# 

| **김윤기 Backend Developer** | **Phone** | 010-3385-2507 |
| --- | --- | --- |
| **Email** | kalsagansteve@gmail.com |
| **Github** | [github.com/kalsteve](http://github.com/kalsteve) |
| **Blog** | [velog.io/@kalsteve98](http://velog.io/@kalsteve98) |
| 함께 개선할 수 있는 주제를 나누고, 항상 최선의 결과를 내기위해 책임감을 갖고 꾸준히 준비하며 개선점을 공유하고 이야기를 나누며, 피드백을 통해 매번 한 단계 성장하는 모습을 꾸준히 보여주는 사람입니다. | | |

# 

# 

## 

## 

## **Experience**

## **Tairot** **(2023.12.26 - 2024.02.03) 2023 Techeer S/W Winter BootCamp 수상작** 인격에 따라 스타일이 다른 **AI 타로 마스터**가 각자의 방식으로 타로를 풀이해주는 서비스 Node.js, Express, Socket.IO, MySQL, AWS(EC2, RDS, Secrets Manager), Jenkins, Docker, Nginx

* Socket.IO를 통해 **실시간 데이터 처리**

실시간 음성 데이터와 텍스트를 실제 타로 마스터가 말하는 것처럼 구현하기 위해 Socket.IO를 사용하여 빠르게 텍스트와 음성 정보를 전달하도록 하여 **자연스러운 타로마스터와의 대화**를 적용하였습니다.

**메시지큐?**

* 이미지 전송에 따른 **부하를 개선**하기 위해 AWS의 S3 사용

이미지를 서버에서 직접 전달하기 보다 문자열을 통해 S3의 이미지를 참조할 수 있게 주소를 전송하여 서버의 부하를 감소시켰습니다.

## **Brain Washer** **(2024.06.24 - 2024.08.12) -** [**link**](https://velog.io/@kalsteve98/%EA%BF%88%EB%A7%8C-%EA%BE%B8%EC%A7%80%EB%A7%90%EA%B3%A0-Brain-Washer)실존하는 특정 인물이 **독설**로 동기 부여해주는 서비스 FastAPI, MySQL, AWS(CloudFront, EC2, RDS, CodeDeploy), GitHub Actions, Docker, Redis, Nginx

* 실시간 채팅 경험을 향상 위한 **비동기 처리**

순차 처리 시 음성과 텍스트를 사용자가 대기해야 하는 문제를 파이썬의 비동기 방식인 Asyncio의 큐(queue) 방식을 통해 개선하였습니다. **( 3분대 -> 1분 이내)**

* **Nginx**를 통한 디도스 공격 방어

실제 도메인 주소로 서비스하게 되는 경우 DDoS 공격이 발생하여 문제가 있는 user agent **차단과 요청을 거부하고, 시간당 요청 수를 제한**하여 문제를 해결하였습니다.

* **Redis**를 통한 서버 부하 개선

실시간으로 생성된 비정형 데이터인 음성 데이터를 S3에 바로 전송하는 경우 요청량에 따른 비용과 서버 부하가 증가하기에 Redis에 **TTL(15분)을 사용한 캐싱**으로 사용자가 저장하려는 내용만 S3에 영구 저장하였습니다.

* 프로젝트 **파일 구조 명확화**

FastAPI의 파일구조가 자유로워 프로젝트를 시작하기 앞서 파일 구조가 명확하게 하여 프로젝트가 협업에 용이하도록 구조화하였습니다.

**관심사를 분리하여 모듈화** 할 수 있도록 model, database, router, service, schemas, config로 분리하였습니다.

## **시공** **(2024.05.16 - 2024.06.21)** 경매 방식을 적용한 **시공 업체 중개** 모바일 서비스 Spring Boot, MySQL, AWS(EC2, RDS,SecretsManager), GitHub Actions, Docker

* Github Actions를 통한 **배포 자동화 및 빌드 속도 개선**

프로젝트를 빌드 할 수 있는 서버를 제공하고 있는 Github Actions를 효율적으로 이용하기 위해 **Gradle 빌드 캐싱** 기능 적용과 **artifact** 기능을 적용하여 빌드 파일을 관리 하였습니다. **(약 2분에서 40초대로 감소)**

Pull & Request시에 **잘못된 빌드 파일**이 실제 서버에 적용하는 것을 막기위해 **빌드가 성공** 되었을 때만 Merge 하여 빌드된 파일을 배포하도록 workflow를 설정하였습니다.

## **Technical Skills**

| **Language** | C, C++, Java, JavaScript, Python, Shell Script |
| --- | --- |
| **Backend** | Node.js(Express), FastAPI, Spring Boot, MySQL, Redis |
| **Frontend** | HTML, CSS, React, Jetpack Compose, Android |
| **Infra** | AWS(CloudFront, EC2, RDS, CodeDeploy, route53, S3), Docker, Nginx, GitHub Actions, grafana, cadvisor, prometheus, slack |

## **Activity**

**Techeer 8기 (2024.08 - )**

**2024 하계 테커 실리콘밸리 소프트웨어 부트캠프 (2023.06.24 - 2024.08.03)**

**Techeer partners (2024.03.23 - 2024.06.23)**

* 실리콘밸리 엔지니어의 SW개발자 스터디 그룹
* 기술 세션, 프론트엔드 React 프레임워크등의 스터디 활동

**2023 동계 테커 실리콘밸리 소프트웨어 부트캠프 (2023.12.26 - 2024.02.03)**

**Techeer partners (2023.09.12 - 2023.12.23)**

* 실리콘밸리 엔지니어의 SW개발자 스터디 그룹
* 기술 세션, 백엔드 Spring Boot 프레임워크등의 스터디 활동

## **Education**

한국공학대학교 (2023.3 - 2025.8)

임베디드시스템전공 (3.53 - 4.5)