

ランキングサイト仕様書

25G1029 大橋慶眞

2025 年 12 月 21 日

GitHub リポジトリ URL

https://github.com/username/webpro_06

目次

1	概要	2
1.1	開発対象システム	2
2	第 1 部：利用者向け仕様	2
2.1	システム概要	2
2.2	利用方法	2
2.3	利用上の注意事項	3
3	第 2 部：管理者向け仕様	3
3.1	システム管理	3
3.2	運用監視	4
4	第 3 部：開発者向け仕様	4
4.1	技術仕様	4
4.2	データ構造	4
4.3	API 仕様	5
4.4	拡張仕様	6
4.5	セキュリティ考慮事項	6

1 概要

本仕様書は、一覧表示をベースとした 3 つの Web アプリケーションの開発仕様を定義するものである。各アプリケーションは独立したランキング情報管理システムとして動作し、Express.js フレームワークを使用して Web サービスとして提供される。

1.1 開発対象システム

1. FANZA ランキングサイト - デジタルコンテンツの売れ行きランキング管理
2. Amazon ランキングサイト - EC 商品の売れ行きランキング管理
3. 観光地ランキングサイト - 日本の人気観光地ランキング管理

2 第 1 部：利用者向け仕様

2.1 システム概要

3 つの独立したランキングサイトを提供し、各分野の人気商品・観光地の情報を閲覧・登録できる Web サービスである。

2.2 利用方法

2.2.1 アクセス方法

各サイトは以下の URL でアクセス可能である：

- FANZA ランキング：http://localhost:8081
- Amazon ランキング：http://localhost:8082
- 観光地ランキング：http://localhost:8083

2.2.2 基本機能

■**ランキング一覧表示** 各サイトのトップページでランキング一覧を表形式で確認できる。表示される情報は以下の通りである：

表 1 各サイトの表示項目

サイト	表示項目
FANZA	ランク、作品名
Amazon	ランク、商品名
観光地	ランク、観光地名

■**詳細情報表示** 一覧のタイトル・商品名・観光地名をクリックすると、詳細情報ページが表示される。詳細ページでは以下の情報を確認できる：

- FANZA：ランク、タイトル、ジャンル、価格、発売日、女優名、商品リンク
- Amazon：ランク、商品名、カテゴリ、価格、評価
- 観光地：ランク、観光地名、都道府県、カテゴリ、年間訪問者数

■**新規データ登録** 「新規登録」リンクから新しいランキングデータを追加できる。必要な情報を入力フォームに記入し、「登録」ボタンを押すことで登録できる。

2.3 利用上の注意事項

- 各サイトは独立して動作するため、データは共有されない
- 登録したデータはサーバー再起動時に初期化される
- 数値項目には適切な値を入力する必要がある

3 第2部：管理者向け仕様

3.1 システム管理

3.1.1 サーバー起動・停止

各サイトは独立した Node.js プロセスとして動作する。

■起動方法

```

1 # ランキングサイトFANZA
2 node fanza_ranking.js
3
4 # ランキングサイトAmazon
5 node amazon_ranking.js
6
7 # 観光地ランキングサイト
8 node tourism_ranking.js

```

■**停止方法** 各プロセスで Ctrl+C を押下してサーバーを停止する。

3.1.2 ポート管理

各サイトは以下のポートで動作する：

表2 ポート割り当て

サイト	ポート番号
FANZA ランキング	8081
Amazon ランキング	8082
観光地ランキング	8083

3.1.3 データ管理

現在の実装では、データはメモリ内の配列として管理されている。サーバー再起動時にはサンプルデータに戻る。

■初期データ 各サイトには5件のサンプルデータが設定されている。

■データバックアップ 本格運用時には、データベースへの移行を推奨する。

3.2 運用監視

3.2.1 ログ監視

各サーバーの起動時にコンソールに起動メッセージが表示される。エラー発生時はコンソールログを確認する必要がある。

3.2.2 アクセス監視

Express.js の標準ログ機能により、アクセス状況を監視できる。

4 第3部：開発者向け仕様

4.1 技術仕様

4.1.1 開発環境

- Node.js (推奨バージョン: 14.x 以上)
- Express.js 4.x
- EJS テンプレートエンジン

4.1.2 ディレクトリ構成

```
1 kadai/ |——
2  fanza_ranking.js      # サイトメインFANZA |——
3  amazon_ranking.js    # サイトメインAmazon |——
4  tourism_ranking.js   # 観光地サイトメイン |——
5  package.json         # 依存関係定義 |——
6  views/               # テンプレートファイル | |——
7    fanza_list.ejs | |——
8    fanza_detail.ejs | |——
9    fanza_new.html | |——
10   amazon_list.ejs | |——
11   amazon_detail.ejs | |——
12   amazon_new.html | |——
13   tourism_list.ejs | |——
14   tourism_detail.ejs | |——
15   tourism_new.html |——
16  public/              # 静的ファイル (未使用)
```

4.2 データ構造

4.2.1 FANZA ランキングデータ

```
1 {
2   id: Number,           // 一意識別子
3   rank: Number,         // ランキング順位
4   title: String,        // 作品タイトル
5   genre: String,        // ジャンル
6   price: Number,        // 価格 (円)
7   releaseDate: String,  // 発売日 (YYYY-MM-) DD
8   actress: String,      // 女優名
9   link: String          // 商品リンク () URL
10 }
```

4.2.2 Amazon ランキングデータ

```
1 {
2   id: Number,           // 一意識別子
3   rank: Number,         // ランキング順位
4   productName: String, // 商品名
5   category: String,     // カテゴリ
6   price: Number,        // 価格 (円)
7   rating: Number        // 評価 () 1-5
8 }
```

4.2.3 観光地ランキングデータ

```
1 {
2   id: Number,           // 一意識別子
3   rank: Number,         // ランキング順位
4   spotName: String,     // 観光地名
5   prefecture: String,   // 都道府県
6   category: String,     // カテゴリ
7   visitorsCount: Number // 年間訪問者数 (万人)
8 }
```

4.3 API 仕様

4.3.1 共通エンドポイント

各サイトで共通の URL 構造を使用する：

4.3.2 レスpons形式

■HTML レスpons 全てのエンドポイントは HTML ページを返す。EJS テンプレートエンジンを使用してサーバーサイドレンダリングを行う。

表 3 API エンドポイント

メソッド	URL	機能
GET	/	一覧表示
GET	/create	新規登録フォーム表示
POST	/create	新規データ登録処理
GET	/:id	詳細情報表示

■リダイレクト POST 処理後は一覧ページ (/) にリダイレクトする。

4.4 拡張仕様

4.4.1 データベース連携

将来的な拡張として、以下のデータベース連携を想定する：

- MongoDB - NoSQL データベース
- MySQL - リレーショナルデータベース
- SQLite - 軽量データベース

4.4.2 追加機能

- データ編集機能
- データ削除機能
- 検索・フィルタ機能
- ページネーション
- ユーザー認証

4.5 セキュリティ考慮事項

- 入力値検証の実装
- SQL インジェクション対策
- XSS 攻撃対策
- CSRF 攻撃対策