Engineer

フロントエンド開発

42 件の実績

システムの各部分を結び付ける

「フロントエンド開発」に分類した実績は、稼働するシステムの一部分を示しています。その部分を他の業務につなげるお手伝いをします。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

サービスを見る お問い合わせ →

比較できる課題を探す

ご自身の状況に近い事例をご覧ください。振り返りでは、技術名だけでは分からない、結果を導いた判断を説明しています。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

フルスタックのGIS地図開発を指揮

PostgreSQL、Mapbox、ReactJS、NodeJSを統括してフルスタックのGIS地図開発を指揮し、統合されたソリューションを提供した。

UI/UX改善200件を提案

GIS地図アプリケーションに200件のUI/UX改善を提案し、機能強化を通じて売上を10倍に伸ばした。

ユーザーインターフェースの設計に貢献

ユーザーインターフェースの設計に携わり、直感的で見やすく、使いやすい画面を実現した。

HTML/HTML5でWebアプリケーション100個を開発

HTML/HTML5、CSS/SCSS、Django、WordPress、Joomlaを用いて100個のWebアプリケーションを開発し、多様なオンライン展開を支えた。

PhotoshopからHTMLへWebサイト30件を設計

PhotoshopのデザインをHTMLへ変換し、独自性と柔軟性を備えたWebサイト30件を設計した。

システムの境界を明確にした海事プラットフォーム

Next.jsのPWAフロントエンド、Go（Huma/Fiber）API、PostgreSQL関数層を分離し、業務ロジックをすべてデータベースに置く、多層の海事プラットフォームを設計した。

利用者の組織コンテキスト切り替えを構築

クライアントのlocalStorageとサーバー側Cookieへの複製を用いて組織の操作コンテキストを切り替える仕組みを設計し、最小権限の認可を維持しながら管理対象の組織として操作できるようにした。

Next.js 16でSSR・SSG・CSRの画面を構築

SSR、SSG、CSRを意図的に使い分けたNext.js 16フロントエンドを構築。再利用可能な事前取得付きサーバーページ生成処理に移行した認証済みページから、N+1の連鎖的なデータ取得を除去した。

Progressive Web Appに完全対応

next-pwa経由のWorkboxランタイムキャッシュで、インストール可能かつオフラインでも使える、完全なProgressive Web App対応を実装した。

iOS 26「Liquid Glass」デザインシステムを設計

iOS 26の「Liquid Glass」を基にデザインシステムと正本となるコンポーネント一覧を設計し、UIのばらつきを静的検査で防いだ。

製品のUIとUX全体を設計

一覧グリッド、カード・表の切り替え、リアルタイムの要件検証、パンくずナビゲーションを含む、製品のUI・UX全体を設計した。

nonce付きCSPでアプリケーションを堅牢化

nonce付きCSP、HSTS、SameSite Cookie、最小権限のデータベースロール、サーバー側の利用権限再検証により、アプリケーションを堅牢化した。

英語・デンマーク語のi18nシステムを構築

458個の名前空間ファイルに言語ごと12,027件のメッセージを持つ英語・デンマーク語の国際化基盤を構築し、用語と1ファイル400行の上限を静的検査で守った。

プラットフォームの基礎技術方針を策定

2025年11月のリポジトリ開設後の数週間に、多層構成、データベース中心のデータアクセス、警告ゼロという基礎方針を定め、9か月後も維持した。

IPTV用Djangoアプリを構築・保守

IPTVプラットフォームのDjango Webアプリケーションを開発・保守し、新機能の提供と性能・安定性の改善を行った。

Hugoサイトを保守し製品のUIを改善

従来のHugo静的Webサイトを保守・改善しつつ、主力の資産管理製品のUI/UX改善にも携わった。

Web開発にSEOとコンテンツを統合

独自のWebサイト開発・デザインにSEO、コンテンツ戦略、コピーライティングを組み合わせ、顧客のWebでの成果につなげた。

Go・SQL・ブラウザーのテスト群を構築

981件のGoテスト、543件のフロントエンド・ブラウザーテスト、494件のSQL動作テストを持つ多層の自動テスト群を構築し、ミューテーションテスト、プロパティベーステスト、アクセシビリティの品質ゲートを加えた。

プラットフォームの運用管理画面を構築

申請、モデレーション、機能フラグ、キャッシュ、診断を扱う約40本の管理ルートと128個のコンポーネントからなる運用管理画面を構築し、データベースへ直接アクセスせずにプラットフォームを運用できるようにした。

企業所有権の申請・審査を構築

利用者による企業所有権の申請から管理者の審査までを実装し、承認時に編集権限を決める認可グラフが更新される、一連の仕組みを構築した。

投稿・フィード・グループの交流機能を構築

投稿、フィード、グループ、メンション、フォロー関係、記念日のまとめを含むソーシャル機能を、製品の他の部分と同じ関数中心のデータ層上に構築した。

3言語の多言語Hugoサイトを構築

53個のpartialを含む108個のHugoテンプレートで多言語静的サイトを構築。1つのコンテンツツリーから3言語・4形式で各ページを公開し、同じツリーから7つの専門分野別サブドメインも構築した。

22静的検査のコミットゲートを構築

HTML、CSS、JavaScript、Python、YAML、Markdown、Shell、リンク、綴り、秘密情報、組版を対象に、1行で説明できる検査22個と詳しい説明が必要な検査5個からなるコミットゲートを構築し、警告扱いや行内抑制を認めない方針にした。

30分も待たずに済む品質検査

最長の検査から順に、最大4並列のワーカープールへ割り当て、ブラウザーを使うサイト品質ゲートを1,636秒から615秒に短縮。名前順では224秒に対して320秒を要するという測定を踏まえた。

27種類の文字サイズを6段階の体系に統一

隣り合うサイズの差が0.6%しかないものもあった27種類の文字サイズを、長3度比による6段階の体系へ置き換え、28種類目を拒否する静的検査を作成した。

32ページを両テーマでアクセシビリティ検査

両配色テーマの代表32ページにWCAG 2.2 Level AA検査を実装し、適合記録とAAA基準1項目の検査を備え、基準1.4.13の部分的な達成を明示した。

自動検査が見逃すアクセシビリティの問題

pointer-eventsで操作を隠してもTab移動に残るフッターリンク8個、3ページの奥付を切り落とすスクロールタイムライン、105ルール中70しか実行しない検査エンジンなど、ブラウザーが正常と報告した11件のアクセシビリティ不備を発見・修正した。

SQLiteエクスポートから495ページを生成

読み取り専用のSQLiteエクスポートから3言語495件の実績ページを生成し、ページのアドレスをデータとして定義することで、文章を直してもURLが変わらずリンクが切れないようにした。

配色を28色に限定

配色を文書化した28色に限定し、描画されるのに未記録、記録されるのに未使用、測定誤り、リテラルによる別表記、既存色との知覚的距離が0.02以内、という場合に失敗する静的検査を設けた。

ライトテーマに欠けた層を発見

両テーマで各多層面に同じ数の層が描画されることを検証し、ライトテーマに作成当初から全面オーバーレイ層が欠けていたことを発見した。

「動きを減らす」設定を正しく反映

クラス付きルールには、全要素のトランジションを無効にするルールが詳細度で負けるため、「動きを減らす」設定でもアニメーションを続ける11個のセレクターを修正し、設定が正しく効くようにした。

637行の印刷スタイルシートを構築

ブラウザーがダーク表示を優先する場合に白地に白で印刷される行内見出しなど、文書化した4つの描画エンジンの挙動に対処する637行の印刷用スタイルシートを構築した。

GeminiとGopherへサイトを複製

同じウェブサイトのソースからGemini版とGopher版を生成する仕組みを構築。この実績では、その時点で各形式777文書を記録し、形式の追加によって変更されたHTMLはゼロバイトだった。

963メタデータブロックを1グラフに統合

会社の説明を1,671回繰り返していた匿名の構造化データ963ブロックを、安定したオリジン識別子から生成する16型の連結グラフ1つへ置き換えた。

CSSバンドルを87 KBから31 KBに削減

base64フォントを取り出してスタイルシートバンドルを87 KBから31 KBに削減し、毎回デプロイされながら参照元がない22ファイル756 KBも除去した。

10,242行の品質ゲート自体に静的検査を適用

リポジトリ最大のコード群でありながら検査対象外だった品質ゲートのJavaScript 10,242行に整形・静的検査を適用し、13件の指摘を抑制せずに修正した。

Swift 6でネイティブmacOSクライアントを構築

固定リビジョンから静的アーカイブとしてリンクしたRustコアのCインターフェース上に、Swift 6とSwiftUIで53ファイル、11,141行のネイティブmacOSメッセージクライアントを構築した。

macOSアプリのイベントパイプラインに上限を設定

毎秒41 MBの増加と、36 GB端末で111 GBの圧縮ページを記録したmacOSの障害を受け、イベントバッファとログ出力に上限を設定。記録された全テスト実行では、コアのログ行が610,996から1,411へ減った。

C上にSwift 6の厳密並行処理を採用

インターフェースに必要な順序がイベントごとのタスク生成で失われると確認し、独自アクターを用いずSwift 6の完全な厳密並行処理を採用。ブロッキングCイベントループとメインアクターを、生産者1つ、消費者1つ、順序1つで接続した。

全17アイコンに読み上げ用の名前を付与

送信ボタンが「arrow up circle, button」と読み上げられる問題を受け、アイコンだけの全操作要素にスクリーンリーダー用の名前を付け、アイコンから8行以内のラベルを必須とする静的検査を作成した。

48トークンのmacOSデザインシステムを構築

プラットフォーム標準のガラス素材を使った48トークンのデザインシステムを構築し、マジックナンバー、固定文字サイズ、「動きを減らす」設定を無視するアニメーションを拒否する静的検査を作成した。

利用者を責めないエラーメッセージに改訂

実際にはプロバイダーがアプリ専用パスワードを要求していたのに、サインイン拒否を利用者の入力誤りとするエラー表示を受け、文書化した文体基準に従ってメッセージを書き直し、プロバイダー名を明示するテストを加えた。

技術的な取り組みを絞り込む

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

自社の状況に当てはめる

お使いのシステムで「フロントエンド開発」がどのように関わっているかをお聞かせください。適したサービスを選び、その業務範囲を定めるお手伝いをします。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/categories/frontend/