Hanamiプロジェクトの名刺管理プロジェクトの引継ぎマニュアル。

2025/04/07 – 岩本建厚作。

・ファイル構成。

コンピューターのスクリーンショット

AI によって生成されたコンテンツは間違っている可能性があります。

1. **App Router の API Routes(**src/app/api/contact/route.ts **)**

* src/app/api/contact/route.ts というファイルパスが示すように、App Router の API Routes を使用
* GET と POST の HTTP メソッドをエクスポートすることで、API エンドポイントとして機能：

この API は以下のような用途で使用されます：

* ビジネスカードデータの一覧取得（GET）
* 新しいビジネスカードの登録（POST）
* マスターデータ（区分、組織、担当者、地域）の管理
* エラーハンドリングと適切なレスポンスの返却

**2 .App Router の API Routes（**src/app/api/upload/route.ts**）**

* src/app/api/upload/route.ts というファイルパスが示すように、App Router の API Routes を使用
* POST メソッドをエクスポートすることで、ファイルアップロード用の API エンドポイントとして機能

この API は以下のような用途で使用されます：

* 画像ファイルのアップロード
* アップロードされたファイルの安全な保存
* 一意のファイル名の生成
* エラーハンドリング

・Libファイル

グラフィカル ユーザー インターフェイス, アプリケーション

AI によって生成されたコンテンツは間違っている可能性があります。

このファイル（SupabaseClient.ts）の役割：

1. **Supabase 接続の一元管理**

* アプリケーション全体で同じ Supabase クライアントを使用
* 接続設定を一箇所で管理

1. **環境変数の検証**

* アプリケーション起動時に必要な設定が揃っているか確認
* 設定ミスを早期に発見

1. **型安全性の確保**

* TypeScript を使用して型安全性を確保
* 環境変数の存在を保証

グラフィカル ユーザー インターフェイス, テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

このコード(logout.ts)の主な特徴：

1. **認証状態の管理**

* Supabase の認証状態をクリア
* セッションを終了

1. **ナビゲーション制御**

* ログアウト後の適切なページ遷移
* ユーザーをログインページにリダイレクト

1. **再利用可能な設計**

* カスタムフックとして実装
* アプリケーション全体で簡単に使用可能

グラフィカル ユーザー インターフェイス, アプリケーション

AI によって生成されたコンテンツは間違っている可能性があります。

このコード(utils.ts)の役割。

1. **コードの可読性向上**

* 条件付きクラス名を簡潔に記述可能
* 複雑なクラス結合を整理

1. **パフォーマンスの最適化**

* 重複する Tailwind クラスを自動的に解決
* 不要なクラス名を削除

1. **保守性の向上**

* クラス名の管理を一元化
* スタイルの一貫性を保持

**・lib/prisma.tsファイル**

グラフィカル ユーザー インターフェイス

AI によって生成されたコンテンツは間違っている可能性があります。

このコードは、Prisma を使用するアプリケーションの基盤となる重要な設定ファイルです。データベース接続を効率的に管理し、開発者体験を向上させながら、本番環境でのパフォーマンスとセキュリティを確保します。

Env.ファイル(local)

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

.env.production(本番環境)

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

Home画面のpage.tsx

グラフィカル ユーザー インターフェイス, アプリケーション

AI によって生成されたコンテンツは間違っている可能性があります。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ

AI によって生成されたコンテンツは間違っている可能性があります。

ログイン画面

サインインと登録が出来る。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ

AI によって生成されたコンテンツは間違っている可能性があります。

・routing機能

グラフィカル ユーザー インターフェイス, アプリケーション

AI によって生成されたコンテンツは間違っている可能性があります。

それぞれのnext.jsのapp router機能による。

routing機能。

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

Link hrefで、routing/tantoやkankeiなどにアクセスできる。

メインナビゲーション。

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

登録画面。

グラフィカル ユーザー インターフェイス, アプリケーション

AI によって生成されたコンテンツは間違っている可能性があります。

区分入力、関係機関名、担当者名、エリアは、

SupabaseのSQLで、直接登録が必要。

UIでは、選択式になっている。

グラフィカル ユーザー インターフェイス, アプリケーション

AI によって生成されたコンテンツは間違っている可能性があります。

画像を選択したら、表示出来る。

ダイアグラム

AI によって生成されたコンテンツは間違っている可能性があります。

確認ボタンを押すと、画像が表示される。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

AI によって生成されたコンテンツは間違っている可能性があります。

・Shinkitouroku画面。

コンピューターのスクリーンショット

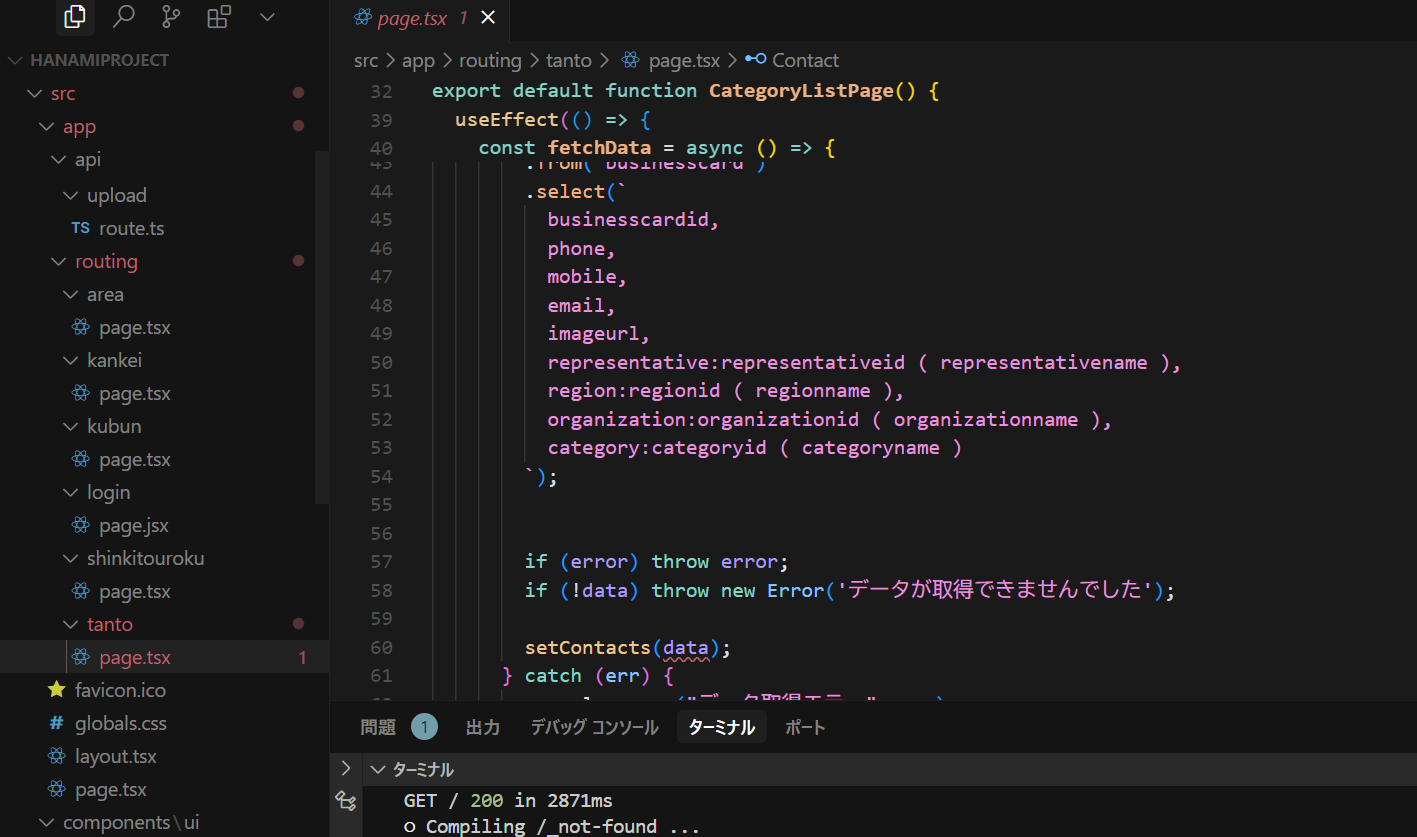
AI によって生成されたコンテンツは間違っている可能性があります。

Shinkitourokuのpage.tsxのhandleSubmitで、

実際の登録処理をしている。

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。



モニター画面に映る文字のスクリーンショット

AI によって生成されたコンテンツは間違っている可能性があります。

categoryIdとorganizationID、representativeID, RegionIDは、

(担当者名 エリア名 関係機関名 区分名)  
実際のSupabaseのテーブルからデータを引っ張って来て、

Selectで、Optionを表示している。

理由は、実際の登録されていないデータを使うと、

テーブルで、競合が起きるため。

(担当者名 エリア名 関係機関名 区分名)

は、supabaseのSQLで直接、管理する事が望ましい。

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

名刺管理テーブルで、名刺を管理している。

Supabaseのinsertで、登録している。

・tanto画面のpage.tsx

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

ReactのUseEffectを使って、

Const dataにbusinesscardのテーブルから、supabaseの、

SQLを使って、selectしたデータをdataに保持している。

パソコン画面のスクリーンショット

AI によって生成されたコンテンツは間違っている可能性があります。

変数publicUrlPrefixで、supabaseに登録すべき、URLを直接、指定している。

パソコン画面のスクリーンショット

AI によって生成されたコンテンツは間違っている可能性があります。

openImagePopupで、名刺画像を表示する機能を作っている。

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

Javascriptのwindow.openを使って画像を表示している。

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

Javascirptのmapを使って、実際取得したデータを表示している。

Kankei/page.tsx

テキスト

AI によって生成されたコンテンツは間違っている可能性があります。

Kankeiのpage.tsxも、

Tanto.tsxと同じ構造。

Supabaseのbusinesscardテーブルから、selectして、データを表示している。

表示機能は、以下の通り。

コンピューターのスクリーンショット

AI によって生成されたコンテンツは間違っている可能性があります。

以上です！