🌸 ゆるVibe Pages - 2時間実装カンバンボード

> *2時間で心に響く詩アプリを完成させる超集中戦略 ⚡*

📋 Backlog（全タスク一覧）

🏗️ インフラ・初期設定

- [ ] **INFRA-01**: Next.js プロジェクト作成
- [ ] **INFRA-02**: Firebase プロジェクト作成・設定
- [ ] **INFRA-03**: 環境変数設定（OpenAI API Key等）
- [ ] **INFRA-04**: Tailwind CSS セットアップ
- [ ] **INFRA-05**: TypeScript型定義作成
- [ ] **INFRA-06**: 基本ディレクトリ構成作成

🔧 API・バックエンド

- [ ] **API-01**: Firebase初期化ファイル作成
- [ ] **API-02**: Firestore操作関数実装
- [ ] **API-03**: Firebase Storage操作関数実装
- [ ] **API-04**: OpenAI GPT-4o API呼び出し関数
- [ ] **API-05**: DALL-E 3 API呼び出し関数
- [ ] **API-06**: `/api/generate` エンドポイント実装

🎨 フロントエンド・UI

- [ ] **UI-01**: トップページレイアウト作成
- [ ] **UI-02**: テーマ入力コンポーネント実装
- [ ] **UI-03**: 詩表示ページレイアウト作成
- [ ] **UI-04**: 固定p5.jsアニメーション実装
- [ ] **UI-05**: SNS共有ボタン実装
- [ ] **UI-06**: 基本ローディングUI実装

🌐 メタデータ・デプロイ

- [ ] **META-01**: 基本OGP設定
- [ ] **DEPLOY-01**: Vercelデプロイ設定
- [ ] **DEPLOY-02**: ビルド・デプロイ実行
- [ ] **DEPLOY-03**: 動作確認

⏰ Sprint Plan（40分ごとの実装計画）

🏃‍♂️ Sprint 1: 爆速セットアップ (0:00 - 0:40)

目標: 開発環境完成 & AI機能基盤作成

In Progress

- [ ] **INFRA-01**: Next.js プロジェクト作成 *(3分)*
- [ ] **INFRA-02**: Firebase プロジェクト作成・設定 *(7分)*
- [ ] **INFRA-03**: 環境変数設定 *(3分)*
- [ ] **INFRA-04**: Tailwind CSS セットアップ *(3分)*
- [ ] **INFRA-05**: TypeScript型定義作成 *(5分)*
- [ ] **INFRA-06**: 基本ディレクトリ構成作成 *(2分)*
- [ ] **API-01**: Firebase初期化ファイル作成 *(7分)*
- [ ] **API-02**: Firestore操作関数実装 *(5分)*

- ☐ **API-03**: Firebase Storage操作関数実装 *(5分)*

Sprint 1 完了条件

- ✅ `npm run dev` でローカル起動成功
- ✅ Firebase接続確認
- ✅ 基本ファイル構成完成

🏃‍♂️ Sprint 2: AI機能実装 (0:40 - 1:20)

目標: OpenAI統合とAPIエンドポイント完成

In Progress

- ☐ **API-04**: OpenAI GPT-4o API呼び出し関数 *(12分)*
- ☐ **API-05**: DALL-E 3 API呼び出し関数 *(12分)*
- ☐ **API-06**: `/api/generate` エンドポイント実装 *(16分)*

Sprint 2 完了条件

- ✅ API経由で詩生成成功
- ✅ DALL-E画像生成成功
- ✅ Firebase保存成功

🏃‍♂️ Sprint 3: UI実装 (1:20 - 2:00)

目標: 美しいページ完成とデプロイ

In Progress

- ☐ **UI-01**: トップページレイアウト作成 *(8分)*
- ☐ **UI-02**: テーマ入力コンポーネント実装 *(7分)*
- ☐ **UI-03**: 詩表示ページレイアウト作成 *(10分)*
- ☐ **UI-04**: 固定p5.jsアニメーション実装 *(8分)*
- ☐ **UI-05**: SNS共有ボタン実装 *(3分)*
- ☐ **UI-06**: 基本ローディングUI実装 *(2分)*

- ☐ **META-01**: 基本OGP設定 *(2分)*

Sprint 3 完了条件

- ✅ 美しいテーマ入力画面完成
- ✅ 詩表示ページ完成
- ✅ アニメーション動作

🏃‍♂️ Final Sprint: デプロイ (2:00+)

目標: 本番公開完了

In Progress

- ☐ **DEPLOY-01**: Vercelデプロイ設定 *(3分)*
- ☐ **DEPLOY-02**: ビルド・デプロイ実行 *(5分)*
- ☐ **DEPLOY-03**: 動作確認・テスト *(7分)*

Final Sprint 完了条件

- ✅ 本番環境デプロイ成功
- ✅ 基本動作確認完了

🎯 超優先度マトリックス（2時間版）

🔴 Critical (絶対必須)

1. **API-04, API-05**: AI生成機能（アプリのコア）
2. **API-06**: 生成エンドポイント
3. **UI-01, UI-02**: テーマ入力機能
4. **UI-03**: 詩表示機能
5. **DEPLOY-02**: デプロイ

🟡 High (重要だが短縮可能)

1. **UI-04**: アニメーション（簡単な固定パターン）
2. **UI-05**: SNS共有ボタン
3. **META-01**: 基本OGP

🟢 Cut (時間不足時は削除)

1. **UI-06**: ローディングUI ▶ 最悪なしでもOK
2. 動的OGP ▶ 静的で妥協
3. エラーハンドリング詳細 ▶ 基本のみ

⚡ 2時間成功の秘訣

🚀 時間短縮テクニック

- **shadcn/ui**: 美しいコンポーネントを即導入
- **Tailwind CDN**: セットアップ時間短縮
- **Firebase Emulator**: スキップしてクラウド直接利用
- **p5.js**: CDNから読み込み、複雑なセットアップ回避
- **テスト**: 手動のみ、自動テスト削除

🗑 削減・簡素化項目

- **TypeScript**: 最小限の型定義のみ
- **エラーハンドリング**: try-catch基本実装のみ
- **バリデーション**: フロントエンドのみ
- **レスポンシブ**: デスクトップファーストで実装
- **SEO**: 基本のOGPのみ

🎯 集中ポイント

- **MVP機能**: 詩生成 + 画像生成 + 表示
- **美しさ**: Tailwindで素早く美しく
- **共有**: X共有機能は必須
- **デプロイ**: Vercel一択

🚨 リスク管理（2時間版）

🔥 超高リスクタスク

- **API-04, API-05**: OpenAI API制限・エラー
  - *対策*: 最初にAPIキー確認、フォールバック用ダミーデータ準備
- **API-06**: 並行処理の複雑性
  - *対策*: 最悪順次処理でも妥協

⚡ 緊急時フォールバック

- **DALL-E失敗**: プレースホルダー画像で妥協
- **p5.js問題**: CSS animationで代替
- **Firebase問題**: ローカルストレージで一時対応
- **デプロイ問題**: 別のホスティングサービス検討

📝 実装メモ（2時間版）

🎨 シンプルデザインガイドライン

- **色調**: ピンク系、Tailwindのrose/pink系パレット

- **レイアウト**: 中央寄せシンプル構成
- **アニメーション**: floating particles（30行以内のシンプル実装）
- **フォント**: システムフォント使用

🔧 技術的簡素化

- **Next.js**: App Router、最小限の機能
- **Firebase**: Firestore + Storage
- **OpenAI**: GPT-4o + DALL-E 3のみ
- **スタイリング**: Tailwind、カスタムCSS最小限

✅ Done

完了したタスクはここに移動

🐾 *2時間で素敵なアプリを完成させるにゃ〜！超集中モードにゃん⚡ ✨*