

The Page Weight Audit Checklist

One video re-encode, six image conversions — 27.7MB cut to 1.2MB, no redesign.

■ FIND THE HEAVY FILES

- ☐ DevTools → Network → hard-reload → sort by size, descending — page weight is almost always 2-3 files
- ☐ Confirm it on disk: `du -sh public/* | sort -rh | head`
- ☐ Check the video you suspect: `ffprobe -v error -select_streams v:0 -show_entries stream=width,height,bit_rate -of default=nw=1 bg2.mp4`
- ☐ Measure on a throttled connection, not office wifi — that's how a 27.7MB page survives for months

■ RE-ENCODE THE HERO VIDEO

- ☐ Scale to the size it actually renders at, CRF 25-28, muted, streamable: `ffmpeg -i in.mp4 -vf scale=1920:-2 -c:v libx264 -crf 27 -preset slow -an -movflags +faststart out.mp4`
- ☐ Generate a poster frame so LCP doesn't wait on the video download: `ffmpeg -i out.mp4 -frames:v 1 -q:v 3 poster.jpg`
- ☐ Confirm the poster paints immediately and the video takes over once it's ready

■ CONVERT AND RESIZE IMAGES

- ☐ Convert screenshots, photos, and UI captures to WebP, resized to the render width: `cwebp -q 80 -resize 1600 0 screenshot.png -o screenshot.webp`
- ☐ Keep PNG only for og:images, logos, and hard-transparency cases
- ☐ Reach for AVIF only on fully-controlled hero/content images — 20-30% smaller than WebP, slower to encode, spottier crawler support

■ SHIP-LIST SANITY CHECKS

- ☐ Set Cache-Control headers so repeat visits skip re-downloading unchanged assets
- ☐ Rename files with spaces — they %20-encode and pollute every log line and cache key
- ☐ Cross-check each image against its alt text for swapped or mismatched files
- ☐ Set a budget and hold to it: under 2MB per first visit, ~1.2MB of that for media — re-run the audit quarterly