

mobile-uat — Responsive Regression Detector

Mobile UAT — Responsive UX Regression Detector

Created: 2026-05-15 (John, after CEO caught snowit.ba landing 248px mobile overflow that source-only verification missed)

Skill path: `~/ .claude/skills/mobile-uat/SKILL.md`

Trigger: `/mobile-uat <url>`, "mob test", "responsive test", "mobile ne valja", "kreiraj mob test"

Author: Vizu/Brad Frost methodology, implemented via Playwright MCP

What it does

Drives a real Chromium browser via Playwright MCP at multiple mobile viewports (iPhone 13 390×844, iPad 768×1024, Android small 360×640) and runs deterministic hard-fail + soft-warn checks.

Hard-fail conditions (verdict = FAIL)

Code	Check	Why it matters
H1	<code>documentElement.scrollWidth > clientWidth + 2</code>	Horizontal page scroll = broken mobile layout
H2	<code><details>:not([open])</code> count > 0 (opt-out for FAQs)	Content hidden behind collapsed elements = user thinks page is empty
H3	Text present on desktop but absent on mobile	Content disappeared between viewports
H4	<code><a></code> , <code><button></code> , <code><summary></code> with bounding rect < 44×44px	WCAG 2.5.5 tap target minimum, iOS HIG
H5	<code><h1>/<h2>/<h3></code> with empty next-sibling chain	Empty section = layout bug

Code	Check	Why it matters
H6	<code><p></code> , <code></code> , <code><td></code> computed font-size < 14px	Microscopic text on mobile = unreadable

Soft-warn conditions (verdict = PARTIAL)

Code	Check
S1	Console errors > 0
S2	Network 4xx/5xx (excluding favicon, analytics)
S3	Cumulative layout shift > 0.1
S4	<code></code> without <code>alt</code> attribute

When to use

- **Mandatory:** after any HTML/CSS deploy to a public web app (companion to `/deploy-verify`)
- **Reactive:** whenever CEO says "doesn't look good on mobile" / "stvari nestale" / "ne valja na telefonu"
- **Audit:** existing site responsive sanity check

When NOT to use

- API-only services (no DOM)
- Native apps (use Paul Hudson / Skybound)
- Sites requiring login (skill is unauth-only — extend for auth scenarios later)

Example invocation

```
/mobile-uat https://snowit.ba/
```

Output: `/tmp/mobile-uat-<run_id>/`

- `SUMMARY.md` — human-readable table per URL × viewport

- `verification.json` — machine-readable verdict
- `screenshots/` — visual evidence per viewport
- `console/` — JS errors
- `network/` — HTTP requests

Real first run (2026-05-15)

Source-only initial run on snowit.ba legal pages reported PASS (0 hard fails). But real-browser run on the landing index.html caught:

Metric	Value
scrollWidth	638px (vs 390px viewport)
Horizontal overflow	248px
Off-screen elements	5 (hero-content, hero-badge, h1, highlight, hero-subtitle)
Small tap targets	13

Root cause: per-page inline `<style>` with `@media (max-width: 1024px)` that set `.hero-content max-width:600px` without `width:100%` — parent grid cell was wider than viewport.

After Vizu fix (commit 37389ef): scrollWidth=390, 0 offscreen, hero readable. Same fix swept across 8 SnowIT pages (index + 7 verticals).

Lesson: source-only static checks are not enough for responsive bugs. Real Chromium with computed styles + bounding rects is mandatory.

Related skills

- `/deploy-verify` — post-deploy gate (general, not responsive-specific)
- `/uat-browser` — generic in-session UAT via Playwright MCP (broader scope)
- `/webapp-testing` — Playwright local-app testing

Cost

Approx \$0.30-\$0.80 per run (Sonnet, 4 viewports × ~3 URLs). Acceptable for any site deploy.

Source

- `~/ .claude/skills/mobile-uat/SKILL.md`
- `~/ .claude/skills/mobile-uat/example-run.md`

Revision #2

Created 2026-05-15 20:10:59 UTC by John

Updated 2026-06-21 20:03:33 UTC by John