

Web アプリケーション仕様書 (F1 2026 年出場チーム , 歴代火影 , W 杯 2022 ベスト 8)

作成者 : _____

2026 年 1 月 15 日

1 概要

本書は , Express と EJS を用いて構築した Web アプリケーションの仕様をまとめたものである . 対象データは , 「F1 2026 年出場チーム」 , 「歴代火影」 , 「W 杯 2022 ベスト 8 」の 3 種類とし , 各データセットに対して CRUD (Create , Read , Update , Delete) を提供する .

本アプリは教育目的の簡易実装であり , データ保存はサーバのメモリ上で行う . そのため , サーバ再起動によりデータは初期状態へ戻る .

2 動作環境

| 項目 | 内容 |
|---------|-------------------------------------|
| OS | Windows / macOS (Node.js が動作する環境) |
| 言語処理系 | Node.js (LTS 推奨) |
| フレームワーク | Express |
| テンプレート | EJS |
| ポート | 8080 |

3 起動手順

プロジェクト直下で次を実行する .

| 手順 | コマンド |
|---------|-------------|
| 依存関係の導入 | npm install |
| 起動 | npm start |

起動後 , ブラウザで `http://localhost:8080/` にアクセスする .

第Ⅰ部 利用者向け仕様

4 利用者の想定

利用者は、データの閲覧を主目的として本アプリを利用する。本仕様の「利用者」は、データの内容を参照し、必要に応じて詳細ページを確認する。

5 画面一覧

| 画面名 | URL |
|------------------|------------------------|
| トップ | / |
| F1 2026 一覧 | /f1_2026 |
| F1 2026 詳細 | /f1_2026/{number} |
| 歴代火影 一覧 | /hokage |
| 歴代火影 詳細 | /hokage/{number} |
| W杯 2022 ベスト 8 一覧 | /wc2022_best8 |
| W杯 2022 ベスト 8 詳細 | /wc2022_best8/{number} |

6 基本操作

6.1 一覧表示

トップ画面から任意のデータセットを選択すると、一覧画面に遷移する。一覧画面では、各行の名称をクリックすることで詳細画面に遷移できる。

6.2 詳細表示

詳細画面では、選択した要素の各項目を表形式で表示する。

第 II 部 管理者向け仕様

7 管理者の想定

管理者は、データの追加・編集・削除を行い、Web 上に表示される内容を更新する。教育用の簡易アプリのため、認証機能は実装しない。実運用では、管理者向け機能を認証で保護することが望ましい。

8 管理機能

8.1 追加 (Create)

各一覧画面の「追加」リンクを選択すると、追加フォーム（静的 HTML）へ遷移する。送信後、対象一覧へ戻り、追加内容が反映される。

8.2 編集 (Update)

各一覧画面の「編集」リンクを選択すると、編集画面へ遷移する。編集画面から更新を送信すると、対象一覧へ戻り、変更が反映される。

8.3 削除 (Delete)

各一覧画面および詳細画面の「削除」リンクを選択すると、対象の行が削除され、対象一覧へ戻る。削除時はブラウザの確認ダイアログを表示する。

9 入力項目と制約

9.1 F1 2026 年出場チーム

| 項目 | 必須 | 制約 |
|---------------|----|---------------------|
| チーム名 (name) | 必須 | 80 文字以内 |
| メモ (note) | 任意 | 制限なし (画面表示上は短文推奨) |

9.2 歴代火影

| 項目 | 必須 | 制約 |
|-----------------|----|------------|
| 名称 (name) | 必須 | 80 文字以内 |
| Roman (roman) | 任意 | 画面表示上は短文推奨 |
| メモ (note) | 任意 | 画面表示上は短文推奨 |

9.3 W 杯 2022 ベスト 8

| 項目 | 必須 | 制約 |
|---------------|----|---------|
| 国・地域 (name) | 必須 | 80 文字以内 |

10 データの永続化について

本アプリはメモリ上の配列で管理するため、サーバ再起動によりデータは初期化される。管理者が更新内容を残す必要がある場合、JSON API を利用して手動でバックアップを取る、あるいは DB を導入する拡張を行う。

第 III 部 開発者向け仕様

11 システム構成

11.1 ディレクトリ構成

```
webpro_f1_hokage_wc2022/
  app.js
  package.json
  views/
    index.ejs
    f1_2026.ejs
    f1_2026_detail.ejs
    f1_2026_edit.ejs
    hokage.ejs
    hokage_detail.ejs
    hokage_edit.ejs
    wc2022_best8.ejs
    wc2022_best8_detail.ejs
```

```

wc2022_best8_edit.ejs
public/
  style.css
  f1_2026_new.html
  hokage_new.html
  wc2022_best8_new.html
README.md
specification.tex

```

11.2 サーバ側の設計

app.js は、以下の役割を持つ。

| 役割 | 内容 |
|-----------|-----------------------------------|
| ルーティング | 各 URL に対して、一覧・詳細・編集などの処理を割り当てる |
| テンプレート描画 | res.render() により EJS を描画する |
| フォーム受信 | express.urlencoded() により POST を扱う |
| 簡易バリデーション | 必須入力と文字数の最小チェックを行う |

12 ルーティング仕様

12.1 F1 2026 年出場チーム

| HTTP | URL | 処理 |
|------|--------------------------|---------------|
| GET | /f1_2026 | 一覧表示 |
| GET | /f1_2026/create | 追加フォームヘリダイレクト |
| POST | /f1_2026 | 追加登録 |
| GET | /f1_2026/{number} | 詳細表示 |
| GET | /f1_2026/edit/{number} | 編集画面 |
| POST | /f1_2026/update/{number} | 更新 |
| GET | /f1_2026/delete/{number} | 削除 |

12.2 歴代火影

| HTTP | URL | 処理 |
|------|----------------|---------------|
| GET | /hokage | 一覧表示 |
| GET | /hokage/create | 追加フォームヘリダイレクト |
| POST | /hokage | 追加登録 |

| | | |
|------|-------------------------|------|
| GET | /hokage/{number} | 詳細表示 |
| GET | /hokage/edit/{number} | 編集画面 |
| POST | /hokage/update/{number} | 更新 |
| GET | /hokage/delete/{number} | 削除 |

12.3 W杯 2022 ベスト 8

| HTTP | URL | 処理 |
|------|-------------------------------|---------------|
| GET | /wc2022_best8 | 一覧表示 |
| GET | /wc2022_best8/create | 追加フォームヘリダイレクト |
| POST | /wc2022_best8 | 追加登録 |
| GET | /wc2022_best8/{number} | 詳細表示 |
| GET | /wc2022_best8/edit/{number} | 編集画面 |
| POST | /wc2022_best8/update/{number} | 更新 |
| GET | /wc2022_best8/delete/{number} | 削除 |

13 データ設計

本アプリでは、配列（JavaScript のオブジェクト配列）としてデータを管理する。`.number` は配列の添字として利用し、`id` は表示用の識別子として保持する。`id` は追加時に最大値へ 1 を加えることで生成する。

14 エラー処理

存在しない番号（添字）へのアクセス、および必須項目未入力はエラーとして扱い、HTTP ステータスコードを返す。本実装では簡易化のため、エラー画面のテンプレートは用意せず、`res.status(400/404).send(...)` を用いる。

15 拡張案

実運用を想定する場合、以下の拡張が有効である。

| 拡張案 | 内容 |
|-----------|-------------------------|
| 認証・認可 | 管理機能をログインで保護する |
| DB 導入 | SQLite / MySQL などに永続化する |
| 入力検証の強化 | 型チェック、範囲チェック、重複チェック |
| テンプレート共通化 | EJS の共通レイアウト化により重複を削減する |