

職務経歴書

担当工程

上流工程から下流工程までを一貫して担当しています。

基本的なチーム構成は営業、デザイナー、フロントCSS、バックエンド、QAの5名体制になります。

| 担当 | 技術 | 内容 | 目標/課題 |
|---------|--------------------------------|---|--|
| 要件定義 | | WEB会議に開発担当として同席します | 口頭での説明力 |
| 詳細設計 | スライド | 画面遷移図を作成して確認を進めます テキストベースで質問・回答を行います | 効率化 |
| DB設計 | | 視認性・拡張性を意識してER図を作成します 規模感: 最大30テーブル、70万レコード 【その他】 インデックス, view, trigger | DBチューニング パーティショニング レプリケーション |
| インフラ | VPS, AWS ロードバランサ スケールアップ | OSからLAMP環境の構築 WAFやFail2banでのセキュリティ対策 Zabbixでのサービス監視 【その他】 LDAP, postfix, AWS-EC2 | AWS, CI/CD AWS Lambda API Gateway Github Action |
| バックエンド | PHP/CakePHP Python | 100 規模感: スケジュール4ヶ月, 見積もり400万円 | チーム開発 OSS開発/貢献 Go/Node.js/Java |
| フロントエンド | React | 使いやすさ・速度を重視します | SSG量 結合テスト |
| アプリ開発 | ReactNative | スムーズに動作するようメモリを意識します | |

サーバー構築

VPS構築をメインに、AWS-EC2の構築や、LDAPサーバー構築、ロードバランサ構築等、幅広く携わりました。
基本的にはマニュアル通りにLAMP環境+メールサーバー構築+セキュリティ対策を行います。
CentOS6からCentOS7、AlmaLinuxへの移行を担当し、脆弱性対策とともにマニュアルの更新を行いました。
サーバー構築に半日~夕方まで時間がかかっていたところ、スクリプトを作成し30分以下に短縮しました。
直近ではビルド構築されていたWEBサーバーを改修し、PHPモジュール等を手動で導入しました。

開発

自社CMSの改修- 汎用的関数のライブラリ化

こだわりは、応答時間を0.8秒以下にすることです。

処理を最適化した上で、改善が見込めない/難しい場合は他言語でも解決できないか模索します。

その結果としてバックエンド部署に所属しながらもフロントエンドにも携わりました。

- 時間のかかるサブコンテンツは、Reactを導入し非同期に
- ダイアログ処理が多いサイトには、Reactを導入し手軽に
- 動画を流す際に負荷が高い場合は、Pythonを利用し編集処理

プロジェクトまとめ

2ヶ月以上かかったプロジェクトの概要をまとめています。

その他にも1週間~2ヶ月未満の案件(HP作成,サーバ構築)に数多く携わっております。

わずかな待機時間には、自社開発、自社CMSの改修、スクリプト作成等に従事しております。

| プロジェクト名 | 期間(目安) | 金額(目安) | 主な技術 |
|----------|--------------|--------|------------------------------------|
| サイネージアプリ | 2022年12月~2ヶ月 | | CakePHP, ReactNative, Python |
| 教育支援システム | 2022年06月~4ヶ月 | 350万円 | CakePHP, React, 言語解析, AWS, Laravel |
| 内覧受付システム | 2021年10月~2ヶ月 | 200万円 | WordPress |
| 販促システム | 2021年07月~2ヶ月 | 200万円 | CakePHP, SMS, OCR, ロードバランサ |
| 学内システム | 2021年03月~4ヶ月 | 300万円 | CakePHP, LDAPS |
| 見積システム | 2020年06月~3ヶ月 | | CakePHP, Backlog |

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。

特徴

自社開発, 5,000円のAndroid端末, モバイル開発

作業内容

| 内容 | 期間 | 技術 | 詳細 |
|--------------|------|-----------------|---|
| 詳細設計 | 約3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 | | 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 | | - 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 | 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。

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。

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

特徴

自然言語処理, React導入, アクセス権限

作業内容

| 内容 | 期間 | 技術 | 詳細 |
|-----------|------|---------|---|
| 詳細設計 | | | スライド |
| DB設計 | 約3日 | | 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 | 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 | 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15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-関数の共通化 | 約5日 | React | 再利用可能なページネーション 編集後に戻る時のConfirm機能 |

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、応用が効く機能だと考えています。

内覧受付システム(2021-10~ 2ヶ月)

スムーズな内覧受付機能を重視したホームページ

特徴

WordPress, ブロックエディタの拡張, API開発

作業内容

| 内容 | 期間 | 技術 | 詳細 |
|----------------|------|-----------|--|
| 設計 | 約1日 | | 物件情報、部屋情報、お知らせ等 カスタムフィールドによるリレーション対応 |
| サーバー構築 | 約30分 | Bash | Bashスクリプトを利用し構築する |
| 開発-管理画面 | 約3日 | WordPress | DBに基づいた登録が可能であること 登録のしやすい登録画面であること |
| 開発-API開発 | 約3日 | WordPress | フロント側の要求に正しくAPIを返すこと |
| 開発-多言語対応 | 約1日 | WordPress | 英語用の登録を可能にする |
| 開発-ブロックエディタの拡張 | 約2日 | プラグイン | ブロックエディタのブロックを追加 |
| 開発-カスタマイズ | 約3日 | プラグイン | お問い合わせとカスタムフィールドを連携させる 物件等の情報CSVをインポート・エクスポートする |

業務での実績

これまでにWordPressを実装してきましたが、
データ登録の大変さ、集計方法、スピード面など、多くの面で悔いが残ってしまいました。

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実装しました。
今思えば、DeepLのAPIで自動翻訳する策も視野に入れるべきだったと思います。

CSVインポート・エクスポート

WPの制限が多いものの、通常のCMSと同じようにインポート対応しました。
実装時はCSVが前提にあったので画像のインポートは実用的ではないと考えていましたが、
ZIP形式なら対応できるのではないかと思います。

API連携

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。
フロントエンドとの連携を深めるためにも、可読性の高いAPI設計書(Swagger)を作成することと、自分自身Reactを
学ぶことを決めました。

販促システム(2021-07~2ヶ月)

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

特徴

SMS, OCR, ロードバランサ, rsync

作業内容

| 内容 | 期間 | 技術 | 詳細 |
|---------|------|------------------|--|
| 詳細設計 | 約4日 | | キャンペーンごとに複製することを考慮する |
| DB設計 | 約2日 | | キャンペーンごとに複製することを考慮する |
| 選定 | 約2日 | | SMS会社 OCR会社 |
| サーバー構築 | 約1日 | Bash | Bashスクリプトを利用し構築する 2つのサーバーをロードバランサを組む 振り分け先の2つサーバーをrsync同期させる |
| 開発-管理画面 | 約20日 | CakePHP, crontab | DBに基づいた登録が可能であること 登録のしやすい登録画面であること crontabによるスケジュール処理 |
| 開発-SMS | 約3日 | SMS | 応募時・登録時・結果発表全てにSMSを用いる 管理画面からのCSV送信にも対応すること 未着時にコールバックを行う |
| 開発-OCR | 約3日 | OCR | レシートをOCRで読み込み、目視での作業の簡略化を図る |
| 開発-フロント | 約3日 | | 外部フロントエンジニアとの連携 |

業務での実績

ロードバランサ

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って負荷を分散しました。
振り分け先のサーバ間をrsyncで同期させることで経験・技術も身につきました。

SMS

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。
到着率が課題となるため多用はできませんがよい経験です。

OCR

目視で行う購入確認作業を簡略化しました。
確認する項目が絞り込まれるだけでも効率的になったかと思います。

学内システム(2021-03~4ヶ月)

教職員と学生(約10,000人)へページを共有するシステム。

特徴

LDAPS, リッチエディタ, アクセス権限, 70万レコード, 20テーブル

作業内容

| 内容 | 期間 | 技術 | 詳細 |
|-----------|------|---------|--|
| 詳細設計 | 約3日 | | 用意された仕様書を読み認識をすり合わせる |
| DB設計 | 約3日 | | リッチエディタ登録ができること あらゆるテーブルにユーザー権限を付与すること あらゆる操作のログを残すこと 複数のページ種別/スラッグ対応 |
| サーバー構築 | 約30分 | Bash | Bashスクリプトを利用し構築する |
| 開発-管理画面 | 約15日 | CakePHP | DBに基づいた登録が可能であること 登録のしやすい登録画面であること |
| 開発-記事入力画面 | 約8日 | JS | WPのようなリッチエディタを実装すること それをHTMLとして出力すること |
| 開発-LDAP連携 | 約5日 | LDAP | アカウントはLDAPから取得すること テスト用のLDAPサーバーを構築すること |
| 開発-通知処理 | 約3日 | CakePHP | ページ追加・更新・削除時に通知を行うこと |
| 開発-フロント | 約10日 | | 登録した内容を正しく表示すること |

業務での実績

WPのようなリッチエディタ開発

WPのリッチエディタを再現しました。

より複雑なページ構成、コンテンツごとのオプションにも柔軟に対応できるようになりました。

複雑な権限管理

ほぼ全てのテーブルに権限を付与することになったので権限確認が大変でした。

緊張感を持ち無事に実装を終えることができました。

LDAP利用

複数システムにわたるアカウント情報をLDAPにて一元管理しました。

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)

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。

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。

特徴

自社開発, Backlog, PDF生成, SSL監視, React導入

業務での実績

PDF生成

見積書・請求書の自動生成を実装しました。

様々なフォーマットを用意することで書類作成作業を簡略化しました。

BacklogのAPI利用

タスク管理に利用していたBacklogと連携させることで、案件ごとの実働工数を算出しました。

Backlogに限らずツールのAPIを活用することで、タスク管理、実働工数、スケジュール管理を一元的に管理したいと考えました。

サイトデータのシステム管理

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。

エクセルやWikiでまとめていたところをシステムに組み込むことで、スクリプトでの一括確認が可能になりました。