

## 프로젝트 기획안 (1차)

2025년 10월 22일

프로젝트 조명	<b>Coco</b> (Cosmetic + Comment)
프로젝트 팀원	<p><b>팀명:</b> <b>Coco</b>(Cosmetic + Comment): 화장품과 후기라는 핵심 요소를 결합한 이름</p> <p><b>팀장:</b> 신태엽 (1명)</p> <p>- <b>주요 역할:</b> 프로젝트 총괄 PM (Project Manager)</p> <p>- <b>상세 책임:</b></p> <ul style="list-style-type: none"><li>• 최종 의사결정 및 팀원 간 의견 조율</li><li>• WBS 기반 전체 개발 일정 수립 및 진행 상황 관리</li><li>• GitHub 브랜치 전략 수립 및 코드 머지(Merge) 관리</li></ul> <p><b>팀원:</b> 이규리, 이준범, 임응규, 최승훈, 추현민 (5명)</p>
프로젝트 주제	<p><b>사용자 후기 중심의 뷰티 아이템 쇼핑몰 구현</b> - '(가칭)뷰티 트윈' 매칭과 AI 기반 개인화를 통해 구매 실패 없는 쇼핑 경험을 제공합니다.</p> <p>단순 상품 판매 쇼핑몰을 넘어, 사용자의 상세한 피부 프로필 데이터를 기반으로 나와 꼭 맞는 '뷰티 트윈'을 찾아주는 커뮤니티를 구축합니다. 신뢰도 높은 리뷰 데이터를 AI로 분석하여 정보 탐색부터 구매까지의 전 과정에서 개인에게 최적화된 새로운 쇼핑 경험을 제공하는 것을 목표로 합니다.</p>

## 타겟층 및 벤치마킹 사이트

### 타겟층

- **주요 타겟:** 광고보다 실제 사용자의 후기를 신뢰하며, 구매 전 뷰티 커뮤니티나 리뷰 앱에서 꼼꼼하게 정보를 탐색하는 20~30대 여성. 특히 자신과 비슷한 피부 타입을 가진 사람의 후기를 적극적으로 찾아보는 '체리슈머(Cherry-sumer)'를 핵심 고객으로 설정한다. 20~30대 남성. 후기를 작성하기 어려운 사람들을 위해 후기 가이드라인을 설정하여 후기 기록을 어렵지 않게 도움을 주도록 독려한다.
- **확장 타겟:** 최신 뷰티 트렌드에 민감하고, SNS를 통해 자신의 경험을 공유하는 것을 즐기는 10대 후반 ~ 20대 초반 사용자.

### 벤치마킹 사이트

- **화해:** 성분 분석과 사용자 리뷰를 결합하여 신뢰도를 확보한 대표적인 뷰티 앱. 특히 사용자 피부 타입에 기반한 리뷰 필터링 기능과 랭킹 시스템은 본 프로젝트의 핵심 기능을 구현하는 데 가장 중요한 벤치마킹 대상이다.
- **오늘의집:** 인테리어 커뮤니티에서 시작하여 커머스로 성공적으로 확장한 사례. 사용자들이 직접 올린 콘텐츠(사진, 노하우)가 어떻게 자연스럽게 구매로 이어지는지에 대한 비즈니스 모델과 커뮤니티 활성화 전략을 벤치마킹하기에 좋다.
- **올리브영 (온라인몰):** 국내 1위 H&B 스토어로서, 방대한 상품 데이터와 실시간 랭킹, 사용자 리뷰 시스템의 UI/UX를 참고하기 좋다. 특히 직관적인 상품 카테고리 구성과 구매 프로세스는 쇼핑몰의 기본 기능을 설계할 때 좋은 레퍼런스가 된다.

## 프로젝트 수행 방향

### (주요 기능 설명)

본 프로젝트는 단순 상품 판매를 넘어, '데이터 기반의 신뢰도 높은 리뷰 커뮤니티'를 구축하는 것이 핵심 목표입니다. 이를 위해 기본적인 쇼핑몰 기능 위에 고도화된 개인 맞춤 기능과 사용자의 참여를 유도하는 차별화된 리뷰 시스템을 구현합니다.

#### 1. 기본 쇼핑몰 기능 구현

- **회원/상품/주문/결제:** 안정적인 이커머스의 기반이 되는 회원 관리, 상품 조회, 장바구니, 주문/결제, 관리자 페이지 등 필수 기능을 우선적으로 구축을 진행합니다.

#### 2. 고도화된 개인 맞춤 기능

- **'내 정보로 필터링' 원클릭 기능:** 사용자가 저장한 피부 프로필(타입, 톤, 고민)을 기반으로, '내 정보로 필터링' 버튼 하나만 누르면 모든 리뷰를 즉시 필터링하여 자신에게 유의미한 정보만 빠르게 탐색할 수 있는 직관적인 환경을 제공합니다.
- **'뷰티 트윈' 매칭 및 추천:** 사용자의 프로필 및 구매 이력을 분석하여 가장 유사한 사용자('뷰티 트윈')를 매칭합니다. '뷰티 트윈'의 리뷰와 구매 상품을 우선 노출하여 광고보다 신뢰도 높은 맞춤형 추천을 구현합니다.

#### 3. AI 기술을 활용한 쇼핑 경험 혁신

- **AI 이미지 기반 상품 검색:** 사용자가 SNS 등에서 본 메이크업 이미지를 업로드하면, AI가 색상과 질감을 분석하여 쇼핑몰 내 유사 상품을 찾아주는 직관적인 검색 기능을 제공합니다.
- **AI 리뷰 요약 및 키워드 추출:** 방대한 텍스트 리뷰를 AI가 분석하여 핵심 내용을 요약하고, #보습감, #지속력 등 핵심 키워드를 추출하여 시각화함으로써 사용자가 제품의 특징을 빠르게 파악할 수 있도록 도움을 줍니다.

#### 4. 신뢰도 및 참여를 높이는 리뷰 시스템

- **리뷰어 보상 및 랭킹:**

- 양질의 리뷰어에게 꿀팁 전도사, 파운데이션 전문가 등의 배지를 부여하고 '명예의 전당' 랭킹 시스템을 도입하여 사용자의 적극적인 참여와 양질의 리뷰 작성을 유도합니다.

- **체계적인 리뷰 작성 환경 구축:**

- 항목별 상세 별점: 전체 별점 외에 발림성, 보습력 등 세부 항목에 대한 별도 평가를 도입하여 리뷰의 구체성을 높입니다.
- 맥락을 담은 포토 리뷰: 사진 첨부 시 자연광/실내조명, 필터 유무 등 촬영 환경 정보를 태그로 추가하여 포토 리뷰의 신뢰도를 확보합니다.
- 리뷰의 가이드 라인: 리뷰의 작성이 어려운 사람들을 위해 가이드라인을 제공하여 리뷰의 작성을 활성화합니다.

- **저평점 리뷰의 데이터화 및 활용:**

- 별점 1~2점 리뷰 작성 시, #트러블유발, #기대이하 등 '실패 원인' 태그를 필수로 선택하게 합니다.
- 이 데이터를 활용하여 비슷한 피부 타입의 사용자에게 "고객님과 같은 피부 타입의 사용자 65%가 #트러블유발 리뷰를 남겼어요."와 같은 '구매 경고 알림'을 제공하고, 더 적합한 대안 상품을 추천합니다.

- **별점 신뢰도 확보:**

- 최소 10개 이상의 리뷰가 누적된 상품에만 평균 별점을 노출하여, 소수 리뷰에 의한 평점 왜곡을 방지합니다.

## 프로젝트 수행 도구 1. 형상 관리 (Version Control)

- **Git:** 소스 코드의 버전을 관리하고 변경 이력을 추적.
- **GitHub:** Git으로 관리되는 프로젝트를 원격으로 저장하고 팀원들과 코드를 공유하는 플랫폼.

## 2. 통합 개발 환경 (IDE)

- **Eclipse 및 IntelliJ IDEA:** Spring Boot 기반의 백엔드 개발을 위한 통합 개발 환경.
- **Visual Studio Code:** 프론트엔드(React, HTML/CSS 등) 개발을 위한 코드 에디터.

## 3. 협업 및 커뮤니케이션 (Collaboration & Communication)

- **Slack:** 팀원 간의 실시간 소통, 개발 관련 이슈 공유 및 빠른 의사결정을 위한 메신저.
- **Notion:** 프로젝트 기획, 회의록 작성, WBS(작업분업구조) 등 문서 관리 및 진행 상황 공유를 위한 협업 툴.

## 4. 디자인 및 기획 (Design & Planning)

- **Figma:** UI/UX 디자인, 와이어프레임 및 프로토타입 제작을 위한 디자인 툴.
- **FigJam / ERDCloud / draw.io:** 데이터베이스 스키마 설계를 위한 ERD(개체-관계 다이어그램) 작성 툴.

## 5. 데이터베이스 관리 (Database Management)

- **SQL Developer:** Oracle 데이터베이스에 연결하여 데이터를 조회하고 관리하는 툴.

- **JPA (Java Persistence API):** 애플리케이션의 객체와 데이터베이스 테이블 간의 매핑을 관리하고, 객체지향적인 방식으로 데이터를 영속화하는 기술.

#### 프로젝트 목적

본 프로젝트의 목표는 **사용자 후기 중심의 뷰티 아이템 쇼핑몰**을 구현하는 것입니다. 소비자는 본 서비스를 통해 나와 꼭 맞는 '**뷰티 트윈**'을 발견하고, 그들의 솔직하고 상세한 리뷰를 통해 **화장품 구매 실패 확률을 획기적으로 줄일 수 있습니다**. 또한, AI 추천과 같은 기술을 통해 자신에게 **최적화된 제품**을 쉽고 재미있게 찾는 새로운 쇼핑을 경험하게 됩니다.

#### 필수 기능

##### 필수 기능 우선순위

[1순위-Core] 기능의 완성을 목표로 진행.

- **[1순위-Core]: MVP의 핵심.** 이 기능이 없으면 서비스의 핵심 가치를 증명할 수 없음.
- **[2순위-Standard]: MVP 출시 이후, 서비스의 완성도와 사용자 편의성을 높이기 위한 주요 기능.**
- **[3순위-Expansion]: 서비스 고도화 및 추가적인 가치 창출을 위한 확장 기능.**

##### A. 기본 쇼핑몰 기능

- **회원 관리:**

- 회원가입, 로그인/로그아웃 [1순위-Core]
- 마이페이지 (주문 내역, 회원 정보 수정 등) [1순위-Core]
- 소셜 로그인 (카카오, 네이버 등) [2순위-Standard]

- **상품:**

- 카테고리별 상품 목록 조회 [1순위-Core]  
(예시)
  - 스킨케어: 클렌저, 토너/패드, 에센스/세럼, 크림, 선크어
  - 메이크업: 파운데이션/쿠션, 아이섀도우, 립스틱/틴트
  - 바디/헤어: 바디워시, 바디로션, 샴푸, 트리트먼트
- 상품 상세 정보 페이지 [1순위-Core]
- 상품 검색 기능 [1순위-Core]

- **주문/결제:**

- 장바구니 기능 [1순위-Core]
- 주문하기 (배송지 입력, 쿠폰/포인트 사용) [1순위-Core]
- 결제 시스템 연동 [1순위-Core]
- 주문 내역 조회 [1순위-Core]

- **관리자 페이지:**

- 상품 등록 및 재고 관리 [1순위-Core]
- 회원 관리 (권한, 상태) [1순위-Core]
- 주문 관리 (주문 확인, 배송 상태 변경) [1순위-Core]

## B. 리뷰 중심 특화 기능

- **상세한 회원 프로필:**

- 필수: 피부 타입(건성, 지성, 복합성, 민감성), 피부 톤(쿨톤, 웜톤, 뉴트럴톤) [1순위-Core]

- 선택: 주요 피부 고민(모공, 트러블, 홍조 등), 퍼스널 컬러 [2순위-Standard]

- 리뷰 작성 기능 (CRUD):

- 별점 평가 (0.5 단위) [1순위-Core]

- 텍스트 리뷰 (최소/최대 글자 수 제한) [1순위-Core]

- (필수) 사진/영상 첨부 기능 [1순위-Core]

- 장점/단점 요약 입력 [1순위-Core]

- (필수) 구매 인증 로직을 통한 리뷰 신뢰도 확보 [1순위-Core]

- '#건성추천', '#인생템'과 같은 사용자 정의 해시태그 기능리뷰 탐색 기능

- 다중 필터: 회원 프로필 정보(피부 타입, 톤, 고민)를 기반으로 리뷰 필터링 [1순위-Core]

- 정렬: 최신순, 도움순, 별점 높은/낮은 순 [1순위-Core]

- '도움이 돼요': 유용한 리뷰를 추천하는 기능 [2순위-Standard]

- 커뮤니티/참여 유도 기능:

- 뷰티 트윈: 나와 프로필(피부 타입, 톤 등)이 유사한 사용자 찾기 및 팔로우 [1순위-Core]

- 리뷰어 랭킹/배지: 우수 리뷰 작성자에게 포인트, 배지 등 보상 제공 [3순위-Expansion]



포함 기술

Backend

- **Framework: Spring Boot 3.2.x**
  - **선정 이유:** 강력한 IoC/DI와 AOP를 통해 객체지향적인 설계가 가능하며, 내장 WAS 및 자동 설정으로 개발 생산성이 매우 높음. 방대한 레퍼런스와 커뮤니티를 통해 안정적인 개발 및 유지보수가 용이하여 선택.
- **Language: Java 21**
  - **선정 이유:** 오랫동안 안정성이 검증된 객체지향 언어이며, Spring 프레임워크와의 완벽한 호환성 및 풍부한 라이브러리를 갖추고 있어 엔터프라이즈급 애플리케이션 개발에 적합.

- **Data Persistence: JPA (Java Persistence API)**

- **선정 이유:** SQL 중심의 개발에서 벗어나 객체 중심으로 데이터를 관리할 수 있어 생산성이 높고 유지보수가 편리합니다. 데이터베이스 교체가 용이하다는 장점을 가지고 있습니다.

- **Security: Spring Security 6**

- **선정 이유:** Spring 기반 프로젝트에서 인증/인가를 구현하는 사실상의 표준으로, 보안 관련 기능을 직접 구현하는 것보다 안정적이고 편리하여 선택했습니다.

## Frontend

- **Core: HTML5, CSS3, JavaScript (ES6+)**

- **Library/Framework: React 18.x**

- **선정 이유:** 컴포넌트 기반 구조로 UI의 재사용성과 유지보수성이 뛰어나고, Virtual DOM을 통해 동적인 화면에서도 높은 성능을 보장합니다. 또한, 풍부한 생태계 덕분에 개발 중 문제 해결이 용이하여 저희 프로젝트에 가장 적합하다고 판단했습니다.

- **State Management: Redux or Recoil**

- **선정 이유:** 사용자 로그인 정보, 장바구니, 피부 프로필 데이터 등 여러 컴포넌트가 공유해야 하는 상태(State)를 하나의 중앙 저장소(Store)에서 관리합니다. 이를 통해 복잡한 데이터 흐름을 예측 가능하고 체계적으로 관리할 수 있습니다.

- **Server-Client-Communication: Axios**

- **선택 이유:** 컴포넌트 기반 아키텍처를 통해 UI를 재사용하고 체계적으로 관리 가능. 특히 사용자와의 인터랙션이 많고 동적인 화면 구성이 필요한 저희 프로젝트에 적합하여 선택.

## Database

- **Oracle 21c**

- **선택 이유:** 대용량 데이터 처리와 트랜잭션 관리에 있어 높은 성능과 안정성을 제공하는 RDBMS의 표준입니다.

## Tools & Architecture

- **VCS (버전 관리): Git, GitHub**

- **선택 이유:** 분산 버전 관리 시스템으로 팀원 간의 협업을 원활하게 하고, 코드 변경 이력을 체계적으로 관리할 수 있어 선택했습니다.

- **Build Tool: Maven**

- **선택 이유:** 프로젝트의 의존성(라이브러리)을 쉽고 체계적으로 관리할 수 있으며, 빌드 과정을 자동화하여 개발 생산성을 높여 줍니다.

- **WAS: Embedded Tomcat (Spring Boot 내장)**

- **선택 이유:** Spring Boot에 내장된 Tomcat을 사용하면 별도의 웹 서버(WAS)를 설치하고 설정하는 과정 없이, 프로젝트를 독립적으로 실행 가능한 JAR 파일 하나로 쉽게 배포할 수 있습니다. 이를 통해 개발 및 배포 과정이 매우 단순화되어 생산성이 향상되고, 서버 환경을 일치시켜 안정적인 운영이 가능합니다.

- **Architecture: MVC (Model-View-Controller) 패턴**

- **선정 이유:** 비즈니스 로직과 사용자 인터페이스를 명확하게 분리하여 코드의 재사용성을 높이고 유지보수를 용이하게 만드는 디자인 패턴이므로 채택했습니다.

[현대이지웰] Java 풀스택 개발자 아카데미

☐ WBS 첨부(OR 별도로 폴더에 업로드)