

과 정 명	[하나은행] Digital hana 路 금융서비스개발 6기/7기	제출 기한	8월12일(화) 23:59까지
과 목 명	백엔드 개발 기초/심화	제출 방법	학습사이트 → 과제 제출

## 쇼핑몰 서비스 - 요구사항

### ❖ 관리자(Admin) 기능

1. 상품 등록 / 조회 / 수정 / 삭제 (이미지 포함)
2. 재고 수량 조정
3. 주문 내역 조회 (검색)
4. 매출 통계 조회
5. 회원 관리 (회원 목록 조회, 삭제 등)

### ❖ 일반 사용자(User) 기능

1. 회원 가입 / 로그인
2. 상품 목록 조회 (검색)
3. 상품 상세 보기
4. 장바구니 담기 / 수정 / 삭제
5. 주문 생성 (장바구니 기반)
6. 내 주문 내역 확인

### ❖ 공통 기능

1. Validation
  - 회원가입(이메일, 비밀번호, 닉네임)과 상품 등록(이름, 가격, 재고 등) 시 JSR-303 기반 유효성 검사를 적용
  - 서버측 예외 발생 시, 공통 에러 메시지 형식으로 클라이언트에 전달
2. Exception 처리
  - 공통 에러 메시지 구조 설계 및 적용
3. Batch & Scheduler
  - 매출 통계 등 정기 작업 자동 처리
4. DB 및 테스트 코드
  - 개발 단계에서 수집된 데이터(DB 테이블의 내용)를 외부 파일로 export ('/resources/data/data.sql')
  - 테스트 코드 기반으로 DB 초기 데이터('data.sql') 삽입 및 마이그레이션 정상 수행 여부 확인
5. 성능 모니터링/상태 점검용 actuator

과 정 명	[하나은행] Digital hana 路 금융서비스개발 6기/7기	제출 기한	8월12일(화) 23:59까지
과 목 명	백엔드 개발 기초/심화	제출 방법	학습사이트 → 과제 제출

## 쇼핑몰 서비스 - 기능 명세

기능	내용
인증/인가	Spring Security 기반 인증 (ADMIN vs. USER) <ul style="list-style-type: none"> <li>ADMIN 아이디/비밀번호 → <a href="#">hanaro/12345678</a></li> <li>로그인 시 JWT 발급 및 사용자 인증</li> <li>ROLE 기반 접근 제어 (ROLE_ADMIN, ROLE_USER)</li> </ul>
파일 업로드	<ul style="list-style-type: none"> <li>파일 원본위치 : '/resources/static/origin'</li> <li>업로드 위치 : '/resources/static/upload'</li> <li>업로드된 날짜별 디렉토리 자동 생성 (예. 2025년 07월25일 업로드한 경우 '/resources/static/upload/2025/07/25'에 파일 저장)</li> <li>파일명 중복 방지, 최대 크기: 1개 512KB, 총 3MB 제한</li> </ul>
스케줄링	<ul style="list-style-type: none"> <li>결제 완료되고 → 배송 준비 (매 5분마다 결제 완료된 주문을 배송 준비 상태로)</li> <li>배송 준비 → 배송 중 (매 15분마다 배송 준비 상태를 배송 중 상태로)</li> <li>배송 중 → 배송 완료 (매 1시간마다 배송 중 상태를 배송 완료 상태로) 변환</li> </ul>
배치 Job	<b>매일 자정에 실행</b> <ul style="list-style-type: none"> <li>일별 매출 통계 저장</li> <li>주문 정보를 집계하여 매출액, 주문 수, 상품별 통계 데이터 생성</li> <li>별도의 테이블에 저장 (통계 테이블)</li> </ul>
로깅	주요 이벤트 로깅 <ul style="list-style-type: none"> <li>console: 표준 콘솔 출력</li> <li>상품 관리 및 주문 관련 비즈니스 로직을 각각 별도의 로그 파일로 출력(business_product.log, business_order.log)</li> <li>rolling: 일일 단위 파일 롤링</li> <li>프로젝트 내부 'logs'폴더에 보존</li> </ul>
Swagger(OpenAPI 3)	<ul style="list-style-type: none"> <li>관리자/사용자용 API 명확히 분리 및 문서화</li> <li>Swagger UI로 테스트 가능</li> </ul>
Validation	회원가입, 상품 등록 시 모든 입력 필드에 유효성 검사 적용
Actuator	시스템 헬스(health), 빈(bean) 정보, 환경(env) 정보, metric 정보를 확인하기 위한 엔드포인트 제공

과 정 명	[하나은행] Digital hana 路 금융서비스개발 6기/7기	제출 기한	8월12일(화) 23:59까지
과 목 명	백엔드 개발 기초/심화	제출 방법	학습사이트 → 과제 제출

# 쇼핑몰 서비스 - 프로젝트 설정 및 제출물

## ❖ 기본 설정 및 프로젝트 구조

- Spring Boot + Gradle 기반 프로젝트 (JDK21)
- application.yml 또는 application.properties 환경 분리
- dbms는 mysql

### README.md 예시

```
## 배치 작업 결과 확인
- 매일 00시에 실행 → logs/business_*.log에 기록됨
- 매출 통계 → DB 테이블 `daily_sales_stats` 확인
```

## ❖ 제출 항목

1. 프로젝트명은 **hanaro**
2. **.gradle, .idea, build, .gitignore** 은 압축 대상에서 제외
  - 배치작업, 스케줄러, actuator, 로그 기록등 확인 방법은 README.md에 문서화
  - 평가에 필요한 모든 파일은 압축에 포함 (예. /logs/business-2025-07-25.log, /resources/static/origin/\*. \* 등 )
  - 파일 이름은 본인 이름의 영문이니셜.zip
3. 기능별 API 명세 (Swagger)
  - http://localhost:8080/swagger-ui/index.html로 접속
4. 설계된 모든 테이블의 샘플 데이터
  - 스키마 : **hanarodb**, account: **hanaro**, password:12345678
  - ERD 이미지 또는 \*.mwb로 제출
5. 로그 파일 출력 결과
  - 기능 명세에서 정해진 위치
6. 파일 업로드에 사용된 파일
  - 기능 명세에서 정해진 위치

과 정 명	[하나은행] Digital hana 路 금융서비스개발 6기/7기	제출 기한	8월12일(화) 23:59까지
과 목 명	백엔드 개발 기초/심화	제출 방법	학습사이트 → 과제 제출

## 쇼핑몰 서비스 - 평가 항목 및 기준

문서에 명시된 요구사항, 기능 명세, 프로젝트 설정, 제출 항목을 정확히 준수하지 않을 경우 감점됩니다

기능	평가 기준	배점	기능	평가 기준	배점
인증/인가	<ul style="list-style-type: none"> <li>JWT 발급 및 검증</li> <li>ADMIN / USER 구분</li> <li>ROLE 기반 API 접근 제한</li> </ul>	10	Swagger (OpenAPI 3)	<ul style="list-style-type: none"> <li>사용자/관리자 API 구분</li> <li>Swagger UI로 모든 API 테스트 가능</li> <li>명세 명확</li> </ul>	10
파일 업로드	<ul style="list-style-type: none"> <li>날짜별 경로 저장 (/resources/static/upload/yyyy/MM/dd)</li> <li>파일명 중복 방지</li> <li>파일 크기 제한 (512KB, 총 3MB)</li> </ul>	5	Validation	<ul style="list-style-type: none"> <li>회원가입, 상품 등록 시 @Valid/@Validated 적용</li> <li>필드별 메시지 반환 확인</li> </ul>	5
스케줄링	<ul style="list-style-type: none"> <li>5분 주기: 결제 완료 → 배송 준비</li> <li>15분 주기: 배송 준비 → 배송 중</li> <li>1시간 주기: 배송 중 → 배송 완료</li> <li>DB 반영 확인 가능</li> </ul>	10	Actuator	<ul style="list-style-type: none"> <li>기능 명세에서 요구한 모든 엔드포인트(/health, /metrics 등) 확인 가능</li> </ul>	5
배치 Job	<ul style="list-style-type: none"> <li>자정마다 통계 실행</li> <li>일자별 주문 수, 매출액, 상품별 통계 생성</li> <li>별도 테이블 저장</li> </ul>	15	코드 품질 및 구조화	<ul style="list-style-type: none"> <li>Controller / Service / Repository 분리</li> <li>중복 코드 최소화</li> <li>명확한 네이밍과 로직 정리</li> </ul>	10
로깅	<ul style="list-style-type: none"> <li>콘솔 로그, business_*.log, rolling log 분리</li> <li>로그 파일 보관 디렉토리 구성</li> </ul>	10	예외 처리 및 에러 응답 명세화	<ul style="list-style-type: none"> <li>공통 예외 처리 (@ControllerAdvice, @ExceptionHandler) 구현</li> <li>Swagger를 통해 예외 응답(4xx / 5xx)의 형식 및 메시지 구조를 문서화 및 확인 가능</li> <li>커스텀 예외 메시지 및 에러 코드 반환</li> </ul>	5
기본 기능 구현	<ul style="list-style-type: none"> <li>관리자 및 사용자 필수 기능 전체 구현 여부</li> <li>필수 기능 목록: 상품 등록/조회/수정/삭제, 회원가입, 로그인, 장바구니, 주문 등</li> </ul>	15			