Digital Disparities: A Comparative Web Measurement Study Across Economic Boundaries

Anonymous Authors

Abstract

While internet usage is slowly catching up globally, it is still unclear how the web experience differs in developing and developed countries. On the one hand, the web has a notoriously large inertia, with many webpages still relying on unencrypted HTTP, deprecated web features, or old and buggy libraries. On the other hand, developing countries are expected to leapfrog and directly adopt the newest technologies by learning from the prior mistakes of more developed countries. Anecdotal evidence suggests that webpages in developing and developed regions differ significantly. In this work, we test this hypothesis by measuring differences in web development practices across the two groups of countries, using multiple dimensions: webpages' size, complexity, security, privacy, quality, technology adoption, and accessibility. Concretely, we collect the largest dataset to date that compares web development practices across developed and developing regions – 200,000 webpages across 20 countries – which we aim to open source along with this publication. Our findings reveal that webpages in developing regions are generally smaller and simpler, utilizing fewer requests — an adaptation that improves the performance over slower network conditions common in these areas. However, these sites are less optimized in other critical aspects: they frequently employ inefficient image formats, include unnecessary JavaScript or CSS code, or lack responsive image designs. Notably, our security assessment shows developing regions lagging in HTTPS adoption and vulnerability mitigation, possibly due to lower awareness of best practices.

1 Introduction

According to the most recent 2023 data from International Telecommunication Union[1] and the World Bank[2], 67% of the global population uses the internet, with a 5% growth in 2023 alone. However, there are large discrepancies in internet adoption based on the economic situation of a country, *i.e.,* while 93% of the developed world is online, only 60% of the developing world is. Moreover, due to the limited availability of fixed broadband services and the prohibitive costs associated with them, users from developing countries mostly rely on mobile internet [3, 15]. This measurable inequality – also called digital disconnect [47] or digital divide [56] – is a missed opportunity for economic growth in developing countries [40].

It is commonly believed that digital inequality also extends to the web. For example, a recent work by Ruth et al. [66] finds that most websites visited in a country are specific to that country, with only few popular global websites. Moreover, the World Bank [40] notes that developed countries tend to host more web-based services. On the contrary, the lack of broadband [23, 41] and modern devices in developing countries might lead to web development optimizing for low-bandwidth conditions, e.g., with more simplistic designs and supported features. Still, websites in these regions are often criticized for poor optimization, resulting in slow load times and higher data usage [26, 71]. However, developing countries can leapfrog [25, 37] and adopt the latest technologies, such as WebAssembly, without the ossification observed for older webpages [50]. Also, users, developers, and regulators in developing countries might be insufficiently concerned with security and privacy.

To the best of our knowledge, no large-scale empirical study has systematically analyzed web development practices across different economic contexts [48]. Most existing research focus on macro-level indicators such as internet penetration rates, smartphone adoption, or general technological infrastructure [59, 69]. To address this research gap, we conduct a large-scale analysis comparing webpages from both developed and developing countries. Our study involves identifying and crawling 200,000 popular webpages, *i.e.,* 10,000 from the top 10 most populated developed and developing nations, using tools like Google Lighthouse for performance audits and Puppeteer to capture detailed network data. This analysis considers multiple web metrics, which we use to explore websites' performance, security, privacy, and technology adoption, as summarized below:

Webpage Size and Complexity: We assessed webpage size, HTTP requests, and multimedia usage to evaluate load times and data consumption. Websites in developing countries are generally smaller and simpler, optimized for slower networks. However, inefficiencies in image formats and JavaScript persist, highlighting opportunities for further optimization despite their reduced size.

Performance Optimization: We analyzed websites' performance by assessing the use of optimized image formats or responsive images or the inclusion of unnecessary JavaScript and CSS files. Despite smaller sizes, websites in developing countries waste more bandwidth due to poorly optimized images and unused code, lacking advanced optimizations, which might hinder user experience.

Security Measures: We evaluated security practices by examining HTTPS usage, outdated libraries, and CSP implementation. Developed countries have adopted HTTPS more widely, but surprisingly, they also contain a higher number of severe security vulnerabilities. Both groups show a widespread lack of CSP policies, leaving websites vulnerable to threats like cross-site scripting.

Privacy Practices: We analyzed tracking scripts and third-party cookies to assess privacy concerns. Websites in developed countries use more trackers and cookies, likely due to data-driven business models. Despite stricter privacy regulations, tracking remains widespread via consent mechanisms. Developing countries have fewer trackers, but weaker privacy regulations remain a concern [19].

Technology Adoption: We examined the adoption of modern web APIs and protocols to assess if developing countries leapfrog older technologies. Surprisingly, some developing countries adopt HTTP/2 and HTTP/3 at rates comparable to or higher than developed ones, likely due to fewer legacy systems. Both groups, however, show similar web API adoption.

[1]https://datahub.itu.int/data/?e=1&i=11624
[2]https://datacatalog.worldbank.org/indicator/715ee78d-bfce-eb11-bacc-000d3a596ff0/Individuals-using-the-Internet----of-population-

2 Related Work

Multiple studies have quantified cellular and broadband connectivity in developing regions, performing measurements, and proposing solutions to improve connectivity. Hassan et al. [44–46] investigated cellular connectivity in rural countries and how solutions such as Software Defined Networking can provide affordable connectivity. Similarly, Marshini et al. [31] explored mobile broadband connectivity in South Africa, while Zaki et al. [70] investigated connectivity challenges in Ghana. Fanou et al. [34] further highlighted the disparities in web content infrastructure across Africa. Research by Koradia et al. [51], Sharma et al. [67], and Bischof et al. [24] focused on cellular network issues in India and Cuba. Beyond connectivity, many research papers propose concrete web optimizations: Lite-Web [27], for example, aims to reduce web complexity through tools such as SlimWeb [28], JSCleaner [29], and Muzeel [53].

Web performance in developing regions are worsened by the diffusion of low-end devices. Naseer et al. [57, 58] noted that JavaScript execution often triggers memory constraints, while rising web complexity and JavaScript-based tracking inflate costs for users. Amjad et al. [21] showed that selective JavaScript blocking can help, though with challenges. Additional research has tackled web accessibility in these areas. Raza et al. [63] proposed enhanced caching to compensate for limited cloud infrastructure, while Ahmad et al. [20] advocated for a new content ecosystem tailored to local needs, essential for users in resource-constrained environments.

Google's AMP [38] and Facebook Lite [4] aim to address web complexity in developing regions. AMP provides a framework for faster web page construction, while Facebook Lite is optimized for low-end phones and slow networks. Google also launched Web Light [8], which transcoded webpages to lighter versions for slow connections, though it was discontinued in 2022 due to functionality issues. Despite these efforts, web complexity continues to impact users in developing regions, who rely on low-end devices and often face poor web performance. Finally, Habib et al. [43] highlighted that the increasing complexity of the web intensifies affordability concerns, restricting users' access to internet services. Their work introduced a fairness metric to redesign the web with a focus on affordability and inclusivity. Unlike [42, 43], we analyze and explore 200k popular web pages, from the most populated developed and developing regions, in terms of their performance, security, privacy, and technology adoption. Additionally, in comparison to [66], which focuses on web traffic distribution and the most visited categories, we delve into the finer details of these pages, comparing factors such as image formats, unnecessary JavaScript usage, and security measures.

3 Methodology

Our methodology combines Google Chrome User Experience (CrUX) dataset [9] with the International Monetary Fund (IMF)'s classification, and with web data (webpage composition, protocol usage, etc.) we crawled using Google Lighthouse and Puppeteer. IMF classifies countries as "advanced" and "emerging market and developing economies" based on per capita income, export diversification, and financial system integration. CrUX data provides real-world web performance metrics like page load times and responsiveness, offering insights into website performance across regions.

Country Selection Criteria. The countries in this study are selected to ensure a balanced representation of both developing and developed economies, following the classifications provided by the IMF. Specifically, we select a sample of 10 developing and 10 developed countries for a comparative analysis of their web ecosystems. The selection process is based on two primary criteria: population size and web data availability within the CrUX database [9].

For developing countries, we identified the ten most populous nations from a cohort of 152 economies classified as "developing" by the IMF [18]. Population size is chosen as the primary criterion to ensure that the selected countries represent a significant portion of the global population within the developing world. Ethiopia, though the 10^{th} most populous developing country[3], was excluded due to insufficient CrUX data that met our inclusion criteria. To complete our sample, we included the Philippines in place of Ethiopia. The ten developing countries selected, in order of population, are: China, India, Indonesia, Pakistan, Brazil, Nigeria, Bangladesh, Russia, Mexico, and the Philippines.

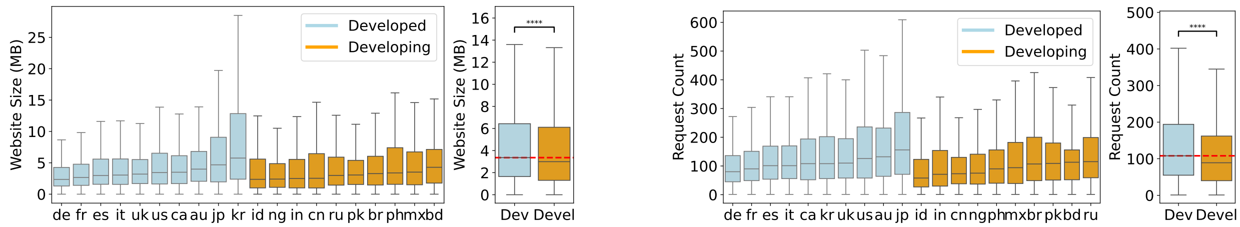
For the developed countries, we focused on the most populous nations classified as "Advanced Economies" by the IMF. This selection ensured that major global economies were adequately represented in our dataset. The ten developed countries, ordered by population, are: the United States, Japan, Germany, France, the United Kingdom, Italy, South Korea, Spain, Canada, and Australia.

Together, the developing countries in our dataset account for a combined population of 4.3 billion people, representing 54.33% of the global population [17]. In comparison, the developed countries in our sample account for 906.01 million people, or 11.3% of the world's population. By using population size and CrUX data availability as selection criteria, our study captures a significant portion of the world's population across both economic classifications, providing a robust foundation for comparing the web ecosystems of developing and developed nations.

Website Selection. To investigate the digital landscape in each country, we retrieve data on 10,000 websites per country from the CrUX dataset. A website is categorized under a particular country if it either has a country code top-level domain (ccTLD) for that country (e.g.,.de for Germany) or if its Whois [33] data indicates a physical registration address in that country. This criterion was used to mitigate the over-representation of globally dominant websites, which could skew the analysis of local web ecosystems. We utilized the `python-whois` package and the `who.is` API [16] to retrieve registration information for the websites. We start from the 10,000 most popular websites for each country – as indicated by the CrUX list – and then apply our selection criteria to filter the relevant websites for analysis. This approach ensures a uniform number of websites for each country, and a comparable distribution of website rankings. In most cases, our 10,000 websites were within the top 50,000 of that country. However, for Bangladesh, Pakistan, Nigeria, and the Philippines, we had to include websites that were less popular to meet our criteria and target sample size. Please refer to appendix A.1 for a distribution of website rankings.

Crawler Configuration. We crawl the selected websites using Google Lighthouse [7] for performance audits, while simultaneously employing Puppeteer [10] to intercept and log network

[3]https://www.worlddata.info/developing-countries.php



(a) Website Size Comparison

(b) Request Count Comparison

Figure 1: Comparison of website size (a) and request count (b) between developed and developing countries.

requests and responses, capturing detailed data such as request headers, responses, and metadata. In the presence of unreachable websites, e.g., due to temporary outages, Lighthouse timeouts, certificate errors, or other technical problems, we supplemented the list with lower-ranked (less popular) websites. For the Lighthouse audits, we emulated an iPhone X mobile environment, with a screen resolution of 375x812 pixels and a device scale factor of 3, ensuring that the mobile version of each page was loaded. This configuration was chosen to reflect common mobile browsing conditions and to avoid bias due to default client configuration as observed in [49]. We run our crawlers across three server machines hosted at two university campuses (Germany and UAE). This selection was motivated by the minimal crawling bias observed for university IP addresses when compared to, for instance, cloud IP addresses [49].

Library Detection. To analyze the usage of JavaScript libraries across the selected websites, we employ multiple methods for library detection. We use the retirejs Python library [12], custom regular expressions to extract library information from request URLs, and data from Google Lighthouse audits. This approach helps capture a broader range of library usage across websites.

Limitations. Our crawler operates from two locations, Germany and UAE, for scalability, though Germany's inclusion as a test country may introduce bias. Ideally, crawlers would run within each studied country, but logistical constraints made this unfeasible. VPNs were also unreliable, as VPN detection could skew results, and many VPN providers rely on cloud servers that lack tracking domains and JavaScript libraries, as noted in prior studies [49]. Since our crawlers don't operate within most of the studied countries or use realistic mobile network conditions, they cannot fully capture Web performance metrics tied to user experience. We mitigate this by using CrUX data, which provides real-world performance metrics from actual devices and networks across all countries studied.

Google Lighthouse also has some limitations. By default, it does not handle cookie banners, which can prevent full webpage rendering. While there are potential solutions [11], their reliability is untested, and we chose not to include them. Additionally, Lighthouse cannot handle user authentication, potentially affecting performance and security measurements [62]. Finally, our library detection is not fully sound, as retirejs may miss libraries, particularly in bundled files, which are challenging to identify [61].

4 Size and Complexity

4.1 Static Footprint

We measure a webpage *size* as the total number of bytes required to load all its content. Figure 1(a) shows that the median website size in developed countries is 3.35 MB, *i.e.,* slightly larger than the 2.99 MB median size in developing countries, both of which exceed the global median of 2.6 MB as per the HTTP archive [22]. Similarly, websites in developed countries require a median of 108 requests to load (Figure 1(b)), compared to 89 requests in developing countries – while the global median is 70 requests. These differences between developed and developing countries are statistically significant($p < 0.001$), highlighting the larger footprints of websites in developed countries. South Korea stands out with exceptionally large websites – median size of 5 MB and upper whisker close to 30 MB – due to higher multimedia content and interactive features supported by the country's fast internet speeds [14].

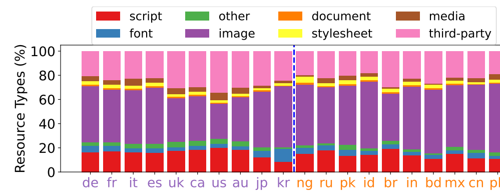


Figure 2: Bandwidth allocation for different resource types.

This result contrasts with earlier findings that focused on public service websites [42], where websites in developing countries were significantly larger and less optimized. In our broader comparison of general websites, however, the opposite is true: websites in developing countries have smaller footprints and fewer resource requests, indicating they are more suitable for slower networks. Despite this, both developed and developing countries exceed the global median in terms of website size and number of requests, suggesting further optimization could reduce data costs, especially for users with limited access to affordable internet services.

4.2 Breakdown by Resource Type

To understand which webpage characteristics contribute to the larger sizes of websites in developed countries, we analyze the sizes of different resources across pages. We distinguish between HTML documents, images, JavaScript, and third-party content, *i.e.,* any resource imported from a third party domain [72]. Figure 2 shows a

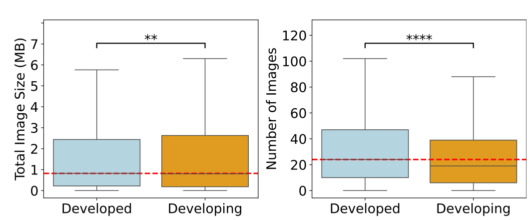
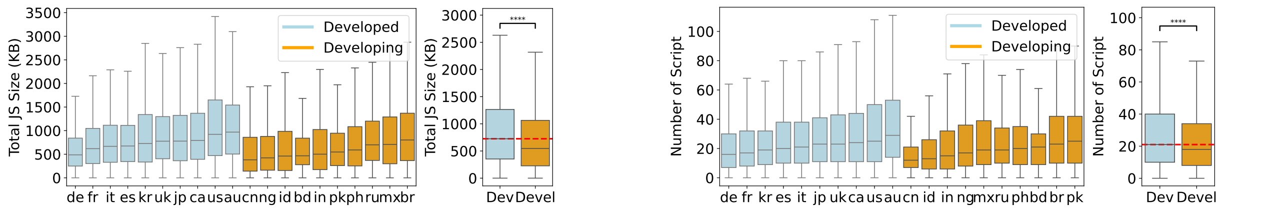


Figure 3: The sum of all image sizes and the number of image requests across developed and developing regions.



(a) Total JS Size Across Countries

(b) Number of JS Requests Across Countries

Figure 4: Comparison of total JS size (a) and number of JS requests (b) between developed and developing countries.

clear contrast in how websites from these two regions allocate bandwidth across webpage components, particularly images. Websites in developing countries dedicate a much larger share of their page size to images, ranging from 48% to 57%. For instance, websites in Indonesia allocate 55.3% of their page size to images, compared to websites in developed countries like the US (28.8%), the UK (36.3%), and Canada (37.2%), where image bandwidth usage is < 40%.

Figure 3 further focuses on image data usage, revealing that websites in developing countries devote a larger proportion of their bandwidth to images, even though they make fewer image requests (19 on average) than websites in developed countries (24 on average). Notably, websites in developing regions allocate more than 50% of their bandwidth to images, averaging 822.75 KB, compared to websites in developed regions, which allocate less than 40% and average 839.62 KB. This indicates that the higher share of image-related bandwidth in developing countries stems from larger image files rather than a larger number of image requests.

By analyzing image size distribution, we find that, indeed, websites in developing countries use significantly larger images compared to those in developed countries (see appendix A.2). For example, 15% of images on websites in developing countries exceed 100 KB, whereas this is the case for only 12% of images in developed countries. This reinforces the observation that websites in developing countries do not host more images but rather use larger images, which increases overall website size and bandwidth consumption. This leads to potential negative effects, particularly for users in regions with slower internet speeds and limited data plans.

4.3 Dynamic Footprint

We here focus on JavaScript (JS), the dominant factor in a website's performance [30], with Chrome spending over 30% of its processing time on it [2]. Figure 4(a) shows the difference in the total size of JS usage across various countries. The median JS size for websites in developed countries such as the US and Australia is 921.71 KB and 968.61 KB, respectively, while in developing countries like Bangladesh and Pakistan, the median JS size is significantly lower, at 466 KB and 547 KB. This trend is consistent across regions, with developed countries generally using more JavaScript code. The overall median JS size for developed countries is 725 KB, compared to 546 KB for developing countries. This shows a clear difference in JS usage, suggesting that websites in developed countries rely more on JS to create richer, more interactive user experiences.

Further supporting this trend, Figure 4(b) shows that websites in developed countries make more JS network requests than those in developing regions. For example, the median number of requests for websites in the US and Australia is 25 and 29, respectively, while in developing nations like Bangladesh and Pakistan, the median number of requests is 21 and 25, respectively. Overall, the median number of JS requests is 21 for developed and 18 for developing countries. This difference highlights again the greater reliance on JS in developed countries. When combined with the median document size (see appendix A.3), which includes all HTML files, where developed countries have a median size of 33.6 KB compared to 26.1 KB in developing regions. It becomes evident that the increased JS usage contributes to the overall websites' size difference.

Figure 5 shows that this complexity is also reflected in the DOM size, or number of DOM elements in a webpage, which represents the structure and complexity of the page. A larger DOM size generally indicates a more complex webpage with more interactive elements, often driven by extensive JS usage. For example, the median number of DOM elements in developed countries like Germany (1,283) and Italy (1,058) is significantly higher than in developing countries such as Pakistan (940) and Russia (926). The overall median DOM size for developed countries is 969 elements, while in developing countries, it is much lower at 633 elements. This larger DOM size in developed countries highlights the increased complexity and interactivity of websites, which is often due to the heavier use of JS to enhance dynamic content and features.

The higher complexity of webpages in developed countries, as indicated by larger document sizes and DOM elements, also impacts browser performance, particularly in terms of *total blocking time (TBT)*, a metric that measures how long a browser's main thread is blocked by tasks, preventing users from interacting with the page. Websites in developed countries have a significantly larger TBT, with median values of 631.0 ms compared to 418.5 ms in developing countries (see appendix A.4). This longer TBT is a direct consequence of larger amounts of JS and HTML code, which must be interpreted before the website becomes fully interactive.

5 Performance Optimizations

5.1 Image Compression

Optimizing image formats and using properly sized images are critical for improving web performance, particularly in regions

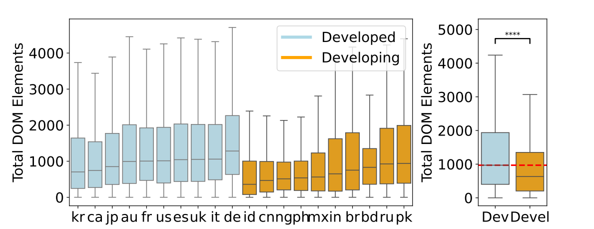
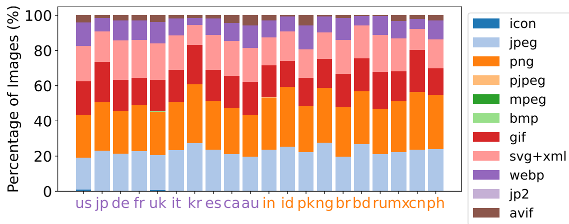
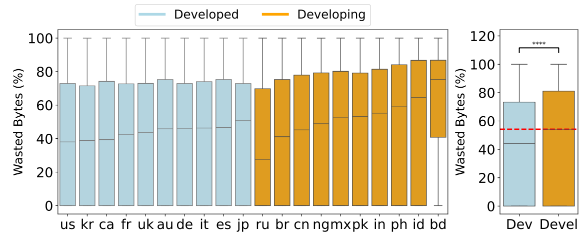


Figure 5: Total number of DOM elements across countries.



(a) Distribution of image format by country.



(b) Wasted image bytes (%) from compression inefficiencies by country.

Figure 6: Comparison of image format distribution (a) and percentage of wasted bytes due to compression inefficiencies (b) across different countries.

with bandwidth constraints. Modern optimized formats such as WebP, AVIF, and SVG are recommended by best practices due to their superior compression capabilities and ability to maintain high image quality with smaller file sizes [13]. While JPEG 2000 (JP2) is not recommended, studies have shown that this format outperforms traditional JPEG in terms of compression efficiency [5].

Figure 6(a) shows the analysis of image formats across countries, which reveals disparities in the adoption of these optimized formats. Developed countries have a higher usage of modern formats like WebP, AVIF, and SVG, contributing to faster page loads and more efficient data usage. For instance, in the United States, modern formats account for approximately 37% of image usage, with SVG at 20.03%, WebP at 13.50%, and AVIF at 3.81%. Traditional formats like JPEG and PNG account for 17.90% and 24.52%, respectively. Similarly, Germany shows substantial adoption of modern formats, with SVG at 22.45%, WebP at 11.33%, and AVIF at 2.69%, totaling about 36% of modern format usage. In contrast, developing countries have lower adoption rates of modern formats; in India, modern formats constitute approximately 28% of image usage—SVG at 15.91%, WebP at 9.64%, and AVIF at 2.68%—while traditional formats like JPEG and PNG dominate at 23.24% and 29.62% respectively. Nigeria exhibits even lower modern format usage at around 24%, with SVG at 14.95%, WebP at 9.02%, and AVIF at 0.52%, whereas JPEG and PNG account for 27.48% and 31.13%.

Next, we analyze image compression wastage, which refers to the proportion of image file sizes that exceed what is necessary for maintaining visual quality. Lighthouse computes this by comparing the size of the rendered image against the actual image size, accounting for the device's pixel ratio to assess whether images are optimally sized. Figure 6(b) shows that developed countries generally have lower image compression wastage rates compared to developing countries. Developed countries like the United States, South Korea, and Canada, which have median wastage rates of approximately 38-39%, also exhibit higher usage of modern formats such as WebP and AVIF in Figure 6, contributing to better management of image sizes and reducing wastage. In contrast, developing countries like India and Bangladesh, with higher wastage rates of 55% and 75%, respectively, show a lower adoption of optimized formats, indicating that inefficient image formats and large file sizes are more commonly used, particularly on smaller devices. This correlation between format usage and wastage emphasizes the importance of adopting optimized formats to improve performance.

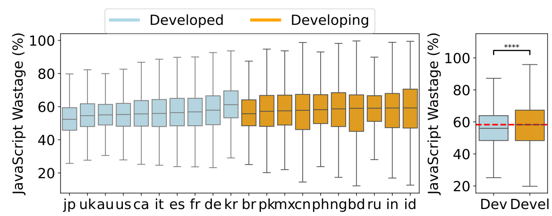


Figure 7: Distribution of unused JavaScript across countries.

5.2 Unused JavaScript and CSS Code

The unused JavaScript (JS) audit highlights code that is downloaded and parsed but never executed, contributing to significant performance inefficiencies. Although this code isn't directly used during page load, it still incurs costs as browsers must download, parse, and sometimes compile it. While tools like Lighthouse help identify this issue, they may overestimate the amount of unused JS by not fully simulating user interactions that might trigger the code later.

Several factors contribute to the presence of unused JS, including the use of pre-built libraries, legacy code, and a lack of regular optimization. Developers often include entire libraries or modules even when only small portions are needed. Additionally, outdated or redundant code can accumulate on websites, remaining in the assets due to limited maintenance or the complexity of refactoring. The increasing reliance on large JS frameworks exacerbates this issue, as these frameworks are often not tailored to the specific needs of individual webpages. Unused JS significantly impacts performance, especially on mobile devices. Mobile phones, particularly low-end models common in developing countries, require up to seven times more processing power than desktop devices to handle JS [30], further slowing down page loads. In regions with slower networks and higher data costs, the burden of downloading and processing unnecessary code is even more problematic.

On average, 55% of JS is unused across all countries, as shown in Figure 7. Developed nations like the United States (82.65%) and Japan (79.76%) show high inefficiencies, while countries like South Korea (93.69%) and Germany (92.65%) fare even worse. The problem is most severe in developing nations such as Bangladesh (99.68%) and China (98.72%), where limited network speeds and strained resources amplify the negative impact of unused JS. CSS wastage is also prevalent, with over 91% of global CSS rules remaining unused. Developing nations like Pakistan and Bangladesh exhibit wastage levels above 95%, while developed countries like Japan and Germany report slightly lower rates, around 92%. This wastage is likely due to outdated frameworks and poor mobile optimization, as discussed further in Appendix A.6.

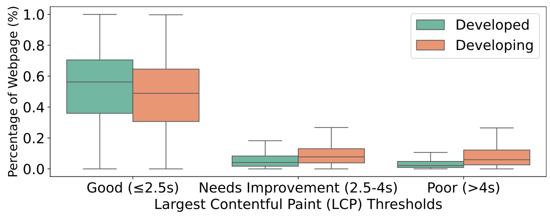


Figure 8: Distribution across developing and developed regions of the percentage of mobile visits which were Good, Needs Improvement, or Poor.

It is crucial to address these inefficiencies, especially in developing regions where slow networks and low-end devices exacerbate the issue. Reducing unused JS and CSS through better development practices will improve performance, particularly for users in regions with limited network and device capabilities.

5.3 Webpage Performance

We conclude the section by analyzing the quality of browsing experience for users in both developed and developing regions. Using the CrUX dataset [9], which anonymously reports web performance metrics from mobile users, we focus on the Largest Contentful Paint (LCP) [6]. LCP measures how quickly the main content of a webpage becomes visible, a key metric recommended by Google as a reliable indicator of page load performance [6]. LCP performance is classified into three categories: Good ($\leq 2.5sec$), Needs Improvement (2.5-4 sec), and Poor ($\geq 4sec$) [1]. CrUX reports the percentage of mobile visits in each country that fall into these categories for the pages under test. Although CrUX does not disclose the total number of reports for privacy reasons, we estimate a minimum of 70.2 million reports [36]. This provides valuable insights into real-world user experiences across different regions.

For visibility reasons, Figure 8 aggregates the results across developing and developed regions. Accordingly, it shows the distribution across developing and developed regions of the percentage of mobile visits which were Good, Needs Improvement, or Poor. For example, 50% of the developed webpages have close to 60% of their mobile visits, which were "Good", whereas this number drops to 50% of the developed regions. Conversely, developing regions have overall more webpages characterized by either "Needs Improvement" or "Poor" performance. This analysis confirm that users in developing regions experience a slower web, however the impact is less dramatic than expected. This is likely due to the overall smaller sizes, and lack of complex JS functionalities.

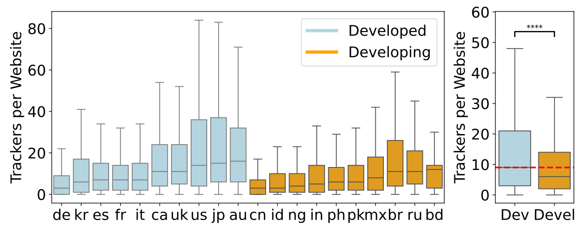
6 Privacy & Security Analysis

Trackers and Third-party Cookies. They play a crucial role in how websites collect data on user behavior, preferences, and interactions. These tools enable personalized advertising, content customization, and detailed analytics. Trackers work by embedding scripts in websites to monitor user actions, while third-party cookies allow tracking across multiple domains. However, these practices raise privacy concerns, as they allow companies to build detailed user profiles. Surprisingly, despite stricter regulations like GDPR, our analysis, as shown in Figure 9(a), indicates a counter-intuitive trend: websites in developed countries tend to use more trackers despite the presence of these regulations. For example, 70.86% of websites in Germany have trackers, while this percentage increases to 93.07% in the United Kingdom and 92.06% in Australia. Other developed countries, such as Canada (89.50%), France (83.96%), and Japan (90.41%), also show high tracker prevalence. On average, websites in developed countries employ approximately nine trackers per site. In contrast, the prevalence of trackers in developing countries shows more variability. In China, 68.01% of websites have trackers, while Bangladesh (90.77%) and Pakistan (90.26%) exhibit similarly high tracker usage. Countries such as Brazil (89.55%) and Russia (91.52%) also have widespread tracker deployment. On average, websites in developing countries use six trackers, significantly fewer than their counterparts in developed countries. The higher tracker prevalence in developed nations can be explained by the fact that these countries have more mature advertising industries and e-commerce platforms that rely heavily on user data for targeted marketing and sales optimization. Additionally, data-driven business models are more prevalent, making user data a valuable asset.

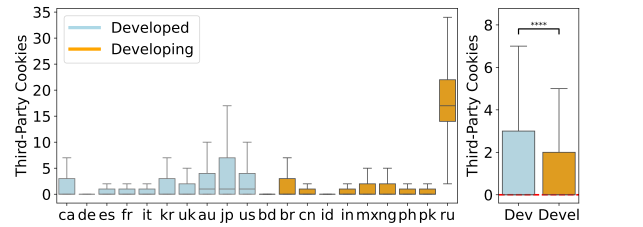
In Figure 9(b), we show the distribution of third-party cookies across both developed and developing countries, showing a low overall prevalence, likely driven by changes in browser policies, e.g., Google Chrome is planning to soon drop support for such cookies. However, Russian websites stand out with a particularly high prevalence of both trackers and third-party cookies. The median number of trackers on Russian websites is 17, and 86% of these websites employ third-party cookies. This extensive use of third-party cookies and trackers in Russia is likely driven by a strong reliance on data-driven advertising models and lack of strict privacy-related legislation. These results are consistent with those of Gotze et al. [39], who also show that almost 90% of Russian websites extensively use both trackers and third-party cookies.

HTTPS Adoption. HTTPS [64] encrypts web content protecting it from interception and manipulation. Figure 10 shows the adoption of HTTPS across both developed and developing nations. In developed countries, only 5% of websites still use unencrypted HTTP, a significant improvement from previously-reported measurements [35]. In contrast, 12% of websites in developing countries still rely on HTTP. This gap is even more concerning in certain countries, with Bangladesh and China having the most websites using HTTP: 25% and 20%, respectively (see appendix A.5). These results show a lag in adopting basic security standards in the developing world, potentially on purpose, leaving users in those regions vulnerable to (un)lawful interception.

Content Security Policy (CSP) Usage. Content Security Policies (CSPs) act as a critical defense mechanism against cross-site scripting (XSS) and other injection attacks, offering a structured way to control which resources can be loaded and executed on a webpage. However, the correct usage of CSP is a known challenge [65]. Figure 11 illustrates CSP usage rates across countries, showing a significant lack of adoption in both developed and developing nations. In developed countries, 75% of websites fail to implement a CSP. The situation is even more concerning in developing countries, where 81.5% of websites do not implement CSP at all. This widespread absence of valid and robust CSPs is not confined to specific regions; over 99% of websites globally, in both developed and developing countries, either lack CSPs altogether or use highly



(a) Prevalence of trackers across countries



(b) Percentage of websites using third-party cookies across countries

Figure 9: Comparison of the prevalence of trackers and third-party cookies across developed and developing countries.

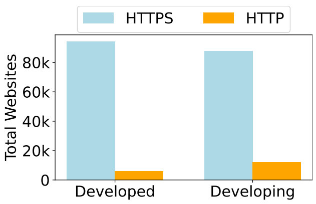


Figure 10: HTTPS adoption across regions.

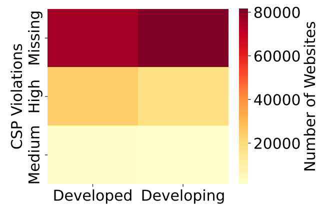


Figure 11: CSP usage rates across regions.

Figure 13: Outdated libraries usage in different countries.

ineffective deployments. These findings underscores a universal struggle with content security policies, as previously reported by developers in the recent work of Roth et al. [65].

Vulnerabilities in Web Ecosystems. When analyzing vulnerabilities, an unexpected trend emerges: websites in developed countries are more prone to severe security issues than those in developing regions. Figure 12 provides a comparison of vulnerability rates across countries, highlighting the higher incidence of critical vulnerabilities in developed nations. For example, in developed countries such as Italy, Canada, Spain, Australia, and South Korea, around 10% of websites exhibit high or critical vulnerabilities. In contrast, the highest rate in the developing world is 7%, observed in Bangladesh. This trend is particularly interesting because, despite higher security standards, developed countries experience more critical security issues. One plausible explanation for this trend is the increased reliance on JS libraries and frameworks in developed regions (see Figure 2). While JS enhances website functionality, it also increases the attack surface. The vast ecosystem of third-party libraries and dependencies, which are not always well-maintained, can contribute to the large amount of observed vulnerabilities. These results are in line with prior measurements [54, 55] that report that 40% of the websites contain a client-side vulnerability.

Outdated and Duplicate Libraries. The use of outdated and/or duplicate libraries can significantly increase a website's exposure to security risks, as these libraries may harbor vulnerabilities that attackers can exploit. Figure 13 illustrates the prevalence of outdated libraries across different countries. Developers across regions often

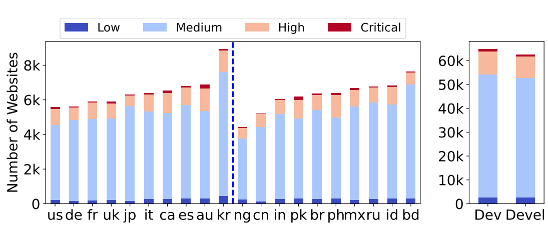


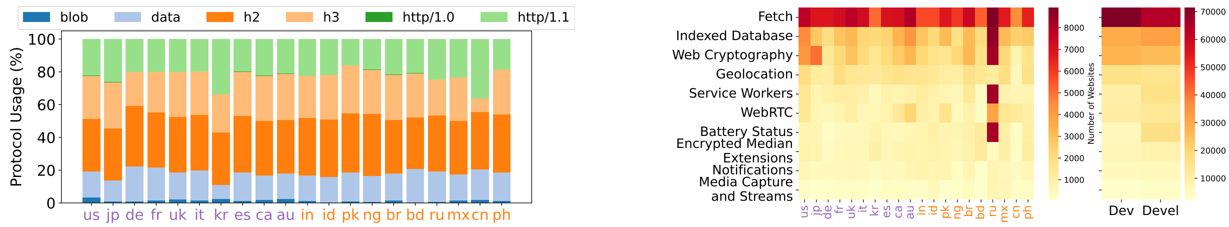
Figure 12: Number of vulnerabilities by country.

hesitate to upgrade to newer versions of libraries. These results are in line with Raula et al. [52] who state that 81.5% of systems include at least one outdated library, and Lim et al. [55] who say that the majority of websites include JavaScript libraries that are more than seven years old. The issue is especially critical in Korean websites, where 94% of websites use at least one older library version, contributing to the higher number of vulnerabilities observed in Korean websites, as shown in Figure 12.

7 Technology & Affordability

Technology Adoption. The adoption of new technologies across countries often reflects various factors such as economic development, technological infrastructure, and user behavior. A common assumption is that developed countries are more likely to adopt new technologies swiftly due to their resources and advanced infrastructures. This section examines this premise by analyzing two factors: 1) the usage of HTTP protocols, 2) the adoption of various Web APIs, with a particular emphasis on security-critical APIs.

Figure 14(a) highlights the usage of HTTP protocols—HTTP/1.0, HTTP/1.1, HTTP/2 (h2), and HTTP/3 (h3)—across countries. Newer versions, such as h2 and h3, offer improvements in both performance and security. Despite expectations that developed countries would lead in adopting these advanced protocols, the data reveals that websites in developing countries are not necessarily more reliant on older technologies than their developed counterparts. For example, Nigerian websites exhibit a high combined usage of h2 and h3, at approximately 65% (37.80% and 27.09%, respectively), which surpasses the combined usage in US websites at around 58% (31.87% for h2 and 26.17% for h3). Similarly, Pakistani websites show high adoption rates of 36.03% for h2 and 29.39% for h3. In contrast, some developed countries like Germany and the United Kingdom show lower combined usage rates of these newer protocols, challenging the assumption that they would naturally lead in adopting cutting-edge technologies. Additionally, the usage of outdated protocols like HTTP/1.0 is minimal across all countries, with slightly higher rates observed in some developed nations. For



(a) Adoption of HTTP protocols across countries

(b) Usage of web APIs across countries

Figure 14: Comparison of the adoption of HTTP protocols and web APIs across developed and developing countries.

instance, US websites record an HTTP/1.0 usage of 0.31%, while Nigerian websites report only 0.02%.

Figure 14(b) shows the adoption of various Web APIs. We first selected the 74 Web API standards, using the same technique as Peter et al. [68], and then focused on 10 APIs that are security- and privacy-critical. These APIs are essential for enabling secure communications and protecting user data. The results show that the adoption of these security-critical APIs is not confined to developed countries. Russian websites demonstrate significant use of the Web Cryptography API, with 8,219 instances, compared to 4,214 in US websites. Similarly, Bangladeshi websites record 1,307 uses of Encrypted Media Extensions, exceeding the 887 instances in US websites. Nigerian websites also exhibit strong engagement with modern web technologies, with 2,318 instances of the Web Cryptography API, reflecting the active adoption of these APIs in developing regions.

These findings suggest that developing countries are not only embracing new technologies but, in some cases, are doing so at rates comparable to or even exceeding those of developed nations. A potential explanation for this trend could be the lack of legacy systems in developing countries, which may facilitate the adoption of newer technologies without the burden of outdated infrastructure. Additionally, the widespread use of mobile internet and the presence of a younger, tech-savvy population may contribute to the rapid adoption of modern web technologies in these regions.

Website Affordability. To assess the affordability of websites, we use the PAW (Price Adjusted Web access) index, proposed by Qazi et al. [60]. This index measures the reduction in webpage sizes needed to meet the UN Broadband Commission's target, which states that broadband services should not exceed 2% of a country's Gross National Income (GNI) per capita [32]. A PAW index of 1 or less indicates affordable access, while values above 1 suggest that webpage sizes or broadband costs are too high relative to income, limiting access to essential online services [60]. See appendix A.7 for details on PAW calculation.

As shown in Figure 15, we calculated the PAW index for 20 countries, grouping them into developed and developing regions. The index values were based on average webpage sizes and mobile broadband prices in each country. The median PAW index for both developed and developing countries was below 1, indicating overall affordability. However, Japan (1.11) was the only developed country with a PAW index above 1. In developing countries, Pakistan (1.14), Nigeria (1.04), Bangladesh (1.19), and the Philippines (1.54) exceeded the threshold, with the Philippines having the highest index. This points to substantial barriers to web access, likely due to high mobile

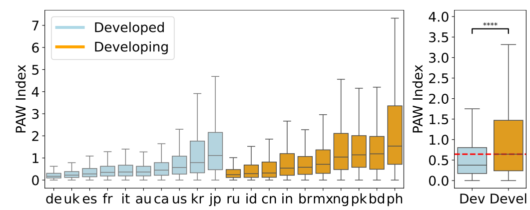


Figure 15: Website affordability (PAW Index) across countries.

data costs or large webpages. Additionally, developed countries showed less variation in PAW values, while developing countries had more inconsistency in affordability across regions, reflecting significant regional disparities.

8 Conclusion

Despite two-thirds of the global population is now online, a significant digital divide persists between developed regions (93% connected) and developing ones (60%). This gap, worsened by dependence on mobile internet and high data costs in developing countries, affects both access and user experience. Websites in these regions often suffer from slower load times, poorer optimization, and limited accessibility. This study aims to provide robust evidence of these disparities by examining the web ecosystem from the perspectives of developed and developing regions.

To address this, we leverage Google CrUX Report to identify the 10,000 most popular websites in the top 10 developing and developed countries (as per International Monetary Fund data). We then systematically crawled these 200,000 webpages using Google Lighthouse for performance audits and Puppeteer to capture detailed network request data. This effort resulted in what we believe to be the largest dataset of its kind, encompassing websites from both regions, which we plan to open-source.

Our analysis reveals that websites in developing countries are generally 10% smaller and use fewer third-party resources and resource-intensive JavaScript (JS), leading to simpler DOM structures and reduced browser blocking time. Yet, these sites also demonstrate weaker optimization practices, wasting 10% more bandwidth on improperly sized images and redundant JS and CSS. We suggest that their smaller size may be due to simpler page structures rather than deliberate optimization. A similar pattern emerges in security practices, with developing regions lagging in HTTPS (88% vs. 95%) and CSP (75% vs. 81.5%) adoption. It remains uncertain whether these differences reflect a lack of awareness or intentional design choices to support surveillance.

References

[1] Core Web Vitals report - Search Console Help.
[2] The cost of JavaScript in 2019 · V8.
[3] Desktop vs Mobile vs Tablet Market Share Asia.
[4] Facebook Lite APK for Android.
[5] JPEG vs. JPEG 2000: Which is better? | Adobe.
[6] Largest Contentful Paint (LCP) | Articles.
[7] Lighthouse.
[8] Load mobile pages faster with Web Light - Google Search Help.
[9] Overview of CrUX | Chrome UX Report.
[10] Puppeteer | Puppeteer.
[11] Removing cookie consent banners from your lighthouse test.
[12] retirejs: Port of RetireJS in Python.
[13] Serve images in modern formats | Lighthouse.
[14] Speedtest Global Index – Internet Speed around the world.
[15] The State of Mobile Internet Connectivity Report 2023 - Mobile for Development.
[16] WHOIS Search, Domain Name, Website, and IP Tools - Who.is.
[17] World Bank Open Data.
[18] World Economic Outlook Database - Groups and Aggregates.
[19] Data and privacy unprotected in one third of countries, despite progress | UNCTAD, Apr. 2020.
[20] Ahmad, T., Zaki, Y., Pötsh, T., Chen, J., Sathiaseelan, A., and Subramanian, L. Gaius: a new mobile content creation and diffusion ecosystem for emerging regions. In *Proceedings of the Tenth International Conference on Information and Communication Technologies and Development* (New York, NY, USA, 2019), ICTD '19, Association for Computing Machinery.
[21] Amjad, A. H., Shafiq, Z., and Gulzar, M. A. Blocking javascript without breaking the web. In *Privacy Enhancing Technologies Symposium (PETS)* (2023).
[22] Archive, H. State of the web. https://httparchive.org/reports/state-of-the-web, 2021.
[23] Bank, W. *World Development Report 2016: Digital Dividends.* World Bank Publications, 2016.
[24] Bischof, Z. S., Rula, J. P., and Bustamante, F. E. In and out of cuba: Characterizing cuba's connectivity. In *Proceedings of the 2015 Internet Measurement Conference* (New York, NY, USA, 2015), IMC '15, Association for Computing Machinery, p. 487–493.
[25] Brezis, E. S., Krugman, P. R., and Tsiddon, D. Leapfrogging: A theory of cycles in national technological leadership, 1991.
[26] Chaqfeh, M., Asim, R., AlShebli, B., Zaffar, M. F., Rahwan, T., and Zaki, Y. Towards a world wide web without digital inequality. *Proceedings of the National Academy of Sciences 120*, 3 (2023), e2212649120.
[27] Chaqfeh, M., Asim, R., AlShebli, B., Zaffar, M. F., Rahwan, T., and Zaki, Y. Towards a world wide web without digital inequality. *Proceedings of the National Academy of Sciences 120*, 3 (2023), e2212649120.
[28] Chaqfeh, M., Haseeb, M., Hashmi, W., Inshuti, P., Ramesh, M., Varvello, M., Subramanian, L., Zaffar, F., and Zaki, Y. To block or not to block: Accelerating mobile web pages on-the-fly through javascript classification. In *Proceedings of the 2022 International Conference on Information and Communication Technologies and Development* (New York, NY, USA, 2023), ICTD '22, Association for Computing Machinery.
[29] Chaqfeh, M., Zaki, Y., Hu, J., and Subramanian, L. Jscleaner: De-cluttering mobile webpages through javascript cleanup. In *Proceedings of The Web Conference 2020* (New York, NY, USA, 2020), WWW '20, Association for Computing Machinery, p. 763–773.
[30] Chaqfeh, M., Zaki, Y., Hu, J., and Subramanian, L. Jscleaner: De-cluttering mobile webpages through javascript cleanup. In *Proceedings of The Web Conference 2020* (2020), pp. 763–773.
[31] Chetty, M., Sundaresan, S., Muckaden, S., Feamster, N., and Calandro, E. Measuring broadband performance in south africa. In *Proceedings of the 4th Annual Symposium on Computing for Development* (New York, NY, USA, 2013), ACM DEV-4 '13, Association for Computing Machinery.
[32] Creative, C. UN Broadband Commission Adopts A4AI "1 for 2" Affordability Target, Jan. 2018.
[33] Daigle, L. WHOIS Protocol Specification. Request for Comments RFC 3912, Internet Engineering Task Force, Sept. 2004. Num Pages: 4.
[34] Fanou, R., Tyson, G., Francois, P., and Sathiaseelan, A. Pushing the frontier: Exploring the african web ecosystem. In *Proceedings of the 25th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2016), WWW '16, International World Wide Web Conferences Steering Committee, p. 435–445.
[35] Felt, A. P., Barnes, R., King, A., Palmer, C., Bentzel, C., and Tabriz, P. Measuring HTTPS adoption on the web. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017* (2017), E. Kirda and T. Ristenpart, Eds., USENIX Association, pp. 1323–1338.
[36] fido128. Answer to "CrUX: How much data is sufficient?", Jan. 2024.
[37] Fong, M. W. Technology leapfrogging for developing countries. In *Encyclopedia of Information Science and Technology, Second Edition*. IGI Global, 2009, pp. 3707–3713.
[38] Google. AMP is a web component framework to easily create user-first web experiences - amp.dev. https://amp.dev, 2019. Accessed: 2019-05-05.
[39] Gotze, M., Matic, S., Iordanou, C., Smaragdakis, G., and Laoutaris, N. Measuring web cookies in governmental websites. In *Proceedings of the 14th ACM Web Science Conference 2022* (2022), pp. 44–54.
[40] Group, W. B. *World development report 2016: Digital dividends.* World Bank Publications, 2016.
[41] GSMA. The state of mobile internet connectivity report 2020. Tech. rep., GSMA, 2020.
[42] Habib, R., Inam, A., Ali, A., Qazi, I. A., and Qazi, Z. A. A First Look at Public Service Websites from the Affordability Lens. In *Proceedings of the ACM Web Conference 2023* (Austin TX USA, Apr. 2023), ACM, pp. 2731–2741.
[43] Habib, R., Tanveer, S., Inam, A., Ahmed, H., Ali, A., Uzmi, Z. A., Qazi, Z. A., and Qazi, I. A. A framework for improving web affordability and inclusiveness. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA, 2023), ACM SIGCOMM '23, Association for Computing Machinery, p. 592–607.
[44] Hasan, S., Barela, M. C., Johnson, M., Brewer, E., and Heimerl, K. Scaling community cellular networks with community cellular manager. In *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation* (USA, 2019), NSDI'19, USENIX Association, p. 735–750.
[45] Hasan, S., Ben-David, Y., Scott, C., Brewer, E., and Shenker, S. Enhancing rural connectivity with software defined networks. In *Proceedings of the 3rd ACM Symposium on Computing for Development* (New York, NY, USA, 2013), ACM DEV '13, Association for Computing Machinery.
[46] Hasan, S., Heimerl, K., Harrison, K., Ali, K., Roberts, S., Sahai, A., and Brewer, E. Gsm whitespaces: An opportunity for rural cellular service. In *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)* (2014), pp. 271–282.
[47] Helsper, E. The digital disconnect: The social causes and consequences of digital inequalities.
[48] Hilbert, M. The bad news is that the digital access divide is here to stay: Domestically installed bandwidths among 172 countries for 1986–2014. *Telecommunications Policy 40*, 6 (2016), 567–581.
[49] Jueckstock, J., Sarker, S., Snyder, P., Beggs, A., Papadopoulos, P., Varvello, M., Livshits, B., and Kapravelos, A. Towards realistic and reproducible web crawl measurements. In *Proceedings of the Web Conference 2021* (2021), pp. 80–91.
[50] Koehler, W. Web page change and persistence - A four-year longitudinal study. *J. Assoc. Inf. Sci. Technol. 53*, 2 (2002), 162–171.
[51] Koradia, Z., Mannava, G., Raman, A., Aggarwal, G., Ribeiro, V., Seth, A., Ardon, S., Mahanti, A., and Triukose, S. First impressions on the state of cellular data connectivity in india. In *Proceedings of the 4th Annual Symposium on Computing for Development* (New York, NY, USA, 2013), ACM DEV-4 '13, Association for Computing Machinery.
[52] Kula, R. G., German, D. M., Ouni, A., Ishio, T., and Inoue, K. Do developers update their library dependencies? an empirical study on the impact of security advisories on library migration. *Empirical Software Engineering 23* (2018), 384–417.
[53] Kupoluyi, J., Chaqfeh, M., Varvello, M., Coke, R., Hashmi, W., Subramanian, L., and Zaki, Y. Muzeel: assessing the impact of javascript dead code elimination on mobile web performance. In *Proceedings of the 22nd ACM Internet Measurement Conference* (New York, NY, USA, 2022), IMC '22, Association for Computing Machinery, p. 335–348.
[54] Lauinger, T., Chaabane, A., Arshad, S., Robertson, W., Wilson, C., and Kirda, E. Thou shalt not depend on me: Analysing the use of outdated javascript libraries on the web. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017* (2017), The Internet Society.
[55] Lim, K., Kwon, Y., and Kim, D. A longitudinal study of vulnerable client-side resources and web developers' updating behaviors. In *Proceedings of the 2023 ACM on Internet Measurement Conference, IMC 2023, Montreal, QC, Canada, October 24-26, 2023* (2023), M. Montpetit, A. Leivadeas, S. Uhlig, and M. Javed, Eds., ACM, pp. 162–180.
[56] Lythreatis, S., Singh, S. K., and El-Kassar, A.-N. The digital divide: A review and future research agenda. *Technological Forecasting and Social Change 175* (2022), 121359.
[57] Naseer, U., and Benson, T. A. Js capsules: A framework for capturing fine-grained javascript memory measurements for the mobile web. *Proc. ACM Meas. Anal. Comput. Syst. 7*, 1 (Mar. 2023).
[58] Naseer, U., Benson, T. A., and Netravali, R. Webmedic: Disentangling the memory-functionality tension for the next billion mobile web users. In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications* (New York, NY, USA, 2021), HotMobile '21, Association for Computing Machinery, p. 71–77.
[59] OECD. Bridging the digital gender divide. Tech. rep., Organisation for Economic Co-operation and Development, 2018.

[60] Qazi, I. A., Qazi, Z. A., Ali, A., Abdullah, M., and Habib, R. Rethinking Web for Affordability and Inclusion. In *Proceedings of the Twentieth ACM Workshop on Hot Topics in Networks* (Virtual Event United Kingdom, Nov. 2021), ACM, pp. 9–15.
[61] Rack, J., and Staicu, C.-A. Jack-in-the-box: An empirical study of javascript bundling on the web and its security implications. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (2023), pp. 3198–3212.
[62] Rautenstrauch, J., Mitkov, M., Helbrecht, T., Hetterich, L., and Stock, B. To auth or not to auth? a comparative analysis of the pre-and post-login security landscape. In *2024 IEEE Symposium on Security and Privacy (SP)* (2024), IEEE, pp. 1500–1516.
[63] Raza, A., Zaki, Y., Pötsch, T., Chen, J., and Subramanian, L. xcache: Rethinking edge caching for developing regions. In *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development* (New York, NY, USA, 2017), ICTD '17, Association for Computing Machinery.
[64] Rescorla, E. Http over tls, 2000.
[65] Roth, S., Gröber, L., Backes, M., Krombholz, K., and Stock, B. 12 angry developers - A qualitative study on developers' struggles with CSP. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021* (2021), ACM, pp. 3085–3103.
[66] Ruth, K., Fass, A., Azose, J., Pearson, M., Thomas, E., Sadowski, C., and Durumeric, Z. A world wide view of browsing the world wide web. In *Proceedings of the 22nd ACM Internet Measurement Conference, IMC 2022, Nice, France, October 25-27, 2022* (2022), C. Barakat, C. Pelsser, T. A. Benson, and D. R. Choffnes, Eds., ACM, pp. 317–336.
[67] Sharma, A., Kaur, M., Koradia, Z., Nishant, R., Pandit, S., Raman, A., and Seth, A. Revisiting the state of cellular data connectivity in india. In *Proceedings of the 2015 Annual Symposium on Computing for Development* (New York, NY, USA, 2015), DEV '15, Association for Computing Machinery, p. 149–157.
[68] Snyder, P., Taylor, C., and Kanich, C. Most websites don't need to vibrate: A cost-benefit approach to improving browser security. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), pp. 179–194.
[69] Union, I. T. Measuring digital development: Ict price trends 2020. Tech. rep., ITU, 2020.
[70] Zaki, Y., Chen, J., Pötsch, T., Ahmad, T., and Subramanian, L. Dissecting web latency in ghana. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (New York, NY, USA, 2014), IMC '14, Association for Computing Machinery, p. 241–248.
[71] Zaki, Y., Chen, J., Pötsch, T., Ahmad, T., and Subramanian, L. Dissecting Web Latency in Ghana. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (Vancouver BC Canada, Nov. 2014), ACM, pp. 241–248.
[72] Zigisova, E. The 2022 Web Almanac: Third Parties. Tech. rep., HTTP Archive, Sept. 2022. Issue: 8 Publication Title: The 2022 Web Almanac Volume: 4.

A Appendix

A.1 Distribution of Website Rankings

Figure 16 displays the distribution of website rankings across different countries in our dataset. The figure illustrates that, for most countries, the 10,000 sampled websites fall within the top 50,000 of that country, indicating a focus on relatively popular websites. However, for countries such as Bangladesh, Pakistan, Nigeria, and the Philippines, less popular websites were included to meet the target sample size of 10,000 websites per country.

A.2 Country-wise Distribution of Image Sizes

Figure 17 illustrates the distribution of image sizes across different countries. The chart highlights that websites in developing countries tend to use larger image sizes compared to those in developed countries. Specifically, 15% of images on websites in developing countries exceed 100 KB, whereas only 12% of images in developed countries fall into this category. This data supports the observation that while websites in developing countries may not host more images overall, they often feature larger images, contributing to greater webpage sizes and potentially increased load times and bandwidth usage.

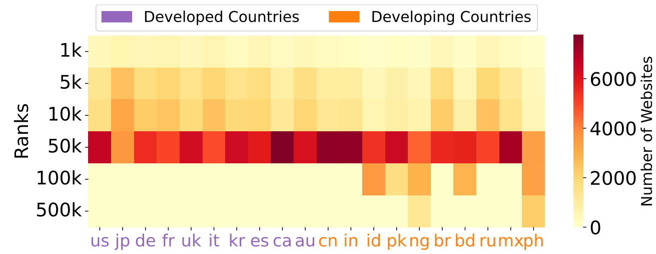


Figure 16: Distribution of website rankings in our dataset per country. The figure shows that in most cases, the 10,000 websites per country are within the top 50,000 of that country. For Bangladesh, Pakistan, Nigeria, and the Philippines, less popular websites were included to meet the target sample size.

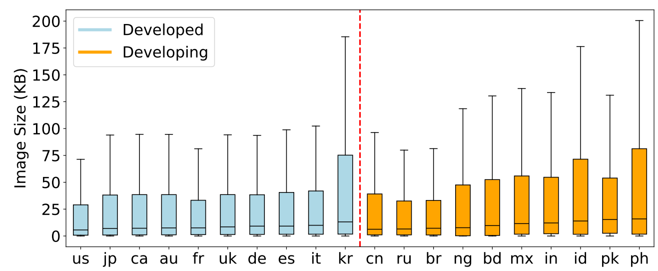


Figure 17: Distribution of image sizes across countries.

A.3 Distribution of Document Sizes

Figure 18 illustrates the distribution of document sizes across different countries. The data indicates that websites in developed countries tend to have larger document sizes compared to those in developing countries.

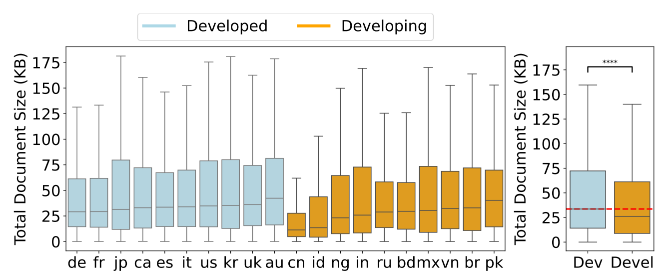


Figure 18: Distribution of document sizes across countries.

A.4 Country-wise Distribution of Total Blocking Time

Figure 19 illustrates the distribution of total blocking time across developed and developing countries. Total blocking time refers to the duration during which the browser's main thread is blocked, preventing user interactions such as clicks, scrolling, or typing. This metric is important for assessing website responsiveness, with longer blocking times leading to poorer user experiences. The data shows significant differences between regions, with developed countries generally exhibiting longer blocking times compared to developing countries. This disparity is largely attributed to the heavier use of JavaScript in websites from developed regions, which increases the complexity of web pages and leads to more frequent blocking of the main thread.

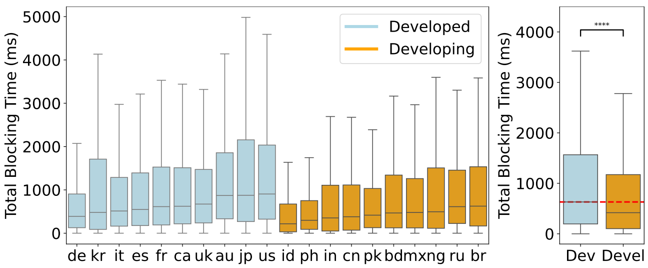


Figure 19: Distribution of total blocking time across countries.

A.5 HTTPS Adoption

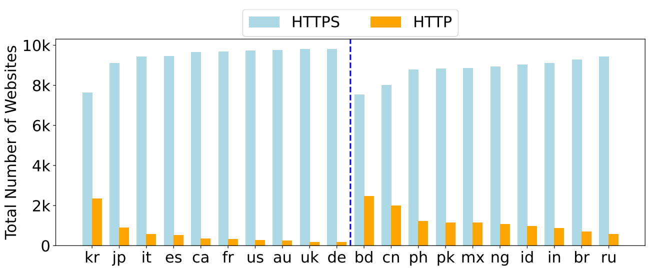


Figure 20: Country-wise HTTPS and HTTP adoption rates.

Figure 20 shows significant variation in HTTPS adoption across countries. Developed nations like the United States and Germany have higher adoption rates, while developing countries like Bangladesh and China exhibit a larger percentage of websites still relying on HTTP. This suggests that security practices vary significantly between regions, potentially exposing users in some areas to greater security risks.

A.6 CSS Wastage Distribution

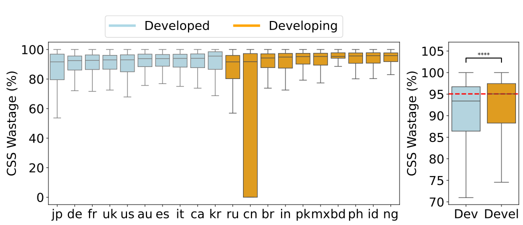


Figure 21: Percentage of unused CSS rules across countries

Figure 21 presents the percentage of unused CSS rules across various countries. Developing countries such as Pakistan and Bangladesh exhibit higher wastage levels, exceeding 95%, while developed countries like Japan and Germany show slightly lower wastage rates, around 92%.

A.7 PAW Analysis

The PAW (Price Adjusted Web access) Index is a measure that evaluates the affordability of accessing web content in a given region by comparing the average broadband prices, local webpage sizes, and income levels to international affordability targets. The formula for the PAW Index is:

$$PAW_i = \frac{P_i}{P_T} \times \frac{W_{j,i}}{W_{global}}$$

where P_i represents the average broadband price in country i, P_T is the target broadband price set at 2% of the country's GNI per capita, $W_{j,i}$ is the size of webpage j in country i, and W_{global} is the global average webpage size. A PAW Index greater than 1 indicates that accessing web content is unaffordable in comparison to the international benchmark, while a value less than or equal to 1 suggests that the region meets the affordability target. This index provides a tool for policymakers and researchers to assess and compare the digital affordability landscape across regions, helping to identify gaps and make informed decisions to promote accessible digital content.