Engineer Software

Frontend-Entwicklung

42 Leistungen

Die Teile Ihres Systems verbinden

Die unter „Frontend-Entwicklung“ gesammelten Arbeiten zeigen einen Teil eines funktionierenden Systems. Wir können helfen, diesen Teil mit dem Rest Ihres Betriebs zu verbinden. Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Eine vergleichbare Herausforderung finden

Folgen Sie einem Beispiel, das Ihrer Situation ähnelt. Die Rückschau beschreibt die Entscheidungen hinter dem Ergebnis und geht über die bloße Technologiebezeichnung hinaus. Full-Stack-GIS-Kartenentwicklung geleitet — Full-Stack-Kartenentwicklung von der Datenbank bis zum Browser

Full-Stack-GIS-Kartenentwicklung geleitet Kontakt aufnehmen →

Full-Stack-GIS-Kartenentwicklung geleitet

Die Full-Stack-Entwicklung einer GIS-Kartenanwendung mit PostgreSQL, Mapbox, ReactJS und NodeJS geleitet und eine integrierte Lösung umgesetzt.

200 UI/UX-Verbesserungen vorgestellt

200 UI/UX-Verbesserungen für die GIS-Kartenanwendung vorgestellt und durch erweiterte Softwarefunktionen eine Umsatzsteigerung um das 10-Fache erreicht.

An der Gestaltung von Oberflächen mitgewirkt

An der Gestaltung von Benutzeroberflächen mitgewirkt und intuitive, visuell ansprechende und benutzerfreundliche Projektoberflächen sichergestellt.

100 Webanwendungen mit HTML/HTML5 entwickelt

100 Webanwendungen mit HTML/HTML5, CSS/SCSS, Django, WordPress und Joomla entwickelt und eine vielseitige Online-Präsenz ermöglicht.

30 Websites von Photoshop in HTML umgesetzt

30 Websites gestaltet und durch die Umsetzung von Photoshop-Entwürfen in HTML individuelle, flexible Lösungen geschaffen.

Eine maritime Plattform mit klaren Systemgrenzen

Eine geschichtete maritime Plattform mit getrenntem Next.js-PWA-Frontend, Go-API (Huma/Fiber) und PostgreSQL-Funktionsschicht entworfen und sämtliche Geschäftslogik in der Datenbank gehalten.

Organisationswechsel für Nutzer entwickelt

Einen Wechsel des Organisationskontexts mit clientseitigem localStorage und serverseitiger Cookie-Spiegelung entworfen, sodass Nutzer im Namen verwalteter Organisationen handeln können, während das Prinzip minimaler Berechtigungen gilt.

Next.js-16-Frontend: SSR, SSG und CSR entwickelt

Das Next.js-16-Frontend mit bewusst gewählten SSR-, SSG- und CSR-Strategien sowie einer wiederverwendbaren Factory für vorab geladene Serverseiten entwickelt, die die N+1-Abfragekaskade auf jeder umgestellten authentifizierten Seite beseitigt.

Progressive Web App vollständig unterstützt

Vollständige Progressive-Web-App-Unterstützung umgesetzt — installierbar und offlinefähig — mit Workbox-Laufzeitcaching über next-pwa.

iOS-26-'Liquid Glass'-Designsystem entworfen

Ein iOS-26-Designsystem im Stil von 'Liquid Glass' und ein verbindliches Komponentenverzeichnis entworfen, das Lint-Regeln gegen abweichende UI-Umsetzungen absichern.

UI und UX des Produkts durchgängig gestaltet

UI und UX des Produkts durchgängig gestaltet — Verzeichnisraster, kombinierte Karten- und Tabellenansichten, laufende Anforderungsprüfungen und Breadcrumb-Navigation.

Anwendung mit Nonce-basierter CSP gehärtet

Die Anwendung mit Nonce-basierter CSP, HSTS, SameSite-Cookies, Datenbankrollen mit minimalen Berechtigungen und erneuten serverseitigen Zugangsprüfungen gehärtet.

Englisch-dänisches i18n-System aufgebaut

Ein englisch-dänisches Internationalisierungssystem mit 12.027 Nachrichten je Sprachversion in 458 Namespace-Dateien aufgebaut, mit per Linter durchgesetztem Wortschatz und einer Grenze von 400 Zeilen je Datei.

Technische Grundlagen der Plattform festgelegt

Die grundlegenden Plattformentscheidungen in den Wochen nach der Eröffnung des Repositorys im November 2025 festgelegt — Schichtenarchitektur, datenbankzentrierter Datenzugriff und Warnungsfreiheit — die neun Monate später weiterhin gelten.

Django-Apps für IPTV entwickelt und gewartet

Django-Webanwendungen für die IPTV-Plattform entwickelt und gewartet, neue Funktionen ausgeliefert sowie Performance und Stabilität verbessert.

Hugo-Website betreut und Produkt-UI verbessert

Die bestehende statische Hugo-Website gepflegt und erweitert sowie UI/UX-Verbesserungen zum zentralen Asset-Management-Produkt beigetragen.

Webentwicklung mit SEO und Inhalten verbunden

Kunden durch die Verbindung individueller Website-Entwicklung und Gestaltung mit SEO, Inhaltsstrategie und Texterstellung zu einem erfolgreichen Webauftritt verholfen.

Testsuite für Go, SQL und Browser aufgebaut

Eine mehrschichtige automatisierte Testsuite aufgebaut — 981 Go-Tests, 543 Frontend- und Browserspezifikationen, 494 verhaltensprüfende SQL-Tests — mit Mutationstests, eigenschaftsbasierten Tests und einer verbindlichen Barrierefreiheitsprüfung.

Verwaltungs-Backend der Plattform entwickelt

Das Verwaltungs-Backend für den Betrieb entwickelt — rund 40 Administrationsrouten und 128 Komponenten für Besitzansprüche, Moderation, Funktionsschalter, Cache und Diagnostik — damit sich die Plattform ohne Datenbankzugriff betreiben lässt.

Unternehmensbesitzansprüche und Prüfung umgesetzt

Die Bearbeitung von Unternehmensbesitzansprüchen durchgängig umgesetzt: Ein Nutzer beansprucht ein Unternehmen, ein Administrator entscheidet und eine Genehmigung verändert den Berechtigungsgraphen, der die Bearbeitungsrechte festlegt.

Soziale Ebene mit Beiträgen und Gruppen aufgebaut

Die soziale Ebene der Plattform entwickelt — Beiträge, Feed, Gruppen, Erwähnungen, Follower-Graph und eine Übersicht anstehender Anlässe — auf derselben funktionszentrierten Datenschicht wie das übrige Produkt.

Hugo-Website mehrsprachig in 3 Sprachen aufgebaut

Eine mehrsprachige statische Hugo-Website mit 108 Vorlagendateien einschließlich 53 Partials aufgebaut, die jede Seite aus einem Inhaltsbaum in drei Sprachen und vier Darstellungsformen veröffentlicht, zusätzlich auf sieben spezialisierten Subdomains aus demselben Baum.

Commit-Prüfung mit 22 Lintern aufgebaut

Eine verbindliche Commit-Prüfung aus 22 einzeiligen Lintern und fünf ausführlicheren Prüfungen aufgebaut, ohne Warnungsstufe und ohne Inline-Unterdrückungen, für HTML, CSS, JavaScript, Python, YAML, Markdown, Shell, Links, Rechtschreibung, Geheimnisse und Typografie.

Ein Sicherheitsnetz, das keine halbe Stunde mehr kostet

Die browsergestützte Qualitätsprüfung der Website von 1.636 auf 615 Sekunden verkürzt, indem die Prüfungen in einem auf vier parallele Ausführungen begrenzten Worker-Pool nach längster Laufzeit zuerst eingeplant werden, nachdem die alphabetische Reihenfolge 320 statt 224 Sekunden benötigt hatte.

27 Schriftgrößen durch 6-stufige Skala ersetzt

27 tatsächlich dargestellte Schriftgrößen mit teils nur 0,6 % Abstand durch eine sechsstufige Skala im Verhältnis einer großen Terz ersetzt und den Linter geschrieben, der eine achtundzwanzigste Größe zurückweist.

Barrierefreiheitsprüfungen auf 32 Seiten in beiden Farbschemata

Prüfungen für WCAG 2.2 Level AA auf 32 repräsentativen Seiten in beiden Farbschemata umgesetzt, mit Konformitätsnachweis, einem geprüften AAA-Kriterium und ausdrücklich benanntem Teilergebnis für Kriterium 1.4.13.

Barrieren, die das Prüfwerkzeug übersieht

11 Barrierefreiheitsfehler gefunden und behoben, die der Browser als unbedenklich gemeldet hatte — acht trotz pointer-events weiter per Tab erreichbare Footerlinks, eine Scroll-Timeline, die das Kolophon auf drei Seiten abschnitt, und eine Regelengine, die nur 70 ihrer 105 Regeln ausführte.

495 Seiten aus einem SQLite-Export erzeugt

495 Leistungsseiten in drei Sprachen aus einem schreibgeschützten SQLite-Export erzeugt und die Seitenadresse als eigene Daten festgehalten, damit eine Satzkorrektur nicht länger die Seite verschiebt und Links bricht.

Farbsystem auf 28 Farben begrenzt

Das Farbsystem auf 28 dokumentierte Farben begrenzt und einen Linter entwickelt, der verwendete, aber undokumentierte, dokumentierte, aber ungenutzte, falsch gemessene, als Literal neu geschriebene oder einer bestehenden Farbe bis auf einen Wahrnehmungsabstand von 0,02 ähnliche Werte zurückweist.

Fehlende Schicht im hellen Farbschema gefunden

Durch die Prüfung gleicher Schichtanzahlen auf allen mehrschichtigen Oberflächen beider Farbschemata festgestellt, dass dem hellen Farbschema seit seiner Entstehung eine ganzseitige Überlagerungsschicht fehlte.

Reduzierte Bewegung wirksam umgesetzt

Die Einstellung für reduzierte Bewegung wirksam gemacht, nachdem elf Selektoren darunter weiterhin animierten, weil eine universelle transition-none-Regel in der Spezifität gegen jede Regel mit einer Klasse verliert.

Druckstylesheet mit 637 Zeilen entwickelt

Ein Druckstylesheet mit 637 Zeilen anhand von vier dokumentierten Verhaltensweisen von Rendering-Engines entwickelt, einschließlich eingebundener Überschriften, die bei Lesern mit dunkler Browsereinstellung weiß auf weiß gedruckt wurden.

Website nach Gemini und Gopher gespiegelt

Gemini- und Gopher-Ausgaben aus derselben Website-Quelle entwickelt; die Leistung dokumentiert für diesen Stand 777 Dokumente je Format, wobei durch das Hinzufügen der Formate null HTML-Bytes geändert wurden.

963 Metadatenblöcke durch einen Graphen ersetzt

963 anonyme Blöcke strukturierter Daten, die das Unternehmen 1.671-mal erneut beschrieben, durch einen verknüpften Graphen mit 16 Typen aus stabilen Identifikatoren auf Basis des eigenen Web-Ursprungs ersetzt.

Stylesheet von 87 KB auf 31 KB verkleinert

Das Stylesheet-Bündel durch das Auslagern einer Base64-Schrift von 87 KB auf 31 KB verkleinert und 756 KB in 22 bei jeder Bereitstellung veröffentlichten, aber nirgends referenzierten Dateien entfernt.

Qualitätsprüfung mit 10.242 Zeilen selbst geprüft

10.242 Zeilen JavaScript der Qualitätsprüfung unter Formatter und Linter gebracht, nachdem sie sich als größter und zugleich ungeprüfter Codebestand des Repositorys erwiesen hatten; 13 Befunde behoben und keinen unterdrückt.

Nativen macOS-Client in Swift 6 entwickelt

Einen nativen macOS-Messaging-Client in Swift 6 und SwiftUI entwickelt — 11.141 Zeilen in 53 Dateien — auf der C-Schnittstelle eines Rust-Kerns, der aus einer festgelegten Revision als statisches Archiv eingebunden wird.

Eine begrenzte Ereignispipeline für eine macOS-App

Ereignispuffer und Protokollierung nach einem dokumentierten macOS-Vorfall mit 41 MB Wachstum pro Sekunde und 111 GB komprimierten Speicherseiten auf einem Rechner mit 36 GB begrenzt; ein dokumentierter vollständiger Testlauf reduzierte Kernprotokollzeilen von 610.996 auf 1.411.

Swift-6-Nebenläufigkeit über C abgesichert

Vollständige strikte Nebenläufigkeit von Swift 6 ohne eigene Actors eingeführt und eine blockierende C-Ereignisschleife über genau einen Produzenten, einen Verbraucher und eine Reihenfolge an den Main Actor angebunden, nachdem ein Task je Ereignis die für die Oberfläche nötige Reihenfolge verloren hatte.

Alle 17 Symbole für Screenreader benannt

Jedes ausschließlich durch ein Symbol dargestellte Bedienelement für Screenreader benannt, nachdem die Sendeschaltfläche als "Pfeil nach oben im Kreis, Schaltfläche" angekündigt worden war, und einen Linter geschrieben, der die Beschriftung innerhalb von acht Zeilen um das Symbol verlangt.

macOS-Designsystem mit 48 Tokens aufgebaut

Ein Designsystem mit 48 Tokens auf dem plattformeigenen Glasmaterial aufgebaut und einen Linter geschrieben, der unbenannte Zahlenwerte, fest codierte Schriftgrößen und Animationen zurückweist, die die Einstellung für reduzierte Bewegung ignorieren.

Fehlermeldungen ohne Schuldzuweisung verfasst

Die Fehlermeldungen der Anwendung anhand eines schriftlichen Tonalitätsstandards neu verfasst, nachdem eine abgewiesene Anmeldung dem Nutzer einen Tippfehler vorwarf, obwohl der Anbieter ein anwendungsspezifisches Passwort verlangte; den Fall mit einem anbieterspezifischen Test abgesichert.

Die technische Arbeit gezielt einsetzen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Das Thema in Ihren Kontext einordnen

Erzählen Sie uns, welche Rolle „Frontend-Entwicklung“ in Ihrem eigenen System spielt. Wir können helfen, die passende Leistung auszuwählen und die damit verbundene Arbeit zu definieren.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://software.engineer.company/de/categories/frontend/