

물류 ERP 시스템 개발

- Git: <https://github.com/hanabe02/vue>
- **주요 역할:**
 - 프론트/백엔드 클라이언트 구현
 - 출고관리 기능 신규 개발
 - 납품관리 기능 개선
 - Google Oauth 2.0 기반 소셜 로그인 기능 구현
 - Spring AOP 기반 공통 로깅 기능 구현
- **개요:**
 - Frontend: Vue3, Nuxt.js, TypeScript
 - Backend: Spring Boot, Java, MyBatis
 - DB: Oracle

건설 ERP

추가

등록

거래처명

관할일자

유종일자

관할담당자

No data available

메인페이지

물류 정보

영업관리

생산 관리

자재 구매 관리

외주 관리

출고 관리

배송 관리

외주발주 등록

외주발주조회

일

시작일&종료일

일

외주번호

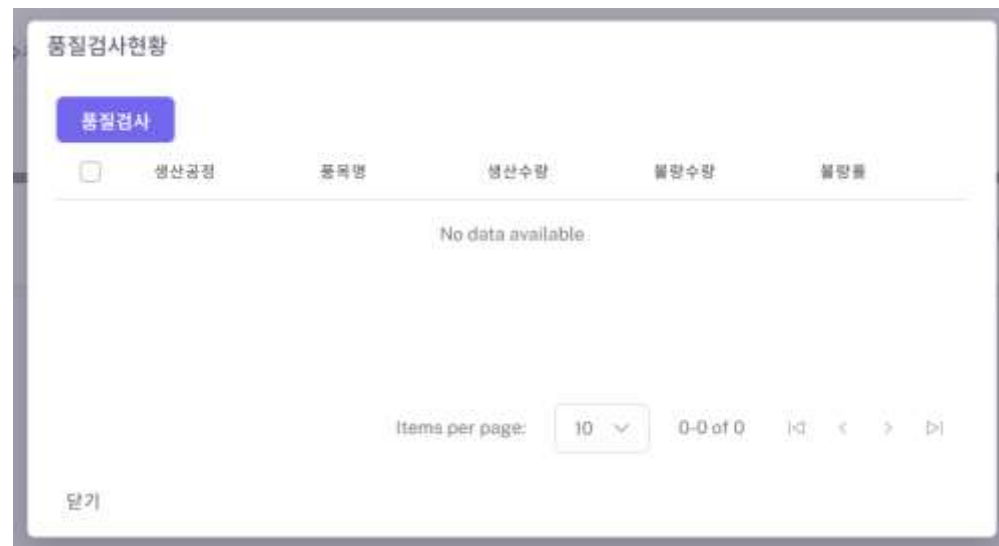
MRP 취합 번호

구매 및 생산여부

품목코드

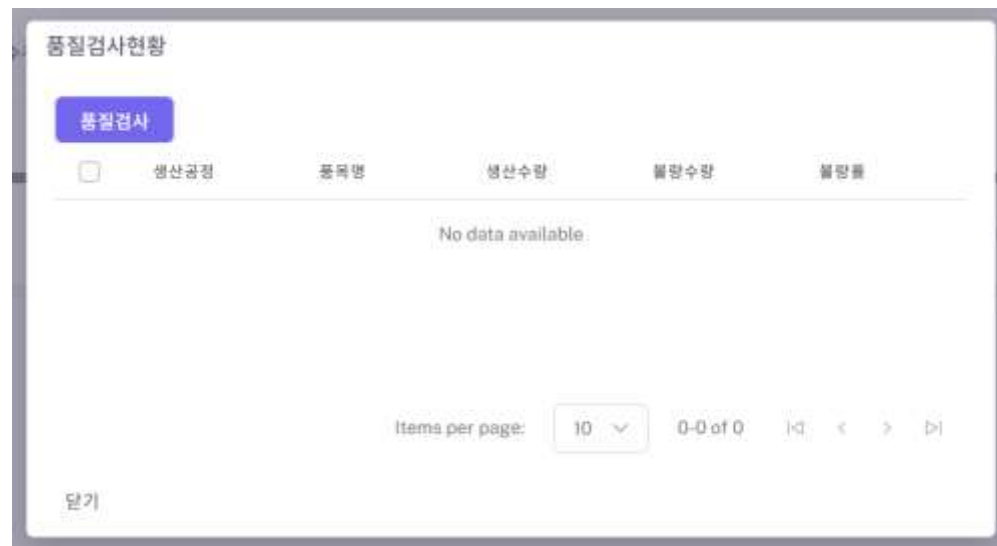
출고 관리

- 기능 개요:
- 출고관리 화면 설계 및 UI 구성
- 출고 등록/ 수정/ 삭제 기능 구현
- 출고 이력 조회 기능 구현
- 품질검사 상태 연동
- (품질검사 결과에 따라 출고 가능 여부를 판단하고 상태 필터링)
- 프론트/백엔드 통합 구현
- (Vue + Nuxt.js 프론트 구현, 백엔드와의 통신을 Axios로 처리)



납품 관리

- 기능 개요:
 - 납품 현황 조회 및 검색 기능 구현
 - (납품처, 품목명 등 다양한 조건 납품 이력 필터링 확인)
 - 납품 등록/ 수정/ 삭제 기능 개선
 - 납기일 관리 기능 추가
 - (납기일 기준 정렬 및 납기 지연 시 시각적 표시 기능 추가)
 - 출고/품질검사 데이터 연동
 - 사용자 편의성 향상



Google OAuth 소셜 로그인

- 기능 개요:
- Google OAuth 2.0 인증 연동
- (사용자가 Google 계정을 통해 로그인할 수 있도록 인증 절차 구현)
- 프론트/백엔드 간 OAuth 토큰 처리 로직 구현
- 회원 정보 자동 등록 및 세션 연동 처리
- 기존 로그인/인증 로직과 통합
- (JWT 기반 로그인과 사용자 세션 처리 방식 유지)
- 보안성 강화
- (HTTPS 기반 통신, OAuth 리디렉션 URI 제한 설정 보안)

Email

admin@demo.com

Password

.....

☐ Remember me [Forgot Password?](#)

Login

Google Login

New on our platform? [Create an account](#)

or

f G t

Google 계정으로 로그인

로그인

vue(으)로 이동

이메일 또는 휴대전화

이메일을 잊으셨나요?

계정 만들기 다음

한국어

도움말 개인정보처리방침 약관

AOP 기반 공통 로깅 구현

- 주요 역할:

- Spring AOP를 활용한 공통 로깅 기능 구현
- 로그 분류 처리
- LoggerAspect 작성
- 공통 예외 처리 및 필터 연동

```
@Around("execution(* kr.co.seoulit.erp..controller..*(..)) || execution(* kr.co.seoulit.erp..servicefacade..*(..))")
public Object logPrint(ProceedingJoinPoint joinPoint) throws Throwable {
    String layer = "";
    String className = joinPoint.getSignature().getDeclaringTypeName();
    String methodName = joinPoint.getSignature().getName();

    if (className.contains(".controller")) {
        layer = "[CONTROLLER]";
    } else if (className.contains(".servicefacade")) {
        layer = "[SERVICE]";
    } else if (className.contains(".dao")) {
        layer = "[DAO]";
    } else {
        layer = "[UNKNOWN]";
    }

    log.info("{} {}.{}, layer, className, methodName);

    return joinPoint.proceed();
}
```

advice

CORSFilter

LoggerAspect

RestfulExceptionHandler

WebConfig

협업 경험

- **협업 방식 및 브랜치 전략**

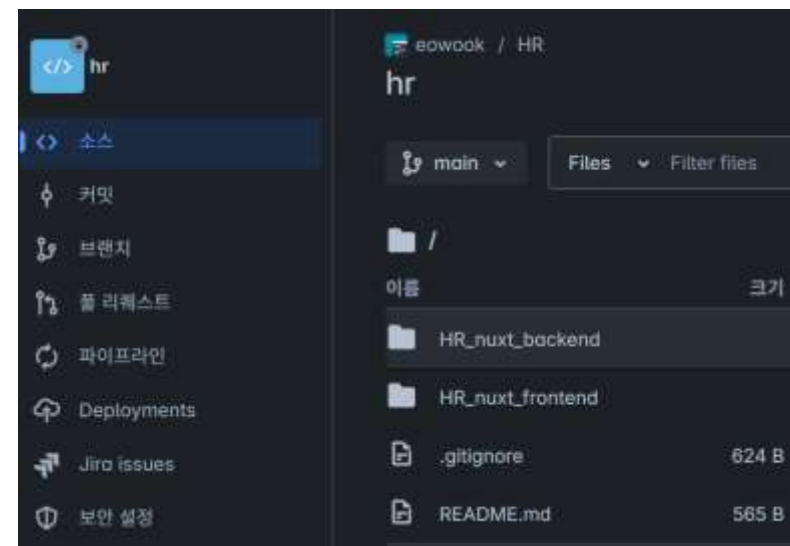
- 3인 팀 프로젝트로 진행, 주요 기능 개발 분담하되,
- 공통 로직 및 추가 기능 개발 과정 팀원 전원 적극 참여
- 형상 관리 Git + Sourcetree + Bitbucket 활용 브랜치 기반

- **브랜치 전략**

- develop 브랜치 기준 feature/기능명 브랜치를 생성
- 병합 시에는 PR을 활용해변경사항을 검토하고, 안정성 확보 후
- develop 에 병합
- 팀원 간 명확한 역할 분담, 브랜치 네이밍 규칙을 통해 충돌 최소화

- **문제 해결 경험**

- Merge 충돌 이슈 발생 시, 테스트 전용 브랜치와
- 별도 레포지토리를 만들어 사전 병합 학습 및 테스트 진행
- 팀원 간 실시간 소통 통해 충돌 원인 파악 해결



느낀점

- 이번 프로젝트를 통해 단순한 기능 구현을 넘어,
- 실제 현업에서 사용하는 개발 방식과 협업 프로세스를 실질적으로 경험할 수 있었습니다.
- ERP 시스템 개발 프로젝트를 통해 단순한 웹 페이지 구현을 넘어,
- 실제 업무 프로세스를 반영한 복잡한 비즈니스 로직을 설계하고 구현하는 경험을 할 수 있었습니다.
- Vue3와 Nuxt.js를 사용하여 컴포넌트 기반 UI를 구성하면서,
- 재사용성과 유지보수성이 높은 프론트엔드 구조를 체계적으로 설계하는 역량을 키웠습니다.
- 특히 **Pinia** 상태관리 도입을 통해 **Vuex**보다 직관적이고 유연한 상태 흐름 관리 방식을 익혔고,
- 출고관리, 납품관리 등 업무 핵심 기능을 구현하며
- 실제 기업 환경에서 필요한 데이터 처리 로직과 화면 흐름에 대한 이해도가 높아졌습니다.
- 또한, Google OAuth 2.0 기반의 소셜 로그인 기능과 Spring AOP를 활용한 **공통 로깅 처리** 구현을 통해
- **보안성과 확장성을 고려한 백엔드 개발 역량**도 함께 강화할 수 있었습니다.