

상세설계서

팀명 : 나혼자한다

팀장 : 서다슬

팀원 : 정솔, 정세홍, 이준혁

목차

1. 서론

- 1.1. 개요
- 1.2. 주요 기능
- 1.3. 개발 환경

2. System 설계

- 2.1. Modules 개요
- 2.2. Modules 상세 설계
- 2.3. Data 설계
- 2.4. 판매량 예측 Model 설계

3. UI

- 3.1. 메인 화면
- 3.2. 물품 상세 화면
- 3.3. 장바구니 화면
- 3.4. 마이페이지 화면
- 3.5. 로그인 화면
- 3.6. 회원가입 화면
- 3.7. 관리자 화면

4. WBS

1. 서론

1.1. 개요

본 문서는 스프링 시큐리티를 통해 관리자와 유저를 구분하여 관리자는 관리자 페이지를 제공함으로써, 각 매장들의 판매실적을 활용해 다음날 각 물품의 재고를 예측하고 효율적으로 관리할 수 있는 홈쇼핑 웹 서비스 개발에 대한 상세설계이다.

1.2. 주요 기능

1.2.1. View

- 1) 로그인 : 로그인을 할 수 있는 **View**를 제공한다.
- 2) 회원가입 : 회원가입을 할 수 있는 **View**를 제공한다.
- 3) 메인 화면 : 모든 물품 리스트를 바둑판 형식으로 나타낸다.
- 4) 물품 상세 : 물품 이미지, 물품명, 물품금액 등 물품에 관련된 정보를 제공한다.
- 5) 장바구니 : 유저가 예약한 물품 목록들을 나타낸다.
- 6) 마이페이지 : 유저 개인정보 확인 및 수정 가능, 구매내역 확인할 수 있다.
- 7) 관리자 : 5일간 매출과 월간 매출을 차트로 볼 수 있고 판매량 예측을 제공받는다.

1.2.2. DataBase

- 1) SiteUser : 사용자의 개인정보를 저장한다.
- 2) Product : 상품의 정보들을 저장한다.
- 3) Cart : 유저가 선택한 **Product** 정보를 저장한다.
- 4) CartItem : 유저가 선택한 **Product** 정보를 저장한다.
- 5) Purchase : 구매내역들을 저장한다.
- 6) PurchaseItem : User가 구매한 정보를 저장한다.
- 7) Review : 유저가 리뷰를 작성했을 때 그 데이터를 저장한다.
- 8) Admin : 판매된 **data**들을 저장한다.

1.2.3. Server

- 1) SiteUser 에 관한 모듈들을 정의한다.
- 2) Product 에 관한 모듈들을 정의한다.
- 3) Cart 에 관한 모듈들을 정의한다.
- 4) CartItem 에 관한 모듈들을 정의한다.
- 5) Purchase 에 관한 모듈들을 정의한다.
- 6) PurchaseItem 에 관한 모듈들을 정의한다.
- 7) Review 에 관한 모듈들을 정의한다.
- 8) Admin 에 관한 모듈들을 정의한다.

1.3. 개발 환경

- OS : Window 10
- 개발 도구 : VS code, Git, Docker, DBeaver
- 개발 언어 : Java, Python, HTML, CSS, JavaScript, MySQL
- 개발 기간 : 7일 (2023.12.14 – 2023.12.20)
- 개발 인원 : 4명

2. System 설계

2.1. Modules 개요

- **User Module:** 로그인, 로그아웃, 회원가입 등 유저와 관련된 메소드들을 모아 놓은 모듈
- **Product Module:** 물품과 관련된 메소드를 모아 놓은 모듈
- **Purchase Module:** 유저가 구매한 물품 내역에 관련된 메소드를 모아 놓은 모듈
- **Review Module:** 유저가 작성한 리뷰에 관련된 메소드를 모아 놓은 모듈
- **Cart Module:** 유저가 최종 구매하기 전에 담아 놓는 장바구니에 관련된 메소드를 모아 놓은 모듈
- **Admin Module:** 5일간 매출과 월간 매출을 차트로 나타내는 기능과 판매량 예측에 관련된 메소드를 모아놓은 모듈

2.2. Modules 상세 설계

Module Name			
Function Name	Input	Output	Explanation

User Module			
IUserRepository()	-	-	User Entity에 대한 CRUD 작업
SiteUser()	-	-	User Entity 정의
UserController()	user_info	frontpage	User MVC Controller
UserCreateForm()	-	-	User 등록 DB 정의
UserDao()	user_info	user_info	User 정보 DB 조작 기능 제공
UserRole()	-	-	User의 역할 정의 (권한 부여)
UserSecurityService()	user_info	user_info	User 정보 로드 후 인증 / 권한 승인
UserService()	user_info	user_info	User 관리 Dao를 통해 기능 제공

user_info			
id	user_address	user_birth	user_id
user_name	user_phone_number	user_pw	

Product Module			
IProductRepository()	-	-	Product Entity에 대한 CRUD 작업
Product()	-	-	Product Entity 정의, review cart과 관계 설정
ProductController()	product_id	frontpage	Product MVC Controller
ProductDao()	product_info	product_info	Product 정보 DB 조작 기능 제공
ProductService()	product_info	product_info	Product 관리 Dao를 통해 기능 제공

product_info			
product_id	product_category	product_content	product_image
product_name	product_price	product_rating	

Purchase Module			
IPurchaseRepository()	-	-	Purchase Entity에 대한 CRUD 작업
IPurchaseItemRepository()	-	-	PurchaseItem Entity에 대한 CRUD 작업
Purchase()	-	-	Purchase Entity 정의, review cart과 관계 설정
PurchaseController()	user_info	frontpage	Purchase MVC Controller
PurchaseDao()	user_info, product_info, purchase_info	purchase_info	Purchase 정보 DB 조작 기능 제공
PurchaseItem()	-	-	PurchaseItem Entity 정의
PurchaseService()	user_info, product_info, purchase_info	purchase_info	Purchase 관리 Dao를 통해 기능 제공

purchase_info			
purchase_item_id	product_id	product_name	product_price
product_total_price	purchase_count	purchase_id	id

Review Module			
IReviewRepository()	-	-	Review Entity에 대한 CRUD 작업
Review()	-	-	Review Entity 정의, review cart과 관계 설정
ReviewController()	review_info	frontpage	Review MVC Controller
ReviewDao()	review_info	review_info	Review 정보 DB 조작 기능 제공
ReviewService()	review_info	review_info	Review 관리 Dao를 통해 기능 제공

review_info			
review_content	review_rating	product_id	id

Admin Module			
get_db()	DB ID	DB	DB 직접 연결
read_root()	-	-	admin 화면 연결
get_sales_data()	DB data	month data	월매출 data 연결
get_day5_data()	DB data	5 day data	5일매출 data 연결
read_products()	DB data	-	물품 정보 리스트 data 연결
train()	sales data	prediction()	3년간의 data로 판매량 예측
prediction()	date	prediction	model_handler를 활용해 예측값 출력

admin_info			
sales_id	date	product_id	quantity

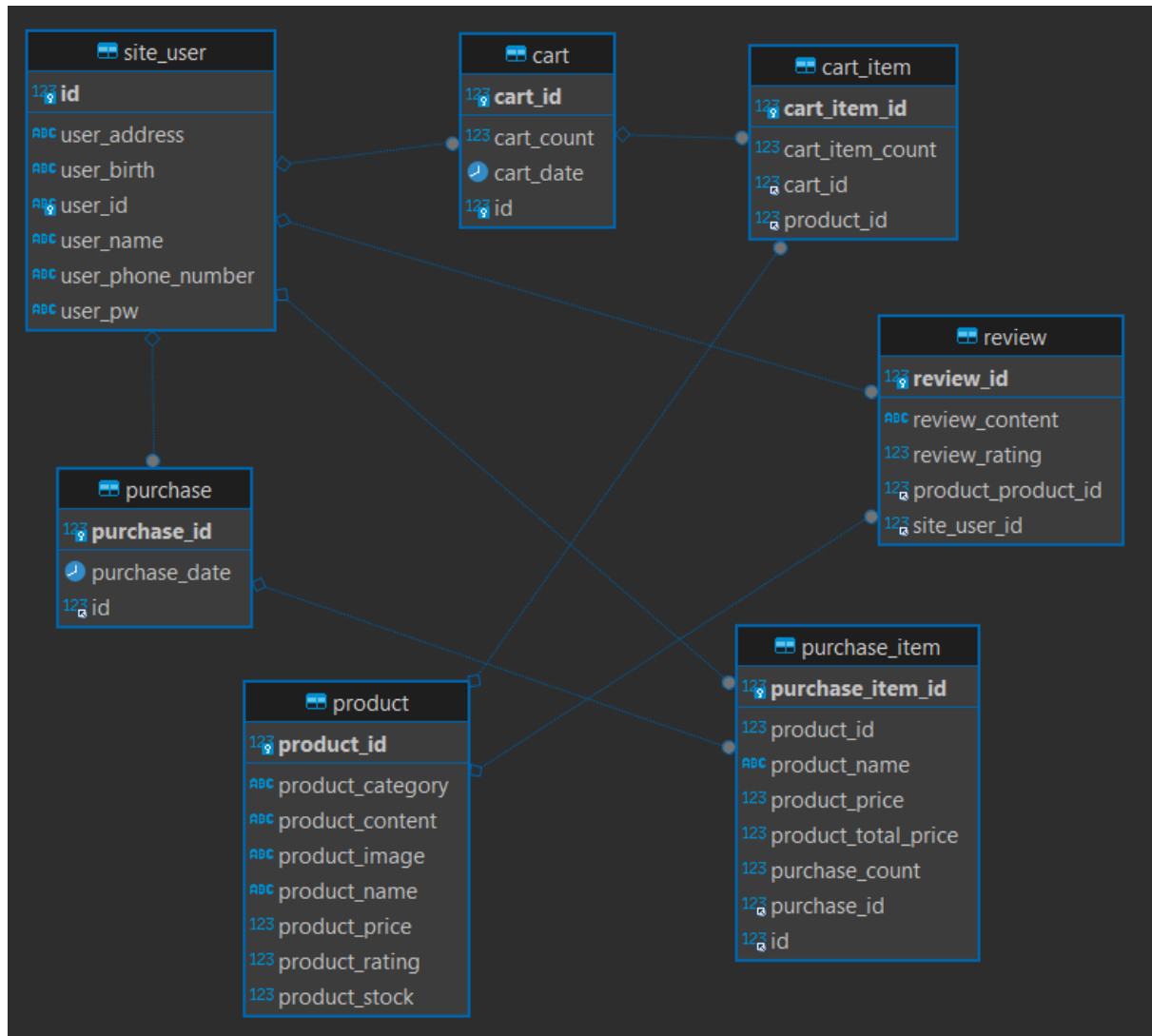
Cart Module			
ICartRepository()	-	-	Cart Entity에 대한 CRUD 작업
ICartItemRepository()	-	-	CartItem Entity에 대한 CRUD 작업
Cart()	-	-	Cart Entity 정의, review cart과 관계 설정
CartItem()	-	-	CartItem Entity 정의
CartController()	user_info, product_info	frontpage	Cart MVC Controller
CartDao()	user_info, product_info, cart_info	cart_info	Cart 정보 DB 조작 기능 제공
CartItemDao()	cart_item_info	cart_item_info	CartItem 정보 DB 조작 기능 제공
CartService()	user_info, product_info, cart_info	cart_info	Cart 관리 Dao를 통해 기능 제공
CartItemService()	cart_item_info	cart_item_info	CartItem 관리 Dao를 통해 기능 제공

cart_info			
cart_count	cart_id	product_id	

cart_item_info			
cart_item_count	cart_date	id	

2.3. Data 설계

2.3.1. ERD 설계 1차 최적화 후 수정 필요 必



[그림 1] 1차 ERD

2.3.2. Entity 설계

(1) site_user

논리명	컬럼명	Datatype	Nullable	설명
ID	id	int	Not NULL	user 고유의 키 값
주소	user_address	varchar	NULL	user 주소
생일	user_birh	varchar	NULL	user 생일
user_id	user_id	varchar	Not NULL	user login id
이름	user_name	varchar	NULL	user 이름
휴대폰	user_phone	varchar	NULL	user phone number
비밀번호	user_pw	varchar	Not NULL	user login pw

(2) product

논리명	컬럼명	Datatype	Nullable	설명
물품ID	product_id	int	Not NULL	product 고유의 키 값
물품 분류	product_category	varchar	NULL	product 물품 분류
물품 내용	product_content	varchar	NULL	product
물품 사진	product_image	varchar	Not NULL	product 사진 링크
물품 이름	product_name	varchar	NULL	product 이름
물품 가격	product_price	varchar	NULL	product 가격
물품 순위	product_rating	varchar	Not NULL	product 순위
물품 재고	product_stock	varchar	Not NULL	product 재고

(3) purchase

논리명	컬럼명	Datatype	Nullable	설명
구매 id	purchase_id	int	Not NULL	purchase 고유의 키 값
구매 날짜	purchase_date	datetime	Not NULL	product 구매 날짜
user id	id	int	Not NULL	유저 id

(4) purchase_item

논리명	컬럼명	Datatype	Nullable	설명
구매 물품 id	purchase_item_id	int	Not NULL	purchase_item 고유의 키 값
물품 id	product_id	datetime	Not NULL	product id
물품 이름	product_name	int	Not NULL	product 이름
물품 가격	product_price	int	Not NULL	product 가격
물품 총 가격	product_total_price	int	Not NULL	product 총 가격
구매량	purchase_count	int	Not NULL	구매량
구매 id	purchase_id	int	Not NULL	구매 id
user id	id	int	Not NULL	유저 id

(5) review

논리명	컬럼명	Datatype	Nullable	설명
리뷰 id	review_id	int	Not NULL	review 고유의 키 값
리뷰 내용	review_content	varchar	Not NULL	review 구매 날짜
리뷰 순위	review_rating	int	Not NULL	review 순위
물품 id	product_id	int	Not NULL	물품 id
user id	id	int	Not NULL	유저 id

(6) admin

논리명	컬럼명	Datatype	Nullable	설명
판매 id	sales_id	int	Not NULL	sales 고유의 키 값
판매 날짜	date	date	Not NULL	sales 판매 날짜
물품 id	product_id	int	Not NULL	물품 id
판매량	quantity	int	Not NULL	판매량

(7) cart

논리명	컬럼명	Datatype	Nullable	설명
장바구니 ID	cart_id	int	Not NULL	cart 고유의 키 값
장바구니 담은 갯수	cart_count	int	Not NULL	cart 담은 갯수
장바구니 담은 날짜	cart_date	datetime	Not NULL	cart 담은 날짜
user ID	id	int	NULL	유저 id

(8) cart_item

논리명	컬럼명	Datatype	Nullable	설명
장바구니 품목 ID	cart_item_id	int	Not NULL	cart_item 고유의 키 값
장바구니 품목 갯수	cart_item_count	int	Not NULL	cart_item 갯수
장바구니 ID	cart_id	int	NULL	장바구니 id
물품 ID	product_id	int	NULL	물품 id

2.4. 판매량 예측 Model 설계

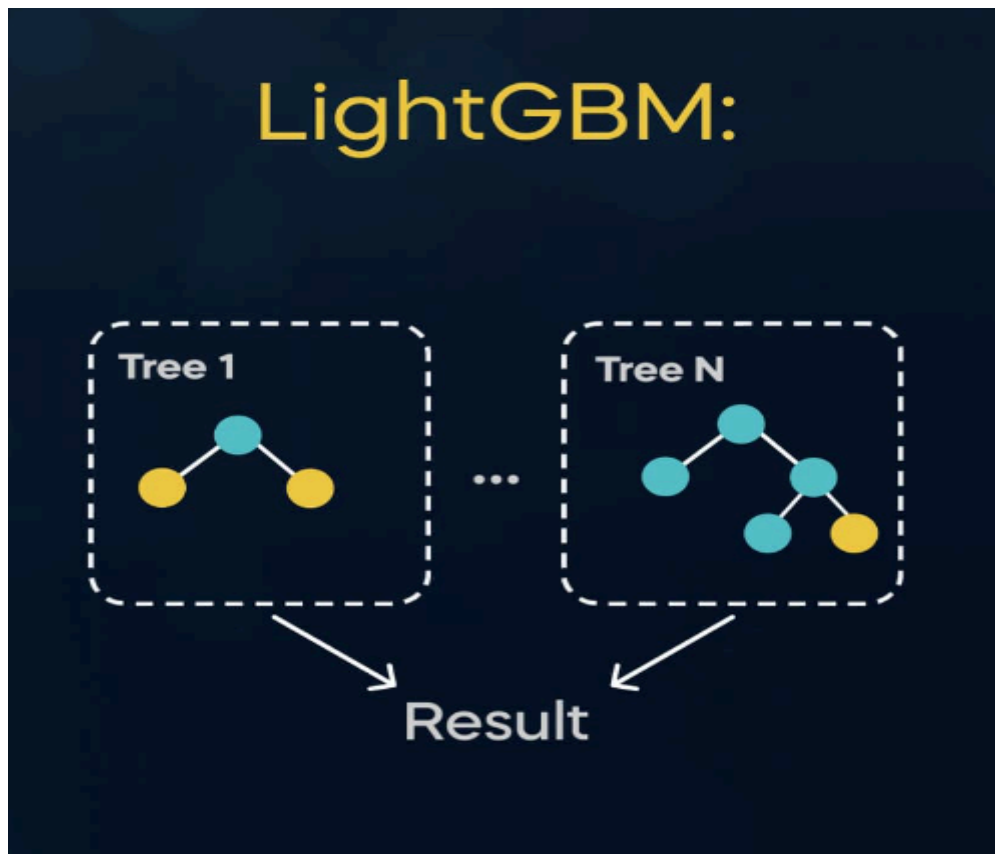
2.4.1. LGMRegressor 사용

▷ 필요한 라이브러리

- pandas
- lightgbm
- scikit-learn

▷ LGMRegressor Model 개요

LGMRegressor는 기계학습에서 사용되는 회귀 모델 중 하나로 LGM(Light Gradient Boosting Machine)은 고성능 그래디언트 부스팅 프레임워크이며 대용량 데이터셋과 대규모 모델 훈련에 효율적이다.



[그림 2] LightGBM

여러개의 결정 트리를 조합하여 강력한 예측 모델을 만드는 앙상블 학습 방법이며 균형 트리 분할로 균형 잡힌 트리가 아닌 리프 노드를 확장하는 방식으로 구성되어 속도를 높였다.

범주형 특징을 최적으로 다루기 위한 효과적인 방법을 제공하며 병렬 처리를 통해 대용량 데이터 셋에 적합하다. 또한 메모리 사용량이 상대적으로 적기 때문에 비 기능 요구사항에 효율적으로 적용시킬 수 있다.

LGMRegressor는 다양한 하이퍼파라미터를 제공하며 그 중 트리의 깊이, 학습률, 트리 수 등이 대표적인 튜닝 대상이 된다.

2.4.2. Cosine Similarity 사용 물품 추천 시스템

▷ 필요한 라이브러리

- pandas
- scikit-learn

▷ Cosine Similarity Model 개요

Cosine Similarity는 추천 모델은 주로 텍스트나 벡터 데이터에 적용되며 사용자-아이템 간의 유사성을 측정하여 추천을 제공하는 모델이다. 이 때 유사성을 측정할 때 **Cosine**을 활용한다.

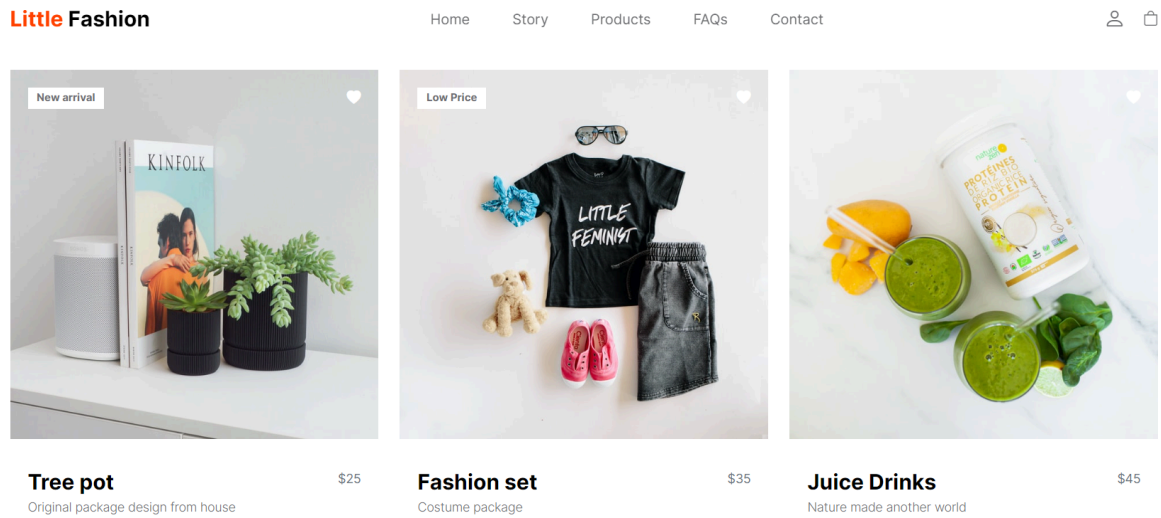
사용자나 아이템을 벡터로 표현해 선호도나 아이템의 특성을 나타낸다. 유사성이 높은 아이템은 추천 목록에 포함시켜 사용자에게 제공한다.

Cosine Similarity는 각도를 기반으로 유사성을 측정하기에 값은 -1에서 1 사이이다. 1에 가까울 수록 유사성이 높다고 판단하고 -1에 가까울 수록 유사성이 낮다고 판단한다.

이 모델은 간단하고 이해하기 쉽고 벡터로 표현하기 쉽다는 장점이 있지만 단점으로는 새로 등장한 아이템에 대한 추천이 어렵고 개인적인 취향을 반영하기 어려울 수 있다.

3. UI

3.1. 메인 화면

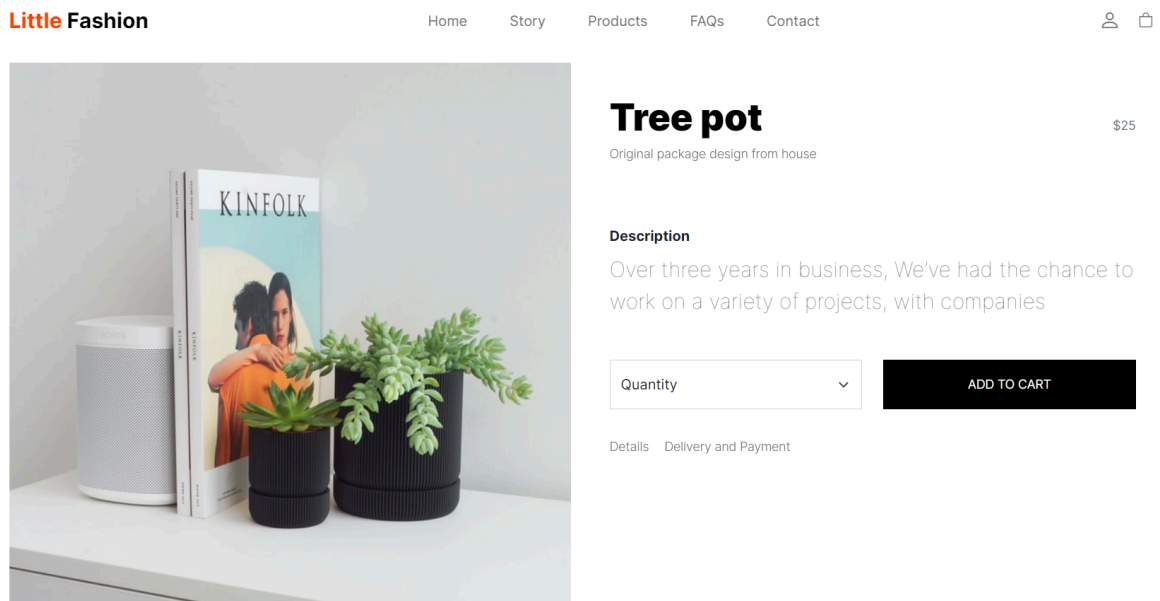


[그림 3] 메인 화면 (참고 템플릿)

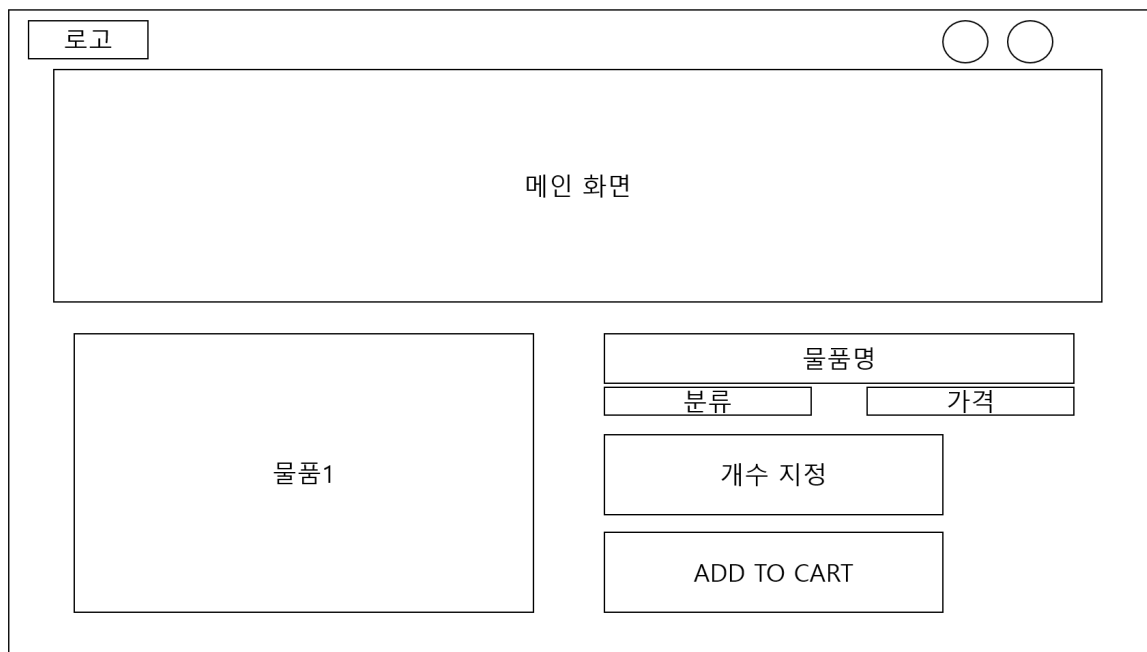


[그림 4] 메인 화면 (구현 디자인)

3.2. 물품 상세 화면



[그림 5] 물품 상세 화면 (참고 템플릿)

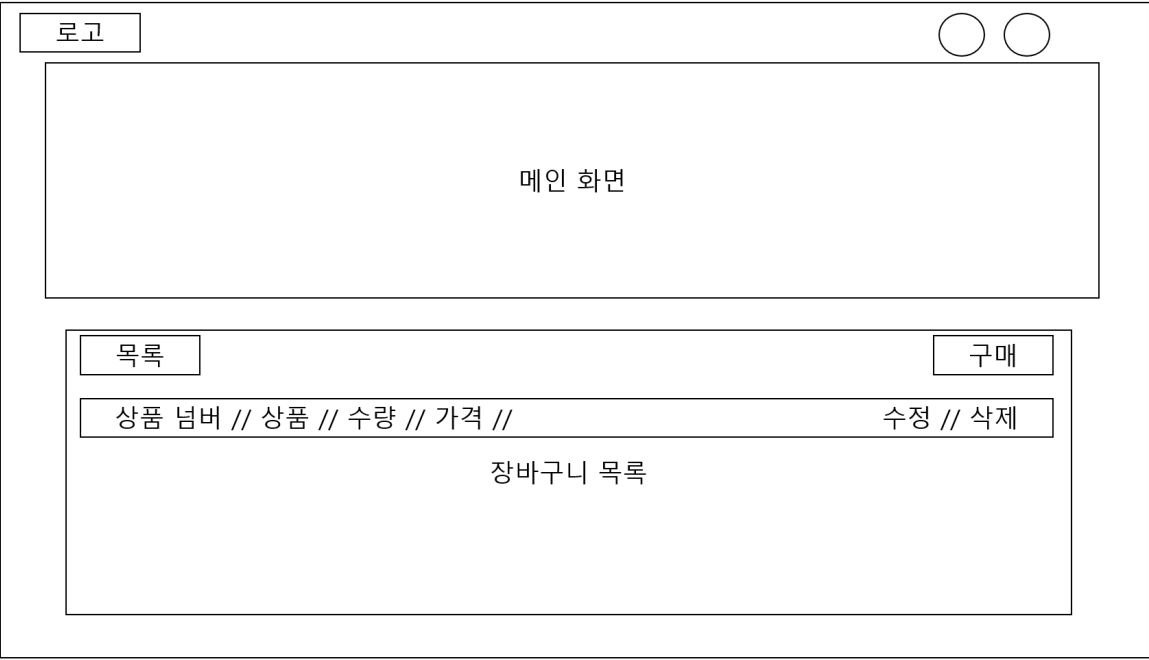


[그림 6] 물품 상세 화면 (구현 디자인)

3.3. 장바구니 화면



[그림 7] 장바구니 화면 (참고 템플릿)

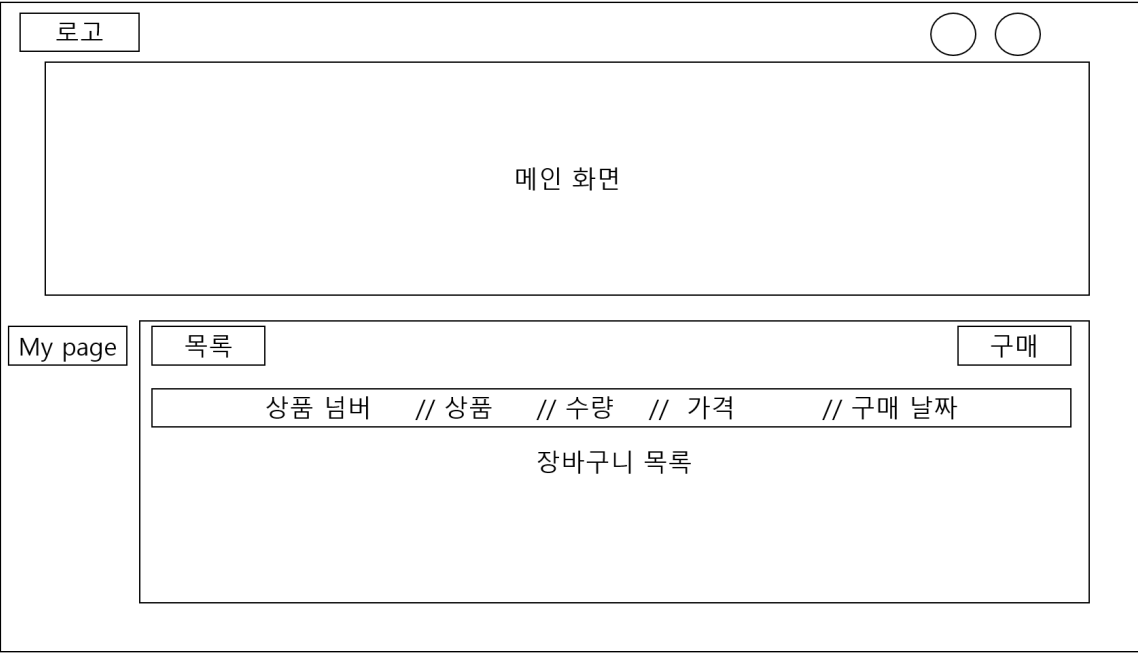


[그림 8] 장바구니 화면 (구현 디자인)

3.4. 마이페이지 화면

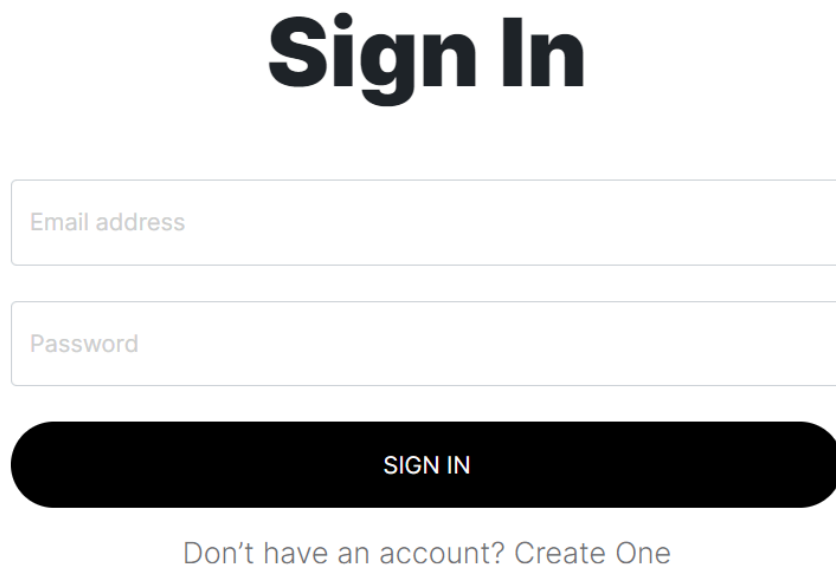


[그림 9] 마이페이지 화면 (참고 템플릿)



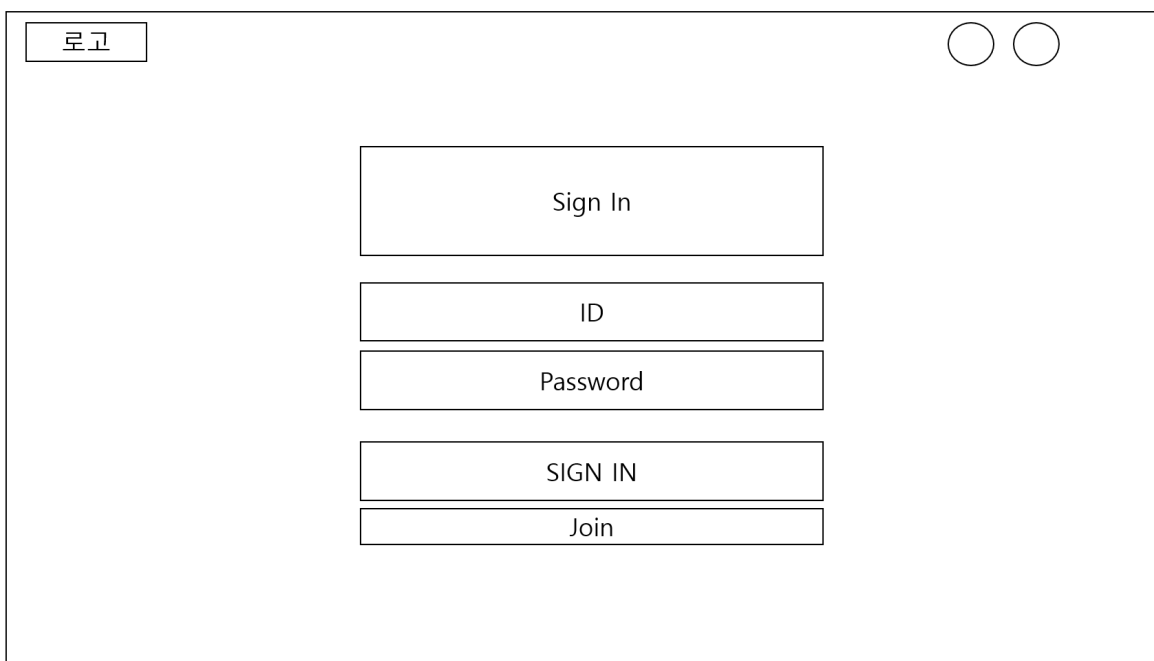
[그림 10] 마이페이지 화면 (구현 디자인)

3.5. 로그인 화면



A minimalist sign-in form template. At the top, the text "Sign In" is displayed in a large, bold, black font. Below it are two white input fields with thin grey borders. The first field is labeled "Email address" in a light grey font, and the second is labeled "Password". Below these fields is a prominent black button with rounded corners, containing the text "SIGN IN" in white, uppercase letters. At the bottom, there is a link that reads "Don't have an account? Create One" in a small, light grey font.

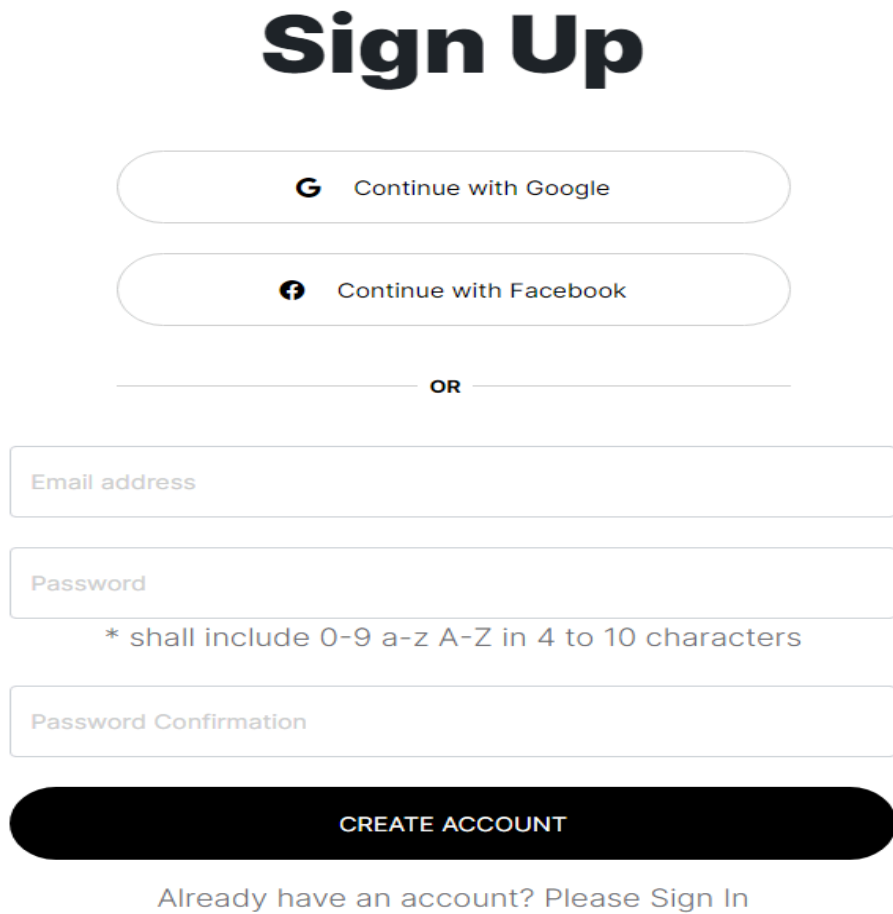
[그림 11] 로그인 화면 (참고 템플릿)




An implemented login screen design within a window frame. The window has a title bar with a "로고" (Logo) button on the left and two circular window control buttons on the right. The main content area is centered and contains a vertical stack of elements: a "Sign In" button, an "ID" input field, a "Password" input field, another "SIGN IN" button, and a "Join" button. All text and buttons are rendered in a clean, sans-serif font.


[그림 12] 로그인화면 (구현 디자인)

3.6. 회원가입 화면



Sign Up

 Continue with Google

 Continue with Facebook

OR

Email address

Password

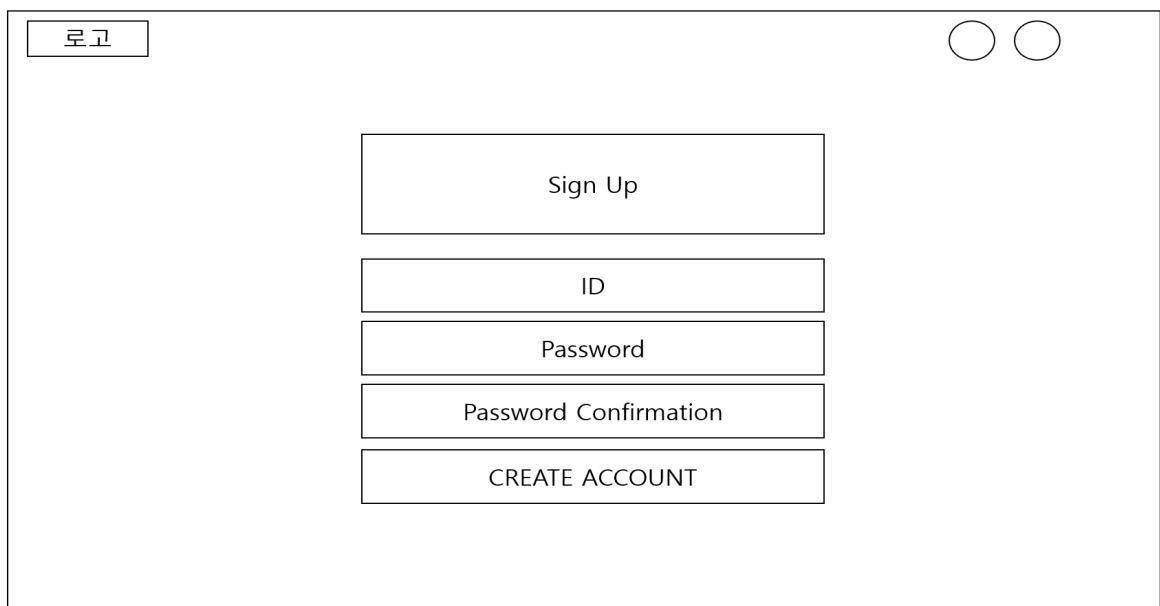
* shall include 0-9 a-z A-Z in 4 to 10 characters

Password Confirmation

CREATE ACCOUNT

Already have an account? Please Sign In

[그림 13] 회원가입 화면 (참고 템플릿)



로고

Sign Up

ID

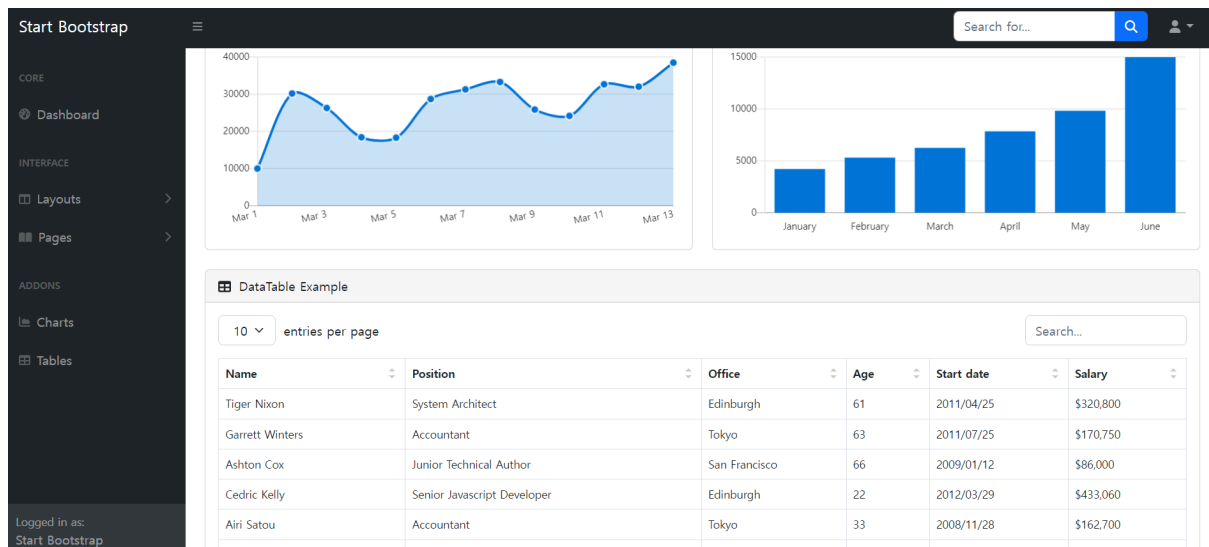
Password

Password Confirmation

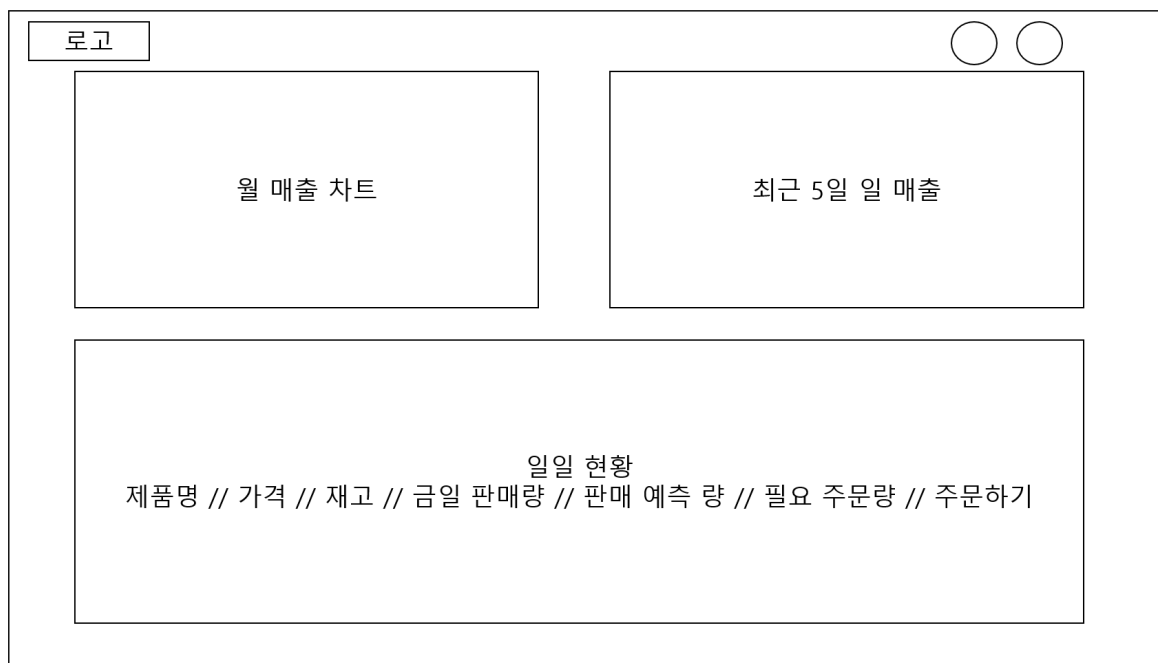
CREATE ACCOUNT

[그림 14] 회원가입 화면 (구현 디자인)

3.7. 관리자 화면



[그림 15] 관리자 화면 (참고 템플릿)



[그림 16] 관리자 화면 (구현 디자인)

4. WBS

Task		담당자	작업 산출물	W1							
				D1(12월 14일)	D2(12월 15일)	D3(12월 16일)	D4(12월 17일)	D5(12월 18일)	D6(12월 19일)	D7(12월 20일)	D8(12월 21일)
1. 계획	1.1. 프로젝트 주제 선정										
	1.2. front 템플릿 선정										
2. 분석 및 설계	2.1. 요구사항 정의		요구사항정의서								
	2.2. 개발환경 및 사용기술 정의		기능정의서								
	2.3. DB 설계		DBTable 정의서								
3. 구현	3.1. View - 메인(품목리스트)		화면정의서								
	3.1. View - 품목상세										
	3.1. View - 장바구니										
	3.1. View - 마이페이지										
	3.1. View - 로그인										
	3.1. View - 회원가입										
	3.1. View - 관리자										
	3.3. 품목리스트 조회		기능정의서								
	3.4. 품목 상세 페이지 조회										
	3.5. 장바구니 조회										
	3.6. 찜 목록 조회										
	3.7. 댓글 목록 조회										
	3.2. 로그인/로그아웃 기능										
	3.1. 회원가입 기능										
	3.8. 관리자 페이지 - 통계										
	3.9. 관리자 페이지 - 예측모델										
	3.10. 예측모델 결과 응답 받는 api 기능										
4. 테스트											
4.1. 서비스 통합 테스트											
5. 안정화											
5.1. 안정화											
6. 발표											
6.1. 발표자료											
6.1. 발표			결과보고서								