Engineer Programvare

{ Frontend-utvikling }

Skjermbildene folk faktisk bruker

Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Avklarer dere fortsatt hva dere trenger?

Fortell oss hva som går galt, eller hva dere prøver å bygge. Vi sier hva det vil kreve, omtrent hva det vil koste, og om vi er de rette til oppgaven.

Fra 1 400 DKK timen, eksklusive mva. — og mindre per time ved et lengre oppdrag.

Ta kontakt Se hva det koster

Utvalgt arbeid som viser denne tjenesten.

Gjør tjenesten til et resultat

La «Frontend-utvikling» støtte et forretningsmål. Vi kan knytte det tekniske arbeidet til brukerne, driften og beslutningene det skal støtte. Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Ta kontakt

Undersøk det relevante arbeidet

Bruk arbeidet på denne siden til å undersøke tilnærmingen vår. Spør om avveiningene og forutsetningene bak hvert resultat. Styrte fullstack-utviklingen av GIS-kart — fullstack-kartutvikling, fra database til nettleser

Styrte fullstack-utviklingen av GIS-kart Ta kontakt →

Styrte fullstack-utviklingen av GIS-kart

Styrte fullstack-utviklingen av GIS-kart med ansvar for PostgreSQL, Mapbox, ReactJS og NodeJS, og leverte en integrert løsning.

Presenterte 200 UI/UX-forbedringer

Presenterte 200 UI/UX-forbedringer for GIS-kartapplikasjonen og økte inntektene 10 ganger gjennom bedre programvarefunksjoner.

Bidro i designprosessene for brukergrensesnitt

Bidro i designprosessene for brukergrensesnitt og sikret intuitive, visuelt tiltalende og brukervennlige grensesnitt i prosjektene.

Utviklet 100 webapplikasjoner med HTML/HTML5

Utviklet 100 webapplikasjoner med HTML/HTML5, CSS/SCSS, Django, WordPress og Joomla, og sikret en variert tilstedeværelse på nett.

Designet 30 nettsteder fra Photoshop til HTML

Designet 30 nettsteder og leverte unike og fleksible løsninger ved å konvertere Photoshop-design til HTML.

En maritim plattform med klare systemgrenser

Utformet arkitekturen for en lagdelt maritim plattform som skiller en Next.js PWA-frontend, et Go-API (Huma/Fiber) og et funksjonslag i PostgreSQL, med all forretningslogikk i databasen.

Bygde bytte av organisasjonskontekst for brukerne

Designet et system for å bytte organisasjonskontekst med localStorage på klienten og speiling i informasjonskapsler på serversiden, slik at brukere kan opptre som organisasjoner de administrerer, med autorisasjon etter prinsippet om minste privilegium.

Bygde Next.js 16-frontenden: SSR, SSG og CSR

Bygde Next.js 16-frontenden med bevisste strategier for SSR, SSG og CSR, og en gjenbrukbar fabrikk for forhåndshentede serversider som fjerner N+1-kaskaden av hentinger fra hver autentisert side som flyttes over på den.

Leverte full støtte for Progressive Web App

Leverte full støtte for Progressive Web App — kan installeres og fungerer uten nettforbindelse — med caching under kjøring i Workbox via next-pwa.

Designet «Liquid Glass»-designsystem for iOS 26

Designet et «Liquid Glass»-designsystem for iOS 26 og et kanonisk komponentregister håndhevet av lint-regler for å hindre at brukergrensesnittet spriker.

Designet produktets UI og UX fra ende til annen

Designet produktets UI og UX fra ende til annen — katalogrutenett, doble kort- og tabellvisninger, fortløpende validering av krav og brødsmulenavigasjon.

Bygde et engelsk/dansk i18n-system

Etablerte et engelsk/dansk internasjonaliseringssystem med 12 027 meldinger per språkversjon fordelt på 458 navneromsfiler, med vokabular håndhevet av linter og et budsjett på 400 linjer per fil.

Tok plattformens grunnleggende tekniske valg

Tok plattformens grunnleggende beslutninger i ukene etter at repoet ble opprettet i november 2025 — lagdelingen, datatilgang med databasen først og kravet om null advarsler — og de står seg fortsatt ni måneder senere.

Vedlikeholdt Hugo-nettsted, bedret produktets UI

Vedlikeholdt og videreutviklet det eldre statiske Hugo-nettstedet, og bidro samtidig med UI/UX-forbedringer til hovedproduktet for forvaltning av eiendeler.

Bygde operatørenes backoffice for plattformen

Bygde back office for operatørene — rundt 40 administrasjonsruter og 128 komponenter for eierskapskrav, moderering, funksjonsflagg, cache og diagnostikk — slik at plattformen kan driftes uten databasetilgang.

Bygde et flerspråklig Hugo-nettsted på 3 språk

Bygde et flerspråklig statisk nettsted i Hugo med 108 malfiler, blant dem 53 partials, som publiserer hver side i fire representasjoner fra ett innholdstre på tre språk, og på nytt under sju fokuserte underdomener bygd fra det samme treet.

Et sikkerhetsnett som ikke lenger koster en halvtime

Kuttet nettstedets nettleserdrevne kvalitetsport fra 1 636 sekunder til 615 ved å kjøre kontrollene med de lengste først gjennom en worker-pool begrenset til fire baner, etter å ha målt at alfabetisk rekkefølge kostet 320 sekunder mot 224.

Byttet 27 skriftstørrelser mot en skala i 6 trinn

Erstattet 27 gjengitte skriftstørrelser, der de nærmeste naboene lå 0,6 % fra hverandre, med en Major Third-skala i seks trinn, og skrev linteren som avviser den tjueåttende.

Tilgjengelighetskontroller på 32 sider i begge temaer

Implementerte WCAG 2.2 Level AA-kontroller på 32 representative sider i begge fargetemaer, med en samsvarsoversikt, kontroll av ett AAA-kriterium og det delvise resultatet for kriterium 1.4.13 uttrykkelig angitt.

Feil i universell utforming som kontrollverktøyet deres overser

Fant og rettet 11 feil i universell utforming som nettleseren meldte var i orden — åtte bunntekstlenker som fortsatt lå i tabulatorrekkefølgen bak pointer-events, en scroll-tidslinje som klippet kolofonen av tre sider, og en regelmotor som kjørte 70 av sine 105 regler.

Lukket fargesystemet ved 28 farger

Lukket fargesystemet ved 28 dokumenterte farger med en linter som avviser en verdi som er malt, men udokumentert, dokumentert, men ikke malt, feilmålt, skrevet om som en literal, eller innenfor en perseptuell avstand på 0,02 fra en som allerede finnes.

Fant at det lyse temaet manglet et helt lag

Fant ut at det lyse temaet hadde manglet et helsides overleggslag helt siden det ble skrevet, ved å kreve at begge temaene maler hver lagdelte flate med like mange lag.

Gjorde innstillingen for redusert bevegelse ærlig

Gjorde innstillingen for redusert bevegelse ærlig etter å ha funnet elleve selektorer som fortsatt animerte under den, fordi en universell transition-none-regel taper på spesifisitet mot enhver regel med en klasse.

Bygde et utskriftsstilark på 637 linjer

Bygde et utskriftsstilark på 637 linjer mot fire dokumenterte atferder i gjengivelsesmotorer, blant dem innfelte overskrifter som ble skrevet ut hvitt på hvitt for enhver leser med nettleseren innstilt på mørkt.

Kuttet stilarkpakken fra 87 KB til 31 KB

Kuttet stilarkpakken fra 87 KB til 31 KB ved å ta ut en base64-kodet skrift, og fjernet 756 KB fordelt på 22 filer som ble publisert ved hver produksjonssetting og som ingenting refererte til.

Lintet selve kvalitetsporten på 10 242 linjer

Satte 10 242 linjer JavaScript i kvalitetsporten under en formaterer og en linter etter å ha slått fast at det var den største kodemengden i repoet og den eneste ingenting leste, og rettet 13 funn uten å undertrykke noen.

Bygde en native macOS-klient i Swift 6

Bygde en native meldingsklient for macOS i Swift 6 og SwiftUI — 11 141 linjer fordelt på 53 filer — over C-grensesnittet til en Rust-kjerne lenket inn som et statisk arkiv fra en låst revisjon.

En avgrenset hendelsesflyt i en macOS-app

Avgrenset hendelsesbuffere og logging etter en registrert macOS-hendelse med vekst på 41 MB i sekundet og 111 GB komprimerte sider på en maskin med 36 GB; en registrert full testkjøring reduserte kjernens logglinjer fra 610 996 til 1 411.

Ga alle 17 ikoner navn for skjermlesere

Ga hver kontroll med bare ikon i grensesnittet et navn for skjermlesere etter å ha funnet at send-knappen ble lest opp som «arrow up circle, button», og skrev linteren som krever etiketten innen åtte linjer fra ikonet.

Bygde et macOS-designsystem med 48 tokens

Bygde et designsystem med 48 tokens på plattformens eget glassmateriale og skrev linteren som avviser et magisk tall, en hardkodet skriftstørrelse eller en animasjon som ignorerer innstillingen for redusert bevegelse.

Leter dere etter Frontend-utvikling?

Bruk de tilknyttede arbeidene til å vurdere relevant erfaring. Fortell oss hva dere skal bygge; vi kan diskutere omfang, fremgangsmåte og overlevering.

Ta kontakt Hele porteføljen

Avgrens oppgaven før bestilling

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gå gjennom kravene med oss

Fortell hva dere trenger innen «Frontend-utvikling», inkludert fristen og teamet som skal vedlikeholde løsningen. Vi kan drøfte en praktisk vei videre.

Ta kontakt

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://software.engineer.company/nb/services/frontend-development/