

프로젝트 일정 계획서

과목정보	과 목 명	프로젝트	과목코드	EO310001
	담당교수	지상문	제출일자	2025.03.12
수행 형태	단독			
프로젝트명	스타벅스 스토어 클론코딩을 통한 리테일 웹서비스 제작하기			
팀 구성원	추지우			

순서	진 행 주 제
1	프로젝트 기획 및 요구사항 분석
2	시스템 설계 및 ERD 작성
3	Spring Boot 기반 백엔드 개발 (1)
4	Spring Security 및 OAuth 2.0 적용
5	Spring 기반 백엔드 개발 (보완)
6	상품 관리 및 장바구니 기능 구현
7	주문 및 결제 시스템 개발
8	관리자 페이지 개발 및 대시보드 구축
9	배포 및 최적화
10	테스트 및 프로젝트 마무리

프로젝트 설명

기존에 존재하는 스타벅스 스토어의 기능 및 요구사항을 분석함으로써 실제 로직이 어떠한지 학습합니다. 또한, 리테일 웹서비스의 필수 기능인 보안, 결제, 관리자 페이지, 배포등의 세부 기능들을 직접 적용합니다, 이를 통해 로직 분석, 백엔드 개발 및 데브옵스 개발 역량을 강화시킵니다.

구체 계획

- 1주차 - 프로젝트 기획 및 요구사항 분석
- 프로젝트 목표 및 기능 정의
 - 스타벅스 공식 웹사이트 분석
 - 사용자 요구사항 정리 (유스케이스 다이어그램)
 - 기술 스택 확정 및 개발 환경 설정 (IntelliJ, VSCode, MySQL Workbench 등)
- 2주차 - 시스템 설계 및 ERD 작성
- 데이터베이스 ERD 설계 (MySQL)
 - API 설계 (RESTful API 엔드포인트 정의)
 - 화면 와이어프레임 제작 (Figma, Adobe XD)
 - GitHub Repository 생성 및 브랜치 전략 수립 (Git Flow)
- 3주차 - Spring Boot 기반 백엔드 개발 (1)
- Spring Boot 프로젝트 세팅 (Gradle 또는 Maven)
 - JPA 및 MyBatis 설정, DB 연결 테스트
 - 회원 가입 및 로그인 기능 구현 (JWT 또는 OAuth 2.0)
 - 예외 처리 및 글로벌 예외 핸들링 적용
- 4주차 - Spring Security 및 OAuth 2.0 적용
- Spring Security 설정 및 JWT 인증 적용
 - OAuth 2.0을 활용한 소셜 로그인 (Google, Kakao, Naver)
 - 사용자 권한(Role) 및 보안 정책 설정
 - Redis를 활용한 세션 관리
- 5주차 - 프론트엔드 개발 (1) - 기본 UI 구현
- HTML/CSS, JavaScript를 활용한 기본 레이아웃 구현
 - Thymeleaf 적용 및 템플릿 엔진 설정
 - 반응형 웹 디자인 적용 (Bootstrap 또는 Tailwind CSS)
 - 로그인/회원가입 UI 개발
- 6주차 - 상품 관리 및 장바구니 기능 구현
- 상품 목록 조회 및 상세 페이지 개발
 - 장바구니 담기, 삭제 기능 구현
 - JPA 활용한 상품 데이터 관리
 - RESTful API 설계 및 데이터 연동
- 7주차 - 주문 및 결제 시스템 개발
- 주문 생성 및 결제 API 연동 (카카오페이, 토스페이먼츠 등)
 - 결제 상태 관리 및 주문 내역 조회 기능 구현
 - Spring Batch 활용한 주문 데이터 처리 자동화
 - 이메일 및 SMS 알림 시스템 적용
- 8주차 - 관리자 페이지 개발 및 대시보드 구축
- 관리자 페이지 개발 (상품 관리, 주문 관리 기능 포함)
 - 매출 데이터 시각화 대시보드 구축 (Chart.js, Recharts 활용)
 - Elasticsearch 적용하여 검색 기능 최적화
 - 관리자 로그인 및 권한 관리 추가
- 9주차 - 배포 및 최적화
- AWS EC2, S3, RDS 환경에서 배포
 - Docker를 활용한 컨테이너화 및 배포 자동화 (GitHub Actions)
 - 성능 최적화 (Redis 캐싱, Lazy Loading 적용)
 - SSL 적용 및 HTTPS 설정
- 10주차 - 테스트 및 프로젝트 마무리
- JUnit 및 Mockito를 활용한 단위 테스트 수행
 - 통합 테스트 및 성능 테스트 진행 (JMeter)
 - 버그 수정 및 코드 리팩토링
 - 프로젝트 발표 및 문서화 정리 (API 명세서, ERD 공유)

라이브러리 및 프레임워크

- **Backend**: Java, Spring Boot, Spring MVC, Spring Security, OAuth 2.0, JPA, MyBatis
- **Frontend**: HTML, CSS, JavaScript, Thymeleaf, React (선택사항)
- **Database**: MySQL, Redis (세션 관리)
- **API & Deployment**: AWS (EC2, S3, RDS), Docker, GitHub Actions (CI/CD)
- **Authentication**: OAuth 2.0 (Google, Kakao, Naver 로그인)
- **DevOps**: Docker, Jenkins (또는 GitHub Actions)