プロジェクト名：Palindrome Checker

ドキュメント名：要求定義書

作成者：ｘｙｚ

作成日：2025年8月4日

**1. はじめに（Introduction）**

説明：この文書の目的と対象範囲を簡潔に記述。

この文書は、Webアプリ「Palindrome Checker」における基本的な要求事項を定義するものである。本アプリは、ユーザーが入力した文字列が回文であるかをチェックする機能を提供する。

**背景（Background / Context）**

アプリを作る理由、ニーズ。

文字列の正規性や対称性を学ぶ教材の一部として、回文を判定する簡易ツールを作成する。

**3. 用語の定義（Terminology, if needed）**

| **用語** | **定義** |
| --- | --- |
| 回文 | 前から読んでも後ろから読んでも同じ文字列 |
| 入力文字列 | ユーザーが入力する対象のテキスト |
| 正規化 | 空白や大文字小文字を無視すること |

**4. 要求事項（Functional Requirements）**

ここがメインの部分！各機能を「〇〇ができること」と書く。

| **No** | **要求項目** | **詳細** |
| --- | --- | --- |
| 1 | 入力機能 | ユーザーが任意の文字列を入力できるテキストボックスを設置すること |
| 2 | 判定実行ボタン | ボタンをクリックすると、入力された文字列が回文かどうかを判定すること |
| 3 | 結果表示 | 判定結果を画面上に表示すること（回文です／回文ではありません） |
| 4 | エラーハンドリング | 入力が空の場合や数字・記号のみの場合、適切なメッセージを表示すること |
| 5 | モバイル対応（任意） | スマホ画面でも表示が崩れないようにすること（レスポンシブ対応） |

**5. 非機能要件（Non-Functional Requirements）**

品質的な側面。性能、使いやすさ、セキュリティなど。

**例:**

* ブラウザ互換性（Chrome / Edge / Firefox）
* 初心者でも直感的に使えるUI
* レスポンシブデザイン対応（任意）

**6. 前提条件・制約事項（Assumptions / Constraints）**

**例:**

* 開発言語はHTML、CSS、JavaScriptのみ使用
* データ保存やログイン機能は不要
* 日本語／英語入力に対応

**7. 将来の拡張（Optional Future Enhancements）**

**例:**

* 入力履歴の保存機能
* ローカライズ（多言語対応）
* バックエンド連携による履歴記録