백엔드 개발자
유예지입니다!

안녕하세요! **백엔드 개발자** 유예지 입니다.

java, Spring, AWS, RestAPI를 사용한 웹 서버를 운영하고 있습니다.
GIthubAction과 AWS CodeDeploy를 통한 **CI/CD구성 경험**이 있습니다.
확장성과 **유지 보수성**이 높은 백엔드 아키텍처를 위해 **SOLID원칙을 지켜 코드를 작성**합니다.

프로젝트

BURGERPUT 2023.05 ~ 진행중 **GitHub :** https://github.com/yellTa

개요
사무 자동화 시스템 구축, 개발 배포 수행

배경
아르바이트 사무 작업 수행 중, Zenput이라는 웹 사이트에 온도 데이터를 입력하는 사무작업이 있었습니다.
Zenput사이트는 식품, 기기별로 온도를 입력해야 했었고, 매장에서 사용하지 않는 식품, 기기의 항목도 포함하고 있었습니다. 이때 매장에서 사용하지 않는 식품, 기기는 온도를 999로 입력해야 했습니다.

목표
- 프로그램을 통해 사용자 대신 필요한 값만 자동으로 입력해주기
- 프로그램을 통해 온도를 입력했을 때 오기입률 0%로 만들기

사용기술
Java, Spring, AWS, MySQL, Github, Selenium(Library)

주요업무
- 코드 리팩토링을 통한 크롤링 속도 **120초->16**초로 8배 개선
- Spring Interceptor와 JWT토큰을 사용한 로그인 시스템 구축
- Git을 활용한 형상관리 수행
- 서버 작동 결과를 메일로 받아보는 모니터링 시스템 구축

성과
- 23년도 10월, 11월, 12월 3개월간 사람이 입력한 오후기기 **오기입률 13%** -> 24년 6월, 7월, 8월 3개월간 프로젝트를 사용해 제품한 오후기기 오기입률 **0%로 개선**
- 4개월간 **총 267회** Burgerput을 사용해 Zenput에 온도가 제출됨
- **사용성을 인정**받아 다른에 매장 배포 준비중

스터디

항해99 코딩테스트 스터디 2기 *2024.06 ~ 2024.07*

항해99 코딩테스트 스터디 3기 *2024.07 ~ 2024.08*

페어 프로그래밍을 통해 함께 알고리즘 풀이
4주차 우수 TIL에 선정됨[TIL링크]

3. 우수 TIL 작성자 : 유예지 (자바/챌린저)

- TIL 링크
- 선정 이유

  예지 님의 TIL은 문제 풀이 로직을 직접 손으로 풀어본 점이 인상적이었고, 코딩 테스트 문제 유형을 분석하여 체계적으로 접근한 점이 돋보여 우수 TIL로 선정되었습니다. 더불어, 이를 토대로 앞으로 보완하고 집중해야 할 과제들을 제시하여 중요한 포인트를 잘 짚어냈습니다.

교육사항

[졸업]인천재능대학교 컴퓨터 정보과 *2016.02~ 2018.02*

솔데스크- 정보보안 침해대응 전문가 과정 *2018.03~2018.10*

- 리눅스/윈도우 관리자 과정 교육
- 쉘 프로그래밍 교육

자격증

- 네트워크관리사 2급 2019.07
- 리눅스 마스터 2급 2019.11
- 정보처리산업기사 2019.11

경력사항

구루멘토링 *2019.11~ 2020.07*

웹 정보보안 연구원

다른 사람을 존중하는 팀 문화의 중요성에 대해 알게됨

연락처

Email : jwb27964@gmail.com
Phone : 010-4064-1306

<사용률 67%의 프로그램을 만들다>

햄버거 프랜차이즈에서 근무하던 시절, Zenput이라는 웹 서비스를 통해 매장 내 기기와 식품의 온도를 입력하는 작업이 있었습니다. 그러나 사용하지 않는 기기나 결품된 식품의 경우, 해당 항목에 999를 입력해야 했습니다. 이 과정에서 온도를 입력하는 곳에 잘못된 온도를 입력하거나, 반대로 999를 입력해야 하는 곳에 온도를 입력하는 오류가 빈번하게 발생했습니다.

문제를 해결하기 위해 java, Spring을 이용해 자동화 프로그램을 직접 만들어 업무에 적용했습니다. 크롤링 라이브러리를 이용해 필요한 데이터를 추출한 후, 사용자가 필요한 데이터를 선택하도록 했습니다. 또한 선택한 값의 온도를 입력하면 나머지 필드는 미들웨어가 자동으로 채워 사용자 대신 Zenput에 입력을 수행했습니다.

그 결과로, 3개월간 사람이 입력한 데이터의 오기입률 13%에서 미들웨어로 제출된 오기입률을 0%로 개선할 수 있었습니다. 또한 4개월 동안 267회 미들웨어를 통해 데이터가 Zenput에 제출되었으며, 8월 오후기기의 전체 온도 데이터 중 67%가 미들웨어를 자동으로 제출 처리되었습니다.

위와 같은 성과로, 다른 매장에서 사용 제의가 들어와 서비스가 확대되는 경험을 겪었습니다. 이런 경험을 바탕으로 불편함을 찾아내고 개선하는 개발자로 거듭나고 싶습니다.

<트러블 슈팅을 하며 깨달은 원리의 중요성>

프로젝트에서 매일 크롤링을 통해 DB에 데이터를 저장하는 작업이 있었습니다. 이때 테이블 레코드의 순서 유지를 위해 테이블의 데이터를 모두 삭제하고 크롤링한 값을 DB에 다시 저장하는 방법을 선택했습니다. 테이블 삭제 작업을 수행하기 위해 Service로직 메서드에 @Transaction을 적용한 후 truncate를 수행하고, Spring JPA를 통해 데이터를 저장했습니다.

하지만, 애플리케이션은 중복된 키 에러를 발했습니다. 원인을 알아보기 위해 JDBC와 Spring JPA, @Transctional의 동작 원리에 대해서 공부했습니다. 원인은 JDBC가 쿼리를 실제 DB에 직접 수행하는 데 반해, Spring JPA의 메소드 쿼리는 서버 캐시 메모리의 영속성 컨텍스트에 먼저 수행하고 Transaction이 끝난 후 실제 DB에 flush되기 때문이었습니다. JDBC로 truncate를 수행 후, 영속성 컨텍스트와 동기화를 맞춰주기 위해 @Modifying(clearAutomatically = true)옵션을 부여해 문제를 해결했습니다.

이 경험을 통해, 사용하려는 기술의 동작 원리를 깊이 이해하지 못한 것이 문제의 근본적인 원인임을 깨달았습니다. 이러한 문제를 방지하기 위해, 기능 구현 전에 필요한 기술의 작동 방식을 명확히 이해하고 이를 문서화하는 작업을 하기로 했습니다. 구현에 필요한 배경지식을 조사한 후, 기능이 개발될 순서를 정의하고, 이를 바탕으로, 단계별로 기능을 개발했습니다. 그 결과, 프로젝트 진행 과정에서 100여 개의 기능 개발 문서와 트러블슈팅 문서를 작성할 수 있었습니다.

이 과정을 통해, "어떤 기술을 써야 하는지" 그리고 "더 나은 결과를 얻기 위해 어떤 방법을 사용해야 할지"를 고민하는 개발자로 성장할 수 있었습니다.

<고대 개발자부터 시작하기>

웹 백엔드 개발의 기본을 다지기 위해 JSP, Servlet, WAS의 개념을 공부했습니다.

기술이 발전하면서 다양한 프레임워크가 등장했고, Spring Framework도 그중 하나였습니다. 모든 웹 프레임워크의 근간은 HTTP 프로토콜을 사용한다는 점에서, 프레임워크를 잘 활용하는 것도 중요하지만 HTTP 통신이 어떻게 발전해왔는지 이해하는 것이 프레임워크를 더 잘 다룰 수 있는 기반이 된다고 생각했습니다.

이러한 이유로 웹의 기본을 공부하기 위해 웹 부스트코스에서 풀스택 강의를 수강하고, JSP와 Servlet을 활용한 간단한 웹 프로젝트를 수행했습니다. Servlet의 doGet()과 doPost() 메서드를 통해 데이터를 받고 JSP로 표현해보면서, Spring의 @GetMapping과 @PostMapping이 같은 역할을 한다는 점을 깨달았습니다. 또한, Tomcat WAS를 직접 구동하여 HttpServletRequest와HttpServletResponse 객체를 사용해 클라이언트와 HTTP 프로토콜을 통한 데이터 통신 과정을 이해할 수 있었습니다.

기존에는 Spring에서 단순히 애노테이션을 추가해 적용하고 구현하는 것이 끝이라고 생각했지만, Servlet과 JSP, Tomcat을 직접 다뤄보면서 그 밑단에서 어떤 방식으로 동작하는지 이해하게 되었습니다.
이 경험으로, 기본 개념에 충실한 학습이 복잡한 기술을 습득하는 데 중요한 기반이 된다는 사실을 깨달았습니다. 이를 계기로, 기본을 중요하게 여기게 되었고, 꾸준히 기본 개념을 되새기며 공부하는 습관을 가지게 되었습니다. 앞으로도 이러한 자세로 새로운 기술을 배우고 적용해 나가고자 합니다.

블로그/ 깃헙링크

[Burgerput 프로젝트 공식 깃헙 링크]
https://github.com/Burgerput

[WebDriver 사용문제
Cache를 지우고 WebDriver를 다운받아 사용할 것인가 vs 라이브러리의 버전을 업그레이드 시킬 것인가]

https://velog.io/@bbubboru22/Chrome-Driver-사용-에러

BURGERPUT 프로젝트 과정

1. 프로젝트 배경

1. 젠풋이란?

젠풋

고객에게 올바른 제품의 품질을 제공하기 위해 식품 및 사용하는 기기의 **온도를 기록하는 시스템**이다.

젠풋의 중요성

젠풋은 **REV**에서 FS(Food Safety)로 분류되며 **감점 시 25점 중 16점**을 차지하는 큰 요소이다.

REV 방문일 기준 30일 간의 모든 젠풋 데이터를 확인한다.

⟶ **즉 방문일 기준 30일간의 모든 기록에 오기입이 없어야 감점이 이루어지지 않는다.**

현재 각 매장별 점장에서 지역장으로 젠풋의 입력 사항을 보고하는 등 젠풋 기록은 매장 운영에 필수적인 사항으로 중요시 여겨지고 있다.

*** REV**

매장의 운영 상태를 분기별로 불시 점검하는 시스템으로 브랜드 점수와 FS 점수를 통해 매장의 등급을 평가한다.
FS에 해당하는 점수 중 25점 이상 감점 시 F를 부여하며, F 등급 3회 연속시 폐점이다.

1. 프로젝트 배경

1. 젠풋이란?

젠풋 오기입의 예시

Date Subr	온도계(°	온도계(°	PHU내 제	PHU내 제(
2024-01-04	31.0 °F	32.0 °F	171.0 °F	999.0 °F
2023-12-31	32.0 °F	31.0 °F	171.0 °F	999.0 °F
2023-12-28	31.0 °F	32.0 °F	176.0 °F	999.0 °F
2023-12-27	31.0 °F	32.0 °F	176.0 °F	999.0 °F
2023-12-18	31.0 °F	32.0 °F	181.0 °F	999.0 °F
2023-12-16	31.0 °F	32.0 °F	999.0 °F	179.0 °F
2023-12-15	32.0 °F	31.0 °F	176.0 °F	999.0 °F
2023-12-14	31.0 °F	32.0 °F	181.0 °F	999.0 °F
2024-01-06	32.0 °F	999.0 °F	161.0 °F	999.0 °F

[기기 입력 오류 자료]

Date Subm	롱치킨	너겟킹
2023-12-11	191.0 °F	4188.0 °F

[식품 입력 오류 자료]

2024년 1월 12일 기준
오답률 : 기기 > 식품

가장 많은 오기입 항목 유형:
사용하지 않는 기기목록 입력

[90일간의 젠풋 오기입률]

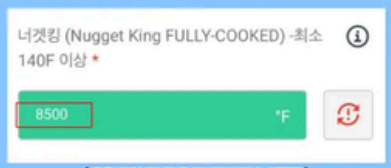
[30일간의 젠풋 오기입률]

3. Burgerput 프로젝트 기대 효과

자체적인 입력값 검증으로 매장에서 사용하지 않는 기기 그리고 식품의 온도를 **잘못 입력할 확률을 감소**시킨다.

⟶ **현재 식품의 최소 범위는 정해져 있지만 최대 범위는 정해져 있지 않아 불가능한 범위(ex: 4803°F)가 입력되는 경우가 발생할 수 있다. Burgerput은 최대값을 설정하여 이를 방지할 수 있다.**

[최소 값만 지정된 식품 요소]



[불가능한 범위가 나오는 식품 요소]

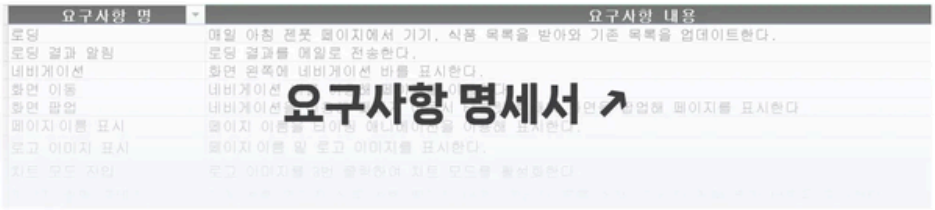
REV에서 **16점**이나 되는 감점 사항을 배제시킬 수 있다.

⟶ **REV 진행 중 FS 항목에서 25점 이상의 감점이 발생하면 즉시 F를 받게 된다. 젠풋은 FS 점수 중 16점에 해당하는 주요 항목으로, 오기입이 없다면 가장 큰 감점 요인을 배제시킬 수 있기 때문에 그 외 FS 점검에 집중할 수 있게 된다.**

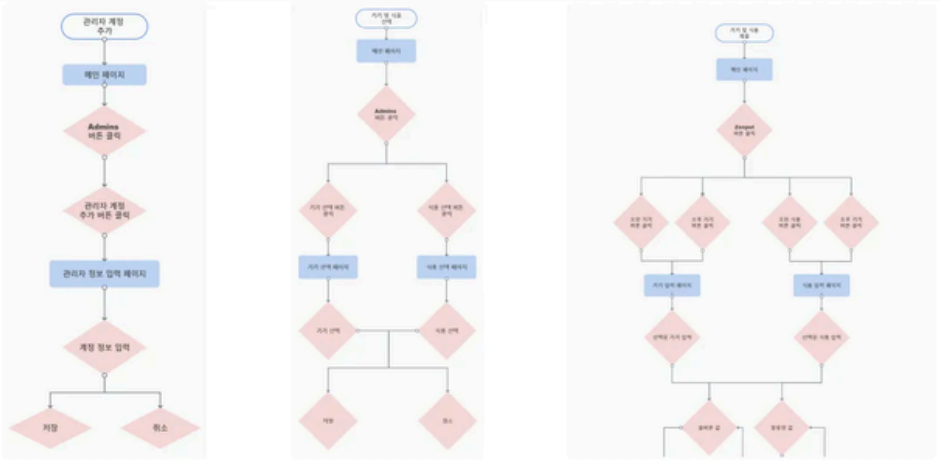
1. 요구사항 명세서

주요 요구사항...

요구사항 명	내용
로딩	매일 아침 젠풋 페이지에서 기기, 식품 목록을 받아와 기존 목록을 업데이트한다.
기기 및 식품 선택	매장에서 사용하는 기기 및 식품을 선택해 목록에 저장한다.
입력 값 검증	지정된 온도 범위(최소~185°F등)를 벗어난 숫자 입력시 값을 검증해 에러를 표시한다.
제출	제출 버튼을 클릭 시 사용자가 입력한 온도 값을 젠풋 페이지에 제출한다.



2. 유저 플로우



3. 순서도