

Webプログラミング レポート

千葉工業大学
情報変革科学部 情報工学科

学生番号 25G1035

氏 名 金子 稜侑

以下に、本システムで作成したソースコードを保管している
GithubのURLを添付する.

< https://github.com/frogama55/webpro_report >

1. 利用者向け仕様書

1 はじめに

本システムは、「千葉工業大学の各学科の偏差値」、「千葉工業大学情報変革科学部情報工学科1年次の授業一覧」、「千葉工業大学情報変革科学部情報工学科の研究室」を一覧表示する Web システムである。本仕様書では、代表として、「千葉工業大学情報変革科学部情報工学科の研究室」の利用方法について説明する。なお、このシステムはパソコンでの利用を想定しているため、本仕様書ではパソコン上での動作等について説明する。

2 システムへのアクセス方法

まず、Web ブラウザを立ち上げる。ここでは、図 1.1 がアイコンである Google chrome を利用する。



図 1.1: Google chrome のアイコン。このアイコンをクリックして起動する。

次に、図 1.2 のようなアドレスバーに `<http://localhost:8080>` を入力し、アクセス（パソコンの Enter キーを押下）する。

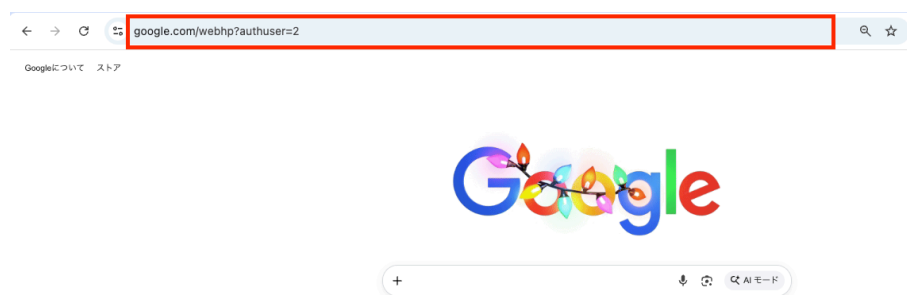


図 1.2: ブラウザの画面。赤枠部分がアドレスバー。

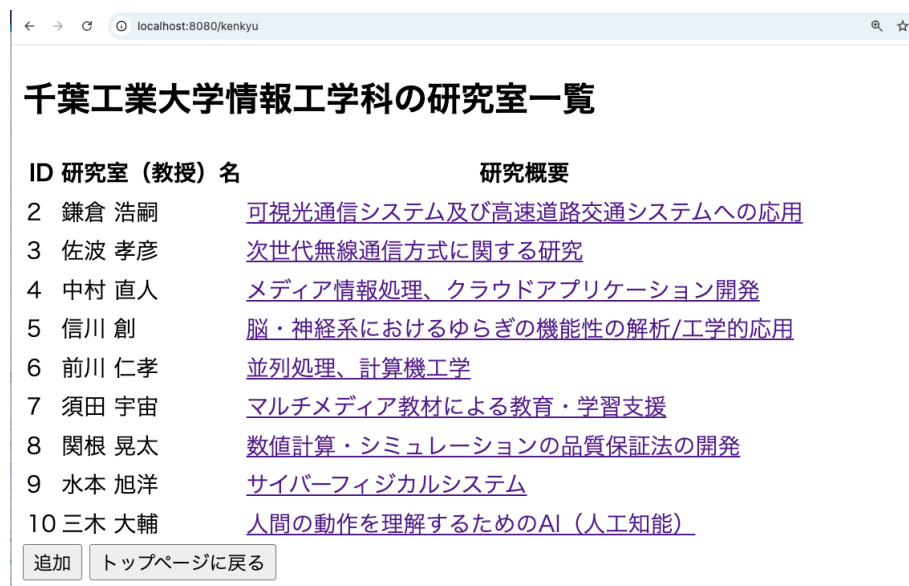
各システムが並ぶトップページが表示されるので、「千葉工業大学情報工学科の研究室一覧」をクリックする。これにより、システムへアクセスできた。

3 システムの機能と操作方法

3.1 研究室情報の一覧

3.1.1 研究室情報の一覧の閲覧

システムへアクセスすると、現在登録されている研究室の一覧が表示される。各行には、データに固有の ID、研究室名、研究概要が表示されている。「トップページへ戻る」をクリックすると、各システムの一覧画面に遷移する。研究概要をクリックすると、各研究室の詳細画面にアクセスすることができる。図 1.3 に研究室の一覧画面、図 1.4 に研究室情報の詳細画面をそれぞれ示す。



ID 研究室 (教授) 名	研究概要
2 鎌倉 浩嗣	可視光通信システム及び高速道路交通システムへの応用
3 佐波 孝彦	次世代無線通信方式に関する研究
4 中村 直人	メディア情報処理、クラウドアプリケーション開発
5 信川 創	脳・神経系におけるゆらぎの機能性の解析/工学的応用
6 前川 仁孝	並列処理、計算機工学
7 須田 宇宙	マルチメディア教材による教育・学習支援
8 関根 晃太	数値計算・シミュレーションの品質保証法の開発
9 水本 旭洋	サイバーフィジカルシステム
10 三木 大輔	人間の動作を理解するためのAI (人工知能)

[追加](#) [トップページに戻る](#)

図 1.3: 研究室一覧画面。

項目	データ
研究室 (教授) 鎌倉 浩嗣 名	
研究概要	可視光通信システム及び高速道路交通システムへの応用
研究内容	高速で点滅させることができるLED光源を用いた可視光通信に関する研究に取り組んでいます。受信機をイメージセンサ（高速カメラ）とする可視光システムの高度道路交通システム（ITS）への応用を目指しています。また、ITS応用については、交通の輸送効率や快適性の向上に寄与する効率のよい車間通信・車路間通信を実現するプロトコルの研究にも取り組んでいます。
キーワード	IT（情報技術）・ITS（インテリジェント交通システム）・移動体通信・信号処理
研究室 サイト URL	
編集	削除
研究室一覧に戻る	

図 1.4: 研究室の詳細画面。ここでは ID2 のページを例示。

研究室の独自サイトがある場合に限り、図 1.4 のように「研究室サイト URL」の部分が青色になり、アクセス可能になる。クリックすると、新しいタブが開き、研究室の独自サイトに遷移する。また、図 1.4 の下部にある「研究室一覧に戻る」をクリックすると、図 1.3 に示した、研究室の一覧画面に戻ることもできる。

3.1.2 研究室情報の編集と削除

図 1.4 の下部にある「編集」「削除」ボタンについて説明する。編集ボタンは、そのページに表示されている研究室データを好きなように編集することができる。クリックすると、図 1.5 のような編集画面に遷移する。

項目	データ
研究室(教授)名	鎌倉 浩嗣
研究概要	可視光通信システム及び
研究内容	高速で点滅させることか
キーワード	IT (情報技術)・ITS (イ
研究室サイトURL	https://www.kama.cs.it-c

送信

研究室一覧に戻る

図 1.5: 研究室情報の編集画面。ここでは ID2 のページを例示。

各項目のデータを書き換えることができる。書き換えたあと、下部にある「送信」ボタンをクリックすることで編集が完了する。また、「研究室一覧に戻る」をクリックすると、編集内容を保存せずに、図 1.3 に示した、研究室の一覧画面に戻ることができる。

3.2 研究室情報の追加

図 1.3 の下部にある「追加」ボタンについて説明する。このボタンをクリックすると、図 1.6 のような追加画面に遷移する。各枠の右側にある項目の通りに、枠にデータを入力し、下部の「送信」ボタンをクリックすると、研究室一覧画面の一番下に、新たな ID とともにデータを追加することができる。



← → 🔍 localhost:8080/public/kenkyu.html

<=研究室（教授）名

<=研究概要

<=研究内容

<=キーワード

<=研究室サイトURL

図 1.6: 研究室情報の追加画面.

4 その他

このシステムの注意点として、システムの仕様上、ここで追加や編集、削除した内容は、サーバーを停止や再起動すると、初期化された状態に戻る。

2. 管理者向け仕様書

1 はじめに

本システムは、Node.js 上で動作する Web アプリケーションサーバであり、「千葉工業大学の各学科の偏差値 (/hensachi)」、「千葉工業大学情報変革科学部情報工学科 1 年次の授業一覧 (/jugyou)」、「千葉工業大学情報変革科学部情報工学科の研究室 (kenkyu)」を一覧表示する 3 つのシステムを 1 つのサーバプロセスで稼働するものである。本システムは、HTTP リクエストを受け付け、EJS テンプレートエンジンを用いて動的に HTML を用意し、クライアント（利用者）へ返す。

2 システムの動作環境

本システムは、パソコンでの利用を想定しているため、本仕様書ではパソコン上での動作等について説明する。開発環境が macOS であるため、macOS での説明を記述している。本システムを稼働させるためには表 2.1 に示す環境が必要である。

表 2.1: 動作環境要件一覧

各環境	必要な動作環境
実行するための環境	Node.js
パッケージ管理ツール	npm (Node.js 内に同梱されている)
OS	Node.js が動作する OS (Windows, macOS, Linux 等)
ネットワーク	TCP ポート 8080 番が使用可能であること

3 システムの導入手順

3.1 ソースコードの入手

Github リポジトリからソースコードをサーバー上の任意のディレクトリにクローン配置する。以下のコードをターミナルを用いて任意のディレクトリで実行することで可能である。

```
$git clone https://github.com/frogama55/webpro_report.git
$cd webpro_report
```

3.2 使用パッケージのインストール

本システムは express パッケージと ejs パッケージを使用している。そのため、package.json が存在するディレクトリにて、ターミナルを用いて以下のコードを実行し、インストールする。

```
$npm install
```

これにより、Node_modules というディレクトリが自動で作成され、パッケージが格納される。

4 システムの起動と停止

4.1 起動方法

IDapp.js が存在するディレクトリにて、以下のコードを実行することで、サーバを起動することができる。

```
$npm start
```

このとき、以下のメッセージが表示されれば、起動に成功している。

```
Example app listening on port 8080!
```

4.2 動作確認

Web ブラウザを起動し、以下の URL をアクセスする。使用する Web ブラウザは、Safari ではなく、Google chrome を推奨する。なぜなら、Safari は Apple 独自のエンジンで動いており、不具合が生じる可能性があるからである。

```
http://localhost:8080/
```

このときに、トップページが表示され、各システムへのリンクが機能していれば、正常に稼働している。

4.3 停止方法

4.1 にてサーバーを起動したターミナル上で、Control キーと C キーを同時に入力することで、サーバプロセスを停止することができる。

5 運用・管理上の注意点

5.1 データの保存について

本システムは簡易データベースとして、プログラム内の変数にデータを保存する仕組みをとっている。このため、サーバを停止したり再起動したりすると、サーバ運用中に追加・編集・削除されたデータは全てリセットされ、初期状態に戻る。

5.2 ポート番号の競合について

本システムは、ポート 8080 番を使用する。起動時に EADDRINUSE 等のエラーが出る場合は、既に他のアプリケーションが 8080 番ポートを使用している可能性がある。その場合、競合しているアプリケーションを停止するか、IDapp.js の最終行に記載されているポート番号の変更が必要となる。

5.3 ログの確認

システムの稼働中、データの追加や編集、削除が行われると、ターミナル上にログが出力される。システムの動作状況を確認する際には、サーバを起動したターミナルの出力を参照することで確認が可能である。

3. 千葉工業大学の各学科の偏差値一覧 システムの開発者向け仕様書

1 データ構造

1.1 変数定義

本システムでは，データベースソフトウェアを使用せず，アプリケーションサーバ（IDapp.js）内の配列変数 `hensachi` にデータを格納する方法を採用している．`hensachi` はオブジェクトの配列変数であり，プロセス終了時にデータは消失する．

1.2 オブジェクトの構造

`hensachi` 配列の各要素は，以下のプロパティを持つオブジェクトである．変数名と実データの対応関係は表 3.1 の通りである [1]．

表 3.1: 変数名と実データの対応関係

変数名	データ型	本システムでの役割	データ例
<code>id</code>	Number	データの管理 ID	1
<code>code</code>	String	学部名	”工学部”
<code>name</code>	String	学科名	”機械工学科”
<code>change</code>	String	入試方式	”一般入試 A 日程”
<code>passengers</code>	Number/String	偏差値	45.0
<code>distance</code>	Number/String	備考	” ”

2 HTTP メソッドとリソース名

表 3.2 に、リソース名と HTTP メソッドを定義する。

表 3.2: 変数名と実データの対応関係

機能	HTTP メソッド	リソース名 (URL)	処理内容
一覧表示	GET	/hensachi	データ一覧画面 (hensachi.ejs) を表示する。
詳細表示	GET	/hensachi/:number	指定 ID の詳細画面 (hensachi_detail.ejs) を表示する。
新規登録 (画面)	GET	/hensachi/create	新規登録フォーム (hensachi.html) へリダイレクトする。
新規登録 (処理)	POST	/hensachi	フォームからデータを受け取り、配列に追加する。
編集 (画面)	GET	/hensachi/edit/:number	編集フォーム (hensachi_edit.ejs) を表示する。
編集 (処理)	POST	/hensachi/update/:number	既存データを更新し、一覧表示画面へリダイレクトする。
削除 (画面)	GET	/hensachi/delete/confirm/:number	削除確認画面 (hensachi_delete.ejs) を表示する。
削除 (処理)	POST	/hensachi/delete/:number	データを null 化し、一覧表示画面へリダイレクトする。

3 ページ遷移

ページ遷移の様子を、図 3.1 に示す。

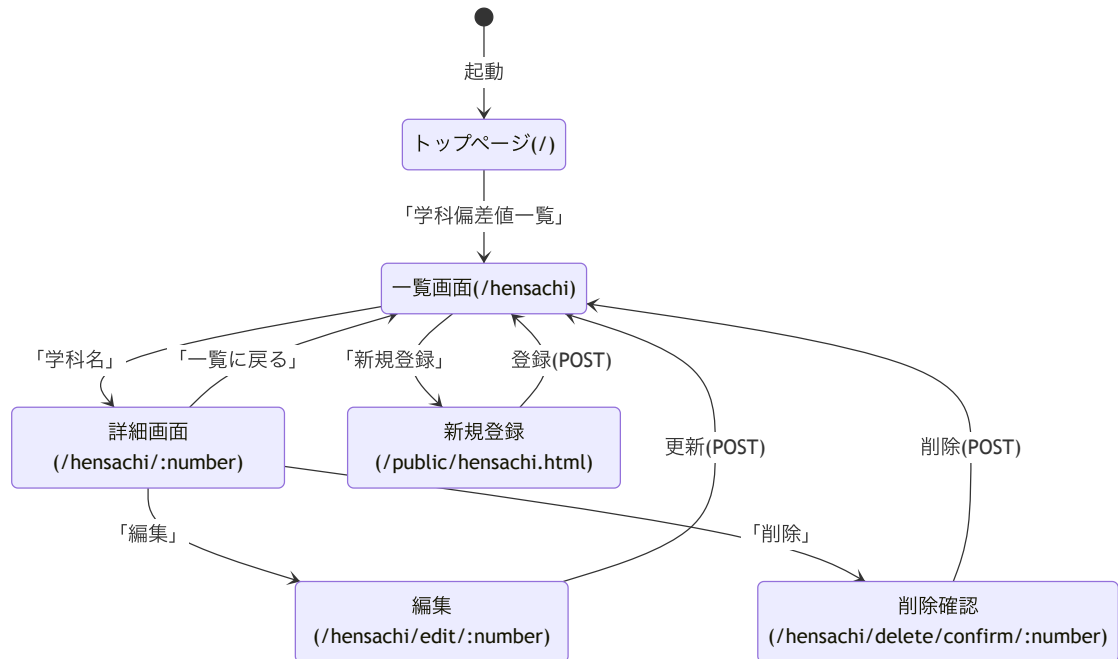


図 3.1: ページ遷移図

4 機能詳細と実装上の注意点

4.1 新規データ追加機能 (Create)

この機能では、`hensachi` 配列の末尾に新しいオブジェクトを追加 (`push`) し、「`hensachi.length + 1`」(一番最後の ID を 1 大きくしたもの) を新しい ID として付与する。追加完了後、`hensachi.ejs` を用いて登録済みデータ一覧ページを表示する。

4.2 一覧表示機能 (Read)

この機能では、EJS テンプレート内で `hensachi` 配列をループ処理し、表形式で表示する。注意点として、削除処理によって、配列内に `null` が含まれる可能性があるため、EJS 側で `if` 文等を用いた確認を行い、`null` 要素を表示しないように制御する必要がある。

4.3 更新機能 (Update)

この機能では、URL パラメータである `:number` を用いて配列にアクセスし、フォームから送信された値 (`req.body`) で詳細を上書きする。処理完了後は、`res.redirect('/hensachi')` により、一覧画面へ遷移する。

4.4 削除機能 (Delete)

配列の要素を削除するために、該当部分に `null` を代入する。これは、`splice` を使用すると、削除した部分にあとのデータが詰められるため、既存データを指す `:number` がずれてしまい、詳細表示や編集時のリンク先が誤ったデータを指すことがあったためである。`ejs` 側では表記上削除されるようにしている。

4. 千葉工業大学情報変革科学部情報工学科1年次の授業一覧 システムの開発者向け仕様書

1 データ構造

1.1 変数定義

本システムでは，データベースソフトウェアを使用せず，アプリケーションサーバ（IDapp.js）内の配列変数 `jugyou` にデータを格納する方法を採用している．`jugyou` はオブジェクトの配列変数であり，プロセス終了時にデータは消失する．

1.2 オブジェクトの構造

`jugyou` 配列の各要素は，以下のプロパティを持つオブジェクトである．変数名と実データの対応関係は表 4.1 の通りである [2]．

表 4.1: 変数名と実データの対応関係

変数名	データ型	本システムでの役割	データ例
<code>id</code>	Number	データの管理 ID	1
<code>code</code>	String	分野名	”教養基礎科目”
<code>name</code>	String	授業名	”英語理解基礎 1 ”
<code>change</code>	String	必修か選択か	”選択”
<code>passengers</code>	Number/String	単位数	1
<code>distance</code>	Number/String	備考	”基礎レベル対象の科目”

2 HTTP メソッドとリソース名

表 4.2 に、リソース名と HTTP メソッドを定義する。

表 4.2: 変数名と実データの対応関係

機能	HTTP メソッド	リソース名 (URL)	処理内容
一覧表示	GET	/jugyou	データ一覧画面 (jugyou.ejs) を表示する。
詳細表示	GET	/jugyou/:number	指定 ID の詳細画面 (jugyou_detail.ejs) を表示する。
新規登録 (画面)	GET	/jugyou/create	新規登録フォーム (jugyou.html) へリダイレクトする。
新規登録 (処理)	POST	/jugyou	フォームからデータを受け取り, 配列に追加する。
編集 (画面)	GET	/jugyou/edit/:number	編集フォーム (jugyou_edit.ejs) を表示する。
編集 (処理)	POST	/jugyou/update/:number	既存データを更新し, 一覧表示画面へリダイレクトする。
削除 (画面)	GET	/jugyou/delete/confirm/:number	削除確認画面 (jugyou_delete.ejs) を表示する。
削除 (処理)	POST	/jugyou/delete/:number	データを null 化し, 一覧表示画面へリダイレクトする。

3 ページ遷移

ページ遷移の様子を，図 4.1 に示す．

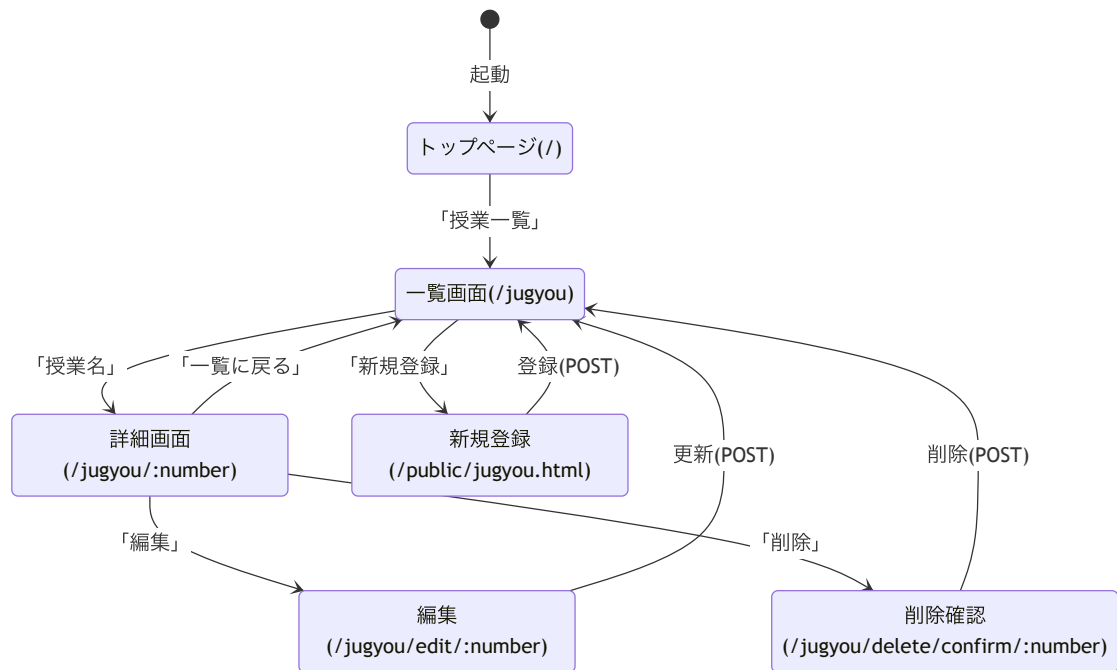


図 4.1: ページ遷移図

4 機能詳細と実装上の注意点

4.1 新規データ追加機能 (Create)

この機能では, `jugyou` 配列の末尾に新しいオブジェクトを追加 (`push`) し, 「`jugyou.length + 1`」 (一番最後の ID を 1 大きくしたもの) を新しい ID として付与する. 追加完了後, `jugyou.ejs` を用いて登録済みデータ一覧ページを表示する.

4.2 一覧表示機能 (Read)

この機能では, EJS テンプレート内で `jugyou` 配列をループ処理し, 表形式で表示する. 注意点として, 削除処理によって, 配列内に `null` が含まれる可能性があるため, EJS 側で `if` 文等を用いた確認を行い, `null` 要素を表示しないように制御する必要がある.

4.3 更新機能 (Update)

この機能では, URL パラメータである `:number` を用いて配列にアクセスし, フォームから送信された値 (`req.body`) で詳細を上書きする. 処理完了後は, `res.redirect('/jugyou')` により, 一覧画面へ遷移する.

4.4 削除機能 (Delete)

配列の要素を削除するために, 該当部分に `null` を代入する. これは, `splice` を使用すると, 削除した部分にあとのデータが詰められるため, 既存データを指す `:number` がずれてしまい, 詳細表示や編集時のリンク先が誤ったデータを指すことがあったためである. `ejs` 側では表記上削除されるようにしている.

5. 千葉工業大学情報変革科学部情報工学科の研究室一覧

システムの開発者向け仕様書

1 データ構造

1.1 変数定義

本システムでは，データベースソフトウェアを使用せず，アプリケーションサーバ（IDapp.js）内の配列変数 `kenkyu` にデータを格納する方法を採用している．`kenkyu` はオブジェクトの配列変数であり，プロセス終了時にデータは消失する．

1.2 オブジェクトの構造

`kenkyu` 配列の各要素は，以下のプロパティを持つオブジェクトである．変数名と実データの対応関係は表 5.1 の通りである [3]．

表 5.1: 変数名と実データの対応関係

変数名	データ型	本システムでの役割	データ例
id	Number	データの管理 ID	1
code	String	研究室（教授）名	”有本 泰子”
name	String	研究概要	”マルチモーダルな情報からコミュニケーションを解明”
change	String	研究内容	”感情の情報工学的な研究と、認知的・心理学的研究の両方を行う文理融合型研究を行います。コミュニケーション場面を研究対象とすることで、感情の表出や知覚に利用される表情や音声・生理反応を個別に分析するだけではなく、それらを統合的に分析し、システムの精度向上を図る仕組みを考えます。”
passengers	Number/String	キーワード	”音声処理・コミュニケーション・認知科学・マルチメディア・心理学”
distance	Number/String	研究室サイト URL	”https://mac-lab.org/”

2 HTTP メソッドとリソース名

表 5.2 に、リソース名と HTTP メソッドを定義する。

表 5.2: 変数名と実データの対応関係

機能	HTTP メソッド	リソース名 (URL)	処理内容
一覧表示	GET	/kenkyu	データ一覧画面 (kenkyu.ejs) を表示する。
詳細表示	GET	/kenkyu/:number	指定 ID の詳細画面 (kenkyu_detail.ejs) を表示する。
新規登録 (画面)	GET	/kenkyu/create	新規登録フォーム (kenkyu.html) へリダイレクトする。
新規登録 (処理)	POST	/kenkyu	フォームからデータを受け取り、配列に追加する。
編集 (画面)	GET	/kenkyu/edit/:number	編集フォーム (kenkyu_edit.ejs) を表示する。
編集 (処理)	POST	/kenkyu/update/:number	既存データを更新し、一覧表示画面へリダイレクトする。
削除 (画面)	GET	/kenkyu/delete/confirm/:number	削除確認画面 (kenkyu_delete.ejs) を表示する。
削除 (処理)	POST	/kenkyu/delete/:number	データを null 化し、一覧表示画面へリダイレクトする。

3 ページ遷移

ページ遷移の様子を、図 5.1 に示す。

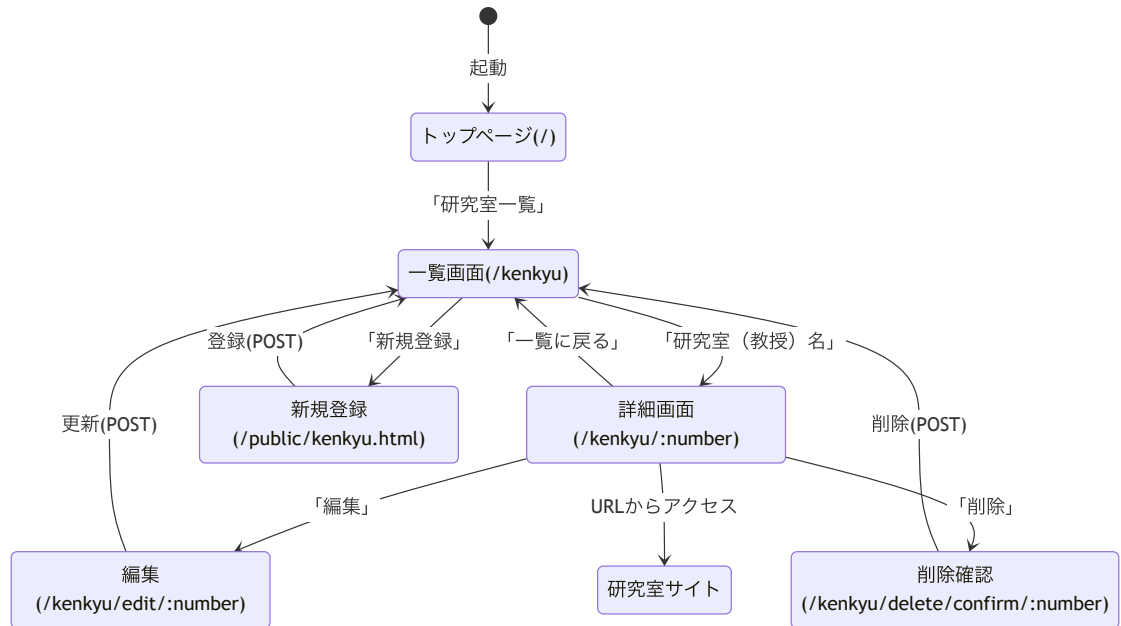


図 5.1: ページ遷移図

4 機能詳細と実装上の注意点

4.1 新規データ追加機能 (Create)

この機能では, `kenkyu` 配列の末尾に新しいオブジェクトを追加 (`push`) し, 「`kenkyu.length + 1`」 (一番最後の ID を 1 大きくしたもの) を新しい ID として付与する. 追加完了後, `kenkyu.ejs` を用いて登録済みデータ一覧ページを表示する.

4.2 一覧表示機能 (Read)

この機能では, EJS テンプレート内で `kenkyu` 配列をループ処理し, 表形式で表示する. 注意点として, 削除処理によって, 配列内に `null` が含まれる可能性があるため, EJS 側で `if` 文等を用いた確認を行い, `null` 要素を表示しないように制御する必要がある.

4.3 更新機能 (Update)

この機能では, URL パラメータである `:number` を用いて配列にアクセスし, フォームから送信された値 (`req.body`) で詳細を上書きする. 処理完了後は, `res.redirect('/kenkyu')` により, 一覧画面へ遷移する.

4.4 削除機能 (Delete)

配列の要素を削除するために, 該当部分に `null` を代入する. これは, `splice` を使用すると, 削除した部分にあとのデータが詰められるため, 既存データを指す `:number` がずれてしまい, 詳細表示や編集時のリンク先が誤ったデータを指すことがあったためである. `ejs` 側では表記上削除されるようにしている.

参考文献

- [1] パスナビ, 千葉工業大学 偏差値・共通テスト得点率 (ボーダーライン) , <https://passnavi.obunsha.co.jp/univ/2170/difficulty/>. (2025/12/28 閲覧).
- [2] 千葉工業大学 学生資料室, 教育課程表, https://kmsk.is.it-chiba.ac.jp/portal/iis/computer/computer_2025.pdf#page=3. (2025/12/28 閲覧).
- [3] 千葉工業大学, 情報変革科学部 情報工学科 研究室, <https://www.it-chiba.ac.jp/navi/page/detail/faculty/20/259>. (2025/12/28 閲覧).