MIDNIGHT WEBS

RESURSĂ GRATUITĂ · LISTĂ DE PERFORMANȚĂ

Listă de verificare pentru viteza site-ului și Core Web Vitals

Google evaluează experiența paginii prin trei cifre: cât de repede apare conținutul principal, cât de repede reacționează pagina și cât de mult „sare". Lista te ajută să citești aceste cifre, să găsești ce îți încetinește paginile și să nu le lași să se degradeze.

Actualizat în noiembrie 2026 · 12 min de citit · 39 verificări

PENTRU CINE ESTE	CE ACOPERĂ	ȚINTE
Oricine răspunde de un site care pare lent sau pică la PageSpeed Insights	6 domenii, de la măsurare la monitorizare	LCP ≤ 2,5 s · INP ≤ 200 ms · CLS ≤ 0,1, la percentila 75 a vizitelor de pe mobil

ÎNCEPE DE AICI

Repară mai întâi cea mai mare imagine de pe pagina cea mai vizitată. Pe majoritatea site-urilor mici, o singură imagine hero prea mare este cea mai importantă cauză a unui Largest Contentful Paint lent.

BIFEAZĂ CĂSUȚA CÂND AI TERMINAT · ESENȚIAL = IMPACT MAXIM

01 Măsoară paginile care contează

Testează șabloanele care aduc vizite și venituri. Un scor perfect pe o pagină pe care n-o deschide nimeni nu schimbă nimic.

- [ ] ESENȚIAL **Învață cele trei ținte**
  Largest Contentful Paint (LCP) ≤ 2,5 s, Interaction to Next Paint (INP) ≤ 200 ms, Cumulative Layout Shift (CLS) ≤ 0,1. Google le evaluează la percentila 75: trei sferturi dintre vizitele reale trebuie să respecte fiecare țintă. INP a înlocuit First Input Delay în martie 2024.

- [ ] ESENȚIAL **Începe cu datele de teren**
  În PageSpeed Insights, citește mai întâi blocul cu experiența utilizatorilor reali. Provine din Chrome UX Report, acoperă ultimele 28 de zile și este ceea ce folosește Google în realitate.

- [ ] **Folosește datele de laborator ca să găsești cauzele**
  Lighthouse (în Chrome DevTools sau PageSpeed Insights) indică fișiere și elemente concrete. Scorurile de laborator variază între rulări, deci testează de trei ori și ia mediana.

- [ ] **Testează fiecare șablon, nu doar pagina principală**
  Pagina principală, categoria, produsul sau serviciul, articolul și checkout-ul. Paginile din același șablon împart de obicei aceleași probleme.

- [ ] **Notează un punct de plecare**
  Data, URL-ul, dispozitivul și cele trei cifre, într-un tabel. Fără un „înainte", un „după" este doar o părere.

- [ ] **Separă mobilul de desktop**
  Mobilul este testul mai strict, iar Google indexează mai întâi versiunea mobilă.

CUM VERIFICI

Verifică raportul Core Web Vitals din Search Console. Grupează adresele în Bun, Necesită îmbunătățiri și Slab, ca să vezi ce șablon trebuie reparat.

02 Repară imaginile și Largest Contentful Paint

Pe majoritatea site-urilor, elementul LCP este o imagine, iar imaginile sunt cele mai grele fișiere din pagină.

- [ ] ESENȚIAL **Găsește elementul tău LCP**
  În Chrome DevTools, panoul Performance, sau în diagnosticul PageSpeed Insights. Repară mai întâi acel element; accelerarea altor imagini nu ajută.

- [ ] ESENȚIAL **Nu aplica lazy-loading imaginii LCP**
  Folosește `fetchpriority="high"` pe ea și elimină `loading="lazy"`. Lazy-loading pentru tot ce e sub zona vizibilă este corect; aplicat imaginii hero, o întârzie.

- [ ] **Folosește formate moderne**
  AVIF sau WebP sunt de obicei mult mai mici decât JPEG la calitate similară. Păstrează un JPEG de rezervă doar dacă trebuie să susții browsere foarte vechi.

- [ ] **Servește dimensiunea potrivită pentru fiecare ecran**
  Folosește `srcset` și sizes, ca un telefon să descarce o imagine de 600 px lățime, nu originalul de 3000 px.

- [ ] **Setează întotdeauna width și height**
  Ca browserul să rezerve spațiu înainte de încărcarea imaginii, ceea ce previne salturile de pagină.

- [ ] **Țintește o imagine hero ușoară**
  Ca țintă de lucru, ține imaginea LCP sub circa 200 KB pe mobil; comprimă până când vezi diferența, apoi fă un pas înapoi.

- [ ] **Folosește un CDN sau un serviciu de imagini**
  Pentru redimensionare, conversie și cache automate, ca editorii să nu poată încărca din greșeală o fotografie de 6 MB.

CUM VERIFICI

Rulează din nou PageSpeed Insights după modificare și confirmă că elementul LCP și timpul lui s-au îmbunătățit. O cascadă în WebPageTest arată când începe de fapt descărcarea imaginii.

03 Redu greutatea JavaScript și a scripturilor terților

JavaScript este principalul motiv pentru care paginile par neresponsive, iar lipsa de răspuns este exact ce măsoară INP.

- [ ] ESENȚIAL **Fă lista tuturor scripturilor terților și a responsabililor lor**
  Analytics, chat, reclame, pixeli, instrumente A/B, bannere de consimțământ, hărți de căldură. Elimină tot ce nu a mai verificat nimeni în ultimul trimestru.

- [ ] ESENȚIAL **Încarcă târziu scripturile neesențiale**
  Folosește `defer` sau `async` și încarcă widgeturile de chat sau videoclipurile doar la interacțiunea vizitatorului. O imagine statică de previzualizare („facade") în locul unui iframe YouTube economisește sute de kilobaiți.

- [ ] **Sparge sarcinile lungi**
  Orice sarcină de peste 50 ms blochează reacția paginii. Găsește-le în panoul Performance și împarte lucrul sau mută-l în afara firului principal.

- [ ] **Elimină codul nefolosit**
  Verifică acoperirea în DevTools. Multe site-uri livrează o bibliotecă întreagă pentru o singură funcție mică.

- [ ] **Auditează managerul de taguri**
  Un singur container poate încărca zeci de taguri pe fiecare pagină. Restricționează tagurile la paginile care le cer.

- [ ] **Testează cu limitarea procesorului**
  Folosește încetinirea 4x din DevTools ca să simulezi un telefon de gamă medie; laptopul tău ascunde problema.

CUM VERIFICI

Apasă pe un meniu, un filtru sau „adaugă în coș" imediat după încărcarea paginii. Dacă există o întârziere vizibilă, INP probabil are probleme; confirmă în datele de teren.

04 Fonturi, CSS și stabilitatea aspectului

Textul trebuie să apară imediat și să rămână pe loc. Fonturile, CSS-ul și bannerele întârziate sunt cauzele obișnuite ale conținutului care sare.

- [ ] ESENȚIAL **Rezervă spațiu pentru tot ce se încarcă târziu**
  Reclame, conținut încorporat, bannere de cookie-uri, bare de newsletter și imagini. Folosește dimensiuni fixe sau proprietatea CSS `aspect-ratio`, ca nimic să nu împingă conținutul în jos.
- [ ] **Găzduiește fonturile local, în WOFF2**
  Elimină o conexiune în plus spre alt domeniu și îți dă control asupra cache-ului.
- [ ] **Limitează fișierele și grosimile de font**
  Două familii și trei grosimi sunt suficiente. Fă subset doar pentru limbile pe care le folosești.
- [ ] **Folosește `font-display: swap` și preîncarcă fișierul-cheie**
  Preîncarcă doar unul-două fonturi folosite deasupra liniei de plutire, cu atributul `crossorigin`.
- [ ] **Redu saltul la schimbarea fontului**
  Potrivește metricile fontului de rezervă cu `size-adjust` și descriptorii înrudiți, ca schimbarea să mute textul abia perceptibil.
- [ ] **Ține CSS-ul critic mic**
  Include inline ce trebuie pentru primul ecran și încarcă restul fără a bloca randarea. Elimină CSS-ul framework-urilor pe care nu le mai folosești.
- [ ] **Preferă SVG inline în locul fonturilor de iconițe**
  Câteva SVG-uri sunt mai ușoare și se randează fără să aștepte un font.

CUM VERIFICI

Încarcă pagina pe o conexiune limitată și urmărește orice mișcare. În Chrome DevTools, opțiunea Layout Shift Regions evidențiază exact ce s-a mutat.

05 Găzduire, cache și răspunsul serverului

Un front-end rapid nu poate salva un server lent. Time to First Byte (TTFB) întârzie tot ce urmează.

- [ ] ESENȚIAL **Măsoară TTFB din locurile unde sunt clienții tăi**
  Un TTFB bun este sub circa 0,8 s, iar unul slab peste 1,8 s. Un server pe alt continent adaugă întârziere înainte să ruleze orice cod.
- [ ] ESENȚIAL **Pune în cache fișierele statice pe termen lung**
  CSS, JS, fonturi și imagini cu nume amprentat pot folosi `Cache-Control: public, max-age=31536000, immutable`. Vizitele repetate descarcă aproape nimic.

- [ ] **Activează compresia Brotli sau Gzip**
  De obicei reduce fișierele text (HTML, CSS, JS, SVG) cu mai mult de jumătate.
- [ ] **Pune un CDN în fața site-ului**
  Servește fișierele dintr-o locație apropiată de vizitator și absoarbe vârfurile de trafic.
- [ ] **Pune HTML în cache cu grijă**
  Sigur pentru paginile publice, dacă îl poți goli la fiecare actualizare; niciodată pentru coș, conturi sau checkout.
- [ ] **Elimină lanțurile de redirecturi**
  Fiecare redirect adaugă un drum dus-întors complet. Leagă direct către adresa finală, inclusiv forma cu https:// și slash final.
- [ ] **Verifică interogările lente și pluginurile grele**
  Dacă TTFB este mare doar pe paginile dinamice, baza de date sau un plugin sunt mai probabile decât rețeaua.

CUM VERIFICI

În fila Network din DevTools, apasă pe documentul principal și verifică „Waiting for server response". Repetă de pe un telefon cu date mobile.

06 Testează pe dispozitive reale și monitorizează după lansare

Performanța derivă. Imaginile, scripturile și funcțiile noi anulează munca bună dacă nimeni nu răspunde de cifre.

- [ ] ESENȚIAL **Testează pe un telefon de gamă medie și pe o conexiune lentă**
  Chrome DevTools oferă Slow 4G și limitarea CPU, dar un telefon Android real de gamă medie este mai onest.
- [ ] ESENȚIAL **Stabilește un buget de performanță**
  De exemplu, imaginea LCP sub 200 KB, JavaScript sub 300 KB comprimat, un număr limitat de taguri ale terților. Stabilește-l cu cei care adaugă funcții.
- [ ] **Măsoară continuu utilizatorii reali**
  Biblioteca JavaScript web-vitals trimite LCP, INP și CLS din vizitele reale în analytics, ca să nu aștepți raportul pe 28 de zile.
- [ ] **Adaugă verificări în procesul de lansare**
  Lighthouse CI sau un test PageSpeed programat te poate avertiza când o modificare încetinește o pagină-cheie.
- [ ] **Revizuiește lunar cele mai lente pagini**
  Repară cea mai mare cauză, notează rezultatul, repetă.
- [ ] **Retestează după schimbări de șabloane, teme sau pluginuri**
  O singură actualizare de plugin poate adăuga 200 KB de script.

CUM VERIFICI

Compară raportul Core Web Vitals din Search Console de luna aceasta cu cel de luna trecută. Numărul adreselor Slabe trebuie să scadă, nu să crească.

Surse și lecturi suplimentare

web.dev: Web Vitals (praguri)
https://web.dev/articles/vitals

web.dev: Interaction to Next Paint
https://web.dev/articles/inp

web.dev: Optimizarea Largest Contentful Paint
https://web.dev/articles/optimize-lcp

web.dev: Time to First Byte
https://web.dev/articles/ttfb

Ai nevoie de ajutor?

Midnight Webs proiectează, securizează și întreține site-uri pentru afaceri mici. Spune-ne ce ai nevoie și stabilim împreună pașii.

midnightwebs.com/ro/estimate

Citește online: midnightwebs.com/ro/resources/website-speed-core-web-vitals-checklist