**Summary**

* **End to End** 프로젝트를 통해 기획부터 배포까지의 전체적인 개발 프로세스를 실제로 경험하였습니다.
* **Django, Spring Boot**를 통해 서버를 구축할 수 있습니다.(AWS 배포)

**Education**

부산대학교 정보컴퓨터공학부 (2019.03 ~ 재학 중)

**Tech Stacks**

* Language : Java, Python
* Backend : Spring Boot, Django
* Database: MySQL, PostgreSQL
* Devops: Docker, AWS(EC2, S3), Github Actions
* ETC:, RabbitMQ, Celery, Nginx

**Project**

[**preview**](https://github.com/2023-Winter-Bootcamp-Team-N/Preview)| 2023.12 ~ 2024.02

유튜브 영상 실시간 요약 및 요약본 생성 서비스 [medium](https://medium.com/@wndjs803/2023-winter-silicon-valley-bootcamp-preview-16a19066ddb5)

* 영상의 자막을 **Chat GPT**를 이용하여 각 시간대별 내용을 요약
  + 실시간으로 사용자에게 내용을 전달하기 위해 **Websocket** 통신 구축
* 요약내용을 바탕으로 이미지를 포함한 요약본 생성
  + 요약본 생성을 비동기로 처리하기 위해 **RabbitMQ**, **Celery** 활용
  + 요약본 내 이미지를 생성하기 위해 **FFmpeg**을 활용해 시간대별 이미지 추출 기능 개발
  + 특정 시간마다 요약본을 자동 생성하기 위해 **Django Scheduler**를 활용
* **Github Action**을 활용하여 **CI/CD** 파이프라인을 구축하여 배포 과정 자동화
* 보안 강화를 위해 **Nginx**를 활용하여 **리버스 프록시**, **HTTPS** 적용

기술 스택: Django, Chat GPT, AWS(EC2, S3), Nginx, RabbitMQ, Celery, FFmpeg, Docker

[**Voilio**](https://github.com/Team-Port)| 2023.09 ~ 2023.11

영상 기반 포트폴리오 플랫폼 서비스

* **Spring Websocket**을 활용하여 실시간 채팅 기능 구현
* **STOMP** 프로토콜을 사용하여 클라이언트와 브로커 간의 통신 처리 및 신뢰성 보장
* 검색 API 구현 시 **페이징**을 적용하여 응답 시간 감소
* QueryDSL을 통해 동적 쿼리를 활용한 검색 필터 개발

기술 스택: Spring Boot, Spring Data JPA, STOMP, MySQL, Docker

[**It’s ME**](https://github.com/2023-Summer-Bootcamp-Team-G) | 2023.06 ~ 2023.08

타인의 시선을 캐릭터로 시각화해주는 서비스 [medium](https://medium.com/@hgh1057/its-me-46d9de903a94)

* **Session**을 활용한 로그인 및 로그인 여부에 따른 예외 처리 구현
* 이미지 생성을 비동기로 처리하기 위해 **RabbitMQ**, **Celery** 활용
* **Uvicorn**을 도입하여 요청을 비동기로 처리하여 동시성을 향상(아직 측정을 못했습니다.)

기술 스택: Django, RabbitMQ, Celery, Uvicorn, Docker, MySQL

**Activities**

2023.08 ~ 진행중 Silicon Valley 개발자 멘토링 그룹 테커

2023.12 ~ 2024.02 [티타임즈 x 테커] 2023 동계 Silicon Valley 부트캠프 수료

2023.06 ~ 2023.08 [티타임즈 x 테커] 2023 하계 Silicon Valley 부트캠프 수료

[**Voilio**](https://github.com/Team-Port)/ Backend 2023.09 ~ 2023.11

영상 기반 포트폴리오 플랫폼 서비스

* **Spring Websocket**을 활용하여 실시간 채팅 기능 구현 [ [blog](https://jueon03.tistory.com/2) ]
* **STOMP** 프로토콜을 사용해 클라이언트, 브로커 간의 통신 처리 및 신뢰성 보장
* **QueryDSL**을 활용하여 **동적 쿼리**를 적용한 검색 API 개발 및 검색 결과 **페이징** 적용

기술 스택: Spring Boot, Spring Data JPA, QueryDSL, STOMP, MySQL, Docker