**フェンリル株式会社　選考課題「ウェブアプリ」　簡易仕様書**

**朴彩恩 パクチェウン**

1. **アプリ名**

TastyMap「テイスティマップ」

1. **対象OSおよびブラウザ（バージョン情報含む）**

－OS：Windows 10 Home

－ブラウザ：Google Chrome 134.0.6998.89（公式ビルド）

1. **開発環境・言語**

－エディタ：Visual Studio Code（VS Code）

－バージョン管理：Git

－Node.js 20.18.3

－npm10.8.2

－言語：JavaScript、HTML、CSS

1. **開発期間**

2025年03月04日～2025年03月18日

1. **機能概要**

－ユーザーの現在地をもとにしたレストラン検索（距離調整可能）

－ジャンル別レストラン検索（複数ジャンル選択可能）

－レストランリスト＆ページング機能

－レストラン詳細情報表示（Google Map API 連携）

－レストランのお気に入り登録機能（登録、解除およびお気に入りリスト表示）

－レストランの定休日と現在日付きの比較による通知機能（定休日の場合、レストラン名の横に「定休日」表示）

1. **フレームワークおよび主要技術スタック（バージョン含む）**

－フロントエンド：React 19.0.0（React Router DOM 含む）

－バックエンド：Express 4.21.2（CORS、Dotenv 含む）

－API通信：Axios 1.8.1

－地図機能：@react-google-maps/api 2.20.6

1. **テーブル定義（ER図）などの設計ドキュメント**

DBの詳細な設計情報については、添付のエクセルファイル「フェンリル課題DB設計ーパクチェウン」と「フェンリル課題DB-ER図ーパクチェウン」をご参照ください。

現在の仕様に基づくレストラン情報およびお気に入りレストラン情報のテーブルに加え、将来的に実装を予定している機能（ユーザー管理、レビュー機能など）を考慮した設計も含まれています。

1. **開発環境構築手順**

( 1 ) Gitのインストール確認（もしGitがインストールされていない場合は、Git公式サイトからダウンロードしてインストールしてください。）

＄git --version

( 2 ) プロジェクトをクローン

＄git clone <https://github.com/chaeeun1002/TastyMap>

＄cd TastyMap

( 3 ) 依存関係をインストール

（ルートディレクトリで実行）

＄npm install

(バックエンドの依存かん系をインストール)

＄cd backend

＄npm install

＄cd ..

（プロントエンドの依存関係をインストール）

＄cd frontend

＄npm install

＄cd ..

( 4 ) プロジェクトの起動（Express + React同時実行）

＄npm start

1. **コンセプト**

近くの美味しいお店を簡単に検索できるグルメ探索サイト

1. **こだわったポイント**

－複数ジャンルを選択できるフィルタリング機能

－お気に入り登録機能で気になるレストラン管理を簡単に処理

－Google Maps APIを使用した詳細情報表示

－定休日表示機能で無駄足を防ぐ

1. **デザイン面でこだわったポイント**

－落ち着いた雰囲気を演出するため、グリーン系のポイントカラーを使用

－ユーザーがより便利に利用できるよう、位置調整用のプログレスバーを実装

－ユーザーのマウス移動に反応するレストランリストのカードアクションを実装

－ページ移動を最小限にするため、レストラン詳細情報をモーダル表示で実装

1. **技術面でアドバイスしてほしいポイント**

－機能拡張のためにDBを追加することになり、現在のバックエンド構造を変更する場合、どのような方法で拡張するのが望ましいかアドバイスを頂きたいです。

－様々な機能のためにライブラリを選定する際、考慮すべきの要素にはどのようなものがあるか、また、実務ではどのような方法で検討を行うのかアドバイスを頂きたいです。

－チームプロジェクトにおける協業および保守性を考慮した際、コメントを効果的に記述する方法や、実務でよく使用されるパターンについてアドバイスを頂きたいです。

1. **自己評価**

( 1 ) プロジェクト構造や環境設定に関する理解不足

新しい言語を学ぶことは、学習を通じて文法や技術を習得することで乗り越えるのができると思っています。しかし、それ以前に開発するプロジェクトの特性を理解し、それにできしたプリジェクト構造や環境設定をどう構築するかについての知識が足りないと実感しました。今後はプロジェクト設計のベストプラクティスについてより深く学んでいきたいです。

( 2 ) クリーンコードの重要性

今回は個人課題として開発を行いましたが、今後実務でチームメンバーと協力しながら開発を進める際には、他の開発者がコードを見たときにすぐに理解できるようなクリーンコードを書くことが求められると感じました。現在は可読性や保守性を考慮したコードの書き方がまだ十分ではないと思いますので、今後はクリーンコードの原則を学び、より理解しやすいコードを書けるように努力したいです。

( 3 ) ユーザー視点でのUI/UX設計の不足

ユーザにとってより使いやすいUI/UXを設計することの重要性を改めて実感しました。しかし、今回の課題では最新のトレンドを把握できておらず、結果として一般的によく使われる「ヘッダーコンテンツ」レイアウトを実装しました。今後は、様々なサイトのUI/UXを直接観察し、それぞれのサイトが持つ独自性や差別化されたレイアウトを参考することで、より良いデザインを考えられるようにしたいと思います。