みんなのノート開発記事　ーHU

こんにちは、チームCでフロントエンドとテックリーダーをしているHUGUANGXINです。

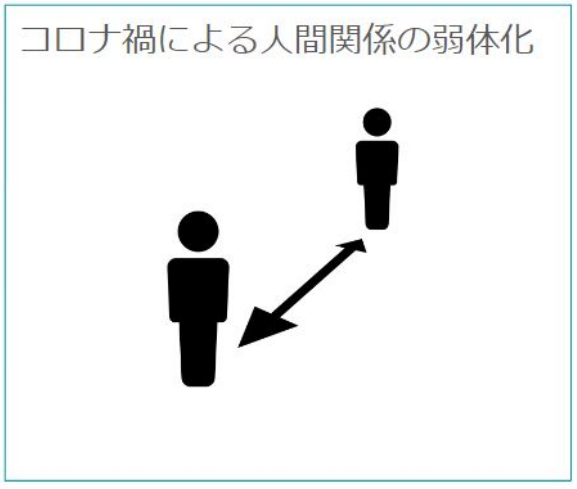
「みんなのノート」の開発を担当しています。

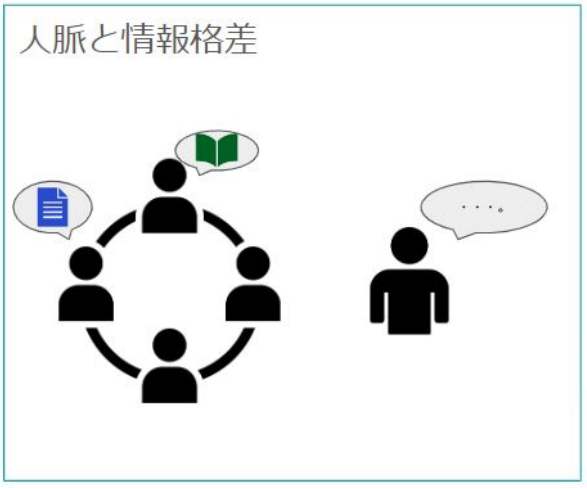
担当しているプロジェクトについて紹介したいと思います。

はじめに

学内で開催された融合デザインプロジェクトは、デザイン系と情報系の学生がランダムにチームを組み、プロジェクトを開発するイベントです。通常、情報系の学生はプログラムな部分を担当し、デザイン系の学生はUIなどのデザインを担当する。プロジェクト開発の経験はあったが、日本の学生たちとこのようなプロジェクトをするのは初めてで、少し緊張した。

みんなのノートとは





みんなのノートはコロナ禍による人間関係の弱体化と情報格差を改善するため岩大生同士が講義ノートなどまとめを

アップロード・閲覧することのできるウェブサイトです。

想定された機能はユーザー登録、ログイン、ノートのアップロードとダウンロード、ノート閲覧、ノート検索。

仕様書

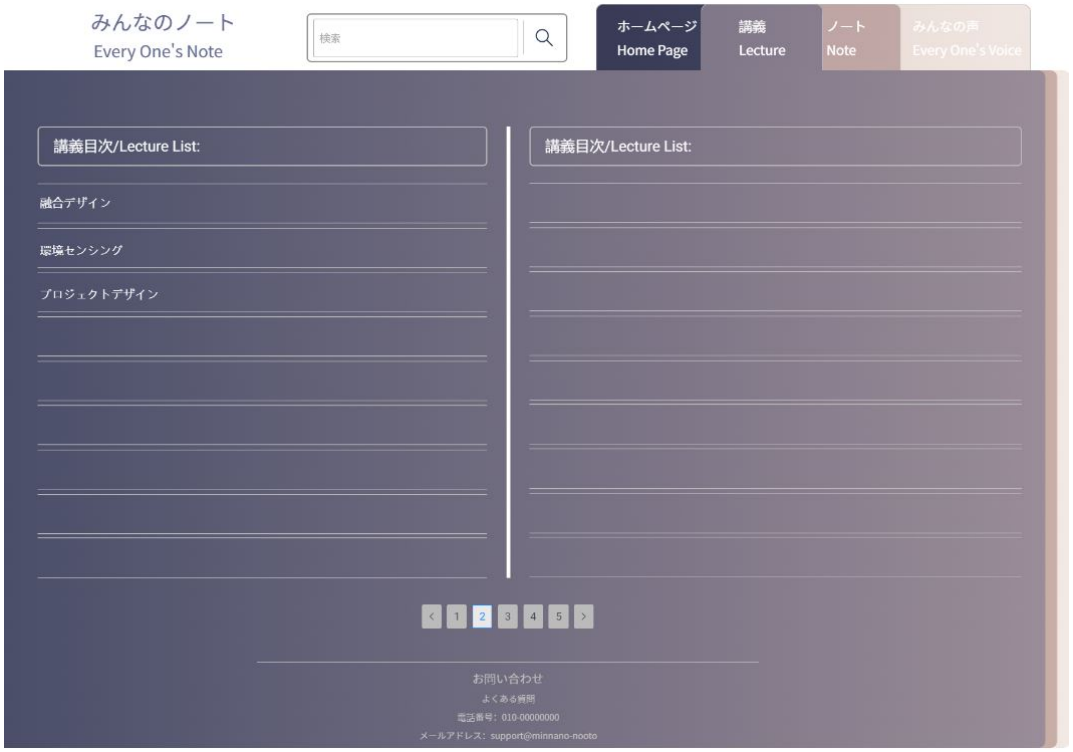
プロダクト要求仕様書(Product Requirements Document)はWebアプリケーション開発で最も重要な部分であり、製品要求仕様書を意味するPRDが無ければ、開発に携わるメンバー間での認識共有ができないためです。

チーム同士がプロダクトに関する前提や認識を共有して、意思疎通を図るためにPRDは欠かせません。

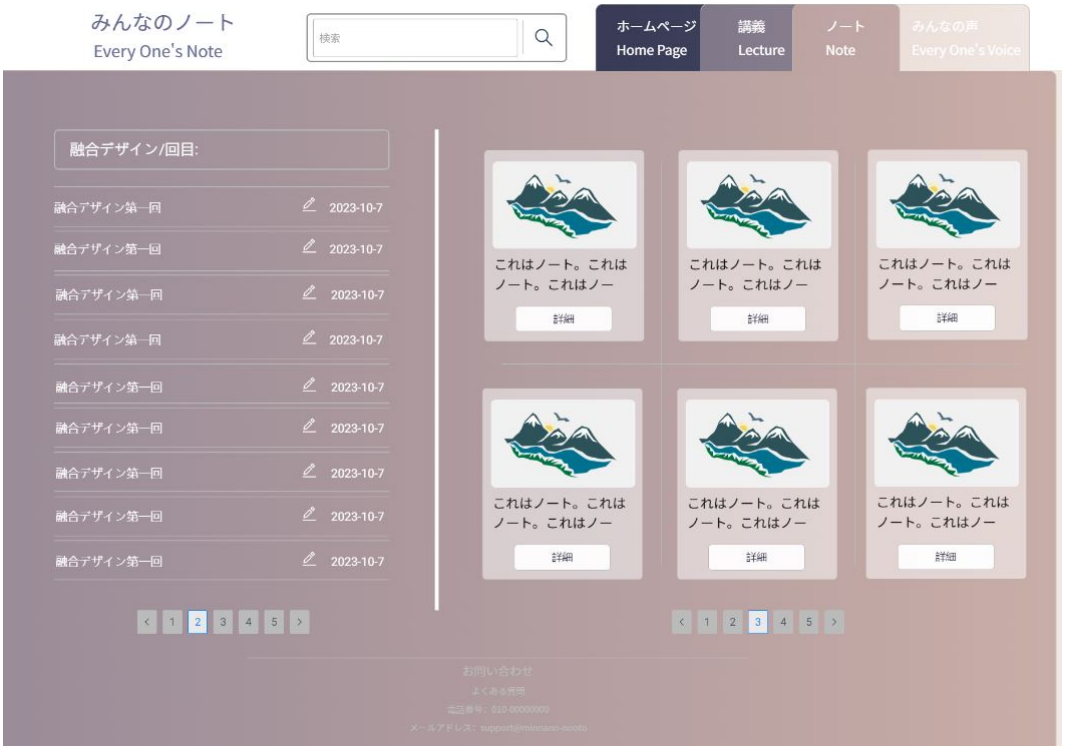
そのため、最初の作業は既存のニーズをもとに各ページのデザインです、ページのデザインはデザイン系のメンバー担当します。何度も修正と改善を重ねた後、デザイン案のいずれかのバージョンがチーム内で選ばれた。

- **ホームページ**

- **講義ページ**



- **ノートページ**



- **みんなの声ページ**

上記**priority**が最も高い4ページがプロジェクトの主要な構成となり、この完成度の仕様書ならフロントエンドに作られます。しかし、その前にこのプロジェクトで使用する技術スタックを決める必要があります。

技術スタックの選択

`"No perfect match, only match perfect."`

プロジェクトの技術アーキテクチャを決めるのは簡単なことではなく、チーム内のメンバーのバックグラウンドや専門分野も異なっていた。したがって、具体的な技術的ソリューションを決定する前に、まずチームのメンバーとコミュニケーションをとり、プロジェクトの要件を盛り込む必要がある。 やみくもに最新技術を使うのは得策ではなく、PMには十分なプロジェクト経験が求められるからだ。

フロントエンド

学生レベルのプロジェクトでは、ほとんどのメンバーがプロジェクト開発やコードを書いた経験がない。 そのため、ReactやVueといったフロントエンドフレームワークの利用は、開発環境の構築が難しく、学習コストが高いという問題があり、開発期間が非常に短い学生プロジェクトにとっては合理的ではありません。

最終的に、フロントエンドの技術スタックとしてネイティブ**HTML＋CSS＋Jquery**を選択することにした。これを選んだ理由は、ネイティブHTMLの学習コストが低く、開発環境を追加インストールすることなく、CDNインポートを通じてJqueryを使用することができるからだ。

バックエンド

バックエンドについては、私たちのプロジェクトは軽量なWebアプリケーションであり、フロントエンドとバックエンドのインタラクションはそれほど大きな要求ではないので、最も適しており、学習しやすい軽量なバックエンドフレームワーク、すなわち**Django**です。

Django は python ベースのバックエンドフレームワークで、コーディングの経験がなくてもすぐに始められる、習得しやすいプログラミング言語です。Djangoのデータベース操作は非常に使いやすく、単にデータモデルを定義することができ、その内部には、すぐに使えるプラグインがたくさん付属しており、ミニマリストの私のための最善の選択と思います。

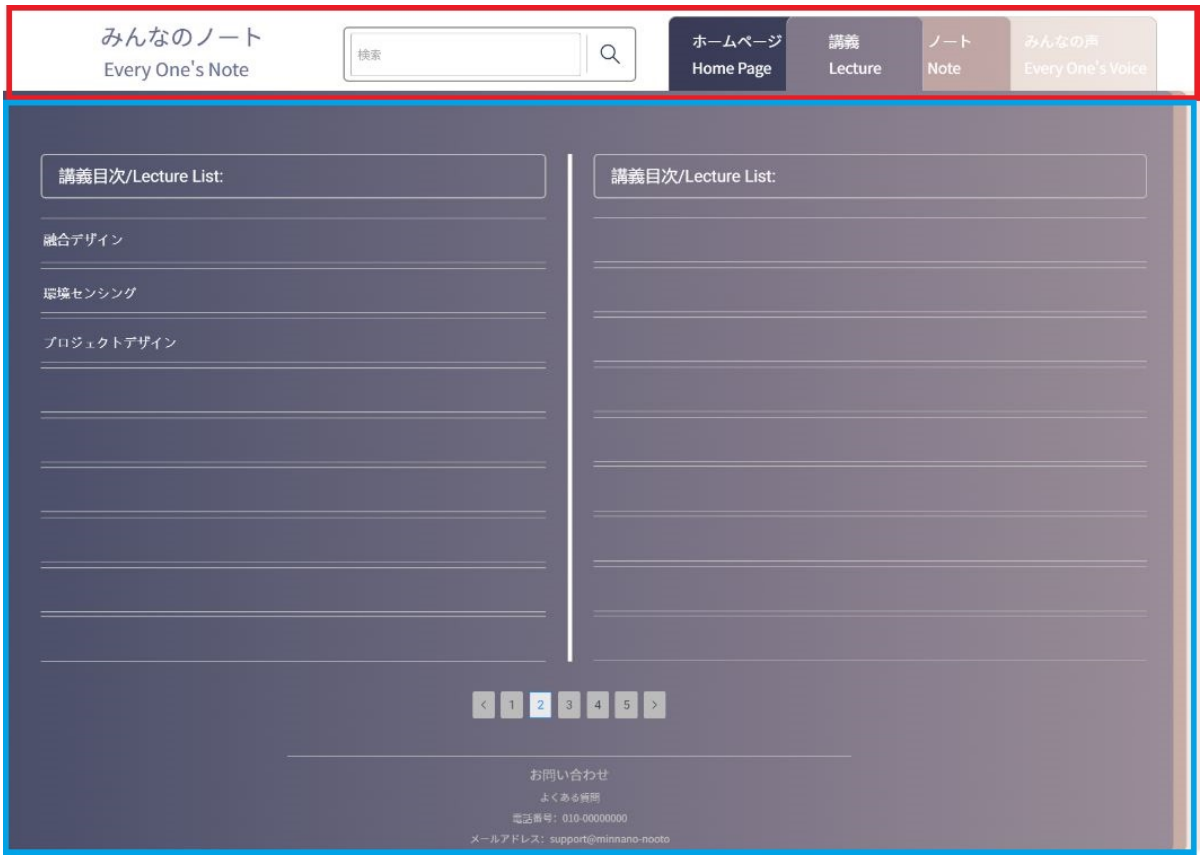
APIデザイン

フロントエンドとバックエンドの並行開発を確実にするためには、APIをデザインする必要がある。**RESTful API**デザインは、これ以上説明するまでもなく、一連の確立されたルールに従っている ([What Is a REST API? Examples, Uses, and Challenges]:https://blog.postman.com/rest-api-examples/)。

ページ構築

上記が完了したら、いよいよフロントエンドページを開始することができる。 ここでは、仕様書に従ってページや機能を実現する方法を詳しく説明します。

まず、実装する具体的なページデザインを分析しよう。



トップページを除けば、他のすべてのページの構成がヘッダー（赤い部分）とページ特定のコンテンツ（青い部分）で構成されていることを見つけるのは難しくない。各サブページは、ヘッダーの右側にある4つのタグを経て画面遷移する。明らかに、これは**シングルページアプリケーション(SPA)**です。

SPAとは？

SPAとは**"Single Page Application"（シングルページアプリケーション）**の略であり、単一のページで、Webアプリケーションを構成する設計構造のことです。SPAが実装されたページでは、遷移を行わずにコンテンツが切り替わるため、ユーザー体験（UX）の向上に繋がります。

従来のWebページでは、操作の度にページ全体を読み込みます。一方で、SPAが実装されたページは、JavascriptでHTMLの一部を差し替えて必要な部分だけを読み込むため、サーバーとの通信量を抑え、アプリケーションのパフォーマンス向上につながります。

私たちのチームのウェブデザイナーはウェブデザインの経験がなかったが、デザイン専門に値するシンプルで美しいSPAを作ることができた！

一般的に、現在な主流のフロントエンドフレームワークを使ってSPAを実装するのは簡単だが、ネイティブHTMLでSPAを実装するのは容易ではありません。**しかし、フロントエンドエンジニアである私たちの仕事は、デザイナーのアイデアを実装することです。**

調べてみると、iframeタグを使ってネイティブHTMLでSPAを実装できることがわかった。

<iframe>タグとは？

<iframe>は HTML の要素で、入れ子になった閲覧コンテキストを表現し、現在の HTML ページに他のページを埋め込むことができます。

それぞれの閲覧コンテキストにはそれぞれの文書があり、URL ナビゲーションができます。それぞれの埋め込み閲覧コンテキストのナビゲーションは、最上位の閲覧コンテキストのセッション履歴で直線化されます。他の閲覧コンテキストを埋め込んでいる閲覧コンテキストは、親閲覧コンテキスト と呼ばれます。最上位 の閲覧コンテキスト（親を持たないもの）は、通常はブラウザーのウィンドウで、 Window オブジェクトで表されます。

---

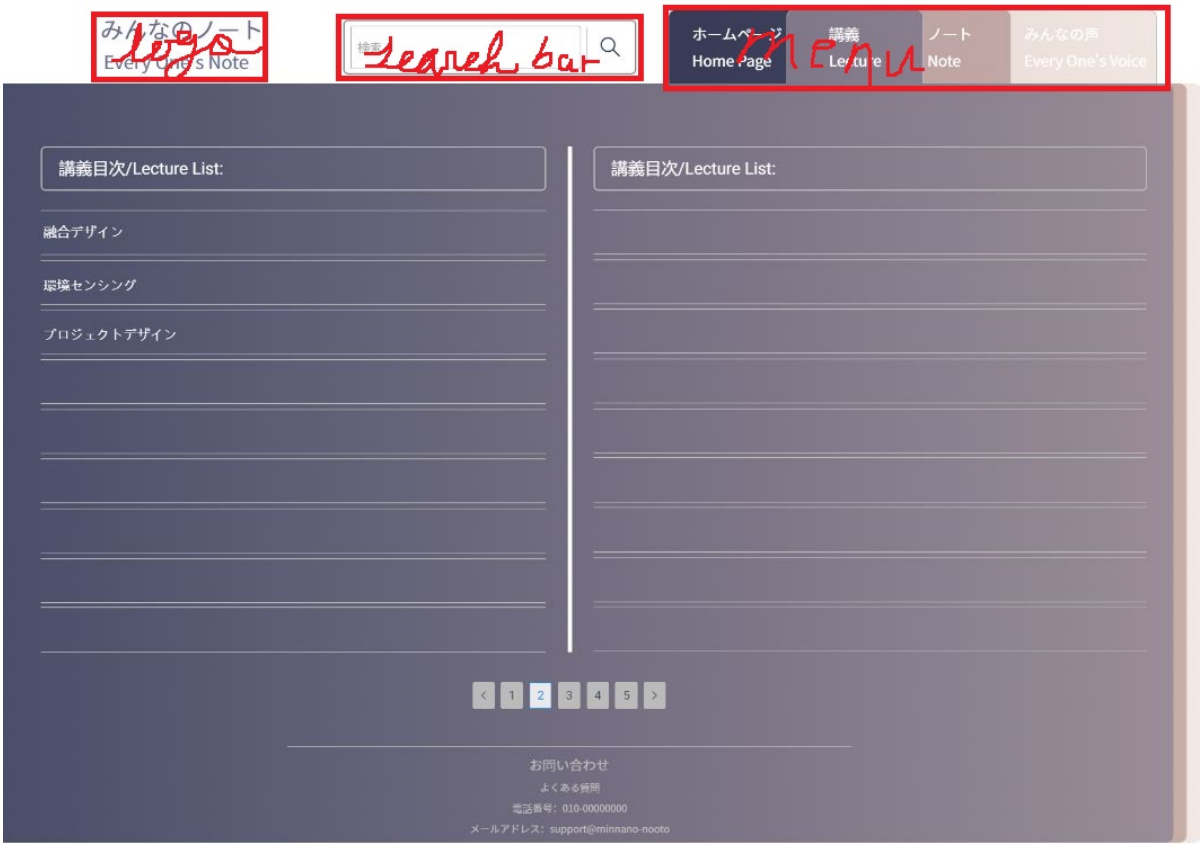
iframeタグの使用は現在のニーズを満たすことができるが、次のようなさまざまな問題を引き起こす可能性がある：

- **必要となるメモリやその他の計算リソースが増加します。**
- **アプリケーションへの悪意のあるコードの挿入が可能になり、機密データが攻撃者に公開される可能性があるため、セキュリティ リスクが生じる可能性があります。**
- **iframe を使用して別のドメインからコンテンツを埋め込むと、クロスドメインの問題が発生し、埋め込まれたコンテンツへのアクセスや操作が困難になる可能性があります。**

しかし、パフォーマンスやセキュリティに無視できる程度の軽量なwebアプリケーションの場合、SPAの開発にはiframeタグを使うことにした。

ヘッダーの実装

引き続き、仕様書に掲載されたヘッダーを分析すると。左のlogo、中央の検索ボックス、右のメニューを構成する3つの要素がある。



まずはこの3つの要素を実際にヘッダーに実装してみよう。

Logo

HTML:

```
<div class="logo">
    <p class="logo-text-jp">みんなのノート</p>
    <p class="logo-text-en">Every One's Note</p>
</div>
```

CSS:

```
.logo-text-jp{
    margin: 0px;
    font-size: 28px;
    font-family: 'source-han-sans-japanese', sans-serif;
    color: #666B92;
}
.logo-text-en{
    margin: 0px;
    font-size: 21px;
    font-family: 'source-han-sans-japanese', sans-serif;
    color: #666B92;
}
```

検索ボックス

HTML:

```
<div class="searchbar">
    <div class="searchbar-border">
    <input type="search" placeholder="検索" class="searchbar-input">
        <div class="searchbar-button-border">
            <button type="button" class="searchbar-button"> </button>
        </div>
    </div>
</div>
```

CSS:

```
.searchbar{
    flex: 1;
    position: relative;
    height: 70px;
    margin-top: 25px;

}
.searchbar-border{
    width: 350px;
    height: 60px;
    border: 2px solid;
    border-radius: 10px;
}
.searchbar-input{
    width: 280px;
    height: 50px;
    margin-left: 5px;
    margin-top: 5px;
    margin-bottom: 5px;
```

```
    border: 1px solid #BBBBBB;
    border-radius: 10px;
    color: #BBBBBB;
}
.searchbar-button-border{
    width: 50px;
    height: 50px;
    border-left: 1px solid;
    float: right;
    margin-top: 5px;
    margin-right: 5px;
}
.searchbar-button{
    width: 29px;
    height: 40px;
    margin-left: 10px;
    background-color: white;
    border:none;
    background-image: url(../image/ze-search\ 1.svg);
}
```

menu

HTML:

```
<div class="header-right">
    <a class="header-tab header-tab-1 header-tab-select" href="#homepage">
        <p class="header-tab-text header-tab-text-adjust">ホームページ</p>
        <p class="header-tab-text">Home Page</p>
    </a>
    <a class="header-tab header-tab-2" href="#lecture">
        <p class="header-tab-text header-tab-text-adjust margin-adjust">講義</p>
        <p class="header-tab-text margin-adjust">Lecture</p>
    </a>
    <a class="header-tab header-tab-3" href="#note">
        <p class="header-tab-text header-tab-text-adjust margin-adjust">ノート</p>
        <p class="header-tab-text margin-adjust">Note</p>
    </a>
    <a class="header-tab header-tab-4" href="#mnnvoice">
        <p class="header-tab-text header-tab-text-adjust">みんなの声</p>
        <p class="header-tab-text">Comment</p>
    </a>
</div>
```

CSS:

```
.header-tab{
    border-radius: 10px 10px 0px 0px ;
    height: 85px;
    width: 150px;
    margin-top: 15px;
    /* text-decoration:none;  */

}
.header-tab-1{
```

```
    z-index: 6;
    background-image: linear-gradient(to right,#393B56,#3D405B);
}
.header-tab-2{
    z-index: 8;
    position: relative;
    left: -20px;
    background-image: linear-gradient(to right,#7F7788,#857C8C);
}
.header-tab-3{
    z-index: 7;
    position: relative;
    left: -40px;
    background-image: linear-gradient(to right,#BCA4A3,#C0A7A4);
}
.header-tab-3-adjustz{
    z-index: 9;
}
.header-tab-4{
    z-index: 6;
    position: relative;
    left: -60px;
    background-image: linear-gradient(to right,#EADDD8,#EEE3DE);
}
.header-tab-text{
    margin: 0;
    font-family: 'source-han-sans-japanese', sans-serif;
    color: white;
    margin-left: 30px;
}
.header-tab-text-adjust{
    margin-top: 20px;
}
.margin-adjust{
    margin-left: 40px;
}
.header-tab-select{
    z-index: 10;
}
.card-select{
    z-index: 10;
}
```

このケースでは、右側メニューの各ラベルが選択されるとz-indexが変わるので、それに適したJavaScriptコードを追加する必要がある。

JavaScript:

```
window.showHeader = function (e) {
        $("#index-header").removeClass("header-disable");
        $(".header-tab-3").removeClass("header-tab-3-adjustz");
    }
    //to subpage
    window.toSearch = function (str) {
        $(".header-tab-3").removeClass("header-tab-3-adjustz");
```

```
        $("#index-header").addClass("header-disable");
        $(".card").removeClass("card-select");
        $(".header-tab-1").removeClass("header-tab-select");
    }
    window.toLecture = function (str) {
        showHeader();
        $(".card").removeClass("card-select");
        $(".card2").addClass("card-select");
        $(".header-tab").removeClass("header-tab-select");
        $(".header-tab-2").addClass("header-tab-select");
    }
    window.toNote = function (str) {
        showHeader();
        $(".card").removeClass("card-select");
        $(".card3").addClass("card-select");
        $(".header-tab").removeClass("header-tab-select");
        $(".header-tab-3").addClass("header-tab-select");
    }
    window.toVoice = function (str) {
        showHeader();
        $(".card").removeClass("card-select");
        $(".card4").addClass("card-select");
        $(".header-tab").removeClass("header-tab-select");
        $(".header-tab-4").addClass("header-tab-select");
        $(".header-tab-3").addClass("header-tab-3-adjustz");

    }
```

実装後の効果:

各サブページの実装

ここまでは順調で、サイトの一般的なヘッダーにiframeタグを埋め込んで、個々のサブページに効果を与えることができる。実装の原理はヘッダーのメニューと同じで、JavaScriptの関数を通してz-indexの値を変更し、サブページの遷移を実現する。

しかし、ホームページはiframeのサブページとして実装することもできますが、フレームのサイズを変更する機能が必要で、その場合、ヘッダーはホームページのiframeによって覆われてしまいます。

```
<div class="content">
            <div class="card4 card">
                <iframe id="mnnvoice-frame" width="100%" height="100%"></iframe>
            </div>
            <div class="card3 card">
```

```
            <iframe id="note-frame" width="100%" height="100%"></iframe>
        </div>
        <div class="card2 card">
            <iframe id="lecture-frame" width="100%" height="100%"></iframe>
        </div>
        <div class="card1 card">
            <iframe id="search-frame" width="100%" height="100%"></iframe>
        </div>
    </div>
```

実装後の効果:

ホームページ

講義ページ

ノートページ

みんなの声ページ

各ページコンテンツの実装

このプロジェクトのページ構成は複雑ではないので、ここでは紹介しないことにしよう。 しかし、参考のために、このプロジェクトで使用した技術をいくつか説明します。

フレックスレイアウト

- フレックスレイアウトとは、Flexible Box Layout Module のことで、その名の通りフレキシブルで簡単にレイアウトが組めます。
- 項目の横並びは、float や inline-block でもできますがフレックスレイアウトを使用すると色々なレイアウトが簡単に出来ます。
- フレックスレイアウトのメリットは「① CSSがシンプルな記述で済む、② 垂直方向の位置を柔軟に調整できる、③ CSSだけで並び順や折り返しを簡単に調整できる」などがあります。

ハッシュタグによるルーティング

このプロジェクトでは、iframeを使うだけではSPAの機能を完全に実現することができず、ユーザーが更新などの操作をするとサイト全体がリセットされてしまいます。したがって、サブページをルーティングする他の方法と組み合わせる必要がある。

```
    function hashchange() {
        let hash = document.location.hash;
        console.log(hash);
        if (hash === '#homepage') {
            window.toSearch();
        }
        if (hash === '#lecture') {
            window.toLecture();
        }
        if (hash === '#note') {
            window.toNote();
        }
        if (hash === '#mnnvoice') {
            window.toVoice();
        }
    }
    $(document).ready(function () {
        hashchange();
        //iframe router
        document.getElementById("search-frame").src = "search"
        document.getElementById("lecture-frame").src = "lecture"
        document.getElementById("note-frame").src = "note"
        document.getElementById("mnnvoice-frame").src = "mnnvoice"
        //page change
        window.addEventListener('hashchange', hashchange)
    });
```

こうすることで、ユーザーは特定のURLから対応するサブページにアクセスすることができる。

ホームページ: index

講義ページ: index/#lecture

ノートページ: index/#note

みんなの声ページ: index/#mnnvoice

サブページ間でのパラメータの受け渡し

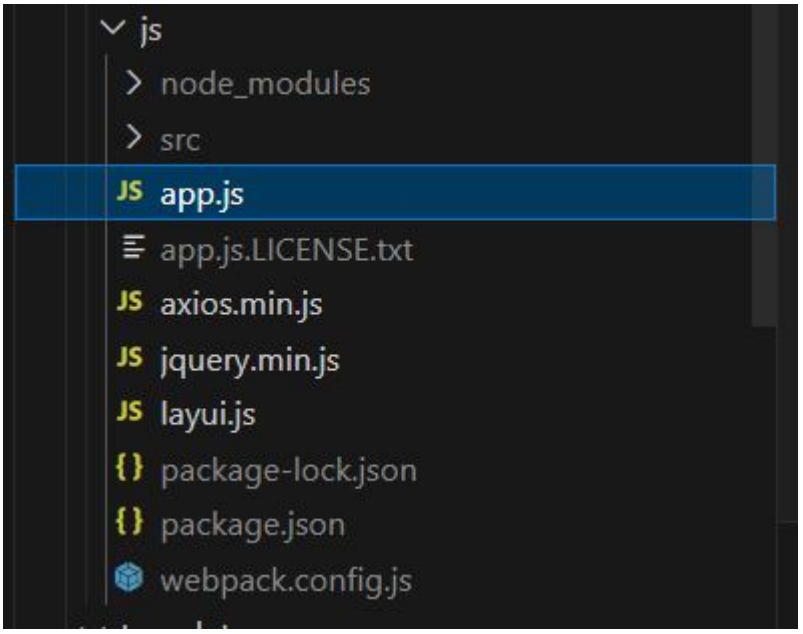
iframeにはもう一つ問題がある。それは、サブページ間でパラメータを渡すのが難しくなるということだが、windowオブジェクトにマップ関数をマウントするだけで、この問題を解決することができる。

```
window.store={
    _store:new Map(),
    set:function(key,value){
        this._store.set(key,value)
    },
    get:function(key){
    return this._store.get(key)
    }
}
```

渡したいパラメータは、store.setメソッドを通してウィンドウ・オブジェクトに書き込めばよく、対応するサブページのstore.getメソッドを呼び出せばパラメータをゲットできる。

Webpackによるモジュールバンドル

モジュールバンドラとは、複数のファイルを１つにまとめて出力してくれるツールのこと。
（複数ファイルをまとめることを「バンドル」と呼ぶ）

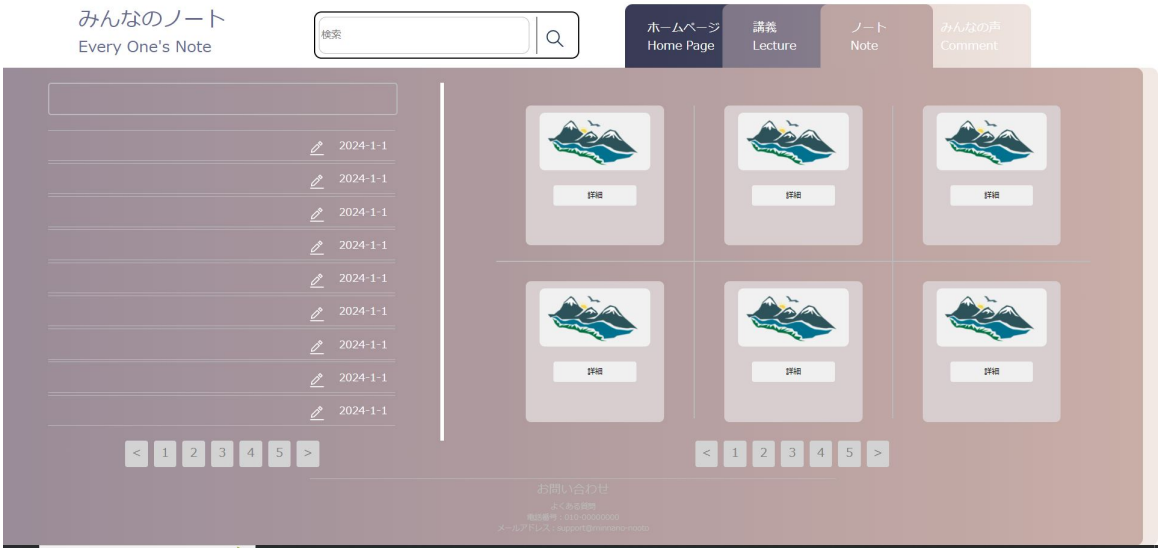


１つのファイルにまとめることで、リクエストも１回だけにし、性能を高めることができます。

各ページの最終成果


みんなのノート
岩手大学学生同士をONE TEAMに
Make the Iwate University's Students
became One Team
検索
お問い合わせ
よくある質問
メールアドレス：support@minnano-nooto


みんなのノート
Every One's Note
検索
ホームページ
Home Page
講義
Lecture
ノート
Note
みんなの声
Comment
講義目次/Lecture List
講義目次/Lecture List
< 1 2 3 4 5 >
お問い合わせ
よくある質問
電話番号：010-00000000
メールアドレス：support@minnano-nooto


みんなのノート
Every One's Note
検索
ホームページ
Home Page
講義
Lecture
ノート
Note
みんなの声
Comment
2024-1-1
2024-1-1
2024-1-1
2024-1-1
2024-1-1
2024-1-1
2024-1-1
2024-1-1
2024-1-1
< 1 2 3 4 5 >
詳細
詳細
詳細
詳細
詳細
詳細
< 1 2 3 4 5 >
お問い合わせ

まとめ

このプロジェクトの経験、特に日本人学生との共同開発から多くのものを得た。 用語や考え方の違いに戸惑い、ストレスを感じることもありましたが、積極的にコミュニケーションをとれば問題ありません！

また、プロジェクト全体の中で最も重要な役割であるテクリーダーを引き受けることを快諾してくれたメンバーの日本人学生にはとても感謝している。 普通、テクリーダーはプロジェクト全体の技術的な天井を代表するもので、プロジェクト・グループ全体で最も重要な役割に近い。

最後に、学んだことをまとめよう。

- **チームメンバー間のコミュニケーションは重要です！**
- **勇気をもって自分を表現する。**
- **自分の考えを他人と共有することも重要です!**