

日本語アクセント投票サイト 拡張機能実装 指示書 & 設計書

このドキュメントは、既存の日本語アクセント投票サイトを拡張するために、AI コーディング支援ツールに渡す「実装指示書」と、開発者向けの「具体的な設計／仕様書」をまとめたものです。本サイトは Next.js 14 / React で構築されたフロントエンドと、Node.js/Express/Prisma のバックエンドで構成されており ^①、JWT による認証を採用しています ^②。以下の指示・設計に従って、機能追加とバグ修正を実施してください。

I. 実装指示書（claude code 向け）

以下の項目を優先順位に沿って実装してください。

1. 日本地図のトップ票色分け表示
2. 都道府県別に最も多く投票された選択肢（アクセント型または汎用投票の候補）を取得し、色分けした日本地図を表示します。既存の `JapanMap.tsx` コンポーネントと `japan-prefectures.json` を活用してください ^③。
3. バックエンドにトップ票を集計する新 API（`GET /api/words/:id/top-by-prefecture`）を追加します。汎用投票の場合は `GET /api/polls/:id/top-by-prefecture` とします。
4. SNS共有機能の追加
5. 投票完了後に「私はXXXに投票しました！みんなも投票して結果を確認しよう！」という文面と、管理者が設定したハッシュタグを含むリンクを生成し、X（旧Twitter）および可能であれば他のSNSにシェアできるようにします。
6. シェア時に使用するサムネイル画像は、アクセントモードでは投票した単語・アクセントの音高を可視化したグラフ画像とし、汎用モードでは管理者が設定した画像を表示してください。
7. イントネーション別読み上げ機能
8. 各アクセント型に対応した音声データを用意または生成し、語詳細ページのスピーカーアイコンから再生できるようにします。
9. Web Speech API などを使用して、NHK式の音高表記を音声合成パラメータに変換する関数を実装してください。
10. バグ修正
11. 地図の色塗りが正しく行われない問題を、新しい集計APIと色マッピングで修正します。
12. 投票数がリアルタイムにページへ反映されない問題を、投票後に最新統計を再取得してレンダリングすることで解決します。
13. 投票時に発生するエラーを調査し、バリデーションを強化することで対処します。レート制限エラーや重複投票エラーは明確なメッセージを返してください。

14. 投票結果にユーザIDが紐付いているか不明なため、匿名投票かログイン必須かを仕様で決め、必要であれば `userId` を投票データに保存してください。

15. 管理者ダッシュボードの作成

16. `admin` ロールのみアクセス可能な新しい管理ページを作成し、投票の作成・編集・削除機能を提供します。汎用投票とアクセント投票を両方扱い、アクセントモードの切り替えはチェックボックスで行います。

17. 投票作成画面では以下の項目を設定できるようにしてください：

- 投票タイトルと説明。
- 選択肢（最大4件まで）。
- アクセントモード ON/OFF。
- 投票締切日時。
- シェア用ハッシュタグ。
- 投票トップ画面用サムネイル画像と、汎用モードの場合は各選択肢のサムネイル画像。

18. JSON ファイルをアップロードして、複数の投票を一括登録する機能も実装します。

19. 汎用投票システムの追加

20. 現在はアクセント投票のみですが、世間の関心を集めるために「きのこの山 vs たけのこの山」のようなテーマで投票できる汎用投票を実装します。選択肢は4つまで、アクセントモードはチェックボックスでOFFにしてください。

21. ユーザーから投票テーマのリクエストを受け付け、管理者ダッシュボードで集計およびCSV/JSONダウンロードできるようにします。

22. SNSやバズリ要素を意識した改良

23. 投票ページにはSEOを考慮したOGPタグを追加し、SNSシェア時に魅力的なカード画像が表示されるようにします。

24. 締切前のカウントダウンや結果発表時の通知メール（またはプッシュ通知）など、ユーザーの再訪を促す仕組みを検討してください。

上記の実装にあたり、不明点や更なる改善案があれば随時記録して対応してください。

II. 具体的な設計／仕様書

1. データモデル

1. Polls	テーブル（新規）	フィールド名	型	説明
		<code>id</code>	UUID/Int	主キー
		<code>title</code>	String	投票タイトル
		<code>description</code>	Text	投票説明
		<code>isAccentMode</code>	Boolean	アクセント投票か汎用投票かを示すフラグ
		<code>options</code>	JSONB	選択肢の配列（最大4件）
		<code>wordId</code>	Int (nullable)	アクセントモード時に単語IDを保持
		<code>deadline</code>	Timestamp	投票締切日時
		<code>shareHashtags</code>	String	シェア用ハッシュタグ（カンマ区切り）
		<code>thumbnailUrl</code>	String	投票ページ用サムネイル画像URL
		<code>optionThumbnails</code>	JSONB	各選択肢のサムネイル画像URL
		<code>createdAt</code>	Timestamp	作成日時
		<code>createdBy</code>	UUID	作成者（管理者）のユーザーID

2. PollVotes	テーブル（新規）	フィールド名	型	説明
				id UUID/Int 主キー pollId UUID/Int Polls テーブルへの外部キー optionIndex Int 選択肢インデックス（0～3） userId UUID (nullable) ログインユーザーID。匿名投票の場合はnull prefecture String 投票者の都道府県コード votedAt Timestamp 投票日時

3. UserVoteRequest	テーブル（新規）	フィールド名	型	説明
				id UUID/Int 主キー title String 提案タイトル description Text 詳細説明 count Int 同じ提案への票数（重複リクエストの集計） createdAt Timestamp 提案日時

2. API 設計

2.1 アクセント投票関連の拡張

メソッド	エンドポイント	内容
GET	/api/words/:id/top-by-prefecture	都道府県ごとに最多投票アクセント型を返す。色分け表示用
GET	/api/words/:id/stats	現行の統計取得APIを活用。投票後はこのAPIを再呼び出して最新表示を更新 ⁴
GET	/api/speech/:accentType/:wordId	指定単語とアクセント型の音声を返す（音声ファイルを生成・配信）

2.2 汎用投票API

メソッド	エンドポイント	内容
GET	/api/polls	公開中の投票一覧を取得（アクセントモード／汎用モードの区別含む）
POST	/api/polls	新規投票を作成（管理者のみ）。JSON またはフォームデータで受付
GET	/api/polls/:id	投票詳細を取得。投票結果と選択肢情報を含む
POST	/api/polls/:id/votes	該当投票に投票する。userId を含める場合は認証チェック
GET	/api/polls/:id/top-by-prefecture	都道府県ごとの最多投票選択肢を返す（汎用モード用）
GET	/api/user-vote-requests	ユーザー投票提案の一覧を取得（管理者ダッシュボード用）
POST	/api/user-vote-requests	ユーザーが投票テーマを提案する

3. 管理者ダッシュボード

1. アクセス制御

- JWT の `role` クレームをチェックし、`admin` ロールのユーザーのみが `/admin` 以下にアクセスできるようにする ²。

3. ページ構成

- Dashboard トップ**：既存投票の一覧、投票数・参加者数など概要表示。
- 投票作成フォーム**：新規投票を登録できるページ。フォーム入力項目は前述した通り。
- 投票編集ページ**：既存の投票を編集・終了・削除できる。
- 投票リクエスト一覧**：ユーザーからの投票提案を表示し、採用・非採用を決定できる。
- インポート機能**：JSON ファイルをアップロードして一括で投票を生成。形式は `Polls` テーブルのスキーマに合わせる。

9. UI 実装

- Next.js の `/app/admin` 以下に管理者用レイアウトとページを作成し、既存の一般ユーザー向けページとは別にナビゲーションを構築する。
- リッチなフォームには React Hook Form や Zod などのバリデーションライブラリを活用する。

4. フロントエンド改修

1. 地図コンポーネント

- 投票結果APIのレスポンスに基づいて都道府県の色を動的に設定します。色のマッピングは設定ファイルで管理し、変更しやすくします。
- 地図の凡例も自動生成し、ユーザーが各色が何を意味するか理解できるようにします。

4. 共有モーダル

- 投票完了時にモーダルを表示し、X・Facebook・LINE などのリンクボタンを提供。
- シェア用のURLやハッシュタグはバックエンドから受け取るか、ページ側で生成します。

7. 音声再生

- 語詳細ページに複数のスピーカーアイコンを用意し、クリックすると該当する音声を再生。音声ファイルがない場合は TTS で生成。

9. 投票UI

- 汎用投票モードでは選択肢をボタンやカード形式で表示し、アクセント投票とは別スタイルにする。
- 締切日時が近づくとカウントダウンタイマーを表示し、ユーザーの注目を引きつける。

5. バックエンド改修

1. サービス層の追加

- `PollService`：投票作成や集計、期限チェックを行うサービス。
- `SpeechService`：音声合成または音声ファイル管理を担当するサービス。

4. リポジトリ層の拡充

5. `PollRepository` : `Polls` テーブルの CRUD および集計を実装。

6. `PollVoteRepository` : 投票登録と統計のクエリを提供。

7. 既存エンドポイントの修正

8. `POST /api/votes` のレート制限とバリデーション強化 ⁵。

9. `/api/words/:id/stats` はキャッシュを使って高速化し、投票後はキャッシュを更新する。

6. その他の技術的考慮事項

- **セキュリティ** : 投票操作やダッシュボード機能には CSRF トークンや認証ミドルウェアを適用し、不正操作を防止します。
- **キャッシュとパフォーマンス** : 投票集計や都道府県別トップ票は Redis に短時間キャッシュし、アクセス負荷を軽減します。
- **テスト** : ユニットテスト・統合テスト・E2E テストを実装し、既存バグの再発防止を図ります ⁶。
- **運用・デプロイ** : CI/CD パイプラインに新しいテストやビルド手順を追加し、管理者ダッシュボードのデプロイも含め本番環境で適切に動作するようにします。

以上が、今回追加される機能を実装するための指示書と具体的な設計／仕様書です。claude code への指示にはこのドキュメントの「実装指示書」部分を渡し、実装過程では「設計／仕様書」を参考にしてください。

¹ ³ GitHub

https://github.com/Klee319/FreeVote/blob/master/ref/product/design/implementation_design.md

² ⁴ ⁵ GitHub

https://github.com/Klee319/FreeVote/blob/master/ref/product/design/api_specification.md

⁶ GitHub

https://github.com/Klee319/FreeVote/blob/master/backend/IMPLEMENTATION_GUIDE.md