

유희진(Yu Hee Jin)

스타트업에서의 실무 경험으로, 기술적인 관점을 넘어 비즈니스적인 관점에서 고객 가치를 최우선으로 생각하고 개발할 수 있습니다.

어떤 기능을 우선적으로 개발할지 우선 순위를 정해 효율적으로 개발할 수 있습니다.

새로운 기술을 배우는 것에 흥미가 있으며, 사내에 필요한 기술이 있다면 빠르게 배우고 실무에 적용할 수 있습니다.

Contact

Email : yuheejin0914@gmail.com

phone: 010-2490-7323

GitHub: <https://github.com/yu-heejin>

블로그: <https://medium.com/@hee98.09.14>

사용 기술

Language	Java, Javascript, Typescript, Python, Kotlin
Backend	Spring Boot(JPA, Querydsl), Node.js express, Flask
Database	MySQL, MongoDB
DevOps/Infra	Docker, GitHub Actions, Nginx, AWS, Linux
Frontend	React, React Native

경력

뷰컴즈(주) - Backend Engineer / 인턴 및 정규직 / 2023.03.02 ~ 2023.10.31

[상세 경력기술서 바로가기 링크](#)

Node.js express , MySQL , MongoDB , React , AWS , GitHub Actions

웹소셜 연재 플랫폼 신규 프로젝트 사업 기획 및 개발 / 정규직 / 2023.07 - 2023.10

- 실제 웹소셜 작가와 소통하는 타 부서(유통팀, 편집팀)와 협업으로 신규 프로젝트 핵심 기능 우선순위 선정 및 프로토타입 구현
 - 프로토타입 구현으로 핵심 기능에 대한 베타 테스트 및 사용자 피드백 수집
 - 개발 초기 단계에서 사용자의 요구 사항과 개발 방향성 조정으로 기획 최적화 가능
- AWS(EC2, DocumentDB, S3), Node.js express 기반 전반적인 개발 환경 및 인프라 설계
 - MongoDB 서버에 해킹 공격이 발생하여, VPC 망 분리와 Database를 Private Subnet으로 이동시키고 인바운드 규칙을 재설정하여 문제 해결
 - 고도화된 인프라 재구축 및 보안 강화로 서비스 안정성 향상
- Node.js express, Mongoose ODM 환경에서 웹소셜 회차 및 댓글 RESTful API 개발
- Next.js, Chakra UI를 활용하여 웹소셜 회차 및 댓글 화면 퍼블리싱 및 Backend 서버와 API 연동

Notion Database API를 활용한 문의 사항 관리 서버 구축 / 정규직 / 2023.07

- Notion Database API를 활용하여, admin 사이트에 기존 서비스 관련 문의 사항 작성 시 팀 Notion 페이지에 자동으로 업로드 되는 기능 개발
 - 문의사항에 대한 문서 자동화로 효율적인 업무 분배 및 유지보수 가능
- Notion 페이지 업로드 시, 팀 메신저로 알림이 오도록 Teams Alert bot을 연동해 실시간으로 업무를 처리할 수 있도록 구현
 - 긴급한 문의 및 오류 사항의 경우 알림 기능을 통해 실시간으로 문제 해결
- 기존 서비스의 불편 사항을 빠르게 개선할 수 있어 서비스 품질 향상 및 수익성 기여

레거시 Backend 서버 리팩토링 프로젝트 / 인턴 / 2023.05 - 2023.06

- Javascript로 작성된 코드를 Typescript로 마이그레이션하여 데이터 타입 안정성 확보
- 기존 Backend 구조를 MVC 구조로 분리하여 전반적인 프로젝트 구조 리팩토링
 - 기존 메소드 하나에 300줄 이상인 코드를 평균 40줄 이내의 함수로 분리하여 책임 분리 및 가독성 향상
 - 중복되는 로직에 대한 모듈화를 통한 코드 재사용성 증가
- 불필요한 table JOIN, 삼중 반복문 등의 로직을 개선하여 코드 및 API 응답 속도 개선
 - 여러 데이터를 Insert하는 경우, 반복계 쿼리로 for문을 통해 Insert하는 로직이 있어 Bulk Insert 쿼리로 변경
 - Update 로직의 경우, 기존 데이터를 Delete 후 Insert하는 로직을 UPDATE, ON DUPLICATED KEY UPDATE, INSERT IGNORE 등의 쿼리로 변경
 - 삼중 반복문, 중복되는 데이터 처리 로직 삭제 및 개선으로 코드 품질 향상
 - 일별 판매량 데이터 그래프 조회 시간의 경우 응답 시간 평균 2s → 2.574ms로 개선
- 테스트 코드 작성으로 로직 개선에 따른 테스트 결과를 예측하고 오류에 대한 안정성 확보
- Swagger API 문서화를 통해 Endpoint와 Request/Response 데이터를 시각적으로 확인하여 원활한 협업 가능

Node.js, Redis Pub/Sub을 이용한 실시간 채팅 서버 구축 / 인턴 / 2023.03 - 2023.04

- Redis를 Message Broker로 활용하여, 특정 채팅방을 구독(sub)한 사용자에게 메시지를 전송(pub)하도록 채팅 서버 구축
 - 작가의 문의에 대한 실시간 커뮤니케이션이 가능하여 사용자 경험 개선
- Publish된 채팅 메시지에 대한 실시간 응답을 위해 알림 기능 구현
 - admin 시스템에 로그인 한 직원에게는 채팅 메시지에 대한 Desktop 터미널로 알림 전송 기능 구현
 - 로그인하지 않은 직원의 경우, 직원의 이메일로 메일을 전송하는 기능 구현
 - 실시간 알림 기능 추가로 더욱 빠른 문의 처리 가능하여 고객 편의성 증가
- MacOS의 경우, MacOS 11 이상의 환경에서 보안 문제로 인해 알림 시 금지 아이콘(🔒)이 표시되어 사용자에게 혼란을 주는 문제 발생
 - 오픈 소스 패키지인 node-notifier 모듈을 직접 custom하여 해당 문제 해결

프로젝트

DEVBRIDGE (2023.12.19 ~)

깃허브 주소

- <https://github.com/devridge-team-project>

배포 주소

- <https://devridge-client.vercel.app/>

프로젝트 소개



개발자들을 이어주는 다리, DEVBRIDGE

- 개발자들을 위한 지식 공유 플랫폼 웹 사이트입니다.
- 커피챗, Q&A, 스터디/프로젝트 모집과 같은 개발자들을 위한 다양한 기능들을 제공합니다.

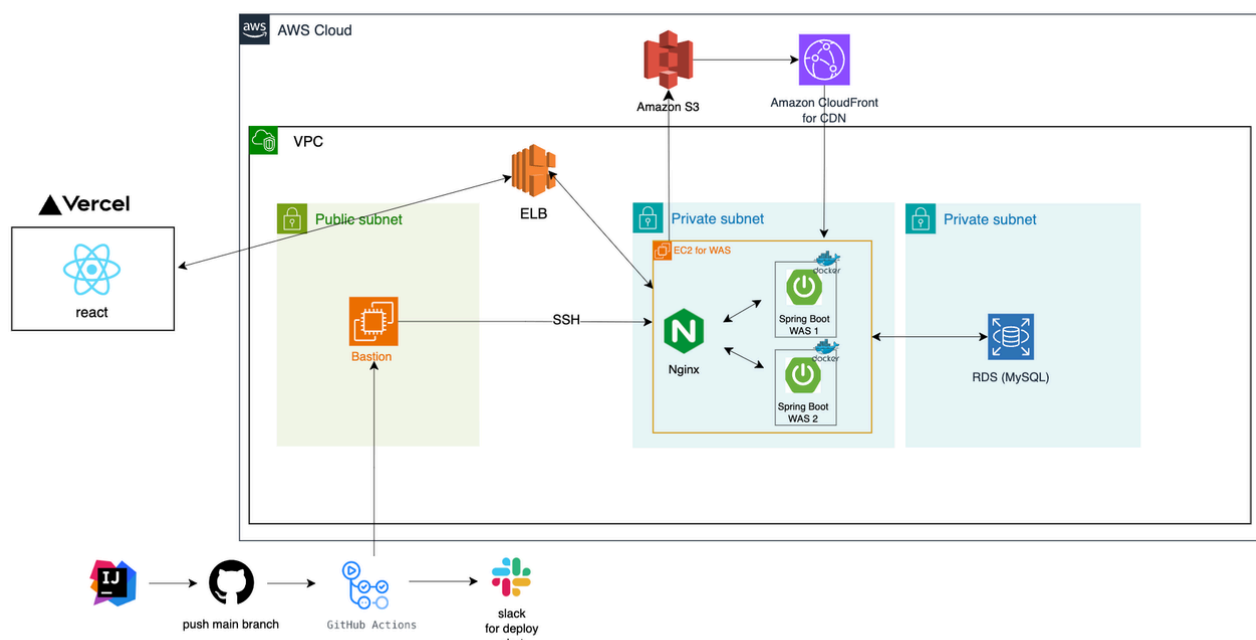
프로젝트 관련 링크

- [Spring Boot에서 Docker 이미지 최적화하기](#)
- [Nginx CORS + https 설정 문제 해결하기](#)
- [\[Q&A 추천/비추천, Q&A 스크랩\] 바인딩 변수 사용, 쿼리문 및 로직 개선](#)
- [GlobalExceptionHandler 리팩토링](#)
- [Server Production 환경 구축하기\(1\) — VPC와 서브넷으로 EC2 추가하기](#)
- [Server Production 환경 구축하기\(2\) — 무중단 배포 도입하기 + GitHub Actions에서 Docker build 시 Cache 적용하기](#)
- [무한 스크롤 기능 구현하기 \(with No Offset\)](#)
- [STOMP와 JWT Token을 활용한 채팅 서버 구축](#)

역할

- Backend
 - GlobalExceptionHandler으로 Requet 데이터에 대한 예외 처리를 통해 시스템 안정성을 높였습니다.
 - S3 Bucket과 연동하여 이미지 객체를 S3에 저장 및 삭제할 수 있는 이미지 서빙 API를 구현했습니다.
 - 이미지를 DB에 직접 저장할 때보다 데이터를 효율적으로 관리할 수 있는 시스템을 구축했습니다.
 - Web Socket과 STOMP pub/sub을 활용한 커피챗(실시간 채팅) 기능을 구현했습니다.
 - JPA, Querydsl을 활용하여 Q&A 게시판 및 댓글 CRUD, 추천/비추천, 스크랩 API를 개발했습니다.
 - 코드 리뷰 문화 도입으로 소프트웨어 품질 향상 및 활발한 소통 문화를 조성했습니다.
 - [129개의 코드 리뷰 과정 바로가기 링크](#)
 - [58개의 코드 리뷰 과정 바로가기 링크](#)
- DevOps
 - GitHub Actions, Nginx, Docker를 활용한 무중단 배포 CI/CD 환경을 구축하여 끊임 없는 서비스 자동화 배포 환경을 구축했습니다.
 - Docker Container를 이용한 Spring Boot 애플리케이션 배포 시 **Layered jar**를 사용하여 배포 속도를 최적화했습니다.
 - 빌드 시간을 평균 6초에서 1초로 감소시켰고, 컨테이너 용량을 약 700MB 크기에서 200MB 크기로 감소시켰습니다.
 - GitHub Actions로 Docker 컨테이너 빌드 시 docker/build-push-action Actions를 사용하여 각 레이어 빌드 시 캐싱 기능을 활용해 전체 배포 속도를 약 30초 정도 감소시켰습니다.
 - VPC, Public/Private subnet을 분리하고, Bastion Host를 이용하여 보안성이 강화된 운영 서버를 구축하였습니다.

프로젝트 구조



사용 기술



Spring Boot를 채택한 이유는 의존성 주입(DI)를 활용해 외부에서 객체의 관계를 결정해 주어서 의존관계가 고정되지 않아 유연성을 확보하고 결합도를 낮출 수 있게 해주고, 개발자가 직접 객체를 관리할 필요가 없기 때문에 편리하게 개발할 수 있다는 장점이 있어 채택하였습니다.

- Frontend: React, TypeScript, Tailwind CSS, react-query, zustand
- Backend: Spring Boot 2.7.5, JDK 11
- DevOps: Docker, Nginx, GitHub Actions, AWS Service (EC2, S3, RDS, CloudFront)
- Database: MySQL

Ladder(Line And Drawing, Draw Especial Recollection) (2022.08.27 ~ 2022.10.01)

깃허브 주소

- <https://github.com/2022-SeongNam-Team-C>

프로젝트 소개



특별한 순간을 더 특별하게 만들어드립니다! 추억을 이어주는 사다리가 되어드릴게요 💕

[프로젝트 상세 소개 바로가기 링크](#)

[데모 영상 바로가기 링크](#)

- 최근 유행하는 '라인드로잉' 기법에서 영감을 얻어 기획하였습니다.
- 사이트에 접속 후 원하는 사진을 업로드 하면, AI 모델이 사진 속 사람을 인식하여 그림처럼 만들어드립니다.

역할

- Backend
 - Flask Email 모듈을 이용하여 변환된 사진을 사용자 이메일로 전송하는 API를 구현해 서비스 사용자 경험을 향상시켰습니다.
 - JWT + Redis를 이용하여 로그인, 로그아웃 기능을 구현함으로써 사용자 정보에 대한 보안성을 강화했습니다.
 - Stateless, 확장성, 보안을 고려하여 JWT token을 이용해 API 요청에 대한 인증 및 인가를 처리했습니다.
 - Redis를 활용한 블랙리스트 기법으로 로그아웃 인증을 관리했습니다.
 - AI 서버와 Backend 서버 사이의 이미지 응답 처리 API를 구현하고, S3 bucket에 처리된 이미지 사진을 업로드하는 기능을 구현해 오류 발생에 대비한 이미지 백업 환경을 구축했습니다.
- ▼ 트러블 슈팅
 - 아래 코드(`cv2.imencode`)의 경우, 이미지 변환 결과가 numpy였기 때문에 S3 Bucket에 변환 결과가 저장되지 않는 문제가 발생했습니다.

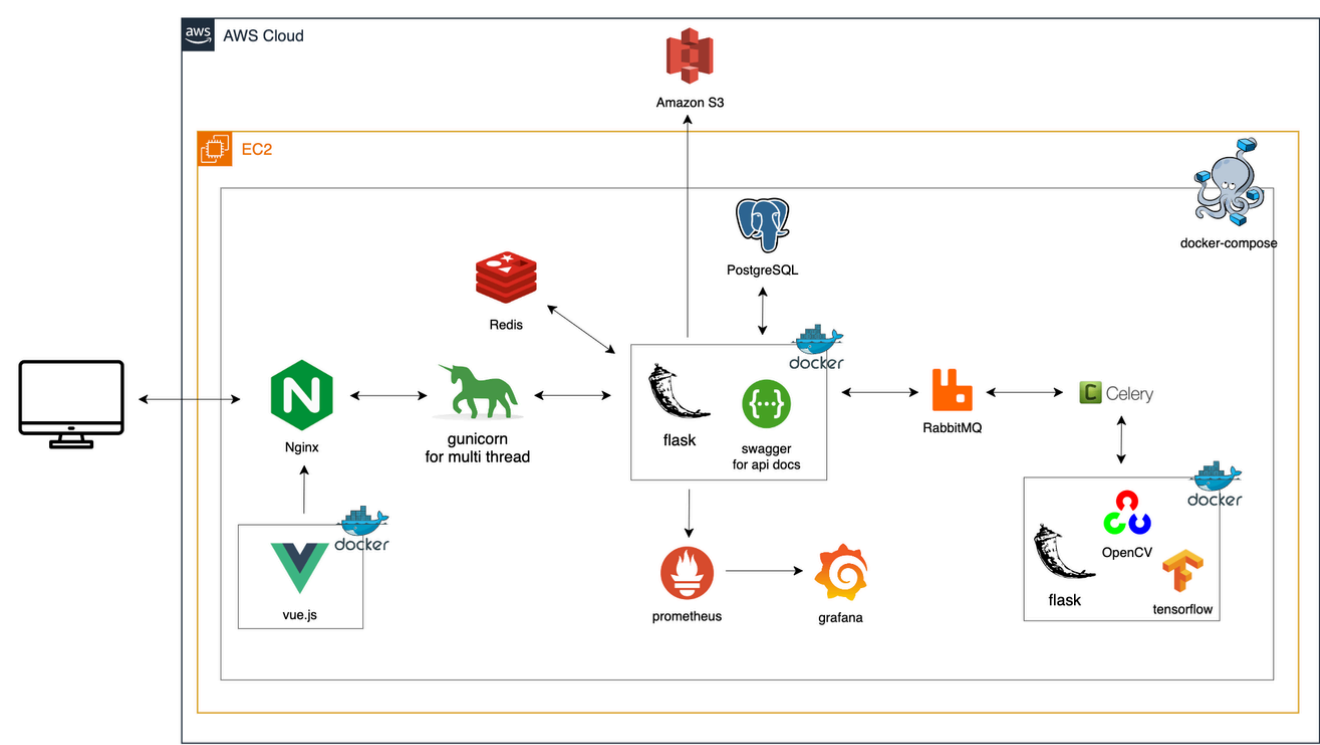
```
plt.figure(figsize=(12, 12)) # fix_img = cv2.cvtColor(result, cv2.COLOR_BGR2RGB) # result
image # plt.imshow(fix_img) days = datetime.today() file_name = days.strftime('%Y-%m-%d-%H-%M-%S') + '.jpg' print(file_name) cv2.imwrite(file_name, result) s3_put_result_image(s3, 'ladder-s3-bucket', cv2.imencode(result), file_name)
```

- `tobytes()` 함수를 사용하여 해당 문제를 해결했습니다.




```
data = cv2.imencode('.jpeg', result)[1].tobytes() print(data) # s3, bucket, file, filename
s3.put_object( Body = data, Bucket = 'ladder-s3-bucket', Key = f'result/{file_name}',
ContentType = 'image/jpeg' )
```

- DevOps
 - Vue + Nginx Dockerfile을 활용하여 Frontend 개발 환경을 통일하고 더 편리한 배포가 가능한 환경을 구축했습니다.
 - nginx.conf 설정을 통한 Reverse Proxy 기능을 구현해 포트 번호의 노출을 막고 보안성을 강화했습니다.
 - Prometheus/Grafana node exporter를 활용한 서버 시스템(CPU, Memory 등) 정보 수집 및 시각화를 진행했습니다.

프로젝트 구조



사용 기술

-  **Flask**는 짧은 시간 동안 빠른 개발을 할 수 있다는 장점이 있으며, AI 모델과 가장 친숙한 언어인 python을 기반으로 동작하는 웹 프레임워크이기 때문에 채택하였습니다.
- 하지만 Flask의 경우 단일 프로세스로 동작하기 때문에 하나의 스레드가 여러 요청을 받을 경우 병목 현상이 일어날 수 있습니다. 따라서 멀티 스레드처럼 동작할 수 있도록 **WSGI인 Gunicorn**을 사용했습니다.
-  **Docker**는 다양한 프로그램들과 실행 환경을 컨테이너로 규격화시켜 프로그램의 배포 및 관리를 단순화하며, 프로그램의 코드뿐만 아니라 실행에 필요한 모든 dependency도 컨테이너에 담기 때문에 다른 환경에서도 빠르게 애플리케이션을 배포할 수 있다는 점에서 채택하였습니다.
-  **Nginx**는 리버스 프록시와 로드밸런싱 기능 구현이 가능하여 보안성을 높일 수 있다는 장점이 있어 사용했습니다.

Police In My Pocket (2022.04.15 ~ 2022.11.30)

깃허브 링크

- <https://github.com/hanium-project>

프로젝트 소개

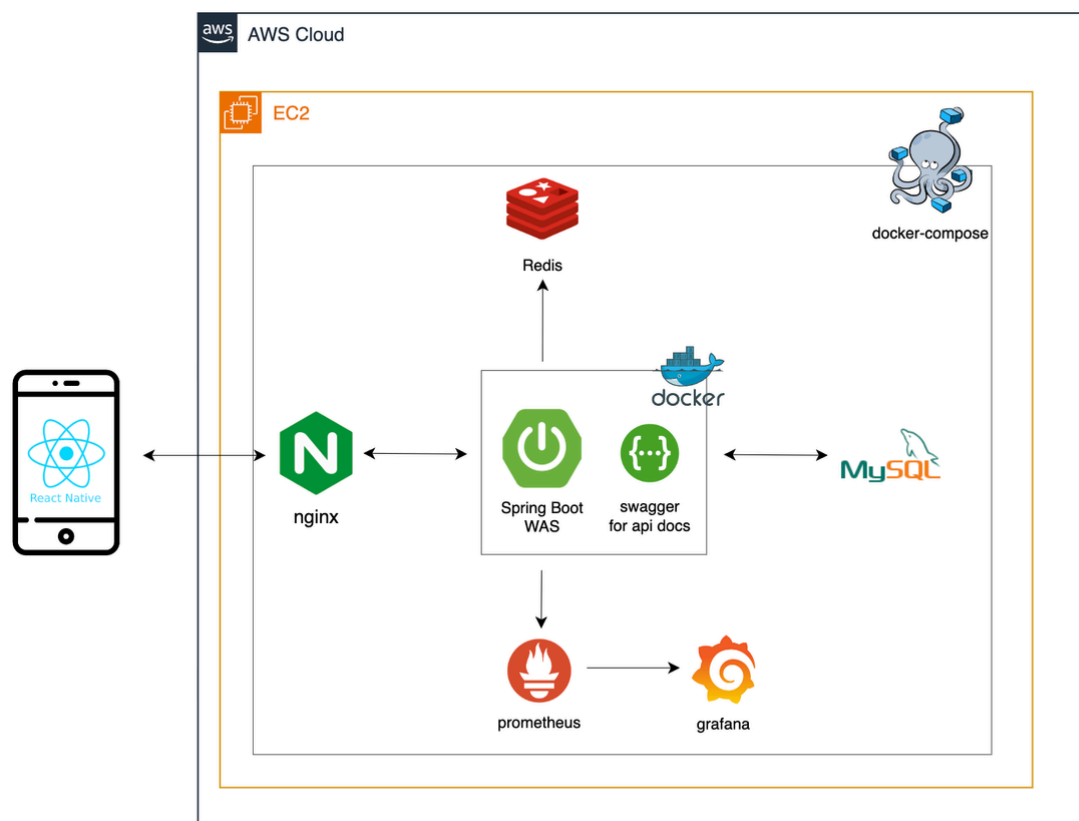
💡 내 손안에 작은 경호 어플리케이션 🚓

- 위급 상황 발생 시 사용자가 신고 버튼을 누르면 인근 파출소 및 사전에 등록한 가족 또는 친구들의 전화번호로 신고 문자를 전송합니다.
- 목적지까지 background로 앱을 실행하여, 사용자의 실시간 위치를 추적하고, 신고가 발생하면 신고가 발생한 위치를 데이터베이스에 저장합니다.
- 저장된 데이터를 통해 내 주변 위치 정보를 지도에 표시하여 위험 상황이 발생한 정보, 치안 시설 정보를 볼 수 있습니다.

역할

- Backend
 - Spring Data JPA를 활용하여 위험 지역, 유저 정보 등의 데이터 관리 API를 구현했습니다.
 - Nurigo Message API를 활용한 문자 전송 기능을 구현해 실시간 위급 상황에 대한 사용자 경험을 개선했습니다.
 - JWT + Redis를 활용하여 로그인 기능 및 블랙리스트 기법을 활용한 로그아웃 기능을 구현해 사용자 정보에 대한 보안성을 강화했습니다.
 - Spring Security를 활용하여 권한에 따른 API 요청을 제한함으로써 서버 및 데이터에 대한 보안성을 높였습니다.
 - Swagger 연동을 통한 API 문서화를 통해 Frontend와 Backend 사이의 더욱 효율적인 협업이 가능하도록 했습니다.
 - Http CORS 설정을 통한 SOP (Same-Origin Policy) 문제를 해결했습니다.
- DevOps
 - Dockerfile, docker-compose를 활용한 개발 환경을 통일하고 더 편리한 배포가 가능하도록 구현했습니다.
 - Nginx를 활용한 Reverse Proxy 환경으로 포트 번호의 노출을 막고 보안성을 강화했습니다.
 - Prometheus, Grafana를 활용한 데이터 모니터링 및 시각화를 통해 서버에 대한 상태를 실시간으로 파악할 수 있었습니다.

프로젝트 구조



사용 기술

- Frontend : React Native
- Backend : Spring Boot 2.7.0, JDK 11
- DevOps : Nginx, Prometheus, Grafana, Docker, docker-compose
- Database : MySQL

프로젝트 성과

- 제 11회 교내 소프트웨어 경진대회 최우수상 수상 (2022.11.24)
 - 실시간 위치 기반 경호 앱 서비스 'Police In My Pocket' 프로젝트 ([발표영상](#))
- 한국정보처리학회 ACK 2022 ICT멘토링 추계학술대회 논문 등재 (2022.11)
 - [긴급 상황 인식 및 자동 알림 신고 앱 시스템\(ID : KIPS_C2022B0250\)](#)

활동

성남시청소년재단 AI 프리인턴십 4기 / 2022.08.27 ~ 2022.10.01 / [프로젝트 Organization](#)

- 사진 속 사람을 인식하여 그림처럼 바꿔주는 웹 사이트 'Ladder(Line And Drawing, Draw Especial Recollection)' 프로젝트 개발
- Flask, Docker, Nginx 등을 활용한 Backend, DevOps 개발 담당

한국정보산업연합회 한이음 ICT 멘토링 프로젝트 / 2022.04.15 ~ 2022.11.30 / [프로젝트 Organization](#)

- 실시간 위치 기반 경호 앱 서비스 'Police In My Pocket' 프로젝트 개발
- Spring Boot, Docker, GitHub Actions를 활용한 Backend, DevOps 개발 담당

성남시청소년재단 AI 프리인턴십 3기 / 2022.01.17 - 2022.02.19 / [프로젝트 Repository](#)

- 지브리 캐릭터 테마 카메라 앱 서비스 'Ghibli Camera App' 프로젝트 개발 ([발표영상](#))
- React Native를 활용한 Frontend 개발 담당

교육

동덕여자대학교 / 정보과학대학 컴퓨터학과 이학사 / 2020.03 - 2024.02

4.05 / 4.5 (전공 평점 4.07)

교내 성적우수장학금 2회 수여 (2학년 1학기, 3학년 1학기)

자격증 및 수상

- SQLD(SQL 개발자) / 한국데이터산업진흥원 / 2023.10.06
- 제 11회 교내 소프트웨어 경진대회 최우수상 / 동덕여자대학교 / 2022.11.24
 - 실시간 위치 기반 경호 앱 서비스 'Police In My Pocket' 프로젝트 ([발표영상](#))
- 한국정보처리학회 ACK 2022 ICT멘토링 추계학술대회 논문 등재 / 2022.11
 - [긴급 상황 인식 및 자동 알림 신고 앱 시스템\(ID : KIPS_C2022B0250\)](#)