Engineer Software

{ Website Development & CMS }

A website you can update yourself

A site that is quick, findable and editable by your own staff, without a bill for every change of wording.

Still working out what you need?

Tell us what is going wrong, or what you are trying to build. We will tell you what it would take, roughly what it would cost, and whether we are the right people for it.

From 1,400 DKK an hour, excluding VAT — and less per hour on a longer booking.

Get in touch See what it costs

Selected work demonstrating this service.

Turn the service into a result

Make “Website Development & CMS” serve a business outcome. We can connect the technical work to the users, operations and decisions it needs to support. A site that is quick, findable and editable by your own staff, without a bill for every change of wording.

Get in touch

Examine the relevant work

Use the related work on this page to examine our approach. Ask about the tradeoffs and the conditions behind each result. Developed 100 web applications using HTML/HTML5 — 100 web applications built

Developed 100 web applications using HTML/HTML5 Get in touch →

Developed 100 web applications using HTML/HTML5

Developed 100 web applications using HTML/HTML5, CSS/SCSS, Django, WordPress, and Joomla, ensuring diverse online presence.

Designed 30 websites from Photoshop to HTML

Designed 30 websites, delivering unique and flexible solutions by converting Photoshop designs to HTML.

Administered 40 websites on Apache and Nginx

Administered 40 websites on Ubuntu Linux hosting servers with Apache and Nginx, ensuring high availability and performance.

Built the Next.js 16 frontend: SSR, SSG and CSR

Built the Next.js 16 frontend with deliberate SSR, SSG and CSR strategies, and a reusable prefetched-server-page factory that removes the N+1 fetch cascade from each authenticated page moved onto it.

Delivered full Progressive Web App support

Delivered full Progressive Web App support — installable and offline-capable — with Workbox runtime caching via next-pwa.

Built and maintained Django apps for IPTV

Developed and maintained Django web applications for the IPTV platform, shipping new features and improving performance and stability.

Maintained a Hugo site and improved product UI

Maintained and enhanced the legacy Hugo static-site website while contributing UI/UX improvements to the primary asset-management product.

Combined web development with SEO and content

Drove client web success by combining custom website development and design with SEO, content strategy and copywriting.

Your code on your own server

Deployed the company's own git forge on Soft Serve, private by default with no web panel and its SSH port bound to loopback behind a jump host, and made the landing page in front of it a build artefact of the main site rather than a hand-kept copy.

Built a multilingual Hugo site in 3 languages

Built a multilingual static site in Hugo across 108 template files including 53 partials, publishing every page in four representations from one content tree in three languages, and again under seven focused subdomains built from that same tree.

Accessibility checks across 32 pages and both themes

Implemented WCAG 2.2 Level AA checks across 32 representative pages in both colour themes, with a conformance record, one AAA criterion checked and the partial result for criterion 1.4.13 stated explicitly.

The accessibility faults your checker misses

Found and fixed 11 accessibility defects the browser reported as healthy — eight footer links still in tab order behind pointer-events, a scroll timeline clipping the colophon off three pages, and a rule engine running 70 of its 105 rules.

Generated 495 pages from a SQLite export

Generated 495 achievement pages across three languages from a read-only SQLite export, with the page address authored as data so that correcting a sentence no longer moved the page and broke the link.

Made the reduced-motion setting honest

Made reduced-motion honest after finding eleven selectors still animating under the preference, because a universal transition-none rule loses on specificity to any rule carrying a class.

Built a 637-line print stylesheet

Built a 637-line print stylesheet against four documented rendering-engine behaviours, including run-in headings that printed white on white for any reader whose browser preferred dark.

Mirrored the site to Gemini and Gopher

Built Gemini and Gopher outputs from the same website source; the achievement records 777 documents in each format at that stage, with zero HTML bytes changed by adding the formats.

Replaced 963 metadata blocks with one graph

Replaced 963 anonymous structured-data blocks that restated the company 1,671 times with one linked graph of 16 types minted from stable origin identifiers.

Cut the stylesheet bundle from 87 KB to 31 KB

Cut the stylesheet bundle from 87 KB to 31 KB by taking a base64 font out of it, and dropped 756 KB across 22 files that were published on every deploy and referenced by nothing.

Fixed a sitemap with one shared timestamp

Fixed a sitemap where 172 of 176 URLs shared one modification timestamp, by taking the date from git history after establishing that the export rewrites every file on every run.

Passed 106 of 106 PDF/UA-1 profile checks

Recorded PDF/UA-1 validation across nine documents with 106 of 106 configured accessibility-profile checks passing, while the archival variant failed one of 146 rules.

Built the visual identity from 2 drawings

Built the company's icon and favicon sets from two hand-made drawings, with every derived asset regenerated by script and a check that fails when a derived file was committed before its source.

Published a readable Tor onion mirror

Published the site as a Tor onion mirror on a readable address beginning engineer, with the key generated off the host so it never reached this repository or a laptop, and left it the one front door of four that is never counted.

Looking for Website Development & CMS?

Use the linked work to judge relevant experience. Tell us what you need to build; we can discuss the scope, delivery approach and handover.

Get in touch Full portfolio

Scope the work before commissioning it

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Talk through your requirements

Tell us what you need from “Website Development & CMS”, including the deadline and the team that will maintain it. We can discuss a practical way forward.

Get in touch

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://software.engineer.company/services/web-development/