PORTFOLIO

KIM-JONGWON

CONTACT

PHONE: 010-8802-5743 | MAIL : growjong8802@gmail.com

Web/Backend Developer

김종원

PROJECTS

HeRoes(인사 관리 시스템) 01

- 프로젝트 소개
- 담당 및 시스템 아키텍처
- 개선 및 문제 해결 사례

OMOS(DEVOPS 프로젝트) 02

- 프로젝트 소개
- 담당 및 시스템 아키텍처 / CI/CD 아키텍쳐 및 설명
- 개선 및 문제 해결 사례

DAZZLE(포토 스팟 공유 플랫폼) 03

- 프로젝트 소개
- 담당 및 시스템 아키텍처
- 개선 및 문제 해결 사례

TWOFAANG(쇼핑몰 프로젝트) 04

- 프로젝트 소개
- 담당 및 시스템 아키텍처
- 개선 및 문제 해결 사례

FUNBIT(가상 화폐 거래소 클론) 05

- 프로젝트 소개
- 담당 및 시스템 아키텍처
- 개선 및 문제 해결 사례

동의보감(헬스케어 서비스) 06

- 프로젝트 소개
- 담당 및 시스템 아키텍처
- 개선 및 문제 해결 사례

이곳에 자신의 매력을 보일 수 있는 소개 글을 기재해 주세요. 개성 있는 자기소개를 입력해주세요. 또는 목차 페이지에서 필요한 내용을 적어도 좋아요.

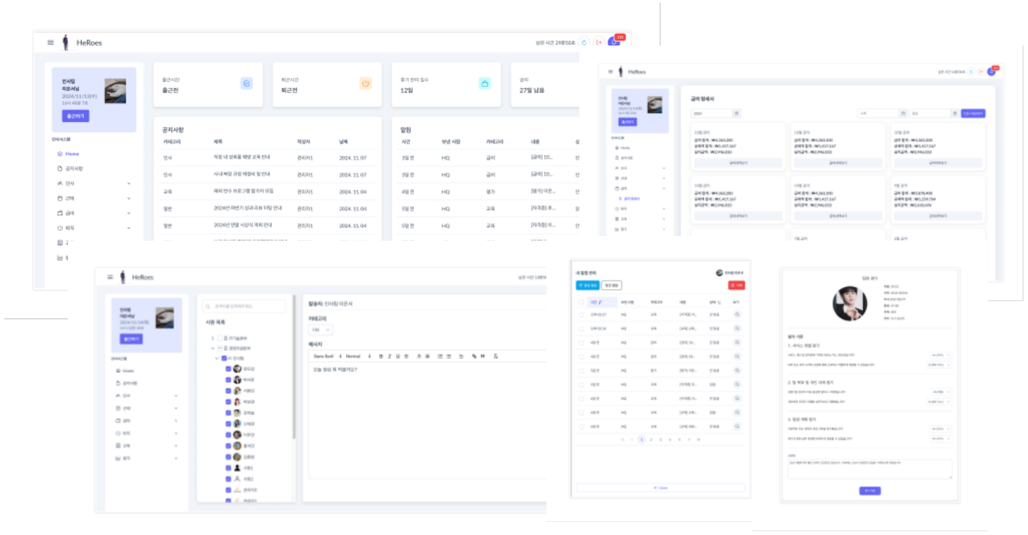
HeRoes

인 사 관 리 시 스 템

프로젝트 소개

HeRoes(인사 관리 시스템)

한화시스템 BEYOND SW CAMP 8기	사용 기술 : Java, SpringBoot, Spring Data JPA, Spring Batch, Spring Security, Vue.js, MariaDB Docker, Kubernetes, Jenkins, ArgoCD, AWS	2024.09.11 ~ 2024.11.08



한화시스템 BEYOND SW CAMP에서 진행한 최종 프로젝트로, 사원들의 근태와 평가 데이터를 기반으로 업무를 자동화하고 팀 캘린더와 교육 프로그램을 제공하여 사원 만족도를 높이는 인사 관리 시스템 개발

Java 기반 Spring Boot 서버와 Vue.js 기반 클라이언트로 구현한 프로젝트로 서버는 JPA를 사용해 관계형 데이터베이스를 관리하며 Spring Batch와 Spring, Security를 활용한 서비스 구성

클라이언트는 Axios를 통해 RESTful API와 통신하며 Pinia와 LocalStorage를 사용한 데이터 관리 구조

AWS EKS에 서비스를 배포하고 EC2, ECR, Route 53, ALB를 활용한 인프라 구축 및 GitHub, Jenkins(CI), ArgoCD(CD)를 활용한 배포 자동화 구성

HeRoes 깃허브 레포지토리 바로가기

담당 및 시스템 아키텍쳐

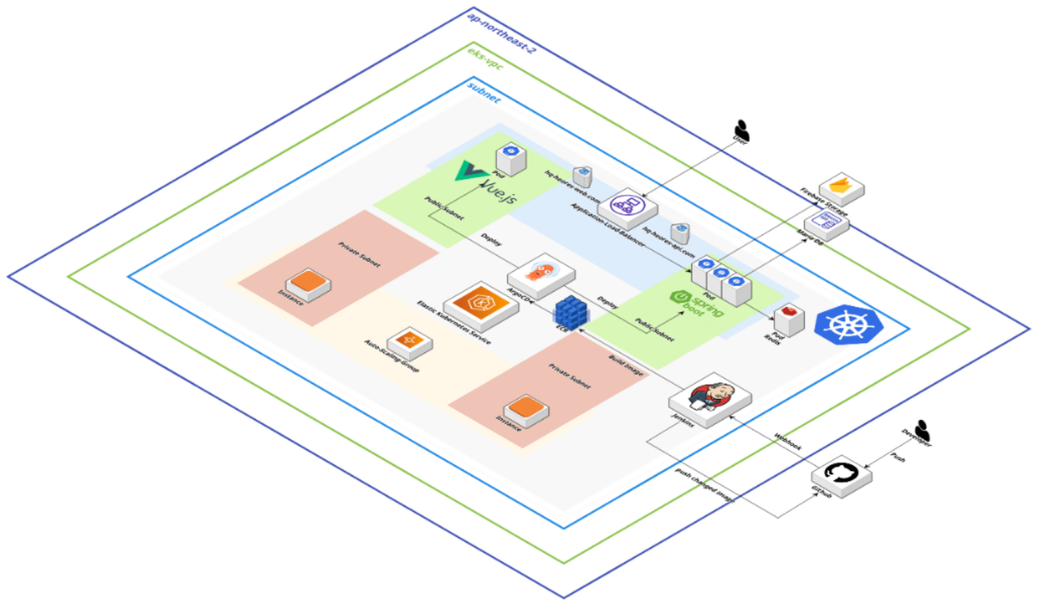
팀원 구성 및 역할

Full Stack 5명 팀장

담당 개발 업무

사용자 인증	사원 로그인, 관리자 로그인, 사원 등록, 비밀번호 재발급 화면 구현
	Spring Security, JWT 기반 사용자 인증, 이메일 인증 구현
평가 시스템	부서 별 평가 양식 관리 페이지, 팀원 평가 및 결과 확인 페이지 구현
	부서 별 평가 양식 관리, 팀장 → 팀원의 하양식 평가 프로세스 구현
	배치를 통한 상/하반기 평가 결과 정산 프로세스 구현
알림 시스템	사원이 다른 사원에게 알림을 발송하고, 개인 알림 관리 기능 구현
	비동기 및 수/발신측 알림 삭제 상태를 통한 삭제 프로세스 구현
CI/CD	AWS EKS에 서비스 배포, Jenkins, ArgoCD를 통한 배포 자동화

시스템 아키텍쳐



HeRoes 깃허브 레포지토리 바로가기

개선 및 문제 해결 사례 - 1

1. 문제 상황

Spring Batch를 통한 데이터 처리 중 엔티티에 Null 값이 반환되는 이슈

2. 원인

JPA를 통해 데이터를 관리하며 영속성 컨텍스트가 엔티티의 생명주기를 관리하고 있었는데, 배치 작업 중 데이터 처리 로직이 끝나게 되면 엔티티 영역에서 DTO 변환 처리를 했기 때문에 값을 반환 할 때 엔티티 생명주기가 끝나게 되며 데이터가 유실되어 Null 값이 반환 됨

3. 해결

기존 DTO 변환을 엔티티 영역에서 서비스 영역으로 옮겨 DTO 변환 작업을 엔티티의 생명주기가 관리되는 서비스 계층으로 이동하여, 배치 작업 중 엔티티가 Detached 상태로 전환되며 발생하는 Null 반환 문제를 방지하는 방안을 통해 해결

4. 평가

- 코드 유지보수성 향상
  - DTO 변환 로직을 서비스 계층으로 이동함으로써 엔티티와 로직 간의 의존성을 줄이고 계층 분리가 명확해져 코드의 가독성과 유지보수성이 향상
- 데이터 신뢰성 확보
  - 배치 작업 중 발생하던 엔티티의 생명주기 종료로 인한 Null 반환 문제가 방지되어 데이터 처리 결과의 정확성과 신뢰성 확보

개선 및 문제 해결 사례 - 2

1. 문제 상황

대량의 알림 발송 및 삭제 시 API 요청으로 인해 서버 과부하와 지연 문제가 발생

2. 원인

알림 발송 및 삭제 작업이 단일 스레드로 처리되고 있어, 메인 스레드가 모든 API 요청을 처리하고 있으며, 처리할 알림이 많을수록 처리 시간 증가와 서버 과부하가 발생

3. 해결

다수의 알림 발송 및 삭제 처리 로직을 @Async와 쓰레드 풀 설정을 통해 비동기 처리하도록 수정하였으며, 그 결과 서버 과부화 문제가 개선

4. 평가

- 성능 개선
  - 다수의 알림 발송 및 삭제 작업을 비동기 처리함으로써, 처리 시간이 99% 이상 개선되었으며, 서버 과부하 문제도 해결
- 확장성 증대
  - 비동기 처리로 대규모 알림 처리 시에도 성능 저하 없이 확장 가능해졌으며, 트래픽 증가에 대비한 시스템 확장성이 확보

개선 및 문제 해결 사례 - 3

1. 문제 상황

AWS EKS에 애플리케이션을 배포한 후 ALB를 통해 도메인 기반 접속을 설정했으나 접근이 되지 않는 문제가 발생

2. 원인

EKS 포드의 상태와 보안 설정에는 문제가 없었으나, 사용하려던 도메인이 ALB의 엔드포인트와 올바르게 연결되지 않아 접근이 불가능한 상태였고. 이 문제는 Route 53이나 다른 DNS 관리 시스템에서 관리하지 않았던 도메인이었기 때문

3. 해결

Cert-Manager와 Route 53을 통해 도메인을 올바르게 등록하고, Route 53에서 라우팅 처리로 ALB 엔드포인트와의 연결을 설정하여 도메인 기반 접속 문제를 해결

4. 평가

- 시스템 안정성 향상
  - 도메인 기반 접속 문제를 해결함으로써 AWS EKS에서 배포된 애플리케이션에 안정적으로 접근할 수 있게 되어 시스템의 신뢰성 향상
- 확장성 증대
  - SSL 인증서 관리와 도메인 라우팅을 자동화함으로써, 보안 및 관리 효율성을 높임

KIM JONG WON PORTFOLIO

OMOS

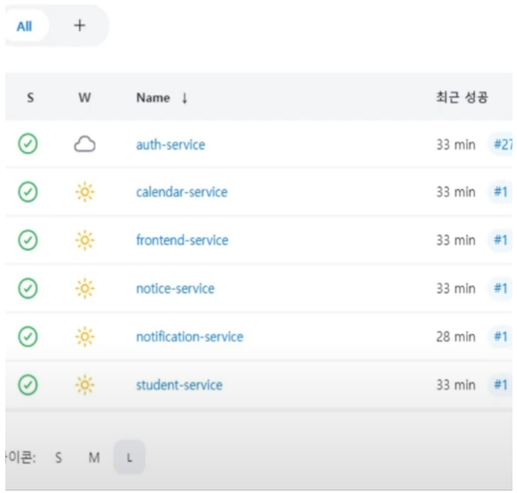
DEVOPS 프로젝트

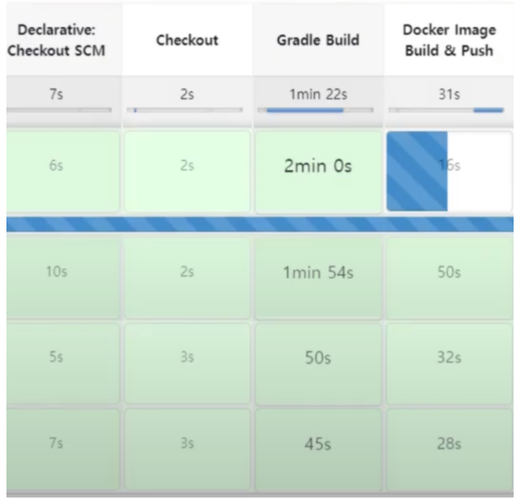
02

프로젝트 소개

OMOS(Devops 프로젝트)

한화시스템 BEYOND SW CAMP 8기	사용 기술 : Java, SpringBoot, Spring Data JPA, Spring Security, Vue.js, MariaDB Docker, Kubernetes, Jenkins, MSA	2024.08.26 ~ 2024.09.09





한화시스템 BEYOND SW CAMP에서 진행한 Devops 프로젝트로, 출결 관리 시스템을 MSA 운영 환경에서 Jenkins를 통해 배포 자동화 파이프라인을 구축하는 프로젝트로 기능적인 부분 보다 CI/CD에 중점을 둔 프로젝트

Java 기반 Spring Boot 서버와 Vue.js 기반 클라이언트로 구현한 프로젝트로 On-Premise 환경에서 Kubernetes 클러스터를 구축하여 어플리케이션을 배포하고 Github와 Jenkins를 통한 CI/CD를 구현

OMOS 깃허브 레포지토리 바로가기

담당 및 시스템 아키텍쳐

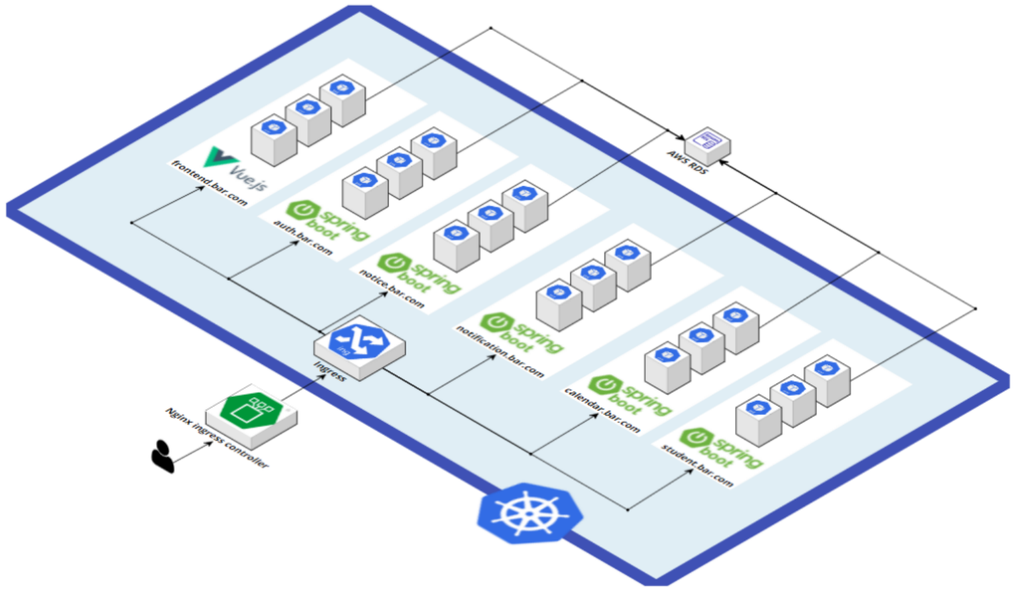
팀원 구성 및 역할

Full Stack 5명 팀장

담당 개발 업무

CI/CD	ON-PREMISE K8S 운영 환경에 MSA 운영 환경 구성
	K8S에 Jenkins 서버를 구성하여 Github 웹훅 기반 CI/CD 구축
사용자 인증	Spring Security, JWT 기반 사용자 인증, 이메일 인증 구현
	Spring Scheduler 기반 인증 데이터 삭제 로직 구현
	Spring Security의 OAuth2 기반 소셜 로그인 구현

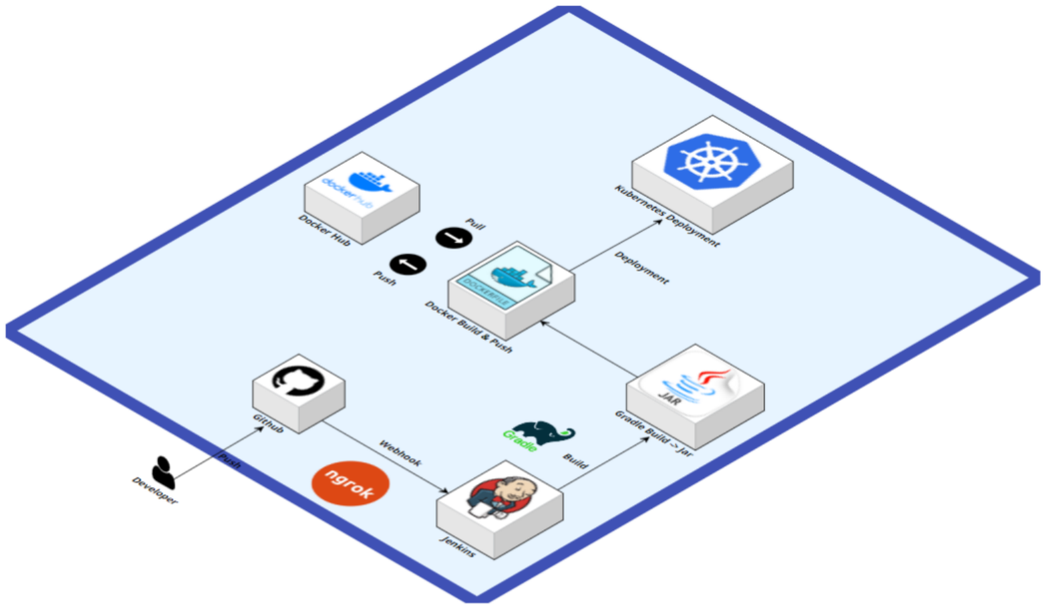
시스템 아키텍쳐



OMOS 깃허브 레포지토리 바로가기

CI/CD 아키텍쳐 및 설명

CI/CD 아키텍처



CI/CD 상세 설명

1. Github 변경 사항 감지
- 소스코드를 Github에 푸시하면, 설정된 Webhook을 통해 Jenkins가 변경 사항을 감지.

2. Jenkins Pipeline 실행
- Jenkins가 Kubernetes에서 새로운 Pod를 생성하여 CI/CD 작업을 진행.
- Jenkins는 Github에서 소스코드를 클론하고, application-private.yaml을 다운로드.

3. 빌드 및 테스트
- Jenkins는 자동으로 빌드 및 테스트를 실행.

4. Docker 이미지 빌드 및 푸시
- 빌드된 .jar 파일을 Docker 이미지로 변환하고 Docker Hub에 푸시.

5. Kubernetes 배포
- 최신 Docker 이미지를 Kubernetes 클러스터에 배포하여 컨테이너 오케스트레이션 실행.

OMOS 깃허브 레포지토리 바로가기

개선 및 문제 해결 사례

1. 문제 상황

MSA를 통한 도입으로 서비스가 독립적으로 분리됨에 따라 클라이언트와 API 서버사이의 API를 요청할 때 브라우저의 CORS 문제가 발생

2. 원인

MSA 환경에서 클라이언트와 API 서버가 서로 다른 도메인을 사용함에 따라 브라우저의 동일 출처 정책에 의해 CORS 요청이 차단되어서 문제가 발생되는 원인을 발견

3. 해결

CORS 문제를 해결하기 위해 Nginx Ingress Controller를 활용하여 클라이언트 요청을 처리하도록 Ingress 설정에서 필요한 CORS 헤더를 명시하여 브라우저가 CORS을 허용하도록 조치하였다.

4. 평가

- 문제 해결
  - 클라이언트와 API 서버 간 CORS 문제가 해결되어 정상적인 데이터 통신이 가능해짐
- MSA 도입
  - Nginx Ingress Controller를 활용해 효율적으로 CORS 설정을 적용했으며, Kubernetes 환경에서 관리의 일관성을 유지

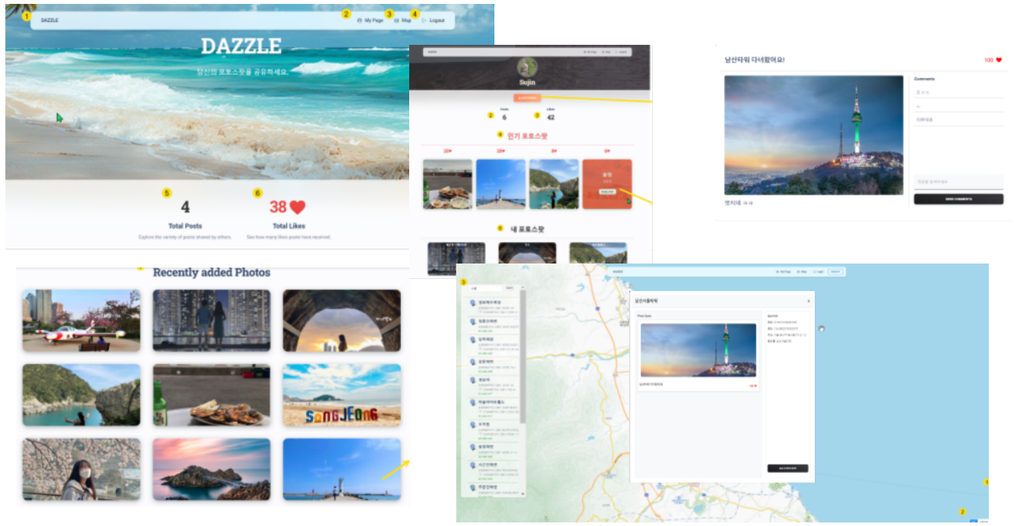
DAZZLE

포토스팟 공유 플랫폼

프로젝트 소개

DAZZLE(포토스팟 공유 플랫폼)

한화시스템 BEYOND SW CAMP 8기	사용 기술 : HTML/CSS, JavaScript, Vue.js, BootStrap5, Firebase(Authentication, Firestore, Stroage), Vercel	2024.08.05 ~ 2024.08.22



한화시스템 BEYOND SW CAMP에서 진행한 프론트 엔드 프로젝트로, 사용자들이 촬영한 사진을 특정 장소와 함께 공유할 수 있는 웹 서비스로 단순히 사진을 공유하는 것을 넘어서, 장소에 대한 정보와 개인적인 경험을 함께 나누는 공간을 제공하는 것을 목표로 한 포토 스팟 공유 플랫폼.

Firebase와 Vue.js 기반 클라이언트로 구현한 프로젝트로 데이터 관리 Firebase의 Storage와 Firestore을 통해 관리 하였으며 사용자 인증은 Authentication을 사용 Axios를 통해 Firebase와 통신하며 Pinia와 LocalStorage를 사용한 데이터를 관리 하였고 Vercel을 사용하여 서비스를 배포

DAZZLE 깃허브 레포지토리 바로가기

담당 및 시스템 아키텍쳐

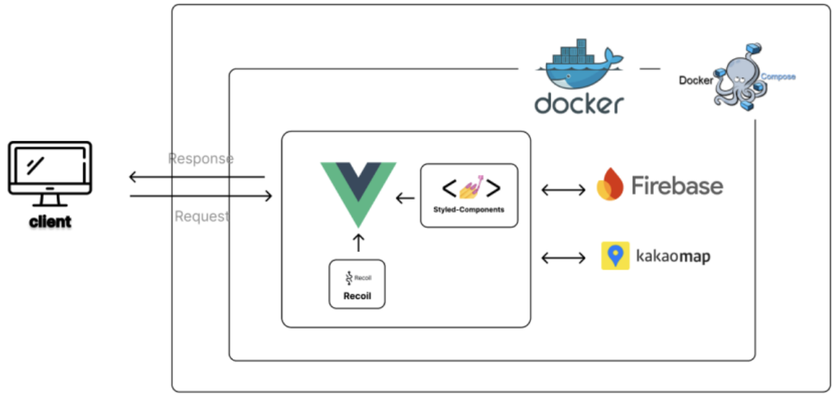
팀원 구성 및 역할

Frontend 5명 팀원

담당 개발 업무

사용자 인증	컴포넌트 기반 회원가입, 로그인 MODAL 화면 구현
	Firebase의 Authentication, Firestore를 통해 사용자 관리
포토 스팟 관리	특정 좌표의 포토 스팟 리스트 조회, 등록 화면 구현
	포토 스팟 상세 MODAL 화면 구현
	Firebase의 Firestore와 Storage를 통해 포토스팟 관리
마이 페이지	내 정보 페이지 및 내 정보 수정 MODAL 구현
	인기 포토스팟 4개 및 모든 포토스팟 리스트 조회 컴포넌트 구현

시스템 아키텍쳐



DAZZLE 깃허브 레포지토리 바로가기

개선 및 문제 해결 사례

1. 문제 상황

애니메이션 없이 이미지들을 정적으로 보여주기 때문에 사용자 경험이 단조롭고, 플랫폼에 대한 흥미를 끌기 어렵다는 이슈가 발생

2. 원인

기존 구현된 컴포넌트들이 이미지 전환이나 동적 요소가 부족하여 사용자 상호작용 요소가 적고, 정적인 화면만 존재하는 상황으로, 전체적인 디자인 수정의 필요성이 생김

3. 해결

AOS(Animate On Scroll) 패키지를 적용시켜 스크롤에 따라 애니메이션 효과를 추가하고, 이미지 전환과 동적 요소를 구현함으로써 사용자 경험을 향상시키고 문제를 해결

4. 평가

- 사용자 경험 향상
  - 사용자에게 보다 동적이고 직관적인 인터페이스를 제공하여 탐색의 재미와 몰입감을 증가
- 콘텐츠 매력도 상승
  - 포토 스팟 이미지들을 보다 효과적으로 전달할 수 있게 되어 사용자들의 관심을 끌 수 있었습니다.

KIM JONG WON PORTFOLIO

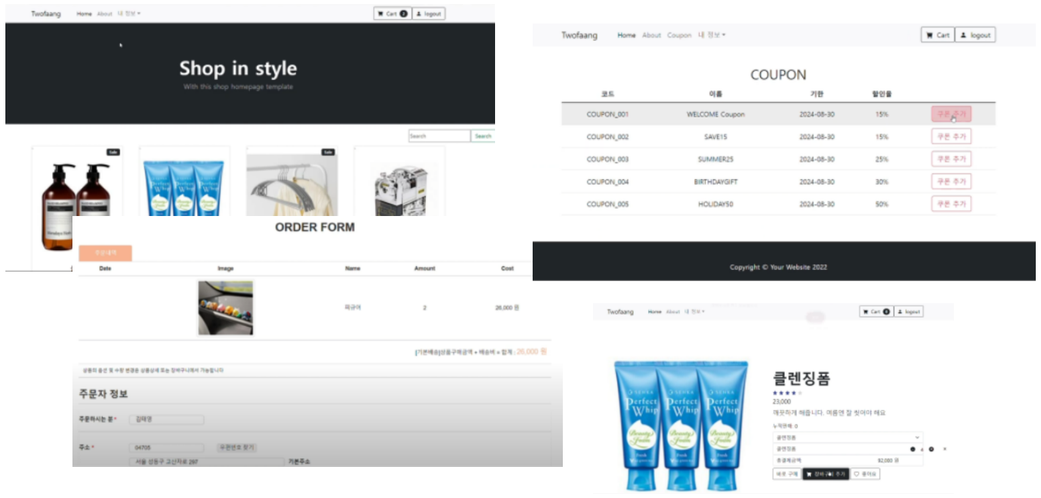
TwoFaang

쇼 핑 몰 프 로 젝 트

프로젝트 소개

Twofaang(쇼핑몰 프로젝트)

한화시스템 BEYOND SW CAMP 8기	사용 기술 : Java, SpringBoot, Spring Data JPA, Spring Security, Thymeleaf, MariaDB	2024.06.13 ~ 2024.07.31



한화시스템 BEYOND SW CAMP에서 진행한 백엔드 프로젝트로, 회원 관리, 상품 구매, 리뷰 및 문의 기능을 제공하여 사용자에게 효율적인 쇼핑 환경을, 관리자에게는 편리한 관리 도구를 지원하는 온라인 쇼핑몰

Java 기반 Spring Boot 서버와 Thymeleaf로 서버사이드 렌더링 화면을 구축했으며, JPA를 활용해 관계형 데이터베이스를 관리하고 Spring Security로 사용자 인증을 구현

HTML 폼을 통한 사용자 입력 데이터를 서버로 전송하는 방식으로 API와 통신하며, 이를 기반으로 데이터 관리 구조를 설계

Twofaang 깃허브 레포지토리 바로가기

담당 및 시스템 아키텍쳐

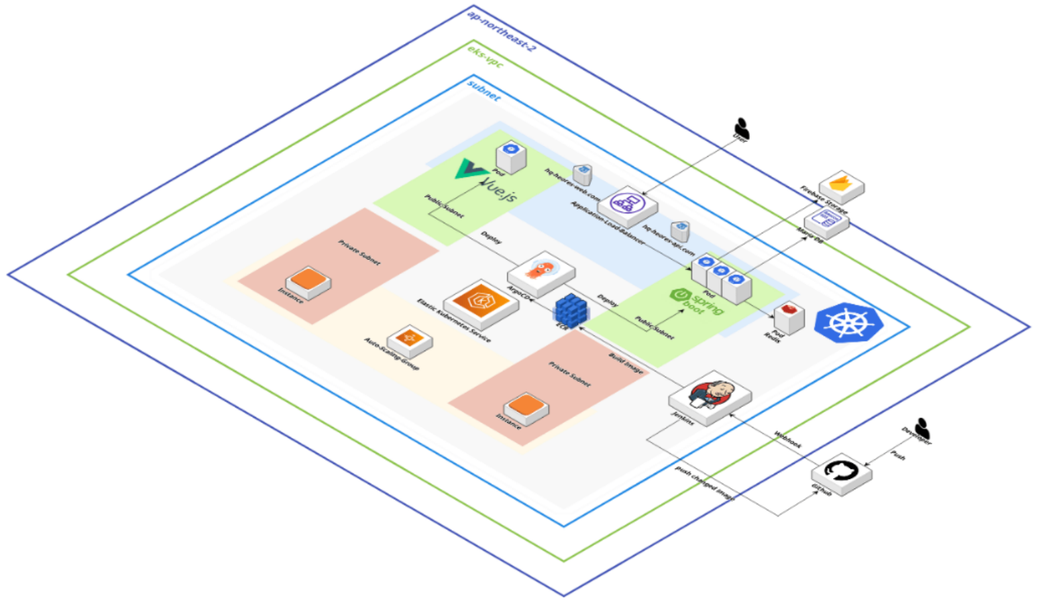
팀원 구성 및 역할

Backend 6명 팀장

담당 개발 업무

사용자 인증	회원 로그인 화면, 비밀번호 재발급 화면 구현
	Spring Security, JWT 기반 사용자 인증, 이메일 인증 구현
	Spring Security의 OAuth2 기반 소셜 로그인 구현
	Spring Scheduler 기반 인증 데이터 삭제 로직 구현

시스템 아키텍쳐



Twofaang 깃허브 레포지토리 바로가기

개선 및 문제 해결 사례

1. 문제 상황

로그인 성공 후 로컬스토리지에 토큰을 저장하고 이를 인증을 처리하려고 했으나, SSR 환경에서는 이 방식이 작동하지 않는 문제가 발생

2. 원인

SSR에서는 서버에서 클라이언트 상태 정보를 유지하거나 접근할 수 없어, 서버에서 로컬스토리지에 접근하거나 AccessToken을 헤더에 포함하지 못하는 원인을 발

3. 해결

AccessToken을 로컬스토리지 대신 쿠키에 저장하여 SSR 환경에서도 인증 요청이 가능하도록 수정하고 API 요청 시 별도의 처리 없이 브라우저가 자동으로 쿠키를 포함하여 요청하도록 설정해서 해결

4. 평가

- SSR 환경 적합성 향상
  - 로컬스토리지가 아닌 쿠키를 이용해 SSR에서도 클라이언트 상태를 유지할 수 있어, JWT 기반 인증이 안정적으로 동작

검토해 주셔서 감사합니다.

THANK YOU

KIM-JONGWON

블로그 바로가기

깃허브 바로가기