

職務経歴書

種別	内容
担当工程	要件定義, 設計, コーディング, テスト, 運用/保守
経験した役割	フロントエンド, バックエンド, インフラ, アプリ開発(Android)
使用技術	PHP/CakePHP, ReactNative, Python, Apache, MySQL
チーム情報	実装x1名(自分), 端末選定x1名, 現職社長x1名

サイネージアプリ(2022-12~ 2ヶ月)

PHP管理画面でプレイリストを作成し、Android端末からモニタに出力するアプリ。

【主な使用技術】 PHP/CakePHP, ReactNative, Python, mysqlworkbench

【担当工程】 要件定義, 設計, コーディング, テスト, 運用/保守

作業内容

内容	期間	技術	詳細
要件定義	約2日		- プレイリストコンテンツ内容の確認 (動画,画像,url,テキスト,,) - コンテンツ表示方法の確認 (表示間隔, モニタ向き,,) - 端末環境の確認 (端末スペック,通信環境,,) - 編集権限の確認 (本社と支店とそれぞれの関係)
DB設計	約1日	mysql workbench	柔軟性を維持しながら管理者と店舗の権限を明確にすること
技術選定	約1日	Google	- 端末の選定 (候補: シングルボードコンピュータか、Android端末) - Android開発言語の選定 (候補: Kotlin,ReactNative,Cordova) - 動画編集言語 (Python)
サーバー構築	約30分	Apache,Bash	自作スクリプトを利用しVPS構築
開発-管理画面	約8日	CakePHP, Shell	DBに基づいた登録が可能であること アプリの要求に正しくAPIを返すこと 動画処理の進捗状況が視覚的に確認できること
開発-アプリ開発	約5日	ReactNative	管理画面で登録したデータと連携ができること プレイリストを正しく再生すること
開発-m3u8拡張子導入	約3日	PHP, FFmpeg	ライブ配信で使われている拡張子(m3u8)を利用する クライアント側の読み込み速度を安定させること
開発-オフライン対応	約5日	ReactNative	プレイリストデータを外部ストレージに保存すること
開発-動画編集ツール開発	約4日	Python, MoviePy	プレイリストデータを1つの動画に変換すること

業務での実績

Android/ReactNativeの開発経験

ストレージ操作の知識が深まり、同時にAndroid端末への苦手意識もなくなりました。
固定端末向けのアプリを開発する際には、これらの知識を積極的に活用したいと考えています。

m3u8の活用

1秒単位で動画を分割することで、サーバー参照型の読み込みを安定化することができました。
使用する場面は限定的ですが、貴重な経験を積むことができました。

Pythonの開発経験

サーバー側での動画編集処理を提案し、実現することができました。
フルスタックエンジニアとしての経験と技術が活かせたと感じています。

教育支援システム(2022-06~ 4ヶ月)

課題に対する生徒の意見を形態素解析し、集計することが主な目的のシステム

【主な使用技術】 PHP/CakePHP, React, 自然言語処理, AWS, mysqlworkbench

【担当工程】 要件定義, 設計, コーディング, テスト, 運用/保守

作業内容

内容	期間	技術	詳細
要件定義	約2日		- クライアント提供の仕様書を読み認識をすり合わせる - 権限の確認(管理者, 学校, 先生, 生徒) - 形態素解析の方法(MeCab)
DB設計	約2日	mysql workbench	クライアント提供の仕様書からDBを設計すること 不明な点は旧システムを参考にする
サーバー構築	約半日	AWS	VPSと同じ要領で構築する
開発-API開発	約24日	CakePHP, Mecab	DBに基づいた登録が可能であること アプリの要求に正しくAPIを返すこと
開発-形態素解析	約3日	MeCab, PHP	旧システムから処理方法が変わらないように構築すること システム側から辞書の追加を行うこと
開発-フロント開発	約14日	React	管理画面で登録したデータと連携ができること
開発-汎用的な関数	約8日	React	再利用可能なページネーションコンポーネント 編集後の戻るConfirm機能の実装

業務での実績

形態素解析

形態素解析を学ぶことで、データ分析の実用性について再認識しました。

お問い合わせから検索方法など、応用が効く分野だと考えています。

内覧受付システム(2021-10~ 2ヶ月)

- プロジェクトカテゴリ: (WEBサービス, 受託開発)
- 担当工程: (設計, コーディング, テスト, 運用/保守)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, WordPress, Apache, MySQL)

スムーズな内覧受付機能を重視したホームページです。

チームの構成と立ち位置

- デザイナー x1名
- フロントエンド x2名
- テーマ制作、API開発 x1名 (自分)

開発内容・課題

1. 多言語化 1つの物件に対する項目量が多いため、日本語の枠の隣に英語用の枠を用意するのではなく、英語用のサイトとして、サイトごと複製することが最善の策だと判断しました。今思えば、DeepLで自動翻訳する策も視野に入れるべきだったかと悔いが残ります。

成果

REST APIを使う際のフロントエンド部署との連携の難しさと、実装した時の使いやすさを再認識しました。

そこで、まずはSwaggerを利用して可読性の高いAPI設計書を作成することと、自分自身でReactを学んで、連携まで実現することを目指すことに決めました。

販促システム(2021-07~2ヶ月)

- プロジェクトカテゴリ: (WEBサービス, 受託開発)
- 担当工程: (設計, コーディング, テスト, 運用/保守)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, SMS, OCR, Apache, ロードバランサ, MySQL)

抽選を行いキャッシュバックを約束することで、購買意欲を高めることを目的とするシステムです。

チームの構成と立ち位置

- CSSとHTMLの担当（外部）
- CSS以外の担当（自分）

開発内容・課題

1. 外部フロントエンジニアとの連携 自社のフロントエンドエンジニアと異なる書き方や仕様に戸惑いましたが、いい経験になりました。
 2. 不正抽選の防止 不正抽選を防ぐための認証および景品の送信のためにSMSを利用しました。選定から行い、料金仕様とAPIの豊富さからTwilioを選びました。
 3. 購入したかどうかを確認する作業の負担 LINEのOCRを導入することで、少しでもおかしい箇所を目視するだけで済む作業に簡略化しました
 4. サーバーへの負荷 全国規模で一斉に応募されることが想定されるため、ロードバランサを使い負荷を分散しました。
-

学内システム(2021-03~4ヶ月)

- プロジェクトカテゴリ: (WEBサービス, 業務システム, 受託開発)
- 担当工程: (設計, コーディング, テスト, 運用/保守)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, LDAPS, Apache, MySQL)

教職員と学生(約10,000人)へページを共有するシステムです。

チームの構成と立ち位置

- CSSとHTMLの担当 x2名
- CSS以外の担当 x1名 (自分)

開発内容・課題

1. 権限 なにより権限が重視されるため、いつも以上に緊張感を持って実装しました。
 - IP制限
 - 操作ログ
 - 部門ごとの閲覧権限
 - ファイルごとのアクセス権限
 - 教職員が登録したページの承認処理等
 - 承認前後のページアーカイブ
 2. 複数システムにわたるアカウント情報を一元管理する アカウント管理をしているLDAPサーバーと連携しました。LDAPの挙動を把握するため、別途LDAPS環境を構築しました。
-

見積システム(2020-06~ 3ヶ月)

- プロジェクトカテゴリ: (WEBサービス, 業務システム, 自社プロダクト)
- 担当工程: (設計, コーディング, テスト, 運用/保守)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, Backlog, Apache, MySQL)

業務の効率化、見積案件の整理、それらに紐づく実働工数の可視化を図るシステムです。入社して間もない頃に開発したシステムですが、多くのスタッフに利用されておりいいものを作れたと実感しています。

チームの構成と立ち位置

- デザイナー x1名
- CSSとHTMLの担当 x2名
- CSS以外の担当 x1名 (自分)

開発内容・課題

1. 見積書・請求書の自動生成 登録した担当者、見積もり内容と金額から、担当者の判子が押された状態の見積書や請求書を生成します。
2. 工数管理 タスク管理に利用していたBacklogと連携させることで、担当者ごとの進捗具合や案件ごとの実働工数を算出しました。
3. サイト管理 システム上でサイトを登録し、SSLの有効期限を監視します。