

Miten siirsin 10 vuotta vanhan WordPress-sivuston Next.js-aikaan yhdessä työpäivässä tekoälyn avulla

Säilyttämällä Google-näkyvyyden entisellään

Kirjoittanut: Karo Tammela, karotammela.fi



Projektipäiväkirja ja tekninen opas liiketoimintakriittisen verkkosivuston modernisointiin.

TL;DR: Projektin ydin pähkinänkuoressa

Lähtötilanne: hiihtogreeni.fi -mökkisivusto toimi raskaalla WordPress/Elementor-yhdistelmällä. SEO-näkyvyys ja sisältö olivat kohtuullisia, mutta ylläpito kankeaa ja se ei ollut omassa hallinnassani. Päivityksien tekeminen saattoi kestää ja palvelupyynnöissä piti aina jonkin aikaa odotella ennen kuin ne tuli hoidetuksi.

Tavoite: Siirtää sivusto moderniin teknologiaan (Next.js 16) täysin SEO-turvallisesti suojellen olemassa olevaa hakukoneliikennettä ja URL-rakenteita.

Toteutus: Tekoälyavusteinen koodaus agentteja ja eri kielimalleja (Kilo Code) hyödyntäen. Rakenteellinen purku, staattinen generointi (SSG) ja millintarkat uudelleenohjaukset.

Lopputulos: Salamannopea, turvallinen ja ylläpitovapaa sivusto. Kaikki vanhat URL-osoitteet ohjaavat oikein, ja roskapostilta suojattu yhteydenottolomake toimii luotettavasti ja roskapostin määrä on tippunut uudistuksen jälkeen nolleen.

Johdanto: Kun "toimiva" ei ole enää riittävän hyvä

Monet yritysten verkkosivut saavuttavat pisteen, jossa ne kyllä tekevät tehtävänsä, mutta niiden ylläpito on muuttunut tarpeettoman raskaaksi. Sisältö on arvokasta, sivusto on ansainnut paikkansa Googlessa ja asiakkaat löytävät sen. Konepellin alla teknologia (kuten vanhentuvat lisäosat) alkaa kuitenkin hidastaa toimintaa.

Tämä oli tilanne Levin Hiihtogreeni.fi -majoitussivuston kohdalla . Sivusto pyöri WordPressillä ja Elementorilla sisältäen suomen- ja englanninkielistä sisältöä, vuosien hakukonehistorian ja paljon brändille tärkeitä valokuvia . Tavoitteena ei ollut vain "tehdä jotain uutta", vaan rakentaa sivusto uudelleen modernilla teknologialla suojellen samalla sitä liiketoiminta-arvoa, joka vanhaan sivustoon oli sitoutunut . Olen koonnut tähän artikkeliin, kuinka tekoälyavusteinen koodaus helpottaa valtavasti sivoustouudistuksen toteuttamisessa säilyttäen edelleen vanhan sivuston SEO-ranking.

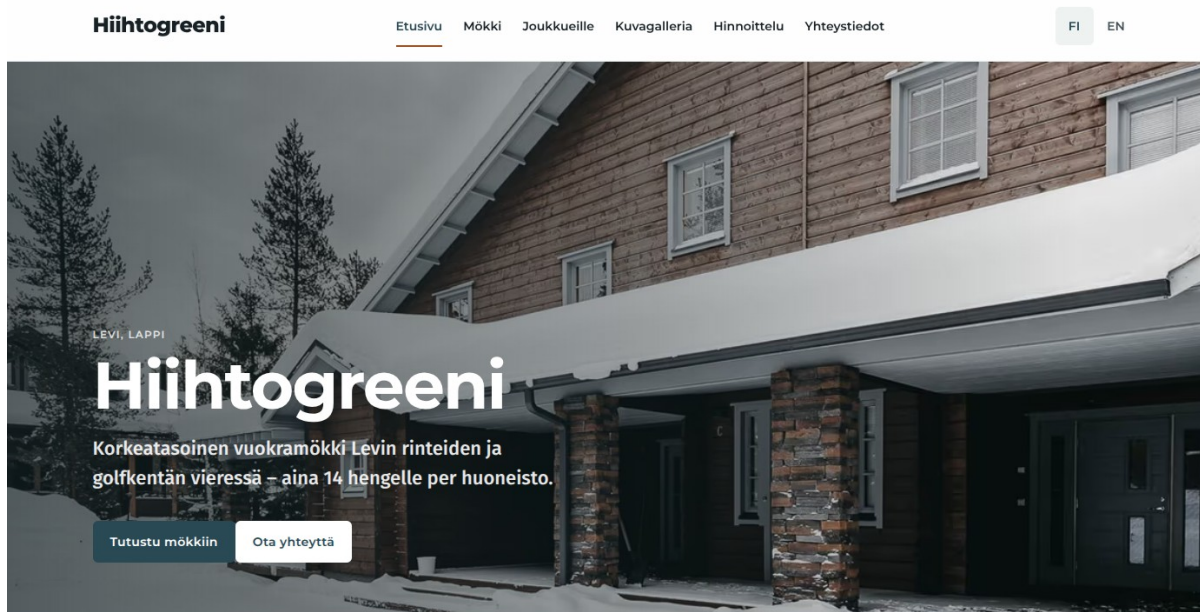
Vaihe 1: Älä rakenna uutta ennen kuin ymmärrät vanhan

Aloitin projektin dokumentoimalla kaiken, mitä vanha sivusto oli syönyt sisälleen. Käytin Kiloclaw-palvelua (Hostattu OpenClaw) vanhan sivuston "raapimiseen" (scraping) . Annoin Openclaw:n tehtäväksi kerätä paitsi näkyvän tekstin, myös sivurakenteet, linkit, kuvaviittaukset ja yhteystiedot . Tekoäly ei saanut heti tehtäväksi koodata uutta etusivua – se olisi ollut nopeaa, mutta tällainen YOLO-tyyppinen vibekoodaus jättää jälkeensä paljon bugeja ja konteksti-ikkunan täyttyessä agentti alkaa sekoilla, jolloin monet kriittiset kohdat voivat jäädä suorittamatta tai agentti keksii niihin jonkun ratkaisun, joka ei ole aidosti toimiva. Sen sijaan pyysin GPT-5.6 Sol -mallia vertaamaan kerättyä dataa live-sivustoon ja luomaan kronologisen migraatiosuunnitelman.

"Tekoäly voi tuottaa koodia erittäin nopeasti, mutta nopeus on arvokasta vain, kun syöte ja tavoitteet ovat oikeat. Visuaalisesti upea uusi sivusto on epäonnistunut projekti, jos tärkeät sivut katoavat Googlestä."

Auditointi paljastikin yllätyksen: kymmenen odotetun sivun lisäksi sivustolta löytyi yhdestoista, vanha englanninkielinen galleria-URL (</en/gallery/>), joka oli edelleen julkinen ja Googlen indeksoima . Jos olisimme vain sokeasti tehneet uuden sivuston, tämä osoite olisi katkennut.

Vaihe 2: Visuaalisuus ja brändin DNA



Teknologiaksi valikoitui Next.js 16, React 19, Tailwind CSS 4 ja TypeScript . Tämä yhdistelmä takaa salamannopean latauksen, sillä suurin osa sivuista pystytään generoimaan staattisena HTML:nä (SSG). JavaScriptiä ladataan selaimeen vain silloin, kun sitä oikeasti tarvitaan, kuten mobiilivalikossa, kuvagalleriassa ja yhteydenottolomakkeessa.

Ennen koodausta määrittelin AI:n kanssa visuaalisen suunnan, jolle annoimme nimen "Arctic calm, timber warmth". Tavoitteena oli säilyttää tunnistettava brändi, mutta uudistaa sitä 2020 luvulle: vahvat talvikuvat, erinomaiset sisätilat, tuttu sinivihreä sävy (#486C7A) ja lämmin oranssi korostusväri.

Uudistuksen myötä nämä elementit tuotiin puhtaampaan Tailwind-teemaan paremmilla kontrasteilla ja saavutettavuudella . Tässä vaiheessa rakensin AI:n avulla jaetun design-systeemin (vaihe 4), joka maksoi tekoälyn osalta tasan 0 € käyttäen ilmaista hy3-mallia (ilmainen viikon kokeilujakso Kilo Codessa). Huomionarvoista on, että navigointi ja kielivalikot renderöidään staattiseen HTML:ään . Sivusto on siis käytettävissä myös silloin, jos selaimen JavaScript jostain syystä kaatuisi.

Vaihe 3: SEO-varmistus ja staattisten tiedostojen pelastus

Tässä vaiheessa voisi mennä isosti vihkoon. WordPress tallentaa kuvat ja tiedostot usein kansioihin tyyliin `/wp-content/uploads/2022/10/...` . Ulkoiset sivustot ja Googlen kuvahaku saattavat yhä linkittää näihin osoitteisiin. Varmistaakseni, ettei mikään mene rikki, loin uuden Next.js-projektin `public/` -kansioon tismalleen saman WordPressin alkuperäisen kansiorakenteen kuville ja tärkeälle PDF-varausehdoille . Jokainen vanha kuva palauttaa nyt aidon 200 OK -vastauksen uudelta palvelimelta – ilman yhtäkään turhaa uudelleenohjausta.

Uudelleenohjaukset (Redirects) ja "410 Gone" SEO-turvallisuuden ydin rakennettiin vaiheessa 8 . Loimme säännöt, jotka käsittelevät vanhat URLit kohteliaasti:

Vanha englanninkielinen etusivu ja roikkumaan jäänyt galleria ohjattiin pysyvästi (301) uusiin osoitteisiin </en/home/> ja </en/photo-gallery/>.

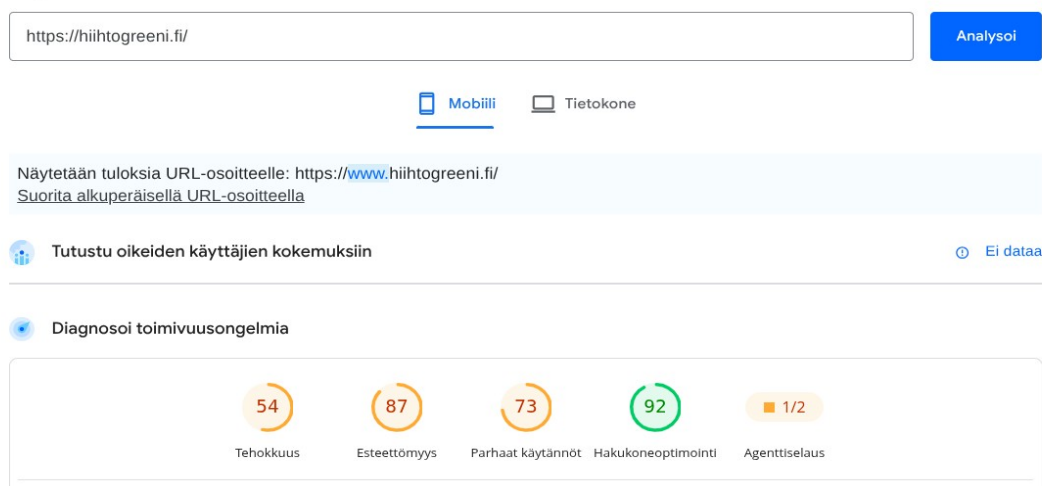
WordPressin vanha sivukartta </wp-sitemap.xml> ohjattiin uuteen sitemap-tiedostoon.

WordPressin vanhat reitit kuten </wp-admin/> , </feed/> ja </wp-json/> eivät ohjaa etusivulle (mikä olisi virhe), vaan ne palauttavat 410 Gone -vastauksen. Se on selkeä signaali hakukoneille, että näitä järjestelmäpolkuja ei enää ole, mikä lopettaa bottien turhan ryömimisen.

Vaihe 4: Suorituskyky, LCP ja manuaalisen työn arvo

Vaikka tekoäly teki ison työn, se ei ole erehtymätön. Vaiheen 10 manuaalisessa laadunvarmistuksessa havaitsin kaksi kriittistä asiaa, jotka vaativat ihmisen puuttumista. Ensinnäkin mobiilivalikko. Se toimi automaattisissa testeissä täydellisesti, mutta todellisella puhelimella valikko jäi sumennetun otsikkopalkin sisään jumiin CSS:n backdrop-filter - ominaisuuden vuoksi. Korjasin asian renderöimällä valikon portaalina suoraan document bodyyn . Tällaiset asiat paljastuvat vain, kun ihminen kokeilee käyttöliittymää oikealla laitteella.

Toiseksi Lighthouse LCP -suorituskyky (Largest Contentful Paint). Mobiilin suorituskykypisteet jäivät aluksi 72:een, koska suuri hero-kuva latautui raskaina JPEG-tiedostoina vanhan WP-rakenteen mukaisesti . Ratkaisin tämän luomalla hero-kuville rinnakkaisen LCP-latausputken uusilla AVIF- ja WebP-formaateilla, hyödyntäen <picture> - elementtiä ja preloadia. Vanhat kuvat jätettiin rauhaan ulkoisia linkkejä varten, mutta uusien formaattien myötä suorituskykypisteet nousivat 95:een (LCP 0.6 sekuntia).



Wordpress-sivuston pagespeed analyysi

Next.js-sivuston pagespeed analyysi

Staattisella sivustolla tarvitaan palvelinratkaisu yhteydenottojen vastaanottamiseen. Valitsin Resendin, joka on minulla useissa projekteissa käytössä. Se konfiguroitiin lähettämään viestit turvallisesti Next.js:n palvelinpuolen (Server Actions / Route) kautta. Selaimessa ei ole mitään API-avaimia, ja palvelin suodattaa roskapostin ja estää tuplaklikkaukset itsenäisesti. Ennen uudistusta lomakkeen kautta saattoi tulla useita useita roskaposteja päivässä. Uudistuksen jälkeen roskapostien määrä on tippunut tasan nolnaan. Mikäli roskapostit alkavat jossain vaiheessa pääsemään läpi, niin onneksi minulla on vielä takataskussa pari niksiä, joilla se voidaan korjata ilman, että käyttäjäkokemus kärsii

Lähetä viesti

Nimi *

Sähköposti *

Puhelin (valinnainen)

Viesti *

Tyyppi

☒ Sähköpostitse ☐ Tekstiviestillä

☐ Minulle saa lähettää tarjouksia ja informaatiota tulevaisuudessa

Lähetä viesti

Lopuksi sivustolle tuotiin aiemmista projekteista portattu oma evästehallinta (Cookie Consent). Se on oikeasti rehellinen: oletuksena se näyttää, että sivusto asettaa tasan yhden välttämättömän evästeen (suostumuksen tallentamisen) . Google Analytics 4 asennettiin Consent Mode v2 -tilassa, jolloin analytiikkakoodi aktivoituu vasta, kun käyttäjä antaa siihen nimenomaisen luvan . Seuraamme vain konversioita (lomakkeen lähetyks, sähköpostin klikkaus, vihreä fee -klikkaukset) keräämättä henkilötietoja .

Projektin luvut ja faktat

Kulutettu aika ja panostus (arvioitu vaiheiden kestosta) :

Tiedonkeruu, suunnittelu ja pohjatyöt (Vaiheet 1-4):	~1 h 50 min
Sivujen sisältö, lomake, SEO ja säännöt (Vaiheet 5-9):	~1 h 20 min
QA, suorituskykykorjaukset ja tuotantojulkaisu (Vaihe 10):	~3 h 30 min
Evästeet ja GA4 (Vaiheet 13-14):	~1h 20 min
Yhteensä aktiivista työaikaa: n.	8 tuntia

Käytetyt kielimallit ja agentit :

Kiloclaw (OpenClaw): Vanhan sivuston "skreippaus" (scrape) ja sisällön eristys.

GPT-5.6 Sol (Kilo Code): Arkkitehtuurisuunnittelu, migraatiostrategia, monimutkaiset tarkistukset ja julkaisuvaiheen auditoinnit.

Qwen3.7 max: Arkkitehtuurin pohjatyöt ja CI-putken pystytys.

Hy3 (ilmainen kokeilujakso ja muutenkin äärimmäisen kustannustehokas): Toistuvat käyttöliittymäkomponentit, komponenttien kasaaminen ja redirect-logiikat (tehokasta rutiinityötä 0 € kuluilla).

Kulutetut eurot (Tekoälyn API-kustannukset) :

Qwen3.7 max ja GPT-5.6 Sol	(Vaiheet 1-2):	~2,50 €
GPT-5.6 Sol	(Vaihe 10 QA):	~2,80 €
GPT-5.6 Sol	(Pre-launch SEO -auditointi):	~3,00 €
GPT-5.6 Sol	(Launch & DNS -tarkistukset):	~0,90 €
GPT-5.6 Sol	(GA4 & Evästeet):	~0,35 €

Tekoälyn kokonaiskustannukset yhteensä: n. **9,55 €**

Muut palvelut: Purelymail-sähköposti 9 € / vuosi.

Yhteenveto: Tekoälyn arvo web-kehityksessä

Hiihtogreeni.fi -sivuston uudistus on malliesimerkki siitä, miten tekoälyä kannattaa ohjelmistokehityksessä käyttää. Tekoäly ei korvaa ihmisen ajattelua liiketoiminnan ja hakukoneoptimoinnin ymmärrystä vielä vaan toimii tehokkaana työkaluna.

Agenttipohjainen koodaus on äärimmäisen nopeaa. Itse en toimita koodia YOLO-modella, vaan pysyn koko ajan matkassa mukana. Minulla on AGENTS.md tiedostossa aina mukana ohjeistus siitä, että agentti koodaa vain yhden taskin kerrallaan, jonka jälkeen käymme sen yhdessä läpi. Se kirjoittaa myös jatkuvasti memomy.md tiedostoa, johon se kokoaa kunkin session aikaansaannokset ja merkkää suunnitelmaan tehdyt tehtävät valmiiksi.

Tässä projektissa suurin arvo ei syntynyt siitä, että koodia syntyi nopeasti. Suurin arvo syntyi siitä, että pystyin ohjaamaan mallit miettimään tarkasti hreflang -tägejä, staattista generointia, 410 Cone -vastauksia ja vanhojen sisältöpolkujen suojelua .

Alle kymmenen euron AI-kuluilla ja reilun työpäivän panoksella vuosikymmenen vanha, hidas WordPress-asennus muutettiin moderniksi, salamannopeaksi Next.js-sovellukseksi, joka turvaa yrityksen digitaalisen omaisuuden pitkälle tulevaisuuteen. Ja mikä tärkeintä: kun uusi sivusto julkaistiin tuotantoon, Googlen silmissä mikään ei ollut kadonnut, kaikki oli vain muuttunut melkoisesti paremmaksi.