carbon metrics and ecological indicators (e.g., EcoIndex, Website Carbon Calculator, SitiGreen).
This approach provided a well-rounded picture of technical efficiency, environmental cost and sustainable design practices across both websites.
Figure 4 presents a recommended framework for developers, outlining best practices derived from this research. It encompasses green front-end and back-end technologies, along with an online deployment strategy that supports environmentally sustainable computing.

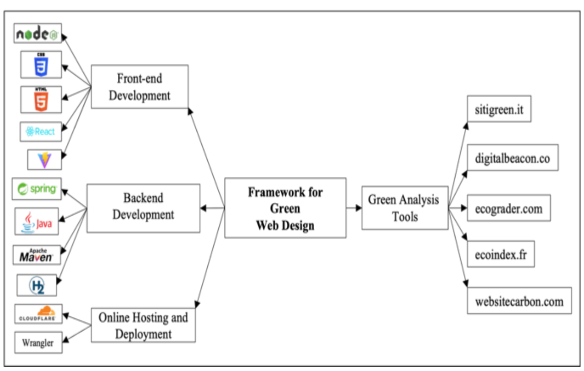

Figure 4-Suggested Practical Framework

VI. CONCLUSION

This research set out to investigate the principles and practical impact of green website design through the development and validation of a custom sustainability framework. By comparing a high-impact existing website (londonlibrary.co.uk) with an optimised clone (greenprototypeforinvestigation.uk), the findings clearly demonstrate that sustainable design choices at the front-end level such as reduced page weight, optimised assets, efficient caching, and DOM simplification can yield reductions in carbon emissions and environmental footprint.

Despite both websites being hosted on infrastructure recognised as green or carbon-offset, the optimised clone produced over 99% fewer CO_2 emissions per visit, used significantly less water, and outperformed the original across all Lighthouse metrics. This validates a central thesis of this research: green hosting is not enough. True digital sustainability must include performance-driven, user-centric, and minimalist design practices.

Therefore, the framework proposed in this study is not only theoretically sound, but practically grounded offering actionable guidelines for developers, designers, and organisations seeking to build environmentally responsible digital platforms.

REFERENCES:

1. O. Hulleberg, H. L. Granum, S. G. Hansen, M. Moen, C. V. Monllaó, and Y. Inal, “The awareness and practices of web developers toward sustainable web design,” in *Proc. Int. Conf. Human-Computer Interaction*, Cham, Switzerland: Springer Nature, 2023, pp. 134–145. M. Willis, J. Hanna, E. Encinas, and J. Auger, “Low power web: Legacy design and the path to sustainable net futures,” in *Extended Abstracts of the 2020 CHI Conf. Human Factors Comput. Syst. (CHI EA '20)*, New York, NY, USA: ACM, 2020, pp. 1–14. [Online]. Available: https://doi.org/10.1145/3334480.3381829
2. M. Arora, I. McClenaghan, and L. Wozniak, “Priorities for net-zero web services,” *Nature Electronics*, 2024. [Online]. Available: https://doi.org/10.1038/s41928-024-01227-8
3. M. González-Sordé, “Exploring the potential of easy language for enhancing website sustainability,” *Universal Access Inf. Soc.*, pp. 1–11, 2024. [Online]. Available: https://doi.org/10.1007/s10209-024-01129-8. B. Dornauer and M. Felderer, “Energy-saving strategies for mobile web apps and their measurement: Results from a decade of research,” *arXiv preprint arXiv:2304.01646*, 2023. [Online]. Available:

https://doi.org/10.48550/arXiv.2304.01646. T. Černý and M. Donahoo, "On energy impact of web user interface approaches," *Cluster Comput.*, vol. 19, 2016. [Online]. Available: https://doi.org/10.1007/s10586-016-0665-7.

4. Connective Web Design, "Sustainable web design," [Online]. Available: https://connectivewebdesign.com/blog/sustainable-web-design. [Accessed: Apr. 25, 2025]. B. Bhanderi, "8 web app design tips to boost user experience in 2025," Creole Studios, 2024. [Online]. Available: https://www.creolestudios.com/web-app-design-tips-for-2025/. [Accessed: Apr. 29, 2025]. Mightybytes, "EcoGrader," [Online]. Available: https://ecograder.com. [Accessed: Apr. 29, 2025]. The Digital Beacon, "The Digital Beacon," [Online]. Available: https://digitalbeacon.co. [Accessed: Apr. 29, 2025]. Piano D, "SitiGreen," [Online]. Available: https://www.sitigreen.it/. [Accessed: Apr. 29, 2025].
5. Wholegrain Digital, "Website carbon calculator," [Online]. Available: https://www.websitecarbon.com/. [Accessed: Apr. 29, 2025]. GreenIT.fr, "EcoIndex," [Online]. Available: https://www.ecoindex.fr/en/. [Accessed: Apr. 29, 2025].
6. The Green Web Foundation, "The Green Web Foundation," [Online]. Available: https://www.thegreenwebfoundation.org. [Accessed: Apr. 29, 2025].