

meijo-cup-result 説明資料

目次

- 使用している技術について
- システム概略図
- フォルダ構成
- 各ソースコードの説明
- 仕様について

使用している技術

フロントエンド

- Next.js ← これについてはこの資料で説明します

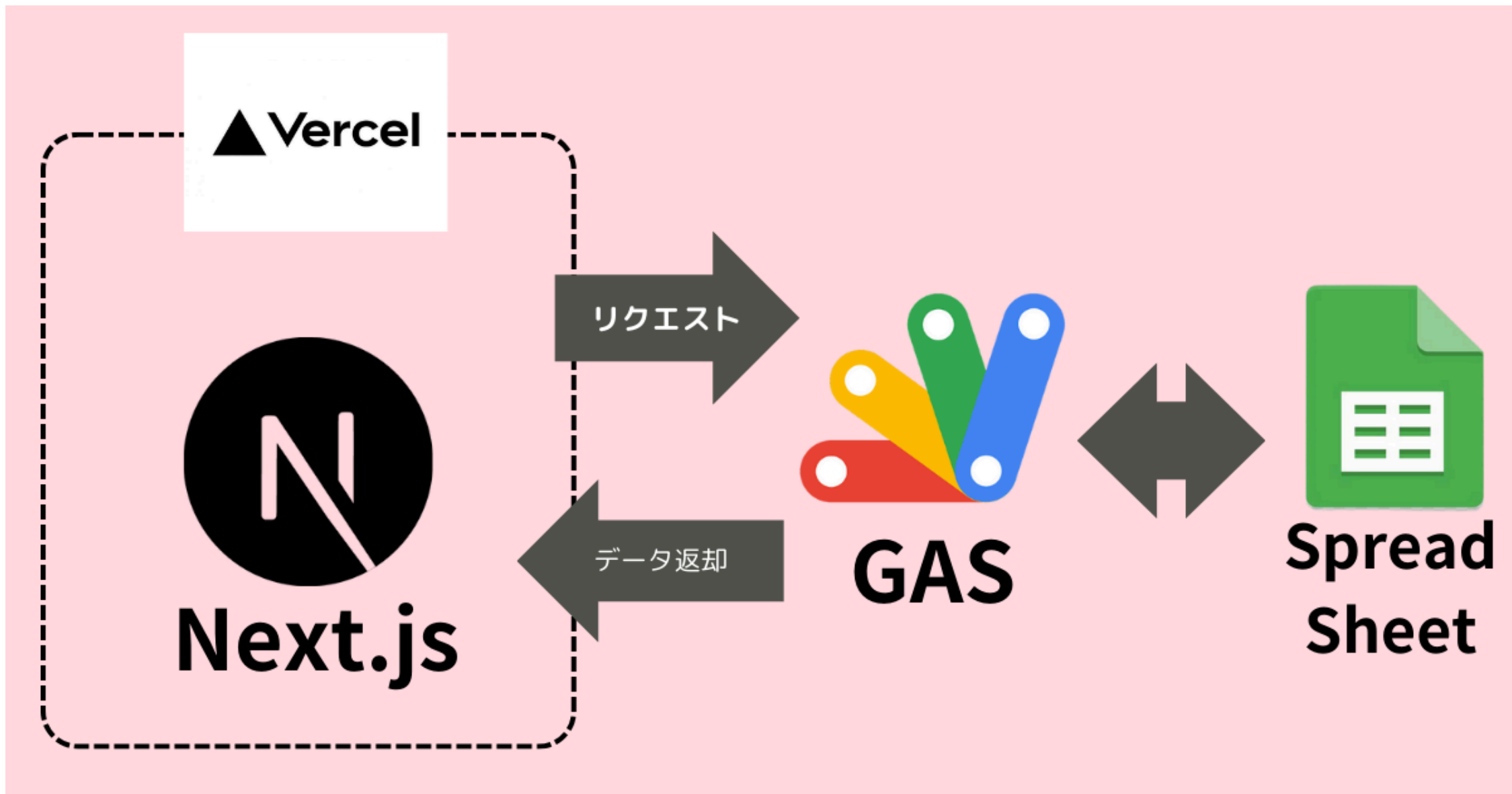
バックエンド

- Google Apps Scripts ← これについては説明しません

ホスティング

- Vercel ← 口頭で説明予定

概略図



フォルダー構成

```
app
├── favicon.ico
├── globals.css
├── layout.js
├── mushin.otf
├── opengraph-image.jpg
├── page.js
├── twitter-image.jpg
├── api
│   └── hello
├── route.js
└── components
    ├── Footer.jsx
    ├── Header.jsx
    ├── Result.jsx
    ├── result.module.css
    └── SlideShow.jsx
```

layout.jsについて

全体のソースコード

```
import "@app/globals.css"

export const metadata = {
  title: "第2回名城CUPレース結果",
  description: "第2回名城CUPのレース結果を確認することができます",
};
export default function RootLayout({ children }) {
  return (
    <html lang="ja">
      <body>
        {children}
      </body>
    </html>
  );
}
```

Next.jsの重要なファイルでエントリーポイントになる！！！！という認識でOK
※ちなみに消しても復活します。あと気にしないでよいです

layout.jsについて

```
export default function RootLayout({ children }) {  
  return (  
    <html lang="ja">  
      <body>  
        {children}  
      </body>  
    </html>  
  );  
}
```

この{children}の部分にpage.jsの内容がはいります。

ちなみにurlをクリックして初めに出るページがlayout.jsというイメージでとりまOK(雑すぎるが...)

layout.jsについて

```
import "@app/globals.css"

export const metadata = {
  title: "第2回名城CUPレース結果",
  description: "第2回名城CUPのレース結果を確認することができます",
};
```

import の部分はCSSをimportしています。これによりすべてのページでglobal.cssに書かれたレイアウトが反映されます。

export const meta の部分は,htmlでいうheadタグに書かれるtitleとdescriptionを設定しています。

page.jsについて

全体のソースコード

```
import Result from "@app/components/Result"

export default async function Page() {
  const rep = await fetch(process.env.NEXT_PUBLIC_API_URL);
  const data = await rep.json();

  console.log(data)

  return (
    <main>
      <Result data={data} />
    </main>
  );
}
```

page.jsについて

```
import Result from "@/app/components/Result"
```

componentsフォルダー内にあるResult.jsxをimportしている
importすることでResultコンポーネントとして利用することができる。

page.jsについて

```
export default async function Page() {  
  const rep = await fetch(process.env.NEXT_PUBLIC_API_URL);  
  const data = await rep.json();  
  便宜上省略...  
}
```

fetch関数について

Next.jsの関数は、通常のjsのfetch関数が拡張されていて、特に指定しない場合、getリクエストを送信される。

今回は、スプレッドシートのデータを取得するために使用

page.jsについて

```
export default async function Page() {  
  const rep = await fetch(process.env.NEXT_PUBLIC_API_URL);  
  const data = await rep.json();  
  便宜上省略...  
}
```

定数 repにはHTTPレスポンスの結果が入る！

ちなみにHTTPのbodyの中にスプレッドシートのデータが入っている！！

なのでjson()という関数を使うとbodyを読み取ることで
スプレッドシートのデータを取得することができる！！

page.jsについて(補足)

HTTPレスポンス

```
Response {  
  status: 200,  
  statusText: '',  
  .....省略.....
```

スプレッドシートのデータ(HTTPレスポンスのbody部分)

```
[  
  {  
    'ゼッケン番号': ,  
    '選手名': 'オオガネカイト',  
    '出場レース': '予備予選',  
    '所属チーム': 'ボラ協botチーム',  
    '順位入力': 3  
  } ,  
  .....スプレッドシートのDBシート行数分だけデータが増えます！  
]
```

page.jsについて

```
export default async function Page() {  
  const rep = await fetch(process.env.NEXT_PUBLIC_API_URL);  
  const data = await rep.json();  
  console.log(data)  
  return (  
    <main>  
      <Result data={data} />  
    </main>  
  );  
}
```

<Result data={data}>でコンポーネント間のデータのやり取りをしている(fetchで取得したスプレッドシートのdataを受け渡していると

Result.jsxについて

。。資料では省略！！知りたかったら聞いてね(´∀`)

指定されたレース結果(決勝の結果)のみを表示させるためのコンポーネント

combine.jsx

省略

組み合わせを表示させるためのコンポーネント

仕様について

セレクトボタンについて

レース組み合わせ表

ウェブサイトへ戻る

2歳本予選



レース結果表へ移動

画像のセレクトを変更する事で特定のレースの組み合わせを確認することができる

仕様について

サイト間の移動と結果と組み合わせ表の移動について

レース組み合わせ表

ウェブサイトへ戻る

2歳本予選



レース結果表へ移動

画像にある左右両端にあるウェブサイトへ戻るで元の公式サイトに戻ることができ、レース結果表へ移動をクリックすることでレースの結果表へ移動することができます！

2歳予備予選

ゼッケン番号	選手名	所属チーム
205	お	ボラ協botチーム
205	お	ボラ協botチーム
203	う	オリジナルチーム
203	う	オリジナルチーム
201	オオガネカイト	ボラ協botチーム
201	オオガネカイト	ボラ協botチーム
202	い	—
202	い	—
204	え	最強（仮）

仕様について

**重複データがある場合
(複数回、同じデータを送信してしめた場合)**

画像のように複数表示
されてしまうので注意

仕様について

レース名の入力が違う場合

歳予備予選		
ゼッケン番号	選手名	所属チーム
205	お	ボラ協botチーム

画像のように一人だけ別の表として作成されてしまうため注意！

(画像は本来 2 歳予備予選の人が別表で表示されてしまっている(´;ω;`))

運用方法

1. 組み合わせまたは結果がわかったらスプレッドシートのDB送信ボタンをクリックする！
2. 変更を反映させるためにVercelにてリデプロイを行う
3. 以上、二つを組み合わせや結果を伝えたいときに行う！

※注意

もしDB送信にミスなどがあった場合、スプレッドシートのDBシートのミスの該当箇所を修正し2のリデプロイの操作をする必要があります。

運用方法

1. スプレッドシートのDB送信ボタンをクリックする！

2歳クラス

2歳予備予選

2歳予備予選送信

ゼッケンNo.	選手名	生年月日	性別	都道府県	所属チーム	順位入力
201	オオガネカイト	2022/04/01	男子	愛知県	ボラ協botチーム	3
202	い	2022/04/02	女子	岐阜県	—	4
203	う	2022/04/03	男子	三重県	オリジナルチーム	2
204	え	2022/04/04	女子	愛知県	最強（仮）	5
205	お	2022/04/05	男子	岐阜県	ボラ協botチーム	1
206	か	2022/04/06	男子	三重県	—	6

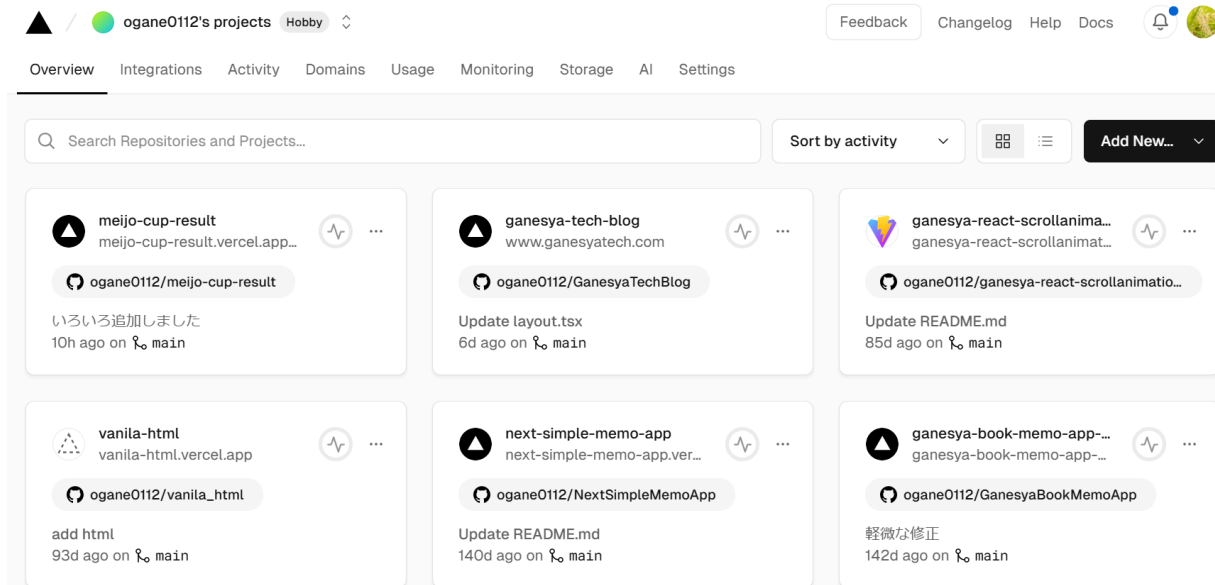
画像の赤枠のボタンが組み合わせ毎にあるので組み合わせが決定次第、ミスがないか確認してからクリックしてください！

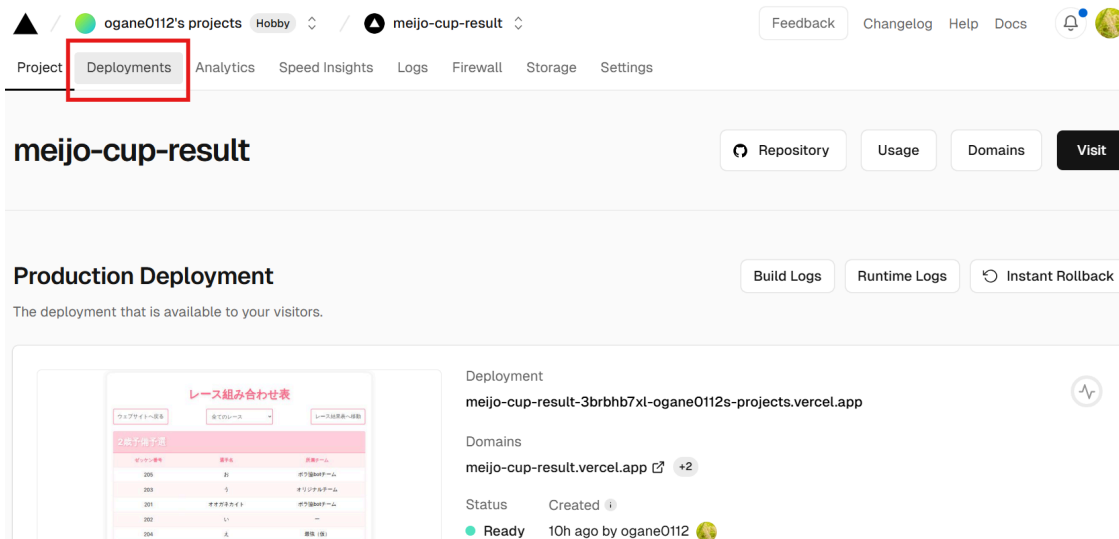
運用方法

2.Vercelにてリデプロイを行う！

↓のURLよりVERCELへアクセスすると画像のような画面になるので該当のプロジェクトを開く

<https://vercel.com/>





運用方法

2.Vercelにてリデプロイを行う

プロジェクトを開くと右の
ような画面になるので赤枠
部分をクリック！
(Deployments)

運用方法

2.Vercelにてリデプロイを行う

Deployments

🔄 Continuously generated from ogane0112/meijo-cup-result

Status 4/5

3brbhb7xl Production Current	● Ready 43s (10h ago)	main 295a03d いろいろ追加しました	10h ago by ogane0112 <div>...</div>
--	---------------------------------------	----------------------------	-------------------------------------

先ほどの操作をするとこのような画面になるので赤枠ボタンをクリック！！！！

※一番上のものが最新のデプロイなので必ず一番上をクリックしてください！

Instant Rollback



Promote



Redeploy

Inspect Deployment

View Source

Copy URL

Assign Domain

Visit 

運用方法

2.Vercelにてリデプロイを行う！

←のようなセレクトメニューが出てくるのでRedeployを選択
以上で操作は終わりです！！！！

運用方法

2.Vercelにてリデプロイを行う！



レース組み合わせ表

ウェブサイトへ戻る | 全てのレース | レース結果表へ移動

2歳予備予選

ゼッケン番号	選手名	所属チーム
205	お	ボラ協botチーム
205	お	ボラ協botチーム
203	う	オリジナルチーム
203	う	オリジナルチーム
201	オオガネカイト	ボラ協botチーム
201	オオガネカイト	ボラ協botチーム
202	い	—
202	い	—
204	え	競強 (仮)
204	え	競強 (仮)
206	か	—

Status
● Ready

Environment
Production [Current](#)

Duration ⓘ
56s (7m ago)

Domains ⓘ

- meijo-cup-result.vercel.app +1
- meijo-cup-result-git-main-ogane0112s-projects.vercel.app ⓘ
- meijo-cup-result-h390nsze2-ogane0112s-projects.vercel.app ⓘ

Source

- main
- 295a03d いろいろ追加しました

[Visit](#)

⋮

上記のような画面になっていれば成功です！！