



# 外国語学習ヘルパー(Foreign Language Helper)

## プロジェクト概要

本アプリケーションは、日本語・英語学習を支援するWebアプリケーションです。JLPT（日本語能力試験）やTOEICなどの資格試験対策を中心に、単語学習、読解練習、模擬試験などの機能を提供します。

## 制作の意図・背景

なぜこのアプリを作ったのか：

- 自分自身が日本語・英語を学習する中で、既存の学習アプリでは物足りない点があった
- 検索した単語を自動的に記録し、その単語でクイズを生成する機能が欲しかった
- 資格試験（JLPT、TOEIC）に特化した学習環境を提供したいと考えた
- 画像からテキストを抽出して音声で読み上げる機能により、復習を効率化したいと考えた
- 複数のデバイスで学習データを同期できるようにしたいと考えた - 特に直ぐに単語の意味を調べることではなく、マウスカーソルを上に置くだけで意味を調べることができるならと思って作りました。

The annual company retreat will be held at the Grand Mountain Resort from March 15th to March 17th. All employees are required to attend this three-day event, which includes team-building activities, planning sessions, and networking opportunities. The resort aims to strengthen collaboration among departments and align everyone with the company's goals for the upcoming fiscal year.

Transportation will be provided from the main office at 7:00 AM on March 15th. Employees who prefer to drive their own vehicles should notify the Human Resources department by March 1st. The resort offers various amenities including a fitness center, spa, and multiple dining options. Accommodations will be arranged in double-occupancy rooms, but single rooms can be requested for an additional fee.

During the retreat, participants will engage in workshops covering topics such as effective communication, project management, and customer service excellence. There will also be

ニーズ：

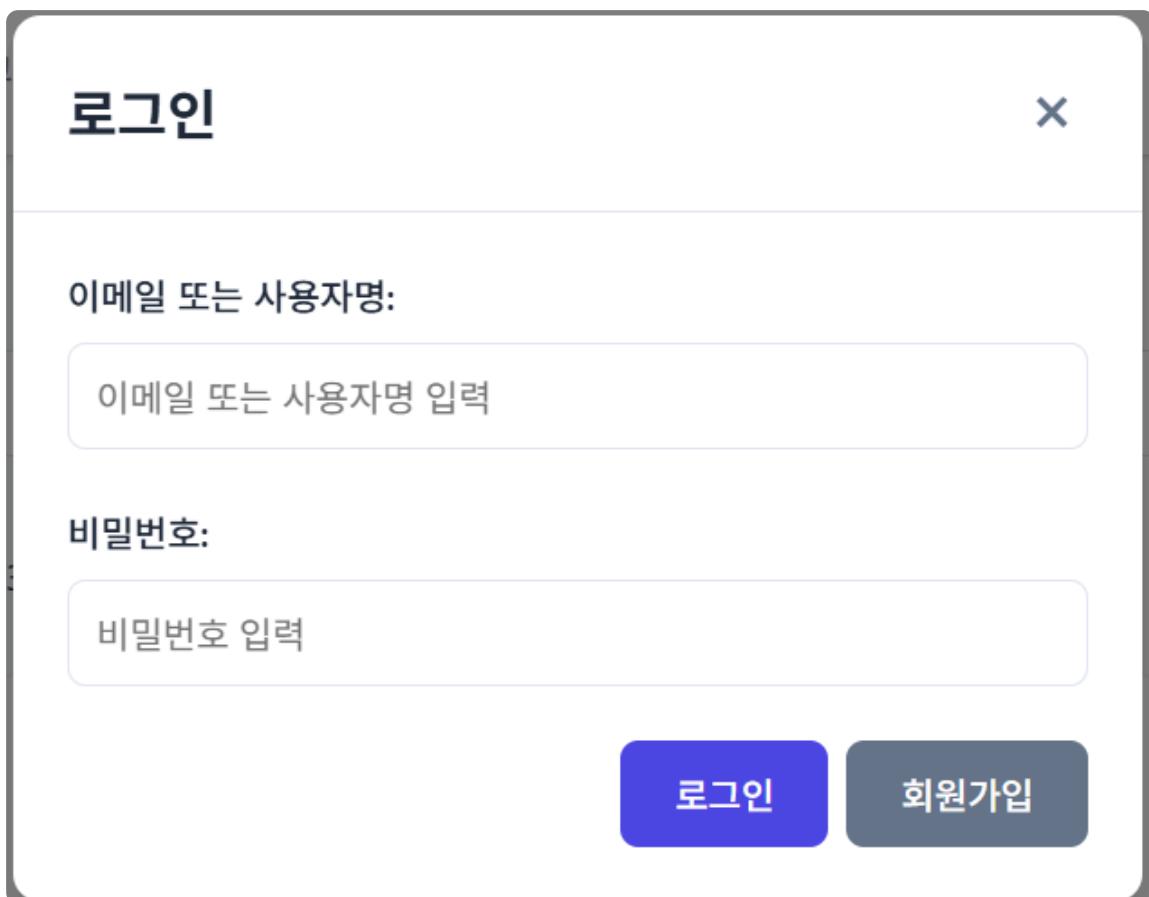
- 資格試験対策に特化した学習アプリの需要
- 学習進捗を可視化し、継続的な学習を支援する仕組み
- オフラインでも利用可能な機能と、オンラインでのデータ同期機能の両立

## ⌚ 主な機能

### 1. ユーザー認証・アカウント管理

- ログイン・サインアップ機能：Supabase Authを使用した認証システム
- プロフィール管理：ユーザー名、メールアドレスの管理

- パスワード変更・アカウント削除：セキュアなアカウント管理機能
- セッション管理：自動ログイン状態の維持



## 2. 単語帳機能

- 単語の追加・削除・編集：ユーザー独自の単語帳作成
- 検索機能：日本語・英語の単語検索
- 習熟度管理：マスター済み/未習得の管理
- 復習回数の記録：学習進捗の追跡

## 3. 辞書機能

- 日本語・英語辞書：ローカル辞書データベースからの検索
- 検索履歴の保存：検索した単語の自動記録
- 漢字の詳細表示：漢字をホバーすると読み方・意味を表示（複合語の場合は個別漢字の意味も表示）
- 単語ホバー機能：読解文内の単語にマウスをホバーすると意味を即座に表示

The screenshot shows a vocabulary learning interface. At the top, there are tabs: '낱' (Nan), '낱장' (Najang) which is highlighted in blue, '독해' (Dokhae), '모의고사' (Moyigoja), and '진행상황' (Jinjeong sanghwang). Below the tabs, the title '낱장' is displayed with a book icon. To the right, it says '목표 자격증: TOEIC Reading'. A search bar contains the placeholder '낱 검색...' and a button labeled '새로고침' (Reset). Below the search bar, it says '총 단어 수: 193' (Total number of words: 193), '학습한 단어: 6' (Learned words: 6), and '학습률: 3%' (Learning rate: 3%). Three words are listed in cards: 'accommodate' (수용하다, 적용시키다), 'accomplish' (성취하다, 완수하다), and 'accurate' (정확한). Each card has two buttons: '상세' (Detailed) and '학습 원료' (Learning material).

## 4. フラッシュカード学習

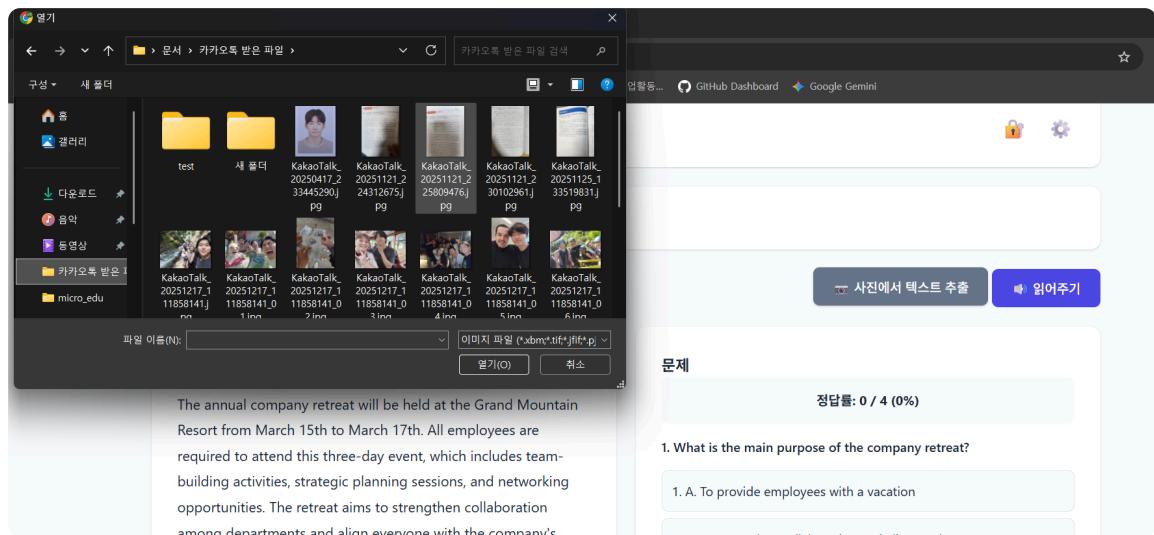
- JLPTレベル別学習：N1～N5レベルに対応した単語学習
- TOEIC単語学習：TOEIC頻出単語の学習
- カードめくり機能：クリックでカードをめくって意味を確認
- 習熟度管理：「知っている」「知らない」ボタンで学習状況を記録
- 前後移動：前後の単語に移動して学習

The screenshot shows a flashcard study interface. At the top, there are tabs: '낱', '낱장' which is highlighted in blue, '독해', '모의고사', and '진행상황'. Below the tabs, the title '낱 학습' is displayed with a book icon. To the right, it says '학습 언어: 일본어 ▾'. In the center, a large white box displays the message '학습할 단어가 없습니다' (There are no words to learn). Below the message, it says '카드를 클릭하여 뒤집기' (Click the card to flip it). At the bottom, there are navigation buttons: '◀ 이전' (Previous), '0 / 0', and '다음 ►' (Next).

## 5. 読解練習

- JLPT読解問題：レベル別の読解問題提供（N1～N5対応）
- 画像からのテキスト抽出（OCR）：Tesseract.jsを使用したOCR機能
- 音声読み上げ（TTS）：Web Speech APIを使用した読み上げ機能

- **音声制御**：読み上げ、一時停止、停止機能
- **テキスト編集機能**：OCRで抽出したテキストを手動で編集可能
- **単語ホバー機能**：読解文内の単語にマウスをホバーすると意味を表示



## 6. クイズ機能

- **検索履歴ベースのクイズ**：検索した単語から自動的にクイズを生成
- **問題数選択**：5問、10問、20問から選択可能
- **4択問題形式**：単語の意味を4択で選択
- **進捗表示**：現在の問題数と進捗バーで表示
- **結果表示**：正答数、正答率を表示

## 7. 模擬試験・レベルテスト

- **模擬試験**：実際の試験形式に準拠した問題
- **レベルテスト**：現在の実力を測定
- **自動採点**：即座に結果を表示
- **進捗表示**：問題の進捗状況を表示

홈

단어장

독해

모의고사

진행상황

## 모의고사 / 레벨테스트

## 모의고사

실제 시험 형식의 모의고사

시작하기

## 레벨테스트

나의 실력을 측정해보세요

시작하기

## 8. 学習進捗管理

- 日次進捗の記録：学習した単語数、クイズスコアの記録
- 連続学習日数のカウント：学習継続のモチベーション向上
- 統計情報の表示：総単語数、習得単語数、クイズ正答率など
- 進捗グラフ：学習進捗の可視化

## 使用技術・ツール

### フロントエンド

- HTML5：セマンティックなマークアップ
- CSS3：レスポンシブデザイン、モダンなUI
- \*\*JavaScript (Vanilla)\*\*：フレームワークを使わない純粋なJavaScript実装
  - ES6+の機能を活用 (async/await、アロー関数、分割代入など)
  - モジュール化されたコード構造

### バックエンド・データベース

- Supabase：BaaS (Backend as a Service)
  - Supabase Auth：ユーザー認証システム
  - PostgreSQL：リレーションナルデータベース
  - \*\*Row Level Security (RLS)\*\*：セキュリティポリシーの実装
  - リアルタイム同期：複数デバイス間でのデータ同期

### 外部API・ライブラリ

- Tesseract.js：クライアントサイドOCR（画像からテキスト抽出）
- Web Speech API：ブラウザネイティブのTTS（Text-to-Speech）機能

- **Supabase JavaScript Client**：データベース操作

## 開発ツール

- **Node.js**：パッケージ管理、データマイグレーションスクリプト
- **npm**：依存関係の管理
- **Git**：バージョン管理

## デプロイ

- **Vercel**：静的サイトホスティング
- **GitHub Pages**：代替デプロイオプション

## 💡 こだわったポイント・工夫した点

### 1. セキュリティの実装

- **Row Level Security (RLS) ポリシー**：ユーザーは自分のデータのみアクセス可能
- **APIキーの適切な管理**：環境変数と `.gitignore` による保護
- **パスワードのハッシュ化**：Supabase Authによる自動処理

### 2. オフライン対応とオンライン同期の両立

- **localStorage フォールバック**：ログインしていない場合もlocalStorageで動作
- **Supabaseとの自動同期**：ログイン時は自動的にクラウドに保存
- **エラーハンドリング**：ネットワークエラー時の適切な処理

### 3. パフォーマンスの最適化

- **メモリ内検索**：辞書データを一度ロードしてメモリ内で検索
- **インデックスの活用**：データベース検索の高速化
- **遅延ローディング**：必要なデータのみをロード

### 4. ユーザー体験 (UX) の向上

- **レスポンシブデザイン**：モバイル・タブレット・デスクトップに対応
- **直感的なUI**：明確なナビゲーション、視覚的なフィードバック
- **リアルタイム更新**：データ変更時の即座なUI更新

### 5. コードの品質

- ・ モジュール化：機能ごとに分離された関数
- ・ エラーハンドリング：try-catch文による適切なエラー処理
- ・ コメント：日本語コメントによる可読性の向上
- ・ 一貫性のあるコーディングスタイル

## データベース設計

### テーブル構成

- ・ **words**：単語マスターテーブル（日本語・英語）
- ・ **profiles**：ユーザープロフィールテーブル
- ・ **user\_vocabulary**：ユーザー単語帳テーブル
- ・ **user\_progress**：学習進捗テーブル
- ・ **search\_history**：検索履歴テーブル

### リレーション

- ・ ユーザーと単語帳：1対多の関係
- ・ ユーザーと進捗：1対多の関係（日次）
- ・ 単語とユーザー単語帳：多対多の関係

## セットアップ・実行方法

### 必要な環境

- ・ Node.js 16以上
- ・ npm または yarn
- ・ モダンブラウザ（Chrome、Firefox、Safari、Edge）

### インストール手順

#### 1. リポジトリのクローン

```
git clone https://github.com/your-username/foreign_lang_helper.git  
cd foreign_lang_helper
```

#### 2. 依存関係のインストール

```
npm install
```

### 3. Supabase設定

- `supabase_config.js.example` をコピーして `supabase_config.js` を作成
- SupabaseプロジェクトのURLとAPIキーを設定
- `supabase_schema.sql` を実行してテーブルを作成

### 4. ローカルサーバーの起動

```
# Windows  
start-server.bat  
  
# または Pythonを使用  
python -m http.server 8000
```

### 5. ブラウザでアクセス

- `http://localhost:8000` にアクセス

## デプロイ方法

Vercelでのデプロイ：

1. Vercelアカウントを作成
2. GitHubリポジトリを接続
3. 環境変数を設定（必要に応じて）
4. デプロイ

詳細は `DEPLOY.md` を参照してください。

## 📁 プロジェクト構成

```
foreign_lang_helper/  
├── index.html      # メインHTMLファイル  
├── app.js          # メインJavaScriptファイル（約3800行）  
├── styles.css       # スタイルシート  
├── supabase_client.js # Supabaseクライアント設定  
├── supabase_schema.sql # データベーススキーマ  
├── package.json     # 依存関係定義  
├── vercel.json      # Vercel設定  
|  
└── jlpt/           # JLPT関連データ  
    └── jlptN1~N5/   # レベル別読解問題
```

```
|   └── vocabulary/ # 単語データ
|
|   ├── toeic/       # TOEIC関連データ
|   |   ├── reading/ # 読解問題
|   |   └── vocabulary/ # 単語データ
|
|   └── docs/        # ドキュメント
|       ├── API_SETUP.md
|       ├── SUPABASE_GUIDE.md
|       └── ...

```

## 🎓 学習したスキル・技術

### 実装を通じて学んだこと

- 認証システムの実装：Supabase Authを使用したログイン・サインアップ機能
- データベース設計：PostgreSQLのスキーマ設計、リレーションの理解
- セキュリティ：Row Level Security (RLS) ポリシーの実装
- 非同期処理：async/awaitを使用した非同期処理の実装
- API連携：外部API（OCR、TTS）の統合
- 状態管理：グローバル状態の管理、データの永続化
- エラーハンドリング：適切なエラー処理とフォールバック機能
- レスポンシブデザイン：モバイルファーストのUI設計

### 現在学習中のスキル

- TypeScriptへの移行
- テストコードの作成（Jest等）
- CI/CDパイプラインの構築
- パフォーマンス最適化の高度なテクニック

## 📝 今後の改善予定

- TypeScriptへの移行
- ユニットテスト・統合テストの追加
- PWA（Progressive Web App）対応
- ダークモード対応
- 多言語対応（UIの多言語化）
- ソーシャル機能（学習記録の共有）
- AIを活用した学習内容の推薦機能

## ポートフォリオとしてのポイント：

- 実務経験はありませんが、このようなWebアプリケーションを開発できるスキルを持っています
- 認証、データベース、セキュリティなど、Webアプリケーションに必要な基本機能を実装しました
- 整ったソースコードと適切なエラーハンドリングを心がけました
- オリジナルの機能（検索履歴からの自動クイズ生成、OCR+TTSによる復習機能、漢字・単語ホバー機能など）を含めています
- 自分の実力に合わせた内容で、コピー&ペーストではなく、理解して実装しました