

Sirong



Resume Blog

Résumé

스펀지 같은 개발자 신원균 입니다.

[> Download as PDF](#)

SKILLS

Language

Java, dart, JavaScript, HTML5, C#, C/C++

Back-End

Spring Framework (MVC, JPA, Batch, Security, Actuator, Cloud, WebFlux ...)

MQ (Kafka, RabbitMQ)

Monitoring (Prometheus, Grafana, pinpoint, nGrinder, Datadog)

Front-End

Flutter, React.js, Vue.js

DevOps

AWS(EC2, RDS, S3, CodeDeploy), GithubActions, Jenkins, ArgoCD, nginx

DB

PostgreSQL, MySQL, MSSQL, H2

OS

MacOS, Linux, Windows

Tools

IntelliJ, Postman, MySQL workbench, VS code, DBeaver, pgAdmin4, Android Studio

Collaborations, Document

Git, Github, bitbucket, Jira, Notion, Slack, Google Workspace, MS office

Work

AITStory

Tmap 전기차 충전 서비스 팀(BE)

2023.09 ~ 2023.12

학부 4-2 학기에 ICT 인턴십으로 근무했던 회사입니다. 해당 회사에서 백엔드 개발자로 일했었고 tmap 내의 전기차 충전 플랫폼의 백엔드 개발을 담당하였습니다.

애자일 프로세스기반으로 근무하였고 매주 진행되던 스프린트 내에서 slack을 통해 의사소통을 진행하였고 Jira에 주어지는 ticket을 기반으로 개발을 진행하였습니다.

근무를 하며 얻은 경험을 간단히 정리하면

- 충전 종료 후 결제 주문서 상 상태가 결제대기인 상태로 남아있는 이슈의 원인이 동시성 문제임을 파악하고 이를 재현하고 알려 해당 문제를 해결한 경험
- 비동기 이벤트 기반 처리를 하는 부분에서 간헐적인 낙관적 락 오류가 발생하는 원인이 실제 충전기기와의 통신에서 지연이 발생하게 되는 것임을 로그를 통해 파악하고 이를 개선하기 위해선 OSIV 옵션을 비활성화 하여 영속성을 짧게 유지시키는게 유리할 것 같다는 의견을 제시해본 경험
- 또한 서버 배포 이후 사용자에게 푸시 알림이 지속적으로 나가는 오류를 발견하고 prd db 조작을 통해 알림 전송을 막고 문제의 원인이 mq내에서 반복적으로 발생하던 오류임을 파악하고 이를 dead-letter queue 및 retry 횟수 제한으로 해결한 경험
- 정산시 충전 서버와 결제 서버와의 차액 발생을 쉽게 해결하기 위한 사후 재결제 기능을 구현하게 되었고, 배치 서버에 등록하고 배치서버 및 메인 서버에 테스트 코드를 작성하여 실 동작여부를 검증해보는 경험

등과 같은 다양한 경험을 해보았습니다.

근무를 하면서 단순히 인턴이라는 생각보단 하나의 팀이라는 생각을 가지고 일할 수 있게 도와주셨고 저도 그 덕분에 더욱 몰입하며 일할 수 있었습니다.

개발을 하면서 가장 중요시 했던 부분은 내가 수정하는 부분이 실제 사용자에게 어떠한 영향을 미칠 수 있는 가 였습니다.

기능적으로는 결함이 없으나 사용자의 입장에서 서비스를 이용하는데 기다림의 시간이 길어지는 경우, 다량의 사용자가 유입된 상황에서는 정상 동작하지 않을 가능성이 있는 경우와 같이 서비스 이용자가 느낄 수 있는 잠재적 불편함

이 존재한다고 생각했고

이를 개선하기 위해서 기존 불필요한 필터링 작업 제거, 불필요한 다량의 데이터 조회 제거, 동시성 문제를 고려한 쿠폰 발급 서비스 개선등의 개발을 진행하였습니다.

이후 개인적으로 트래픽이 몰리는 상황을 만들어 보고자 기존 진행했던 프로젝트에 부하를 걸어보고 이에 발생하는 동시성 문제 및 응답 지연 문제를 해결해보는 프로젝트를 진행하게 되는 계기가 되었습니다.

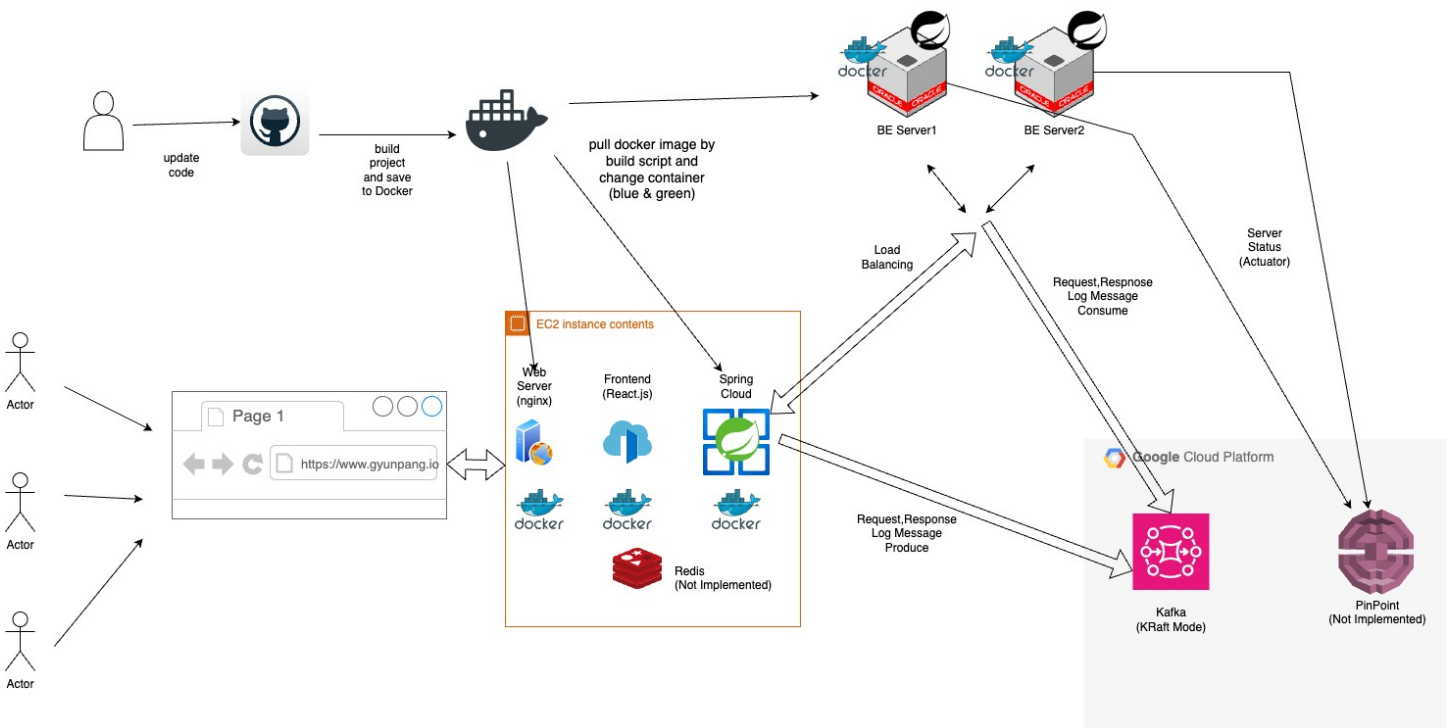
PROJECTS

Gyunpang

개인 프로젝트

2024.03 ~

프로젝트 구조



프로젝트 배경

실제 서비스 도중 발생할 수 있는 여러 상황들을 만들어보고 이 상황 속에서도 강건하게 유지되는 시스템을 구축해보고자 시작한 프로젝트입니다.

인터넷 쇼핑이라는 도메인을 이용하여 선착순 쿠폰, 재고 처리 등 다량의 사용자가 접근하는 상황에서 발생하는 동시성 문제와 특가 이벤트와 같이 일시적으로 급격하게 증가하는 트래픽에 대처하는 방법을 익혀보는 것이 목적인 프로젝트입니다

사용한 기술 스택

BE : Spring (MVC, WebFlux, JPA, Security, Cloud, Kafka)

Devops : nginx, docker

MQ : Kafka(Kraft)

Cloud : aws ec2, oracle cloud, google cloud platfrom

Cache : (Redis)

Monitoring : (Pinpoint)

구현한 내용

추후 기능별 정리 후 업데이트 예정입니다. **개인 블로그** 에서 현재 진행 상황을 확인하실 수 있습니다.

공강구조대

팀 프로젝트 (3인)

2023.03 ~ 2023.06

현재 프로젝트의 코드는 **공강구조대** 깃허브에서 확인하실 수 있습니다.

현재는 비용 문제로 프로젝트의 운영을 지속할 수 없어 스토어 및 서버의 배포를 중단한 상태 입니다

프로젝트 시연 영상은 **여기** 에서 확인하실 수 있고,

프로젝트의 PPT는 **여기** 에서 다운받으실 수 있습니다.

프로젝트 배경

이 프로젝트는 졸업 설계 프로젝트이며, 팀원끼리 브레인스토밍을 한 결과, 시간표와 통학이라는 소재를 이용해 우리가 대학생살을 하며 필요했던 부분을 채워보자는 의견으로 좁혀졌습니다. 팀원들 모두 통학을 해본 경험이 있었고 통학하는 과정에서 공강시간이 크게 생기는 경우 다시 집으로 돌아가기에는 시간이 너무 무의미하게 사용되어 집으로 갈 수는 없지만 그렇다고 마땅히 할 일을 생각해내지 못하고 시간을 무의미하게 보낸 경우가 많았습니다. 이런 경험을 토대로, 공강시간에 할 수 있는 다양한 활동을 추천해주고 시간을 효율적으로 관리할 수 있게 하는 프로젝트를 진행해보자는 의견이 나와 진행하게 된 프로젝트 입니다.

담당한 부분 : 프로젝트 기획 및 설계, 프론트 엔드, 백엔드

사용한 기술 스택

교내 OPEN API를 통해 강의 정보를 받아오고, 프론트는 Flutter, 백엔드는 Spring Boot를 사용하였고, DB는

MySQL을 사용했습니다. 또한, 서버는 EC2를 통해 배포 하였고, RDS를 사용하여 원격으로 DB를 관리하였습니다. 그리고, github actions와 AWS S3, AWS CodeDeploy를 이용하여 CI/CD를 구현하였습니다.

구현한 내용

해당 프로젝트에서 저는 배포와 관련된 부분 (CI,CD설정 및 AWS설정)을 담당하였고, 현재 **무중단 배포** 중에 있습니다. 또, 프로젝트의 **인증**을 **JWT** 를 이용하여 수행하였고 Filter에서 인증기능을 수행하였습니다. 그리고, 프론트엔드와 원활한 소통을 위해 서버에 **스웨거**를 적용하였습니다.

또, 다음의 SMTP서버를 이용하여 **이메일 인증 기능** 을 구현하였고, 이외의 로그인 및 회원가입등 사용자와 관련된 부분에 대한 기능 개발(엔티티 설계, 서비스 로직 및 API 설계 등)을 담당하였습니다. 그리고, *Jsoup* 을 이용하여 학교 비교과 페이지인 Uostory페이지의 정보들을 **크롤링**하여 DB에 저장하는 기능을 구현하였으며 이를 **Spring Scheduler**에 등록하여 매일 오전 5시에 수행되게끔 설정하였습니다. 그리고, *FCM* (FireBase Cloud Messaging)을 사용하여 서버에서 Flutter 기기로 알림을 보내는 기능을 구현하였고, Flutter에서 이를 받아 처리하는 부분을 구현하였습니다. 이외에도, 친구관련 기능, 그룹관련 기능, 카테고리 관련 기능, 알림 관련 기능을 담당하여 개발을 하였습니다.

또한, 플러터를 이용하여 UI 개선작업 및 오류 메세지 핸들링, 화면 재배치등의 작업을 수행하였습니다.

협업 도구

팀원과의 소통은 Notion, Swagger, 카카오톡, Discord를 이용하여 필요시마다 적극적으로 소통을 하였습니다.

프로젝트 회고 프로젝트 구현에 주어진 시간 자체가 짧았고, 인원 또한 3명이었으며 심지어 프론트엔드가 1명인데 Flutter를 처음 사용해본 상태라 개발을 빠르게 해내지 못하였습니다. 프로젝트 자체가 활동하는 유저의 수가 어느정도 존재해야 신규 유입 사용자가 서비스를 100%활용할 수 있다는 한계점을 가지고 있는 프로젝트였다고 생각합니다. 이에 대한 대책으로 프로젝트에서 제공하는 빈 시간 찾기, 그룹 공지사항, 게시판,댓글 등을 통해 이미 존재하는 그룹인 동아리를 우선적으로 유저로 끌어드리려 하였습니다. 하지만, 개발 자체의 시간이 꽤 소요되었고 플레이스토어에는 업로드가 되어 있었으나, 앱스토어의 경우 개발자 등록비용이 10만원이고 해당 비용을 캡스톤 지원금으로 지불할 수 없다고 하여, 교내 중앙 동아리에 협조를 구하기 힘들었습니다. 때문에, 안드로이드 사용자를 대상으로만 테스트를 진행할 수 있었고 동아리 내에서 모든 동아리원이 저희 어플을 사용할 수 없었기에 유저수 유치에 실패하게 되었습니다.

또한, 개발 일정에 쫓겨 개발을 하다보니 테스트 코드를 제대로 작성하지 못한 상태로 Postman을 이용해 자체적으로 테스트만 진행하였습니다. 또, 트래픽 테스트 및 성능 모니터링도 다른 툴을 사용한 것이 아닌, 직접 ec2인스턴스에 들어가 로그를 보는 형식으로 진행하였습니다.

그래서, 우선 Spring Actuator와 프로메테우스, 그라파나를 이용하여 모니터링 시스템을 구축하고, 핀포인트도 도

입하여 서버의 성능을 실시간으로 모니터링할 환경을 구성하였습니다. 그 후, 부하 테스트를 수행하고 병목 지점을 찾아 성능 개선 작업을 수행하였고 동시성 문제를 고려하여 코드를 수정하였습니다.(V2)

자세한 내용은 [블로그](#) 및 [깃허브](#)를 참고 해주시면 감사하겠습니다.

또, 많은 트래픽 상황에 따른 대규모 데이터 처리 및 스트림 처리에 대해 카프카를 공부한 후 프로젝트 성능 및 모니터링 개선 방안을 세워 추가적으로 작업을 진행할 계획에 있습니다.(V3)

스포티파이 클론코딩

팀 프로젝트 5인

2023.03 ~2023.04

프로젝트에 대한 상세 코드는 [Homebrewtify_BE](#) 깃허브에서 확인 부탁드립니다.

프로젝트 배경

이 프로젝트는 서울소재 대학생간 진행하는 프로그래밍 동아리인 Homebrew클럽에서 진행한 프로젝트로 약 한달 간 스포티파이 웹 페이지를 참고하여 Homebrewtify라는 음악 스트리밍 웹 애플리케이션을 만드는 것이 목적인 프로젝트 입니다.

해당 프로젝트는 프론트엔드 2명 백엔드 3명으로 이루어져 있고 그 중 저는 백엔드를 담당하고 있습니다.

담당한 부분 : 프로젝트 요구사항 정리 및 ERD설계, 백엔드

사용한 기술 스택

프론트는 React, 백엔드는 Spring Boot를 사용하였고, DB는 MySql을 사용했습니다. 또, Kaggle의 spotify dataset을 기반으로 음원 데이터베이스를 구성하였습니다.

구현한 내용

위 프로젝트에서 저는, *Spring Batch*를 이용하여 spotify dataset (11만 4천곡)을 설계한 DB에 맞게 파싱하고 저장하는 작업을 했으며, 이 과정에서 3분 이상 걸리던 배치 작업을 중복 제거, 로직 개선, 자료구조 변경 등을 통해 45초 까지 줄였습니다. 또, 프로젝트에서 발생하는 예외를 잡는 controller advice를 만들어 예외를 핸들링 하였으며, 엔티티간의 지연로딩 및 관계 매핑을 하였습니다. 그리고, 팔로우 기능, 좋아요 기능, 내 라이브러리 기능, 플레이리스트 기능, 최근 재생목록 기능, 로그인 및 회원가입에 필요한 컨트롤러 및 서비스를 구현하였습니다. 프로젝트에 *Spring Security*를 적용하였고, 인증 방식은 JWT를 이용하였으며 AccessToken은 인증 헤더에 담아 클라이언트의 로컬 스토리지에 저장하였고 RefreshToken의 경우 httpOnly쿠키로 클라이언트에게 전달 후 DB에 저장하였습니다.

협업 도구

프론트엔드 팀원들과 소통을 위해 프로젝트에 Swagger를 적용하였고, Slack 및 매주 진행하는 회의를 통해 협업을 하였습니다.

마찬가지로 백엔드 팀원들과도 Slack 및 매주 회의를 진행하며 협업을 진행하였습니다.

프로젝트 회고 데이터 셋 자체가 11만개 정도 되어 서버가 응답을 보내는데에 시간이 오래 소요될 것이라 예상했으나, 적당한 쿼리 수정 및 페이징 처리로도 충분히 빠른 응답을 내보낼 수 있었습니다. JWT를 처음으로 공부하고 도입했던 프로젝트로, 스프링 동작 과정에 대해 좀 더 깊이 공부할 수 있게된 계기가 되었고 인증,인가 방식 공부에도 큰 도움이 된 프로젝트 였습니다. 다만, 프론트엔드에서 처리할 부분이 한달안에 하기에는 다소 많아 기간내에 완성을 하진 못하였습니다. 서버에서 스포티파이 open api를 통해 노래를 받아오고 프론트로 보내는 방식도 생각해보았으나, 보안적 이슈도 있었고 그렇게 되면 통신 비용이 너무 증가하게 될 것이라 판단하여 이 방식은 선택하지 않았습니다. 결국, 백엔드에서 구현한 내용은 모두 반영되었으나, 노래 자체를 API로 받아오는 부분 및 기타 프론트엔드 작동 오류 부분을 수정하지 못한 채 프로젝트를 마무리하게 되었습니다.

비록, 전체적으로 완성된 프로젝트는 아니었지만 Spring Batch, JWT등의 기술을 습득하게 되었고, 자료구조의 소중함을 다시금 깨닫게 해 준 프로젝트였습니다.

음악맞추기

개인 프로젝트

2023.01 ~ 2023.02

자세한 프로젝트 내용은 **음악맞추기** 깃허브의 GuessMusic 프로젝트 pdf를 참고 해 주시면 감사하겠습니다.

프로젝트 배경

이 프로젝트를 진행하게 된 계기는, 제가 좋아하는 스타크래프트 유즈맵인 음악 맞추기 맵을 플레이 하다 느낀 불편함 때문이었습니다. 워낙 많이 하다보니, 다양한 맵이 필요했고 원하는 장르 또한 부족하다고 느꼈고 직접 유즈맵을 만들기에는 너무 많은 시간이 필요하다는 것을 알게되었습니다. 때문에, 친구들과 함께 맵을 쉽게 만들고 플레이 할 수 있는 웹을 만들고 싶다는 생각이 들어 직접 만들어 보기로 결정했습니다.

사용한 기술 스택

위 프로젝트에서 사용한 기술의 경우, 1인 프로젝트다 보니 SSR방식을 택했고 프론트엔드에서는 Thymeleaf, Vue.js, javascript, HTML5을 사용했고, 백엔드의 경우 Spring Boot 3.0.1, Spring Security을 사용하였고, DB는 PostgreSQL을 사용하였습니다. 그리고, 소켓 통신을 통해 사용자 간 채팅을 가능하게 하였습니다. 배포는 EC2를 통해 하였고 RDS를 사용하여 원격으로 DB를 관리하였습니다.

구현한 내용

사용자는 로컬 및 소셜 로그인을 통해 서비스에 들어올 수 있게 하였으며, 중복로그인 및 재로그인 시 세션count를

초기화 하여 항상 정상적으로 동작할 수 있게끔 하였으며 로그인 한 사용자는 만들어진 방에 참가하거나 직접 생성하여 방에 들어갈 수 있게 하였으며, 방에 접속한 경우 방장과 일반 참여자를 구분하여 화면을 다르게 표현하였습니다. 게임이 시작되면 각 사용자 별로 정답 점수를 표시하고 타이머를 이용하여 힌트를 표시하고 웹소켓을 통해 채팅 기능을 적용하여, 사용자간 대화 및 정답 입력이 가능하게 하였습니다. 정답을 맞춘 사용자가 나오게 되면 3초 동안 정답 공지 메시지를 출력하며 그 동안은 정답처리를 하지 않게끔 기능을 구현하였습니다. 로그인 시 관리자 계정으로 로그인 한 경우, 관리자 페이지로 이동이 가능하게 되며, 관리자 페이지에서 게임을 생성,수정,삭제할 수 있게끔 하였습니다. 또한, 페이징 처리를 통해 많은 양의 데이터를 불러오는데에 무리가 없게 하였습니다.

프로젝트 회고 첫 개인프로젝트이자 아직 유일한 개인 프로젝트 입니다. 평소 친구들과 자주 했던 스타크래프트 유즈맵에서 시작한 프로젝트인지라 해당 유즈맵과 유사한점이 많다고 생각합니다. 다만, 스프링을 처음으로 학습한 상태로, 그것도 짧은 강의만 듣고 무작정 구글링을 해가며 진행했던 프로젝트 였습니다. 요청마다 응답의 속도가 조금씩 차이나서 많은 곡을 재생하는 경우 어떤 브라우저에선 소리가 조금 밀리는 현상이 발견되었고 당시 여러 방법을 시도했으나 결국 고치지 못한 상태로 마무리 한 프로젝트 였습니다. 해당 프로젝트를 통해, 스프링을 통한 백엔드 설계, Thymeleaf 및 Vue.js를 이용하여 프론트엔드 설계를 경험해볼 수 있었고, OAuth및 SpringSecurity에 대해 처음 공부해볼 수 있었습니다. 특히, Spring Security의 경우 세션 로그인 방식에서 어떻게 이 로그인 정보를 유지할 수 있는 지에 대해 공부해볼 수 있는 기회가 되어 좋았습니다.

시간적 여유가 된다면, 기존에 파일을 서버내에 저장하던 방식에서 클라우드를 이용하는 방식으로 변경하고, 노래가 동시에 재생될 수 있도록 하여 프로젝트를 V2버전으로 업그레이드하고 배포해볼 생각입니다.

HomebrewChatting

팀 프로젝트 6인

2023.05 ~ 2023.05

현재 프로젝트는 HomebrewChattingBE 에서 확인하실 수 있습니다.

프로젝트 배경

이 프로젝트는 서울소재 대학생간 진행하는 프로그래밍 스터디 Homebrew클럽에서 진행한 프로젝트로, 채팅기능 및 인증기능을 메인으로 하는 Homebrew동아리 홈페이지를 제작하는 프로젝트 입니다.

사용한 기술 스택

React, Spring Boot를 통해 프론트엔드, 백엔드 개발을 진행하였으며 채팅을 위해 Websocket을 사용하였습니다. Github action, S3, CodeDeploy를 활용하여 프론트엔드 및 백엔드 CI/CD를 하였고, AWS certificate을 통해 HTTPS 인증을 진행하였고, 하나의 도메인에 프론트,백엔드 서버 모두를 돌리기 위해 Nginx를 이용하여 리버스 프록시를 적용하였습니다.

구현한 내용

프로젝트에서 채팅 기능 및 CI/CD작업, Nginx관련 설정을 담당하였습니다. 프로젝트 초기 원활한 프로젝트 진행을 위해 AWS S3,CodeDeploy,Github Action을 이용하여 CI/CD작업을 우선적으로 수행하였습니다. 그 후, Stomp

를 이용하여 프론트와 소켓 통신을 진행하였고, 웹 소켓 업그레이드를 위해 Nginx에 관련 설정을 하였고 이를 통해 클라이언트와 서버가 안전한 wss연결으로 채팅을 진행할 수 있게끔 하였습니다. 또, 채팅에 필요한 채팅 상태, 내용 및 DB 저장등 관련 API 및 로직들을 구현하였습니다.

프로젝트 회고 프로젝트 팀원 모두 과제 및 시험 때문에 활동을 많이 하지 못하여, 완성도면에서 많이 떨어지는 프로젝트라고 생각합니다. 프론트엔드를 담당한 팀원분을 포함해 일부 인원끼리 모여 급하게 백엔드 전반을 뒤집고 프로젝트를 진행하였고 그 결과, 기존 목표였던 소셜 로그인(JWT) 및 그룹 채팅 기능 완성에 실패하게 되었고, 로컬 로그인 및 1대1채팅 기능만 급하게 만든 상태로 프로젝트를 마무리하게 되었습니다.

채팅의 경우 1:1 채팅 및 그룹 채팅에 필요한 기능 및 API 구현은 모두 끝낸 상태로 프로젝트가 종료하였기에 추후에 프로젝트를 다시 개선시키고자하는 생각은 없습니다 !

디너서비스 웹

팀 프로젝트 2인

2022.11 ~ 2022.11

자세한 프로젝트 내용은 **미스터대박** 깃허브의 프로젝트 pdf를 참고 해 주시면 감사하겠습니다.

프로젝트 배경

고객이 원하는 음식을 고객이 원하는 때에 배달하는 서비스를 구축하라는 비즈니스 요구사항을 바탕으로 만들어진 소프트웨어 공학 시간에 진행한 프로젝트 입니다.

사용한 기술 스택 React,Node.js를 이용하여 프론트,백엔드 개발을 진행하였습니다. DB는 MySQL을 사용하였습니다.

구현한 내용

주어진 비즈니스 요구 사항을 바탕으로 분석,설계 산출물을 만들고 이를 바탕으로 실제 웹을 구축하였습니다. 해당 서비스는 고객을 위한 주문 인터페이스가 존재하고, 관리자의 조리 및 배달 그리고 재고 관리를 위한 인터페이스가 존재합니다. 재고는 DB를 구축하여 관리하였고 이를 바탕으로 품질 관리를 하였습니다. 해당 프로젝트에서 프론트엔드의 경우 페어 프로그래밍을 통해 개발을 진행하였고, 백엔드는 제가 전적으로 담당하였습니다.

프로젝트 회고 학교 수업에서 제시한 비즈니스 요구사항에 맞춘 프로젝트로 해당 프로젝트로 백엔드 개발에 조금 흥미를 가지게 되었고, 현재의 제가 될 수 있었다고 생각합니다.

지금 보면 너무나 단순한 기능만을 구현한 프로젝트이지만 그 당시엔 모든게 너무 새로웠고 HTTP가 뭔지도 CSR,SSR이 뭔지도 잘 몰랐던 터라 개발에 많은 어려움이 있었고, 같이 진행한 동기와 거의 매일 밤까지 코딩 공부를 했던 기억이 납니다. 프로젝트를 추후에 개선할 생각은 없습니다 !

EDUCATION

서울시립대학교

2018.03 ~ 2024.02

컴퓨터과학부

2021.3 ~ 2021.6 학업 우수 장학

2021.9 ~ 2021.12 교내 교양 수학 튜터 활동

2022.9 ~ 2022.12 데이터 통신 조교 활동

수료 강의

삼성전자 dx 하계 S/W 알고리즘 특강 (우수 수료, pro 취득)

2023.07 ~ 2023.08

LANGUAGE

TOEIC

2022.07.24

845

TOEIC Speaking

2023.09.24

150 (IH)

* * * * *

Wongyun Shin



Back-End Developer with Spring

