

職務経歴書

自己 PR

- 私は 13 年間のバレーボール競技経験を持ち、高校時代にはインターハイ準優勝、春の高校バレーで 3 位の実績を持っています。さらに、社会人としては V リーグという日本のトップバレーボールリーグに所属するなど、スポーツでの高い競技力を持っています。この経験から、チームでの連携や高い目標に向かっての努力の大切さを学びました。エンジニアとしては、初めての会社でフロントエンドの開発を 1 年間担当した後、現在はバックエンドをメインに開発しています。しかし、その経験を活かしてサブタスクとしてフロントエンドの開発も行っており、フルスタックとしての幅広いスキルを持っています。

スキル

- 言語
 - JavaScript ・ TypeScript ・ Go
- フレームワーク等
 - React ・ Next.js
- RDB
 - MySQL
- クラウド
 - AWS
 - API Gateway ・ AppSync ・ Aurora ・ CloudWatch ・ Cognito ・ DynamoDB ・ ECR ・ ECS ・ EC2 ・ Fargate ・ IAM ・ Lambda ・ SES ・ SQS ・ S3
- SaaS/PaaS
 - AWS
 - Bitbucket ・ CircleCI ・ GitHub ・ GitHub Actions
- その他
 - Docker ・ GraphQL ・ RSpec

職務要約

- 勤務先
 - 受託会社
- 役割
 - DX アプリケーション開発
- 業務内容
 - クライアントからの準委任契約のもと、労働時間を基にアプリの新規機能の開発、運用保守を担当しております。
 - クライアントとのコミュニケーションを密に取りながら、要件定義や設計、開発、運用までの一連の流れを担当。

職務経歴詳細

- タクシー産業 DX プロダクト
 - 期間
 - 2022/1/5 ~ 2022/3/31
 - チーム編成
 - PM 1 人
 - 営業アシスタント 1 人
 - 開発者 7 人
 - 言語・フレームワーク
 - React・Next.js
 - Jest・Storybook
 - インフラ
 - AWS
 - ツール
 - GitHub・GitHub Actions
 - プロダクトが属する業界
 - タクシー業界
 - プロダクト概要
 - タクシーアプリと連動して使用する日本初のタクシーデリバリーに特化したサービス
 - ウェブ上で注文から決済まで完結できるサービス
 - チームの特徴・課題
 - React に知見ある人が少ない中での開発
 - 具体的なタスクは次の通り
 - 各画面のモジュールの実装
 - レイアウトが統一されているためコンポーネント化して各画面によって開閉のタイミング、表示する文言を変更できるように実装
 - カレンダーの実装
 - ライブラリーを使用せずにカレンダーを実装して配送日時を選択しバックエンドに送信できる機能
 - 2 つの画面で表示するカレンダーのコンポーネント化
 - 2 つの画面でレイアウトが異なるため、props を使用して style で変更する処理
 - 選択された値をバックエンドに送る処理はカスタムフックを作成して共通化
 - ライブラリー(swiper)を使用して画像のスライド機能の実装
 - スライド機能のコンポーネント化
 - 画面によって表示するレイアウトのサイズ、スライドのアクションを変更できるように実装
 - 作成した画面のユニットテスト
 - storybook を使用して UI の構築
- 不動産 DX プロダクト
 - 期間
 - 2022/4/1 ~ 2022/8/31
 - チーム編成
 - PM 1 人
 - 営業アシスタント 1 人
 - 開発者 2 人
 - 言語・フレームワーク

- React
- インフラ
 - AWS
 - API Gateway・AppSync・Aurora・CloudWatch・Cognito・DynamoDB・ECR・ECS・EC2・Fargate・IAM・Lambda・SES・SQS・S3
- ツール
 - GitHub・GitHub Actions
- プロダクトが属する業界
 - 不動産業界
- プロダクト概要
 - 賃貸入居者と物件管理会社のやりとりをオンラインで完結するアプリケーション
 - 従来は、解約や騒音、水漏れなどの連絡は紙や電話を使用した方法が一般的で非常に非効率であった
 - 上記アプリケーションを導入することで、チャットベースでやりとりが行えるため、忙しい平日や夜中でも管理会社に連絡が可能になる
 - 他の機能として、掲示板機能、契約情報の確認、入居中サービス、各種申請など全てアプリケーション上から行える
- チームの特徴・課題
 - 導入数が業界首位の入居者管理アプリケーションを開発元の他社から引き継ぐことになり、一人目のリードフロントエンジニアとして参加
 - アサイン後、開発基盤を整備したことで、新機能実装や導入企業数が順調に増加し、東証グロース市場への上場に大きく貢献した
- チームでの役割
 - 新規機能開発、レビューの担当
 - 手を動かすだけの開発だけではなく、画面設計・実装・テスト・リリース作業も担当
- 具体的なタスクは次の通り
 - メールを配信できる数万人のユーザーをページング機能を用いて画面に表示
 - 表示されたユーザーの中から配信の有無を選択してチェックされているユーザーのみ id をバックエンドに送信する機能
 - 一括チェックボタンを使用して配信できる全ユーザーの id をバックエンドに送信する機能
 - 配信する内容を新規で作成、また過去に配信した内容を表示し編集でき、フロントで作成した内容をバックエンドに送信する機能
 - 複数の画面で使用されている右カラムの共通化
 - 7 画面で使用されている右カラムはコンポーネントを使用してレイアウトを共通化することでコード量を削減
 - 画面の中のスクロール位置を取得して、一番下までスクロールした際にデータをリクエストする実装
 - API が未実装の際にフロント側でレイアウト、リクエストの確認ができるように AppSync で処理を記して mock 化
 - GraphQL ファイルにバックエンドと連携して構造体、フィールド名の記述
- チームの課題と自身が工夫したこと
 - ドキュメントの不足により開発スピードに影響が出ていたため全てドキュメントとして作成しチームに共有することで、チーム全体の業務効率化や生産性向上に大きく貢献した
 - 具体的な内容は次の 4 つ
 - バックエンドにリクエストを送る方法
 - mockAPI の作成手順
 - フロント側のデプロイ手順
- 観光産業 DX プロダクト

- 期間
 - 2022/9/1 ~ 2023/3/31
 - チーム編成
 - PM 2 人
 - 営業アシスタント 1 人
 - 開発者 2 人
 - 言語・フレームワーク
 - React・Next.js
 - Jest・Storybook
 - ツール
 - GitHub・GitHub Actions
 - プロダクトが属する業界
 - 観光産業界
 - チームでの役割
 - 新規機能開発、レビューの担当
 - 手を動かすだけの開発だけではなく、画面設計も担当
 - 具体的なタスクは次の通り
 - Figma をもとにフロント画面の作成
 - react-pwa の導入
 - 対象デバイスがスマホのため、各画面のレスポンシブ化
 - QR コードの生成
- 健康産業 DX プロダクト
 - 期間
 - 2022/11/1 ~ 2023/3/31
 - チーム編成
 - PM 2 人
 - 営業アシスタント 1 人
 - 開発者 2 人
 - 言語・フレームワーク
 - React・Next.js
 - Jest・Storybook
 - ツール
 - Bitbucket
 - プロダクトが属する業界
 - 健康産業界
 - チームでの役割
 - 新規機能開発、レビューの担当
 - 手を動かすだけの開発だけではなく、画面設計も担当
 - 具体的なタスクは次の通り
 - Figma をもとにフロント画面の作成
 - react-chartjs-2 を使用したチャートの作成
 - react-hook-home を使用したバリデーション
 - 不動産 DX プロダクト
 - 期間
 - 2023/4/1 ~ 現在
 - チーム編成
 - PM 1 人

- 営業アシスタント 1 人
 - 開発者 2 人
- 言語・フレームワーク
 - React
 - Go
 - Serverless Framework
- インフラ
 - AWS
 - API Gateway・AppSync・Aurora・CloudWatch・Cognito・DynamoDB・ECR・ECS・EC2・Fargate・IAM・Lambda・SES・SQS・S3
- ツール
 - Bitbucket・Docker・GitHub・GitHub Actions・MySQL・Redmine
- プロダクトが属する業界
 - 不動産業界
- プロダクト概要
 - 2022/4/1 にアサインしていた時と同様
- チームでの役割
 - バックエンド、フロントエンドの新規機能開発、レビューの担当
 - 手を動かすだけの開発だけではなく、画面設計・テーブル設計・API 設計・実装・テスト・リリース作業も担当
- 具体的なタスクは次の通り
 - Go + クリーンアーキテクチャ + Lambda を用いた各種 API の作成
 - AppSync + Lambda + DynamoDB を使用した GraphQL 及びリゾルバーの作成
 - RDB や DynamoDB のテーブル設計およびインデックスの設定
 - IVR 認証機能の実装（バックエンド）
 - メディア 4u が提供している API と連携して実装
 - Controller 層、Repository 層の testing を使用したテーブル駆動のテスト
 - 通知の一括既読機能の実装