

# PORTFOLIO

パク・ジュヒョン  
Park JuHeoung



<https://github.com/jhpdevs>

# 目次

自己紹介

プロジェクト

私について



# 自己紹介

名前

パク・ジュヒョン

生年

1999年11月21日 (満22歳)

大学

永進専門大学

専攻

ウェブデータベース

資格証

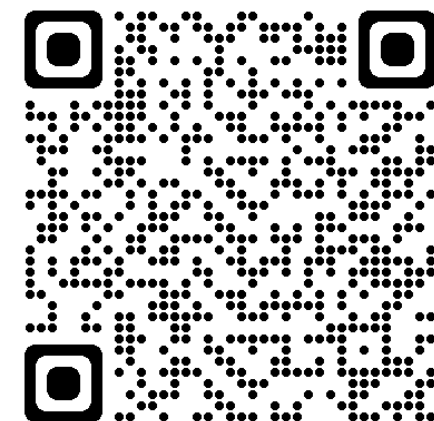
JLPT N2

連絡

jhpdevs@g.yju.ac.kr  
+82(10)2309-2437



<https://github.com/jhpdevs>







開発期間・人数

22年02月～05月・6人

サービス概要

チャットとビデオチャットで世界中の人と意思疎通できるコミュニケーションサービス

特徴

クライアントの負担を軽減するためにメディアサーバーを実装

機能

- ① チャットボット ( 翻訳 )
- ② ビデオチャット ( 自動字幕翻訳 )
- ③ GROUPで仲間が探せる

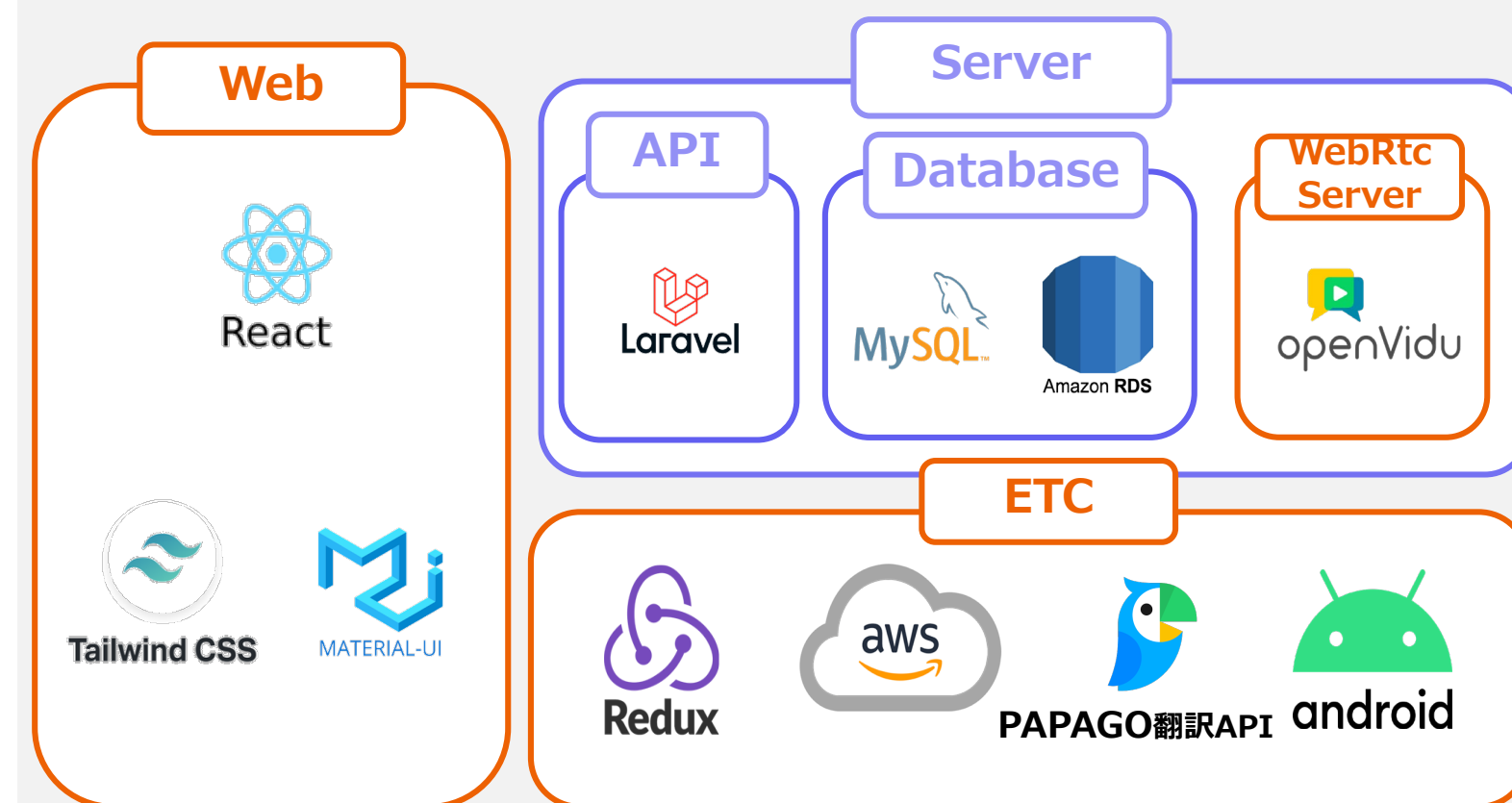


<https://github.com/mankai-wdj>

MANKAI



担当分野



# 私が担当した部分（フロントエンド担当）



React.jsとTailwindを使ってチャット、ビデオチャットの  
フロントエンドを担当しました!

# 私が担当した部分（ビデオチャット）

名前変更

The screenshot displays a video chat application interface. On the left, a shared screen shows the AWS Management Console with a table of EC2 instances. The table has columns for Name, Instance ID, and Instance State. The instances listed are 'Mankaproxy-env' (ID: i-0bb666a9aee19cc6e, State: 실행 중) and 'mankai-opensidu' (ID: i-0f84f621225f980e7, State: 실행 중). Below the table, details for the 'Mankaproxy-env' instance are shown, including its AMI (ami-036a8ecff227a5373) and platform (Linux/UNIX). A red box highlights the 'Mankaproxy-env' instance. In the center, a video feed shows a person wearing a white face mask, with a caption '박·주·형·말·하·는·중' (Park Ju-hyeong is speaking). On the right, a chat window shows a list of participants: '박·주·형·말·하·는·중' and '박·주·형·말·하·는·중'. A red arrow points from the '名前変更' (Name Change) label to the '이름 변경' (Change Name) button in the chat window. Below the chat window, a message says '처음まして、ジョンです。' (Hello, I'm John). A red arrow points from the 'チャット' (Chat) label to the chat window. At the bottom, a status bar shows a URL 'http://localhost:3000' and buttons for '공유 중지' (Stop Sharing) and '숨기기' (Hide).

Name	인스턴스 ID	인스턴스 상태
Mankaproxy-env	i-0bb666a9aee19cc6e	실행 중
mankai-opensidu	i-0f84f621225f980e7	실행 중

인스턴스: i-0bb666a9aee19cc6e(Mankaproxy-env)

Linux/UNIX(주요) ami-036a8ecff227a5373

모니터링 플랫폼 세부 정보


비밀성 Linux/UNIX

画面共有

Kurento Media Server(WebRtc)基盤の  
OpenVidu Serverを利用して機能を実装しました。



# 私が担当した部分（ビデオチャット・自動字幕翻訳）



パク・ジュヒョン

박주형 말하는중

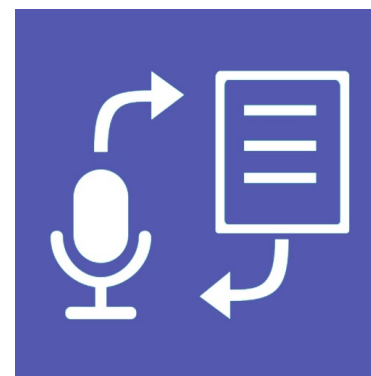
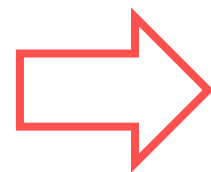
안녕하세요 박주형 입니다. (자동자막)  
こんにちは、パクジュヒョンです。 (자동자막)

①音声認識をして話している人の  
音声をテキストに変換(STT)

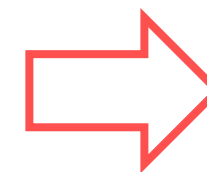
②テキストを自国語に翻訳



①言っている



②音声認識APIを使って  
音声をテキストに変換

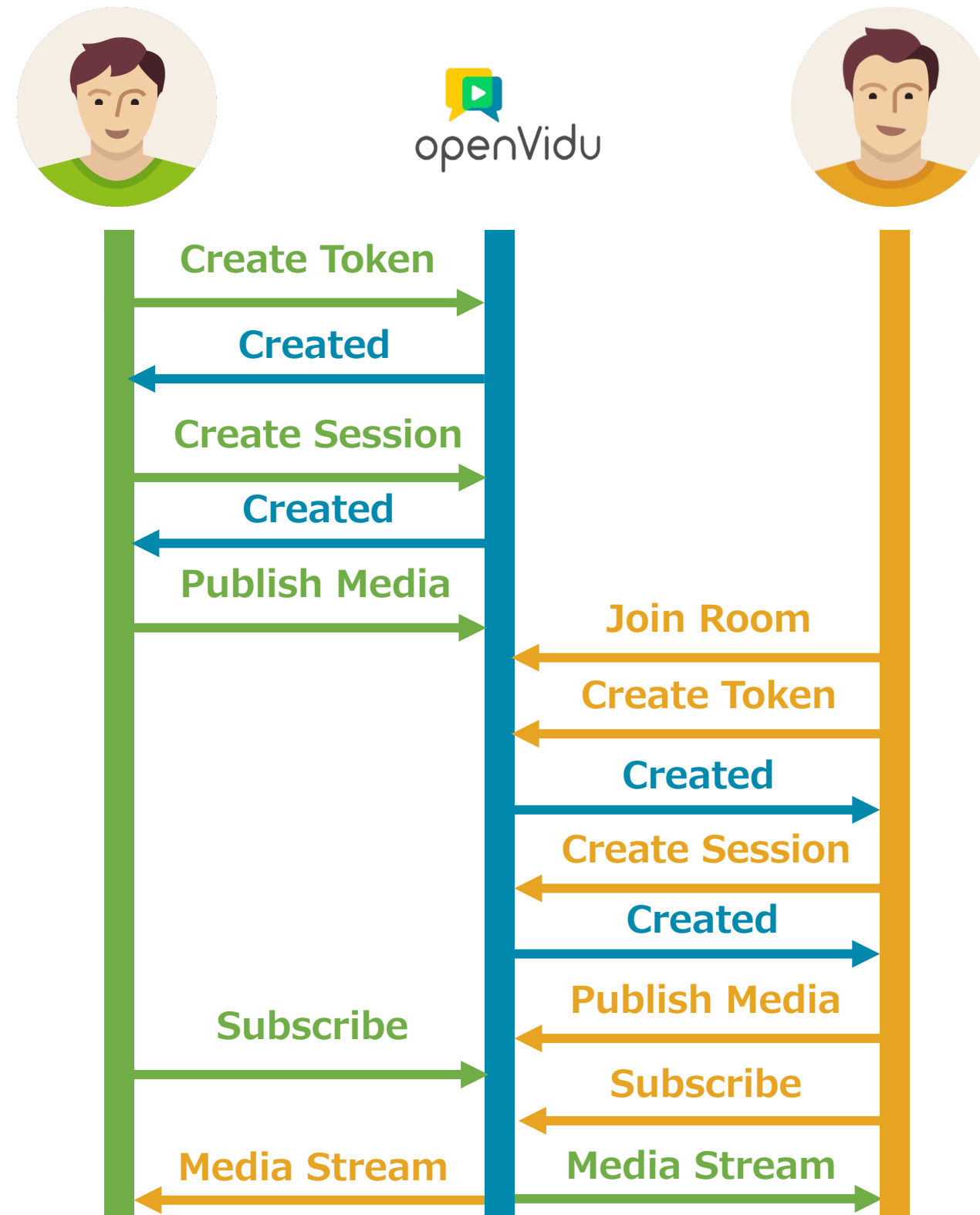


③変換されたテキストを翻訳APIを  
使って翻訳



# 担当分野詳細

## ・ビデオチャットプロセス・



## OpenViduメディアサーバーを使った理由

- ① WebRTCのさまざまなコネクション方式はP2P、SFU、MCU方式があります。P2P方式の場合、クライアントへの負担が大きいため、多くのコネクションが必要な当サービスには適合しないと判断しました。そして、OpenViduは状況を考慮してSFU、MCUのコネクション方式を自動的に選択するために選択しました。
- ② 画面共有、チャットを簡単に実装することかできます。
- ③ Roomに接続したユーザーがいない場合、Roomを自動的に消します。



# Checkmate

開発期間・人数

21年06月～08月・4人

サービス期間

21年09月～22年01月

サービス概要

出席と出席管理をもっと便利に！

特徴

- ① Geolocation APIを使ってGPSセンサーがない機器も位置情報を獲得可能
- ② レスポンシブウェブでモバイルもPCもOK!

効果

- ① 出席チェックにかかる時間を節約
- ② 出席管理の煩雑さを解決



GitHub

<https://github.com/jhpdevs/checkmate-vue-laravel> (PC)

<https://github.com/JHPDEVs/CheckMate-Android> (Android)

デモサイト

<https://www.ckme.live>

担当分野

Web



Server

DB



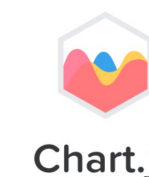
API



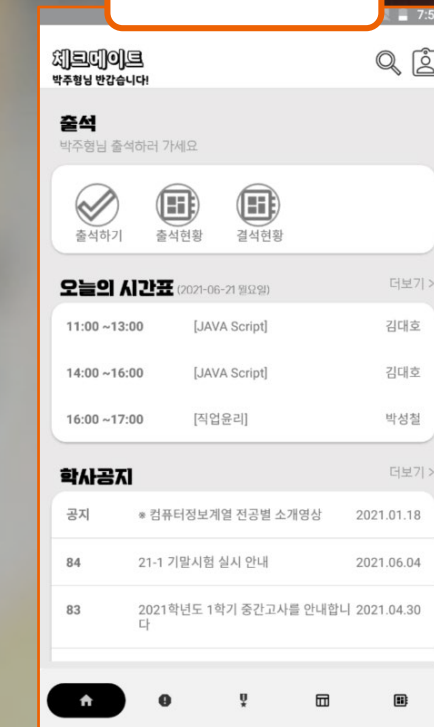
Hosting(NAS)



ETC

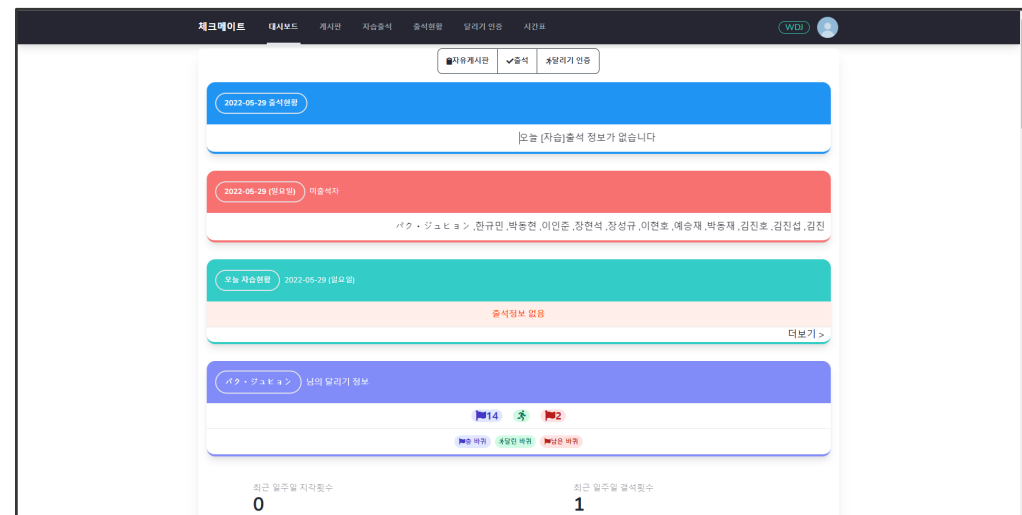


Android



# サービスの流れ・学んだこと

## Client(PC)



## Client(MOBILE)



## Backend Server

### 회원 관리

### 출席 관리

### 掲示物 관리

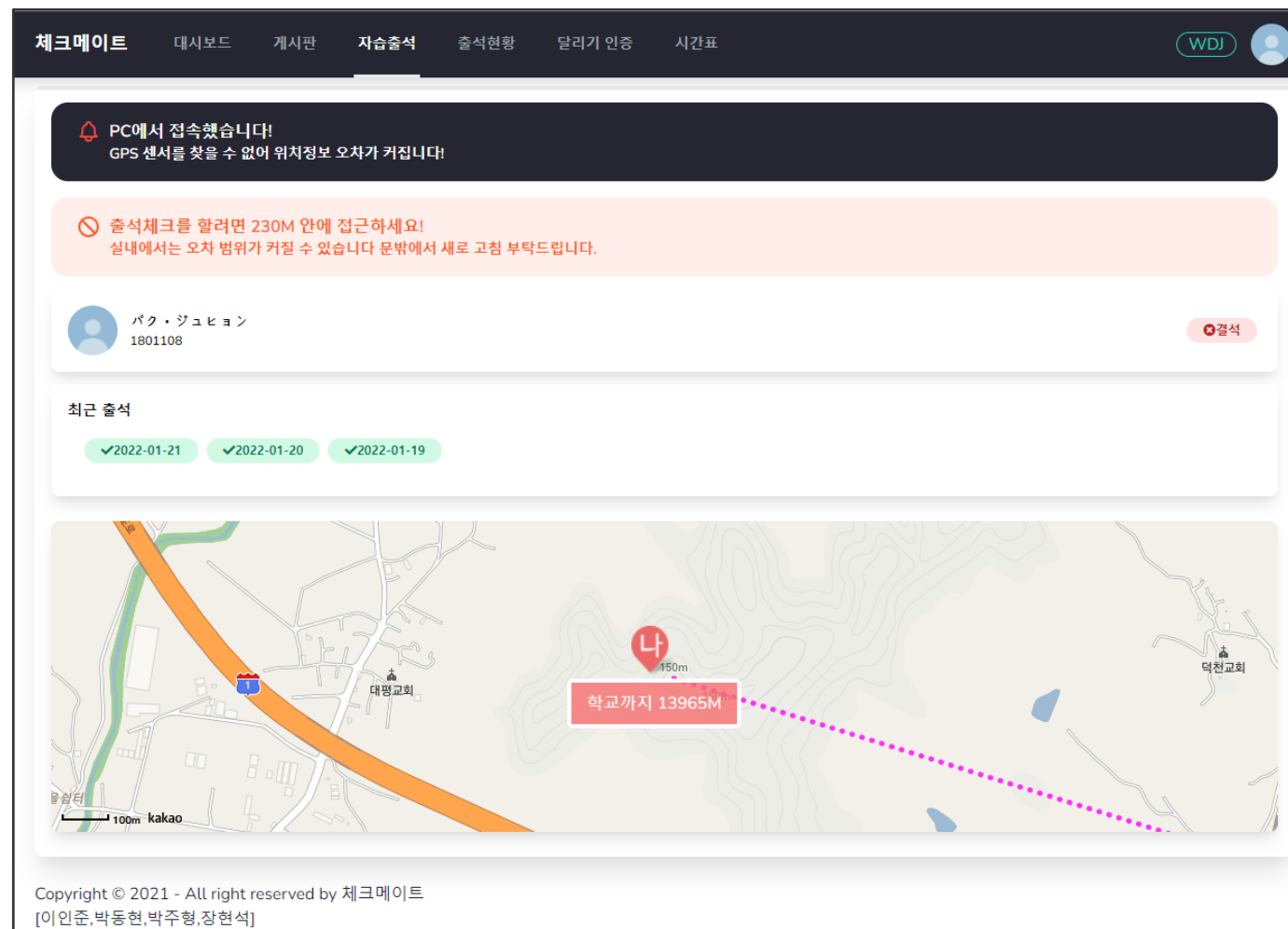
## 主要ポイント

- ① Geolocation APIを使ってGPSセンサーがない機器も位置情報を獲得可能
- ② どんな環境でも利用できるようにレスポンシブウェブデザインを実装

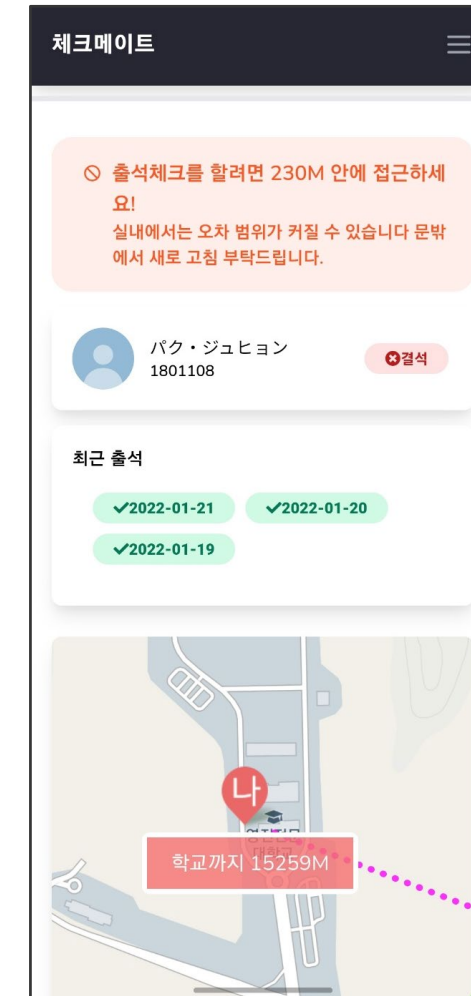
## 学んだこと

- ① 位置情報はセキュリティのためhttps環境で得られることがわかりました。
- ② Nas ( Synology)にホスティングしながらサーバの知識も学べる機会になりました。

# 私が担当した部分（フロントエンド担当）



PC ScreenShot  
Desktop (画面の解像度：1920x1080)

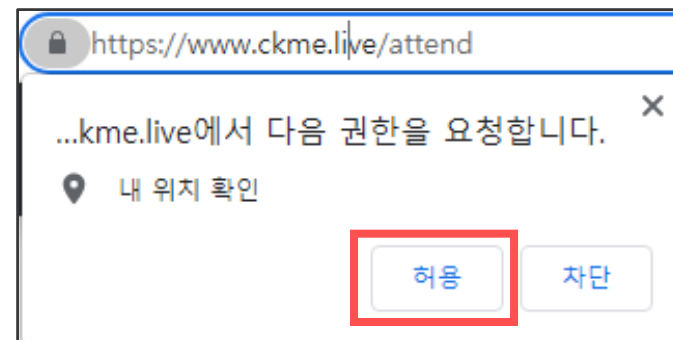


Mobile ScreenShot  
Galaxy S20(画面の解像度：3200x1440)

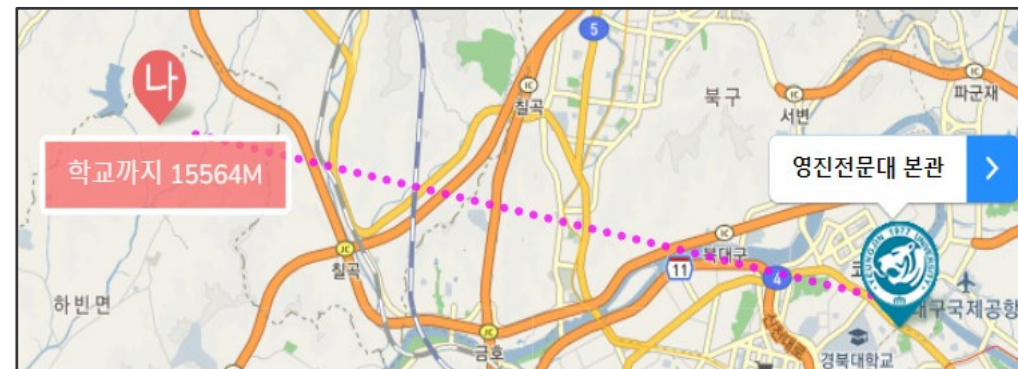
**Vue.js**と**Tailwind**を使って全体的なフロントエンド担当しました!  
**レスポンシブウェブ**でデザインされ、PCもモバイル(iPhone)も使用可能！



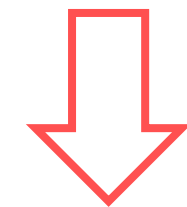
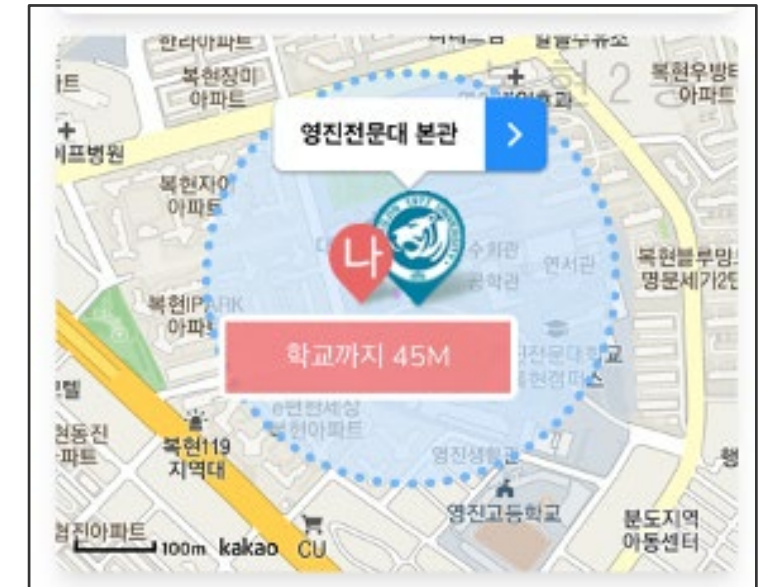
# 私が担当した部分（出席）



① Geolocation API를使って現在の位置情報を得る  
(GPSセンサーがない機器はIPで位置情報を得る)



② 現在の位置の座標と目的地の座標をつないで直線を作ります。  
(Kakao Map API使用)



③ 直線の長さ(距離)が230M以内なら出席ボタンを表示します。



# 個人プロジェクト

## FoodMap

開発期間

21年11月～12月

サービス概要

ダイエットのサポート!

特徴

- ① OpenApi基盤の公共ビックデータ活用
- ② プロキシサーバーを構築してCORSエラー解決

機能

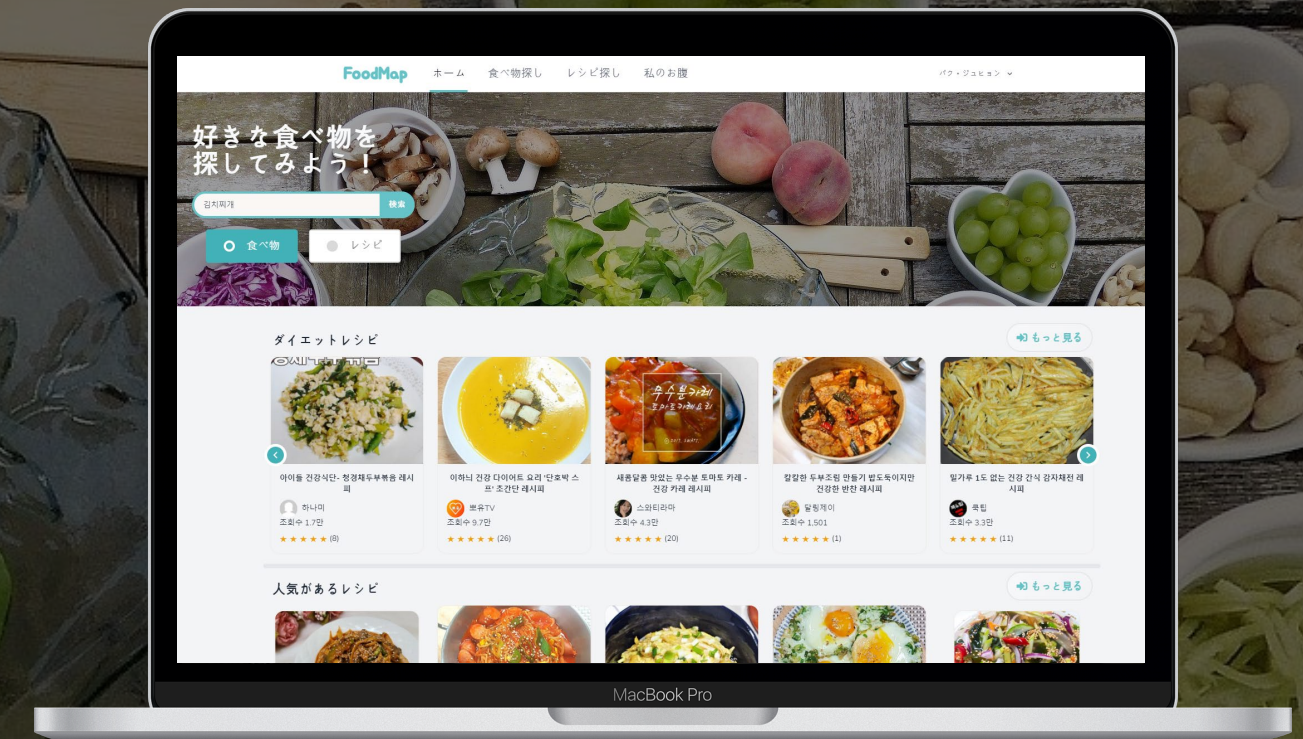
- ① 食べ物の栄養成分確認
- ② レシピ確認
- ③ 1日あたり必要な栄養分自動計算



<https://github.com/jhpdevs/foodmap-web>

デモサイト

<http://foodmap.dothome.co.kr> (PC)



Web



Vue.js



Tailwind CSS

daisyUI

Server

DB



SQLite

API



Laravel

Proxy Server

Cors-anywhere  
( node )

ETC

栄養分(API)

DATA . GO . KR



docker





## サービス流れ



レシピAPI Server

レシピデータ

DATA 공공데이터포털 .GO.KR

栄養成分 API Server

栄養成分データ

Proxy Server



Backend

食べた食べ物管理



食べた食べ物のデータ

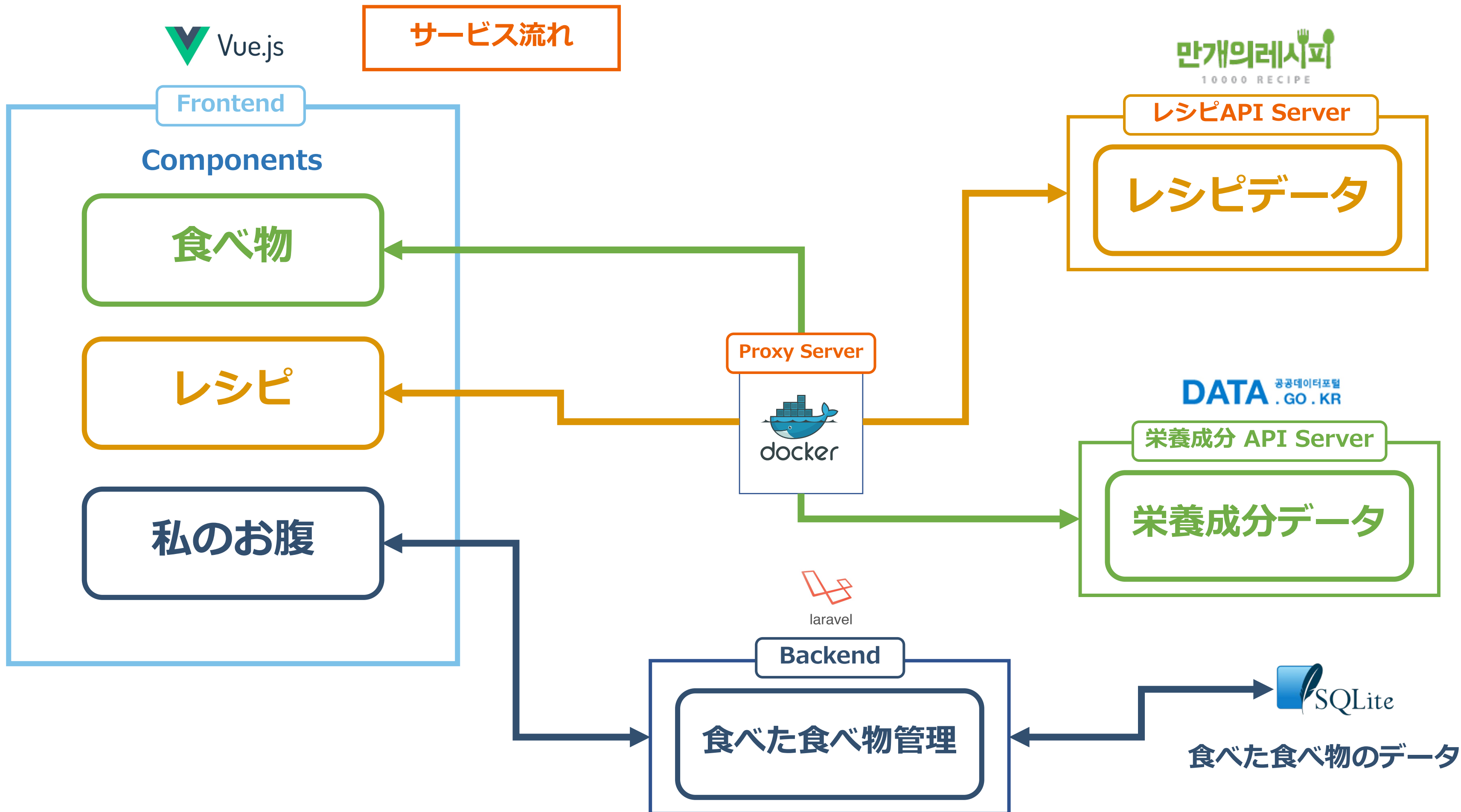
Frontend

Components

食べ物

レシピ

私のお腹



## サービス紹介

### 食べ物の栄養成分確認

製品名

전국(대표) **콩치김치찌개**

전국(대표) **돼지고기김치찌개**

検索をして栄養成分をみたい  
食べ物を選択する

1



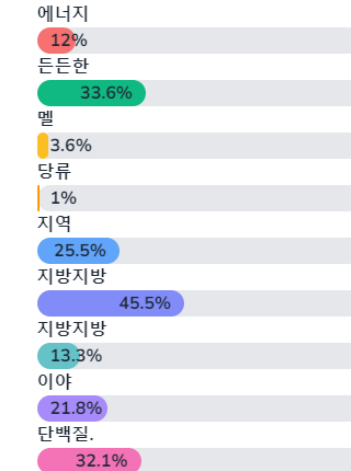
콩치김치찌개 (270g)



영양정보

1회 제공량	270g
1회 제공량	241
지하철	1일 영양성분 기준치
672mg	33.6%
9g	3.6%
당류 1g	1%
지방 13g	25.5%
지방지방 1g	45.5%
지방지방 2g	13.3%
65mg	21.8%
19mg	32.1%

1일 영양섭취추천 (%)



ボタンを押すと私のお腹に  
保存

量調節可能

栄養成分

分類

270G 콩치

단위

自動計算!

270

g

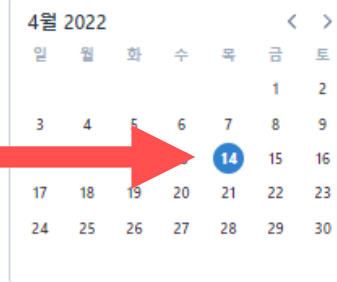
1일 영양성분 기준치 대비 %

12% [ 241 /2000]kcal

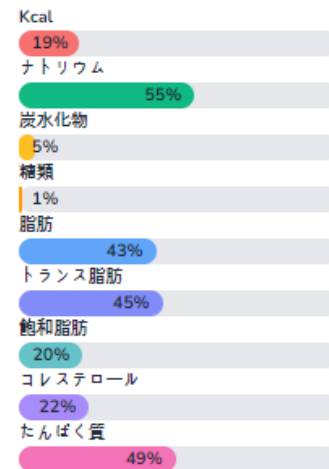
🍴 콩치김치찌개 270G食べる!

日付を選択して自分が食べた  
食べ物を確認可能

日付を選択してください。



1日栄養摂取基準 (%)



食べた食べ物の栄養成分自動計算  
(私のお腹)

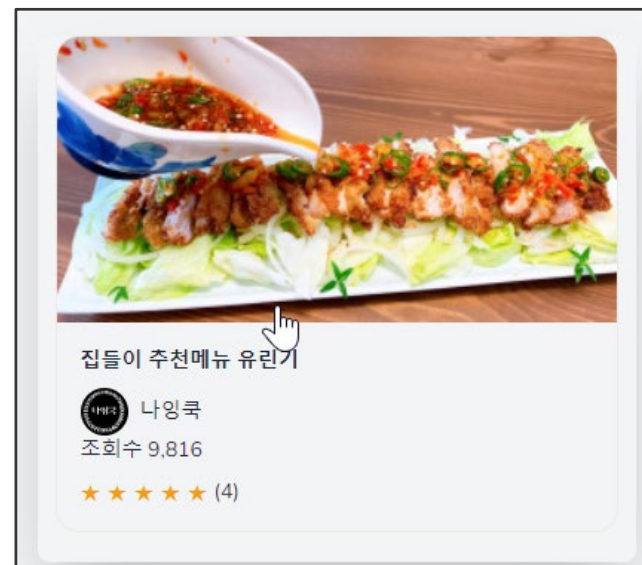
2

私のお腹

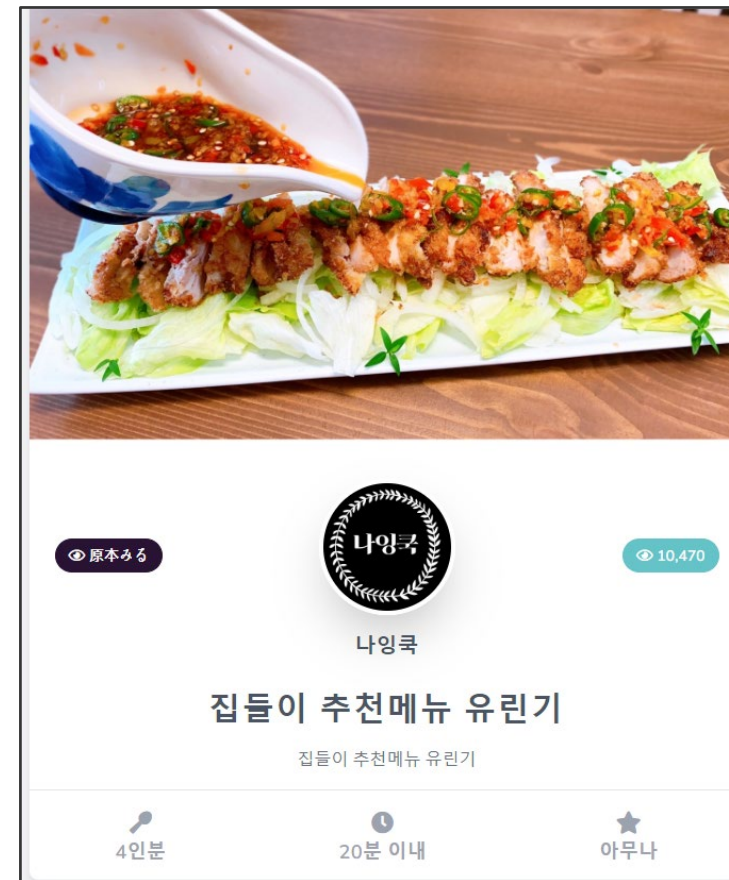
食べ物名	カロリー	炭水化物	たんぱく質	脂肪	糖類	ナトリウム	トランス脂肪	飽和脂肪	コレステロール
콩치김치찌개 270g	241 kcal	9g	19g	13g	1g	672mg	1g	2g	65mg
두부부침 100g	139 kcal	2g	10g	9g	0g	429mg	0g	1g	0mg

## 食べ物のレシピ確認

3



検索を試みたい  
① レシピ選択



材料Ingredients			
닭가슴살 혹은 닭다리살 400g	양상추 1/3개	양파 1/2개	계란 2개
빵가루 2C	미림 1T	소금 조금	후추 조금
간장 4T	레몬즙 2T	식초 4T	설탕 4T
다진마늘 1T	후추 조금	청양고추 2개	빨간파프리카 조금
노랑파프리카 조금			

レシピの多様な  
情報を確認できる！



## 学んだこと

- ① フロントエンドとバックエンドのドメイン名が違うとセセキュリティのためCORSエラーが発生することが分かりました。
- ② CORSエラー解決法の中にPROXYサーバ実装があることが分かりました。

## 感じたこと

開発をするためには  
サーバの知識も知っていなければならない  
ことを感じました。

# ありがとうございます！

パク・ジュヒョン

jhpdevs@g.yju.ac.kr

<https://github.com/jhpdevs>