

Web プログラミング 最終課題レポート

251134 横居 克憲

2025 年 12 月 26 日

ソースコード (GitHub リポジトリ URL):

https://github.com/shibakatsu-oss/webpro_06

1 利用者向け仕様書

1.1 システム概要

本システムは、うまい棒の情報を管理・閲覧するための Web アプリケーションである。利用者は、うまい棒の味、価格、カロリーなどを一覧で確認できるほか、詳細情報の閲覧や新規登録が可能である。

1.2 操作方法

1. **一覧画面:** トップページにアクセスすると、現在登録されているうまい棒の一覧が表示される。
2. **詳細表示:** 一覧の「味」のリンクをクリックすると、そのうまい棒の詳細情報（価格・カロリー）が表示される。
3. **新規登録:** 一覧画面下部の「新規追加」ボタンを押すと、登録画面に遷移する。必要な情報を入力して「登録」を押すとデータが保存される。
4. **削除:** 一覧画面または詳細画面にある「削除」リンクをクリックすると、該当データが削除される。

2 管理者向け仕様書

2.1 システムのセットアップ

本システムは Node.js 上で動作する。以下の手順でサーバーを起動する。

```
$ npm install express ejs
$ node app5.js
```

2.2 動作環境

- Node.js
- Web ブラウザ

3 開発者向け仕様書

3.1 システム 1: うまい棒管理システム

3.1.1 データ構造

データは配列 `umaiboData` 内のオブジェクトとして管理する。各オブジェクトは以下のプロパティを持つ。

プロパティ名	型	説明
id	数値	データ固有の ID
flavor	文字列	うまい棒の味 (例: メンタイ)
price	数値	価格 (円)
calorie	数値	カロリー (kcal)

3.1.2 画面遷移とリソース

HTTP メソッドとリソース名 (URL) の対応は以下の通りである。

- **一覧表示** (GET `/umaibo`): 全データを一覧表示する。
- **詳細表示** (GET `/umaibo/:number`): 指定 ID の詳細を表示する。
- **新規登録フォーム** (GET `/umaibo/create`): 登録画面へ遷移する。
- **新規登録処理** (POST `/umaibo`): データを追加し一覧へ戻る。
- **編集フォーム** (GET `/umaibo/edit/:number`): 編集画面を表示する。
- **更新処理** (POST `/umaibo/update/:number`): データを更新する。

- **削除処理** (GET /umaibo/delete/:number): データを削除する.

3.2 システム 2: スマブラキャラクター一覧システム

3.2.1 データ構造

データは配列 `smashData` 内のオブジェクトとして管理する.

プロパティ名	型	説明
id	数値	データ固有の ID
name	文字列	ファイター名 (例: マリオ)
series	文字列	出典作品 (例: スーパーマリオ)
number	文字列	参戦ナンバー (例: 01)

3.2.2 画面遷移とリソース

- **一覧表示** (GET /smash): 全キャラを一覧表示する.
- **詳細表示** (GET /smash/:index): 指定キャラの詳細を表示する.
- **新規登録フォーム** (GET /smash/create): 登録画面へ遷移する.
- **新規登録処理** (POST /smash): データを追加する.
- **編集フォーム** (GET /smash/edit/:index): 編集画面を表示する.
- **更新処理** (POST /smash/update/:index): データを更新する.
- **削除処理** (GET /smash/delete/:index): データを削除する.

3.3 システム 3: スプラトゥーン武器一覧システム

3.3.1 データ構造

データは配列 `splaData` 内のオブジェクトとして管理する.

プロパティ名	型	説明
id	数値	データ固有の ID
name	文字列	ブキ名 (例: わかばシューター)
sub	文字列	サブウェポン (例: スプラッシュボム)
special	文字列	スペシャルウェポン (例: グレートバリア)

3.3.2 画面遷移とリソース

- **一覧表示** (GET /splatoon): 全ブキを一覧表示する.
- **詳細表示** (GET /splatoon/:index): 指定ブキの詳細を表示する.

- **新規登録フォーム** (GET /splatoon/create): 登録画面へ遷移する.
- **新規登録処理** (POST /splatoon): データを追加する.
- **編集フォーム** (GET /splatoon/edit/:index): 編集画面を表示する.
- **更新処理** (POST /splatoon/update/:index): データを更新する.
- **削除処理** (GET /splatoon/delete/:index): データを削除する.