



스스로 고민하는 개발자

“ 고민하고 개발하는 ”

누구를 위한 서비스인지

어떤 기능이 필요한지

스스로 고민하고

서비스를 책임질 줄 아는

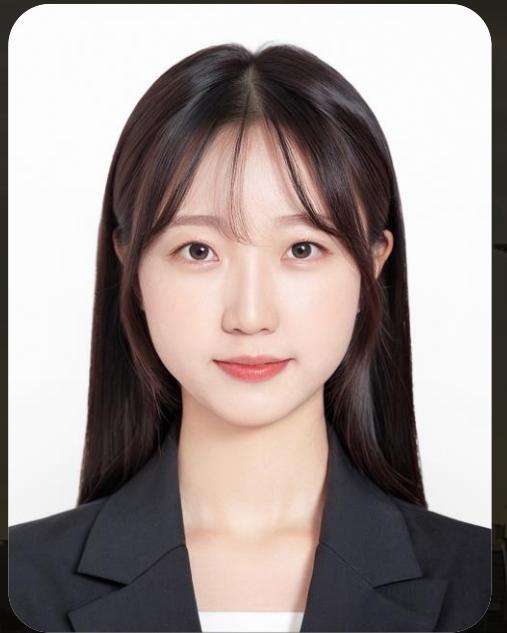
개발자로 일하고 싶습니다

+ 82 010-5153-6560

dev.hanakwon@gmail.com

<https://coday.tistory.com/>

PROFILE



권하나
KWON HANA

1993.10.05

+ 82 010-5153-6560

dev.hanakwon@gmail.com

<https://khndev.github.io/>

EXPERIENCE

2018 단국대학교(죽전 본캠) 졸업
행정학 전공

2020 근로복지공단 근무 (행정 직무)

2022 멀티캠퍼스 JAVA 개발자 교육 이수

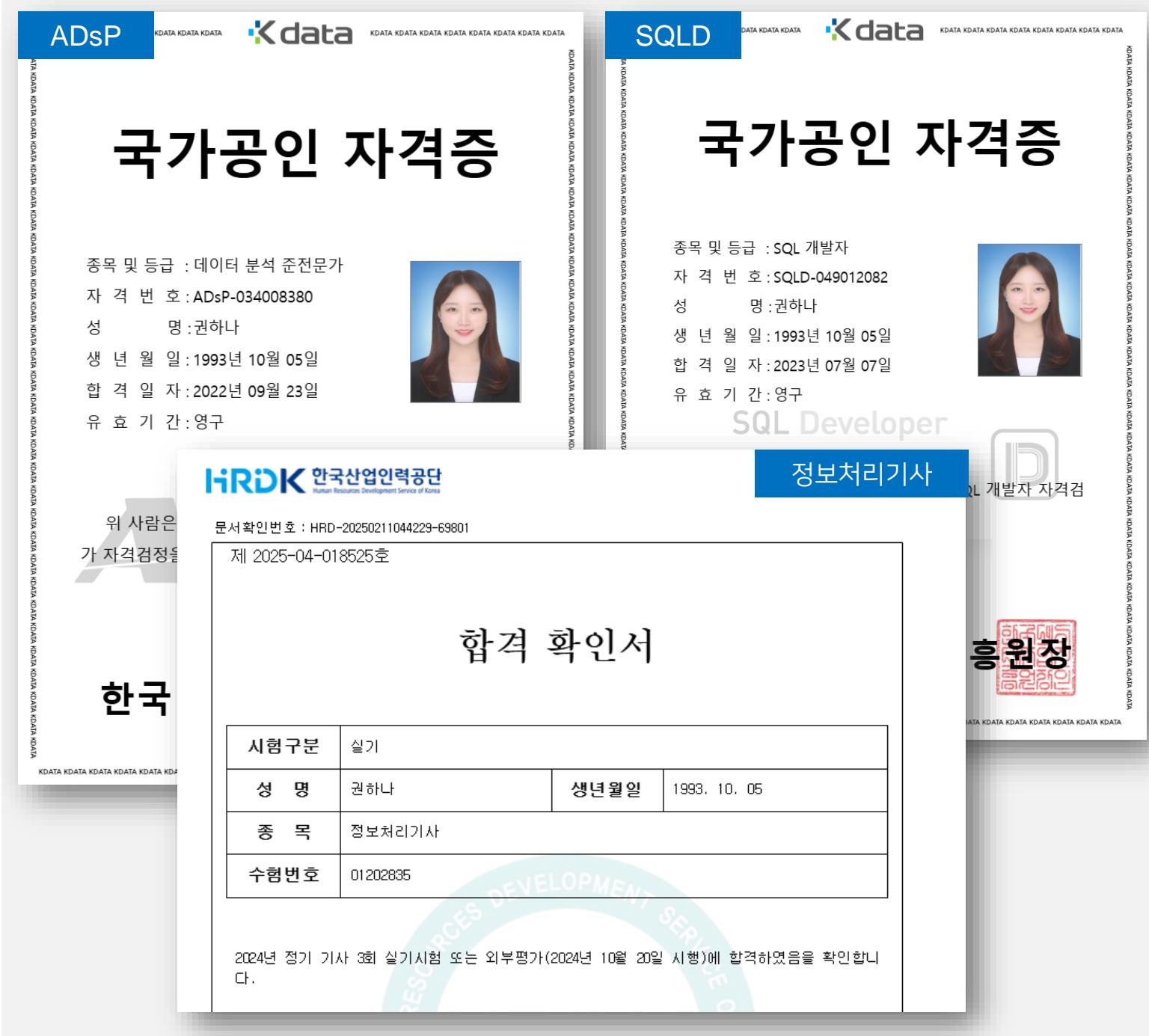
2023~2024 (주)태화이노베이션 재직 (개발 직무)

SKILLS



SKILLS

전문 자격증 취득



SKILLS

스킬

숙련도

활용경험 및 수준

Java



- 경력) 전자화 작업장 및 스캐너 프로그램 개발 및 유지보수
- 경력) 은행 솔루션 개발 프로젝트(신청서 데이터 관리 시스템) 수행
- Java 기반 멀티캠퍼스 교육 수료
- Java 활용 개인/팀프로젝트 3건 진행
- 정보처리기사 자격증 취득으로

DBMS
(MySQL)



- MySQL 사용 프로젝트 3건 진행
- 중복 조회/필터링 기능 구현 및 최적화 경험
- 데이터분석 준전문가 (ADsP),
SQL개발자(SQLD) 자격증 취득으로 SQL문 작성 능력 향상
- 데이터 분석/관리, SQL문 작성 능력 확보
- MyBatis, JPA 기반 프로젝트 진행 경험

Spring & SpringBoot
Framework



- Sping Framework, Spring boot 활용 5건의 프로젝트 진행
- MVC 패턴 이해와 GET, POST 요청/응답 개발 경험
- 은행 솔루션 개발 프로젝트(신청서 데이터 관리 시스템) 수행
- Restful 방식의 서비스 설계 및 구현 경험
- 스케줄러 및 스프링 시큐리티 설정을 통한 프로젝트 최적화
- Spring 기반 전자정부 표준 프레임워크 사용 경험

JS, HTML, CSS



- Next.js와 Spring Boot 연동 프로젝트 수행 (은행 솔루션 개발)
- React ,AngularJS, Vue.JS 를 통한 다양한 UI/UX 프로젝트 진행
- 사용자 입력 데이터 처리 및 동적 페이지 렌더링 구현 경험
- CSS 및 애니메이션 전반 작업 경력 보유

Git



- 팀 프로젝트에서 GitLab과 GitHub를 통한 협업 경험
- 코드 리뷰 및 브랜치 전략 관리 (Merge Request 활용)
- 기능별 업로드(Issues Open/Closed) 경험

AWS



- AWS EC2 활용한 '도서관리 프로젝트' 배포 경험
- Linux 환경 경험 및 명령어 학습

SKILLS

업무노트 작성

전주진행사항(2024.04.22 ~ 2024.04.26)		전주진행사항(2024.08.26 ~ 2024.08.30)			
업무내용	업무내용				
* KB은행 방카 서식솔루션(DNI-HTML) 개발	<p>* DNI 개발</p> <ol style="list-style-type: none"> 1. 이용자시스템 수정 <ul style="list-style-type: none"> - 발송박스 등록 화면 개발 2. 발송박스 입력시 발송박스번호로 증복체크 3. 발송박스 추가 후 최종 저장 시 분류명세번호 생성 기능 4. 발송박스 입력 시 매핑된 커버페이지 존재 유무에 따른 조회 5. 발송박스번호 없으면 커버페이지 검색창으로 커서 옮기는 기능 추가 6. 원장(문서번호)로 바코드 입력하는 기능 추가 7. 원장(문서번호) 바코드 입력시 매핑된 커버페이지번호 및 원장정보 조회 8. 발송박스 존재하면 기존 문서에 원장(문서번호 입력하여) 추가, 발송박스 없으면 발송박스 새로 입력 후 원장 입력 기능 9. 커버페이지에 매핑되는 발송명세번호(연계정보) 다건 삭제 10. 최종 저장 시 증복체크 발송박스번호, 발송명세번호, 원장(문서번호) 증복체크 11. 원장(문서번호) 입력 시 고객사, 커버페이지, 원장(문서번호)로 증복체크 12. 기존 발송박스에 커버페이지 존재시, 원장 추가 저장할 때 기존 발송박스의 발송명세번호로 모두 저장 13. 커버페이지 없는 원장(문서번호) 조회 14. 커버페이지 없는 원장은 커버페이지 생성하여 최종 저장 				
* KB은행 방카 서식솔루션(DNI-HTML) 개발					
- 메인화면 기능 개선					
* 전자문서 열람 목록 조회 카운트 방식으로 수정					
* 그래프 조회 n+1 현상 수정					
* 그래프 작아지는 오류 수정					
* 메뉴 헤더 고정					
* 스크롤 내릴 때 사이드바 이동 가능하게 수정					
* 메인화면 타입 오류 수정					
* KB은행 방카 서식솔루션(DNI-HTML) 개발					
- 로그아웃 기능 개발					
* 유저 아이콘 클릭 시 유저정보 출력 기능 구현					
* 토큰 삭제 기능 구현					
* 리덕스 유저정보 삭제					
7. pdf처리도 다시 count해서 보내기 (완료)					
8. 스크롤로 바꾸고 헤더 고정 연구하기 (완료)					
9. 메인화면 count로 바꾸기 (완료)					
10. 전자문서 열람 목록 조회 카운트 방식으로 수정 (완료)					
11. 그래프 작아지는 오류 수정 (완료)					
12. 페이지네이션 딜레이 추가 (완료)					
▼ 24/04/25 로그인과 역할 연결					
1. 유저 아이콘 클릭 시 유저정보 출력 기능 생성 (완료)					
2. 유저사진->로그아웃 클릭 -> 로그아웃 팝업 -> 확인 (완료)					
3. 서버 토큰 액세스토큰으로 그 행 지우기 -> 토큰삭제 = 디비삭제 (완료)					
4. 리덕스 저장된 유저정보 지우기 (완료)					
5. 로그인 페이지로 리다이렉트 (완료)					
▼ 24/04/30 메인화면					
1. 그래프 그래프 각각 클릭 되게 바꾸기 (완료)					
2. 스크롤 : overflow 적용을 하위에 해서 스크롤 기능이 안되는 것 수정하기 (완료)					
3. 메뉴에 들어갔다가 나오면 위에 네비게이션바 선이 생기는 오류 (완료)					

디버깅을 통한 오류해결

```

429 ▾
430
431 ▾
432
433
434 ▾
435 ▾
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452 ▾

async searchAndSaveCharacter() {
  if (!this.characterName) return; // 캐릭터 이름이 없으면 중단
  try {
    const ocidResponse = await axios.$get(`ocidResponse = {data: {...}, status: 200, statusText: ''}, headers: AxiosHeaders`);
    `http://localhost:8081/api/characters/ocid`;
    {
      params: {
        name: this.characterName,
        personalColor: this.personalColorAnalysis,
      },
    );
    this.characterInfo = ocidResponse.data.characterInfoDTO; ocidResponse = {data: {...}, status: 200, statusText: ''};
    this.message = "";
    this.characterCashItem = ocidResponse.data.searchedCashItemDTO; ocidResponse = {data: {...}, status: 200, statusText: ''};
    this.characterCashFace = ocidResponse.data.searchedCashFaceDTO;
    // 이미지가 로드된 후 extractColors 실행
    const img = new Image(); img = img {alt: '', src: '', srcset: '', sizes: '', crossOrigin: 'anonymous', ...};
    img.crossOrigin = "Anonymous"; // 크로스 도메인 이미지 처리
    img.src = this.characterInfo.character_image;
    img.onload = async () => {
  
```

DB 테이블 구조 및 ERD 설계

finalproject.tn_code: 10 행 (총) (대략적)

CODE_S...	GROUP_SEQ	PARENT_CODE_SEQ	CODE_NAME	CODE_DEPTH	CODE_SORT
1	1	(NULL)	전체	1	1
2	1	(NULL)	개발	1	2
3	1	(NULL)	게임개발	1	3
4	1	(NULL)	디자인	1	4
5	1	2	백엔드/서버 개발자	2	1
6	1	2	프론트엔드/웹퍼블리셔	2	2
7	1	3	게임 서버 개발자	2	1
8	1	3	게임 클라이언트 개발자	2	2
9	1	4	웹/앱 디자이너	2	1
10	1	4	그래픽 디자이너	2	2

SKILLS

Google cloud 배포

Dockerfile X

Plugins supporting Dockerfile files found.

```
1 FROM openjdk:21-jdk-slim
2
3 WORKDIR /app
4
5 COPY build/libs/*.jar app.jar
6
7 EXPOSE 8080
8
9 ENTRYPOINT ["java", "-jar",
10
11 ]
```

Databases

Search anything

Status Subscription Subscription type

Status	Database name	Subscription
●	mefit-ranking	#2582248 mefit-ranking

Connect 2.6MB / 30MB (8.8%) Redis Stack

Items per page: 10 << 1 of 1 >> Page 1 of 1

AWS 배포

aws 서비스 검색 [알트+S]

EC2

A newer release of "Amazon Linux" is available.
Version 2023.1.20230705:
Version 2023.1.20230719:
Version 2023.1.20230725:
Run "/usr/bin/dnf check-release-update" for full release and version update info

```
'          #'
~\_\#\#\#_      Amazon Linux 2023
~~\_#\#\#\`_
~~  \#\#\|
~~  \#/   https://aws.amazon.com/linux/amazon-linux-2023
~~  V~'__->
~~  /
~~.-.  /
~/m/'
```

Last login: Thu Jul 27 21:50:20 2023 from 13.209.1.61
[ec2-user@ip-172-31-39-199 ~]\$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.33 MySQL Community Server - GPL
Copyright (c) 2000, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

K w o n H a n a

p o r t f o l i o

01

P R O J E C T , M E F I T



BABY INFO

온앤온

캐릭터 정보 퍼스널컬러 컬러워즈 컬러픽 로얄스타일

검색

캐릭터 정보 Lv: 285 스카니아 여 온앤온 아크메이지(썬,풀)

퍼스널컬러 블루 브라이트 메인컬러 서브컬러

캐릭터 이미지

빨간색 스텔라 헤어	출신 조용한 얼굴	홍조 라벤더 피부
헤어 발간색: 40% 조록색: 60%	성형 에메랄드: 28% 갈색: 72%	피부
빛의 향연	은은한 밤그레	매력만점
모자 계열: 전체 색상 계열 색: 311 채: 3 명: 14	얼장	눈장
투명 귀고리	고상한 명작	별빛 걸음
귀걸	상의 계열: 정록색 계열 색: 121 채: 33 명: 42	신발 계열: 전체 색상 계열 색: 172 채: 0 명: 99
투명 장갑	천의무봉	사카엘의 검
장갑	망토	무기

Copyright 2025. Mefit. 메이기의 컬러를 찾아드립니다.
Developed by 엘리시움@엔득 & 엘리시움@우멍 후원하기

Data powered by NEXON OPEN API
This is an independent project not associated with NEXON Korea.

↳ 메인화면



1. 메핏: 메이플스토리 코디사이트 개발(백엔드 & 프론트엔드 개발)

2024.10~2025.01 (팀프로젝트 : 2명)

· 개발 주요 사항

- Spring Boot 기반의 백엔드 REST API 설계 및 구현
- Vue.js를 활용한 프론트엔드 개발 및 Ajax 비동기 통신 구현
- 메이플스토리 공식 API, 데이터 크롤링을 통한 캐릭터 정보 및 가챠 데이터 수집
- 퍼스널컬러 분석 기능과 캐릭터 코디 시뮬레이션 기능 제공
- Redis Cloud를 이용한 실시간 캐릭터 랭킹 시스템 구현
- Docker 컨테이너를 사용하여 Google Cloud 에 배포

· 기술 스택

··· 개발 환경

- Java 21
- IntelliJ
- JavaScript
- DBeaver
- JPA
- vue.js
- MySQL
- Visual Studio Code

··· 오픈소스 & 라이브러리

- Maple API
- Redis Cloud
- Google Cloud
- vuety CSS

· 기획 이유

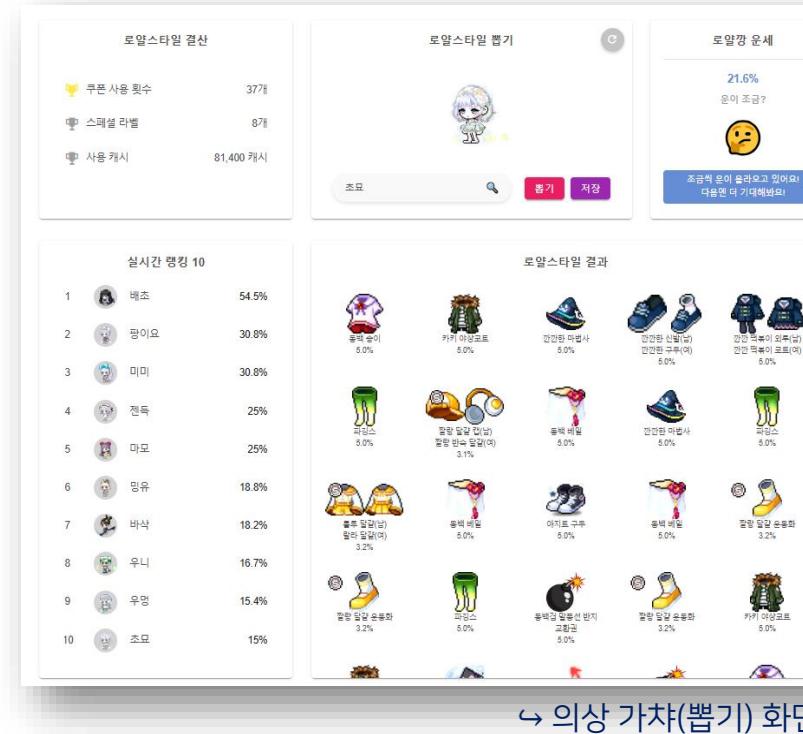
메이플스토리 유저들이 캐릭터 코디에서 가장 중요하게 생각하는 요소는 '색상과 스타일'입니다. 그러나 기존 게임 내 시스템에서는 유저들이 색상 조합이나 스타일을 체계적으로 관리하거나, 원하는 스타일을 미리 시뮬레이션해볼 수 있는 기능이 부족했습니다. 이러한 문제를 해결하기 위해 캐릭터의 외형 정보와 퍼스널컬러, 메인/서브컬러 분석 기능을 기반으로 한 플랫폼을 기획하게 되었습니다.

· 개발 관련 설명

메이플스토리 유저들은 가챠를 통해 희귀 의상을 얻기 위해 실제 게임 내에서 캐시를 사용하지만, 가챠의 확률적 특성상 원하는 아이템을 획득하기 어려운 경우가 많습니다.

이를 해결하기 위해 가챠 시뮬레이션 기능을 개발하여, 유저들이 게임 외부에서 미리 가챠 결과를 체험할 수 있도록 했습니다.

Nexon 페이지에서 제공하는 공식 확률 정보를 크롤링하여 매달 업데이트되는 가챠 아이템 목록을 자동으로 DB에 저장함으로써, 사용자는 최대 100회까지 시뮬레이션을 실행할 수 있습니다.

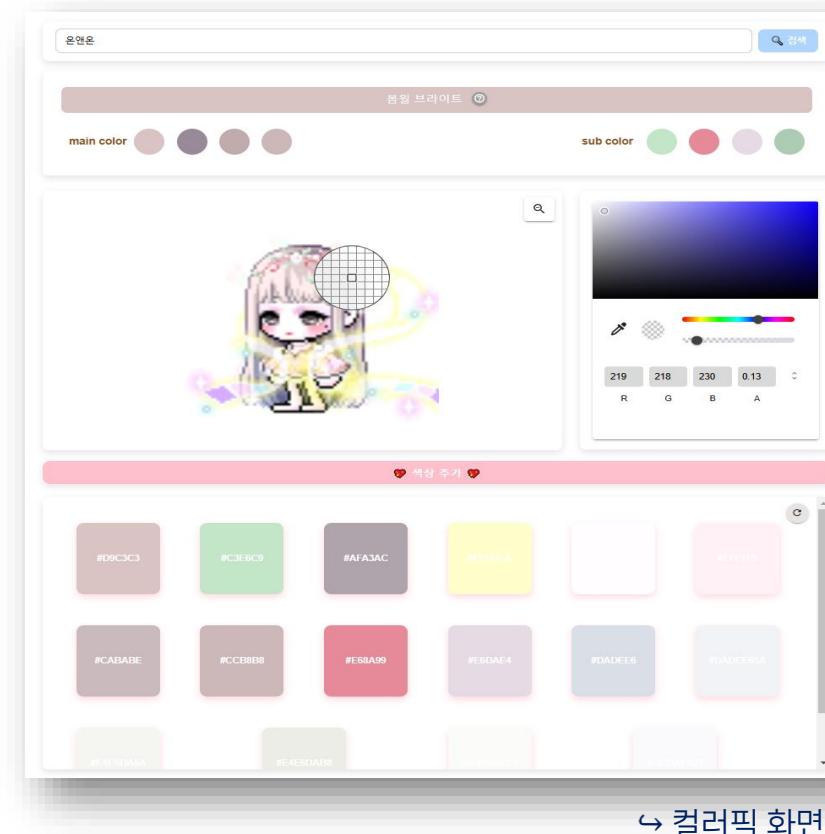


메이플스토리 코디에서 가장 중요한 염색 기능을 좀 더 손쉽게 할 수 있도록 직접 색을 찍어보며 저장할 수 있는 기능을 개발했습니다.

컬러픽커(Color Picker)를 통해 캐릭터 이미지 또는 다른 화면의 색상을 추출하고 저장할 수 있도록 했습니다.

사용자가 추출한 메인컬러와 서브컬러는 퍼스널컬러 분석 기능과 연동되어, 12가지 퍼스널컬러 타입에 따라 추천 캐릭터와 코디를 제공하고 있습니다.

이러한 기능들은 사용자가 손쉽게 색상 기반으로 스타일을 조합하고 분석하는 데 기여하였습니다.



· 담당 구현 기능 (참여율 : 70%)

1. 캐릭터 정보 검색 및 조회 기능

- 메이플스토리 공식 API를 통해 캐릭터의 외형 정보, 모션, 감정 등을 조회 및 커스터마이징

2. 퍼스널컬러 분석 기능

- 메인컬러와 서브컬러를 기반으로 12가지 퍼스널컬러 타입 분류 및 분석

3. 로얄스타일 가챠 시뮬레이션

- 메이플스토리 페이지 크롤링을 통해 주 단위로 가챠 목록과 확률을 DB에 자동 저장

- 사용자가 최대 100회까지 가챠 시뮬레이션 실행 가능

4. 실시간 인기 랭킹 시스템

- Redis Cloud를 활용하여 '가장 운이 좋은 캐릭터' 보여주는 실시간 랭킹 구현

5. 컬러픽커(Color Picker) 기능

- 캐릭터의 색상을 추출하여 화면상에 저장하는 기능 구현

6. 투표 및 선호도 기반 랭킹 기능

- 사용자들이 코디에 좋아요를 누르면 계절별(봄/여름/가을/겨울)로 랭킹을 집계하여 표시

- local storage를 이용해 30분 내 재투표 불가 기능 구현

7. 배포 및 운영

- Docker를 활용해 백엔드는 Cloud Run, 프론트엔드는 Google Cloud에 배포

- 자동 확장(Auto-scaling) 적용 및 Dockerfile 구성으로 효율적인 배포 환경 구축

· 트러블슈팅

◦ 퍼스널컬러 분석 로직 문제

☞ 문제점

퍼스널컬러를 분석하는 API가 존재하지 않았으며, 기존의 퍼스널컬러 분석 사이트들은 단순히 하나의 색상만 입력하여 톤을 판단하는 방식에 그쳐 참고할 레퍼런스가 부족했습니다. 이에 따라 초기 개발에 오랜 시간이 걸렸고, 개발 이후에도 분석 결과가 부정확하게 나타나는 문제가 발생했습니다.

☞ 초기 해결

먼저 색상에 대한 이론적 기초를 학습했습니다. 색상(Hue), 채도(Saturation), 명도(Value)의 개념을 이해하고, 각 값이 캐릭터의 외형 색상에 어떻게 영향을 미치는지 연구했습니다. 이후 직접적으로 적용할 수 있는 아이디어를 바탕으로 로직을 개선하였고, 채도와 명도를 바탕으로 퍼스널컬러와 메인컬러가 시각적으로 적절하게 보이도록 조정하였습니다.

☞ 최종 해결

개선된 로직을 적용한 후에도 다양한 캐릭터 외형에 대한 테스트가 필요했기 때문에, 테스트 페이지를 제작하여 파일 업로드 기능을 통해 약 1,000개의 캐릭터 아바타를 한 번에 분석하는 작업을 수행했습니다. 이를 통해 최종적으로 퍼스널컬러, 메인컬러, 서브컬러 개념이 완성되었습니다. 사용자는 분석된 색상을 기반으로 보다 정확하고 직관적인 코드를 조합할 수 있게 되었습니다.



↳ 퍼스널컬러 분석 화면

· 배운점 & 아쉬운점

◦ 배운점

- 주요 기능들을 주도적으로 개발하며 프로젝트 진행 및 의사 결정 능력 향상
- 색상 이론(Hue, Saturation, Value)을 학습하고 퍼스널컬러 분석 로직을 개선한 경험
- 동료와의 지속적인 코드 리뷰와 피드백을 통해 협업과 의사소통 능력 강화

◦ 아쉬운점

- 초기 기획 단계에서 퍼스널컬러 분석과 관련한 자료 조사가 미흡해 개발 시간이 늘어난 점
- 캐릭터 외형 데이터의 대규모 테스트 환경을 사전에 충분히 준비하지 못한 점
- 일정 및 리소스 제한으로 일부 기능(모바일 최적화 등)이 완벽히 구현되지 못한 점

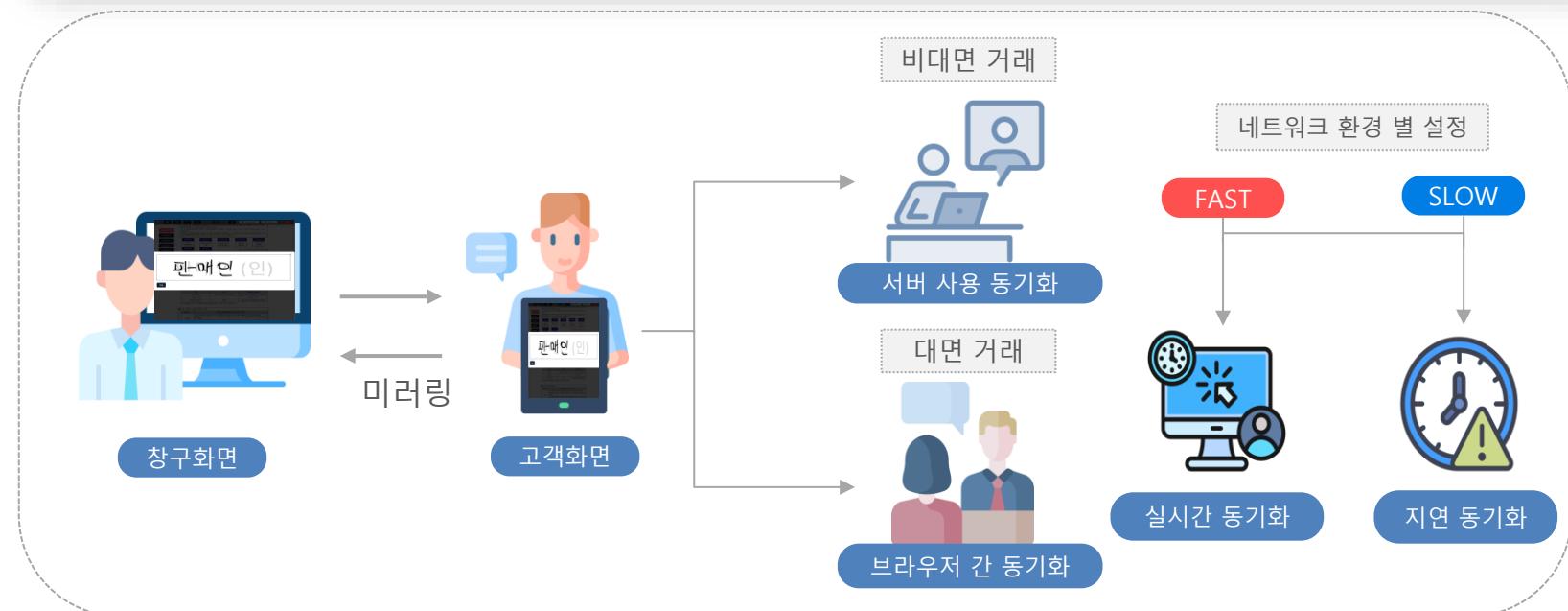
K w o n H a n a

p o r t f o l i o

0 2

P R O J E C T

은행 업무 자동화 프로그램



2. 은행 업무 자동화 프로그램 개발(백엔드 & 프론트엔드 개발)

2024.02 ~ 2024. 05 (사내 개발팀)

· 개발 주요 사항

- 종이 신청서를 스캔하여 HTML 형식으로 변환 및 전산화 기능 개발
- 체크박스 및 서명 인식 기능 구현을 통해 신청서 항목 자동 판별 및 데이터 저장
- 신청서 데이터 전송 및 DB에 저장 처리
- Spring Boot 기반 REST API로 백엔드 서버 개발
- 프론트엔드에서 Next.js 활용하여 사용자 인터페이스 구축

· 기술 스택

· 개발 환경

- | | | |
|----------------------|----------------|-----------|
| - Java 21 | - SpringBoot | - Oracle |
| - NextJS | - IntelliJ | - DBeaver |
| - Visual Studio Code | - tailwind css | |

· 개발 목적

은행에서는 종이 신청서를 디지털 방식으로 전환하기 위해 전산화 솔루션이 필요했습니다. 기존에는 신청서를 수작업으로 입력하고 관리했으나, 본 프로젝트를 통해 신청서를 스캔하여 자동으로 HTML 형식으로 변환하고, 체크박스 및 서명 인식 결과를 데이터베이스에 저장하는 기능을 구현했습니다.

이를 통해 신청서 입력 프로세스를 자동화하여 처리 속도를 개선하고 오류를 줄이는 것이 목표였습니다.

· 개발 관련 설명



은행 신청서와 전자문서를 전산화하는 프로젝트로, 종이 신청서를 디지털 데이터로 변환하여 관리하는 서비스입니다. 주요 기능으로 신청서 데이터를 스캔하여 HTML로 변환하고, 체크박스 및 서명 인식 데이터를 데이터베이스에 저장하는 작업이 포함되었습니다.

· 트러블슈팅

◦ 권한 정보 미반영 문제

문제: 로그인 후 사용자 권한에 따라 접근 가능한 메뉴를 로딩해야 했지만, 권한 정보 API가 비동기적으로 실행되는 동안 메뉴가 먼저 로딩되면서, 권한 정보가 반영되지 않은 메뉴가 표시되는 문제가 발생했습니다.

해결: useEffect 안에서 async/await로 권한 정보가 로드된 후에만 메뉴를 로딩하도록 로직을 수정했습니다. 이를 통해 메뉴 로딩 시점과 권한 상태 변경 시점을 일치시켜 문제를 해결했습니다.



메인화면 관리 기능과 권한·역할별 메뉴 접근 제어를 담당했습니다. 메인화면에서는 공지사항, 서식 관리, 전자문서 관리 현황 데이터를 미리보기 형식으로 표시하여 사용자가 한눈에 파악할 수 있도록 구현했습니다.

· 담당 구현 기능 (팀 내 참여율 : 20%)

◦ 메인화면 관리

- 공지사항, 서식 관리, 전자문서 현황 데이터를 메인화면에 실시간으로 표시
- 데이터 조회 시 중복 호출을 방지하고 페이지 로딩 속도 최적화
- 각 메뉴별 현황 데이터를 카운트 방식으로 정리하여 한 화면에 요약 제공

◦ 메뉴관리

- 계층형 트리 구조로 메인메뉴와 서브메뉴 조회 및 표시
- 메뉴 등록 시 자동 렌더링 및 권한 체크 기능 개발
- 메뉴 추가, 삭제, 순서 변경 기능 및 검색·페이지네이션 기능 구현

◦ 권한 및 역할 관리

- 사용자 역할(Role)에 따른 권한 조회 및 메뉴 접근 제어 기능 개발
- 로그인 후 리덕스에 담긴 사용자 권한 정보에 따라 동적 메뉴 로딩 기능 구현
- 권한별 메뉴 관리 화면에서 권한 설정(체크박스 선택)을 통한 동적 변경 기능 개발

· 배운점 & 아쉬운점

◦ 배운점

- DB 설계부터 주도적으로 하며 구조적인 설계 방식의 중요성을 깨달음
- 메뉴 관리 기능에서 트리 구조와 동적 추가/삭제 구현을 통한 UI/UX 개선 경험
- 권한과 접근 제어 로직을 구현하며 보안과 사용자 경험(UX)의 중요성을 깨달음

◦ 아쉬운점

- 초기 설계에서 권한 및 메뉴 관리 DB설계를 명확히 하지 못해 개발 시간이 늘어난 점
- 주요 기능 중 하나인 HTML 변환 자동화 기능에 직접 참여하지 못한 점

K w o n H a n a ————— p o r t f o l i o

0 3

P R O J E C T

전자화 작업장

3. 전자화작업장 유지보수 및 추가개발

(백엔드 & 프론트엔드 개발) 2023.10 ~ 2024. 10 (사내 개발팀)

· 개발 주요 사항

- 전자 문서 스캔 및 전산화 솔루션 유지보수 및 추가 개발
- 업무 흐름에 따른 개발 : 문서 준비, 스캔, 교정, 품질검사, 마감 작업
- 작업 현황 관리 및 문서 상태별 실시간 데이터 제공
- 자동화 프로세스 개선 및 주요 기능 개발 (검사, 보관박싱 등)

· 기술 스택

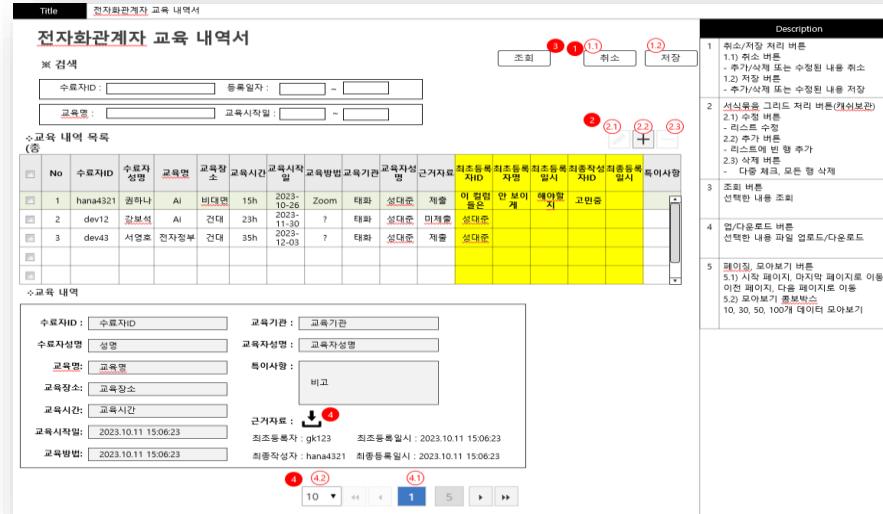
· 개발 환경

- | | | |
|-----------|----------------------|---------------|
| - Java 17 | - JavaScript | - Ms sql |
| - C# | - AngularJS | - Spring boot |
| - DBeaver | - Visual Studio Code | - bootstrap |

· 개발 목적

- 종이 문서를 전자 문서로 전환하여 관리와 검색 속도를 개선하고자 했습니다.
- 스캔, 품질검사, 교정, 박싱 등 전자 문서화 작업을 자동화하여 처리 효율성을 높이고자 했습니다.
- 데이터 무결성과 정확성을 확보하기 위해 다단계 검증 시스템을 도입하였습니다.

· 개발 관련 설명



사내 직원들이 교육을 등록하고 관리할 수 있는 전자화 교육 화면을 개발하였습니다.
교육 프로그램 목록과 세부 과목 정보를 제공하여 사용자가 필요한 교육을 쉽게 확인하고 등록할 수 있도록 구현하였습니다.
타 화면과 데이터 연동을 통해 전체 교육 관리 기능이 유기적으로 동작하도록 개선하였습니다.

· 트러블슈팅

· 데이터 중복 출력 문제

문제: LEFT OUTER JOIN으로 인해 중복 데이터가 조회되는 문제가 발생하였습니다.

해결: 쿼리 조건을 최적화하고, 중복을 제거하기 위한 필터 로직을 추가하여 문제를 해결하였습니다.

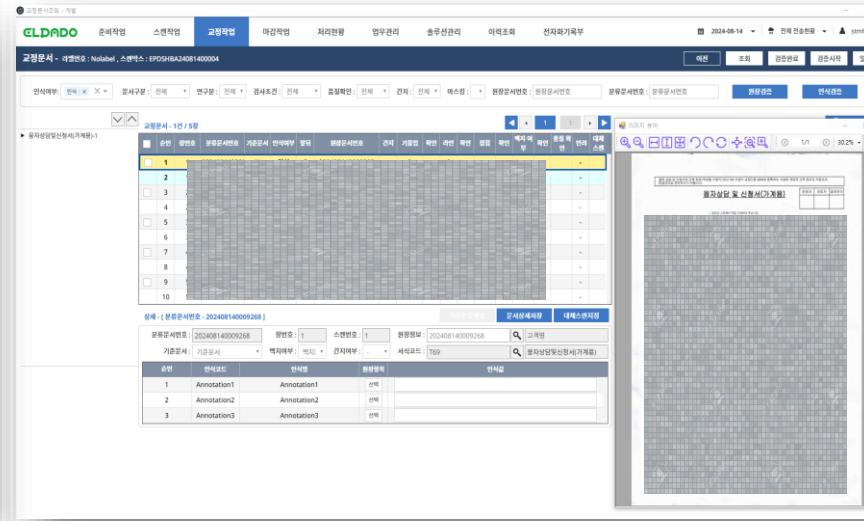
· 데이터 동기화 문제

문제: 전자화 교육 프로그램 수정 시 상태값이 'N'으로 변경되지 않는 문제가 발생했습니다

원인: 상태값 변경 로직이 프로그램 내 일부 화면에서만 적용되고 다른 화면에서는 호출되지 않아,

데이터 갱신이 일관되지 않은 상황이 발생하였습니다.

해결: 상태값 변경 로직을 공통 서비스로 통합하여 모든 화면에서 일관되게 호출되도록 개선했습니다.



문서 이미지 뷰어에 타사의 OCR API를 연동하여 문서 유형을 자동으로 판별하는 기능을 추가하였습니다.
OCR 인식 정확도를 높이기 위해 OCR이 잘못 인식하는 위치정보를 조정하고 주민등록번호와 발급일 형식이 일치할 경우 주민등록증으로 판단하도록 구현하였습니다.

· 담당 구현 기능 (팀 내 참여율 : 30%)

1. 전자화 교육 화면 개발

- 기본적인 CRUD 개발

- 타 화면과 연계하여 교육 프로그램 기능 추가

(예: 교육 프로그램 관리 시 서식 정보 연동)

- 상태값 변경 로직 보완 및 프린트 기능 개발

2. 서식 및 태그 관리 기능 개선

- 검색 조건 최적화하여 서식 인식 태입 및 태그명 검색 정확도 개선

- 서식 관리 화면에서 관계사와 작업사 선택 항목 구현

- 코드 관리 기능에서 중복 검색 및 처리 로직 보완

3. 문서 이미지 뷰어의 OCR 기반 문서 유형 판단 기능 추가 개발

- 문서 이미지 뷰어에 타 회사의 OCR API를 연동하여 문서 유형을 자동으로 판별하는 기능 개발

- OCR 결과로 주민등록번호와 발급일 형식을 검증하여 주민등록증 여부를 판단하는 로직 구현

- 인식 오류를 해결하기 위해 OCR 위치 조정 및 구조 파악 로직 추가

· 배운 점 & 아쉬운 점

· 배운 점

- 문서 이미지 뷰어와 같은 복잡한 화면에서 타사의 API 연동 작업을 통해 외부 API 연동 후

추가 개발, 예외 처리 경험을 쌓음

- Angular와 Bootstrap을 활용한 화면 개발 및 로직 최적화 작업을 통해 UI/UX 개선 방법을 학습

· 아쉬운 점

- 사용하는 Java와 Angular의 버전이 낮아 최신 기능들을 충분히 활용하지 못한 점

K w o n H a n a

p o r t f o l i o

0 4

P R O J E C T

농협은행 전자화작업장

4. 농협은행 전자화작업 프로그램 유지보수 및 추가개발

(백엔드 & 프론트엔드 개발) 2024.01 ~ 2024. 10 (사내 개발팀)

· 개발 주요 사항

◦ 전담 맞춤형 기능 개선

- 농협 시스템 사용 부서로부터 직접 요청받은 작업 지연 문제에 대한 개선 작업을 주도.

◦ 테스트 및 패치

- 출장 지원을 통해 현장 환경에서 발생한 문제를 신속히 파악하고, 최신 빌드 배포 및 패치를 직접 수행하여 사용자 불편을 최소화.

· 기술 스택

◦ 개발 환경

- Java 8

- JSP

- My Batis

- Spring(전자정부프레임워크) - AngularJS

- Ms sql

- bootstrap

- DBeaver

· 담당 구현 기능 (팀 내 참여율 : 100%)

- 사용자 맞춤 검색 기능: 사용자가 원하는 조건으로 데이터를 필터링할 수 있는 다중 조건 검색 기능 구현

- 출력 기능 다양화: PDF, Excel 등 다양한 포맷으로 데이터 일괄 출력 가능하도록 선택 옵션 추가

- 권한별 접근 제어: 사용자의 역할에 따라 출력 기능과 데이터 접근 권한을 제한하는 로직 추가

K w o n H a n a

p o r t f o l i o

0 5

P R O J E C T
CODE:ONE



The screenshot shows the homepage of the Code:One website. At the top, there's a navigation bar with '채용' (Recruitment) and '일정' (Schedule). Below the navigation, there's a large diagram illustrating user roles and their interactions:

- 일반회원 (General Member):** Represented by three people. They can log in as a company member and change their profile to a company member.
- 기업회원 (Company Member):** Represented by a building icon. It shows how a general member can change their profile to a company member.
- 서비스 목록 (Service List):** Includes icons for **BLOG**, **스터디 모집/관리**, **일정관리**, **중고거래**, **강의샵 강의 구매**, **채용**, and **마이페이지 작성 글 확인**.
- 관리자페이지 관리**: Shows how a general member can change their profile to a company member.
- 채용일정**: Shows a timeline for recruitment processes.

Below the diagram, there's a search bar with fields for '회사 이름을 입력해주세요' (Enter company name), '직무' (Job title), '경력' (Experience), '지역' (Region), and a '검색' (Search) button. Underneath the search bar, there are filters for 'Java', 'Python', 'JavaScript', 'React', 'Vue', '직무 근무', '유연 근무', '점심 제공', '시차 출근제', '인센티브', '교육', and '노트북 지원'.

At the bottom, there's a section titled '채용중인 글만 보기' (Only view posts from companies currently recruiting) showing five job listings:

- 와랩스 ♥ 게임 서버 개발자 Python 미술(4~8년)
- 센스톤 ♥ 월별 디자이너 Vue 디자인(4~8년)
- 모두의연구소 ♥ 2명 디자이너 React 경력무관
- 풀리토 ♥ 프론트엔드/웹마일러서 JavaScript 인턴
- (주)페이하이 ♥ 게임 클라이언트 개발자 Vue 시니어(9년 이상)

At the very bottom, there's a magnifying glass icon and text: "Back-end : <https://github.com/leechangsin/codeone>" and "Front-end : <https://github.com/kwatermelon/CodeoneReact.git>".

5. 개발자 포털사이트 서비스 개발 (백엔드 & 프론트엔드 개발)

2023.04 ~ 2023. 05 (5명)

· 개발 주요 사항

- 네이버 클라우드 Chat Bot 활용하여 서비스 추천 기능 개발
- Spring Boot 기반 REST 백엔드 서버 개발
- I'm Port API를 활용한 구매서비스 개발
- Google, KakaoTalk, Naver 소셜 로그인 연동 기능 개발

· 기술 스택

- | | | |
|----------------------|------------------|-----------------------|
| ··· 개발 환경 | ··· 오픈소스 & 라이브러리 | |
| - Java 11 | - MyBatis | - 국세청 공공 API |
| - SpringBoot | - Git, Maven | - ImPort 결제 API |
| - React | - MySQL | - FullCalendar 일정 API |
| - Eclipse | - Workbench | - Kakao API |
| - Visual Studio Code | | - jQuery |

· 기획 이유

'Code:One'은 흩어져있는 개발 관련 정보를 한 사이트에 모은 포털사이트입니다.

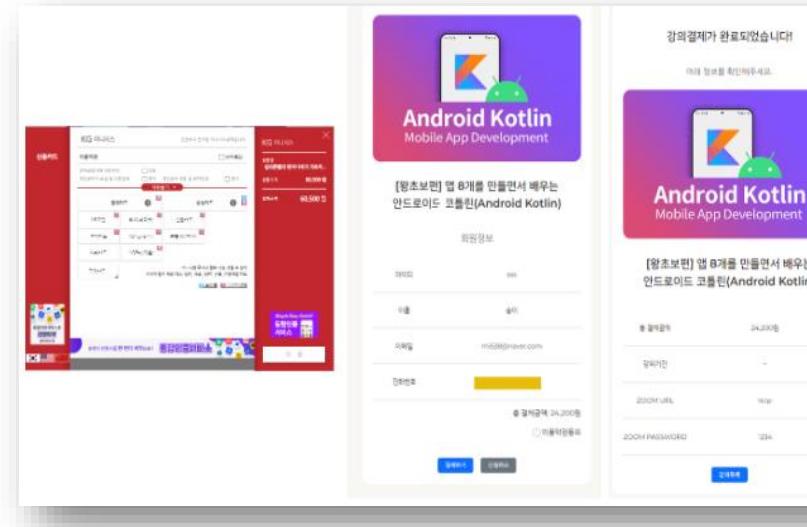
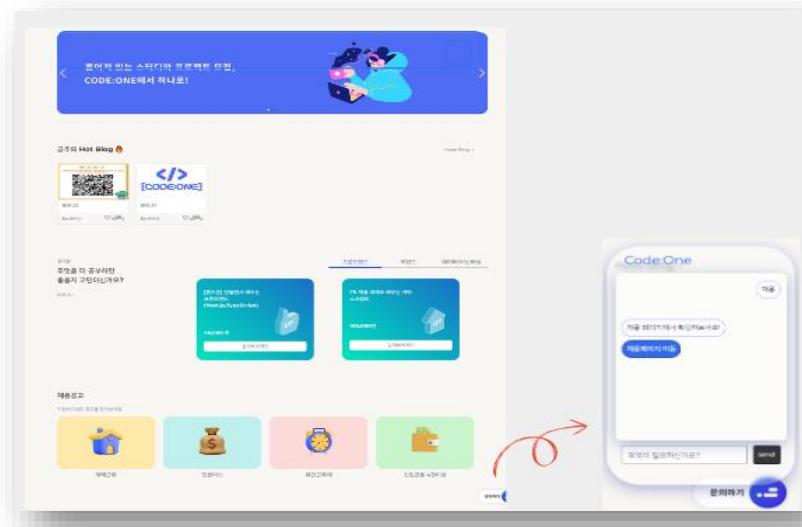
블로그, 스터디, 일정관리, 중고거래, 강의샵, 채용 관련 정보가

각각 다른 사이트에 흩어져 있어서 초보 개발자로서 불편한 점이 많았습니다.

그래서 개발자에게 필요한 모든 것을 한 곳에 모은 All-In-One 사이트

Code:One을 만들었습니다.

· 개발 관련 설명



RESTful API를 통해 SpringBoot 환경에서 네이버 클라우드 Chat Bot 서비스를 구현하였고 사용자가 원하는 서비스를 입력하면 Chat bot이 문자 또는 이미지로 응답할 수 있도록 여러 차례 엔진을 업데이트했습니다. 클라우드 서비스의 장점을 느낄 수 있는 경험이었습니다.

I'mPort API를 활용해 강의샵에서 사용자에게 강의를 판매하는 기능을 구현했습니다. 결제한 사람만 강의 정보를 확인할 수 있도록 React의 localStorage 로그인 정보를 이용, 페이지 접근 제한 기능을 추가해 보안에 노력을 기울였습니다.

· 트러블슈팅

· CORS 오류 해결

초기 해결: 사업자 등록번호 API 이용한 화면이 cors(Cross-Origin Resource Sharing) 오류가 나서 임시로 chrome 브라우저를 관리자 권한으로 실행하는 방법을 이용했습니다.

최종 해결: 디버깅을 통해 오류 메시지를 분석해 브라우저의 cors 정책 강화를 원인으로 확인했고 클라이언트 단의 withCredentials 옵션을 false로 설정해 요청을 보낼 때 쿠키와 같은 인증 정보를 함께 보내지 않도록 처리했습니다. 이로 인해 브라우저가 cors 정책을 더 유연하게 적용하여 오류를 해결했습니다.

· 담당 구현 기능 (참여율 : 30%)

· 강의샵 & 채용 게시판 & Chat Bot

- I'm Port API 활용해 강의 구매(카드결제) 기능 구현
- DB를 통해 좋아요와 좋아요 해제 시 카운팅 기능 구현
- 기업 회원만 접근 가능한 기업회원 전용 게시판 구현
- 로그인 DB 활용해 기업회원 변경/기업회원만 접근 허가하는 보안 기능 구현
- 국세청 사업자등록번호 API 활용한 기업회원 인증 기능 구현
- HeidiSQL 이용, MySQL 서브쿼리 작성으로 채용정보 중복 필터링 기능 구현
- axios를 통한 비동기 방식의 작성 데이터 추출
- 카카오맵 API를 활용해 회사 위치 확인 기능 구현
- FullCalendar API 활용하여 채용일정 동기화 기능 구현
- 네이버 클라우드 엔진 빌드를 통한 Chat Bot 기능 구현

· 배운점 & 아쉬운점

· 배운점

- 필터링 기능 구현하며 복잡한 SQL문 작성 능력 향상
- 기업회원 외 접근 제한 기능 추가하며 DB관리의 중요성 느낌
- 결제 기능 구현하며 웹사이트 구현 시 보안의 중요성을 느낌

· 아쉬운점

- JPA 활용하려 했으나, ORM이 익숙치 않아서 활용하지 못한 점
- 비용 문제로 Chat bot 엔진을 더 빌드하지 못한 점

K w o n H a n a

p o r t f o l i o

0 6

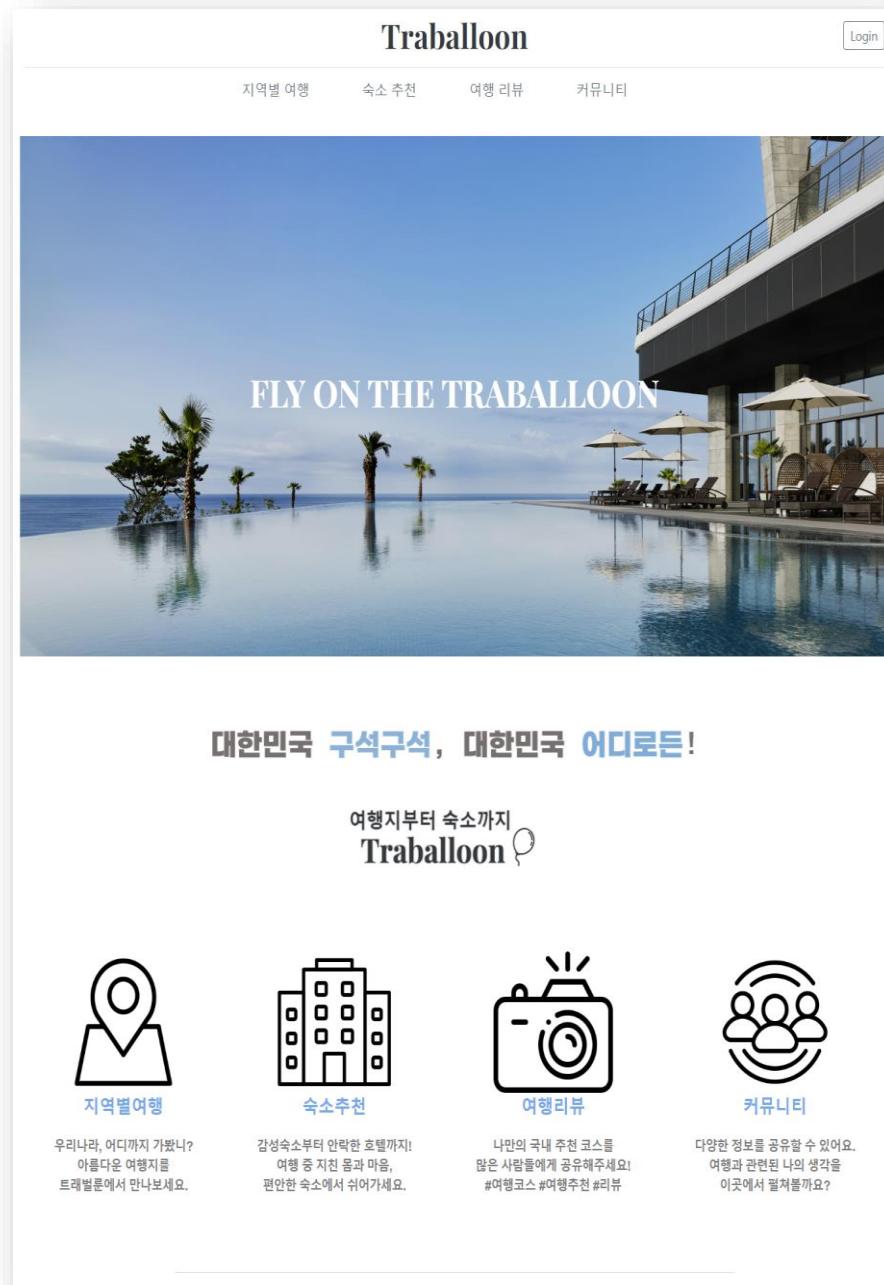
P R O J E C T

Traballoon

Traballoon

6. 국내 여행 사이트 서비스 개발 (백엔드 & 프론트엔드 개발)

2023.03 (4명)



<https://github.com/bdy4119/Tiles-Traballoon.git>

· 개발 주요 사항

- Spring Framework 기반 백엔드 서버 개발
- Ajax 비동기 통신 활용한 화면 구성

· 기술 스택

· 개발 환경

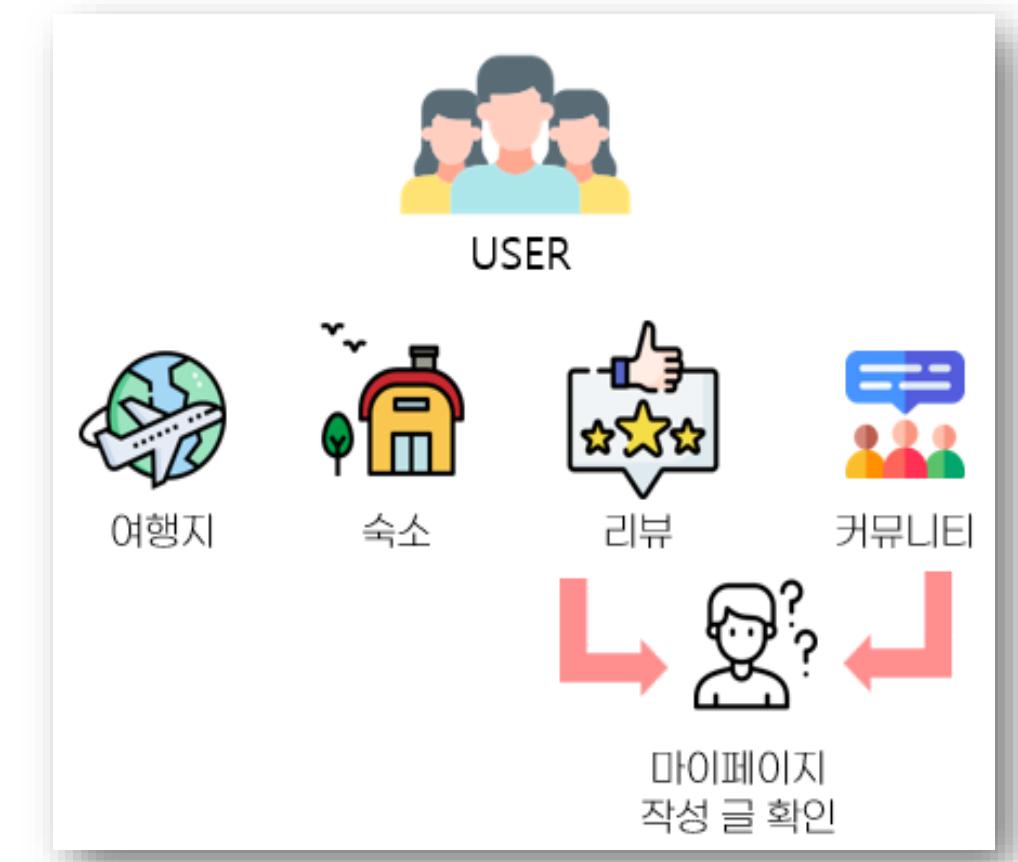
- Java 11
- Spring
- JavaScript
- Eclipse
- JSP
- Hikari CP
- MySQL
- Workbench

· 오픈소스 & 라이브러리

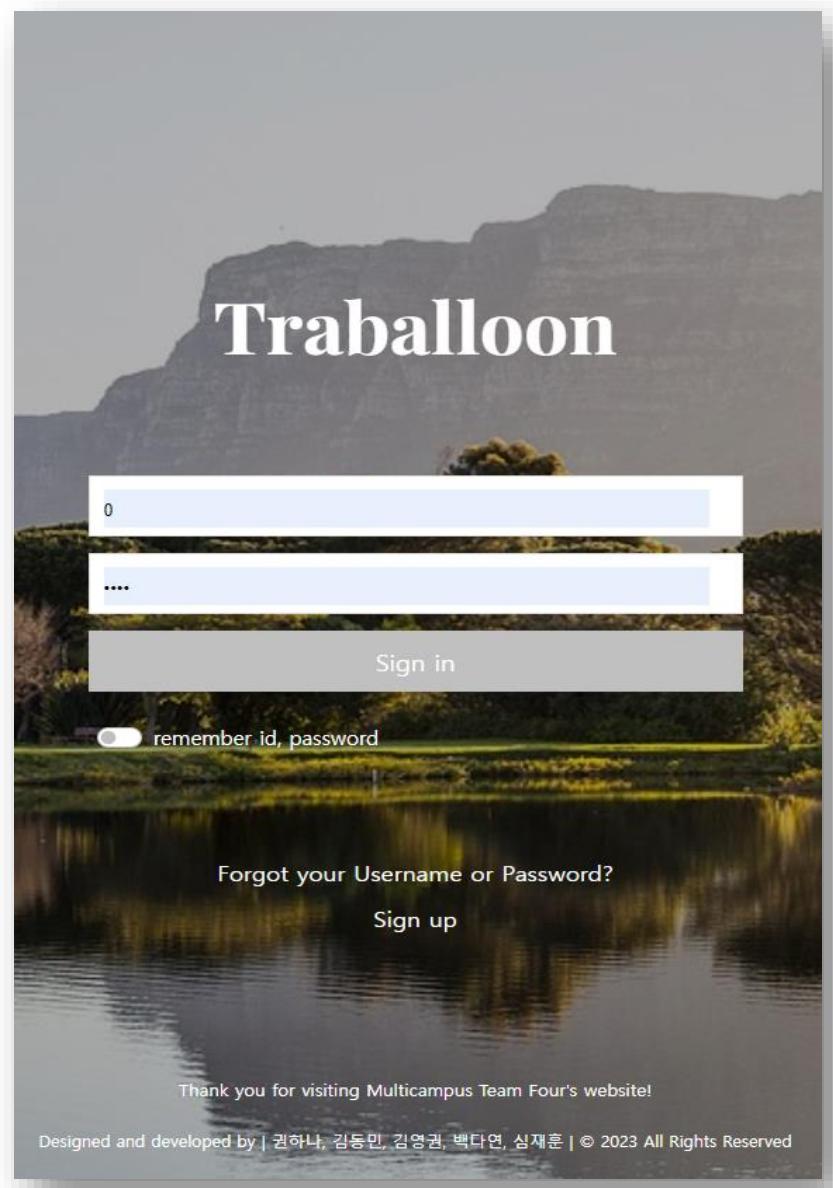
- Kakao API
- Summer-note API
- jQuery
- Bootstrap

· 기획 이유

코로나가 완화되면서 국내 여행에 대한 수요가 급증했지만, 국내 여행에 대한 상세한 정보를 한 눈에 보는 커뮤니티를 찾기가 어려웠습니다. 여행지에 대한 정보를 파악하면서 회원 간 정보를 공유한다면 사용자가 원하는 사이트를 개발할 수 있지 않을까? 하는 마음으로 만들었습니다.



· 개발 관련 설명



세션을 통해 Spring 환경에서 회원가입, 로그인, 회원정보 수정, 회원탈퇴 기능을 구현했습니다. 이 기능을 구현하면서 DB 관리의 중요성을 몸소 느꼈습니다.

```
<!-- 6. DB -> 카카오맵 연결 -->
<br><br>

<div style="...">
    <div class = "mapimage" style="text-align: left; ">
        <img src= "./hotelimage/detail2.png" > <br>
    </div>
</div>

<div id="map" style="..."></div>
<script>
    var container = document.getElementById('map');
    var options = {
        center: new kakao.maps.LatLng(<%=hotelmap %>),
        level: 3
    };

    var map = new kakao.maps.Map(container, options);

    var markerPosition = new kakao.maps.LatLng(<%=hotelmap %>);

    var marker = new kakao.maps.Marker({
        position: markerPosition,
        map: map
    });

</script>
<br><br>
<div class="border-top-gray" style="..."></div>
```

KakaoMap API를 활용해 숙소 위치를 사용자에게 보여주는 기능을 구현했습니다. 지도 확대와 축소가 되도록 사용자 UI 측면도 고려했습니다.

· 담당 구현 기능 (참여율 : 30%)

- 회원가입/로그인 & 숙소추천 게시판
 - 세션 활용해 회원가입, 로그인, 탈퇴 기능 구현
 - DB를 통해 사진, 상세 텍스트, 조회수 추출
 - Ajax 비동기 통신을 사용해 댓글 작성과 최신, 조회수순, 댓글순 정렬 기능 구현
 - limit 기능을 활용한 페이지 처리
 - 알고리즘 활용하여 썸네일 3x3 정렬 구현
 - 카카오맵 api를 사용해 숙소 위치 전달
 - 프로젝트 전반 tiles 설정과 CSS 전반

· 배운점 & 아쉬운점

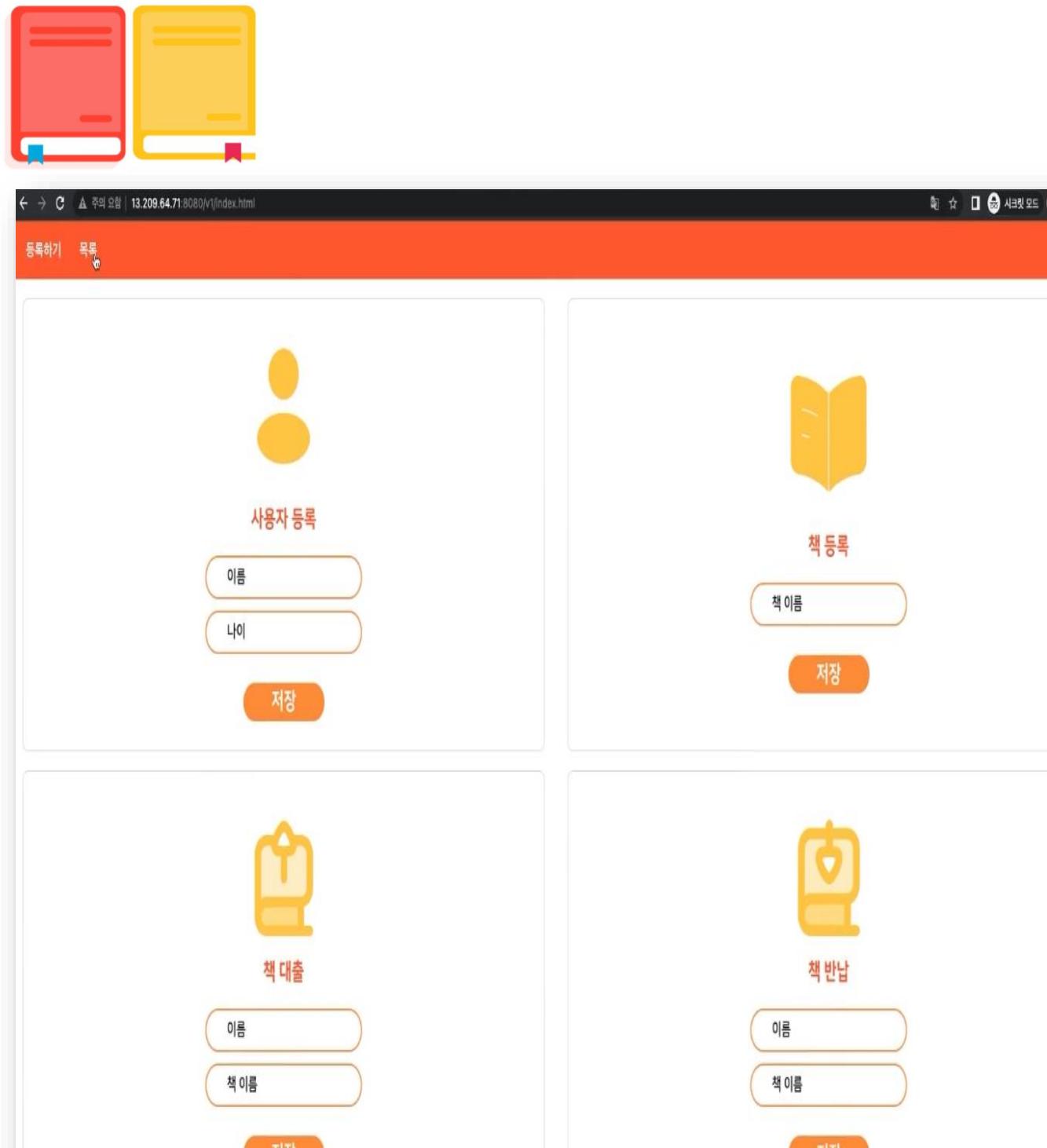
- 배운점
 - Java를 사용한 데이터 처리 및 Spring Framework 활용 능력 향상
 - MySQL 활용법, 흐름 이해 등 DB 이해도 강화
 - RDBMS(관계형 데이터베이스) 활용으로 서브쿼리, 조인 스킬 향상
 - Front-End와 Back-End간 get, post에 따른 Ajax 비동기 통신 학습
- 아쉬운점
 - 단기간 프로젝트로 구현하고 싶은 기능을 다 구현하지 못했음

K w o n H a n a ————— p o r t f o l i o

07

P R O J E C T

도서관리 서비스



7. 도서관리 서비스 개발 (백엔드 개발)

2023.07 (개인)

· 개발 주요 사항

- Spring Framework 기반 백엔드 서버 개발
- AWS EC2 가상서버를 이용해 서비스 배포
- Spring Data JAP 를 이용해 회원 조회, 등록, 도서 등록, 대여, 반납 기능 구현

· 기술 스택

- | | |
|--------------|------------------|
| ··· 개발 환경 | ··· 오픈소스 & 라이브러리 |
| - Java 8 | - JSP |
| - SpringBoot | - AWS |
| - JavaScript | - MySQL |
| - IntelliJ | - GitHub |

· 기획 이유

Code:One 프로젝트 이후 SpringBoot를 활용해 CRUD, 배포를 경험할 수 있는 REST 백엔드 서버를 혼자 구현하고 싶다는 생각을 했습니다. 이런 개인 프로젝트를 하기 위해서는 회원 등록, 책 등록, 대출, 반납 등의 기본 기능을 갖춘 도서관리 서비스가 적합하다고 판단했습니다.

· 개발 관련 설명

```
//두번째 방법(더욱 깔끔하게-> 자바8버전의 스트림을 활용한 코드)
public List<UserResponse> getUsers(){
    return userRepository.findAll().stream() Stream<User>
        .map(UserResponse::new) Stream<Object>
        .collect(Collectors.toList()); //결과를 리스트로 바꾸기
}
```

JPA와 Java 8 버전의 스트림을 이용해 클린코드로 프로젝트를 개발했습니다. SQL문을 작성하지 않고 JPA 메소드를 이용했고, 스트림을 사용해 더 간결하고 가독성이 좋은 코드를 작성했습니다.

```
/bin/
Last login: Thu Jul 27 21:50:20 2023 from 13.209.1.61
[ec2-user@ip-172-31-39-199 ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

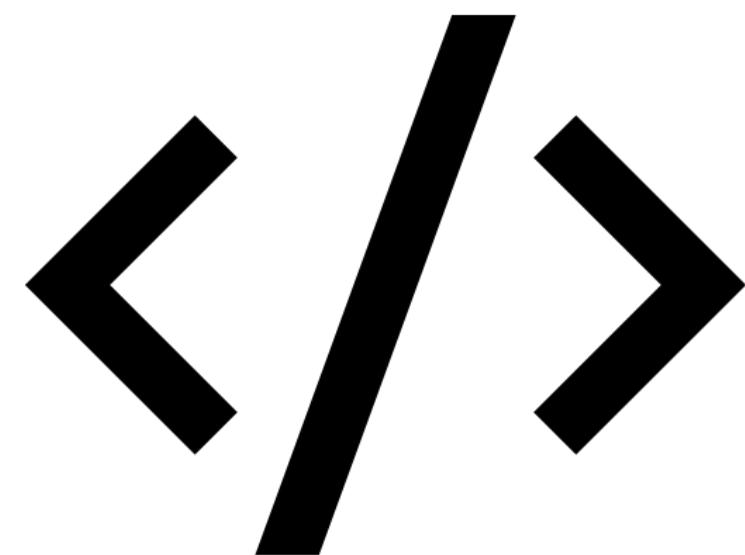
AWS의 EC2 가상서버를 이용해 프로젝트를 배포했습니다. Git과 연동하여 최신 코드 유지와 지속적인 모니터링에 노력을 기울였습니다. 자동화된 배포 프로세스를 이용하여 AWS 서비스의 장점을 느꼈습니다.

· 담당 구현 기능 (참여율 : 100%)

- DB생성, 회원등록과 도서출납 CRUD 전반
- Spring Data JPA 를 이용해 CRUD 기능 구현
- DB 활용하여 회원 조회, 등록, 도서 등록, 대여, 반납 기능 구현
- Git과 연동하여 AWS EC2 가상서버를 이용해 프로젝트 배포

· 배운점 & 아쉬운점

- 배운점
 - API 스펙을 확인하며 개발해 실무 능력에 기여
 - 스프링 빈, 컨테이너, 트랜잭션 관련 코드 학습으로 스프링 활용 능력 향상
 - JPA를 활용한 DB 조회, 수정, 업데이트 방법 습득
 - intelliJ 환경에서 내장 DB 사용 방법 습득
 - AWS EC2를 활용한 배포 후 리눅스 명령어 습득, 오류 해결 능력 향상
 - 클린코드 작성방법 습득
 - Postman 활용, 백엔드단 구현 능력 향상
- 아쉬운점
 - AWS 비용 문제로 인한 짧은 배포 기간



“필요 기술 학습”

서비스의 기술 스택을 활용하는 프로젝트를
진행하며 업무에 빠르게 **적응**하겠습니다.



“맞춤 솔루션 개발”

고객의 입장에서 맞춤 **솔루션**을 고민하고
개발하는 개발자가 되겠습니다.

K w o n H a n a

p o r t f o l i o

END

+ 82 010-5153-6560

dev.hanakwon@gmail.com

<https://coday.tistory.com/>