



고민을
해결로 바꾸는
개발자 최성장입니다.

+82 010-8967-9399

palga65@gmail.com

Blog. blog.naver.com/growingdev95

PROFILE



최성장

CHOI SUNGJANG

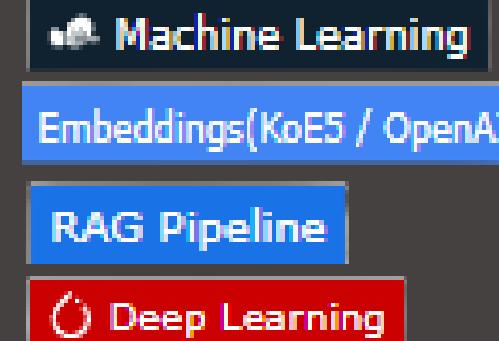
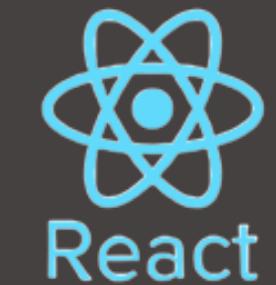
1995.01.02

+82 010-8967-9399

palga65@gmail.com

<https://growingchoi.github.io/>

SKILLS



CAREER & Education

2020.06 중국인민대학교 졸업(학사)

2020.11~2021.04 한국ICT인재개발원
응용SW엔지니어링 교육 이수(JAVA)

2021.06 ~ 2024.11 (주)우리C&C SI 개발팀 근무

2025.03 ~ 2025.09 SKNetwork AI-FAMILY 부트캠프 수료

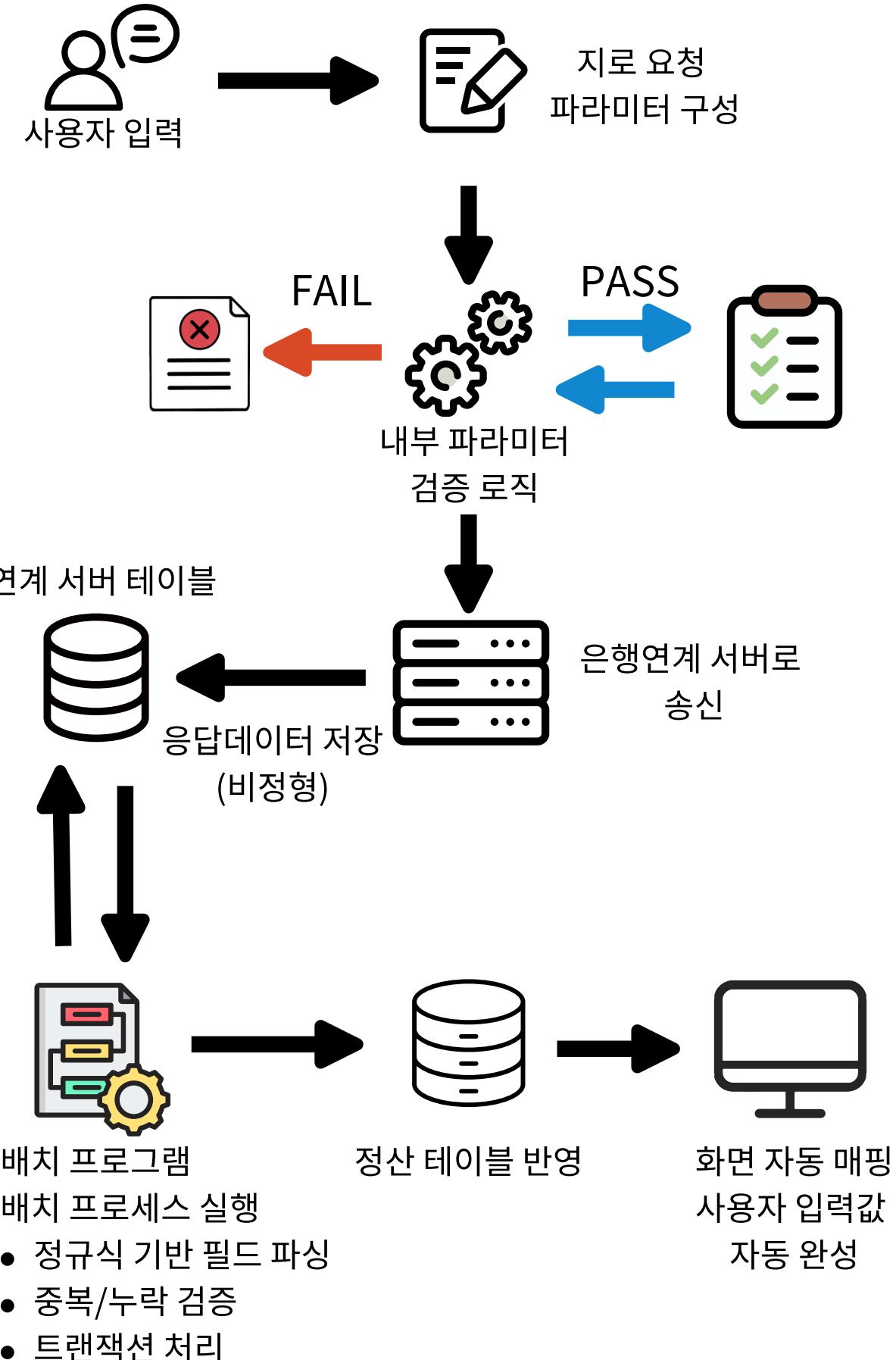
01.

Project

국가 예산 정산 시스템 고도화

CAREER EXPERIENCE

[A] 지로(은행)기반 연계 프로세스



1. 국가 예산 정산 시스템 고도화(백엔드 개발)

기간: 2022.07 ~ 2023.04 / 2023.10 ~ 2024.12

[개요]

국가 예산 집행 시 필요한 지로(은행) 거래 응답을 자동으로 수신·파싱하여 정산 화면에 자동 반영하는 기능을 개발했습니다.

[담당 업무]

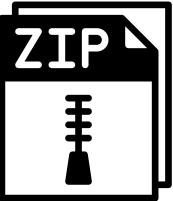
- [A] 지로(은행) 기반 연계 개발
 - 지로 요청 파라미터 구성 및 송신 로직 개발
 - 사용자 입력값 기반으로 NH은행에 전달되는 요청 값을 생성
 - 요청 값 검증 및 예외상황 처리 로직 설계
 - 지로 응답 데이터 수신·파싱·저장 배치 개발
 - NH은행 응답 문자열을 배치 프로그램에서 주기적으로 수신
 - 기존 split 기반 파싱 실패 문제 해결 → 정규식 기반 파싱 로직 재구성
 - 자동 매핑 기능 개발
 - 파싱된 응답 데이터를 정산 화면의 입력값에 자동 반영
 - 데이터 누락·중복을 방지하기 위한 검증 로직 추가

CAREER EXPERIENCE

[B] 전자우편(우체국) 연계 프로세스



영수증 데이터 조회



ZIP 압축 처리

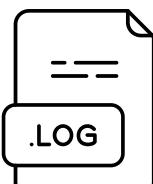
- BufferedOutputStream write
- flush 단계별 수행
- close 처리



전자우편(우체국)

API 연동

PASS FAIL



오류 시
관리자 전용
오류 표시

[C] SQL 성능 개선(Before → After)

[Before]

```
SELECT ...
  FROM A
  JOIN B
 WHERE A.col LIKE '%...%' ← Full Scan
   AND ( SELECT ... )      ← 단단 서브쿼리
```

[After]

```
WITH filtered AS (
  SELECT ...
    FROM A
   WHERE A.col = ? ← 인덱스 타는 조건
)
SELECT ...
  FROM filtered
  JOIN B
```

• [B] 전자우편(우체국) 연계 개발

- ZIP 패키징 및 우체국 API 자동 발송
 - 영수증·첨부파일 ZIP 변환 후 우체국 전자우편 API 자동 발송
 - flush 누락으로 발생한 메모리 증가 문제를 write → flush → close 구분 적용으로 개선
 - finally 기반 자원 해제 구조로 안정성 강화
- 발송 안정성·후처리 기능 강화
 - 요청·응답 전체 로그 기록
 - 발송 실패 시 관리자 전용 오류 메시지 제공
 - 장애 추적 및 원인 파악을 위한 로그 구조 개선

• [C] SQL 실행 계획 분석 및 성능 개선

- Full Scan 발생 원인 분석
 - LIKE '%검색%' 조건과 중첩 서브쿼리로 Full Scan 발생
 - 대용량 정산·통계 테이블 조회 시 응답 지연 문제 확인
- 쿼리 구조 최적화
 - WHERE 조건을 인덱스 활용 가능한 형태로 재작성
 - 서브쿼리를 WITH절 기반 선필터링 구조로 변경해 중간 연산 감소
 - 조회 컬럼 축소 및 JOIN 순서 정리로 전체 쿼리 흐름 최적화

CAREER EXPERIENCE

• 트러블 슈팅 : [A] 지로 응답 데이터 파싱 실패 문제

▷ 문제점

NH은행 지로 응답 데이터는 비정형 문자열로 제공되었고, 기존 `split` 기반 파싱 방식은 필드 누락, 구분자 인식 오류, 자동 매핑 실패 등으로 인해 정산 화면에서 데이터가 정확히 반영되지 않는 문제가 발생했습니다.

초기 개발 단계부터 파싱 로직 구현에 많은 시간이 들었으며, 개발 완료 후에도 케이스별 파싱 오류가 지속적으로 발견되었습니다.

▷ 초기 해결

우선 NH 지로 응답 구조를 분석하여 반복적으로 등장하는 패턴, 숫자/문자 조합, 길이 규칙 등을 정리하고 이후 필드별 규칙을 기반으로 정규식을 적용해 파싱 정확도를 높이는 방향으로 구조를 개선했습니다.

그러나 케이스별 예외 패턴이 많아, 정규식만으로는 100% 안정적인 파싱을 보장하기 어려운 상황이었습니다.

▷ 최종 해결

• 파싱 프로세스를 3단계 구조로 재설계 • 배치 내부

- | | |
|---------------------|-----------------|
| 1. 응답 문자열 사전 정규화 | 1. 중복 데이터 검증 |
| 2. 필드별 정규식 매칭 | 2. 파싱 예외 처리 |
| 3. 결과 검증 및 누락 필드 보정 | 3. 실패 데이터 로그 기록 |

기능을 추가해 문제 추적 가능하도록 개선했습니다.

▷ 배운점 & 아쉬운 점

▷ 배운점

- 비정형 문자열 처리에서는 규칙 기반 정규식 설계가 핵심임을 체감
- 전체 프로세스를 "정규화 → 파싱 → 검증" 단위로 분리할 수록 안정성이 높아짐
- 배치 · 트랜잭션 설계의 중요성을 실제 운영 환경에서 경험함

▷ 아쉬운 점

- 초기 단계에서 더 많은 응답 데이터를 수집했다면 정규식 설계 시간을 줄일 수 있었을 점
- 정상/오류 로그 규칙을 더 세분화했다면 장애 대응 속도가 더 좋아질 여지가 있다는 점

CAREER EXPERIENCE

• 트러블 슈팅 : [B] 전자우편 연계(메모리 누수 문제)

⚠ 문제점

영수증·첨부파일을 ZIP 파일로 패키징하는 과정에서 flush 단계가 누락되어 메모리 가지속적으로 증가하는 문제가 있었습니다.

대량 파일 처리 시 스트림이 정상 해제되지 않아 메모리 립(memory leak)이 발생했고, 우체국 API 발송 실패 시 로그가 충분히 남지 않아 원인 파악이 어려웠습니다.

! 해결 과정

ZIP 처리 로직을 전반적으로 점검하여 write → flush → close 단계를 명확히 분리하고, 예외 발생 여부와 관계없이 스트림을 반드시 종료하도록 try-catch-finally 구조를 적용했습니다.

또한 API 요청/응답 로그를 모두 기록하고, 실패 건은 관리자 전용 오류 메시지로 즉시 확인할 수 있도록 후처리 기능을 추가했습니다.

✓ 결과

ZIP 처리 과정의 메모리 사용량이 약 30% 감소했고, 대량 발송 시에도 안정적으로 동작할 수 있었습니다. 로그 체계가 정비되면서 발송 실패 원인을 빠르게 확인할 수 있어 운영 효율이 향상되었습니다.

• 트러블 슈팅 : [C] Full Scan 및 조회 지연 문제

⚠ 문제점

정산·통계 화면에서 대용량 데이터를 조회할 때 LIKE '%검색%' 조건이 인덱스를 사용하지 않아 Full Scan이 발생했습니다. 중첩 서브쿼리가 많아 실행계획이 불안정했고, 이로 인해 조회 응답 속도가 전반적으로 느려지는 문제가 있었습니다.

! 해결 과정

실행계획(PLAN) 분석을 통해 Full Scan의 원인을 파악한 뒤, WHERE 조건을 인덱스가 적용되는 형태로 재작성했습니다. 중첩된 서브쿼리는 WITH 절 기반 선필터링 구조로 변경하여 중간 연산량을 줄였고, JOIN 순서를 정리해 전체 쿼리 흐름을 단순화했습니다.

✓ 결과

쿼리 최적화 후 주요 화면의 조회 속도가 약 40% 개선되었고, CPU·IO 사용량도 안정적으로 낮아졌습니다. 선필터링 기반 구조로 재작성하면서 향후 유지보수 난이도 크게 줄어들었습니다.

CAREER EXPERIENCE

▶ 배운점 & 아쉬운 점

▷ 배운점

- 외부기관 API 연계 과정에서 예외 처리·로그 체계·자원 해제의 중요성을 체감함
- ZIP·스트림 처리 등 파일 기반 로직은 작은 누락도 운영 장애로 이어질 수 있음을 경험함
- SQL 성능 개선은 실행계획 분석과 선필터링/조인 구조 설계가 핵심임을 학습함
- DBA 협업을 통한 실행계획 검증 프로세스의 필요성을 깨달음

▷ 아쉬운점

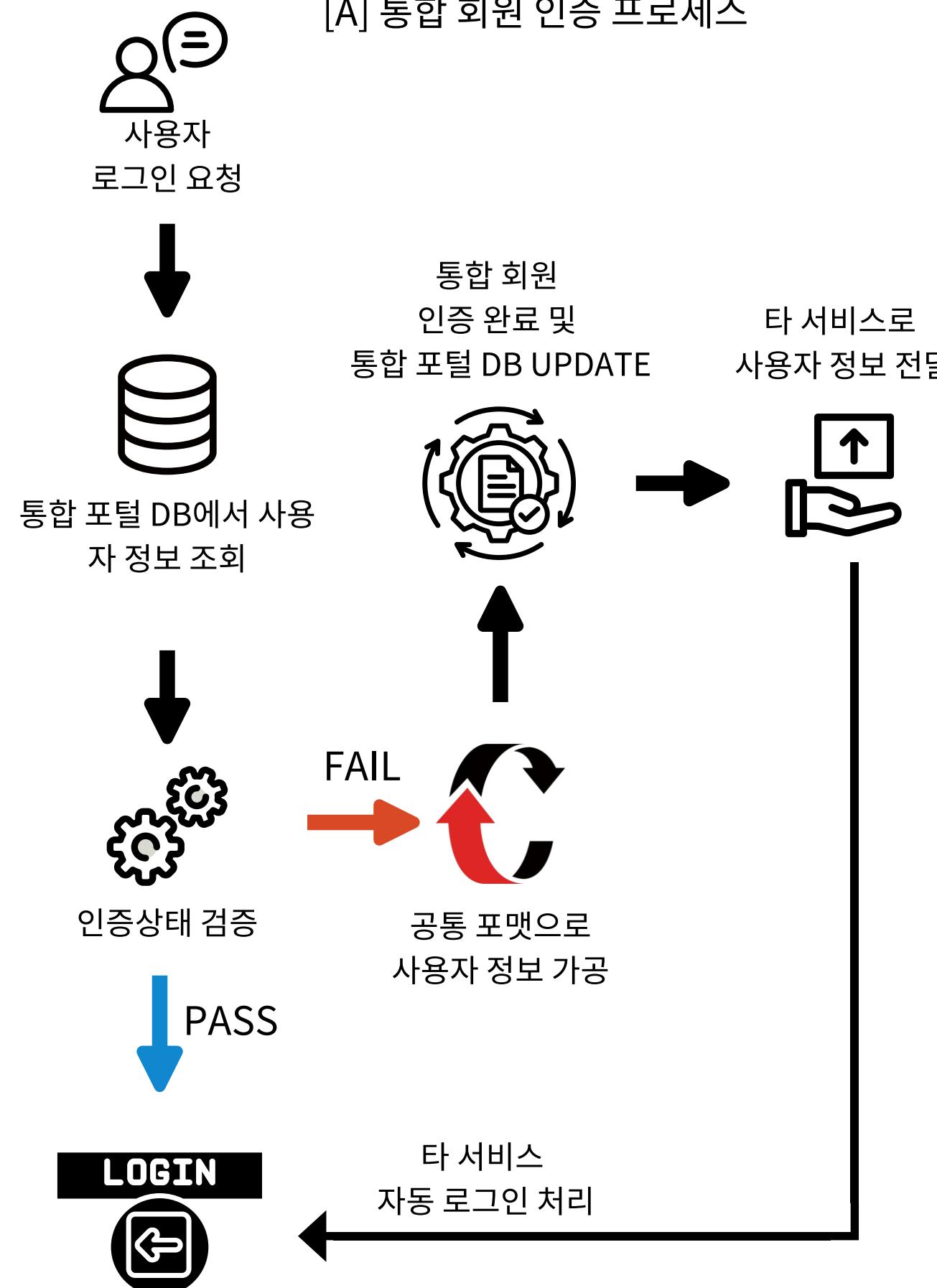
- 초기에 스트레스 테스트와 다양한 케이스 검증을 충분히 준비하지 못해 문제 발견이 늦어진 점
- 로그 및 모니터링 체계를 더 촘촘하게 구성했다면 장애 대응 속도를 높일 수 있었던 점
- 운영 환경을 고려한 테스트 시나리오 설계가 부족했다고 느꼈던 점

02.

Project

무역•투자 통합 플랫폼 인증•권한 고도화

CAREER EXPERIENCE



2. 무역·투자 통합 플랫폼 인증·권한 고도화(백엔드 개발)

기간: 2023.06 ~ 2023.10

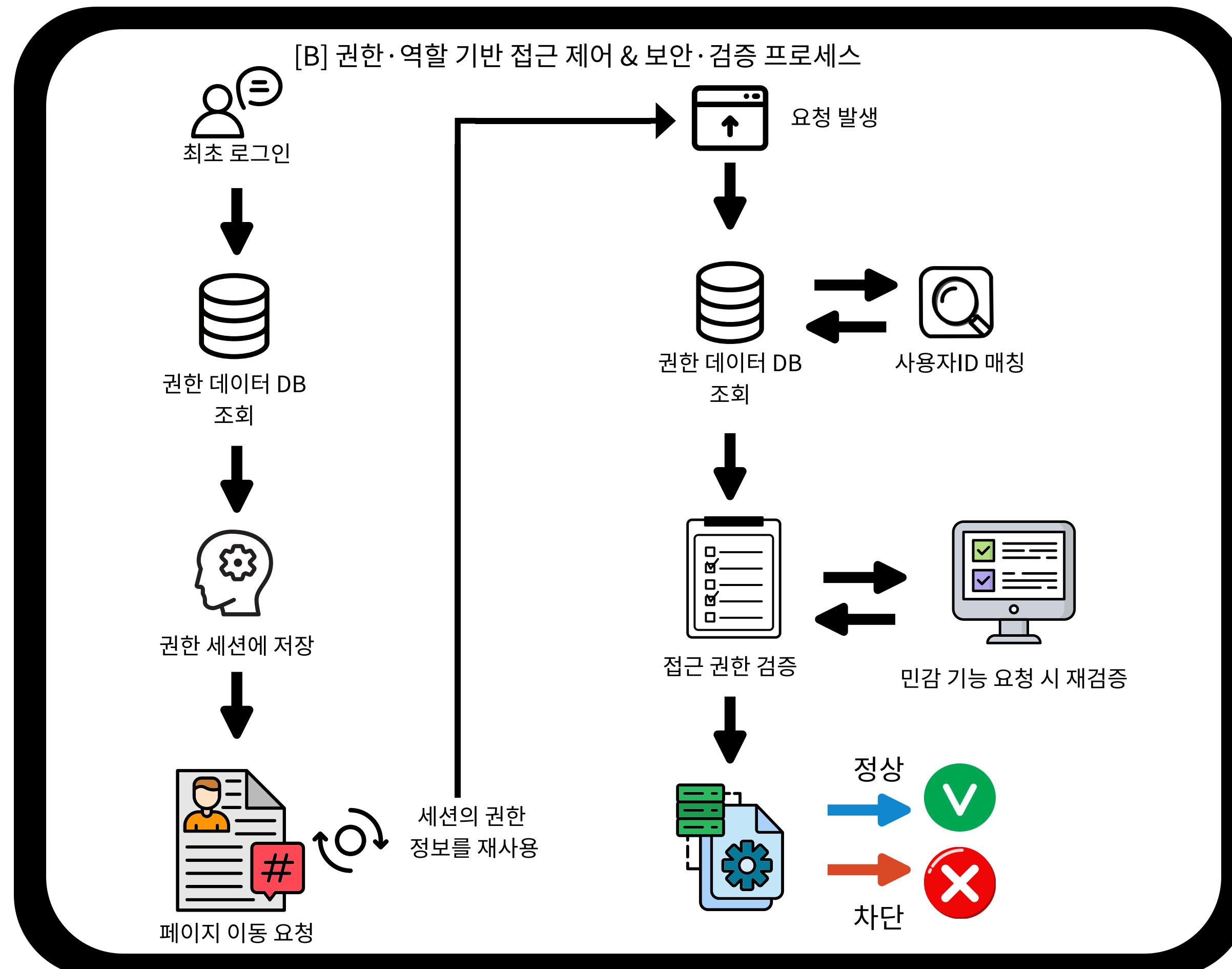
[개요]

국내·외 다양한 무역·투자 기관의 계정 정보를 통합 관리하는 플랫폼으로, 분산된 로그인 시스템을 하나의 인증 체계로 통합하고 기관별 권한·역할을 일관성 있게 관리할 수 있도록 고도화하고 사용자 인증 흐름의 안정성 강화와 권한 기반 접근 제어를 중점적으로 수행했습니다.

[담당 업무]

- [A] 통합 회원 인증 개발
 - 통합 회원 인증 로직 개선
 - 기관별 회원 정보 정규화 및 단일 로그인 흐름 구성
 - 분산된 인증 절차를 하나의 절차로 통합
- [B] 권한·역할 기반 접근 제어 고도화
 - 역할·권한 매핑 및 접근 제어 구조 개선
 - 사용자 그룹별 권한 매핑 및 기능 단위 접근 로직 강화
 - 세션 기반 권한 캐싱 적용으로 반복 조회 제거 및 성능 개선
 - 보안·검증 로직 강화
 - 인증·요청 과정 보안 로직 고도화
 - 세션 무결성 검증 및 토큰 기반 암호화 적용
 - 개인정보 전달 최소화 및 보안 취약점 제거

CAREER EXPERIENCE



CAREER EXPERIENCE

• 트러블 슈팅 : [A] 권한 정보 캐싱 누락으로 인한 접근 오류 문제

⚠ 문제점

로그인 이후 세션에 저장된 권한 정보가 페이지 이동 과정에서 초기화되며 일부 사용자에게 간헐적인 ‘권한 없음’ 오류가 발생했습니다. 특정 사용자 그룹에서만 동일 문제가 반복되어 권한 로딩 시점과 세션 갱신 로직 간 충돌 가능성이 확인되었습니다.

! 해결 과정

권한 정보 로딩·저장을 담당하는 구간을 전수 점검하고, 로그인 완료 시점에서 1회 만 권한을 세션에 적재하는 구조로 재설계했습니다.

페이지 이동 시 중복 검증을 유발하던 권한 조회 로직을 정리했고, 세션 만료 기준·재발급 기준을 독립적으로 구분하여 세션과 DB 간 권한 정보가 일관되게 유지되도록 개선했습니다.

✓ 결과

페이지 이동 시 발생하던 권한 초기화 오류가 더 이상 재현되지 않았습니다. 권한 정보 조회 횟수가 감소하여 전체 응답 속도가 향상되었고, 권한 구조가 복잡한 관리자 계정에서도 정상적으로 동작함을 확인했습니다.

• 트러블 슈팅 : [B] 인증 전달 과정의 개인정보 노출 위험

⚠ 문제점

기관 간 연동 과정에서 사용자 정보를 URL 파라미터로 전달하는 구간이 존재해 민감 정보가 브라우저 히스토리·서버 로그에 노출될 가능성이 있었습니다. 특히 외부기관 연계 시 중간자 공격(MITM)에 취약한 구조였습니다.

! 해결 과정

기존 URL 기반 전달 방식을 제거하고 사용자 정보를 토큰 기반 암호화 전달 방식으로 변경했습니다.

토큰은 서버에서만 복호화되도록 설계했으며, 요청마다 세션 값·사용자 식별자를 교차 검증하는 로직을 추가했습니다.

마지막으로 불필요하게 전달되던 데이터 필드를 정리해 데이터 노출 가능성을 최소화했습니다.

✓ 결과

URL·로그·중간 경유지에 개인정보가 남지 않는 구조로 전환되어 인증 전달 구간의 보안 수준이 크게 향상되었습니다. 기관 간 연동 테스트에서도 동일 취약점이 재발하지 않았습니다.

CAREER EXPERIENCE

▶ 배운점 & 아쉬운 점

↳ 배운점

- 권한·역할 구조는 인증 흐름, 데이터 정규화, 세션 구조가 함께 고려되어야 함을 학습함
- 세션 기반 권한 캐싱처럼 작은 구조 개선도 전체 서비스 응답속도에 직접적인 영향을 줌을 체감함
- 기관 간 인증 전달에서는 보안·운영·개발 관점을 모두 고려한 설계가 필요함
- 민감정보 전달 경로 점검과 토큰 기반 암호화의 중요성을 이해함

↳ 아쉬운 점

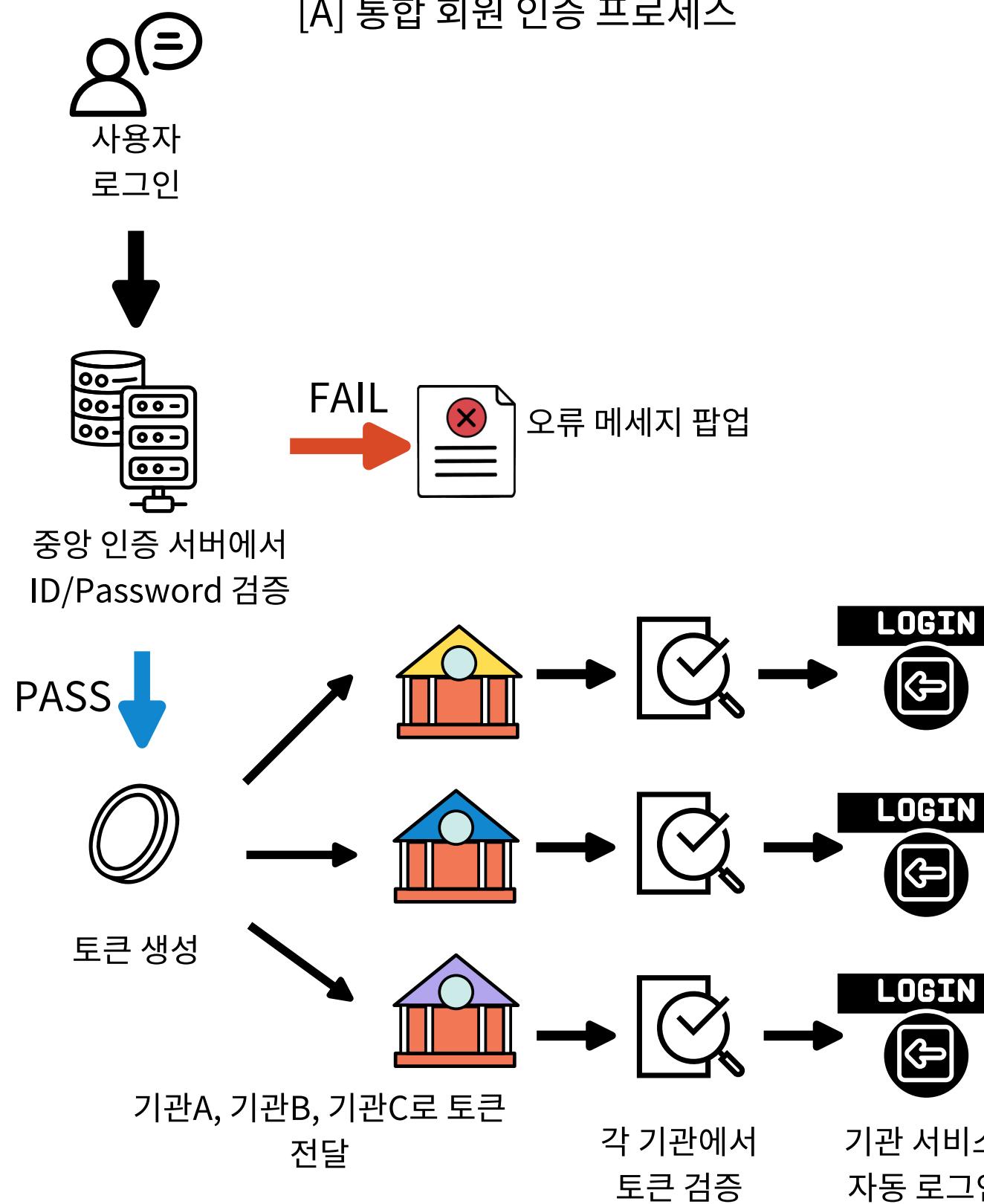
- 초기 단계에서 다양한 사용자·권한 조합 테스트가 충분하지 않아 일부 이슈 발견이 늦어진 점
- 로그인/권한 흐름이 복잡해짐에도 모니터링·로그 체계를 촘촘하게 구성하지 못한 점
- 기관별 인증 규칙을 초기에 정리하지 못해 테스트·검증 과정이 길어진 점

03.

Project

무역기반 SSO 인증 시스템 개발

CAREER EXPERIENCE



3. 무역기반 SSO 인증 시스템 개발(백엔드 개발)

기간: 2022.04 ~ 2022.06

[개요]

여러 무역·통관·지원 기관에서 각각 운영하던 로그인 시스템을 하나의 SSO 체계로 통합하는 프로젝트였습니다. 중앙 인증 서버를 두고, 각 기관은 표준화된 인증 API로 연동하는 구조로 재설계하였습니다.

[담당 업무]

- [A] SSO 인증 시스템 개발
 - 통합 회원 인증 로직 개선
 - 사용자 ID/PW를 중앙 인증 서버에서 검증하는 기본 로그인 흐름 구현
 - 인증 성공 시 기관 시스템으로 전달되는 공통 인증 결과 포맷 구성(세션 생성, 사용자 식별값 매핑 등)
 - SSO 기반 회원가입 연동 기능 개발
 - 기관별 사용자 정보를 통합 회원 테이블로 등록하는 API 개발
 - 기관마다 다른 입력 스펙을 공통 포맷으로 정규화하여 저장 구조 개선
 - 인증 관련 예외 처리 및 검증 로직 보완
 - 로그인 실패·중복 가입·정보 누락 등 기본 인증 예외 처리 구현
 - 요청 로그 및 실패 로그 구조화

CAREER EXPERIENCE

• 트러블 슈팅 : 기관별 회원가입 파라미터 불일치

⚠ 문제점

기관별로 제공하는 가입 정보가 필드명·포맷·필수항목이 제각각이라 통합 회원 테이블에 저장할 때 유효성 오류와 누락이 반복되었습니다.

❗ 해결 과정

기관별 입력 규칙을 비교해 공통 포맷을 먼저 정의하고, 회원가입 API에서 필드 매팅
· 정규화 로직을 추가하여 데이터 저장 과정에서 발생하던 오류를 줄였습니다.

✓ 결과

회원가입 시, 파라미터 불일치로 가입이 안되는 문제가 해결되었고 기관 추가 시 같은 로직을 재사용할 수 있게 되어 연동 시간도 단축되었습니다.

▶ 배운점 & 아쉬운 점

↳ 배운점

- 기관별 규칙이 다른 환경에서는 입력 포맷 정규화가 가장 중요한 기본기임을 체감함
- 로그인·회원가입처럼 단순해 보이는 기능도 예외 처리와 로그 구조 설계가 핵심임을 학습
- 여러 기관과 협업하면서 요구사항 정리·스펙 명확화의 필요성을 크게 느낀

↳ 아쉬운점

- 초기 단계에서 기관별 입력 스펙을 충분히 수집 못하여 수정 작업이 길어졌던 점
- 운영 환경의 다양한 케이스를 미리 준비했다면 문제 발견 속도가 빨랐을 것 같음
- 공통화 문서를 더 체계적으로 관리했더라면 후속 시스템 연동이 한층 더 수월했을 것을 느낌

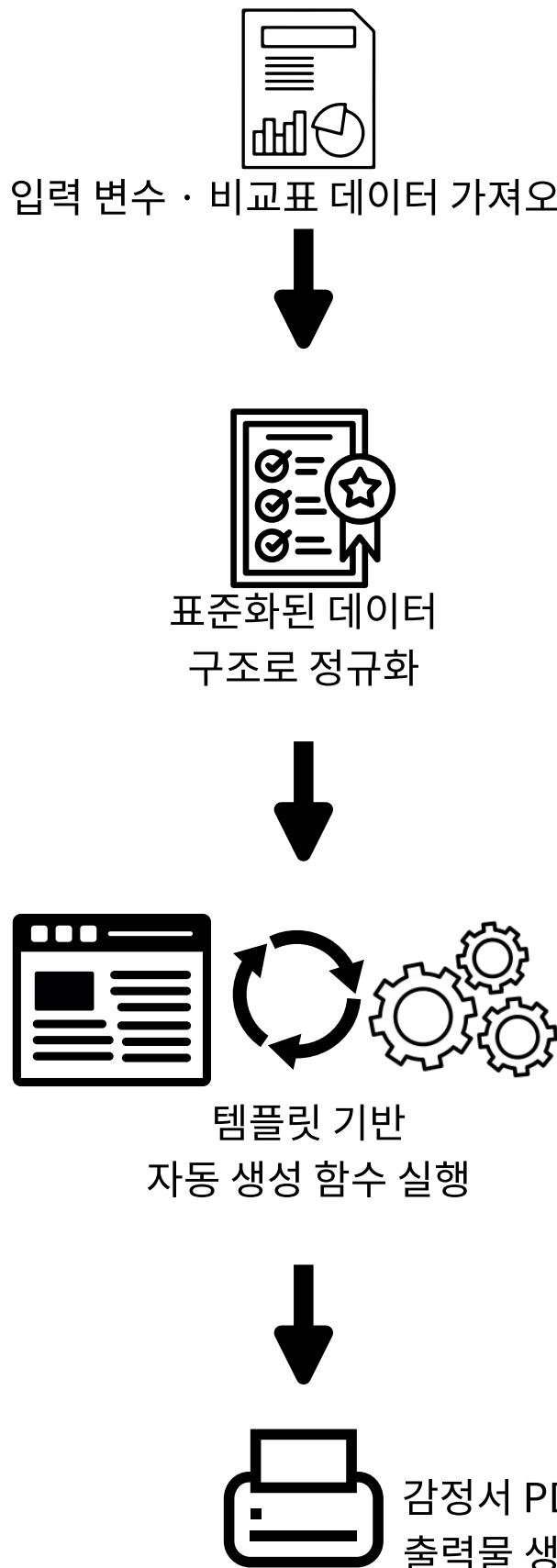
04.

Project

감정평가 자동작성 시스템 개발

CAREER EXPERIENCE

[A] 감정평가 자동 작성 프로세스



4. 감정평가 자동작성 시스템 개발(백엔드 개발)

기간: 2021.10 ~ 2022.04

[개요]

감정평가 업무의 자동화를 목표로 한 시스템으로 감정서 자동작성, 수익환원법 계산 엔진, 대용량 파일 업로드 기능 개선 등 감정평가 실무 효율을 높이는 핵심 기능을 담당했습니다.

[담당 업무]

- [A] 감정평가 자동작성 시스템 개발
 - 감정서 자동작성 기능 개발
 - 감정 항목·비교표 등의 데이터를 조합해 감정서가 자동 생성되도록 로직 구현
 - 데이터 구조를 표준화하여 필드 매팅 오류를 줄이고 유지보수성 향상
 - 수익환원법 계산 함수 개발
 - 수익률을 계산하는 수식 함수개발
 - 계산 모듈을 독립 구조로 분리
 - 대용량 이미지 업로드 및 전송 기능 개선
 - 큰 이미지 파일 업로드 시 응답 지연이 발생하던 문제 개선
 - 비동기 업로드 적용 및 업로드 진행률 표시 기능 구현

CAREER EXPERIENCE

• 트러블 슈팅 : 감정서 자동작성 시 필드 매팅 오류 문제 & 이미지 업로드 시 응답 지연 및 서버 부하 문제

⚠ 문제점

- 감정서 자동작성 기능 초기 버전에서는 감정 항목·비교표 데이터 구조가 화면/DB 마다 달라 필드 매팅이 정확히 일치하지 않아 오류·누락·잘못된 값 생성이 반복되었습니다.
- 고해상도 이미지가 많을 때 업로드 시간이 길어지고, 동기식 처리로 인해 서버에서 응답 지연·스레드 점유 증가 문제가 발생했습니다.

! 해결 과정

- 화면/DB/엑셀 입력 구조를 모두 비교하여 단일 표준 데이터 구조로 통합하고 매팅 규칙을 JSON 스키마 형태로 관리하여 필드 간 불일치가 발생하지 않도록 재설계 했고 자동화 로직이 동작하기 전 단계에서 사전 유효성 검증을 추가했습니다.
- 이미지 업로드를 비동기 방식으로 전환하여 요청-처리 흐름을 분리하고, 서버 부하를 줄이기 위해 이미지 압축·리사이즈 로직을 추가했습니다.
또한 업로드 진행률을 사용자에게 표시하도록 UX를 개선했습니다.

✓ 결과

- 필드 오류 발생 빈도가 크게 감소했고, 감정서 자동 생성 성공률이 안정적으로 올라갔습니다.
- 대용량 이미지 업로드 시 응답 체감 속도가 크게 향상되었고, 서버 CPU 사용량도 안정적으로 유지되었습니다.

▶ 배운점 & 아쉬운 점

- .. 배운점
 - 자동화 기능 개발에서 데이터 구조 통합이 가장 핵심임을 체감함
 - 비동기 처리와 파일 전송 구조의 중요성을 경험함

.. 아쉬운점

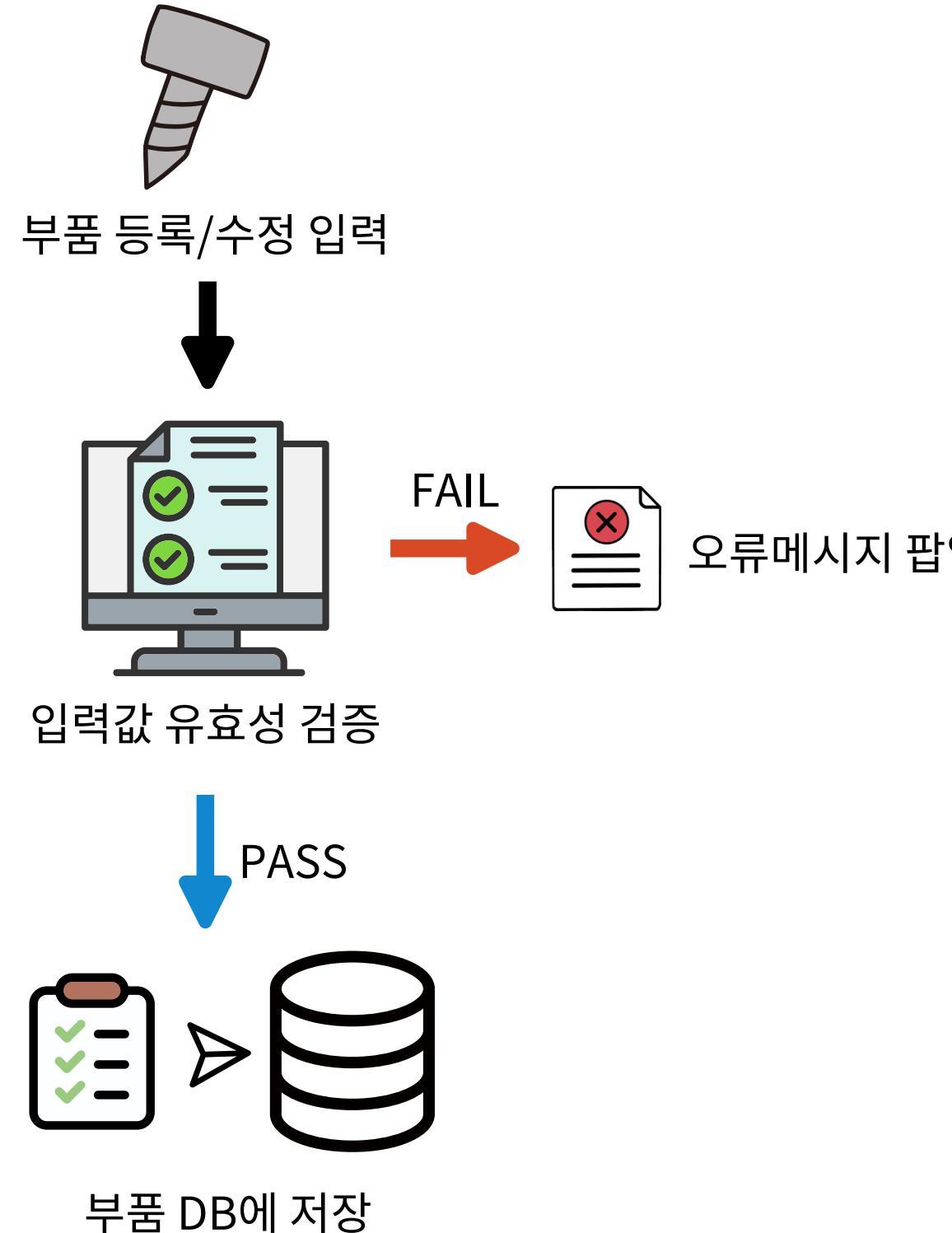
- .. 아쉬운점
 - 감정평가 도메인 지식을 더 빨리 습득했다면 기능 설계 시간을 줄일 수 있었던 점
 - 초기 테스트 케이스가 부족해 예외 상황을 늦게 발견했던 점

05

Project

자동차 재제조 부품 유통 플랫폼

CAREER EXPERIENCE



3. 자동차 재제조 부품 유통 플랫폼 개발(백엔드 개발)

기간 : 2021.08 ~ 2021.10

[개요]

자동차 재제조 부품을 등록·조회·수정·관리하는 플랫폼 개발에 참여했으며, 초기 개발자로서 입력값 검증과 CRUD로직 중심 기능을 담당했습니다.

[담당 업무]

- [A] 자동차 재제조 부품 유통 플랫폼 백엔드 개발
 - 부품 등록·수정 기능 개발
 - 등록/수정 화면 입력값 유효성 검증 처리
 - 검증 실패 시 오류 팝업 및 메시지 노출
 - 부품 데이터 검증 및 저장 로직 개발
 - 검증된 입력값만 DB에 반영되도록 서버단 검증 추가
 - 부품 DB 저장 처리 및 결과 메시지 구성

CAREER EXPERIENCE

• 트러블 슈팅 : 잘못된 형식의 데이터가 DB에 저장되던 문제

⚠ 문제점

부품 코드·수량 등 주요 입력값이 검증 없이 저장되어 데이터 정합성 문제가 반복적으로 발생했습니다.

! 해결 과정

정규식 기반 입력 검증을 추가하고, 서버단에서도 동일한 유효성 검사를 수행하도록 로직을 보완했습니다.

✓ 결과

잘못된 데이터 저장률이 크게 줄고, 관리자 검증 시간이 단축되었습니다.

▶ 배운점 & 아쉬운 점

∴ 배운점

- 클라이언트·서버 이중 검증 구조가 안정성 확보에 중요하다는 것을 경험함
- CRUD 중심 기능에서도 UI 피드백 및 오류 메시지의 품질이 UX에 큰 영향을 미친다는 점을 학습함

∴ 아쉬운점

- 초기 검증 케이스 정리가 부족해 예외 처리 보완 작업이 반복되었던 점

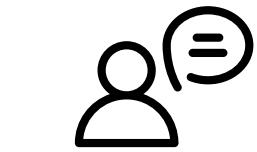
06

Project

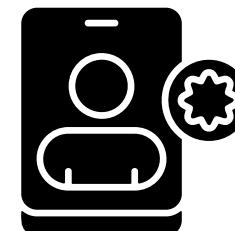
내부 고객 관리 및 공지사항 시스템

CAREER EXPERIENCE

[A] 로그인 후 세션 기반 인증 상태 유지



로그인 요청



세션 생성



사용자 인증 상태 유지

3. 내부 고객관리 및 공지사항 시스템 개발

기간 : 2021.06 ~ 2021.07

[개요]

웹 시스템의 로그인 처리와 사용자 인증 상태 유지를 담당했습니다. 초기 개발자로서 웹 인증 흐름에 대한 기초를 체득한 프로젝트입니다.

[담당 업무]

- [A] 로그인 처리 기능 개발
 - 사용자 로그인 요청을 받아 입력된 ID/Password 검증
 - 인증 성공 시 사용자 정보를 기반으로 세션 생성
 - 세션 기반 인증 상태 유지 로직 개발
 - 세션을 통해 사용자 인증 상태를 유지하는 구조 구현

CAREER EXPERIENCE

• 트러블 슈팅 : 세션 타임아웃 미적용 문제

⚠ 문제점

세션 만료가 정상 동작하지 않아, 로그아웃된 것으로 보이지만 실제로는 인증이 유지되는 문제가 발생했습니다.

! 해결 과정

세션 타임아웃 설정을 재확인하고, 요청마다 세션을 검사하는 로직을 추가했습니다.
만료 시 로그인 페이지로 자동 이동하도록 흐름을 정비했습니다.

✓ 결과

인증 상태 관리가 정상화되었고 내부 서비스 보안이 개선되었습니다.

▶ 배운점 & 아쉬운 점

- ∴ 배운점
 - 로그인 → 세션 생성 → 인증 유지의 기본 웹 인증 흐름을 명확히 이해함
 - 세션 기반 인증 구조를 실무에서 직접 경험한 첫 프로젝트였음

∴ 아쉬운점

- 로깅·모니터링 도구 사용 경험 부족으로 디버깅 시간이 길었던 점

본 경력기술서는 프로젝트 경험을
중심으로 요약한 내용이며,

필요한 부분은 제 업무 경험 범위 안에서
차분하게 설명드릴 수 있습니다.

앞으로도 더 나은 개발자가 되기 위해
꾸준히 배우고 성장하겠습니다.

CHOI SUNGJANG

PORTFOLIO

END