

PILLAR

LEARN · WEB FOUNDATIONS

Mobile-First Design for the Global South

In LATAM, Southeast Asia, and much of Africa, the smartphone is not a secondary device. It is the only device. This piece teaches the design constraints that follow from that reality, the performance budgets that protect against them, and the widening gap in AI tooling that builders need to know about.

9 min read

Last updated: June 10, 2026

PILLAR MEDIA & ENTERTAINMENT · PILLARME.COM/LEARN

Mobile-first is not an aesthetic preference in the Global South. It is a structural constraint imposed by device economics, data costs, and network reality, and most modern AI tools are quietly making the gap worse.

The thesis

- Why mobile-only access defines the majority of users in LATAM, Indonesia, and the Philippines, and what that means for layout decisions
- The performance budget (FCP < 1.8s, LCP < 2.5s, CLS < 0.1) that separates usable sites from abandoned ones on a \$200 Android
- How to run a 360px mobile-first audit in Chrome DevTools, including throttling for Slow 3G conditions
- Why most current AI tools (Runway, NotebookLM, Sora) degrade or fail on mobile, and how that shapes builder workflows in the Global South
- How AI Overviews and Discovery surfaces increasingly favor mobile-optimized sites with strong Core Web Vitals

01 — The framework: The Mobile-Only Stack

The Mobile-Only Stack

The Mobile-Only Stack is a four-layer mental model for builders shipping to readers who have one device, one screen, and one expensive data plan. Each layer constrains the one above it, and ignoring any single layer breaks the experience for the majority of your audience.

1

Device layer: the \$200 Android

Design and test against a mid-range Android with 3-4 GB of RAM, a 360px viewport, and a battery that is already at 40 percent. The average smartphone in LATAM costs between \$150 and \$250, compared with \$800+ in the US. Anything you ship needs to run on that device, not on the M-series laptop you built it with.

2

Network layer: Slow 3G as default

In Argentina, mobile data costs roughly 8x more per GB than in the US. In rural Indonesia and across much of sub-Saharan Africa, users routinely operate on 3G or congested 4G. Treat Slow 3G in DevTools throttling as your baseline test condition, not as an edge case, and budget every kilobyte as if the user is paying for it directly.

3

Interface layer: thumb-zone first

On a one-handed phone, the comfortable thumb-reach zone is the bottom two-thirds of the screen. Primary actions, navigation, and CTAs belong there. Touch targets need to be at least 44px square, forms need to lose half their fields, and any layout that depends on hover states is broken by definition.

4

Tooling layer: the AI mobile gap

Most current AI products are built mobile-second or desktop-only. Runway, NotebookLM, Sora, and many code-generation tools degrade significantly or simply fail on a mid-range Android. Builders in the Global South need to know which tools work on their devices, which work only on their clients' devices, and where the gap creates strategic openings.

02 — The data.

73%

Share of LATAM e-commerce transactions that happen on mobile

STATISTA, 2024

96%+

Share of Indonesian internet users who access the web via mobile

DATAREPORTAL, DIGITAL 2024 INDONESIA

76%

Share of Philippine internet users who are mobile-only

DATAREPORTAL, DIGITAL 2024 PHILIPPINES

\$150-\$250

Typical smartphone price in LATAM (vs. \$800+ in the US)

COUNTERPOINT RESEARCH, REGIONAL SMARTPHONE PRICING

8x

How much more expensive mobile data is per GB in Argentina vs. the US

CABLE.CO.UK WORLDWIDE MOBILE DATA PRICING INDEX

< 2.5s

Largest Contentful Paint target for a Core Web Vitals pass

GOOGLE WEB.DEV CORE WEB VITALS GUIDANCE

Why mobile-only is the default, not the exception

When North American and European product teams say "mobile-first," they usually mean a design that adapts gracefully from a 1440px monitor down to a phone. In LATAM, Africa, and Southeast Asia, the assumption inverts. The phone is the only screen the user owns. DataReportal's 2024 figures put Indonesian mobile internet usage above 96 percent, the Philippines at 76 percent mobile-only, and across LATAM, Statista reports that 73 percent of e-commerce transactions now happen on a phone. These are not preferences. They are the consequence of household economics: a single \$200 Android serves an entire family for browsing, banking, work, and communication.

The design implication is severe. A homepage that loads in 1.4 seconds on your MacBook may take 9 seconds on a mid-range Samsung A-series in Lagos or Lima. A hero image that looks crisp at 1920px may eat half a user's daily data budget. A form with eight fields that converts at 12 percent on desktop may convert at under 2 percent on mobile, not because the offer is wrong but because the keyboard occludes the submit button and the layout shifts every time the user taps.

The performance budget that actually matters

Core Web Vitals are not a Google compliance exercise. They are the closest thing the industry has to a shared standard for whether a site is usable on a constrained device.

The three numbers that matter most are First Contentful Paint under 1.8 seconds, Largest Contentful Paint under 2.5 seconds, and Cumulative Layout Shift under 0.1. Hit those at 360px on Slow 3G and your site is genuinely usable. Miss them and you are losing the majority of your audience before they see your headline.

The performance budget cascades into specific design choices. Hero videos are usually wrong. Web fonts loaded synchronously are usually wrong. Images larger than 100KB are almost always wrong. WebP and AVIF cut image weight by 25-50 percent compared to JPEG at equivalent quality. Lazy-loading below-the-fold media is non-negotiable. Server-side rendering or static generation beats client-side hydration for first paint. None of this is exotic. It is just the discipline of treating bandwidth as the user's money, because in Argentina, Nigeria, or the Philippines, it literally is.

The AI tooling gap and what it means for builders

A quieter shift is happening underneath the design conversation. The current generation of AI products is overwhelmingly built for desktop-class devices. Runway and Sora assume a powerful GPU on the other end of a fast connection. NotebookLM works on mobile but with degraded interaction patterns. Many code-generation IDEs are unusable on a phone. This is not a small problem for builders working in or for the Global South: it means the AI productivity multiplier is unevenly distributed by geography in a way that mirrors and may widen existing economic gaps.

There are two strategic implications. First, if you are building for a mobile-only audience, you cannot assume your users have access to the same AI-augmented experiences your San Francisco competitors are shipping. That is sometimes a constraint and sometimes an opening. Second, AI Overviews and Discovery surfaces, the new interfaces through which users find content, increasingly weight mobile-optimized signals heavily. A site that nails Core Web Vitals, ships clean mobile schema, and answers questions directly is more likely to be cited by ChatGPT, surfaced in Perplexity, and pulled into Google's AI Overviews. Mobile-first design is no longer just a UX choice. It is an AEO choice. [Pillar Authority](#) ([/authority/](#)), exists to help operators make those choices deliberately.

Designing for the thumb, the keyboard, and the data plan

Once you accept the device and network layer, the interface layer follows mechanically. Primary navigation lives at the bottom of the screen, where the thumb can reach it. Touch targets are at least 44 pixels on a side, with 8 pixels of padding between adjacent tap zones. Forms lose half their fields, because every keystroke on a glass keyboard is a cost the user is paying. Modal dialogs that cannot be dismissed by tapping outside them are user-hostile. Carousels that auto-advance are user-hostile. Hover-only menus do not exist on touch devices, full stop.

The most undervalued discipline is offline tolerance. A user on a Buenos Aires bus loses signal three times in ten minutes. A user in Jakarta crosses between LTE and 3G constantly. Your site needs to fail gracefully: cached pages should load, form drafts should persist locally, and any state that depends on a live connection should make that dependency visible. Service workers, IndexedDB, and progressive enhancement are not advanced techniques here. They are baseline professionalism.

03 — Watch: a real walkthrough

04 — Apply this to your work

Run this audit on your homepage and your highest-traffic landing page this week. Each item maps to a specific Core Web Vitals or interaction target and can be done in under an hour.

1. Open your homepage in Chrome DevTools, set the device to a 360px viewport (e.g., Galaxy S8), and toggle CPU throttling to 4x slowdown. Record the actual First Contentful Paint, Largest Contentful Paint, and Cumulative Layout Shift values.
2. Apply network throttling: Slow 3G. Reload. If your LCP is above 2.5 seconds or your FCP is above 1.8 seconds, you have a budget problem, not a design problem.
3. Audit every image on the page. Anything over 100KB needs to be converted to WebP or AVIF, served at the rendered size (not the source size), and lazy-loaded if it is below the fold.
4. Inspect your forms. Count the fields. Cut the count in half. Ensure every input has a touch target of at least 44px and that the submit button is reachable with one thumb without the keyboard occluding it.
5. Disable JavaScript in DevTools and reload. Critical content should still render. If your page is blank, you are excluding users on flaky connections by default.
6. Check your fonts. Self-host or use font-display: swap. A 200KB web font that blocks rendering for 4 seconds on 3G is worse than a system font that loads instantly.
7. Run a Lighthouse audit on mobile preset. Take the Performance, Accessibility, and SEO scores seriously. A score under 70 on mobile means you have material work to do before AI Overviews will reliably cite you.

Frequently asked questions.

Is "mobile-first" still the right term, or should we just say "mobile-only"?

Both terms are useful, but they describe different audiences. Mobile-first is a design methodology: start with the smallest viewport and progressively enhance. Mobile-only is an audience reality: the user has no other device. In LATAM, Indonesia, and the Philippines, the audience is genuinely mobile-only for the majority of users, which is why this piece is titled the way it is. Using the methodology is the only honest response to the audience condition.

How much should I really care about Core Web Vitals if my audience is in the Global South and not searching on Google?

More than you think, for two reasons. First, the same constraints that Core Web Vitals measure (paint speed, layout stability) are exactly what determine whether a user on a 3G connection abandons your site. Second, Discovery surfaces beyond Google including AI Overviews, Perplexity citations, and ChatGPT browsing all increasingly weight mobile signals. The technical floor for being cited by AI is rising, and mobile performance is part of that floor. [Pillar Authority \(/authority/\)](#) tracks how those signals translate into citations.

What about AI tools? Should I be building with them if my audience cannot use them?

Yes, but with discipline. You can use desktop-class AI tools to build, design, and operate your site. What matters is that the output runs on a \$200 Android. The mistake is shipping an AI-augmented frontend (live transcription, real-time generation, heavy client-side ML) and assuming your users have the device to run it. Use AI for production. Ship the lightest possible artifact.

Are there any AI tools that work well on mid-range Android?

ChatGPT and Claude both have functional mobile apps and web experiences that work on mid-range Android, though they can be data-heavy. Perplexity is generally lighter. NotebookLM is usable but degraded. Runway, Sora, and most video-generation tools are effectively desktop-only at present. The gap is real and shapes which workflows are accessible to builders in the Global South.

I am a designer working with a US client whose audience is diaspora. Do these rules apply?

Yes, often more than the client realizes. Diaspora audiences frequently use the device and data plan profile of the country they emigrated from, especially for content in the heritage language. If your client's audience includes Latin American diaspora in the US, Filipino diaspora across the Gulf, or African diaspora in Europe, the Mobile-Only Stack is the right frame. Audit at 360px on Slow 3G and you will catch problems your client's analytics may be obscuring.
