



연구논문/작품 최종보고서

2023 학년도 제 1학기

제목 : OCR 모델을 통한 식단 기록 및 건강관리 웹사이트

허종원 (2017313150)

2023 년 월 일

지도교수: 정윤경 서명_____

계획(10)	주제(20)	개념(20)	상세(30)	보고서(20)	총점(100)

* 지도교수가 평가결과 기재

1. 요약

본 작품에서는 사용자의 건강한 식습관 개선을 돕기 위해, 인공지능 기반의 영양 분석 웹사이트를 개발하고자 한다. 이 웹사이트는 사용자가 업로드 한 식단 사진의 영양성분을 OCR 기술을 이용해 텍스트로 추출하며, 그 결과를 바탕으로 식단의 영양적 특성을 분석하고 평가한다. 이를 통해 사용자에게 맞춤형 식단 개선 방향을 제시하고, 개인별 건강한 식생활 지원을 제공한다. 또한, 이 웹사이트는 사용자가 식단 기록을 관리하고 성장을 추적할 수 있는 기능도 갖추고 있다

본 작품에서는 다음과 같은 핵심 기능을 구현하였다.

- OCR 기술을 활용한 식단 사진의 텍스트 추출: 사용자가 업로드한 식단 사진에서 영양성분 정보를 정확하게 인식하고 추출할 수 있도록 OCR 알고리즘을 개선하였다.
- 키워드 기반의 영양 분석 및 평가: 추출된 텍스트로부터 키워드를 분석하여, 식단의 영양소 비율, 칼로리, 비타민 및 미네랄 등 주요 영양성분을 측정하고 평가한다.
- 개인별 식단 기록 및 추적 기능: 사용자가 웹사이트에 기록한 식단 정보를 저장하고, 개선된 식습관의 변화와 성장을 시각적으로 파악할 수 있는 추적 기능을 제공한다.

이러한 기능들을 통해 사용자는 영양분석 웹사이트를 이용하여 건강한 식생활을 도모하고, 지속적인 식단 개선을 통해 전반적인 건강 상태를 향상시킬 수 있다. 이 프로젝트는 사용자의 건강한 식습관 개선을 돕는 차세대 웹사이트 개발에 한 발짝 다가갈 수 있는 좋은 발판이 될 것이다

2. 서론

2.1. 제안 배경 및 필요성

2.1.1 건강관리

코로나 이후 집에 있는 시간이 늘어남에 따라 건강관리에 대한 관심이 대폭 증가하였고, 건강한 식습관과 자기 관리에 대한 노력이 중요시 되고 있다. 대한민국은 고령화가 빠르게 진행되고 있으며, 만성질환, 비만, 스트레스 등 다양한 건강 이슈가 대두되고 있다. 이러한 상황 속에서 국민들의 삶의 질 향상을 위한 다양한 노력들이 진행되고 있다.

GDP가 오르고 선진국으로 발전해가면서 국민들이 취미생활과 여가시간에 쏟는 시간이 늘어나고 있다. 그 중에서 가장 중요하게 생각하는 것은 운동과 건강이다. 이에 따라, 레저와 스포츠 산업의 성장과 다양한 종류의 피트니스 센터와 프로그램

들이 등장하고 있다.

실제로 식단관리에 긍정적인 사람이 늘어났으며 건강 관련 식품 섭취 비율도 늘어나고 있다. 이는 건강기능식품 시장의 확대와 건강 관련 애플리케이션 및 서비스의 발전을 보여주는 지표이다. 또한, 신체 건강뿐만 아니라 정신 건강에 대한 인식도 증가하고 있어, 마음의 건강을 돌보는 다양한 프로그램과 서비스가 활성화되고 있다.



[그림 1] 2020 건강기능 식품 시장 소비자 실태조사



[그림 2] 아시아 국가 개인건강 관리 인식 조사 결과

건강 관리의 중요성 및 필요성은 나이에 국한하지 않고 나타나고 있으며, 특히 20-30대의 경우 바디프로필에 대한 관심도가 높아져 더욱 식단 관리의 중요성도 높아지고 있다. 이러한 세대는 자기관리와 건강한 몸매를 추구하며, 신체 건강을 개선하는 데 중점을 두고 있다.

하지만 바디프로필을 찍기 위해 극단적 다이어트를 하여 오히려 건강을 해치는 경우도 늘어나고 있다. 이는 일시적인 목표 달성을 위해 장기적인 건강을 희생하는 부작용을 초래할 수 있다. 건강한 식단이란 정해진 범위 내에서 적정량의 영양성분을 섭취하는 것으로, 균형 잡힌 식사와 적절한 영양 섭취가 중요하다.

건강한 식단은 피로 회복과 생활에 활기를 불어 넣고, 호르몬 균형과 면역력 강화에 도움이 될 수 있다. 또한, 신체적 건강뿐 아니라 정신적 안정감과 행복감도 향상시킬 수 있다. 이러한 상황에서 건강관리 웹사이트는 매우 유용한 도구가 될 수 있다. 이를 통해 사용자들은 전문가의 조언을 얻거나, 올바른 건강 관리 방법에 대한 정보를 공유할 수 있다.

따라서, 건강관리 웹사이트의 활용을 통해 건강한 식단 구성 및 올바른 운동 방법을 습득하고, 정기적인 체크업과 건강 관리의 중요성을 인식하는 것이 중요하다. 이를 통해 개인의 건강 수준을 높이고, 삶의 질을 개선하며, 건강한 사회를 만들어 갈 수 있다.

건강관리 웹사이트의 가장 큰 장점은 건강 정보를 쉽게 접할 수 있다는 점이다. 건강에 관한 다양한 정보를 제공하며, 사용자는 이를 기반으로 자신의 건강 상태를 파악하고 관리할 수 있다. 이를 통해 개인은 올바른 건강 정보를 습득하고, 실생활에 적용할 수 있는 방법을 찾을 수 있다.

특히, 최근에는 인공지능 기술을 활용한 건강 서비스도 등장하고 있다. 예를 들어, 심전도 검사, 혈압 측정 등의 건강 검진을 인공지능이 자동으로 수행하여 건강 상태를 파악할 수 있으며, 또한, 개인 맞춤형 운동 계획 및 식단 추천, 건강한 수면 습관 등을 제공하여 개인의 건강 상태를 효과적으로 관리할 수 있다.

그러나 건강관리 웹사이트를 이용할 때에는 신뢰할 수 있는 정보를 선택하는 것이 중요하다. 허위 정보나 과장된 광고에 현혹되지 않도록 주의를 기울여야 한다. 이를 위해 전문가의 의견을 참고하거나, 과학적 근거를 바탕으로 한 자료를 찾아보는 것이 좋다.

결론적으로, 건강관리 웹사이트는 건강 정보의 접근성을 높여주고, 개인의 건강 관리를 돕는 효과적인 도구로 활용될 수 있다. 인공지능 기술을 활용한 건강 서비스의 발전은 더욱 개인화된 건강 관리를 가능케 하며, 건강한 삶을 지속할 수 있는 지원을 제공한다. 이를 통해 건강한 사회를 만들어 가는데 큰 도움을 줄 것이다.

2.1.2 OCR 모델을 통한 분석 서비스

OCR(광학 문자 인식) 모델은 이미지나 스캔된 문서에서 문자를 인식하고, 컴퓨터가 이해할 수 있는 텍스트로 변환하는 기술이다. 이 기술은 디지털화와 정보의 효율적 관리에 큰 도움을 주며, 시간과 노력을 절약할 수 있다. OCR 모델은 의료, 금융, 교육 등 여러 산업 분야에서 다양하게 사용되며, 특히 문서 처리와 관련된 작업에서 탁월한 성능을 발휘한다.

긍정적인 사례로 의료 서류에 대한 OCR 인식을 통해 의사들은 빠르고 정확하게 환자 정보를 얻을 수 있게 도와주며 환자 치료에 필요한 정보를 더 빠르게 확보하고, 의료진들의 업무 효율성을 높이는 데 도움을 준다. 또한, OCR 인식을 이용해 금융 문서나 수표를 빠르게 처리하며 금융 서류 처리 과정에서 발생하는 인적 오류를 줄일 수 있다.

교육 분야에서도 OCR 모델은 학생들의 성적표나 시험지를 자동으로 채점하거나, 서적과 논문의 텍스트를 디지털로 변환하는데 사용되어, 교육자원의 공유와 활용을 용이하게 한다. 또한, 공공 부문에서는 기록 보관과 관리를 효율적으로 수행할 수 있게 해주며, 정보 검색 시간을 단축시키는 역할을 한다.

OCR 모델을 기반으로 하는 구체적인 서비스로는 모바일 앱으로, OCR 인식을 이

용하여 스캔한 문서를 PDF 파일로 변환해주는 Adobe Scan, 노트 앱으로, OCR 인식을 이용하여 스캔한 문서나 이미지 내의 텍스트를 인식하여 검색이 가능한 Microsoft OneNote 등이 있다. 이 외에도, 여러 번역 앱에서 OCR 기술을 활용하여 외국어 텍스트를 캡처하고 해당 언어를 실시간으로 번역해주는 서비스가 제공되고 있다. OCR 모델은 지속적으로 발전해 나가고 있으며, 앞으로 딥러닝 기술의 도입으로 저화질 이미지 자체를 고화질 이미지로 변환하여 인식할 수 있는 방향으로 발전하고 있다. 또한, 여러 자연어 처리 모델과 접목하여 다국어 인식 기능 개발하여, 다양한 언어의 문서를 인식하고 변환하는 연구가 활발히 진행되고 있다.

이처럼 OCR 모델을 활용한 서비스는 개인과 기업, 국가 차원에서 모두 효율적인 정보 관리와 소통을 가능케 하며, 경제적 이익과 시간 절약을 가져온다. 또한, 장애인들에게는 보조 기술로서 접근성을 높여주는 역할도 한다.

그러나 OCR 모델의 성능은 폰트, 해상도, 배경 노이즈 등 다양한 요소에 영향을 받을 수 있다. 이로 인해 오류가 발생할 수 있으며, 정확도를 높이기 위한 연구와 개선이 지속적으로 필요하다. 또한, OCR 모델의 활용에는 개인정보 유출 위험이 따르므로, 이를 고려한 보안 대책과 정책이 마련되어야 한다. 또한, 국제적인 협력을 통해 다양한 언어와 지식을 공유하며, 인공지능 기술의 발전을 촉진시켜야 할 것이다.

결국, OCR 모델을 통한 분석 서비스는 현대 사회에서 다양한 분야에서 활용되며 지속적인 발전을 거듭하고 있다. 이를 통해 정보 접근성을 높이고, 글로벌 시대에 걸맞은 효율적인 커뮤니케이션을 가능케 할 것으로 기대된다.

2.2. 목표

본 작품은 최근 늘어나고 있는 헬스 어플들에서 보여주는 식단 분석에서 한층 더 나아가 제대로 된 정보를 알리고자 영양 성분 사진에서 텍스트를 추출하여 저장함으로써 정확한 정보만을 사용자에게 제공한다. 이를 통해 사용자는 일일 영양소 섭취 목표를 쉽게 확인하고 관리할 수 있다.

또한, 식품의약품안전처에서 제공하는 Open API를 통해 식품의 검색을 제공하고 그 정보를 논문을 통해 입증된 계산 공식을 통해 인식된 식단 정보를 기반으로 사용자의 영양 섭취량을 계산하고, 권장 섭취량과 비교하여 그 정보를 기반으로 사용자의 식습관을 분석하여 건강한 식습관을 유지하도록 도와준다. 이를 통해 개인별 맞춤형 건강관리가 가능해지며, 사용자가 섭취해야 할 영양소를 체계적으로 파악할 수 있다.

COVID-19 대유행 이후, 개인들은 건강관리에 대한 중요성을 더욱 인식하게 되었습니다. 이러한 상황에서, 개인들이 자신의 장기적인 건강을 유지하기 위해 영양 성분과 칼로리 등을 포함한 자신의 식습관을 기록하고 분석하는 것은 높은 기대가치를 가질 것입니다.

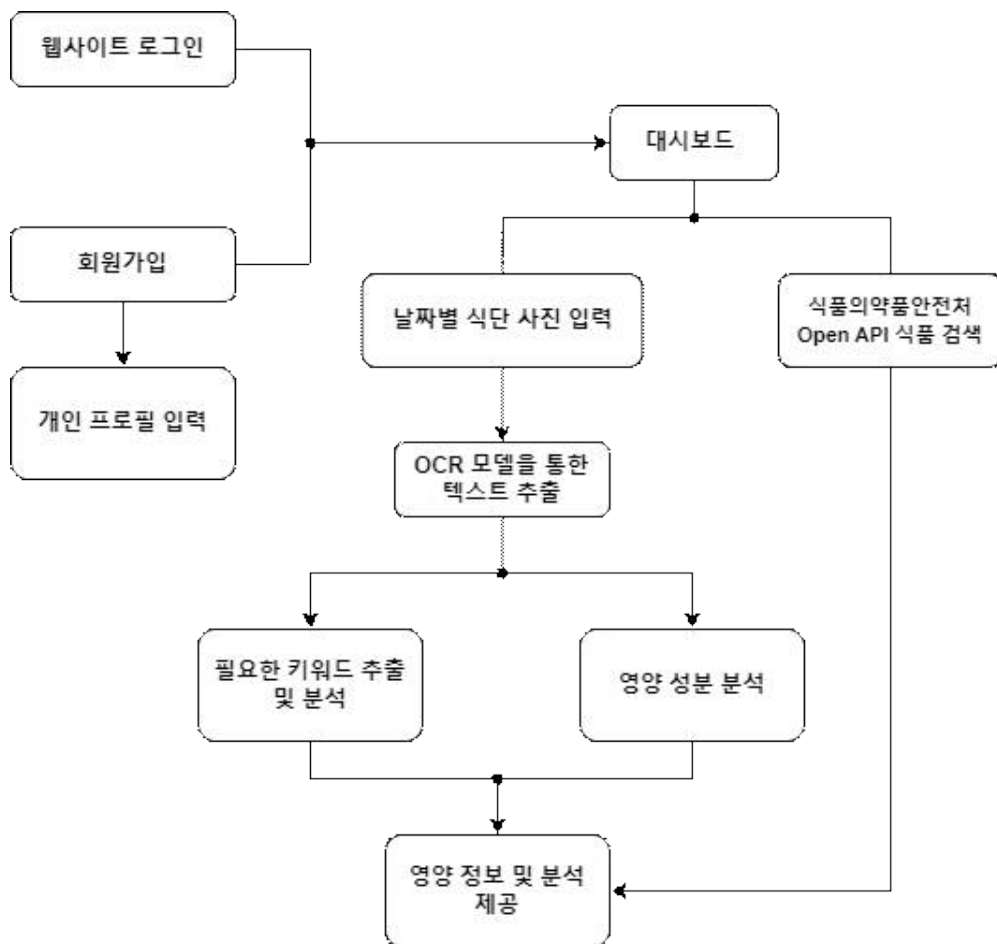
OCR 모델을 통한 식단 기록 및 건강관리 웹사이트의 목표는 사용자들의 건강한

식습관 형성을 지원하고, 개인화된 건강 정보를 제공하여 건강한 생활 방식을 도모하는 것이다. 이를 통해 예방적인 건강관리의 중요성을 강조하고, 사용자들이 스스로 건강한 삶을 추구할 수 있는 도구를 제공함으로써 건강한 사회를 만드는데 이바지할 것이다. 이 웹사이트는 또한 다양한 연령대와 직업군을 대상으로 하여, 보다 넓은 사용자층에게 건강한 식습관의 중요성을 전달할 수 있다.

2.3. Overview

2.3.1. 서비스 흐름

본 작품에서 제공하는 서비스의 전체적인 흐름은 다음 그림과 같다. 이후는 각 항목에 대한 세부 설명이다.



[그림 3] 서비스의 전체적인 흐름

① 회원 가입 및 로그인

이 웹사이트에서는 자체 서버에서 만들어진 REST API를 사용하여 회원 가입 및 로그인 기능을 구현한다. 회원 가입 과정에서 사용자는 개인 정보(키, 몸무게, 나이,

성별)와 운동량, 운동정보를 입력해야 한다. 이러한 정보를 바탕으로 시스템은 사용자의 기초대사량 및 운동대사량을 계산하여 데이터베이스에 저장한다

② 대시보드

대시보드는 캘린더 형태로 제공되며, 기본적으로 사용자가 섭취할 수 있는 운동대사량을 해당 날짜에 보여준다. 사용자가 원하는 날짜를 클릭하면 Modal이 표시된다. 이 Modal에서 사용자는 권장되는 영양 비율을 확인하고, 탄수화물, 단백질, 지방의 필요 섭취량을 파악할 수 있다. 또한, 사용자는 식단 영양 정보를 포함한 사진을 업로드하거나, 식품을 검색하여 식단에 추가할 수 있다. 추가된 식단을 기준으로 섭취한 칼로리도 함께 표시되어 사용자가 영양소 섭취 상태를 한 눈에 확인할 수 있다. 이를 통해 사용자가 건강한 식습관을 유지하고 개선할 수 있도록 도움을 제공한다.

③ OCR 모델을 통한 텍스트 추출

네이버 Clova OCR 모델을 사용하여 사용자가 업로드 한 식품 영양성분 사진에서 텍스트를 추출한다. OCR 모델에는 일반 텍스트 추출을 위한 General 모델과 표 형식의 데이터를 추출할 수 있는 Template 모델이 함께 적용되어 정확도를 높인다. 사용자는 자신이 섭취한 식품의 영양성분 사진을 업로드하면, OCR을 통해 추출된 텍스트가 화면에 표시된다. 영양성분 정보는 1회 제공량 당으로 제공되므로, 사용자가 제공량을 조절할 수 있는 기능도 제공된다. 이를 통해 사용자는 영양 정보를 쉽게 확인하고, 자신의 식단에 적합한 식품 선택을 도움 받을 수 있다.

④ Open API를 통한 추가 식품 검색

OCR을 통한 텍스트 추출이 만족스럽지 않거나 추가적인 식품 정보가 필요한 사용자를 위해, 식품의약품안전처에서 제공하는 Open API를 활용하여 식품의 영양성분을 조회할 수 있는 검색 화면을 제공한다. 사용자는 편리하게 원하는 식품의 영양 정보를 찾아볼 수 있으며, 동일하게 1회 제공량 당 영양성분이 제공된다. 이를 통해 사용자는 자신의 식단에 맞춘 영양 성분을 확인하고, 제공량을 조절하여 자신이 필요한 영양소를 섭취할 수 있도록 지원된다. 이 기능은 사용자가 다양한 식품 정보를 쉽게 얻을 수 있게 해 주며, 개인화된 식단 관리를 돕는다.

⑤ 영양 정보 및 분석 제공

사용자가 업로드 한 식단을 기반으로 총 칼로리를 제공해주는 것뿐만 아니라, 각

개인의 건강 상태와 운동 목적에 맞춰 적절한 3대 영양소(단백질, 탄수화물, 지방) 비율도 고려한다. 개인의 운동량과 운동 목적을 반영하여, 영양소별 필요한 칼로리 정보를 산출하고 사용자에게 제공한다. 이를 통해 사용자는 자신에게 맞는 균형 잡힌 식단을 구성하고, 올바른 영양 섭취를 추구할 수 있다. 또한, 이러한 진단을 통해 사용자는 자신의 식습관을 개선하고, 건강한 생활을 유지하는데 도움을 받을 수 있다.

2.3.2. 개발

① Front-end 개발

Front-end는 JavaScript와 Vue.js를 사용하여 개발하며, 서비스 흐름에 따라 사용자에게 보여지는 페이지를 구성한다. 페이지 개발 중 기능상 구현이 필요한 부분은 가능한 경우 오픈소스 라이브러리를 활용하여 효율성을 높인다. 로그인, 회원가입, 사진 업로드, 유저 정보 호출, 식품 검색 등 Back-end와의 통신이 필요한 경우에는 Rest API를 활용하여 원활한 데이터 교환을 가능하게 한다. 또한 사용자 경험(UX)을 고려하여 직관적이고 편리한 인터페이스를 제공함으로써, 이용자들이 서비스를 쉽게 이해하고 사용할 수 있도록 구성한다.

② Back-end 개발

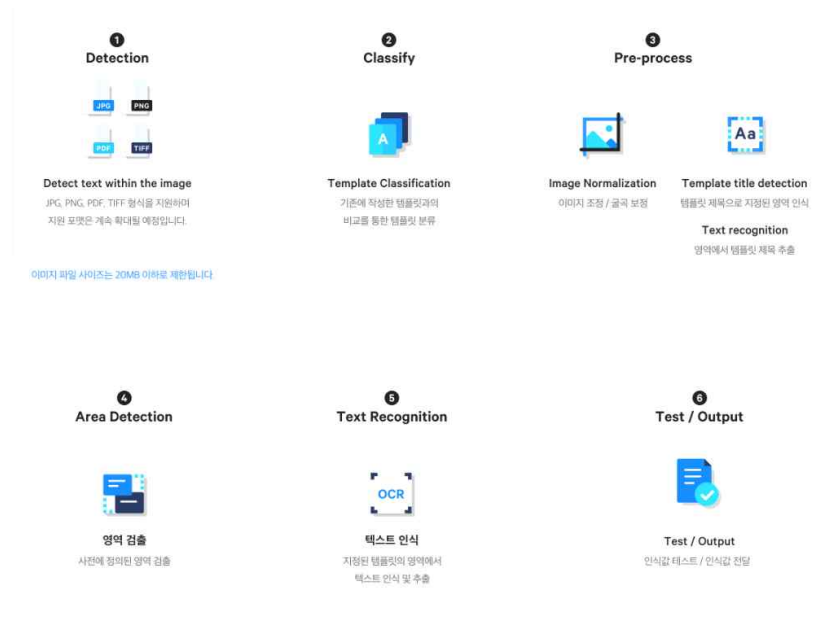
Back-end는 Front-end와의 원활한 통신을 위해 REST API를 개발하며, 웹 서버 구축에는 Node.js 기반의 Express.js를 사용하고, 데이터베이스로는 MySQL을 활용한다. 사용자가 업로드한 이미지를 base-64로 인코딩한 후 OCR API를 호출하는데, 이 과정은 Secret Key의 보안상 이유로 Back-end에서 처리한다. 또한, 사용자 정보와 식품 데이터를 관리하고 저장하기 위해 데이터베이스 스키마를 설계하며, 필요한 경우 데이터베이스 쿼리 최적화를 통해 성능 향상을 도모한다. 이와 함께, 서버 측 여러 처리 시스템을 구축하여 안정적인 서비스 운영을 지원한다.

3. 관련연구

3.1 OCR Model

OCR은 Optical Character Recognition의 약자로, 이미지에서 텍스트를 추출하는 기술을 의미한다. 대표적인 OCR 사용 사례 중 하나는 번호판을 인식해 텍스트로 변환하는 과속 단속 카메라이다.

본 프로젝트에서는 사용자가 자신이 섭취한 음식에 대한 사진을 업로드하면, 그 영양성분을 분석하고 기록하기 위해 네이버 Clova OCR 모델을 활용하여 기능을 구현하였다. 이를 통해 사용자가 손쉽게 음식의 영양 성분을 확인할 수 있으며, 시간과 노력을 절약할 수 있다. [1]네이버 Clova OCR 모델은 아래의 6단계의 과정을 거친다.



[그림 4] 네이버 Clova OCR 과정

네이버 Clova OCR 모델은 이미지 내의 문자를 인식하는 컨볼루션 신경망 (Convolutional Neural Network, CNN) 기반의 딥러닝 OCR 모델입니다. 이 모델은 관련 논문인 "End-to-end Text Recognition with Convolutional Neural Networks" 에서 제안된 방법을 활용하여 문자 인식을 수행합니다.

이 논문에서는 입력 이미지로부터 특징 추출, 문자 영역 검출, 문자 인식을 한 번에 수행하는 방법을 제시하며, 이를 위해 여러 레이어로 구성된 CNN 구조를 사용합니다. 이 방식은 기존의 방법들에 비해 높은 인식 정확도를 보이고, 일반화 능력이 뛰어납니다.

네이버 Clova OCR 모델은 이와 유사한 CNN 구조를 사용하며, 입력 이미지를 전처리한 후 특징 맵을 추출합니다. 이 과정에서 이미지의 밝기, 대비 등을 조정하고, 문자와 배경을 분리합니다. CNN을 통해 학습된 모델은 문자 영역을 찾고, 문자 단위로 분할하여 인식을 수행합니다. 추가적으로, LSTM(Long Short-Term Memory)을 활용하여 문자 인식 결과를 연결해 단어나 문장을 추론합니다.

또한, 네이버 Clova OCR 모델은 방대한 데이터셋을 통해 학습됩니다. 이 데이터셋은 대량의 이미지와 해당 이미지의 문자 정보로 구성되어 있어, 모델은 다양한 모양, 크기, 폰트 등의 문자를 학습하고 높은 인식 정확도를 달성합니다. 이를 통해 사용자에게 효율적이고 정확한 OCR 서비스를 제공하게 됩니다.

3.2. 기초 대사량과 운동대사량 및 필수 영양 성분

기초대사량(Basal metabolic rate, BMR)은 생명유지에 필요한 최소한의 에너지량을 나타내며, 환경 온도, 식사, 운동 등 외부 요인의 영향을 받지 않는 상태에서의 대사량을 의미합니다. 반면, 운동 대사량(Exercise metabolic rate, ERM)은 운동에 따라 소비되는 에너지량으로, 운동 강도, 종류, 지속시간 등에 따라 변동됩니다.

기초대사량을 예측하는 여러 공식들은 오랜 시간 동안 연구를 통해 발전해왔습니다. 대표적인 예로 해리스-베네딕 방정식, 뮐플린-세인트 지뢰르 방정식, 칼딩-테너 방정식 등이 있습니다. 이러한 공식들은 체중, 키, 나이, 성별 등의 변수를 활용하여 기초대사량을 추정합니다.

기초대사량과 운동 대사량을 결합하여 일일 에너지 소비량을 추정하는 방법은 개인의 신체 상태와 활동 수준에 따른 칼로리 섭취량을 계획하는 데 중요한 역할을 합니다. 이를 통해 개인의 건강한 체중 유지, 체중 감량, 혹은 체중 증가 목표에 맞게 올바른 식단과 운동 계획을 세울 수 있습니다.

[표 1] BMR 공식들의 종류

이름	년도	구분	식
Harris-Benedict 식	1919	남성	$66.5 + (13.75 \times \text{체중kg}) + (5.003 \times \text{키cm}) - (6.75 \times \text{나이})$
		여성	$655.1 + (9.563 \times \text{체중kg}) + (1.850 \times \text{키cm}) - (4.676 \times \text{나이})$
개 정 된 Harris-Benedict 식	1984	남성	$88.4 + (13.4 \times \text{체중kg}) + (4.8 \times \text{키cm}) - (5.68 \times \text{나이})$
		여성	$447.6 + (9.25 \times \text{체중kg}) + (3.1 \times \text{키cm}) - (4.33 \times \text{나이})$
Mifflin-St Jeor 식	1990	기본식	$(10 \times \text{체중kg}) + (6.25 \times \text{키cm}) - (5 \times \text{나이})$
		남성	기본식 + 5
		여성	기본식 - 161

기초대사량에 맞는 탄단지비율은 개인의 목적, 건강 상태, 생활 환경 등에 따라 다르기 때문에 일반적인 기준은 존재하지 않습니다. 그러나 일반적으로 권장되는 탄단지비율은 다음과 같습니다.

[2] American Heart Association:

- 탄수화물: 총 열량의 45 ~ 65%
- 단백질: 총 열량의 10 ~ 35%
- 지방: 총 열량의 20 ~ 35%

[3] 세계보건기구 (World Health Organization):

- 탄수화물: 총 열량의 55 ~ 75%
- 단백질: 총 열량의 10 ~ 15%
- 지방: 총 열량의 15 ~ 30%

이러한 기준은 일반적인 건강한 성인을 대상으로 한 것이며, 개인의 건강 상태나 목적에 따라 조절될 수 있습니다. 또한, 최근에는 탄수화물 중에서도 저지방 고섬유 다이어트에 따른 고단백 탄수화물에 더욱 집중하는 추세입니다. 지방은 단일 불포화 지방과 다중 불포화 지방으로 구분하여 섭취량을 조절하는 것이 중요하게 강조되고 있습니다. 이를 통해 심혈관 질환의 위험을 낮추고, 전반적인 건강을 유지하는데 도움이 됩니다.

3.3. REST API

본 작품을 구현하는 과정에서 로그인 및 OCR 모델을 이용한 사진 분석과 같은 Front-end와 Back-end 간의 소통은 REST API를 활용하여 개발했습니다. REST API는 로이 필딩(Roy Fielding)이 제안한 것으로, 웹의 장점을 최대한 활용할 수 있는 구조로 현재도 널리 사용되고 있습니다. REST의 주요 특징은 다음과 같이 4가지로 나열할 수 있습니다.

- **Stateless (무상태성):** 각 작업을 수행하기 위해 상태 정보를 별도로 저장하거나 관리하지 않습니다. 이 때문에 세션, 쿠키 정보를 따로 저장 및 관리하지 않아 API 서버는 들어오는 요청만 처리할 수 있습니다.
- **Cacheable (캐시 가능):** HTTP의 기존 웹 표준을 사용함으로써, 웹에서의 기존 인프라를 그대로 활용할 수 있습니다. 이를 통해 HTTP가 가진 캐싱 기능을 사용할 수 있습니다.
- **Self-descriptiveness (자기 기술성):** REST API 메시지만 보고도 어떤 행위를 하는지 쉽게 이해할 수 있는 구조를 가지고 있습니다.
- **Uniform (일관성):** URI로 지정한 리소스에 대한 조작을 통일되고 제한된 인터페이스로 수행하는 아키텍처입니다.

이러한 특징을 유지하기 위해 REST API를 설계할 때 고려해야 할 중요한 항목이 있습니다. 첫 번째로는 URI에는 정보의 자원을 표시해야 한다는 것입니다. 예를 들어, 회원 정보를 가져오는 API를 생성할 때는 'member'와 같은 자원의 내용을 명시해야 합니다. 두 번째로는 자원에 대한 행위는 HTTP Method (예: GET, POST, PUT, DELETE 등)로만 표현하는 것입니다. 이를 통해 API 설계가 명확하고 일관성 있는 구조를 갖출 수 있습니다.

[표 2] 각 REST API Method의 역할 및 특징

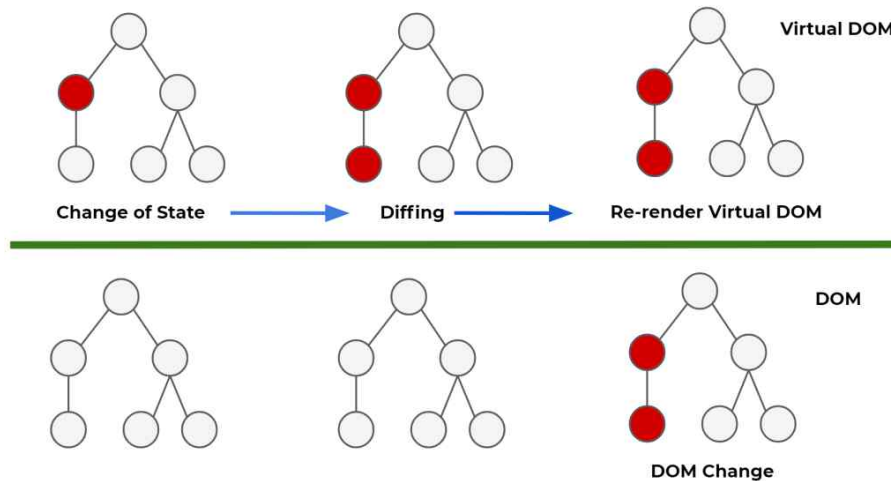
Method 종류	역할	Body, Content-Type 작성 유무	URL을 통한 Parameter 전송
GET	데이터 조회	X	O
POST	데이터 생성	O	X
PUT	데이터 수정	O	O
DELETE	데이터 삭제	X	O

3.4. Vue.js

Vue.js는 사용자 인터페이스를 구축하기 위한 JavaScript 프레임워크로서 HTML, CSS 및 JavaScript를 기반으로 구축되며, 가벼움과 유연성, 뛰어난 성능으로 인기 있는 자바스크립트 프레임워크 중 하나입니다. Vue.js의 주요 장점은 다음과 같습니다.

- 유연성: Vue.js는 유연한 아키텍처를 가지고 있어 다른 자바스크립트 라이브러리나 프레임워크와 쉽게 통합할 수 있습니다. 컴포넌트 기반 아키텍처를 사용함으로써 코드의 재사용성이 높아지고, 프로젝트의 유지 보수가 용이해집니다.
- 성능: Vue.js는 가벼우면서도 빠른 성능을 제공합니다. 이는 가상 DOM(Virtual DOM)을 사용하여 렌더링 성능을 향상시키고, 리액티브 데이터 바인딩으로 더욱 빠른 반응성을 가능하게 합니다. 기본적으로 번들 크기가 작아 코드를 더 빠르게 다운로드하고 실행할 수 있습니다.
- 커뮤니티: Vue.js는 활발하게 성장하고 있는 커뮤니티를 가지고 있어 개발자가 Vue.js를 사용하여 개발하는 데 도움이 됩니다. 이 커뮤니티는 많은 자료, 튜토리얼, 예제 프로젝트 등을 제공하므로, 학습 과정이 더욱 원활해집니다.
- 진입 장벽 낮음: Vue.js는 초기 학습 곡선이 낮아 개발자들이 빠르게 프로젝트에 적용할 수 있습니다. 문서화가 잘 되어 있고, 다양한 플러그인 및 확장 기능을 제공하여 사용자가 필요한 기능을 쉽게 구현할 수 있게 합니다.

3.4.1. 가상 DOM



[그림 5] 가상 DOM을 통한 실제 DOM의 업데이트

HTML 문서의 각 요소를 계층적으로 표현하여 생성, 변형, 삭제할 수 있도록 돕는 인터페이스인 DOM (Document Object Model)은 트리 구조를 가지고 있기 때문에 부모 노드에서 발생한 수정 사항이 자손 노드들에 영향을 주고, 결과적으로 UI 반응성이나 사용자 경험에 부정적인 영향을 준다. 이 문제를 해결하기 위해 React는 사용자와 실제 DOM 사이에 가상 DOM을 구현하는 방법으로 접근했다.

가상 DOM은 실제 DOM의 복사본으로 브라우저 등 실제 화면에 보이지 않고 메모리 상에만 존재하는 객체이다. 사용자의 조작에 의해 가상 DOM에 발생한 변화는 곧바로 실제 DOM에 적용되지 않고, 효율적인 알고리즘을 통해 실제 DOM에 어떤 변화를 일으켜야 할지 계산한다. 그 후, 다른 컴포넌트에 영향을 주지 않고 실제 변화가 발생한 DOM이 포함된 컴포넌트만 업데이트되도록 한다. 이를 더불어 'diffing'과정이라고도 부른다.

이러한 선택적 업데이트는 비교적 적은 연산과 시간을 소모하여 성능 향상에 기여하며, Front-end의 발전에 따라 복잡하고 동적인 페이지를 만드는 데 매우 유리한 특징으로 간주된다. 더불어, 가상 DOM을 통해 UI 렌더링 과정에서의 성능 저하를 최소화하고, 웹 애플리케이션의 빠른 반응성과 사용자 경험을 향상시키는 데 큰 역할을 한다.

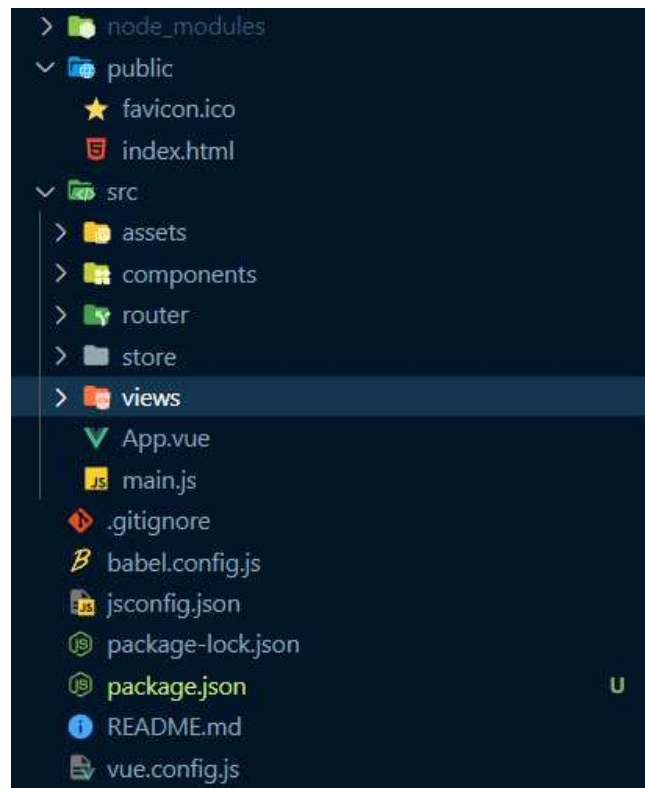
4. 제안 작품 소개

4.1 Front-end

4.1.1 개발환경 구성

일반적으로 Vue 프로젝트를 개발하고 실행하기 위해서는 Babel, Webpack 등의

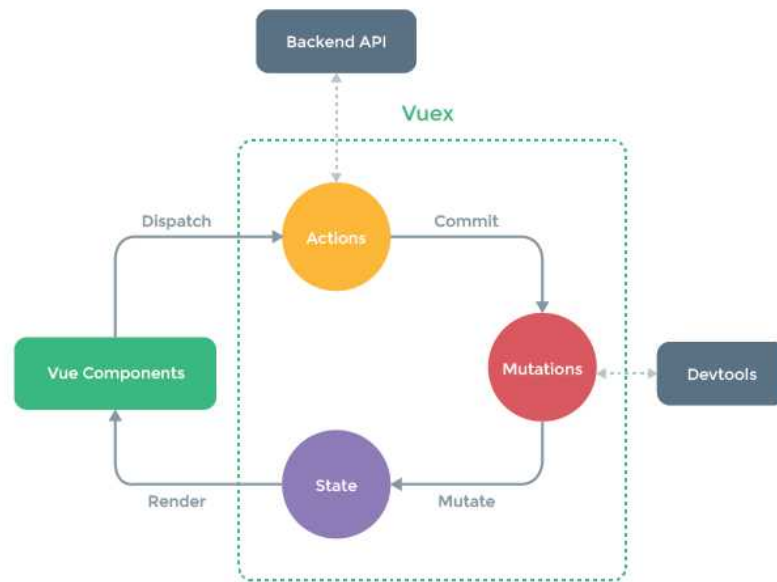
복잡한 설정이 필요하다. 이러한 과정이 Vue 프로젝트 시작에 어려움을 초래하기 때문에, 빠르고 간편하게 Vue 프로젝트를 시작할 수 있는 Vue-CLI라는 도구가 제공되고 있다. Vue-CLI는 Babel, Webpack, HMR(Hot Module Replacement), ES6+ 문법 지원, CSS 후처리 등 Vue 개발에 필수적인 환경을 구축해주며, 기존 기능 개선이나 새로운 기능 추가 시에는 Vue-CLI의 패키지 버전만 업데이트하면 되어 관리가 용이하다. 이로 인해 초기 설정 및 개발에 소요되는 시간을 절약할 수 있다. Vue-CLI를 활용하여 구성한 Vue 프로젝트의 구조는 아래 그림과 같다.



[그림 6] Vue-CLI 기본 구조

또한, 추가적인 구현의 편의성을 위해 스타일 라이브러리인 Vuetify를 CSS 파일 형태로 분리하여 전역에서 사용할 수 있도록 저장하였다. 이렇게 함으로써 프로젝트 전반에 걸쳐 일관된 디자인과 레이아웃을 적용할 수 있다.

또한, 싱글 페이지 애플리케이션(SPA)인 Vue에서 데이터를 효과적으로 관리하기 위해 상태 관리 라이브러리인 [4] Vuex를 사용하였다. Vuex는 컴포넌트 간의 데이터 공유와 상태 관리를 효율적으로 처리할 수 있게 도와주어, 프로젝트의 유지보수 및 확장성을 향상시키는 데 도움을 준다.



[그림 7] Vuex의 상태 관리 방법

4.1.2 Front-end 구조

Vue에서 컴포넌트(Component)란 애플리케이션을 구성하는 기본 단위로, 재사용 가능한 UI 요소를 의미한다. MVC (Model-View-Controller) 패턴과 달리, 의존성이 낮아 재활용이 쉽다는 장점이 있다. 이는 방향 데이터 바인딩을 사용하여 프로토타입과 뷰 구성 요소 간의 동적 조정을 가능하게 하는 ViewModel을 구성함으로써 가능하다.

본 작품 역시 컴포넌트를 통해 UI를 재사용 가능한 조각들로 나누어 개발하였다. 최상단 컴포넌트인 App이 존재하며, 이 App 컴포넌트는 각각의 라우팅에 따라 적절한 View를 보여주는 역할을 한다. View는 화면 구성을 위한 다양한 컴포넌트들로 구성되어 있으며, 하나의 화면이나 장면을 나타낸다.

View 컴포넌트는 다양한 하위 컴포넌트들을 포함하여 사용자에게 보여질 화면을 구성한다. 이러한 구조는 UI의 재사용성을 높이고, 코드의 가독성과 유지보수성을 향상시킨다. 또한, 이를 통해 개발자들은 각 컴포넌트를 독립적으로 관리하고 테스트할 수 있어, 전체 애플리케이션의 확장성과 안정성이 높아진다.

- **App:** App 컴포넌트는 최상단 컴포넌트로서, 자식 컴포넌트로 View를 포함하며, 이 View 컴포넌트는 Login, Feature, Calendar와 같은 다양한 하위 컴포넌트들로 구성되어 있다. 각 컴포넌트들에서는 스코프된 형태로 Style 태그를 사용하여 독립적인 스타일링이 가능하다. 그러나 App 컴포넌트는 모든 컴포넌트의 최상단에 위치하기 때문에, 전역적으로 사용할 CSS 파일들을 import하여 일관된 스타일 및 레이아웃

웃을 적용할 수 있다.

이러한 구조를 통해 각 컴포넌트의 스타일이 서로 간섭하지 않고, 전체 애플리케이션에 일관된 디자인을 적용할 수 있게 된다. 또한, App 컴포넌트를 통해 전역 스타일을 쉽게 관리하고 필요한 경우 수정할 수 있어, 프로젝트의 유지보수와 확장성이 향상된다. 이렇게 최상단 컴포넌트인 App을 통해 프로젝트의 구조와 스타일링을 체계적으로 관리할 수 있다.

- **Login:** 로그인 및 회원가입을 처리하는 컴포넌트로, 하위 컴포넌트로 회원가입 Modal인 Register 컴포넌트를 포함한다. Feature 컴포넌트의 헤더에서 회원가입 버튼을 클릭하면, Register 컴포넌트가 활성화되어 화면에 표시된다. 사용자는 개인 프로필 정보를 입력한 후 회원가입을 진행할 수 있다.

회원가입 과정에서 Back-end로 POST 요청을 보내 데이터를 저장하게 되고, Back-end는 생성된 사용자의 고유 ID인 UUID를 반환한다. 이후 사용자가 로그인을 완료하면, UUID를 라우팅 주소에 포함하여 Calendar 화면으로 진입하게 된다. 이렇게 구성된 로그인 및 회원가입 컴포넌트는 사용자 인증 및 회원 정보 관리를 체계적으로 처리할 수 있으며, 사용자 경험을 향상시키는 데 도움이 된다.

- **Feature:** 첫 화면은 사용자가 서비스를 처음 접할 때 만나는 화면으로, 해당 서비스에 대한 설명을 제공하여 사용자들이 서비스를 이해하는 데 도움을 준다. 이 화면은 하위 컴포넌트로 Header를 포함하고 있다.

Header 컴포넌트는 로그인과 회원가입 버튼을 가지고 있으며, 사용자가 로그인 버튼을 클릭하면 Login 화면으로 이동한다. 반면에 회원가입 버튼을 클릭할 경우, 역시 Login 화면으로 이동하지만 이때는 회원가입 모달이 활성화된 상태로 진입하게 된다. 이렇게 구성함으로써 사용자에게 직관적이고 편리한 인터페이스를 제공할 수 있다. 이처럼 첫 화면의 구성은 사용자들이 서비스의 목적과 기능을 이해하고, 원활하게 회원가입 및 로그인 과정을 진행할 수 있도록 돕는다.

- **Calendar:** Calendar 컴포넌트는 크게 Header와 Calendar 두 가지 하위 컴포넌트를 가지고 있으며, 각 날짜를 선택하면 해당 날짜의 정보를 보여주는 Modal 형태의 하위 컴포넌트인 Day를 포함하고 있다. 사용자가 로그인을 완료하고 반환된 UUID를 사용하여, 선택한 날짜를 기준으로 해당 월 전체의 칼로리 데이터를 조회할 수 있다. 사용자가 상세한 날짜 정보를 확인하고 싶을 경우, 해당 날짜를 클릭하면 Back-end에서 세부 데이터를 조회하는 Modal이 표시된다. Day Modal은 Main Calendar View에서 보여지는 전체 칼로리 정보와 함께 각 영양소 별 필요 칼로리를 보여준다. 사용자는 사진을 업로드하여 OCR 모델을 통해 텍스트를 추출할 수 있고, 검색 버튼을 통해 오픈 API를 사용하여 필요한 식품의 영양 정보를 받아올 수 있다.

영양 정보는 기본적으로 1회 섭취량 당 데이터로 제공되며, 사용자는 섭취한 양을 입력한 후 저장할 수 있다.

4.1.3 Vuex 상태 관리

본 작품은 Vue의 상태관리 라이브러리인 Vuex를 사용하여 개발하였다. Vuex는 Flux 아키텍처에서 영감을 받아 개발되었으며 크고 복잡한 SPA(Single Page Application)에서 다양한 컴포넌트 사이의 상태를 효과적으로 관리하고 추적할 수 있도록 도와준다.

본 작품에서는 calendar, external, users 라는 이름의 Module 형태로 나누어 각각의 State, Getters, Mutations를 가진다.

- **State:** 애플리케이션의 중앙 저장소로, 모든 컴포넌트가 공유하는 데이터를 저장합니다. 단일 소스로 데이터를 관리하여 일관성 및 데이터 무결성을 유지할 수 있습니다. 컴포넌트에서는 Vuex의 state를 사용하여 데이터를 읽을 수 있습니다.

본 작품에서는 여러 컴포넌트에서 사용되는 데이터에 대해 저장하며 Mutations를 통해 저장되고 Getters를 통해 호출하는 방식으로 사용되고 있다.

- **Getters:** State에서 데이터를 가져와 계산하는 데 사용되는 메서드입니다. Getters는 Vue 컴포넌트의 computed 속성처럼 작동하며, 캐싱된 결과를 반환합니다. 이를 통해 성능이 개선되고 중복 코드를 줄일 수 있습니다. 컴포넌트에서는 Vuex의 getters를 사용하여 필요한 계산이나 필터링이 적용된 데이터를 가져올 수 있습니다. 본 작품에서는 State에 저장된 값을 각각의 Components에서 호출 할 때 사용되고 있습니다.

- **Mutations:** Mutations는 Vuex의 state를 변경하는 함수입니다. Mutations은 상태 변경에 대한 기록을 남기기 때문에 디버깅이 용이하며, 예측 가능한 상태 변경이 가능합니다. Mutations는 동기적으로 실행되어야 하며, 컴포넌트에서는 일반적으로 Mutations를 직접 호출하지 않고 Actions를 통해 호출합니다. 이렇게 함으로써 상태 변경의 로직을 분리하고 명확하게 관리할 수 있습니다. 본 작품에서는 Actions를 통해 API 호출 후 도출된 결과를 수정하거나 필요한 정보만 State에 저장하기 위해 사용됩니다.

- **Actions:** Actions는 비동기 작업이나 여러 Mutations를 연속해서 호출할 수 있는 메서드입니다. 컴포넌트에서는 주로 Actions를 호출하여 상태를 변경하며, 이때 Actions는 필요한 비동기 작업을 수행한 후, 결과를 Mutations에 전달하여 상태를 변경합니다. 본 작품에서는 Actions에서 API Instance를 호출해 Rest API를 비동기적으로 호출합니다.

4.2 Back-end

① 회원가입 및 로그인 API

- **URL:** /users/login

Method: GET

Description: 실제 유저가 email 과 password를 입력하고 로그인을 시도했을 때 Database에서 email과 password에 해당하는 userUuid를 반환한다.

Request

Name	Type	Description	Required
email	String	User의 로그인시 입력하는 email	O
password	String	User의 로그인시 입력하는 password	O

Response

Name	Type	Description	Required
userUuid	String	User 고유의 uid 값	O

- **URL:** /users

Method: GET

Description: 로그인 시도 후 반환된 userUuid를 기반으로 Database에 저장된 유저정보를 반환한다.

Request

Name	Type	Description	Required
userUuid	String	User 고유의 uid 값	O

Response

Name	Type	Description	Required
email	String	User의 로그인시 입력하는 email	O
password	String	User의 로그인시 입력하는 password	O
name	String	회원 가입시 입력한 User의 실제 이름	O
nickname	String	회원 가입시 입력한 User의 닉네임	O
weight	String	회원 가입시 입력한 User의 몸무게	O
height	String	회원 가입시 입력한 User의 키	O
age	String	회원 가입시 입력한 User의 나이(만)	O
sex	String	회원 가입시 입력한 User의 성별	O
BMR	Number	입력정보를 토대로 계산한 기초대사량	O
purpose	String	회원 가입시 입력한 User의 운동 목적	O
exercise	Number	회원 가입시 입력한 User 운동량	O

- **URL:** /users

Method: POST

Description: 회원가입 시 사용되는 API로 입력된 정보로 Database에 유저정보를 등록한다.

Request

Name	Type	Description	Required
email	String	User의 로그인시 입력하는 email	O
password	String	User의 로그인시 입력하는 password	O
name	String	회원 가입시 입력한 User의 실제 이름	O
nickname	String	회원 가입시 입력한 User의 닉네임	O
weight	String	회원 가입시 입력한 User의 몸무게	O
height	String	회원 가입시 입력한 User의 키	O
age	String	회원 가입시 입력한 User의 나이(만)	O
sex	String	회원 가입시 입력한 User의 성별	O
BMR	Number	입력정보를 토대로 계산한 기초대사량	O
purpose	String	회원 가입시 입력한 User의 운동 목적	O
exercise	Number	회원 가입시 입력한 User 운동량	O

Response

Name	Type	Description	Required
Success	Boolean	API 호출 결과에 대한 Boolean	O

② **OCR 모델 및 날짜별 칼로리 데이터 API**

- **URL:** /calendar/calories

Method: GET

Description: 로그인 이후 진입하는 Dashboard 즉 Calendar 화면에서 해당 달의 각 날짜에 총 얼마의 칼로리를 섭취했는지 제공하는 API로 Database에서 해당 유저의 uuid와 날짜를 기반으로 칼로리 데이터를 제공한다.

Request

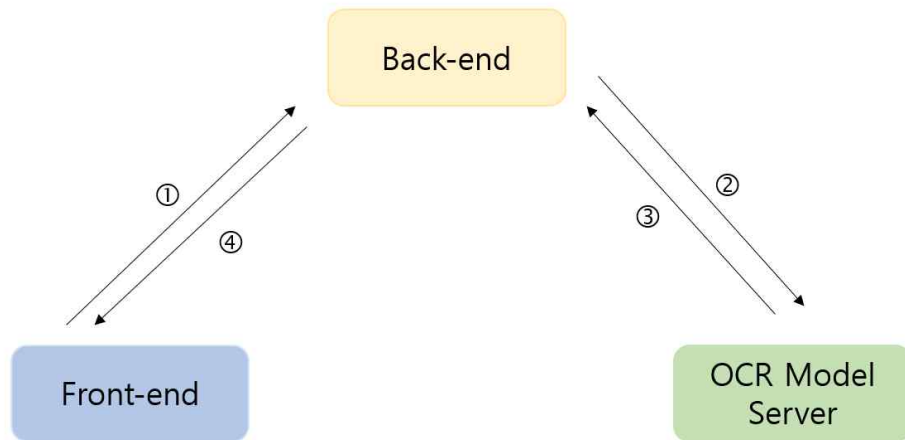
Name	Type	Description	Required
uuid	String	User 고유의 uid 값	O
searchDate	date	조회 하고자하는 날짜	O

Response

Name	Type	Description	Required
fat	String	날짜별 총 섭취한 지방량	○
carbohydrate	Number	날짜별 총 섭취한 탄수화물량	○
protein	Number	날짜별 총 섭취한 단백질량	○
calories	Number	날짜별 총 섭취한 칼로리	○
day	date	조회 한 날짜	○

- **URL:** /calendar/ocr

Method: POST



[그림 8] OCR 텍스트 추출 과정

Description:

①, ② : 유저가 식단 영양성분을 이미지로 업로드하면 그 이미지를 Base64형태로 encoding 한 후 Naver OCR API에 인자로서 전달한다.

③, ④ : 이미지에서 추출된 텍스트가 결과 값으로 전달된다. Back-end에서 해당 정보들 중 필요한 정보들만 정제한 뒤 데이터베이스에 저장 후 Front-end에 반환 값으로 전달한다.

- **URL:** /calendar/diet

Method: GET, POST, PUT, DELETE

Description: OCR 텍스트 추출이나 검색을 통해 얻은 결과를 저장, 수정, 삭제 등 변환하여 사용하기 위해 호출되는 API이다.

- **GET:** 사용자가 특정 날짜를 클릭했을 때 보여주는 정보를 조회하기 위해 사용된다.

- **POST:** 사용자가 OCR이나 검색을 통해 얻은 데이터를 저장할 때 사용되며 각

개별 데이터를 저장하기 전 전체 칼로리를 계산해 저장 후 각각의 데이터를 저장한다.

- **PUT:** 대부분의 영양성분은 1회 제공량 당 정보로 주어지기에 사용자가 먹은 만큼의 수량을 수정할 때 사용하는 API로 수정 후 이전 수량과 차이를 계산하여 총 칼로리를 계산해 저장 후 각각의 데이터를 수정한다.

- **DELETE:** 사용자가 자신이 먹은 음식이나 혹은 잘못된 결과를 삭제할 때 사용하는 API로 저장된 총 칼로리에서 해당 데이터의 칼로리와 수량을 계산하여 저장 후 해당 음식 데이터를 삭제한다.

Request

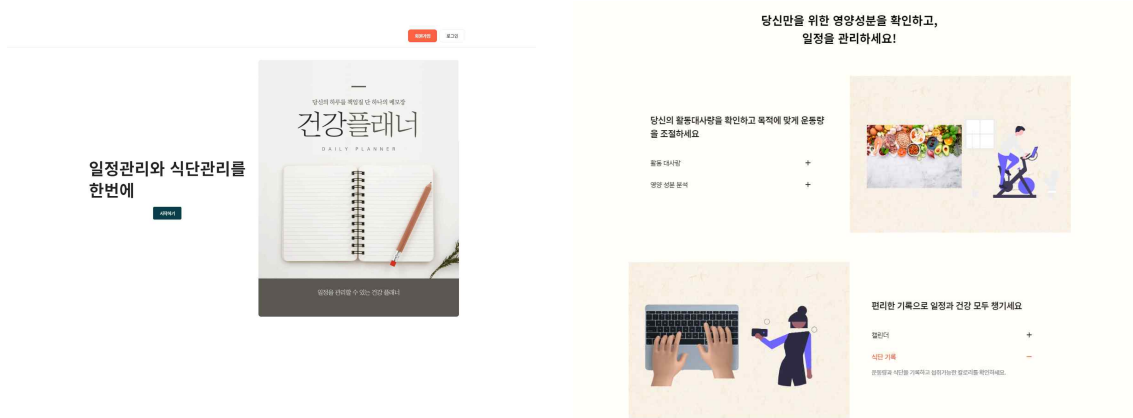
Method	Name	Type	Description
Get	uuid	String	User 고유의 uid 값
	date	date	조회 하고자 하는 날짜
Put	dietUuid	String	User가 입력한 고유의 식단 uid값
	count	Number	변경하고자 하는 수량 값
Delete	uuid	String	User 고유의 uid 값
	dietUuid	String	User가 입력한 고유의 식단 uid값
Post	uuid	String	User 고유의 uid 값
	productName	String	식단 이름
	carbohydrate	Number	1회 제공량 당 탄수화물
	protein	Number	1회 제공량 당 단백질
	fat	Number	1회 제공량 당 지방
	eachCalories	Number	1회 제공량 당 칼로리
	servingWT	Number	1회 제공량 (g)

Response

Method	Name	Type	Description
Get,Put,Delete	Success	Boolean	API 호출 결과에 대한 Boolean
Post	productName	String	조회한 식단 이름
	carbohydrate	Number	조회한 식단의 탄수화물 량
	protein	Number	조회한 식단의 단백질 량
	eachCalories	Number	조회한 식단의 칼로리
	servingWT	Number	조회한 식단의 1회 제공 량
	count	Number	식단의 섭취량을 보여줌. Default 1 삽입

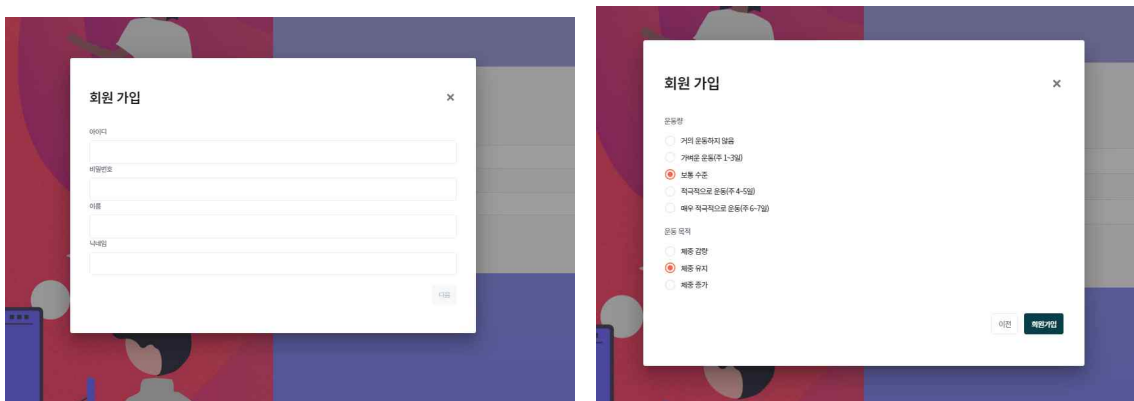
5. 구현 및 결과분석

5.1 구현

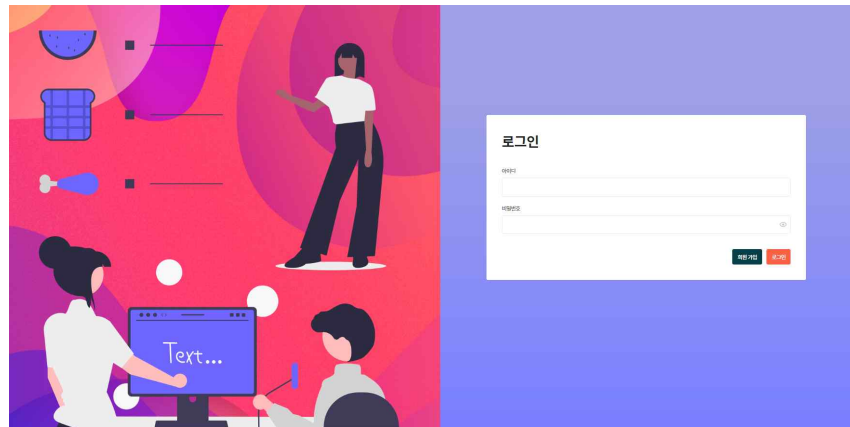


[그림 9] 웹사이트 메인 화면

웹사이트에 처음 접속하면 나타나는 화면이다. 메인 화면의 경우 간단하게 웹사이트에 대한 소개가 적혀있다. 유저는 시작하기 버튼을 누르면 로그인 화면으로 이동할 것이고 상단 헤더의 회원가입 버튼을 누르면 아래의 회원가입 Modal이 뜬 상태로 로그인 페이지로 접속하게 된다. Modal에서 아이디, 비밀번호, 키, 몸무게 등 개인 정보와 함께 그림과 같이 운동량과 운동 목적을 선택하고 회원가입이 진행된다. 로그인 버튼을 누르면 로그인 페이지로 접속된다.



[그림 10] 회원 가입 화면



[그림 11] 로그인 화면

위 화면에서 아이디와 패스워드를 입력하여 접속하면, 첫 화면으로 아래의 캘린더 페이지가 나타난다. 이전에 회원가입 시 입력한 정보를 바탕으로, 하루에 섭취 가능한 운동대사량이 계산되어 각 날짜에 표시된다. 또한 날짜별로 사용자가 섭취한 총 칼로리량도 계산되어 캘린더에 나타난다.

[회원가입](#)
[로그아웃](#)

3월 2023						
일	월	화	수	목	금	토
			1 섭취량: 0 kcal 목표 대사량: 2435 kcal	2 섭취량: 0 kcal 목표 대사량: 2435 kcal	3 섭취량: 0 kcal 목표 대사량: 2435 kcal	4 섭취량: 0 kcal 목표 대사량: 2435 kcal
5 섭취량: 0 kcal 목표 대사량: 2435 kcal	6 섭취량: 0 kcal 목표 대사량: 2435 kcal	7 섭취량: 0 kcal 목표 대사량: 2435 kcal	8 섭취량: 0 kcal 목표 대사량: 2435 kcal	9 섭취량: 0 kcal 목표 대사량: 2435 kcal	10 섭취량: 0 kcal 목표 대사량: 2435 kcal	11 섭취량: 0 kcal 목표 대사량: 2435 kcal
12 섭취량: 0 kcal 목표 대사량: 2435 kcal	13 섭취량: 0 kcal 목표 대사량: 2435 kcal	14 섭취량: 0 kcal 목표 대사량: 2435 kcal	15 섭취량: 0 kcal 목표 대사량: 2435 kcal	16 섭취량: 0 kcal 목표 대사량: 2435 kcal	17 섭취량: 0 kcal 목표 대사량: 2435 kcal	18 섭취량: 0 kcal 목표 대사량: 2435 kcal
19 섭취량: 0 kcal 목표 대사량: 2435 kcal	20 섭취량: 0 kcal 목표 대사량: 2435 kcal	21 섭취량: 0 kcal 목표 대사량: 2435 kcal	22 섭취량: 0 kcal 목표 대사량: 2435 kcal	23 섭취량: 0 kcal 목표 대사량: 2435 kcal	24 섭취량: 0 kcal 목표 대사량: 2435 kcal	25 섭취량: 0 kcal 목표 대사량: 2435 kcal
26 섭취량: 0 kcal 목표 대사량: 2435 kcal	27 섭취량: 0 kcal 목표 대사량: 2435 kcal	28 섭취량: 0 kcal 목표 대사량: 2435 kcal	29 섭취량: 0 kcal 목표 대사량: 2435 kcal	30 섭취량: 0 kcal 목표 대사량: 2435 kcal	31 섭취량: 0 kcal 목표 대사량: 2435 kcal	

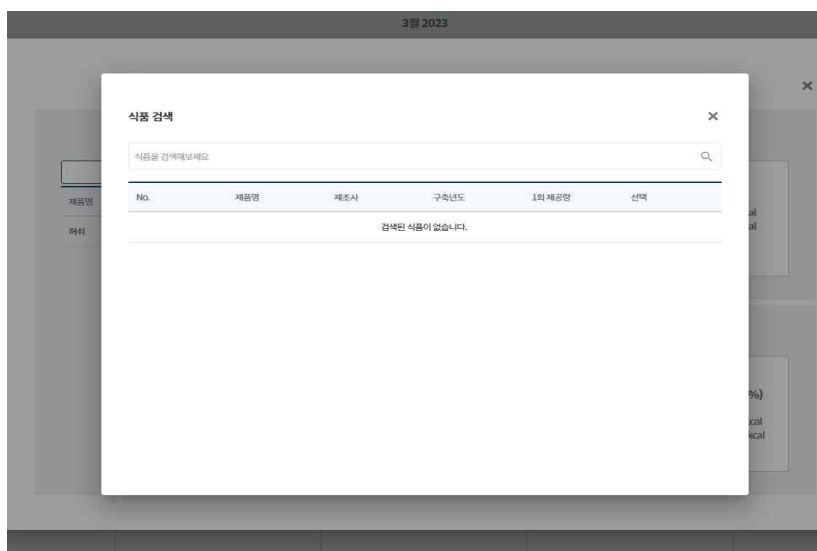
[그림 12] 캘린더 화면

캘린더 화면의 각 날짜는 클릭이 가능하면 클릭하게 되면 [그림 13] 화면이 나타난다. 상단 헤더에는 닉네임과 로그아웃 버튼이 존재한다. 배경 색으로 오늘의 날짜를 알 수 있으며 화살표를 통해 다른 달의 정보도 볼 수 있다.



[그림 13] 날짜별 Modal 화면

위 그림은 캘린더의 각 날짜를 클릭했을 때 나타나는 창으로 사용자가 자신이 먹은 식품의 사진을 올리거나 검색을 할 수 있다. +버튼을 통해 사진을 올리면 OCR 텍스트 추출이 진행되어 식품이 추가되고 각 영양성분 별 칼로리와 총 섭취량 역시 날짜별로 화면에 표시된다. 식품의 경우 1회 제공량 당으로 되기에 수정 버튼을 통해 수량과 이름을 수정할 수 있다. 잘못된 정보나 추출이 있는 경우 삭제가 가능하고 아래 [그림 14] 화면에서 볼 수 있듯이 검색 버튼을 통해 Open API로 식품을 검색하고 추가 할 수 있으며 해당 방식으로 추가한 식품 역시 수정, 삭제가 가능하다.



[그림 14] 검색 Modal 화면

5.2 결과 분석

5.2.1 OCR 텍스트 추출

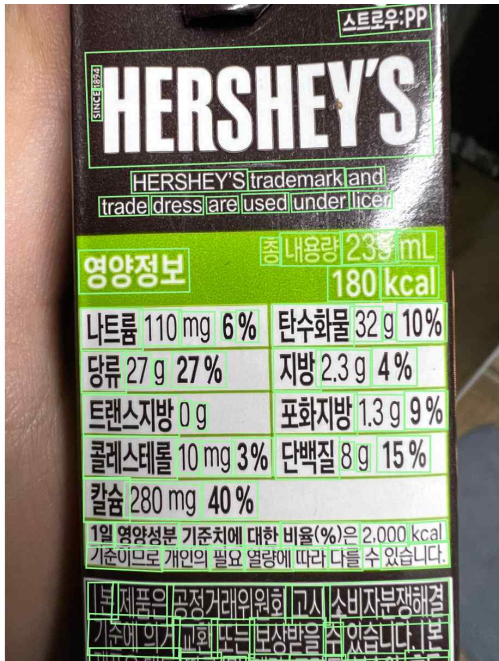
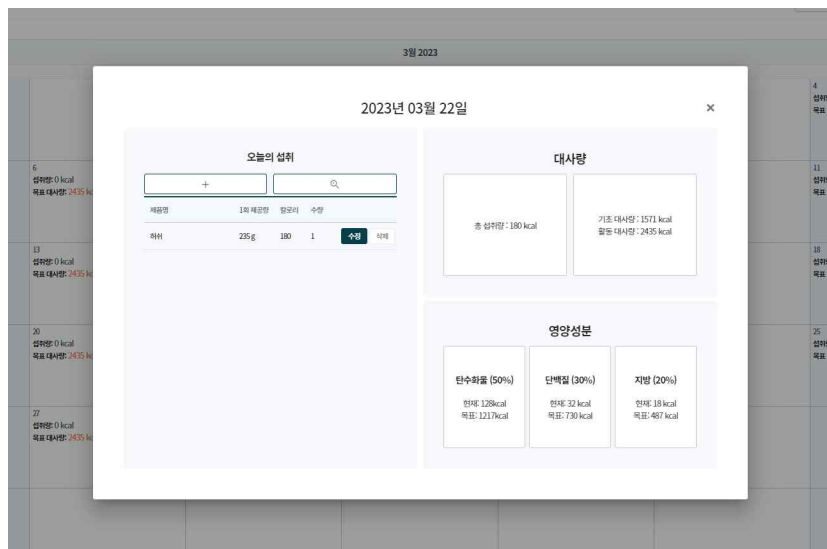


Table 02

기능성표시	
기능성 성분함량 (총 내용량 당)	총다당제 함량으로서 198 mg (기타원료 유래 다당제 11,603 mg)
1일 섭취 기준량	총다당제 함량으로서 100~420 mg
섭취시 주의사항	균형 잡힌 식생활을 권장합니다.
이상사례가 있는 경우 섭취를 중단하고 전문가와 상담이 필요합니다. 본 제품은 질병의 예방치료를 위한 제품이 아닙니다.	
ml(25 m/x36회) 영양정보 총 내용량 900 1회(25 ml) 당 25 kcal	
1회 당 1일 영양성분 기준치에 대한 비율	100ml당
나트륨 mg 0 %	0 mg 0%
탄수화물 g 2 %	23 g 7%
당류 g 3 %	12 g 12%
지방 g 0 %	0 g 0%
트랜스지방 g 0 %	0 g 0%
포화지방 g 0 %	0 g 0%
콜레스테롤	0 %

[그림 15] 올바른 추출 예시

위 그림과 같이 영양정보를 OCR을 통해 텍스트 추출을 진행한다. 기본적으로 표 모양을 인식하는 옵션까지 가지고 있기 때문에 주로 표 형태로 주어지는 영양정보를 잘 인식하여 나타내는 것을 볼 수 있다. 추출한 텍스트들 상당히 많은 배열 구조로 이어져 있기 때문에 해당 데이터들에서 필요한 정보만 추출하여 Front-end에 전달한다. 아래 사진은 [그림 15] 사진을 추출하여 저장한 모습이다.



[그림 16] OCR 데이터 추출 후 화면

6. 결론 및 소감

6.1 결론

본 작품을 통해 건강관리의 기본적이고 중요한 측면에 초점을 맞추어, OCR을 활용한 텍스트 추출 및 분석과 Open API를 이용하여 데이터를 제공하는 웹사이트 서비스를 개발하였다. 이를 통해 사용자들이 손쉽게 식단을 기록하고 분석할 수 있게 되었다.

코로나 팬데믹 이후 건강에 대한 인식이 크게 상승하였고, 헬스나 운동 등의 취미 활동에 대한 관심도 높아지는 추세이다. 이에 따라 식단과 운동 기록의 필요성이 점점 더 증가하고 있다. 본 작품에서 개발한 서비스는 바디 프로필이나 대회를 준비하는 사람, 체중 조절을 원하는 사람과 같은 특정 사용자 뿐만 아니라 다양한 대중에게도 도움이 될 수 있다.

이 서비스는 사용자들이 자신의 건강을 쉽게 관리할 수 있도록 돕는 동시에, 개인화된 추천과 분석을 제공함으로써 건강한 생활습관 형성에 기여할 것이다. 또한, 사용자들의 피드백과 지속적인 개선을 통해 더욱 발전하고 다양한 기능을 제공할 수 있는 플랫폼으로 성장할 가능성이 있다.

결론적으로, 본 졸업 작품을 통해 개발한 웹사이트 서비스는 현재 사회의 건강관리에 대한 요구와 기대에 부응하면서, 더 많은 사용자들이 편리하게 이용할 수 있는 서비스로서 지속적인 성장과 발전을 이루어 나갈 것으로 기대된다. 이를 통해 건강한 사회를 위한 한 걸음을 내딛는 데 기여할 수 있기를 바란다.

6.2 소감

1) Front-end

이전에는 주로 회사에서 정해진 프로세스에 따라 개발을 진행하며, 기존 기능의 발전이나 개선에 초점을 맞추는 경험을 했다. 그러나 이번 졸업 작품을 통해 현업에서 다루는 것보다는 작은 규모이긴 하지만, 개발물의 시작부터 끝까지 모든 과정을 직접 경험해 볼 수 있었다. 이 과정에서 서비스 설계부터 구현까지 모든 단계를 직접 수행하며, 내가 알고 있다고 생각했던 이론적 지식이 실제 구현과 어떻게 다른지를 깨닫게 되었다.

특히 웹 개발에서는 적절한 프레임워크와 라이브러리 사용이 중요하다. 이번 작품에서는 Vue.js의 Webpack 설정부터 Vuetify, Vuex, API 설정까지 모두 직접 설계하고 구성했다. 이 과정에서 식품의약품안전처에서 제공하는 식품 영양성분 정보 API와 Naver OCR Model API를 활용해 사진에서 텍스트를 추출하는 기능을 구현하였다. 이를 통해 웹사이트에 모델을 접목하고 검색 기능을 통해 사용자에게 데이터를 제공하는 서비스를 구축할 수 있었다.

이런 경험을 통해 개발 프로젝트의 전체 흐름을 파악하고, 실제 구현 과정에서 다양한 문제를 해결하는 능력을 향상시켰다. 또한, 여러 기술 스택을 활용해보며 기술적 역량을 넓힐 수 있었다. 이와 같은 경험은 앞으로의 개발 활동에 있어 더욱

자신감을 가지고 창의적인 솔루션을 제시하고 구현하는 데 도움이 될 것이다. 이렇게 소중한 경험을 통해 개인적으로 큰 발전을 이룰 수 있었으며, 앞으로도 지속적으로 배우고 성장하는 개발자로 거듭나겠다는 다짐을 하게 되었다.

2) Back-end

웹 개발 과정에서 Front-end와 Back-end 간의 소통이 필수적이기 때문에, 비록 Front-end에 중점을 두었지만, 어느 정도의 Back-end 지식은 갖추고 있었다. 그러나 전문적인 Back-end 공부를 하지 않아, 간단한 API 구성 정도만 가능한 수준이었다. 이번 졸업 작품에서는 Front-end와 Back-end의 설계를 모두 수행해야 했고, Front-end에서 다루는 데이터의 종류와 타입 등이 Back-end 설계와 밀접하게 연결되어 있어서, 단계별로 작업하는 데 많은 어려움을 겪었다.

Front-end에 익숙함 때문에, Front-end 구조에 맞춰 Back-end 설계를 생각하게 되었고, 결과적으로 전체적인 설계보다는 부분적인 설계에 초점을 맞추게 되었다. 이러한 습관을 개선하기 위해 Back-end와 Front-end 간의 소통과 네트워크에 관한 공부를 반복하며 이해를 높이고, 구조를 변경하면서 RESTful 형식을 지키려 노력했다. 이 과정을 통해 Back-end뿐만 아니라 Front-end에 대한 이해도 깊어지는 계기가 되었다.

예를 들어, REST API를 작성할 때의 규칙, 각각의 REST API 메소드에서 JSON 데이터를 다룰 때 Back-end와 Front-end에서 처리해야 하는 작업 등에 대해 다시 한번 살펴볼 수 있었다. 또한, OCR 모델을 API 호출을 통해 가져오고 Open API를 사용하는 과정에서 Back-end에서 이미지를 받아오고 인코딩하는 방법, Secret Key와 Token을 이용한 CORS(Cross-origin Resource Sharing) 및 API 헤더에 대한 이해가 필요했다. 이러한 작업들은 구현과 테스트 과정에서 발생하는 오류를 찾아내기가 매우 어려웠기 때문에, 많은 시간과 노력이 필요했으며, 성공적으로 해결했을 때의 뿌듯함도 컸다.

최종적으로 이번 졸업 작품을 통해 Back-end 개발에 대한 지식과 경험을 확장하고, Front-end와 Back-end 간의 상호 작용에 대한 이해를 높일 수 있었다. 이 경험은 앞으로 웹 개발 프로젝트에서도 큰 도움이 될 것이며, 더욱 전문적인 웹 개발자가 되기 위해 계속 노력하고 성장하겠다.

7. 참고문헌

[그림 1] 한국건강기능식품협회

<https://m.post.naver.com/viewer/postView.naver?volumeNo=31074801&memberNo=40955866>

[그림 2] International Self-care Day in PHILIPS

[1] 네이버 클로버 OCR <https://clova.ai/ocr/?lang=ko>

[그림 4] <https://www.ncloud.com/product/aiService/ocr>

[2] <https://www.heart.org/en/healthy-living/healthy-eating>

[3] <https://www.who.int/publications/i/item/924120916X>

[4] <https://v3.vuex.vuejs.org/kr/>

[표 1] Mifflin, M. D., St Jeor, S. T., Hill, L. A., Scott, B. J., Daugherty, S. A., & Koh, Y. O. (1990). A new predictive equation for resting energy expenditure in healthy individuals. The American Journal of Clinical Nutrition, 51(2), 241-247.

Harris, J. A., & Benedict, F. G. (1919). A biometric study of human basal metabolism. Proceedings of the National Academy of Sciences, 4(12), 370-373.

Ainsworth, B. E., Haskell, W. L., Herrmann, S. D., Meckes, N., Bassett Jr, D. R., Tudor-Locke, C., ... & Leon, A. S. (2011). 2011 Compendium of physical activities: a second update of codes and MET values. Medicine and Science in Sports and Exercise, 43(8), 1575-1581.