



반응형 웹/앱 가구 쇼핑몰 사이트

조원 : 정성윤, 우수정, 임세훈, 김수훈

ILOKE

CONTENTS



O1 기획

O2 벤치마킹

O3 마인드 맵

O4 GUI

O5 작업 툴

O6 요구사항 정의서

O7 시스템 구조도

O8 Use Case

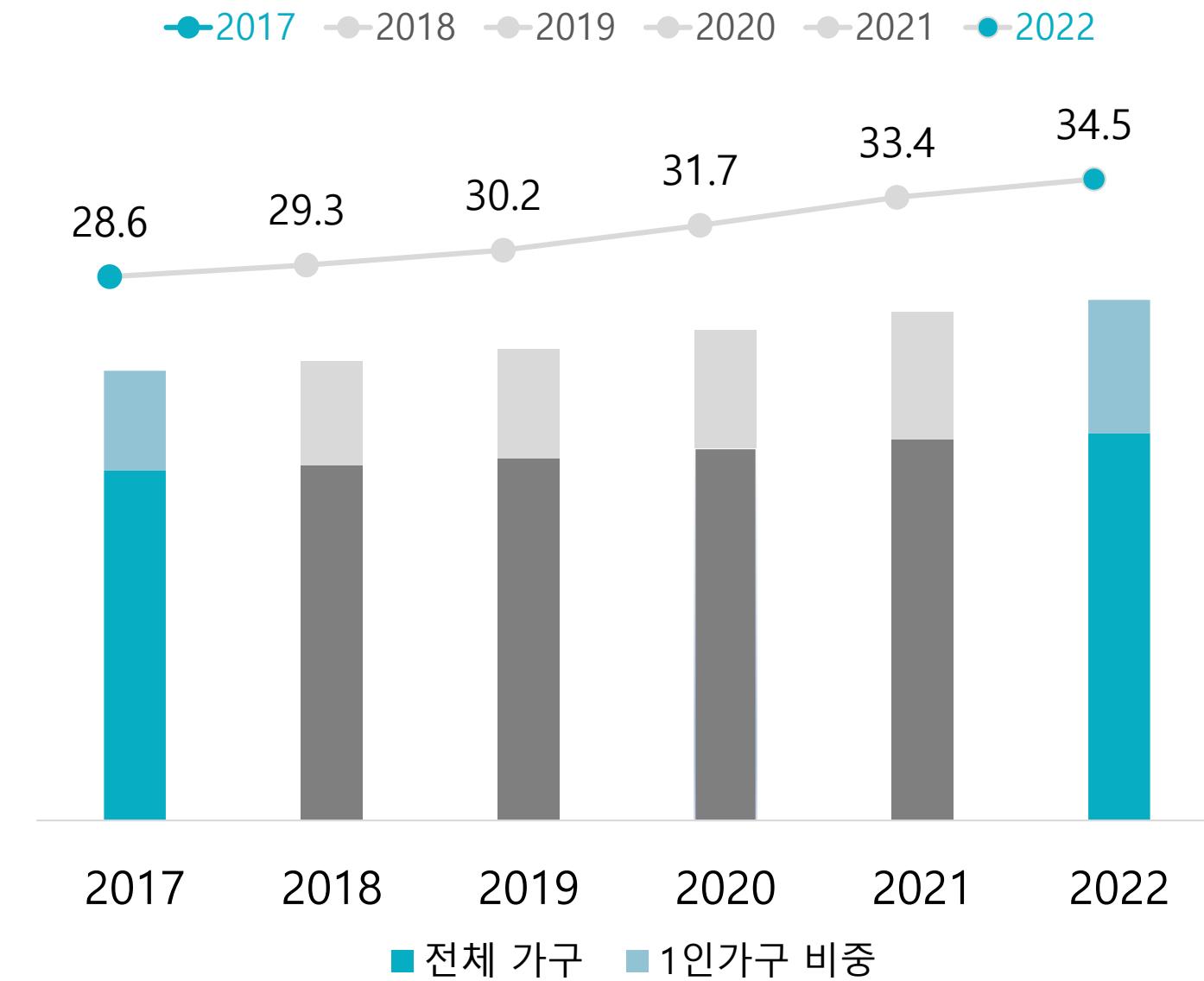
O9 ERD

10 역할 분담

11 시간 계획서

기획

1인 가구 추이

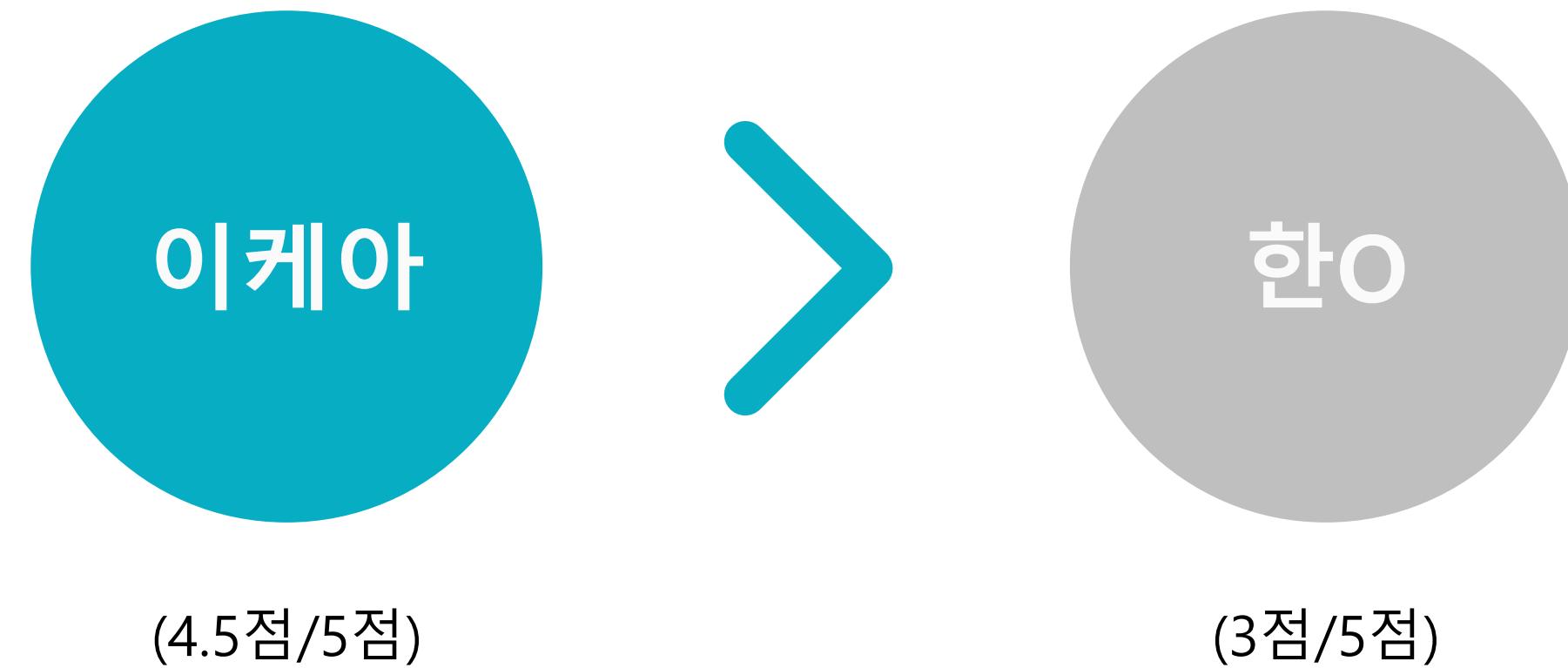


단위: 가구, % / 자료 = 통계청 (2023.12.12) <https://www.chosun.com/>

1인 가구가 증가하는 시대에 1인 가구를 위한 상품에 대한 접근성이
중요

기획

2030 남녀 가격 대비 만족도 조사

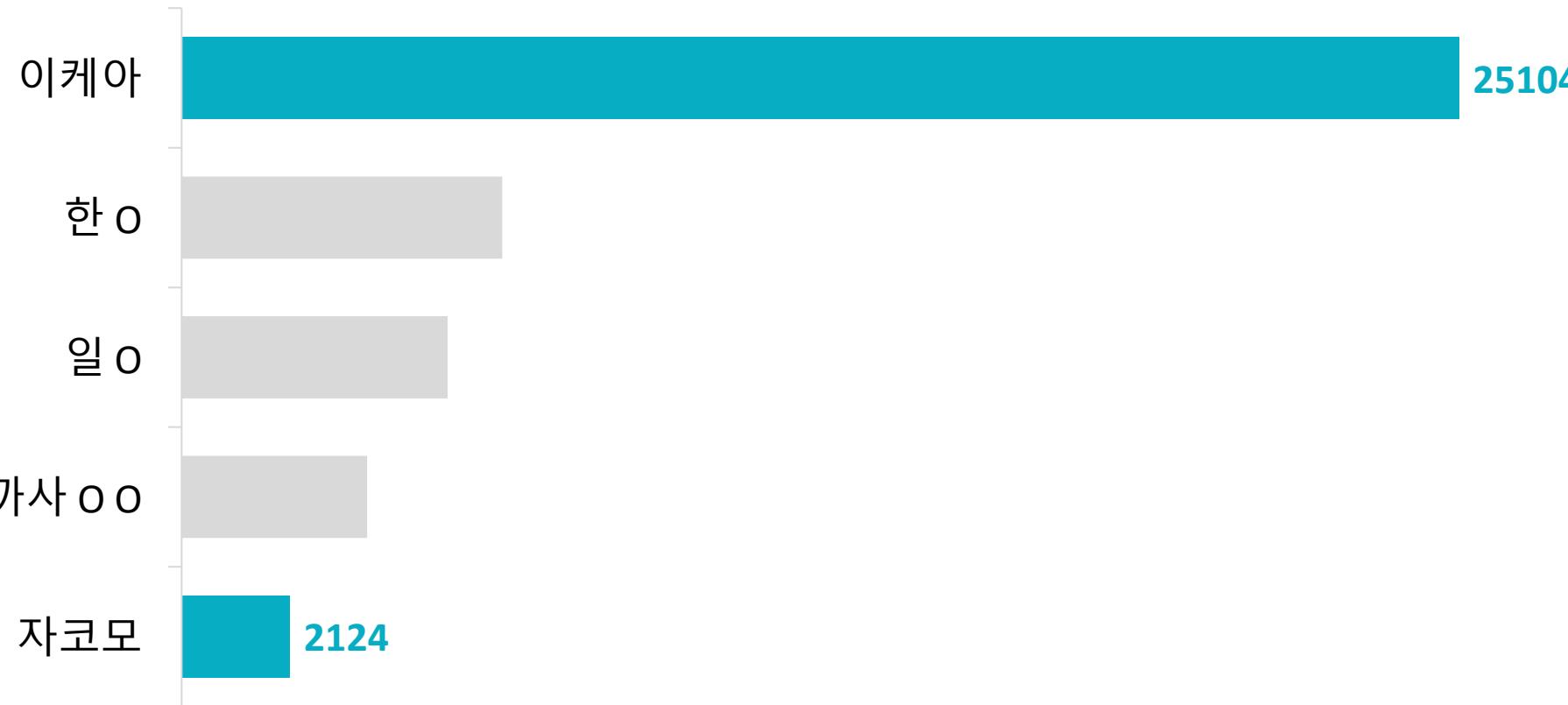


출처 : 데일리팝 ([2016.12.09](#))

1인 가구 비율이 높고 혼자 사용하기 가성비 좋은 제품 찾기
때문

기획

랭키파이 브랜드 가구 순위 차트



출처 : 랭키파이 (2024.04.24)

키워드 검색량과 구글 트렌드 점수를 이용해 랭키파이에서 산출한

트렌드 지수 **이케아 1위**

기획

고객 인지도 1위 '자코모', 베스트 소파 브랜드 대상

▣ 오한준 기자 | ○ 승인 2024.04.18 17:10 | ▴ 댓글 0



사진=자코모 제공

소파 전문 브랜드 자코모가 '2024 대한민국 대표 브랜드 대상'에서 소파브랜드 부문 대상을 수상하였다. 자코모는 모든 문항에서 압도적인 1위를 달리며 명실상부 국내 대표 소파 브랜드임을 입증했다.

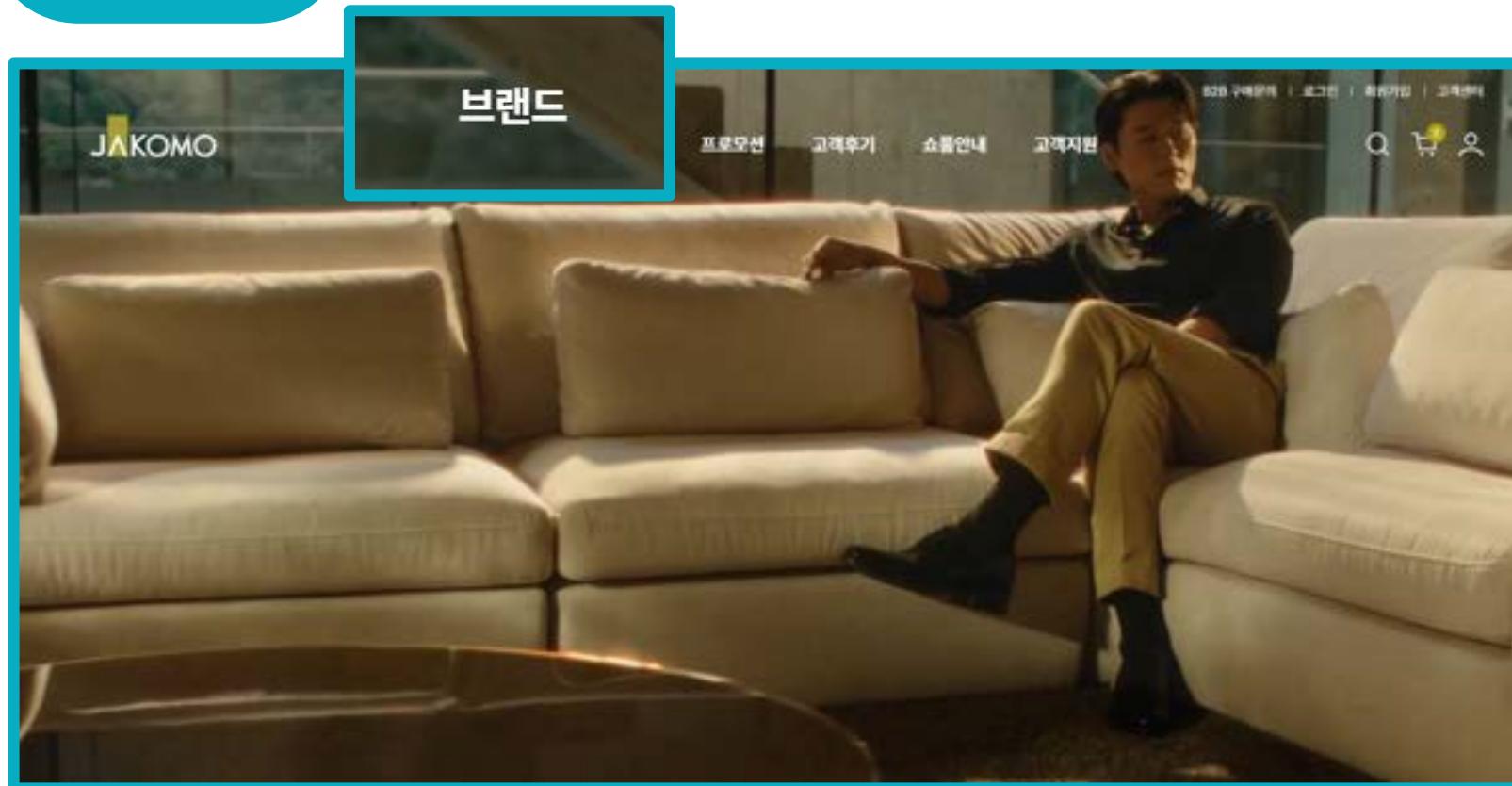
출처 : 데일리경제 (2024.04.18) <http://www.kdpress.co.kr/>

2024 대한민국 대표 베스트 소파 브랜드 대상

고객 인지도 **자코모 1위**

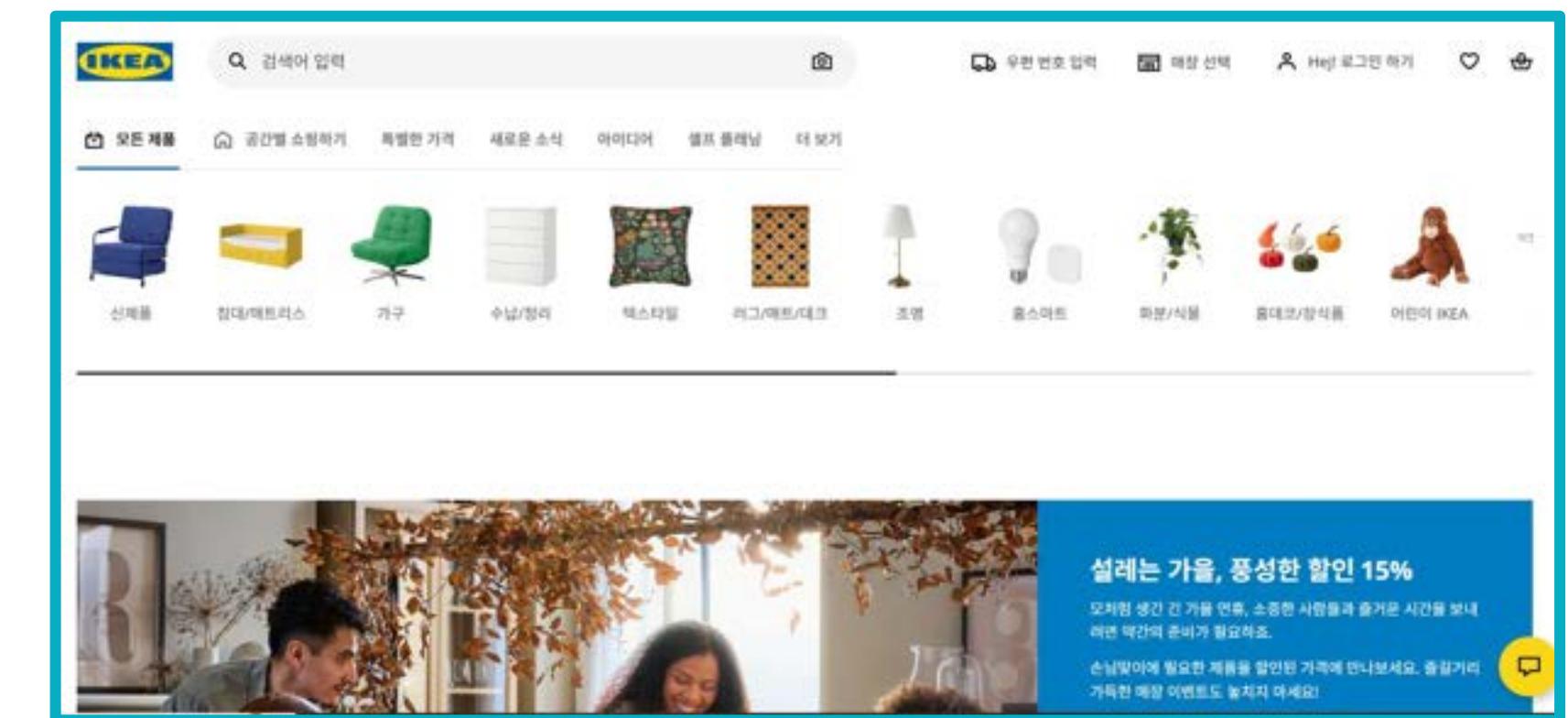
기획

자코모



브랜딩을 위한 큰 사진과 첫 카테고리에 [브랜드](#) 나열
출처 : 자코모 홈페이지 <https://jakomo.co.kr/main/index.php>

이케아



상품 위주 나열이며 브랜드 관련 내용은 좌측 상단 로고 배치가 전부
출처 : 이케아 홈페이지 <https://www.ikea.com/kr/ko/>

브랜드 친화적 메인 화면 레이아웃 구성을 가진 자코모

벤치마킹



장점

We Make Quality
Jakomo Since 1986

40여 년의 길이가 만든 불비른 제품 철학과
사랑 깊은 장인의 정성을 담아 다르게, 더 바르게 만듭니다.

[View More](#)

출처 : 자코모 홈페이지 <https://jakomo.co.kr/main/index.php>

세련된 디자인으로 품격 높은 브랜드 이미지
표현

기획전

JAKOMO in DRAMA

-5%

자코모 in 드라마
드라마 속 자코모 소파

-5%

편안한 가실 속 미니멀라이프
간편하면서도 세련된 사이즈 소파 추천

-5%

최적의 휴식 할길 소파
작중을 흐리울 위한 다양한 기능성 소파

-10%

공식몰 단독 소파 할인전
보석급 가죽재와 사연을 수 있는 할인

나눔에 대한

인재를 양성합니다

재능을 후원합니다

생각을 함께합니다



당신의 성장과 더불어 삶을 공유합니다

나눔에 대한

인재를 양성합니다

재능을 후원합니다

생각을 함께합니다

행복을 환원합니다

기획전과 이벤트가 다양하며, 시각적으로
화려

벤치마킹



단점

스툴 사이즈



*사이즈는 측정방식에 따라, 수작업 제작 특성상,
다소 차이가 있을 수 있습니다.



비기닝 모듈 1인 한팔(우형) 천연면피 소가죽 소파

900,000원



출처 : 자코모 홈페이지 <https://jakomo.co.kr/main/index.php>

제품 상세 정보 부족
상품 비용이 높아 구매에 부담

벤치마킹



장점

더 낮은 새로운 가격

SEGERÖN 세게뢴
야외커피테이블, 화이트/베이지, 73x73 cm

₩119,000

기존가: ₩139,000

★★★★★ (0)

치수

높이: 30 cm
길이: 73 cm
폭: 73 cm



SEGERÖN 세게뢴
야외커피테이블

제품 번호
905.111.72

폭: 44 cm
높이: 7 cm
길이: 73 cm
무게: 10.69 kg
패키지: 1

출처 : 이케아 홈페이지 <https://www.ikea.com/kr/ko/>

정보 제공 풍부
이동, 재배치가 간단한 조립식 가구
위주

벤치마킹



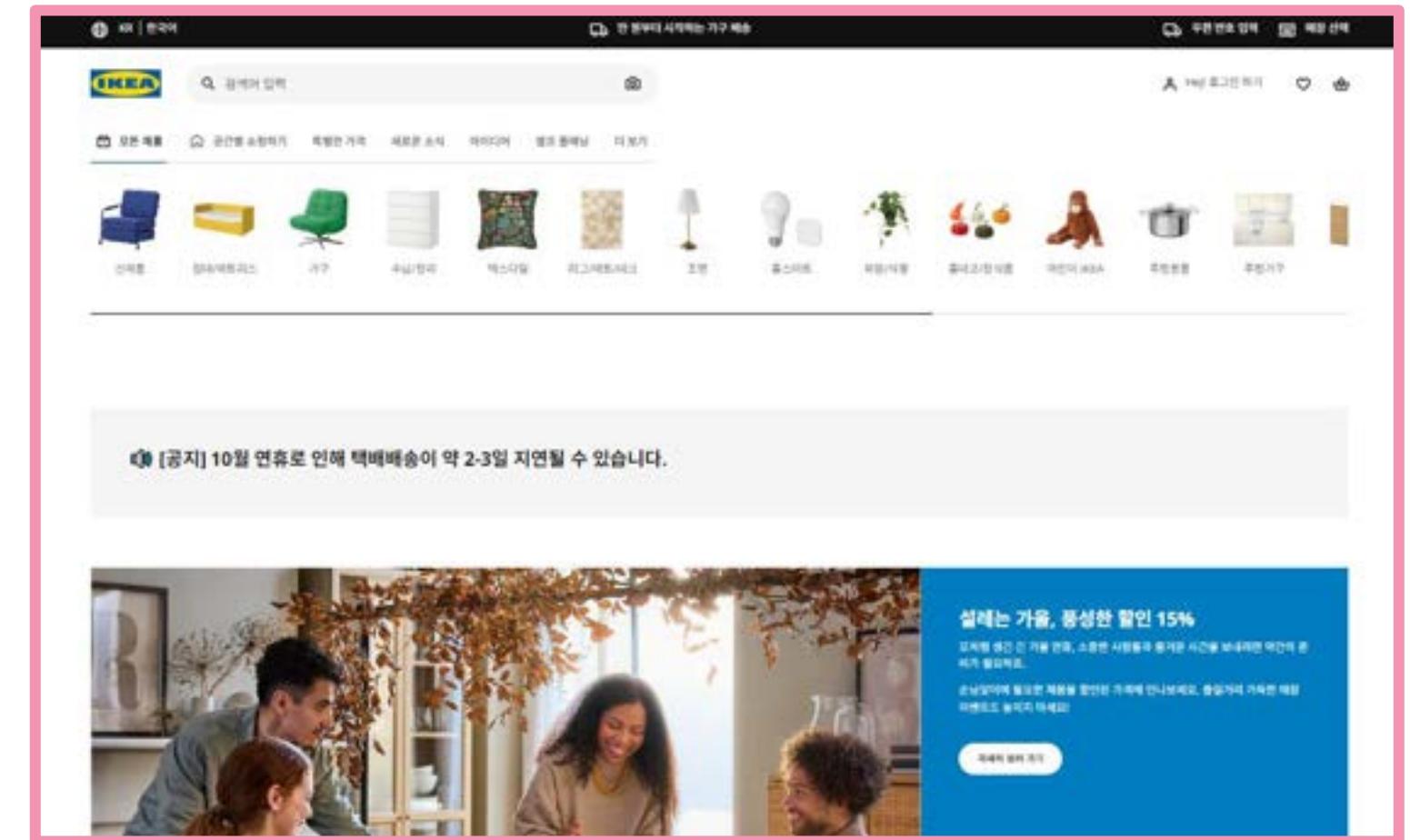
단점

출처 : 이케아 홈페이지 <https://www.ikea.com/kr/ko/>

IKEA Family	고객문의	쇼핑하기	서비스	IKEA 이야기
지금 IKEA Family에 무료로 가입하고 다양한 멤버 전용 혜택을 누리세요. 자세히 보기	고객 서비스 고객 문의 질문 고객지원센터 배송조회 교환환불 품질보증 제품리콜 피드백 부품 신청	쇼핑하기 차주 묻는 질문 고객지원센터 배송조회 교환환불 제품 사용 팁 / 가이드 브로슈어 / 제품 구매 안내 결제 옵션 기프트 카드	IKEA 서비스 전화 주문 IKEA for Business 셀프 플래닝 IKEA 모바일 앱	브랜드 소개 집에서의 생활 지속가능한 생활 뉴스레터 제품정보 구매 상담 서비스 인테리어 디자인 서비스 바이백 서비스

IKEA Business Network

여러분의 더 나은 비즈니스 환경을 위한 다양한 혜택들을 받으세요. 자세히 보기	IKEA Family 가입하기	IKEA Business Network 가입하기
--	----------------------------------	--



The screenshot shows the IKEA Korea website's main page. At the top, there's a navigation bar with links for '모든 제품' (All Products), '공간별 솔루션', '직영점 가격', '새로운 소식', '마리디어', '결제 플랫폼', and '더 보기'. Below the navigation is a grid of icons representing various product categories: 신작품 (New products), 청소/백화점 (Cleaning/Hypermarket), 가구 (Furniture), 주님/침실 (Bedroom), 책스다일 (Books), 리그메트리너 (Liggett), 조명 (Lighting), 풀스티커 (Stickers), 화분/식물 (Plants/flowers), 퍼포먼스 (Performance), 아동/인테리어 (Children/Interior), and 주방가구 (Kitchen). A central banner states '[공지] 10월 연휴로 인해 배송이 약 2-3일 지연될 수 있습니다.' (Notice: Due to the October holidays, delivery may be delayed by about 2-3 days). Below the banner is a photo of four people in a living room setting. To the right of the photo is a blue promotional box for '설레는 가을, 풍성한 할인 15%' (Exciting autumn, 15% discount) with fine print about delivery times.

이벤트 다소 부족
시각적으로 화려하지 않아 사이트 자체의 매력은 없음
기업 브랜딩이 약함

벤치마킹 - 결론

JAKOMO, IKEA의 UI/UX 디자인, 상품 정보 제공 방식, 마케팅 및 프로모션을 바탕으로 벤치마킹

○1. 브랜딩

업체와 제품의 이미지 및 설명이 얼마나 매력적인지 평가

○2. 가시성

레이아웃과 네비게이션이 직관적인지 평가

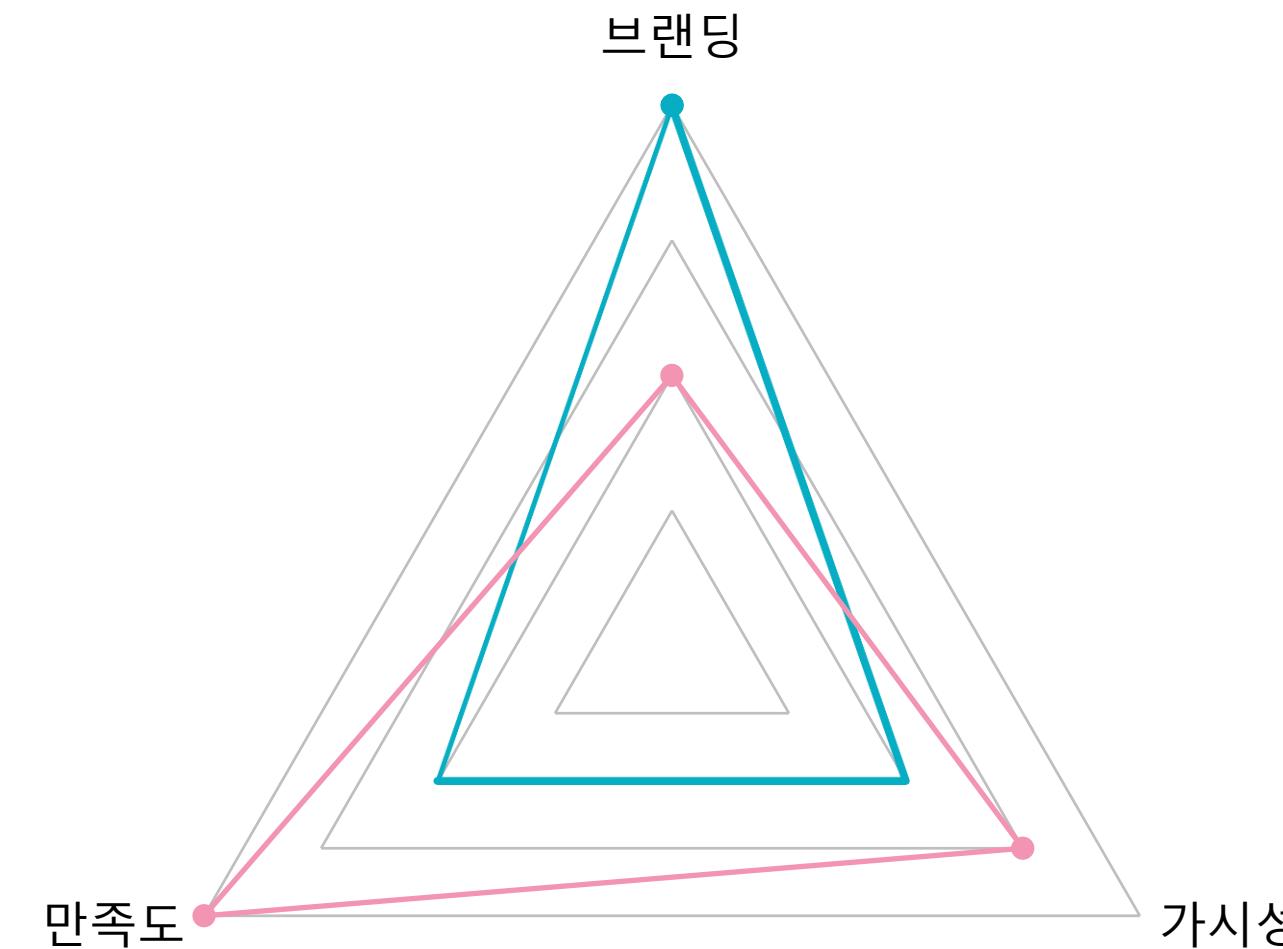
○3. 만족도

판매중인 상품이 편리한지 평가

벤치마킹 - 결론

벤치마킹 웹 사이트 분석 그래프

■ 자코모 ● 이케아

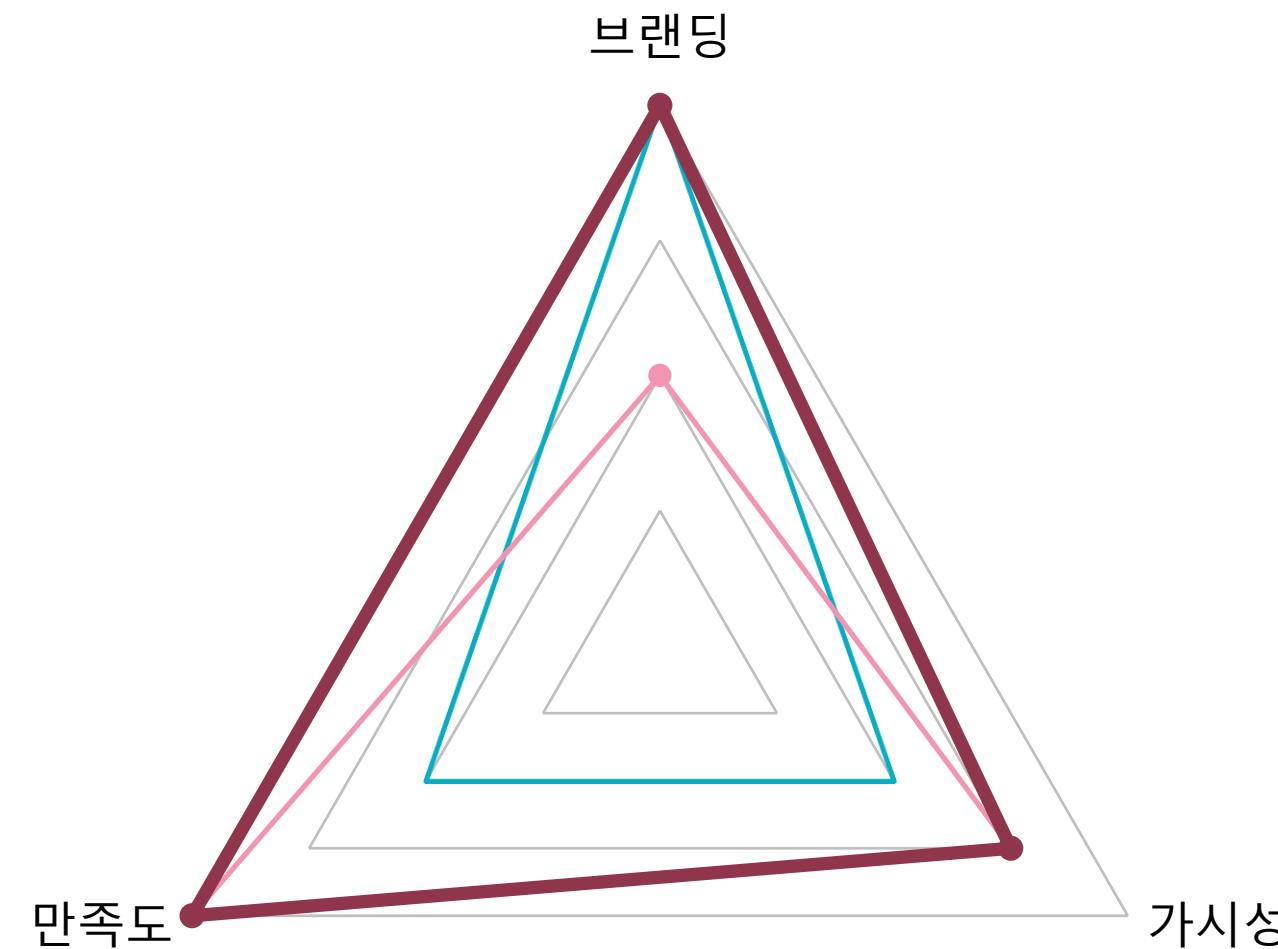


자코모의 세련된 디자인과 브랜딩
+
이케아의 편리하고 풍부한 상품 정보 제공 방식

벤치마킹 - 결론

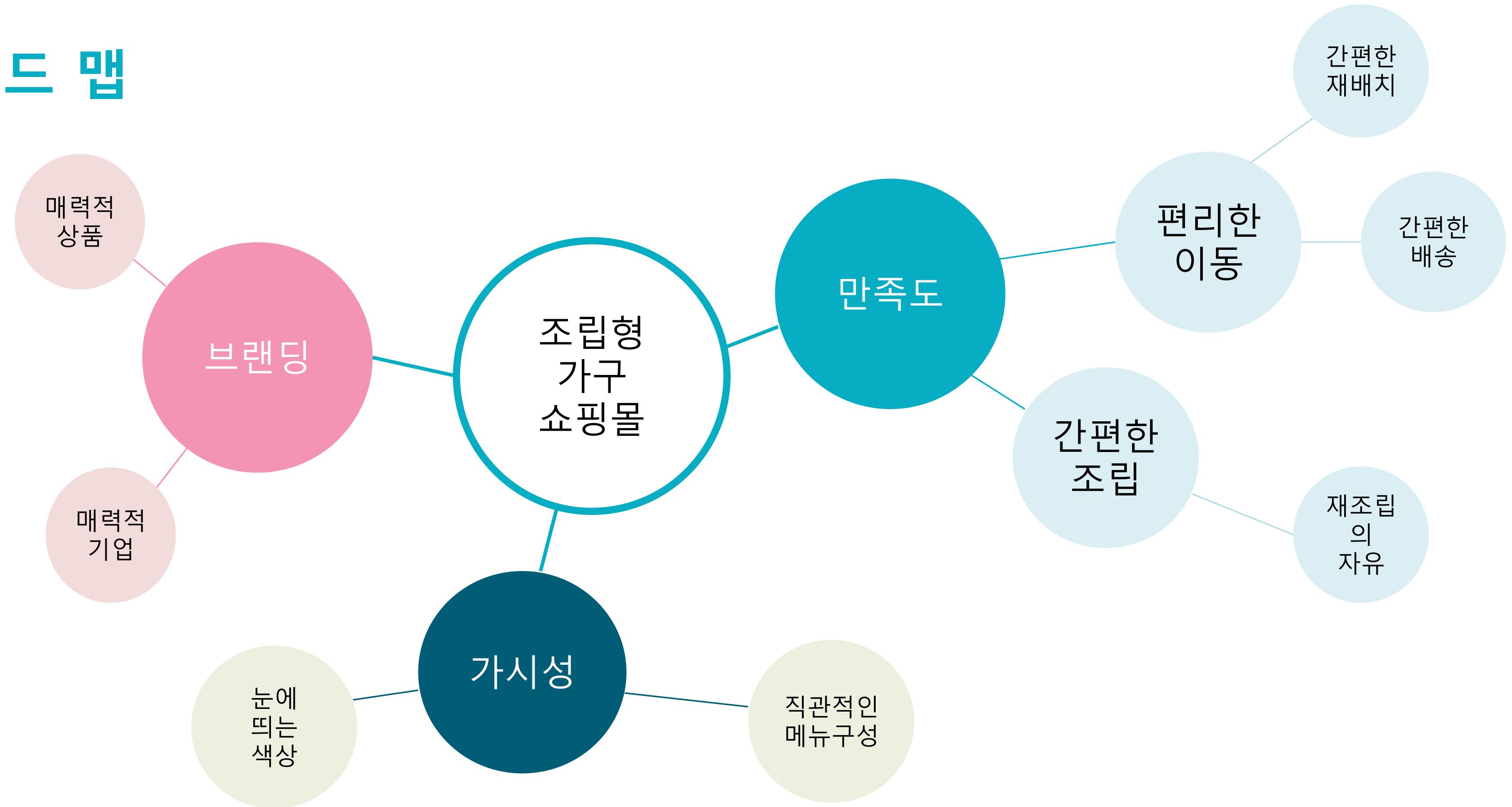
웹 사이트 방향성

● 자코모 ● 이케아 ● 이로케



세련된 디자인과 브랜딩,
편리하고 풍부한 상품 정보 제공 방식을 가진
1인 가구를 위한 가구 쇼핑몰 사이트

마인드 맵



GUI - 색상선정



메인 포인트 색상
분홍 #F294B2
초록 #06ADC3



서브 색상
진분홍 #8F364D
진초록 #005D75



주요 타깃 연령층인 2030의 젊고 활기찬 이미지를 반영하여
보색 대비를 통해 밝고 에너지 있는 느낌을 강조,
균형을 위해 가볍거나 부담스럽지 않은 부드러운 분홍-민트 색조 사용

GUI – 폰트선정

한글 - Noto Sans Kr

가나다라마바사아자차카타파하

가나다라마바사아자차카타파하

가나다라마바사아자차카타파하

가나다라마바사아자차카타파하

가나다라마바사아자차카타파하

영어 – Roboto

AbCdEfGhIjKlMnOpQrStUvWxYz

AbCdEfGhIjKlMnOpQrStUvWxYz

AbCdEfGhIjKlMnOpQrStUvWxYz

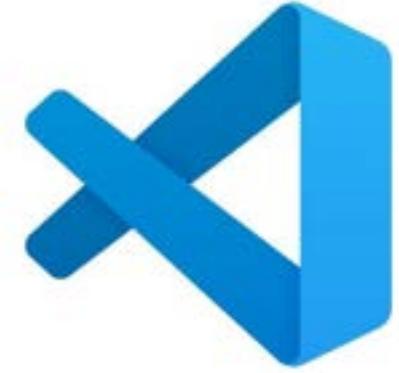
AbCdEfGhIjKlMnOpQrStUvWxYz

AbCdEfGhIjKlMnOpQrStUvWxYz

저작권 문제가 없어 무료로 사용이 가능하며
깔끔한 디자인에 기존 타 사이트에서도 많이 사용하는 폰트이기에 사용자 친숙

작업툴

IDE



VSCode Ver_1.93.1

OS



Windows 10



Windows 11

Windows 10 Ver_22H2

Windows 11 Ver_23H2

런타임 환경



Node.JS
Ver_22.12.0

DBMS



MariaDB

MariaDB Ver_11.5.2

OA

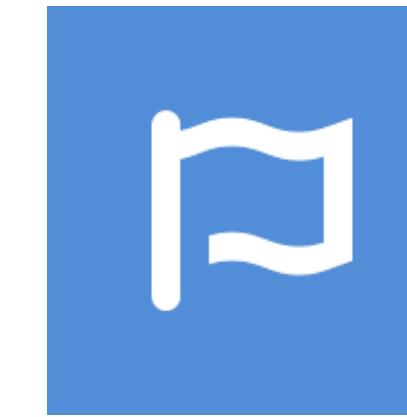


PowerPoint Office16



Excel Office16

디자인



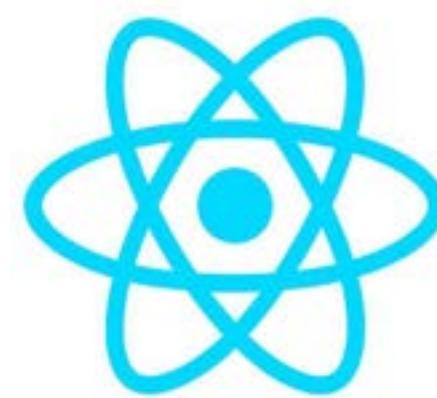
FontAwsome Ver_6.6.0



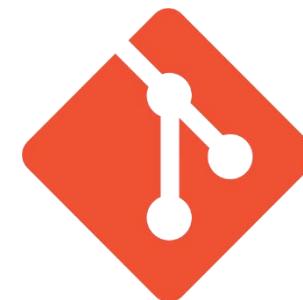
그림판 Ver_11.2406.42.0

작업툴

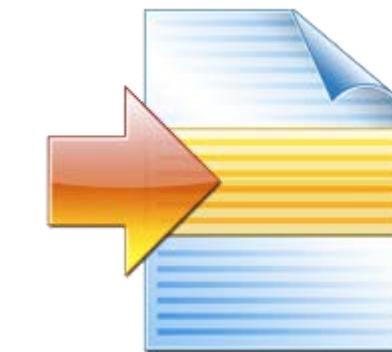
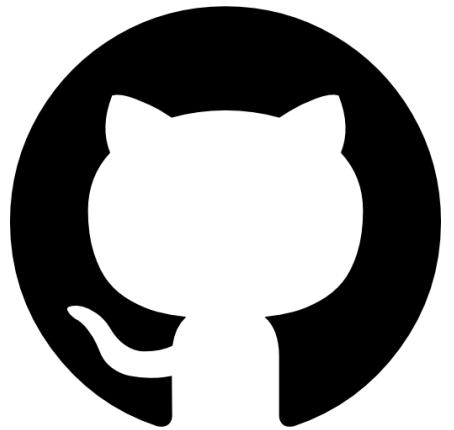
웹 클라이언트



협업도구



git



카카오톡

Git Ver_2.46.0.windows.1

Github

SourceTree Ver_3.4.19

WinMerge Ver_2.16.42.1

요구사항 정의서 - 사용자

우선순위	분류	소분류	기능	중요도	전제조건
1	장바구니	상품 목록	장바구니 담기 버튼을 누르면 해당 상품 및 옵션정보, 추가금, 수량 등이 저장되며, 저장된 정보를 장바구니로 가져와 상품목록을 조회.	상	홈페이지 우측 상단 ↓ 장바구니 아이콘 ----- 상품리스트 ↓ 장바구니 아이콘
		선택 상품 삭제	각 상품목록의 체크박스를 선택 후 선택 상품 삭제버튼을 누르면 해당 상품이 삭제.		
		쇼핑 계속하기	클릭 시 전체상품으로 연결.		
		합계 표시 구역	선택된 상품 금액이 실시간으로 반영.		
		상품 주문	주문 페이지로 연결.		
2	상품주문	주문 상세 내역	장바구니에서 보여줬던 상품 정보들을 조회.	상	장바구니 ↓ 상품주문 버튼
		장바구니 가기	장바구니로 연결.		
		주문자 정보	주문자 정보는 고정된 회원정보로 사용.		
		배송 정보	휴대폰 번호, 받으실 분, 받으실 주소 등을 필수적으로 입력하고 유효성 검사를 진행.		
		결제 정보	사용자가 보유한 잔여 마일리지를 표시. 사용자가 사용할 마일리지 양을 입력 시 실시간으로 업데이트. 잔여 마일리지와 결제금액을 변동하여 업데이트. 가지고 있는 양 이상의 마일리지를 사용하려고 할 시 잔여 마일리지만큼만 사용.		
		결제수단 선택	결제 방법을 신용카드, 휴대폰 중에 체크박스로 선택.		
		최종 결제 금액	마일리지 입력 값, 할인율에 따라 값 변동.		
		동의란	체크박스에 체크해야 결제 가능.		
3	주문완료	결제하기	필수입력사항 유효성 검사(이름-한글만/ 주소-한글,영어,숫자만/ 이메일/ 휴대폰번호) 후 주문 완료 페이지로 연결. 데이터를 추가. 그 후 주문완료 페이지로 이동.	중	상품주문 ↓ 결제하기 버튼
		주문 요약	결제수단, 주문번호, 받으실 분 이름, 주소, 휴대폰번호, 결제금액 등을 화면에 표시.		
		확인 버튼	클릭 시 메인페이지로 연결.		

요구사항 정의서 - 사용자

우선순위	분류	소분류	기능	중요도	전제조건
1	회원가입 정보입력	아이디	영어 대/소문자+숫자 조합 최소 4자 이상, 12자 이하 가능. 조건 일치시, 불일치시 각각 메시지 출력. 중복확인기능 및 메시지 출력. 입력하지 않고 회원가입 누를 시 아이디가 유효하지 않다는 메시지창 팝업.	상	<p>회원가입 안내 ↓ 쇼핑몰 회원가입 버튼 ↓ 필수약관동의 후 다음단계 버튼</p>
		비밀번호	영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합, 10자 이상. 조건 일치시, 불일치시 각각 메시지 출력. 입력하지 않고 회원가입 누를 시 비밀번호가 유효하지 않다는 메시지창 팝업.		
		비밀번호 확인	위 비밀번호 조건과 동일. 위 비밀번호와 일치, 불일치 한다는 메시지 출력. 불일치시 회원가입 버튼 누르면 비밀번호가 일치하지 않습니다 메시지창 팝업.		
		이름	한글 2자~6자 입력 가능. 입력하지 않고 회원가입 누를 시 이름을 입력해주세요 메시지창 팝업.		
		닉네임	선택사항이지만 입력 시 중복체크 및 메시지 출력. 한글, 영문자, 숫자 가능. 특수문자 불가능. 최소 2자 이상 10자 이하. 조건 일치시, 불일치시 각각 메시지 출력.		
		이메일	로컬파트(영문자, 숫자, . _ %+- 만 가능) @ 도메인(영문자와 숫자, .- 만 가능). 최상위도메인(영문자만 허용, 최소 2글자 이상). 도메인 선택 시 로컬파트 뒤에 자동으로 입력. 조건 일치시, 불일치시 각각 메시지 출력.		
		휴대폰 번호	번호형식은 xxx-xxxx-xxxx. 미입력 또는 조건 불일치 시 회원가입 버튼 누를 시 휴대전화 번호 형식이 유효하지 않습니다 메시지창 팝업.		
		주소	한글, 숫자, 영문자, 공백, 특수문자 '-'만 허용, 최소 8자 이상.		
		생년월일	선택사항. 미입력도 통과.		
		가입완료	회원 가입 완료.		
2	회원가입 완료	홈으로, 로그인 버튼	회원가입 완료 페이지 (회원이름)님의 회원가입을 축하합니다라는 메시지가 출력. 홈으로 버튼 누를 시 메인페이지로, 로그인 버튼 누를 시 로그인 페이지로 이동.	하	회원가입 정보입력 ↓ 회원가입 버튼

요구사항 정의서 - 사용자

우선순위	분류	소분류	기능	중요도	전제조건
1	회원가입 안내	쇼핑몰 회원가입	회원가입 페이지로 이동.	하	메인페이지 우측 상단 회원가입
2	회원가입 약관동의	이용 약관 동의	약관 전체동의 체크시 전체동의 체크 및 개별 체크 .	하	회원가입 안내 ↓ 쇼핑몰 회원가입 버튼
		이전	회원가입 진입 전 회원가입 안내화면으로 이동.		
		다음	회원가입 정보입력 페이지로 이동(필수항목 체크시에만 가능. 필수항목 미 체크시 필수 약관에 모두 동의해주세요 라는 메시지 출력.		
3	아이디 찾기	이름, 이메일 입력	도메인 선택시 @와 도메인이름이 자동으로 이메일 아이디 뒤에 붙음.	중	홈페이지 접속 ↓ 헤더 우측 로그인 ↓ 회원 로그인 폼 하단 아이디 찾기 비밀번호 찾기 버튼
		아이디 찾기 버튼	데이터베이스에 등록된 아이디와 이메일 일치 시 찾기 완료. 불일치시 찾기 불가.		
		아이디 찾기 완료	찾고자 하는 회원 아이디 출력.		
		비밀번호 찾기, 로그인하기	비밀번호 찾기 페이지, 로그인 페이지로 이동.		
4	비밀번호 찾기	이름, 아이디 입력	이름과 아이디 입력. 유효성 검사는 하지 않음.	중	홈페이지 접속 ↓ 헤더 우측 로그인 ↓ 회원 로그인 폼 하단 아이디 찾기 비밀번호 찾기 버튼
		다음	이름과 아이디가 일치 시 비밀번호 재설정 가능. 불일치시 회원정보가 일치하지 않습니다 라는 메시지 출력.		
		아이디 찾기	아이디 찾기 페이지.		
		비밀번호 재설정 페이지	영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합, 10자 이상 재설정. 조건 불충족 조건에 맞게 입력하라는 메시지 출력. 조건은 충족하지만 위 아래 비밀번호 불일치시 비밀번호가 일치하지 않습니다 메시지 출력		
		새 비밀번호 입력	유효성 조건에 맞고 두 비밀번호가 일치해야 함		
		확인	위 조건들을 충족시키면 비밀번호 변경완료 메시지와 함께 비밀번호 변경완료 화면 이동.		
		비밀번호 변경 완료 페이지	비밀번호가 성공적으로 변경되었다는 메시지와 로그인 버튼 누를 시 로그인 페이지로 이동.		

요구사항 정의서 - 사용자

우선순위	분류	소분류	기능	중요도	전제조건
1	로그인	로그인 페이지 로그인 버튼	로그인 화면 아이디, 비밀번호 미입력시 '입력란을 작성하세요' 메시지. 아이디, 비밀번호 불일치시 '아이디 또는 비밀번호를 잘못입력하셨습니다.' 출력. 아이디와 비밀번호 일치시 정상 로그인 되면 메인 화면으로 이동	상	홈페이지 접속 ↓ 헤더 우측 상단 로그인
2	마이페이지	진행 중인 주문 최근 주문 정보 마일리지 회원 정보 수정	배송준비, 배송중, 배송완료, 구매확정 / 취소, 반품 몇 건씩 인지 표시 배송준비, 배송중 상태에서는 주문취소 가능, 배송완료 상태에서는 구매확정, 반품완료만 가능구매확정인 상태에서는 후기 등록 가능. 구매 확정될 시 구매 총 금액의 10%가 마일리지로 적립됨. 현재 보유 중인 마일리지 표시. 마일리지 점수 클릭 시 사용 및 적립 내역이 담긴 표가 출력.(적립 날짜, 적립내용, 유효날짜, 변동사항 확인 가능) 돌아가기 버튼 누를 시 마일리지 내역 표는 사라지고 기존 주문 정보 확인 내역들 출력 회원 정보 수정 전 비밀번호 인증비밀번호 일치 시 회원 정보 수정 페이지 진입. 아이디 제외한 개인정보 수정 가능. (유효성은 회원가입과 동일)	상	홈페이지 접속 ↓ 헤더 우측 상단 사람 아이콘 클릭
3	메인화면	메인 배너 영상 섹션3	제품 등을 슬라이드 형식으로 보여주고 각 배너 클릭 시 해당 페이지로 이동.	중	홈페이지 접속
5	푸터	대표번호 링크	회사정보, 고객센터 전화번호와 운영시간, 이메일 표시 및 SNS링크 바로가기	하	홈페이지 접속
6	네비게이션	메인, 서브메뉴	메뉴에 마우스 올리면 메뉴명 색상 변경, 하단에 하위 서브메뉴 활성화	중	홈페이지 접속

요구사항 정의서 - 사용자

우선순위	분류	소분류	기능	중요도	전제조건
1	상품 리스트	상품 전체보기	등록 된 총 상품 개수 출력 등록 된 상품의 이름, 가격 데이터 출력 할인율이 있다면 원가에 자동 계산되어 최종금액으로 출력	상	네비게이션 ↓ 전체상품 ↓ 전체보기
2	상품 상세	상품 정보 조회	등록 된 상품의 이름, 가격 데이터 출력 상품 최종 선택 후 장바구니에 구매 할 상품 데이터 전달 할인율이 있다면 원가에 자동 계산되어 최종금액으로 출력	상	상품 리스트 클릭
		옵션선택	옵션 선택 후 추가 시 유료 옵션일 경우 옵션가격이 포함되어 최종금액으로 출력		
		슬라이드	선택한 이미지 확인 가능. 이미지 목록 슬라이드 기능		
3	게시물 리스트	전체후기	작성 및 수정 가능	상	네비게이션 ↓ 고객후기 - 전체후기 고객센터 - 공지사항 고객센터 - 상품문의 고객센터 - FAQ
		포토후기	작성(파일 업로드) 및 수정 가능		
		공지사항	읽기 가능		
		상품문의	전체 글 조회 및 작성, 수정 가능 (답변처리 유무 확인 가능)		
		FAQ	읽기 가능		
		1:1 문의	본인 작성 글 조회 및 작성, 수정 가능 (답변처리 유무 확인 가능)		
4	게시물 상세		각 상세 페이지에 맞는 (수정, 삭제, 목록으로) 기능 활성화	중	게시물 리스트 클릭

요구사항 정의서 - 관리자

우선순위	분류	소분류	기능	중요도	전제조건
1	관리자 로그인	로그인	관리자 페이지는 접근 권한이 관리자인 아이디로만 로그인 가능	상	관리자 페이지 접속
2	상품등록	등록	새로운 상품의 이름, 가격, 사진, 재고, 설정 가능한 옵션/상세정보 키워드 선택, 상품 사용 여부, 가격/할인 데이터 등록	상	관리자 페이지에서 상품등록 페이지 접속
3	상품목록	수정	등록 된 상품 정보 수정 후 저장	상	관리자 페이지에서 상품목록 페이지 접속
		조회	등록 된 전체 상품 리스트 조회 및 각 상품 상세 조회		
4	옵션관리	등록	상품 등록 시 필요한 옵션 / 옵션값 신규 등록	상	관리자 페이지에서 옵션 관리 페이지 접속
		수정	등록 된 옵션/옵션값 수정		
		삭제	등록 된 옵션/옵션값 삭제		
5	키워드 관리	등록	상품 검색 시 필요한 상품 상세정보 키워드 / 키워드 값 신규 등록	중	관리자 페이지에서 상세정보 키워드 페이지 접속
		수정	등록 된 상품 상세정보 키워드 / 키워드 값 수정		
		삭제	등록 된 상품 상세정보 키워드 / 키워드 값 삭제		

요구사항 정의서 - 관리자

우선순위	분류	소분류	기능	중요도	전제조건
1	주문관리	시작일	시작일을 기점으로 그 이후의 데이터만 포함하여 조회할 수 있는 필터	상	관리자 페이지에서 주문 조회 페이지 접속
		종료일	종료일을 기점으로 그 이전의 데이터만 포함하여 조회할 수 있는 필터		
		주문 상태	"전체", "입금대기", "입금완료", "배송준비", "배송시작", "배송완료" 의 상태별로 나누어 조건에 맞는 데이터 조회		
		결제 방법	"전체", "휴대폰", "신용카드" 의 상태별로 나누어 조건에 맞는 데이터 조회		
		조회	"전체", "구매확정", "구매미확정" 의 상태별로 나누어 조건에 맞는 데이터 조회		
		초기화	필터에 맞는 조건들을 선택한 후, 조회버튼을 클릭하면 필터에 맞는 데이터들을 조회		
2	환불관리	시작일	시작일을 기점으로 그 이후의 데이터만 포함하여 조회할 수 있는 필터	상	관리자 페이지에서 환불 관리 페이지 접속
		종료일	종료일을 기점으로 그 이전의 데이터만 포함하여 조회할 수 있는 필터		
		상태	"전체", "취소요청", "반품요청", "교환요청", "취소완료", "반품완료", "교환완료" 의 상태별로 나누어 조건에 맞는 데이터 조회		
		조회	필터에 맞는 조건들을 선택한 후, 조회버튼을 클릭하면 필터에 맞는 데이터들을 조회하는 기능		
		초기화	모든 필터와 조건들을 초기화		
3	정산관리	일별 조회	각 일별에 해당하는 상품 데이터의 총 매출액을 조회	중	관리자 페이지에서 정산 관리 페이지 접속
		주별 조회	각 주별에 해당하는 상품 데이터의 총 매출액을 조회		
		월별 조회	각 월별에 해당하는 상품 데이터의 총 매출액을 조회		
		상품별 조회	각 상품 데이터의 총 매출액을 조회		

요구사항 정의서 - 관리자

우선순위	분류	소분류	기능	중요도	전제조건
4	게시판 답변 관리	시작일	시작일을 기점으로 그 이후의 데이터만 포함하여 조회할 수 있는 필터	상	관리자페이지에서 답변 관리 페이지 접속
		종료일	종료일을 기점으로 그 이전의 데이터만 포함하여 조회할 수 있는 필터		
		게시판 구분	조회할 게시판 종류를 구분하는 필터		
		제목	제목을 검색할 수 있는 필터		
		작성자	작성자를 검색할 수 있는 필터		
		내용	내용 클릭 시 상세 내용 확인 및 코멘트 달수 있는 항목 표시		
		조회	필터에 맞는 조건들을 선택한 후, 조회버튼 클릭 시 필터에 맞는 데이터들을 조회		
		삭제	게시글 삭제 처리		
5	게시판 등록 관리	시작일	시작일을 기점으로 그 이후의 데이터만 포함하여 조회할 수 있는 필터	중	관리자페이지에서 등록 관리 페이지 접속
		종료일	종료일을 기점으로 그 이전의 데이터만 포함하여 조회할 수 있는 필터		
		게시판 구분	조회할 게시판 종류를 구분하는 필터		
		제목	제목을 검색할 수 있는 필터		
		작성자	작성자를 검색할 수 있는 필터		
		내용	내용 클릭 시 상세 내용 확인 및 코멘트 달수 있는 항목 표시		
		조회	필터에 맞는 조건들을 선택한 후, 조회버튼 클릭 시 필터에 맞는 데이터들을 조회		
		글쓰기	필터에 맞는 게시판에 글 작성		
		수정	선택한 글 제목 및 내용 수정		
		삭제	게시글 삭제 처리		

요구사항 정의서 - 관리자

우선순위	분류	소분류	기능	중요도	전제조건
4	회원 관리	등록	회원등록 버튼 클릭시 회원 신규 등록	상	관리자페이지에서 회원 정보 관리 페이지 접속
		내용	회원 정보 클릭시 상세 정보 확인		
		수정	회원 상세 정보 품에서 수정 버튼 클릭 시 회원 상세 정보 수정		
		삭제	회원 상세 정보 품에서 삭제 버튼 클릭 시 회원 상세 정보 삭제		
		마일리지 내역 조회	회원 정보 클릭 시 회원의 사용되거나 적립된 마일리지 내역 출력		
		회원 id 조회	회원 아이디를 검색하여 회원 정보를 조회		
		회원 이름 조회	회원 이름을 검색하여 회원 정보를 조회		
		회원 구분 조회	회원 구분별로 검색하여 회원 정보를 조회		
		회원 등급 조회	회원 등급별로 검색하여 회원 정보를 조회		
		가입 시작일 조회	가입 기간으로 검색하여 회원 정보를 조회		
		검색 초기화	초기화 버튼을 클릭 시 검색 필터 초기화 및 회원 데이터 목록 재정렬		

요구사항 정의서 – API

우선순위	분류	소분류	기능	중요도	전제조건
1	지도 api	매장 위치 확인	메인의 고객센터-> 지점으로 이동	중	홈페이지 고객센터 ↓ 지점 ----- 결제 진행 ↓ 지점 선택 버튼
		매장 위치 검색	검색창 또는 우측 목록과 페이지네이션을 통해 원하는 지점을 선택, 정보와 위치를 확인할 수 있음..		
		배달 지점 선택	장바구니->결제 진행 페이지에서 지점 선택 버튼을 눌러 모달창을 통해 배송받을 지점을 선택 가능		
2	스마트 에디터 api (Quill 스마트에디터)	주문 상세 내역	장바구니에서 보여줬던 상품 정보들을 조회.	중	장바구니 ↓ 상품주문 버튼
		장바구니 가기	장바구니로 연결.		
		주문자 정보	주문자 정보는 고정된 회원정보로 사용.		
		배송 정보	휴대폰 번호, 받으실 분, 받으실 주소 등을 필수적으로 입력하고 유효성 검사를 진행.		
		결제 정보	사용자가 보유한 잔여 마일리지를 표시. 사용자가 사용할 마일리지 양을 입력 시 실시간으로 업데이트. 잔여 마일리지와 결제금액을 변동하여 업데이트. 가지고 있는 양 이상의 마일리지를 사용하려고 할 시 잔여 마일리지만큼만 사용.		
		결제수단 선택	결제 방법을 신용카드, 휴대폰 중에 체크박스로 선택.		
		최종 결제 금액	마일리지 입력 값, 할인율에 따라 값 변동.		
		동의란	체크박스에 체크해야 결제 가능.		
		결제하기	필수입력사항 유효성 검사(이름-한글만/ 주소-한글,영어,숫자만/ 이메일/ 휴대폰번호) 후 주문 완료 페이지로 연결. 데이터를 추가. 그 후 주문완료 페이지로 이동.		
3	(카카오) 연동 로그인 api	카카오로그인	클릭 시 필수 동의항목 체크 후 회원가입	상	로그인 ↓ 카카오 로그인 버튼
		회원 아이콘 클릭	클릭 시 가입한 카카오 계정 이름으로 알림창.		

시스템 구조도 - 전체

관리자 페이지

PORT : 3000
(react)

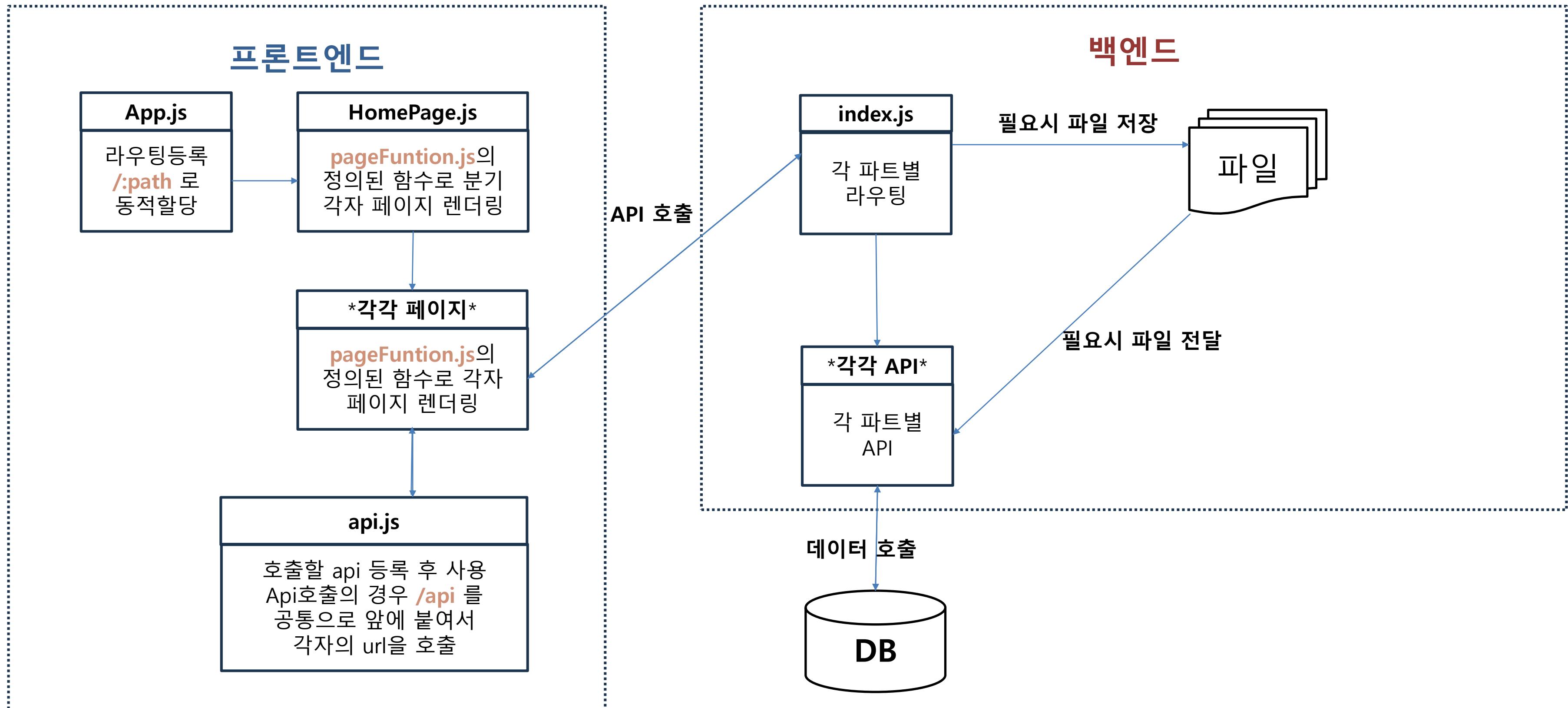
데이터 통신
Axios 사용

사용자 페이지(쇼핑몰)

PORT : 5500
(LiveServer)
(작업 경과 확인하여 react로
マイグ레이션 예정)

데이터 통신
Ajax 사용 예정

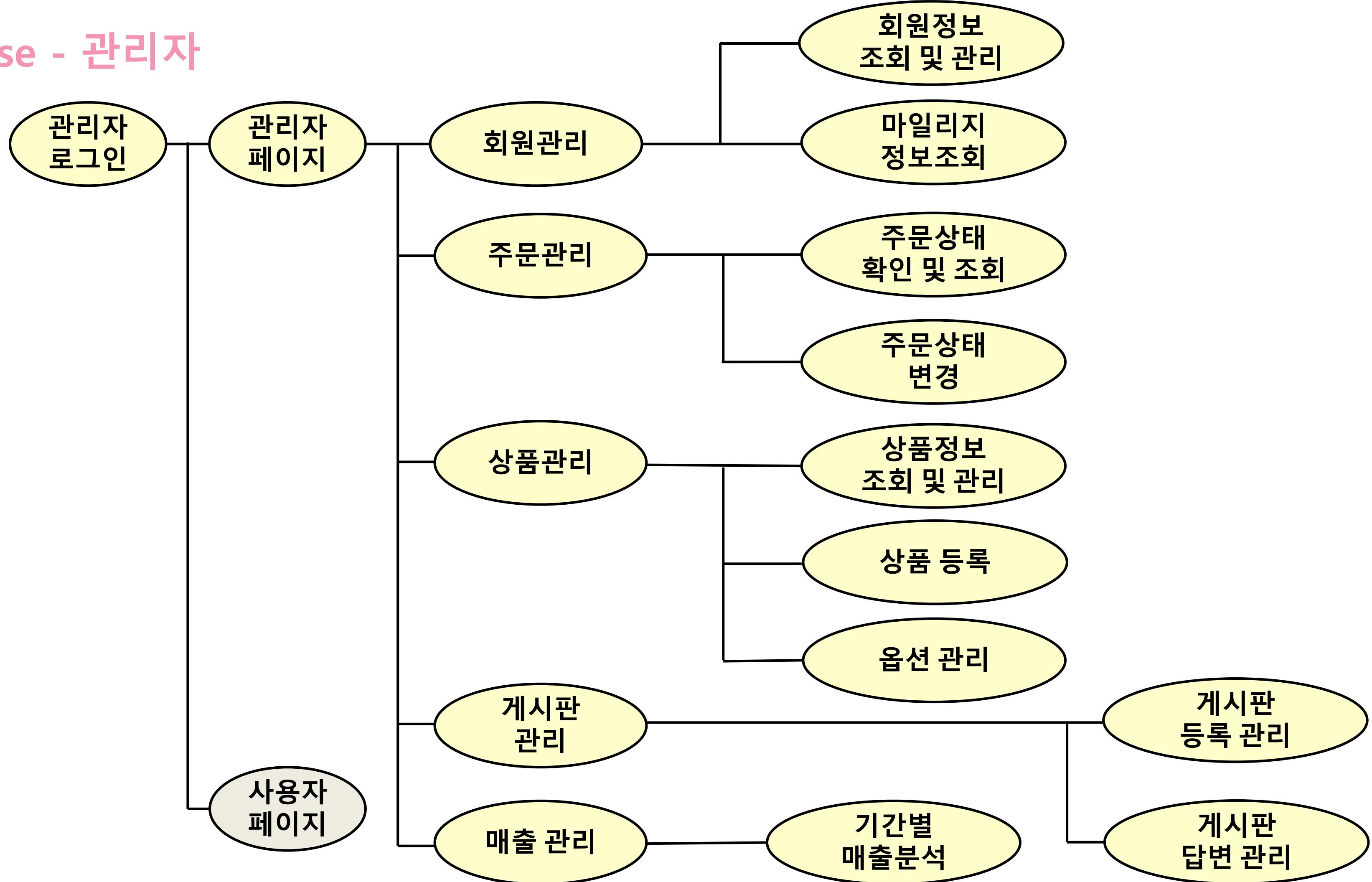
시스템 구조도 - 관리자 시스템



Use Case - 관리자



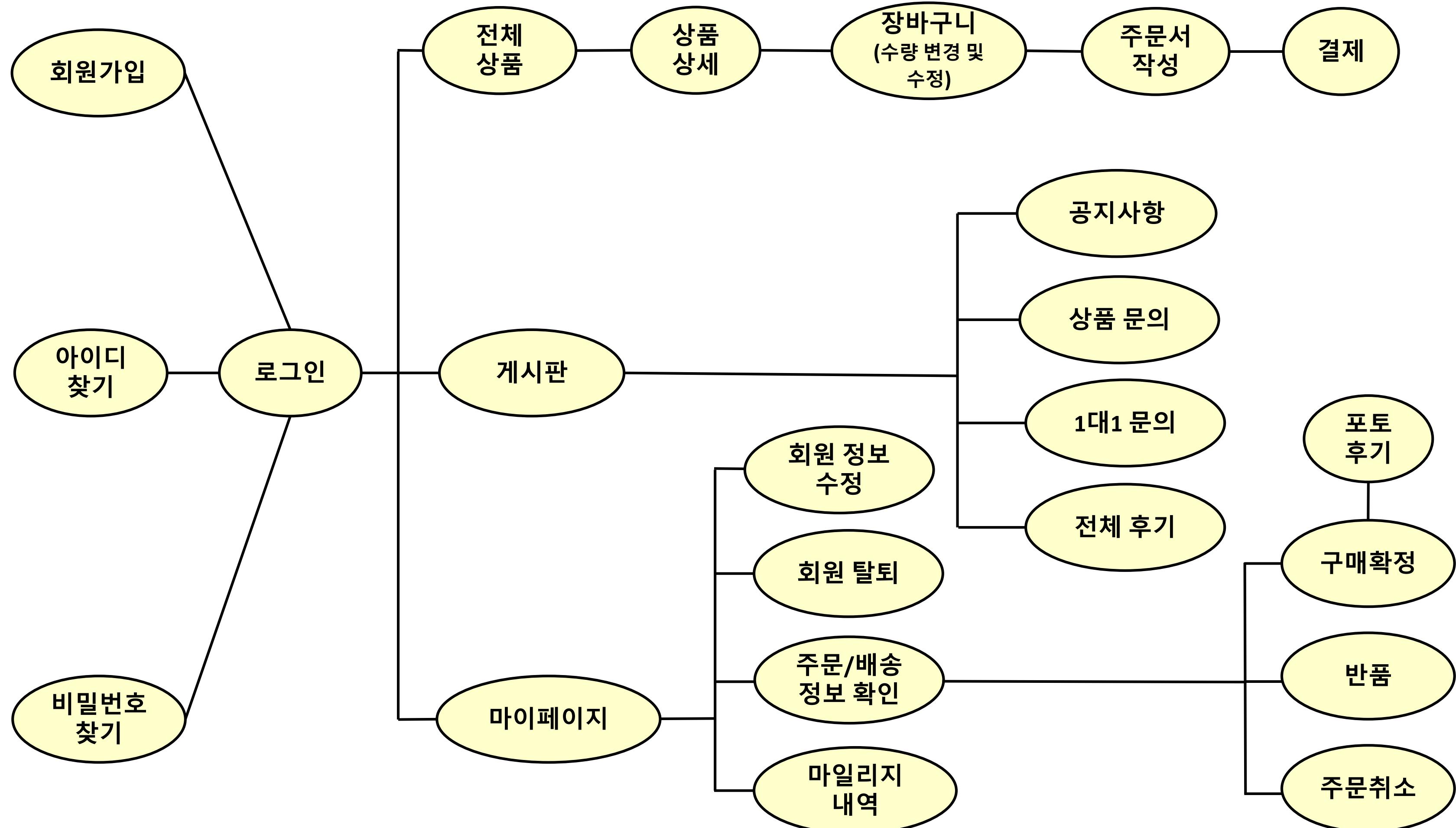
관리자



Use Case - 사용자



사용자



ERD - 전체



ERD – 회원 관리

mem_info		mem_info	
Column	회원id	id	Domain
회원id	비밀번호	pw	Domain VARCHAR(255)
회원명	회원명	name	Domain VARCHAR(100)
닉네임	닉네임	nick	Domain VARCHAR(50)
생년월일	생년월일	birth_date	Domain DATE
휴대전화	휴대전화	phone_num	Domain VARCHAR(20)
이메일	이메일	email	Domain VARCHAR(100)
주소	주소	addr	Domain VARCHAR(255)
접근권한	접근권한	role	Domain ENUM('관리자','회원','비회원')
회원구분	회원구분	type	Domain ENUM('일반','블랙리스트','휴면계정')
회원등급	회원등급	level	Domain ENUM('브론즈','실버','골드','VIP')
가입일시	가입일시	join_date	Domain DATETIME
최근접속일	최근접속일	last_date	Domain TIMESTAMP
잔여마일리지	잔여마일리지	m_remain	Domain INT
작성자	작성자	reg_id	Domain VARCHAR(60)
작성일시	작성일시	reg_date	Domain DATETIME
수정자	수정자	upt_id	Domain VARCHAR(60)
수정일시	수정일시	upt_date	Domain DATETIME

회원 상세 정보를 담은 테이블

ERD – 회원 관리

마일리지 mileage				
	순서	turn	Domain	INT
PK	회원id	id	Domain	VARCHAR(50)
	적립날짜	earn_date	Domain	DATETIME
	적립내용	description	Domain	TEXT
	유효날짜	valid_date	Domain	DATE
	변동내역	change_val	Domain	INT
	작성자	reg_id	Domain	VARCHAR(60)
	작성일시	reg_date	Domain	DATETIME
	수정자	upt_id	Domain	VARCHAR(60)
	수정일시	upt_date	Domain	DATETIME

회원 ID를 참조하여
회원 마일리지 정보를 담은 테이블

ERD – 주문/배송

주문

			Domain	INT
⑦	주문번호	order_id	Domain	INT
회원ID	member_id	Domain	INT	
주문일자	order_date	Domain	DATETIME	
결제수단	pay_method	Domain	ENUM('휴대폰','신용카드')	
마일리지 순서	used_mileage	Domain	INT	
구매확정여부	is_confirmed	Domain	CHAR(1)	
요청사유	req_reasons	Domain	VARCHAR(255)	
요청거절사유	refuse_reasons	Domain	VARCHAR(255)	
작성자	reg_id	Domain	VARCHAR(50)	
작성일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(50)	
수정일시	upt_date	Domain	DATETIME	

회원ID를 참조하여
주문 정보를 담은 테이블

orders

상품별 주문

			Domain	INT
⑦	주문상품id	order_product_id	Domain	INT
주문번호	order_id	Domain	INT	
상품 번호	product_id	Domain	INT	
상품 수량	quantity	Domain	INT	
상품 단가	unit_price	Domain	INT	
작성자	reg_id	Domain	VARCHAR(50)	
작성일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(50)	
수정일시	upt_date	Domain	DATETIME	

주문번호와 상품번호를 참조하여
주문별 상품 정보를 담은 테이블

ERD – 주문/배송

장바구니				
Key	장바구니ID	cart_id	Domain	INT
	상품ID	product_id	Domain	INT
	회원ID	member_id	Domain	VARCHAR(50)
	갯수	quantity	Domain	INT
	작성자	reg_id	Domain	VARCHAR(50)
	작성일시	reg_date	Domain	DATETIME
	수정자	upt_id	Domain	VARCHAR(50)
	수정일시	upt_date	Domain	DATETIME

회원 ID와 상품ID를 참조하여
장바구니 정보를 담은 테이블

ERD – 주문/배송

배송				deliveries
배송 번호	delivery_id	Domain	INT	
주문번호	order_id	Domain	INT	
회원ID	member_id	Domain	INT	
주문 상태	order_status	Domain	ENUM('배송준비','배송시작','배송완료','취소요청','반품요청','교환요청','취소완료','반품완료','교환완료')	
배송준비날짜	prep_date	Domain	DATETIME	
배송시작날짜	start_date	Domain	DATETIME	
배송완료날짜	done_date	Domain	DATETIME	
취소요청날짜	cancel_req_date	Domain	DATETIME	
반품요청날짜	return_req_date	Domain	DATETIME	
교환요청날짜	exchange_req_date	Domain	DATETIME	
취소완료날짜	cancel_done_date	Domain	DATETIME	
반품완료날짜	return_done_date	Domain	DATETIME	
교환완료날짜	exchange_done_date	Domain	DATETIME	
작성자	reg_id	Domain	VARCHAR(50)	
작성일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(50)	
수정일시	upt_date	Domain	DATETIME	

상품별 주문 테이블의 주문번호
 회원 정보 테이블의 회원 ID를 참조하여
 배송 정보를 담은 테이블

ERD – 상품/옵션

상품

products				
상품id	id	Domain	VARCHAR(14)	
상품명	name	Domain	VARCHAR(100)	
가격	price	Domain	INT	
할인율	discount	Domain	INT	
상세이미지	detail_exist	Domain	BOOLEAN	
등록재고수량	quan	Domain	INT	
사용여부	status	Domain	BOOLEAN	
혜택	benefit	Domain	TEXT	
등록자	reg_id	Domain	VARCHAR(60)	
등록일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(60)	
수정일시	upt_date	Domain	DATETIME	

상품 ID 포함
상품 정보를 담은 테이블

상품 디테일

product_detail				
하위값고유코드	code	Domain	INT	
코드	main_code	Domain	INT	
상품id	id	Domain	VARCHAR(14)	
순서	tum	Domain	INT(2)	
옵션키워드구분	type	Domain	CHAR(1)	
등록자	reg_id	Domain	VARCHAR(60)	
등록일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(60)	
수정일시	upt_date	Domain	DATETIME	

상품 ID를 참조하여
상품 상세 정보를 담은 테이블

ERD – 상품/옵션

옵션키워드항목		option_keyword		
코드	code	Domain	INT	
구분	type	Domain	BOOLEAN	
명칭	name	Domain	VARCHAR(100)	
필수선택여부	req	Domain	BOOLEAN	
사용여부	status	Domain	BOOLEAN	
등록자	reg_id	Domain	VARCHAR(60)	
등록일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(60)	
수정일시	upt_date	Domain	DATETIME	

상품에 관한 옵션 항목정보를
담고 있는 테이블

옵션키워드하위값		opKey_value		
하위값고유코드	code	Domain	INT	
코드	main_code	Domain	INT	
구분	type	Domain	BOOLEAN	
값 명칭	name	Domain	VARCHAR(100)	
값 가격	price	Domain	INT	
사용여부	status	Domain	TINYINT(1)	
등록자	reg_id	Domain	VARCHAR(60)	
등록일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(60)	
수정일시	upt_date	Domain	DATETIME	

옵션 키워드 항목을 참조하여
옵션의 하위옵션 정보 테이블

ERD – 게시판

게시판			
			board
고유id	id2	Domain	VARCHAR(14)
분류코드	code	Domain	CHAR(2)
제목	title	Domain	VARCHAR(100)
내용	contents	Domain	TEXT
첨부이미지코드	img_no	Domain	VARCHAR(14)
상품ID	products_id	Domain	VARCHAR(14)
별점	score	Domain	INT(1)
분류	division	Domain	TINYINT
조회수	read_no	Domain	INT(3)
등록자	reg_id	Domain	VARCHAR(60)
등록일시	reg_date	Domain	DATETIME
수정자	upt_id	Domain	VARCHAR(60)
수정일시	upt_date	Domain	DATETIME

게시판 고유 ID를 포함한
게시판 정보를 담은 테이블

답변				
				answer
고유id	id	Domain	VARCHAR(14)	
순번	turn	Domain	INT(3)	
코멘트	comment	Domain	text	
delete_yn	delete_yn	Domain	TINYINT(1)	
등록자	reg_id	Domain	VARCHAR(60)	
등록일시	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(60)	
수정일시	upt_date	Domain	DATETIME	

게시판 고유 ID를 참조하는
게시판 답변 테이블

ERD – 게시판

파일관리				file_manage
고유id	id	Domain	VARCHAR(14)	
순서	turn	Domain	int(2)	
요청url	api_url	Domain	VARCHAR(50)	
원본파일명	org_name	Domain	VARCHAR(100)	
변경파일명	new_name	Domain	VARCHAR(255)	
확장자명	ext	Domain	VARCHAR(10)	
저장 경로	path	Domain	VARCHAR(255)	
파일크기	size	Domain	BIGINT	
설명	dsct	Domain	TEXT	
등록자	reg_id	Domain	VARCHAR(60)	
등록일자	reg_date	Domain	DATETIME	
수정자	upt_id	Domain	VARCHAR(60)	
수정일자	upt_date	Domain	DATETIME	

게시판에 업로드 되는
파일관리 테이블

역할분담

게시판

정성윤

- 관리자
공지, FAQ 글 작성, 전체후기, 포토후기 글 삭제 전환, 상품문의, 1:1 문의
답변작성

글 읽기, 전체후기, 포토후기, 상품문의, 1:1문의 글 작성 및 삭제 및 읽기와 답변

여부 확인 주문 및 배송, API

임세훈

- 관리자
주문, 환불, 정산조회 및 취소, 환불 요청에 대한 승인, 거절 기능
상태, 날짜(일, 주, 월별) 결제수단, 상품별 조회 기능
- 사용자
장바구니페이지, 주문페이지, 주문완료페이지, 마이페이지
주문데이터 동적 이동 및 옵션값과 옵션추가금 반영
취소, 반품요청, 구매확정, 후기등록 등의 기능, 마일리지 변동량 확인 가능
(카카오)지도 api, 현재 위치기반 매장 표시 및 선택

역할분담

상품관리

우수정

- 관리자
상품 이름, 이미지, 사용 옵션/키워드 설정 등록, 재고 등록, 상품 가격, 할인율 자동계산
- 사용자
등록된 상품 목록 조회
등록된 상품 목록 수, 상품 정보 조회, 옵션값 추가금, 정보 반영 후 장바구니로 전달

회원관리

김수훈

- 관리자
회원 등록, 회원정보 조회(회원 ID, 이름, 구분, 등급, 가입날짜 별), 수정, 삭제 및 마일리지 내역 조회
- 사용자
회원가입, 로그인, 로그아웃, 마일리지 상세내역 조회, 회원정보 수정, 아이디 및 비밀번호 찾기, 회원 탈퇴기능

시간계획서01

우선순위	기능	11월												12월										
		13	14	15	18	19	20	21	22	25	26	27	28	29	2	3	4	5	6	9	10	11	12	13
1	주제선정																							
2	파트분배																							
3	요구사항																							
4	사이트분석																							
5	Use Case																							
6	ERD																							
7	공지사항																							
7	FAQ																							
8	전체후기																							
9	포토후기																							
10	상품문의																							
11	1대1문의																							
12	상품 조회																							
13	상품 등록																							
14	상품 수정																							
15	메인 페이지																							
16	상품 리스트																							
17	상품 상세																							

테스트

완료

시간계획서02

시간계획서03



반응형 웹/앱 가구 쇼핑몰 시연 화면

O1 장바구니

Description List

장바구니					
		수량	상품금액	할인/적립	합계금액
01	<input checked="" type="checkbox"/> 상품/옵션 정보	3개	130,000원	0% (-0원)	390,000원
02	<input checked="" type="checkbox"/> 상품1번 상품1번 나쁜짓을하면은~ / 노랑	2개	165,800원	0% (-0원)	331,600원
03	<input checked="" type="checkbox"/> 상품1번 상품1번 나쁜짓을하면은~ / 초록	1개	155,800원	0% (-0원)	155,800원
04	<input type="button" value="선택 상품 삭제"/>			<input type="button" value="쇼핑 계속하기"/>	
05			상품금액 877,400원	배송비 0원	합계 877,400원
06	<input type="button" value="상품 주문"/>				

체크박스를 통해 상품선택 가능

데이터를 통해 불러온 장바구니 상품정보

선택한 상품 삭제 가능

전체상품 페이지로 이동

각 금액을 자동으로 합산해서 표시함

주문으로 이동

○1장바구니

Description List

01

```
try {
  // 옵션 데이터를 파싱 (JSON 문자열인지 확인)
  const parsedOptions = typeof cart.options === 'string'
    ? JSON.parse(cart.options) // JSON 문자열일 경우 파싱
    : cart.options; // 객체일 경우 그대로 사용

  // 옵션 값을 보기 좋은 형식으로 표시
  if (parsedOptions && typeof parsedOptions === 'object') {
    return Object.values(parsedOptions).join(' / '); // 값만 표시
  } else {
    return "옵션: 없음";
  }
} catch (error) {
  console.error("옵션 처리 오류:", error);
  return "옵션 데이터를 표시할 수 없습니다.";
}
})()
```

01

옵션값을 장바구니에 데이터마다 표시

02

장바구니 테이블에 데이터를 추가하는 쿼리

02

```
const insertCartQuery = `
INSERT INTO cart (member_id, product_id, quantity, options, reg_id, option_unit_price)
VALUES (?, ?, ?, ?, ?, ?, ?)
`;
```

O2 주문

Description List

01 데이터를 통해 받아온 주문정보

02 장바구니로 이동

03 주문의 총금액을 자동합산

주문상세내역

01 장바구니 > 02 주문서작성/결제 > 03 주문완료

상품/옵션 정보

수량

상품금액

할인/적립

합계금액

01

상품1번
초코초코 / 초록

3개

100,000원

0%

390,000원

상품1번
나쁜짓을하면은~ / 노랑

2개

100,000원

0%

331,600원

상품1번
나쁜짓을하면은~ / 초록

1개

100,000원

0%

155,800원

02

장바구니 가기

3 종류의 총 상품금액

877,400원

+

배송비

0원

03

합계
877,400원

O2주문

Description List

- O1** 배송정보를 입력하면 DB에 저장
- O2** 보유 마일리지를 조회하며 보유한 양 이하로 사용가능
- O3** 사용한 마일리지 양만큼 총 결제금액이 감소됨

O1 정보

■ 받으실 분	임세훈
■ 받으실 곳	어딘가
전화번호	02-579-6269
■ 휴대폰 번호	010-5796-6269

상품 합계 금액 877,400원

배송비 0원

할인 및 적립 적립 마일리지

쿠폰 사용 쿠폰 조회 및 적용

마일리지 사용 점(보유 마일리지 : 10,026,039점)

최종 결제 금액 877,400원
(마일리지 사용: -0원)

O2주문

Description List

01 결제수단 신용카드, 휴대폰 선택 가능

02 약관동의 유효성

03 주문완료로 이동

결제수단 선택 / 결제

01

신용카드 휴대폰

일반결제

최종 결제 금액 877,400원

02

(필수) 소비자 위약금 부과에 대한 고지와 구매하실 상품의 결제정보를 확인하였으며, 구매진행에 동의합니다.

소비자 위약금 부과에 대한 고지

자코모 소파는 제품의 디자인별 • 색상별 • 가죽 종류별 • 고객 요청사항 등 주문 요건에 따라 각각 완성되는 제품이 달라지는 특이사항으로,
고객이 제품을 주문한 날(결제가 완료된 날)로부터 3일 후 제작에 착수하는 '주문제작상품'입니다.

근거법률(전자상거래 등에서의 소비자보호에 관한 법률 제17조 제2항 제6호, 동법 제19조, 동법 시행령 제21조)에 따라 '청약의 철회가 제한되는 재화'에 해당하므로,
주문청약 취소 요청 시 고객님께 위약금이 부과되거나 교환 • 반품이 불가할 수 있으니 구매 결정 이전, 제품의 온라인 상 상세페이지의 내용(제품 특성 및 명시사항 등)을 반드시 확인해주시기 바랍니다.

03

결제하기

O2주문

할인 및 적립	적립 마일리지
쿠폰 사용	쿠폰 조회 및 적용
O1 마일리지 사용	99999 점(보유 마일리지 : 10,026,039점)
최종 결제 금액	777,401원 (마일리지 사용: -99,999원)
결제수단 선택 / 결제	
일반결제	<input checked="" type="radio"/> 신용카드 <input type="radio"/> 휴대폰

Description List

- O1** 사용된 마일리지 양 표시
- O2** 사용한 양만큼 결제금액 감소

O2주문

01

```
const Order = () => {
  const [formData, setFormData] = useState({
    ordererName: "임세훈",
    ordererPhone: "01057966269",
    ordererMobile: "01057966269",
    ordererEmail: "sehoon@naver.com",
    recipientName: "임세훈",
    recipientAddress: "어딘가",
    recipientPhone: "02-579-6269",
    recipientMobile: "010-5796-6269",
    recipientComment: "",
    mileage: "",
    deposit: "",
    agree: false,
    items: [],
    total_price: 0,
    pay_method: "신용카드",
  });
}

const [errors, setErrors] = useState({}); // 유효성 에러 상태 추가

const userId = sessionStorage.getItem('userId'); // 로그인한 사용자 ID 가져오기
```

Description List

01

주문페이지의 각 STATE와 기본값 설정

O2주문

01

```
<li>
  <input
    type="radio"
    name="pay_method"
    value="신용카드"
    checked={formData.pay_method === '신용카드'}
    onChange={(e) => setFormData({ ...formData, pay_method: e.target.value })}>
  />
  <label>신용카드</label>
</li>
<li>
  <input
    type="radio"
    name="pay_method"
    value="휴대폰"
    checked={formData.pay_method === '휴대폰'}
    onChange={(e) => setFormData({ ...formData, pay_method: e.target.value })}>
  />
  <label>휴대폰</label>
</li>
```

02

```
<tr>
  <th>전화번호</th>
  <td>
    <input
      type="text"
      name="recipientPhone"
      value={formData.recipientPhone}
      onChange={handleInputChange}
      className={styles.input220}>
    />
  </td>
</tr>
```

Description List

01

결제수단 선택 및 체크박스 관련코드

02

배송정보입력 관련코드

O2주문

01

```
export const rFetchOrderInfo = (orderId, userId) => API.get(`/order/${orderId}`, {  
  headers: { 'user-id': userId }  
});
```

02

```
// useEffect로 주문 정보 가져오기  
useEffect(() => {  
  const fetchOrderInfo = async () => {  
    try {  
      const { data } = await rFetchOrderInfo(orderId, userId); // orderId로 요청  
  
      setOrderInfo(data || {});  
  
    } catch (error) {  
      setError("주문 정보를 불러오는 중 오류가 발생했습니다.");  
    } finally {  
      setIsLoading(false);  
    }  
  };  
  
  if (orderId) {  
    fetchOrderInfo();  
  }  
}, [orderId, userId]);
```

Description List

01

주문정보를 요청하는 서버에 요청하는 코드

02

주문정보를 받아오는 코드

O3 주문완료



01 장바구니 > 02 주문서작성/결제 > 03 주문완료

주문이 정상적으로 접수 되었습니다.

감사합니다.

주문정보	
4수단	신용카드
주문번호	30
주문일자	2024-12-31
주문상품	상품1번
주문자명	임세훈
배송정보	수령자: 임세훈 주소: 어딘가 핸드폰 번호: 02-579-6269
총 결제금액	777,401원

01

02

확인

Description List

01

데이터에 저장된 주문정보를 동적으로 불러옴

02

확인버튼 메인으로 이동

O3 주문완료

01

```
<tr>
    <th>주문번호</th>
    <td>{orderInfo?.order_id}</td>
</tr>
<tr>
    <th>주문일자</th>
    <td>{orderInfo?.order_date}</td>
</tr>
<tr>
    <th>주문상품</th>
    <td>{orderInfo?.product_summary}</td>
</tr>
<tr>
    <th>주문자명</th>
    <td>{orderInfo?.orderer_name}</td>
</tr>
<tr>
    <th>배송정보</th>
    <td>
        <strong>수령자:</strong> {orderInfo?.receiver_name}
        <br />
        <strong>주소:</strong> {orderInfo?.receiver_address}
        <br />
        <strong>핸드폰 번호:</strong> {orderInfo?.receiver_phone}
    </td>
</tr>
```

Description List

01

데이터에 저장된 각종 주문정보를 동적으로 받아오는 코드입니다.

○4. 회원가입 정보입력



전체상품 고객후기 고객센터

회원가입

약관동의

卓

호학인

四

정보입력

가입완료

회원가입 | 로그인



Description List

○4. 회원가입 정보입력

localhost:5500 내용:

아이디가 유효하지 않습니다. 조건을 충족하는 아이디를 입력해주세요.

확인

03

■ 아이디

■ 비밀번호

■ 비밀번호 확인

■ 이름

닉네임

■ 이메일

이메일 입력

직접 입력 ▼

■ 휴대전화

- 포함하여 입력하세요

■ 상품 구매 시 해피콜 등 고객님과 연락에 꼭 필요한 정보임으로 정확히 입력해주세요.

□ 정보/이벤트 메일 수신에 동의합니다.

■ 주소

생년월일

02 月-日-月-日 01

취소

회원가입

Description List

01 회원가입 버튼

02 메인 페이지로 이동

03 아이디 미 입력 시 또는 조건에 맞지 않게 입력 시 alert 출력

O4. 회원가입 정보입력

01

회원가입

약관동의 정보입력 가입완료

■ 아이디 아이디는 숫자와 영문 소문자 포함 최소 4자 이상 12자 이하이어야 합니다.

■ 비밀번호

02

약관동의 정보입력 가입완료

■ 아이디 이미 사용 중인 아이디입니다.

■ 비밀번호

	* id varchar(50)	* pw varchar(255)	* name varchar(100)
>	asdff1134	tnttn1234	니난뇨뇨
>	bbb123	12341234	김루피는바보
>	ccc123	tngns0328@	손흥민

Description List

01

아이디 유효성 검사

(숫자, 영문자포함 최소 4자 이상 12자 이하)

충족 못할 시 메시지 출력

02

데이터베이스내 데이터와 중복 확인. 메시지 출력

O4. 회원가입 정보입력

01

회원가입

약관동의 정보입력 가입완료

■ 아이디 사용 가능한 아이디입니다.

■ 비밀번호

■ 비밀번호 확인

■ 이름

닉네임

■ 이메일

■ 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요
 정보/이벤트 메일 수신에 동의합니다.

■ 휴대전화

Description List

01

아이디 유효성 및 중복확인 통과 시 메시지 출력

O4. 회원가입 정보입력

약관동의 정보입력 가입완료

■ 아이디 ksh1234
사용 가능한 아이디입니다.

■ 비밀번호 01

비밀번호는 영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합으로 10자 이상이어야 합니다.

■ 비밀번호 확인
비밀번호가 일치하지 않습니다.

■ 비밀번호 02
localhost:5500 내용:
비밀번호가 유효하지 않습니다. 조건을 충족하는 비밀번호를 입력해주세요.
확인

■ 비밀번호 확인

비밀번호는 영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합으로 10자 이상이어야 합니다.

■ 비밀번호 확인

비밀번호가 일치합니다.

브랜드 고객센터

Description List

01

비밀번호 유효성 검사

(영문, 숫자, 특수문자 중 2가지 이상 조합 + 10자 이상)

충족 못할 시 메시지 출력

02

유효성 충족과 비밀번호 일치 둘 중 하나라도 충족 못할 시
통과 불가 및 alert 메시지 출력

O4. 회원가입 정보입력

Description List

회원가입

약관동의 **정보입력** 가입완료

■ 아이디	ksh1234	사용 가능한 아이디입니다.
■ 비밀번호	•••••••• 01	비밀번호가 유효합니다.
■ 비밀번호 확인	••••••••	비밀번호가 일치합니다.
■ 이름		
닉네임		
■ 이메일	이메일 입력	직접 입력 ▾
<ul style="list-style-type: none">■ 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요<input type="checkbox"/> 정보/이벤트 메일 수신에 동의합니다.		

01

유효성 및 비밀번호 일치 시 확인메시지 출력

O4. 회원가입 정보입력

01

localhost:5500 내용:

이름이 유효하지 않습니다.

확인

브랜드

고객센터

■ 비밀번호	*****	비밀번호가 유효합니다.
■ 비밀번호 확인	*****	비밀번호가 일치합니다.
■ 이름	@@@	

Description List

01

이름 유효성 검사

(한글만 최소 2자 이상 6자 이하. 공백 불가)

충족 못할 시 alert 메시지 출력

O4. 회원가입 정보입력

01

■ 이름

김수훈

닉네임

ㅎビズヒ 뽕r己r뽕I己|뽀™

닉네임은 한글, 영문, 숫자 포함 2자 이상 10자 이하로 입력해주세요.

02

■ 이름

김수훈

닉네임

강아지

* id varchar(50)	* pw varchar(255)	* name varchar(100)	nick varchar(50)
asdff1134	tnttn1234	니난뇨뇨	후루루루
bbb123	12341234	김루피는바보	강아지
ccc123	tngns0328@	손흥민	홍민소니

이미 사용 중인 닉네임입니다.

03

■ 이름

김수훈

닉네임

훈수김

사용 가능한 닉네임입니다.

Description List

01

닉네임 유효성 검사

(한글, 영문, 숫자 포함 2글자 이상 10자 이하. 특수문자 불
충족 못할 시 메시지 출력)

02

데이터베이스내 데이터와 중복 확인. 메시지 출력

03

유효성 충족 및 중복 없을 시 사용 가능 메시지 출력

O4. 회원가입 정보입력

■ 이메일

01 가나다1234@gmail.com 직접 입력 ▾

유효한 이메일 형식이 아닙니다.

- 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요
- 정보/이벤트 메일 수신에 동의합니다.

02 ksh1234@gmail.com 직접 입력 ▾

유효한 이메일 형식입니다.

- 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요
- 정보/이벤트 메일 수신에 동의합니다.

03 ksh1234@naver.com 직접 입력 ▾

유효한 이메일 형식입니다.

- 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요
- 정보/이벤트 메일 수신에 동의합니다.

■ 휴대전화

- 포함하여 입력하세요

- 상품 구매 시 해피콜 등 고객님과 연락에 꼭 필요한 정보를 제공해주세요.
- 정보/이벤트 SMS 수신에 동의합니다.

Description List

01

이메일 유효성 검사

로컬파트(영문자, 숫자, ._%+- 만 가능)

@ 도메인(영문자와 숫자, .- 만 가능).

최상위도메인(영문자만 허용, 최소 2글자 이상)

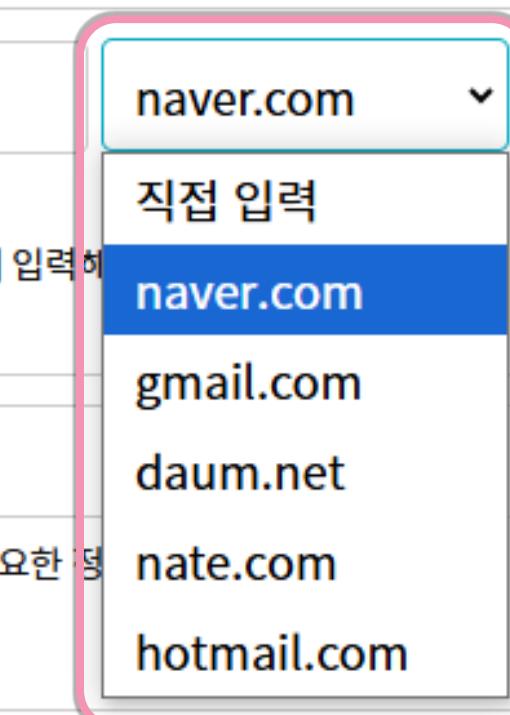
충족 못할 시 메시지 출력

02

유효성 충족 시 메시지 출력

03

Select의 도메인 선택 시 자동으로 이메일 아이디 뒤에 붙



○4. 회원가입 정보입력

01

localhost:5500 내용:

휴대전화 번호 형식이 유효하지 않습니다. xxx-xxxx-xxxx 형태로 입력해주세요.

고객센터

확인

■ 이름

김수훈

■ 닉네임

훈수김

사용 가능한 닉네임입니다.

■ 이메일

ksh1234@naver.com

naver.com ▾

유효한 이메일 형식입니다.

- 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요
- 정보/이벤트 메일 수신에 동의합니다.

■ 휴대전화

01012341234

- 상품 구매 시 해피콜 등 고객님과 연락에 꼭 필요한 정보임으로 정확히 입력해주세요.
- 정보/이벤트 SMS 수신에 동의합니다.

Description List

01

휴대전화 유효성 검사

(숫자만 허용. xxx-xxxx-xxxx의 형태만 허용)

충족 못할 시 alert 메시지 출력

O4. 회원가입 정보입력

01 **브랜드**

localhost:5500 내용:
주소는 한글, 숫자, 영문자, '-'만 허용하며, 최소 8자 이상이어야 합니다.

고객센터

확인

■ 이름

닉네임
사용 가능한 닉네임입니다.

■ 이메일 ▼
유효한 이메일 형식입니다.
■ 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요
 정보/이벤트 메일 수신에 동의합니다.

■ 휴대전화
■ 상품 구매 시 해피콜 등 고객님과 연락에 꼭 필요한 정보임으로 정확히 입력해주세요.
 정보/이벤트 SMS 수신에 동의합니다.

■ 주소

Description List

01

주소 유효성 검사

(한글, 숫자, 영문자, 공백, 특수문자 '-'만 허용, 최소 8자)
충족 못할 시 alert 메시지 출력

O4. 회원가입 정보입력

■ 이름	김수훈
닉네임	훈수김 <small>사용 가능한 닉네임입니다.</small>
■ 이메일	lsh1234@naver.com naver.com
■ 휴대전화	<small>꼭 필요합니다. 정확히 입력해주세요</small>
■ 주소	<small>꼭 필요한 정보입니다. 정확히 입력해주세요</small>
생년월일	1993-04-19 <small>삭제 오늘</small>
취소 회원가입	

Description List

01

생년월일은 선택사항

달력모양 클릭 시 달력 출력 및 생년월일 선택

O4. 회원가입 정보입력

localhost:5500 내용:
회원가입 성공!

브랜드

고객센터

확인

■ 이름
김수훈

■ 닉네임
훈수김
사용 가능한 닉네임입니다.

■ 이메일
ksh1234@naver.com naver.com ▾
유효한 이메일 형식입니다.
■ 아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요
 정보/이벤트 메일 수신에 동의합니다.

■ 휴대전화
010-1234-1234
■ 상품 구매 시 해피콜 등 고객님과 연락에 꼭 필요한 정보임으로 정확히 입력해주세요
 정보/이벤트 SMS 수신에 동의합니다.

■ 주소
서울특별시 서초구 강남대로 331

생년월일
1993-04-19

01

취소

회원가입

Description List

01

모든 정보를 조건에 맞게 입력 후 회원가입 버튼 클릭 시
회원가입 성공이라는 alert 메시지 출력

02

데이터베이스에 입력한 회원정보 저장

	* id varchar(50)	* pw varchar(255)	* name varchar(100)	nick varchar(50)
>	ksh1234	tngns1234@@	김수훈	훈수김

	birth_date date	phone_num varchar(20)	email varchar(100)	addr varchar(255)
	1993-04-19	010-1234-1234	ksh1234@naver.com	서울특별시 서초구 강남

O5. 회원가입 완료

전체상품

고객후기

고객센터

01

회원가입완료

약관동의

정보입력

가입완료



회원가입이 완료되었습니다.

02

김수훈님의 회원가입을 축하합니다.
알차고 실속있는 서비스를 찾아뵙겠습니다.

03

홈으로

로그인

04

Description List

01

회원가입 완료 페이지

02

가입한 회원 이름과 함께 축하 메시지 출력

03

메인 페이지로 이동

04

로그인 페이지로 이동

○6. 회원가입 안내 및 혜택



전체상품 고객후기 고객센터

이로케 회원만의 혜택

회원가입 후 이로케에서 드리는 특별한 혜택을 놓치지 마세요!



신규 회원
가입 시

M 10,000점 지급



이메일/SMS
수신 동의 시

M 최대 10,000점 지급

※ 마일리지는 최초 회원가입 시 수신 동의하셔야 지급 됩니다.

※ 마일리지는 장바구니의 결제 금액이 500,000원 이상일 경우, 결제 건별로 1회 결제 시 최대 40,000점을 사용 가능합니다.

02

쇼핑몰 회원가입

이미 쇼핑몰 회원이세요? [로그인](#)

Description

01

회원가입 클릭 시 회원가입 안내 페이지로 이동

02

쇼핑몰 회원가입 버튼(클릭 시 약관동의 페이지로 이동)

○7. 회원가입 이용약관

Description List

회원가입

01

약관동의

정보입력

가입완료

이로케의 모든 약관을 확인하고 전체 동의함 (전체동의는 선택항목도 포함됩니다)

(필수) 이용약관 동의

제1조(목적)

이 약관은 (주)자코모 회사(전자거래 사업자)가 운영하는 자코모 사이버 몰(이하 "몰"이라 한다)에서 제공하는 인터넷 관련 서비스(이하 "서비스"라 한다)를 이용함에 있어 사이버몰과 이용자의 권리·의무 및 책임사항을 규정함을 목적으로 합니다.

※ 「PC통신등을 이용하는 전자거래에 대해서도 그 성질에 반하지 않는한 이 약관을 준용합니다.」

제2조(정의)

① "몰"이란 (주)자코모 회사가 재화 또는 용역을 이용자에게 제공하기 위하여 컴퓨터등 정보통신설비를 이용하여 재화 또는 용역을 거래할 수 있도록 설정한 가상의 영업장을 말하며, 아울러 사이버물을 운영하는 사업자의 의미로도 사용합니다.

② "이용자"란 "몰"에 접속하여 이 약관에 따라 "몰"이 제공하는 서비스를 받는 회원 및 비회원을 말합니다.

③ '회원'이라 함은 "몰"에 개인정보를 제공하여 회원등록을 한 자로서, "몰"의 정보를 지속적으로 제공받으며, "몰"이 제공하는 서비스를 계속적으로 이용할 수 있는 자를 말합니다.

④ '비회원'이라 함은 회원에 가입하지 않고 "몰"이 제공하는 서비스를 이용하는 자를 말합니다.

(필수) 개인정보 수집 및 이용 동의

개인정보 수집 및 이용은 회원가입, 고객상담, 본인여부 확인, 주문/결제 및 배송 서비스 이용에 있다.
각 이용 목적과 그 항목, 기간은 아래로 정의된다.

- 수집 및 이용 목적 : 이용자 식별 및 본인여부 확인

- 수집 항목 : 이름, 아이디, 비밀번호

- 보유 및 이용기간 : 회원탈퇴 후 5일까지

- 수집 및 이용 목적 : 민원 등 고객 고충처리

- 수집 항목 : 이메일, 휴대폰번호

- 보유 및 이용기간 : 회원 탈퇴 후 5일까지

- 수집 및 이용 목적: 주문/결제 시 상품 배송

- 수집 항목: 구매자 정보, 상품 구매/취소/반품/교환/환불 정보, 수령인 정보

- 보유 및 이용기간: 회원 탈퇴 후 5일까지

- 소지 미 이요 모저 · 마 14 세 미마 아도 화이

(선택) 마케팅 정보 수신 및 활용 동의

(주)자코모에서 제공하는 이벤트 및 혜택 등 다양한 정보를 문자, 이메일 등으로 받아보실 수 있습니다.

마케팅 정보 수신 및 활용 동의 여부와 관계없이 회원가입 및 서비스를 이용하실 수 있습니다.

또한 서비스의 중요 안내 사항 및 주문/배송에 대한 정보는 마케팅 정보 수신 동의 여부와 관계없이 발송됩니다.

1. 수집 및 이용 목적

- 인구 통계학적 특성에 따른 서비스 제공 및 광고 게재

- 필수 이용 및 반송 빙드 페인 서비스 이용에 대한 통계 등 업무와 관련된 각종 통계자료의 작성 및 서비스 개발

- 01 전체 동의 체크박스 클릭 시 전체 약관 동의 체크
 - 02 개별 체크도 가능

○7. 회원가입 이용약관

(필수) 개인정보 수집 및 이용 동의

개인정보 수집 및 이용은 회원가입, 고객상담, 본인여부 확인, 주문/결제 및 배송 서비스 이용에 있다.
각 이용 목적과 그 항목, 기간은 아래로 정의된다.

- 수집 및 이용 목적 : 이용자 식별 및 본인여부 확인
- 수집 항목 : 이름, 아이디, 비밀번호
- 보유 및 이용기간 : 회원탈퇴 후 5일까지
- 수집 및 이용 목적 : 만원 등 고객 고충처리
- 수집 항목 : 이메일, 휴대폰번호
- 보유 및 이용기간 : 회원 탈퇴 후 5일까지
- 수집 및 이용 목적: 주문/결제 시 상품 배송
- 수집 항목: 구매자 정보, 상품 구매/취소/반품/교환/환불 정보, 수령인 정보
- 보유 및 이용기간: 회원 탈퇴 후 5일까지

스저 미 이요 모저 · 마 14세 미마 아도 화이

(선택) 마케팅 정보 수신 및 활용 동의

(주)자코모에서 제공하는 이벤트 및 혜택 등 다양한 정보를 문자, 이메일 등으로 받아보실 수 있습니다.
마케팅 정보 수신 및 활용 동의 여부와 관계없이 회원가입 및 서비스를 이용하실 수 있습니다.
또한 서비스의 중요 안내 사항 및 주문/배송에 대한 정보는 마케팅 정보 수신 동의 여부와 관계없이 발송됩니다.

1. 수집 및 이용 목적

- 인구 통계학적 특성에 따른 서비스 제공 및 광고 게재
- 접속, 이용 및 방문 빈도 파악, 서비스 이용에 대한 통계 등 업무와 관련된 각종 통계 자료의 작성 및 서비스 개발
- 신규 서비스 개발과 이벤트 행사에 따른 정보 전달 및 맞춤 서비스 제공
- 제품 안내, 홍보 및 이벤트 등의 마케팅, 광고성 정보 전달

2. 수집 및 이용 항목

회원 정보(이름, 휴대폰번호, 이메일, 성별, 생년월일, 가입일시)
매 및 배송 서비스 이용 정보(장바구니 상품, 구매일시, 주소, 받으시는 곳)

01 수집 및 이용 항목은 마케팅 및 분석, 프로모션의 목적에 따라 달라질 수 있으며 수집 시점에 동의 여부를 별도 확인함

보우 미 이요 기가

I (필수) 약관에 모두 동의해주세요.

02

03

이전단계

다음단계

Description List

01

필수 사항 미 체크 시 메시지 출력

02

회원가입 안내 페이지로 이동

03

회원가입 정보입력으로 이동

O8. 회원가입 핵심 코드

01

```
// 회원가입 처리 라우터
router.post('/join', async (req, res) => {
  const { id, pw, name, nick, birth_date, phone_num, email, addr } = req.body;

  // 필수 항목이 누락된 경우 클라이언트에 에러 응답 반환
  if (!id || !pw || !name || !email || !phone_num || !addr) {
    return res.status(400).json({ success: false, message: '필수 항목이 누락되었습니다.' });
  }

  try {
    // 회원 정보를 mem_info 테이블에 삽입하기 위한 SQL 쿼리
    const sql = `
      INSERT INTO mem_info (id, pw, name, nick, birth_date, phone_num, email, addr, reg_id)
      VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
    `;
    // SQL 쿼리에 바인딩될 데이터 배열
    const data = [
      id,
      pw,
      name,
      nick || null,
      birth_date || null,
      phone_num,
      email,
      addr,
      id,
    ];

    const [result] = await conn.execute(sql, data);

    res.json({ success: true, message: '회원가입이 완료되었습니다.' });
  } catch (error) {
    console.error('회원가입 오류:', error);
    res.status(500).json({ success: false, message: '회원가입 중 오류가 발생했습니다.' });
  }
});

module.exports = router;
```

Description List

01

회원가입 요청 처리 및 데이터베이스에 회원 정보를 저장
회원가입 처리 라우터

○8. 회원가입 핵심 코드

01

```
const handleSubmit = async (e) => {
  e.preventDefault();

  // 아이디 유효성 검사 실패 시 제출 방지
  if (!idValid) {
    alert("아이디가 유효하지 않습니다. 조건을 충족하는 아이디를 입력해주세요.");
    return;
  }

  // 아이디 중복 확인 로직
  if (idDuplicate) {
    alert("이미 사용 중인 아이디입니다. 다른 아이디를 입력해주세요.");
    return;
  }

  // 비밀번호 확인 로직
  if (formData.pw !== formData.pw2) {
    alert("비밀번호가 일치하지 않습니다.");
    return;
  }

  // 비밀번호 유효성 검사
  if (!pwValid) {
    alert("비밀번호가 유효하지 않습니다. 조건을 충족하는 비밀번호를 입력해주세요.");
    return;
  }

  // 이름 유효성 검사
  if (!nameValid) {
    alert("이름이 유효하지 않습니다.");
    return;
  }
```

Description List

01

각 유효성 검사 및 중복 확인을 종합하여 검증 완료 되었을 때 데이터를 서버로 전송하는 함수

```
// 서버로 보낼 데이터 준비 (pw2 제외)
const dataToSend = {
  id: formData.id,
  pw: formData.pw,
  name: formData.name,
  nick: formData.nick,
  email: formData.email,
  phone_num: formData.phone_num,
  addr: formData.addr,
  birth_date: formData.birth_date,
};

try {
  const response = await memJoin(dataToSend);
  if (response.data.success) {
    alert("회원가입 성공!");
    navigate(`joinFin?name=${encodeURIComponent(formData.name)}`);
  } else {
    alert("회원가입 실패: " + response.data.message);
  }
} catch (error) {
  alert("회원가입 중 오류가 발생했습니다.");
}
```

O8. 회원가입 핵심 코드

```
const express = require("express");
const router = express.Router();
const conn = require("../utils/dbUtil");

// 아이디 중복 확인 라우터
router.post("/", async (req, res) => {
  const { id } = req.body;

  if (!id) {
    return res.status(400).json({ message: "아이디를 입력해주세요." });
  }

  try {
    const [results] = await conn.query(`SELECT COUNT(*) AS count FROM mem_info WHERE id = ?`, [id]); // 아이디 중복 확인 함수
    const isDuplicate = results[0].count > 0;

    res.json({ isDuplicate });
  } catch (error) {
    res.status(500).json({ message: "서버 오류 발생" });
  }
});

module.exports = router;
```

01

02

03

```
const checkIdDuplicate = async (id) => {
  if (!id) {
    setIdMsg("");
    setIdDuplicate(false);
    return;
  }

  // 유효성 검사 실패 시 중복 확인 생략
  if (!validId(id)) {
    return;
  }

  try {
    const response = await checkIdDup(id);
    if (response.data.isDuplicate) {
      setIdMsg("이미 사용 중인 아이디입니다.");
      setIdDuplicate(true);
    } else {
      setIdMsg("사용 가능한 아이디입니다.");
      setIdDuplicate(false);
    }
  } catch (error) {
    setIdMsg("아이디 중복 확인 중 오류가 발생했습니다.");
    setIdDuplicate(true);
  }
};
```

Description List

01

아이디 중복 확인 요청 라우터

02

아이디 중복 확인 요청을 처리하는 라우터 등록

03

아이디 중복확인을 위한 API 호출

04

아이디 중복 확인 함수

O9. 아이디 찾기

아이디 찾기

가입 시 입력한 정보를 정확히 입력해 주시기 바랍니다.

01. 아이디 찾기 버튼

02. 회원 가입 정보 입력의 이메일과 같은 형식으로 Select의 도메인 선택 시 자동으로 도메인이 붙음

03. 비밀번호 찾기 페이지로 이동

04. 로그인 페이지로 이동

Description List

01. 아이디 찾기 버튼

이름과 이메일 주소가 가입된 정보와 일치 시 찾기 가능

02. 회원 가입 정보 입력의 이메일과 같은 형식으로

Select의 도메인 선택 시 자동으로 도메인이 붙음

03. 비밀번호 찾기 페이지로 이동

04. 로그인 페이지로 이동

O9. 아이디 찾기

01

localhost:5500 내용:
등록된 정보가 없습니다.

확인

Description List

01

이름과 이메일 주소가 가입된 정보와 일치 하지 않을 시
아이디 찾기를 눌렀을 때 alert 메시지 출력

Q	* id varchar(50)	* pw varchar(255)	* name varchar(100)	nick varchar(50)	birth_date date	phone_num varchar(20)	email varchar(100)
>	ksh1234	tngns1234@@	김수훈	훈수김	1993-04-19	010-1234-1234	ksh1234@naver.com

아이디 찾기

가입 시 입력한 정보를 정확히 입력해 주시기 바랍니다.

김훈수

ksh1234@naver.com

naver.com

아이디 찾기

비밀번호 찾기

로그인하기

O9. 아이디 찾기

Description List

아이디 찾기

가입 시 입력한 정보를 정확히 입력해 주시기 바랍니다.

김수훈	
ksh1234@naver.com	naver.com

아이디 찾기

Q	* id varchar(50)	* pw varchar(255)	* name varchar(100)	nick varchar(50)	birth_date date	phone_num varchar(20)	email varchar(100)
> ksh1234 tngns1234@@ 김수훈 훈수김 1993-04-19 010-1234-1234 ksh1234@naver.com							

아이디 찾기

01

회원님의 아이디는
ksh1234 입니다.

02

비밀번호 찾기

로그인하기

03

01 이름과 이메일이 가입 정보와 일치 시 해당하는 아이디가 출력됨

02 비밀번호 찾기 페이지로 이동

03 로그인 페이지로 이동

10. 아이디 찾기 핵심코드

```
router.post('/', async (req, res) => {
  const { name, email } = req.body; // 회원으로부터 이름과 이메일을 추출

  try {
    // 입력받은 이름과 이메일을 기반으로 데이터베이스에서 회원 정보를 조회
    const sql = 'SELECT id FROM mem_info WHERE name = ? AND email = ?';
    const [rows] = await conn.execute(sql, [name, email]);

    // 조회된 결과가 없는 경우
    if (rows.length === 0) {
      return res.status(404).json({ success: false, message: '등록된 정보가 없습니다.' });
    }
    // 조회 성공 시 아이디와 이름을 클라이언트에 응답
    res.json({ success: true, id: rows[0].id, name: rows[0].name });
  } catch (error) {
    console.error('아이디 찾기 오류:', error);
    res.status(500).json({ success: false, message: '서버 오류가 발생했습니다.' });
  }
});

module.exports = router;
```

```
const FindId = () => {
  const [name, setName] = useState(''); // 사용자 이름 상태
  const [email, setEmail] = useState(''); // 사용자 이메일 상태
  const navigate = useNavigate();

  // 아이디 찾기 처리 함수
  const handleFindId = async () => {
    try {
      const response = await findId(name, email); // 서버로 아이디 찾기 요청 전송
      navigate(`/findId2/${response.data.id}`); // find_id2.js로 이동하며 찾은 아이디 전달
    } catch (error) {
      alert(error.response?.data?.message || '서버 오류');
    }
  };
}
```

Description List

01

- 01 사용자 이름과 이메일을 기반으로 등록된 아이디를 반환하는
아이디 찾기 라우터

02

- 02 아이디 찾기 요청 처리 함수

10. 아이디 찾기 핵심코드

Description List

01

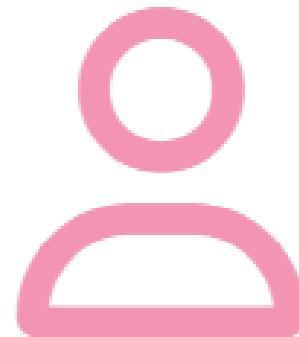
이메일 도메인 선택 시 기존 입력된 로컬 파트와 결합하여
자동으로 이메일 주소를 완성

01

```
<select  
  className={styles.emailDomain}  
  onChange={(e) => setEmail((prevEmail) => `${prevEmail.split('@')[0]}@${e.target.value}`)}  
>  
  <option value="">직접입력</option>  
  <option value="naver.com">naver.com</option>  
  <option value="daum.net">daum.net</option>  
  <option value="nate.com">nate.com</option>  
  <option value="hotmail.com">hotmail.com</option>  
  <option value="gmail.com">gmail.com</option>  
</select>  
...
```

11. 비밀번호 찾기

비밀번호 찾기



회원님의 이름과 아이디를 입력해 주세요.

01 [다음](#)

아이디를 모르시나요? [아이디 찾기](#)

02

Description List

- 01 이름과 아이디가 가입 정보와 일치 시 비밀번호 재설정 페이지로 이동
- 02 아이디 찾기 페이지로 이동

11. 비밀번호 찾기

비밀번호 찾기



회원님의 이름과 아이디를 입력해 주세요.

김훈수	* id varchar(50) ksh1234	* pw varchar(255) tngns1234@@	* name varchar(100) 김수훈
-----	--------------------------------	-------------------------------------	-------------------------------

01

회원정보가 일치하지 않습니다.

다음

아이디를 모르시나요? 아이디 찾기

Description List

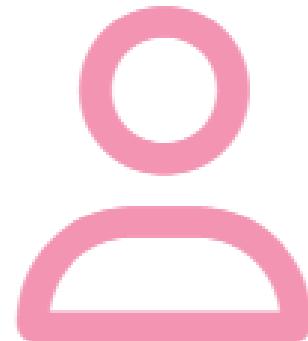
01

이름과 아이디가 가입 정보와 일치 하지 않을 시
메시지 출력 및 비밀번호 재설정 페이지로 이동 불가

11. 비밀번호 찾기

01

비밀번호 재설정



새로운 비밀번호를 등록해 주세요.

02

새 비밀번호 입력
새 비밀번호 확인

03

확인

Description List

01

이름과 아이디가 가입 정보와 일치 시 비밀번호 재설정 페이지로 이동

02

새 비밀번호 입력 및 확인
두 비밀번호가 일치해야 하며 유효성은 기존 회원가입과
(영문, 숫자, 특수문자 중 2가지 이상 조합 + 10자 이상)

03

확인 버튼
유효성 통과 및 비밀번호 일치 해야함

11. 비밀번호 찾기

01 비밀번호는 영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합으로 10자 이상이어야 합니다.

확인

02 비밀번호가 일치하지 않습니다.

확인

Description List

01

새 비밀번호가 유효성에 충족하지 못할 시 메시지 출력
(영문, 숫자, 특수문자 중 2가지 이상 조합 + 10자 이상)

02

새 비밀번호가 유효성은 충족해도 일치하지 않을 시
메시지 출력

11. 비밀번호 찾기

Description List

01 유효성 통과 및 새 비밀번호 일치 시 비밀번호 변경 완료

01

비밀번호 변경 완료



비밀번호가 성공적으로 변경되었습니다.

02 로그인 페이지로 이동

01

02

로그인

02

	* id varchar(50)	* pw varchar(255)	* name varchar(100)
>	ksh1234	tngns1234@@	김수훈

	* id varchar(50)	* pw varchar(255)	* name varchar(100)
>	ksh1234	tngns4321!!	김수훈

12. 비밀번호 찾기 핵심코드

```
// 비밀번호 재설정 라우트
router.post('/', async (req, res) => {
  const { userId, userPw } = req.body;

  // 비밀번호 유효성 검사 (숫자와 영문자를 모두 포함한 10글자 이상)
  const pwRegex = /^(?=.*[a-zA-Z])(?=.*[0-9]).{10,}$/;

  if (!pwRegex.test(userPw)) {
    return res.status(400).json({ success: false, message: '비밀번호는 숫자와 영문자를 모두 포함해야 합니다.' });
  }

  try {
    // 비밀번호 업데이트 쿼리
    const sql = 'UPDATE mem_info SET pw = ? WHERE id = ?';
    const [result] = await conn.execute(sql, [userPw, userId]);

    // 업데이트된 행이 없으면 존재하지 않는 사용자 ID로 판단
    if (result.affectedRows === 0) {
      return res.status(404).json({ success: false, message: '존재하지 않는 아이디입니다.' });
    }

    res.json({ success: true, message: '비밀번호가 성공적으로 변경되었습니다.' });
  } catch (error) {
    console.error('비밀번호 변경 오류:', error);
    res.status(500).json({ success: false, message: '서버 오류가 발생했습니다.' });
  }
});

module.exports = router;
```

Description List

O1

O1

사용자가 입력한 새로운 비밀번호를 확인 후 해당 아이디
비밀번호를 재설정하는 API 라우터

12. 비밀번호 찾기 핵심코드

01

```
const FindPwReset = ({ userId }) => {
  //const { userId } = useParams(); // URL 파라미터에서 userId 가져오기
  const [userPw, setUserPw] = useState(''); // 새 비밀번호 입력 상태
  const [confirmPw, setConfirmPw] = useState(''); // 새 비밀번호 확인 상태
  const [errorMsg, setErrorMsg] = useState('');
  const navigate = useNavigate();

  const handleSubmit = async () => {
    // 비밀번호 유효성 검사 (영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합 10자 이상)
    const pwRegex = /^(?!.*\s)(?=.*[a-zA-Z])(?=.*\d|.*[@#$%^&*])[a-zA-Z\d!@#$%^&*]{10,}$/; // 영문

    if (!pwRegex.test(userPw)) {
      setErrorMsg('비밀번호는 영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합으로 10자 이상이어야 합니다.');
      return;
    }

    if (userPw !== confirmPw) {
      setErrorMsg('비밀번호가 일치하지 않습니다.');
      return;
    }

    try {
      // 백엔드에 비밀번호 변경 요청
      const response = await findPwReset(userId, userPw); // `findPwReset`은 API 호출 함수
      if (response.data.success) {
        alert('비밀번호가 성공적으로 변경되었습니다.');
        navigate('/findPwComplete'); // 성공 시 완료 페이지로 이동
      } else {
        setErrorMsg(response.data.message || '비밀번호 변경에 실패했습니다.');
      }
    } catch (error) {
      console.error('비밀번호 변경 실패:', error.response?.data?.message || '서버 오류');
      setErrorMsg(error.response?.data?.message || '서버 오류');
    }
  };

  const fields = [
    { type: 'password', id: 'pw_input', placeholder: '새 비밀번호 입력', value: userPw,
      onChange: (e) => setUserPw(e.target.value), required: true },
    { type: 'password', id: 'pw_input_chk', placeholder: '새 비밀번호 확인', value: confirmPw,
      onChange: (e) => setConfirmPw(e.target.value), required: true },
  ];
}
```

Description List

01

새 비밀번호를 입력, 비밀번호 변경 요청을 보내는
비밀번호 재설정 화면 컴포넌트

13. 비밀번호 찾기 공통 컴포넌트화

01

```
import styles from './inputForm.module.css'

const InputForm = ({ fields, errorMessage, buttonText, onSubmit, className = '' }) => {
  return (
    <div className={`${styles.inputForm} ${className}`}>
      {fields.map((field, index) => (
        <div key={index} className={styles.inputField}>
          <input
            type={field.type}
            id={field.id}
            placeholder={field.placeholder}
            value={field.value}
            onChange={field.onChange}
            required={field.required || false}
          />
          {field.error && <p className={styles.fieldError}>{field.error}</p>}
        </div>
      ))}
      {errorMessage && <p className={styles.formError}>{errorMessage}</p>}
      <div className={styles.inputBtnDiv}>
        <button type="button" className={styles.submitButton} onClick={onSubmit}>
          {buttonText}
        </button>
      </div>
    </div>
  );
}

export default InputForm;
```

Description List

01

비밀번호 찾기에서 공통으로 쓰이거나 품을 컴포넌트화

```
<section>
<div className={styles.findPwContainer}>
  <div className={styles.findPwTitle}>비밀번호 찾기</div>

  <div className={styles.findMsgDiv}>
    <i id="mypage_i" className={`fa-regular fa-user ${style}`}>
    <p>회원님의 이름과 아이디를 입력해 주세요.</p>
  </div>

  <InputForm
    fields={fields}
    errorMessage={errorMessage}
    buttonText="다음"
    onSubmit={handleNextClick}
  />
```

```
<section>
<div className={styles.resetContainer}>
  <div className={styles.resetTitle}>비밀번호 재설정</div>

  <div className={styles.resetPwMsg}>
    <i id="mypage_i" className={`fa-regular fa-user ${style}`}>
    <p>새로운 비밀번호를 등록해 주세요.</p>
  </div>

  <InputForm
    fields={fields}
    errorMessage={errorMsg}
    buttonText="확인"
    onSubmit={handleSubmit}
  />
</div>
```

14. 로그인

Description List

The screenshot shows the ILIKE login page with the following elements:

- Header:** ILIKE logo, navigation links (전체상품, 고객후기, 고객센터), account status (0원), and member icons.
- Header Buttons:** 회원가입 (pink) and 로그인 (white).
- Member Login Section (01):** Contains input fields for 아이디 and 비밀번호, a large blue "로그인" button, and smaller buttons for 회원가입, 아이디 찾기, and 비밀번호 찾기.
- SNS Convenient Login Section:** Three colored buttons for PAYCO 아이디 로그인 (red), 네이버 아이디 로그인 (green), and 카카오 아이디 로그인 (yellow).

01 로그인 페이지

14. 로그인

회원 로그인

The diagram illustrates a login form with the following components:

- 아이디** (ID) input field
- 비밀번호** (Password) input field
- 로그인** (Log In) button (highlighted with a red border)
- 회원가입** (Join) link (highlighted with a black background)
- 아이디 찾기** (Find ID) link
- 비밀번호 찾기** (Find Password) link

Below the form, four numbered circles point to specific elements:

- 01: **로그인** 버튼 (Log In button)
- 02: **회원가입** 페이지로 이동 (Move to Join page)
- 03: **아이디 찾기** 페이지로 이동 (Move to Find ID page)
- 04: **비밀번호 찾기** 페이지로 이동 (Move to Find Password page)

SNS 간편 로그인

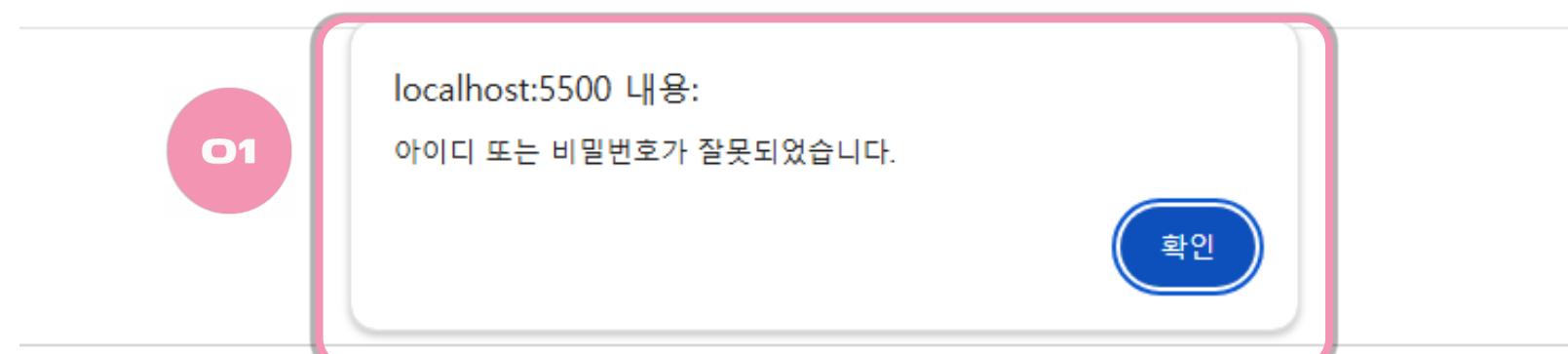
The diagram shows three SNS login buttons:

- PAYCO 아이디 로그인** (PAYCO ID Login) - Red button
- 네이버 아이디 로그인** (Naver ID Login) - Green button
- 카카오 아이디 로그인** (Kakao ID Login) - Yellow button

Description List

- 01** 로그인 버튼
- 02** 회원가입 페이지로 이동
- 03** 아이디 찾기 페이지로 이동
- 04** 비밀번호 찾기 페이지로 이동

14. 로그인



Description List

- 01 잘못된 아이디와 비밀번호 입력 시 alert 메시지 출력.

회원 로그인

A screenshot of a member login form. It includes two input fields: one with placeholder 'ksh1234' and another with placeholder '.....'. Below these is a teal '로그인' (Login) button. At the bottom are three buttons: '회원가입' (Sign Up), '아이디 찾기' (Find ID), and '비밀번호 찾기' (Find Password).

SNS 간편 로그인



14. 로그인



회원 로그인

ksh1234
.....

로그인

회원가입 아이디 찾기 비밀번호 찾기

SNS 간편 로그인

PAYCO 아이디 로그인
네이버 아이디 로그인
카카오 아이디 로그인

01

localhost:5500 내용:
김수훈님, 환영합니다!

확인

02

Origin http://localhost:5500

Key	Value
userId	ksh1234
userName	김수훈
auth	true

Events Console Application > ▲ 6 🔍 1 ⌂ : X

http://localhost:5500

Origin http://localhost:5500

Key	Value
userId	ksh1234
userName	김수훈
auth	true

Local storage Session storage IndexedDB Cookies Private state tokens Interest groups Shared storage Cache storage Storage buckets Background services Back/forward cache

Description List

01

아이디와 비밀번호 일치 시 회원 이름과 함께
환영 메시지 출력 후 로그인
메인 페이지로 이동

02

로그인과 동시에 세션 스토리지에 회원 아이디, 회원이름
저장

15. 회원 마일리지 조회

The image consists of three vertically stacked screenshots of a mobile application interface. The top two screenshots show the main header bar with '전체상품' (All Products), '고객후기' (Customer Reviews), and '고객센터' (Customer Service). The top screenshot shows a pink circular callout '01' pointing to a user icon with a plus sign and a shopping cart icon. The middle screenshot shows a pink circular callout '02' pointing to the same icons. The bottom screenshot shows a pink circular callout '03' pointing to a box containing a coin icon and the text '마일리지 7,000점'. The header bar also includes the text '김수훈님, 환영합니다.' (Welcome, Mr. Kim) and a '로그아웃' (Logout) button. Below the header, there is a banner for 'ILOKE' and a section for '진행 중인 주문' (Pending Orders) showing four items: 배송준비 (Delivery Preparation), 배송중 (In Delivery), 배송완료 (Delivery Complete), and 구매확정 (Purchase Confirmation). The pending order section also displays the number of items: 0건 (0 items).

Description List

- 01 사람 아이콘 클릭 시 마이 페이지로 이동
- 02 마이 페이지
- 03 점수 클릭 시 마일리지 내역 출력

15. 회원 마일리지 조회

Description List



01 회원 마일리지 내역 출력

(적립 날짜, 적립 내용, 유효 날짜, 변동 내역)

02 돌아가기 버튼

클릭 시 기존 배송 정보 출력

01

마일리지 내역

적립 날짜	적립 내용	유효 날짜	변동 내역
2024-12-31T10:57:42.000Z	마일리지 사용 - 소파	2025-12-30T15:00:00.000Z	-2,000점
2024-12-18T07:16:46.000Z	구매 적립 - 가죽소파	2025-12-17T15:00:00.000Z	1,000점
2024-11-29T05:54:27.000Z	구매 적립 - 소파	2025-11-28T15:00:00.000Z	500점
2024-11-29T05:54:27.000Z	구매 적립 - 의자	2025-11-28T15:00:00.000Z	200점
2024-11-29T05:54:27.000Z	마일리지 사용 - 의자	2025-11-28T15:00:00.000Z	-300점

02

돌아가기

16. 회원 탈퇴

회원탈퇴

01. 회원탈퇴 안내

01

이로케 탈퇴 안내
회원님께서 회원 탈퇴를 원하신다니 저희 쇼핑몰의 서비스가 많이 부족하고 미흡했나 봅니다.
그동안 저희 이로케를 이용해주셔서 감사드립니다.

■ 아울러 회원 탈퇴 시의 아래 사항을 숙지하시기 바랍니다.
회원 탈퇴를 신청하시면, 보유하신 마일리지 및 회원 정보는 삭제됩니다.
탈퇴 신청 후에는 복구가 불가능하므로 신중히 진행해 주시기 바랍니다.

02. 회원탈퇴 하기

02

비밀번호

비밀번호 입력

이전으로

탈퇴

03

04

Description List

01

회원 탈퇴 안내문

02

비밀번호 입력

03

이전으로 버튼(마이 페이지로 이동)

04

회원 탈퇴 버튼(비밀번호 일치 시 회원 탈퇴)

16. 회원 탈퇴

01

localhost:5500 내용:
정말로 탈퇴하시겠습니까? 탈퇴하면 복구할 수 없습니다.

확인 취소

02. 회원탈퇴 하기

비밀번호

이전으로

탈퇴

02

localhost:5500 내용:
회원탈퇴가 완료되었습니다.

확인

02. 회원탈퇴 하기

비밀번호

이전으로

탈퇴

Description List

01 정말로 탈퇴 하실지 alert 메시지 출력

02 회원 탈퇴 완료

(확인 누를 시 자동 로그아웃 및 메인 페이지 이동)

16. 회원 탈퇴

01

localhost:5500 내용:

탈퇴한 회원입니다. 로그인할 수 없습니다.

확인

회원 로그인

ksh1234

.....

로그인

회원가입

아이디 찾기

비밀번호 찾기

02

	* id varchar(50)	* pw varchar(255)	* name varchar(100)	nick varchar(50)	birth_date date
>	ksh1234	tngns1234!!	김수훈	훈수	1993-04-19

phone_num varchar(20)	email varchar(100)	addr varchar(255)	role enum('관리자','회원','비')
010-4321-4321	ksh1234@gmail.com	경기도 남양주시 다산중	탈퇴회원

네이버 아이디 로그인

카카오 아이디 로그인

Description List

01

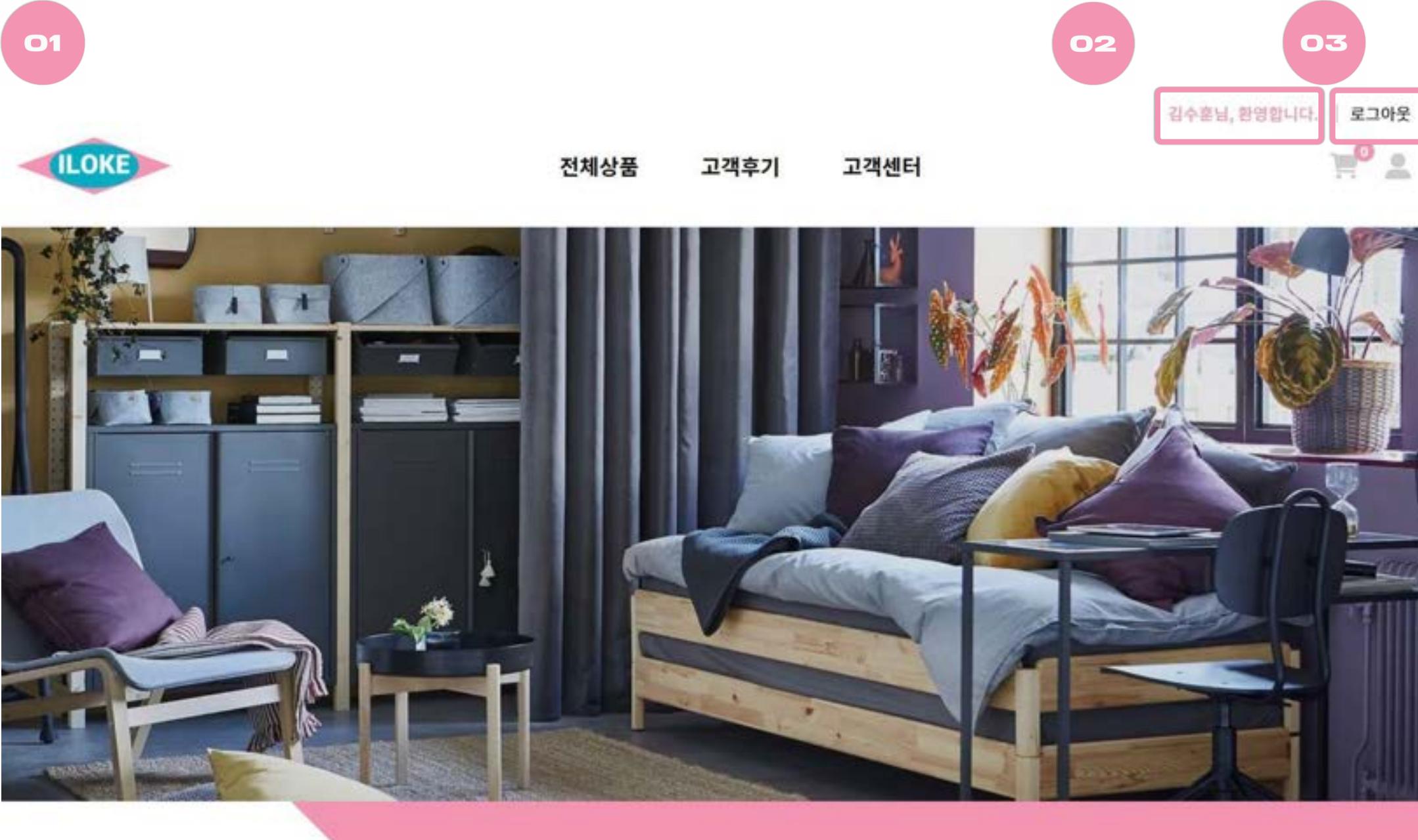
재 로그인 시 탈퇴 회원은 로그인 불가

02

수정된 회원 데이터

탈퇴 시 회원 -> 탈퇴회원으로 변경
로그인 불가

17. 로그인

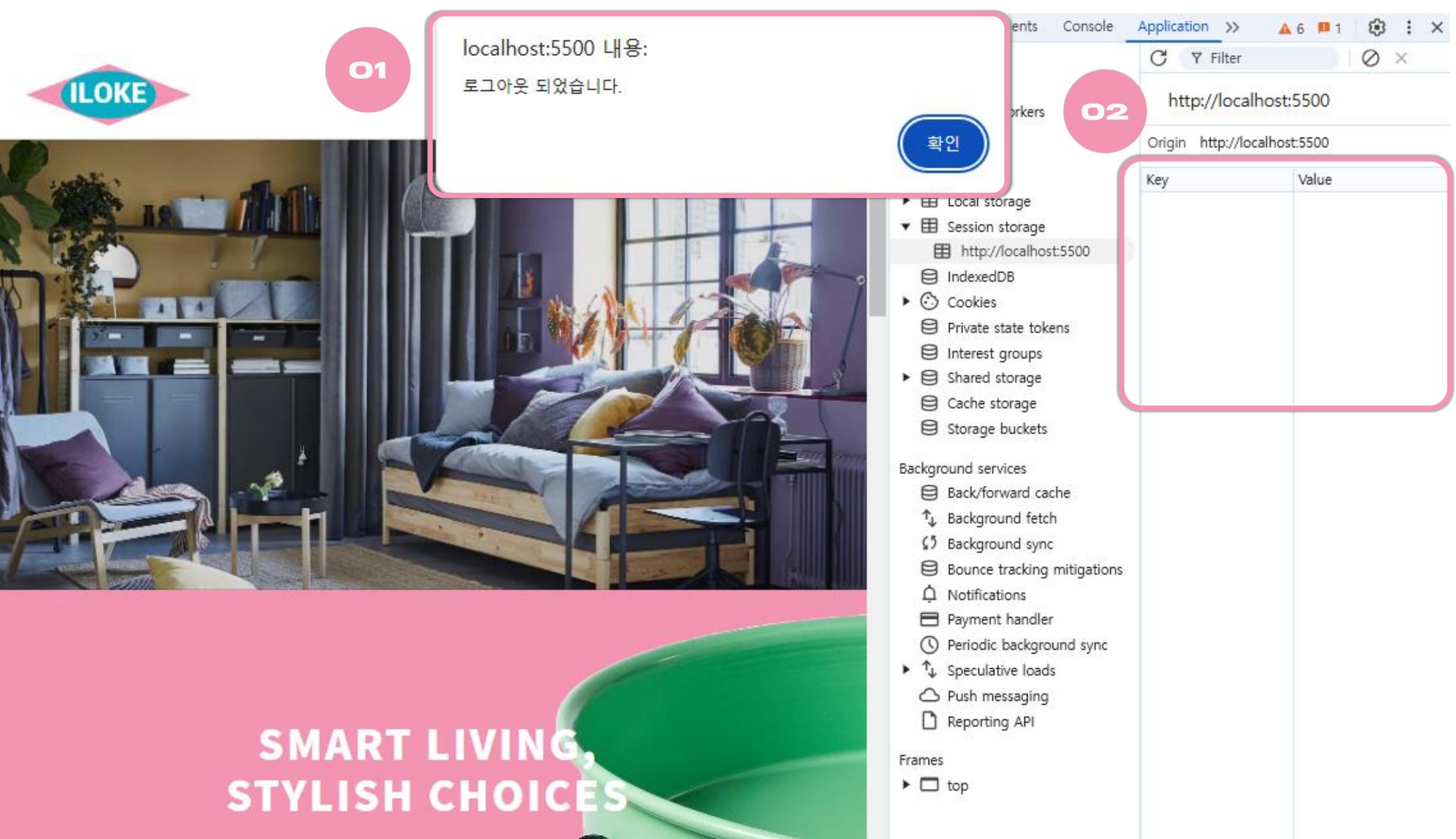


Description List

- 01 로그인 된 메인 페이지
- 02 회원 이름 출력과 환영 메시지
- 03 로그인 시 로그아웃 버튼 활성화

18. 로그아웃

Description List



로그아웃 클릭 시 로그아웃 되었다는 alert 메시지 출력

로그아웃과 동시에 세션 스토리지에 저장되어 있던 회원 이름과 회원 아이디 클리어

19. 로그인 핵심 코드

```
router.post('/', async (req, res) => {
  const { id, pw } = req.body;

  try {
    // 입력받은 아이디에 해당하는 회원 정보를 데이터베이스에서 조회
    const sql = 'SELECT * FROM mem_info WHERE id = ?';
    const [rows] = await conn.execute(sql, [id]);

    // 아이디가 존재하지 않을 경우 에러 응답 반환
    if (rows.length === 0) {
      return res.status(401).json({ success: false, message: '아이디가 존재하지 않습니다.' });
    }

    const user = rows[0]; // 조회된 회원 정보

    // 탈퇴한 회원인지 확인
    if (user.role === '탈퇴회원') {
      return res.status(403).json({ success: false, message: '탈퇴한 회원입니다. 로그인할 수 없습니다.' });
    }

    // 비밀번호 검증 (평문 비교인 경우)
    if (user.pw !== pw) {
      return res.status(401).json({ success: false, message: '비밀번호가 일치하지 않습니다.' });
    }

    // 로그인 성공 시 세션에 userId 저장
    req.session.userId = user.id;

    // console.log('세션에 저장된 userId:', req.session.userId);

    res.json({ success: true, message: '로그인 성공!', user: { id: user.id, name: user.name } });
  } catch (error) {
    console.error('로그인 오류:', error);
    res.status(500).json({ success: false, message: '서버 오류가 발생했습니다.' });
  }
});

module.exports = router;
```

Description List

01

01

사용자 로그인 처리 라우터

19. 로그인 핵심 코드

01

```
const Login = () => {
  const handleLogin = async (e) => {
    e.preventDefault();

    try {
      // 서버로 로그인 요청을 보냄 (memLogin 함수 호출)
      const response = await memLogin(id, pw);

      if (response.data.success) {
        // 로그인 성공 시 서버에서 반환된 사용자 이름을 추출
        const userName = response.data.user.name;

        // 세션 스토리지에 사용자 정보 저장
        sessionStorage.setItem('userId', id);
        sessionStorage.setItem('userName', userName);
        sessionStorage.setItem('auth', 'true');

        // storage 이벤트를 수동으로 발생시킴
        window.dispatchEvent(new Event('storage'));

        alert(`#${userName}님, 환영합니다!`);

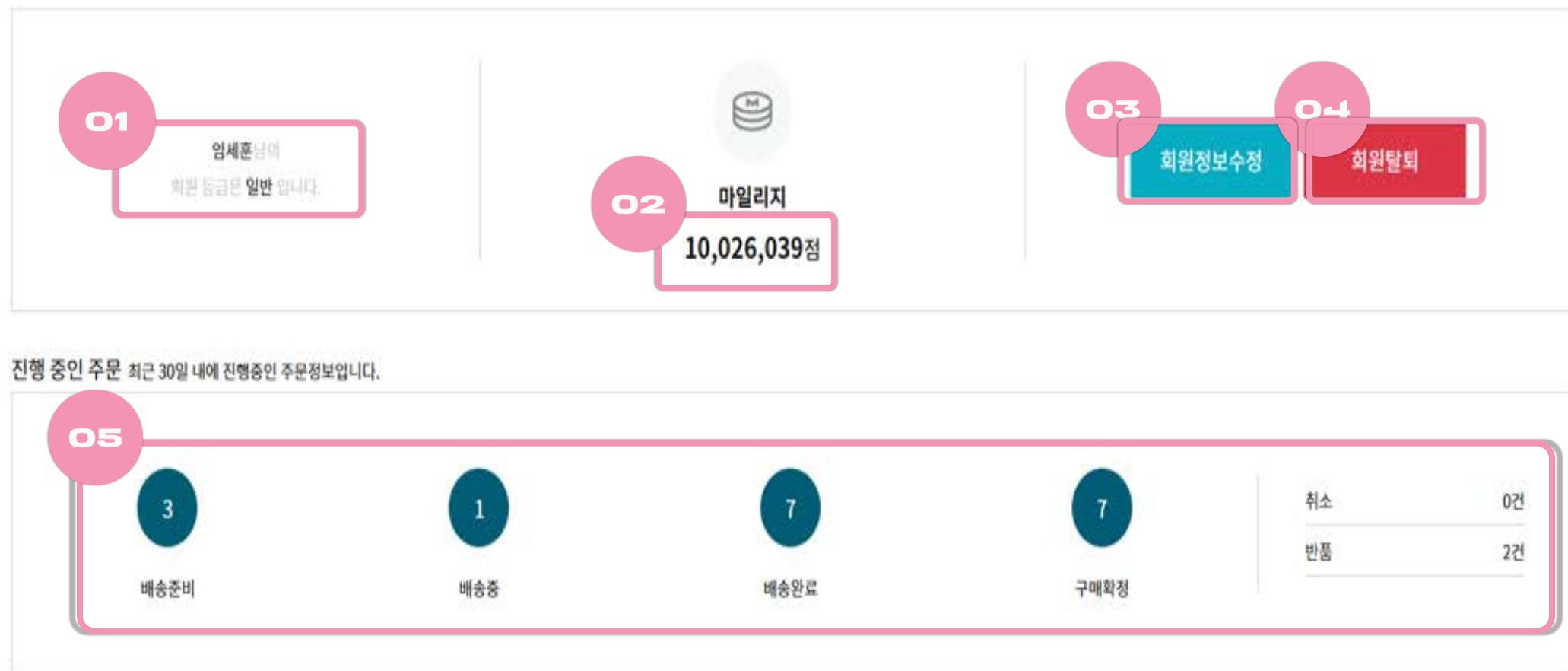
        navigate('/home'); // 홈 페이지로 이동
        //window.location.reload();
      } else {
        alert(response.data.message || '로그인에 실패했습니다.');
      }
    } catch (error) {
      if (error.response) {
        // 403 상태 코드 처리
        if (error.response.status === 403) {
          alert('탈퇴한 회원입니다. 로그인할 수 없습니다.');
        } else if (error.response.status === 401) {
          alert('아이디 또는 비밀번호가 잘못되었습니다.');
        } else {
          alert('서버 오류가 발생했습니다. 다시 시도해 주세요.');
        }
      } else {
        alert('네트워크 오류가 발생했습니다. 다시 시도해 주세요.');
      }
    }
  };
};
```

Description List

01

사용자가 입력한 아이디와 비밀번호 인증 후
성공 시 세션 스토리지에 사용자 정보를 저장 및
로그인 하게하는 함수

20. 마이페이지



Description List

- 01** 회원에 대한 정보를 DB에서 불러옴
- 02** 보유한 마일리지양을 DB에서 불러옴
금액 클릭시 마일리지 상세내역 조회가능
- 03** 회원정보수정기능
- 04** 회원탈퇴기능
- 05** 주문을 주문상태에 따라 진행상황을 확인가능

20. 마이페이지

The screenshot shows a user profile summary at the top with a balance of 10,026,039 points. Below it is a table of transaction history.

적립 날짜	적립 내용	유효 날짜	변동 내역
2024-12-30T06:46:48.000Z	주문 취소로 인한 마일리지 반환	2024-12-29T15:00:00.000Z	20,000점
2024-12-30T06:46:07.000Z	구매확정 마일리지 적립	2024-12-29T15:00:00.000Z	15,580점
2024-12-30T06:44:39.000Z	상품 구매 마일리지 사용	2024-12-29T15:00:00.000Z	-50,000점
2024-12-30T02:42:07.000Z	주문 취소로 인한 마일리지 반환	2024-12-29T15:00:00.000Z	3,000점
2024-12-30T02:41:23.000Z	구매확정 마일리지 적립	2024-12-29T15:00:00.000Z	30,400점
2024-12-30T02:39:49.000Z	상품 구매 마일리지 사용	2024-12-29T15:00:00.000Z	-20,000점
2024-12-30T01:26:39.000Z	주문 취소로 인한 마일리지 반환	2024-12-29T15:00:00.000Z	1,800점
2024-12-30T01:25:39.000Z	구매확정 마일리지 적립	2024-12-29T15:00:00.000Z	28,060점
2024-12-30T01:23:23.000Z	상품 구매 마일리지 사용	2024-12-29T15:00:00.000Z	-1,000점
2024-12-29T18:23:19.000Z	상품 구매 마일리지 사용	2024-12-29T15:00:00.000Z	-1,800점
2024-12-29T11:17:43.000Z	구매확정 마일리지 적립	2024-12-28T15:00:00.000Z	76,773점
2024-12-29T11:09:08.000Z	구매확정 마일리지 적립	2024-12-28T15:00:00.000Z	76,773점

Description List

- 보유한 마일리지양을 DB에서 불러옴
금액 클릭시 마일리지 상세내역 조회가능
- DB에서 불러온 마일리지 변동내역
- 돌아가기 버튼을 통해 기존 마이페이지로 이동

20. 마이페이지

Description List

주문목록 / 배송조회				
01	날짜/주문번호	상품명/옵션	상품금액/수량	주문상태
최근 주문 정보				
2024/12/30 29	상품 이미지 _상품1번	155,800	배송완료 구매확정됨	후기등록
2024/12/30 28	상품 이미지 _상품 정보 없음	304,000	취소완료	
2024/12/30 27	상품 이미지 _상품 정보 없음	280,600	배송완료 구매확정됨	후기등록

- 01 기간으로 주문조건을 검색할 수 있는 필터
- 02 주문을 데이터로 불러와 기간순으로 정렬
- 03 주문상태에 따른 버튼 생성
 - 배송준비, 배송중: 취소요청
 - 배송완료(구매확정): 후기등록
 - 배송완료(구매미확정): 구매확정, 반품요청

20. 마이페이지

2024/12/29	상품 이미지 상품 정보 없음	1	202,224	배송준비	01 취소 요청
2023/12/07	상품 이미지 상품 정보 없음	11	120,000	배송원료 구매확정됨	02 후기등록
2023/12/03	상품 이미지 상품 정보 없음	7	100,000	배송완료	01 구매확정 01 반품요청
					03 이전 페이지 1 / 2 다음

Description List

- 01** 주문상태 변경에 따른 버튼 생성
- 02** 구매확정시 구매확정표시
- 03** 페이지네이션 기능

20. 마이페이지

Description List

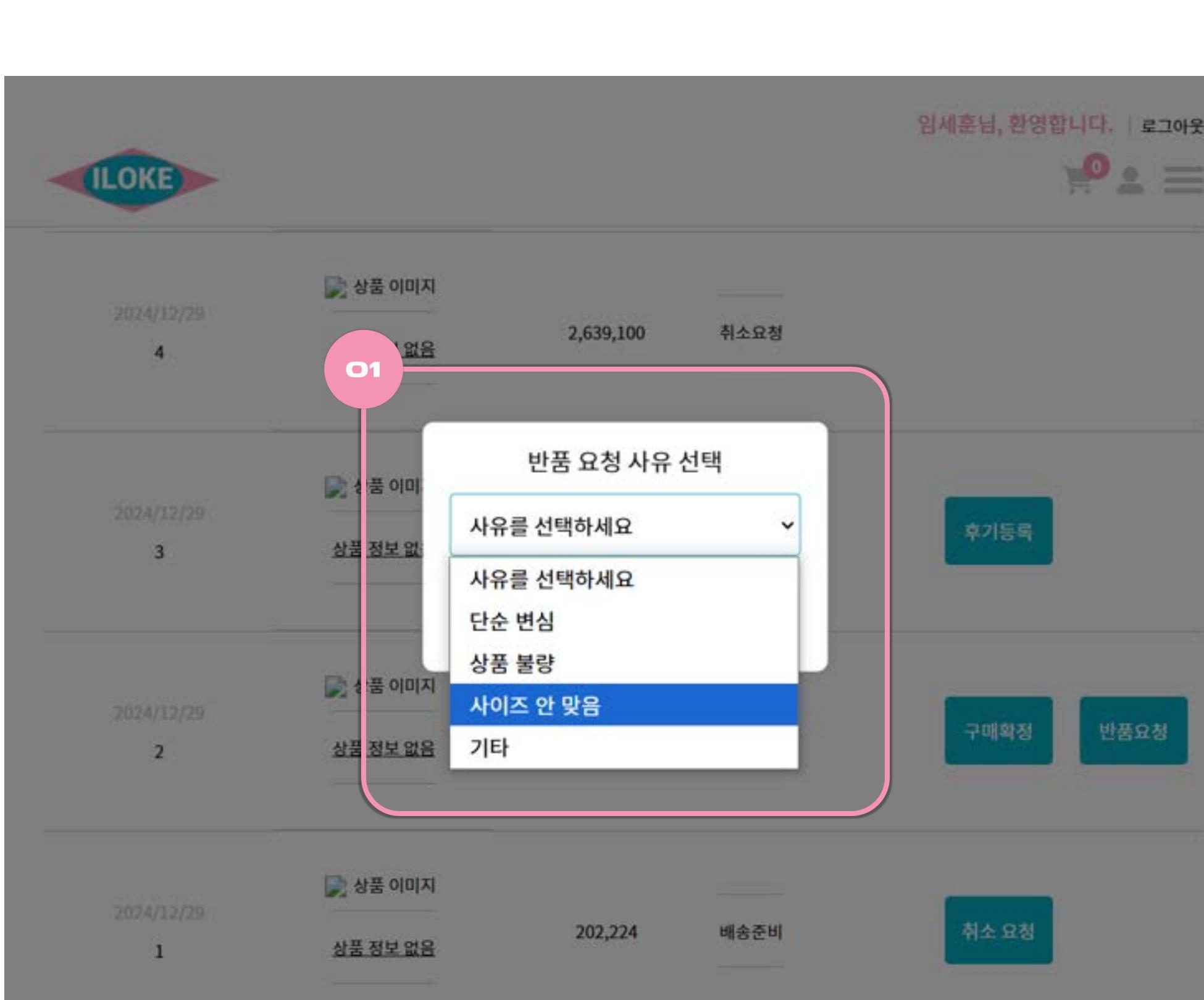
01

취소요청 모달창과 함께 사유선택
사유와 요청시기가 데이터로 등록됨



20. 마이페이지

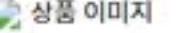
Description List



- 01** 반품요청 모달창과 함께 사유선택
사유와 요청시기가 데이터로 등록됨

20. 마이페이지

Description List

2024/12/30 27	 상품 정보 없음	280,600	배송완료 구매확정됨	후기등록
2024/12/30 26	 상품 정보 없음	534,000	반품요청	
2024/12/29 4	 상품 정보 없음	2,639,100	취소요청	
2024/12/29 3	 상품 정보 없음	767,730	배송완료 구매확정됨	후기등록

01

사유를 선택하고 확인버튼 클릭시
반품요청,취소요청 상태로 변경

20. 마이페이지

Description List

01

```
const [modalVisible, setModalVisible] = useState(false);
const [selectedOrder, setSelectedOrder] = useState(null);
const [reason, setReason] = useState('');
const [requestType, setRequestType] = useState('') // 취소 or 반품 구분
```

02

```
const [totalOrders, setTotalOrders] = useState(0); // 총 주문 수 추가
const [currentPage, setCurrentPage] = useState(1); // 현재 페이지 추가
const ordersPerPage = 10; // 한 페이지당 10개의 주문
```

03

```
const [startDate, setStartDate] = useState("");
const [endDate, setEndDate] = useState("");
const [recentOrders, setRecentOrders] = useState([]);
const [orderCounts, setOrderCounts] = useState({
    배송준비: 0,
    배송중: 0,
    배송완료: 0,
    구매확정: 0
});

const [returnCancelCounts, setReturnCancelCounts] = useState([
    취소: 0,
    반품: 0,
]);
```

01

요청사유와 모달창과 관련된 코드

02

페이지네이션 관련 코드

03

주문상태에 따른 숫자변경 코드

20. 마이페이지

Description List

01

```
const handleConfirmRequest = async () => {
  if (!selectedOrder || !reason) {
    alert("사유를 선택해주세요.");
    return;
  }

  const requestData = {
    orderId: selectedOrder.id,
    reason: reason,
    requestType: requestType
  };

  try {
    const response = await uRequestCancelOrReturn(memberId, requestData); // API 호출

    if (response.data.success) {
      alert(`#${requestType} 요청이 완료되었습니다.`);
    }
  } catch (error) {
    console.error(error);
  }
};
```

01

요청사유 전달 코드

02

백서버에 데이터를 요청하는 api

02

```
export const uRequestCancelOrReturn = (memberId, requestData) =>
  API.patch('/mypage/request',
    requestData,
    {
      headers: { 'user-id': memberId }
    }
  );
```

20. 마이페이지

Description List

01

```
const [userInfo, setUserInfo] = useState({ name: '***', m_remain: 0, type: '일반회원등급' });
```

01

회원정보받아오는 state 설정

02

```
const cancelReasons = ['단순 변심', '상품 불량', '배송 지연', '기타'];
const returnReasons = ['단순 변심', '상품 불량', '사이즈 안 맞음', '기타'];
```

02

취소사유와 반품사유 설정

02

```
const memberId = sessionStorage.getItem('userId');

const [isLoading, setIsLoading] = useState(true); // 로딩 상태 추가

useEffect(() => {
  if (!memberId) {
    alert('로그인 정보가 없습니다. 다시 로그인 해주세요.');
    window.location.href = '/login';
  } else {
    setIsLoading(false);
  }
}, [memberId]);
```

03

Session값으로 사용자 정보를 받아와
비로그인시 접근 불가능하게 만듬

20. 마이페이지

01

```
// 마일리지 내역 가져오기
const fetchMileageHistory = async () => {
  setLoading(true); // 로딩 시작
  try {
    const response = await rFetchMileageHistory(memberId);
    setMileageHistory(response.data.mileageHistory);
    setShowMileageTable(true);

  } catch (error) {
    console.error("마일리지 내역 가져오기 오류:", error.message);
    alert("마일리지 정보를 가져오는 중 오류가 발생했습니다.");
  } finally {
    setLoading(false); // 로딩 종료
  }
};
```

02

```
const [showMileageTable, setShowMileageTable] = useState(false);
const [mileageHistory, setMileageHistory] = useState([]);
```

Description List

01

마일리지 내역 조회 함수

02

마일리지 내역 조회를 위한 state 설정

21. 쇼핑몰 메인화면 스와이퍼



메인배너

section3 슬라이드

This image displays a product slider section titled 'PRODUCTS' in large white letters. It includes a navigation arrow '〈 〉' at the top right. Below the title are four thumbnail images of interior scenes: a blue wall with a green plant, a red sofa on a blue carpet, a red sofa in a room with yellow and blue walls, and a pink armchair. Each thumbnail has placeholder text 'Lorem ipsum dolor' below it. A pink circular callout '02' is located in the bottom-left corner of the slider area.

21. 쇼핑몰 메인화면 스와이퍼 - 메인배너

01

html – js 형식 스와이퍼를 리액트 방식으로 변경

01

```
<Swiper
  tag='section'
  className='main-banner main-con'
  wrapperTag='div'
  loop= {true}
  autoplay = {
    {delay: 4000 , disableOnInteraction: false}
  }
  modules={[EffectFade]}
  effect="fade"
  speed = {1000}
>
  <SwiperSlide className="banner-img01"></SwiperSlide>
  <SwiperSlide className="banner-img02"></SwiperSlide>
  <SwiperSlide className="banner-img03"></SwiperSlide>
  <SwiperSlide className="banner-img04"></SwiperSlide>
  <SwiperSlide className="banner-img05"></SwiperSlide>
</Swiper>
```

21. 쇼핑몰 메인화면 스와이퍼 - section3 슬라이드01

01

```
<Swiper
  tag='div'
  className='product-swipe'
  wrapperTag='ul'
  loop= {true}
  slidesPerView = {1}
  spaceBetween = {30}
  modules={[{Navigation}]}
  navigation={{
    //스와이퍼 외부에 컨트롤러 설정하기
    prevEl: prevRef.current,
    nextEl: nextRef.current,
  }}
  onBeforeInit={(swiper) => {
    // 초기화 전에 네비게이션 버튼을 swiper에 할당
    swiper.params.navigation.prevEl = prevRef.current;
    swiper.params.navigation.nextEl = nextRef.current;
  }}
  autoplay = {
    { delay: 3000 ,
      disableOnInteraction: false}
  }
  speed = {1000}
  breakpoints={
    1200: { slidesPerView: 3, spaceBetween : 60 },
    800: { slidesPerView: 2, spaceBetween : 30 }
  }
>
```

01

html – js 형식 스와이퍼를 리액트 방식으로 변경

21. 쇼핑몰 메인화면 스와이퍼 - section3 슬라이드02

02

이미지를 map 반복문 사용하여 스와이퍼에 적용

02

```
>
  [
    "main_product09.jpg",
    "main_product10.jpg",
    "main_product11.jpg",
    "main_product12.jpg",
    "main_product13.jpg",
    "main_product14.jpg",
    "main_product15.jpg",
    "main_product16.jpg",
  ].map((image, index) => (
  <SwiperSlide
    tag='li'
    key={index}>
    <Link to={'/'}>
      <span className="product-img">
        <img
          src={`./content/img/main/${image}`}
          alt={`Product ${index + 1}`}
        />
      </span>
      <span className="product-tit">Lorem ipsum dolor</span>
    </Link>
  </SwiperSlide>
))
</Swiper>
```

22. 쇼핑몰 상품 리스트

내게 맞는 상품 찾기
상품 전체보기

01

총제품수 3

상품1
10,000원

상품2
300,000원

상품3
3,000원
10% 2,700원

02

01 등록된 상품 개수

02 등록된 상품 정보

22. 쇼핑몰 상품 리스트

01

```
const [listVal, setListVal] = useState([]); //데이터 초기값
const [chkVal, setChkVal] = useState(null); //옵션 체크박스 초기값
let listLength = listVal.length; //리스트 개수
let listTxt = listLength;
```

02

```
// 데이터 연결
useEffect(()=>{
  if(!listLength || listLength == 0){
    listTxt = 0
  }
  goodsSelect()
  .then(res =>{
    // console.log(res.data); // data 확인 - 배열
    setListVal(res.data)
  })
  .catch(err => {
    console.error('상품리스트 db 프론트 불러오기 오류 ', err)
  })
},[listLength])
```

01

초기값 설정

02

등록된 상품 목록 들고오기

22. 쇼핑몰 상품 리스트

03

```
<Link to={`/goodsDetail/${item.id}`} className={styles.product_item} key={idx}>
  <div className={styles.product_image_link}>
    
    <p>{item.name}</p>
    <span className={styles.original_price}>{slash}</span>
    <div className={styles.price_container}>
      <div className={styles.price_details}>
        <b className={styles.discount_percent}>{discount}</b>
        <strong className={styles.discounted_price}>{newFnPrice.toLocaleString()}원</strong>
      </div>
    </div>
  </div>
</Link>
```

03

저장된 상품 정보 가져오기

23. 쇼핑몰 상품 상세



공식몰단독

상품1

10,000원

무이자

카드사별 무이자 혜택 안내

추가 혜택

네이버페이 최대 3% 적립
페이코 포인트 결제시 1% 적립

02

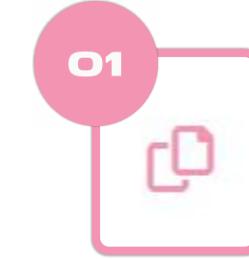
부분추가 (+50000원)

소 (+500원)

옵션 추가

총 합계금액

0원



01

현재 보고있는 상품 상세 페이지 링크 복사하기

02

선택된 옵션/키워드 목록, 가격 불러오기

23. 쇼핑몰 상품 상세

01

```
const [goodsInfo, setGoodsInfo] = useState("")//상품정보
const [thumbsSwiper, setThumbsSwiper] = useState(null); //슬라이드 썸네일

const [orPrice, setOrPrice] = useState('');//원가 문자열(ex 10,000)
const [fnPrice, setfnPrice] = useState(0)//숫자열로 변경된 최종가

const [opOne, setOpOne] = useState(null)//옵션1(1자 작업명 color-option)
const [opTwo, setOpTwo] = useState(null)//옵션2(1자 작업명 leather-option)
const [opThree, setOpThree] = useState(null)//옵션3(1자 작업명 stool-option)

const [opList, setOpList] = useState({})//옵션리스트
const [loading, setLoading] = useState(true)
const [error, setError] = useState(null)

const [colorOption, setColorOption] = useState("none");
const [leatherOption, setLeatherOption] = useState("none");
const [stoolOption, setStoolOption] = useState("none");
const [optionList, setOptionList] = useState([]);
const [totalPrice, setTotalPrice] = useState(0);
const [isHeartActive, setIsHeartActive] = useState(false);

const [productId, setProductId] = useState(null) // ✅ 상품 ID 상태 추가
const memberId = sessionStorage.getItem('userId'); // 사용자 ID를 세션에서 가져옴

let optionTitArr = [];//옵션명 배열 전체 출력
let optionTit = [];//옵션명 중복 제거 배열
let optionMainArr = [];//옵션명 배열 전체 출력
let optionMain = [];//옵션명 중복 제거 배열
let optionValCal = {};//옵션값
let optionValCalAll = [];//옵션값
let optionData = {};//옵션가격
let optionName = {};//옵션명
let optionNameArr = [];//옵션명
let opValArr = {};
```

01

초기값 설정

23. 쇼핑몰 상품 상세 - 링크 복사

02

현재 주소값을 가져와서 클립보드에 복사

02

```
<button onClick={handleClipboardCopy} className={styles.clipboard_btn}>
  <i className="fa-regular fa-copy"></i>
</button>
```

```
const handleClipboardCopy = () => {
  navigator.clipboard.writeText(window.location.href);
  alert("URL 주소를 복사했습니다.");
};
```

23. 쇼핑몰 상품 상세 - 옵션/키워드 목록, 가격 불러오기

```
{goodsInfo.length && !goodsInfo.opkey_value_status ? (  
  <div className={styles.product_options}>{/* 옵션 */}  
    {  
      optionMain.map((st, i) => { //옵션 메인코드  
        console.log('Rendering optionMain:', st);  
        const opNameIdx = optionTit[i]//옵션명  
        const optionValues = opValArr[st] || [];//이걸 나눠야해  
        const optionPrices = optionData[opNameIdx] || [];//옵션금액  
        const optionNames = optionName[opNameIdx] || [];//옵션값  
        // console.log(st)  
        // console.log(optionMainArr)  
  
        return (  
          <select  
            // className="color_option"  
            className={  
              i === 0 ? "color_option" :  
              i === 1 ? "leather_option" : "stool_option"  
            }  
            value={  
              i === 0 ? colorOption :  
              i === 1 ? leatherOption : stoolOption  
            }  
            onChange={  
              i === 0 ? handleColorChange :  
              i === 1 ? handleLeatherChange : handleStoolChange  
            } //옵션개수에 따라 핸들러 달라짐  
            key={i}>  
            <option value="none" data-chrg="0">= {opNameIdx} =</option>  
            {  
              optionValues.map((value, idx) => (  
                <option key={idx}  
                  value={st.toString() + optionValues[idx].toString()} //option의 값  
                  data-chrg={optionPrices[idx]} //해당 option의 가격  
                >  
                {optionNames[idx]} ({+optionPrices[idx]})  
              </option>  
            ))}  
          </select>  
        )  
      })  
    }  
  
    <button onClick={handleOptionSelect}>옵션 추가</button>  
  </div>
```

03

옵션 셀렉트 순서/개수에 따라 다른 클래스명, Value값, 핸들러가 주어져서 각 옵션명과 가격을 가져온다.

23. 쇼핑몰 상품 상세 - 옵션/키워드 목록, 가격 불러오기

```
04
<div className={styles.option_wrap}>
  {optionList.map((option, index) =>
    (<div className={styles.option_bg} key={`${option.key}-${index}`}>
      /* {console.log(option)} */
      <span className={styles.option_select}>
        {option.color} / {option.leather}
        /* / {option.stool} */
      </span>
      <div className={styles.option_display}>
        <div className={styles.option_quantity}>
          {option.quantity > 1 ? (
            <>
              <button onClick={() => decreaseQuantity(option.key)}>-</button>
              <span>{option.quantity}</span>
              <button onClick={() => increaseQuantity(option.key)}>+</button>
            </>
          ) : (
            <>
              <button onClick={() => removeOption(option.key)}>삭제</button>
              <span>{option.quantity}</span>
              <button onClick={() => increaseQuantity(option.key)}>+</button>
            </>
          )}
        </div>
        <div className={styles.option_amount}>
          <strong>{(option.price * option.quantity).toLocaleString()}원</strong>
        </div>
      </div>
    )})
  <div className={styles.option_calc}>
    <dl className={styles.option_total}>
      <dt>총 합계금액</dt>
      {/* <dd>{fnPriceNum}원</dd> */}
      <dd>{totalPrice.toLocaleString()}원</dd>
    </dl>
  </div>
</div>
```

04

가져온 셀렉트 값의 옵션 가격을 원가와 더해서 최종 가격으로 출력.

= 가죽 추가 =

= 사이즈 =

옵션 추가

전체추가 / 소

삭제 1 + 110,500원

총 합계금액 110,500원

23. 쇼핑몰 상품 상세 - 옵션/키워드 목록, 가격 불러오기

05

선택한 옵션목록 출력 > 주문 정보로 보내짐

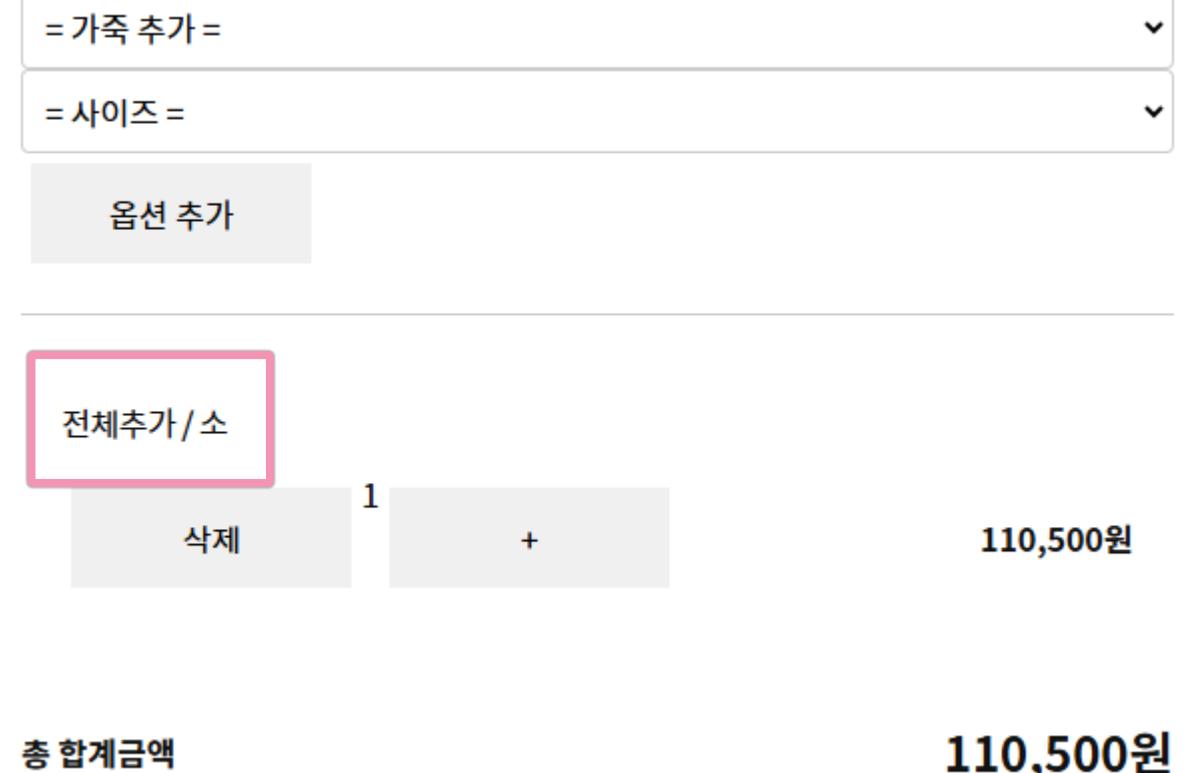
05

```
let optionValCalMapped = {} // 결과 객체//여기서 옵션명 받아와야함

if(optionValCalAll && optionNameArr && optionMainArr) {
    // 두 배열을 순회하여 객체에 "조합된 키": "옵션명" 형태로 저장
    optionValCalAll.forEach((optionValue, index) => {
        const optionName = optionNameArr[index]; // 해당 인덱스의 옵션명
        const optionMain = optionMainArr[index]; // 해당 인덱스의 메인 코드

        // 새로운 키 생성: mainCode + optionValue (예: 42 + 1 = 421)
        const combinedKey = optionMain * 10 + optionValue;

        // 새로운 키와 옵션명을 객체에 맵핑
        optionValCalMapped[combinedKey] = optionName;
        // console.log(optionValue); //옵션값 코드 전체다
    });
}
```



24. 게시물 - 전체후기

Description List

고객후기

전체후기

01

조회기간

02

03

04

게시판번호	제목	조회수	작성자	등록일자
bd201412020005	애들이 좋아해요	4	kkkk123	2024-12-02 12:19:17
bd202412230001	民生人1	1	aaa1	2024-12-23 16:01:21

검색 필터
- 조회기간

검색된 전체후기 리스트
- 게시판번호
- 제목
- 조회수
- 작성자
- 등록일자

페이지

글쓰기

01 검색 필터
- 조회기간

02 검색된 전체후기 리스트
- 게시판번호
- 제목
- 조회수
- 작성자
- 등록일자

03 페이지

04 글쓰기

24. 게시물 - 전체후기 - 글쓰기

Description List

고객후기				
전체후기				
게시판번호	제목	조회수	작성자	등록일자
bd201412020005	애들이 좋아해요	4	kkkk123	2024-12-02 12:19:17
bd202412230001	스스로	1	aaal	2024-12-23 16:01:21

< 1 >

01 작성자 : 로그인 시 저장된 세션값 표시

02 제목 : 제목 작성란

03 본문 : 내용 작성란

04 목록으로 : 리스트로 돌아가기

05 저장 : 작성한 내용 저장

글쓰기

글쓰기

01

서서유

02

제목을 입력해주세요.

03

내용

04

목록으로

05

저장

24. 게시물 - 전체후기 - 상세보기

Description List

고객후기

전체후기

조회기간 : 오늘 7일 15일 1개월 3개월 1년 | 2024-12-01 ~ 2024-12-31 | 조회

게시판번호	제목	조회수	작성자	등록일자
bd201412020005	애들이 좋아해요	4	kkkk123	2024-12-02 12:19:17
bd202412230001	애들이 좋아해요	1	aaa1	2024-12-23 16:01:21

글쓰기

전체후기

본문
애들이 좋아해서 리뷰 남겨봅니다

01 수정 **02** 삭제 **03** 목록으로

01 본문 : 작성된 내용

02 수정 : 게시글 내용 수정

03 삭제 : 게시글 삭제

04 목록으로 : 리스트로 돌아가기

24. 게시물 - 포토후기

Description List

01

전체 후기와 동일

01

고객후기

포토후기

조회기간 오늘 7일 15일 1개월 3개월 1년 2024-12-01 ~ 2024-12-31 조회

게시판번호	제목	조회수	작성자	등록일자
bd202412030001	1번이미지	2	usuuj215	2024-12-03 13:46:30
bd202412030002	2번이미지	3	ml1523	2024-12-03 13:46:30
bd202412230003	우리 페이지가	3	aaa1	2024-12-23 16:51:49
bd202412230004	딸기사진	1	aaa1	2024-12-23 16:52:32

1

24. 게시물 - 포토후기 - 글쓰기

고객후기

포토후기											
조회기간	오늘	7일	15일	1개월	3개월	1년	2024-12-01	-	2024-12-31	조회	
게시판번호		제목		조회수	작성자		등록일자				
bd202412030001		1번이미지		2	usluj215		2024-12-03 13:46:30				
bd202412030002		2번이미지		3	ml1523		2024-12-03 13:46:30				
bd202412230003		우리 페이지가		3	aaa1		2024-12-23 16:51:49				
bd202412230004		딸기사진		1	aaa1		2024-12-23 16:52:32				

1

글쓰기

글쓰기

포토후기	작성자	01	저서유
제목	제목	02	제목을 입력해주세요.
본문	본문	03	내용
첨부파일	첨부파일	04	파일 선택 선택된 파일 없음
		05	목록으로
		06	저장

Description List

- 01 작성자 : 로그인 시 저장된 세션값 표시
- 02 제목 : 제목 작성란
- 03 본문 : 내용 작성란
- 04 파일 선택 : 사진 첨부
- 05 목록으로 : 리스트로 돌아가기
- 06 저장 : 작성한 내용 저장

24. 게시물 - 포토후기 - 상세보기

Description List

고객후기

포토후기

조회기간 : 오늘 | 7일 | 15일 | 1개월 | 3개월 | 1년 | 2024-12-01 ~ 2024-12-31 | 조회

게시판번호	제목	조회수	작성자	등록일자
bd202412030001	1번이미지	2	usuuj215	2024-12-03 13:46:30
bd202412030002	2번이미지	3	ml1523	2024-12-03 13:46:30
bd202412230003	우리 페이지가	3	aaa1	2024-12-23 16:51:49
bd202412230004	딸기사진	1	aaa1	2024-12-23 16:52:32

글쓰기

포토후기

본문
잘될상인가!

01

02

03

04

05

수정

삭제

목록으로

```
graph TD; A[포토후기] --> B[본문  
잘될상인가!]; B --> C[01]; C --> D[02]; D --> E[03]; E --> F[04]; F --> G[05]; G --> H[목록으로]
```

01 본문 : 작성된 내용

02 사진 : 등록한 사진 표기

03 수정 : 게시글 내용 수정

04 삭제 : 게시글 삭제

05 목록으로 : 리스트로 돌아가기

24. 게시물 - 공지사항

Description List

01

전체 후기와 동일

01

고객센터

공지사항

조회기간 오늘 7일 15일 1개월 3개월 1년 2024-12-01 ~ 2024-12-31 조회

게시판번호	제목	조회수	작성자	등록일자
bd202412050001	공지사항1	5	dc4638	2024-12-05 13:46:30
bd202412060001	공지사항2	4	dc4638	2024-12-06 12:15:06
bd202412060002	세번째공지	1	dc4638	2024-12-06 12:15:25
bd202412060004	두번째 공지사항	1	dc4638	2024-12-06 12:18:03
bd202412200008	이건 신기한 공지사항	0	dc4638	2024-12-20 15:57:25
bd202412300001	필독 무탁드립니다.	0	admin123	2024-12-30 17:22:03
bd202412300002	공지사항입니다.	0	admin123	2024-12-30 17:23:04
bd202412300004	판매에 대한 공지	0	admin123	2024-12-30 17:23:52
bd202412300005	계속되는 공지	0	admin123	2024-12-30 17:24:06
bd202412300007	공지사항1	0	admin123	2024-12-30 17:24:40

< 1 2 >

24. 게시물 - 공지사항 - 상세보기

고객센터

공지사항

조회기간

오늘 7일 15일 1개월 3개월 1년

2024-12-01 ~ 2024-12-31

조회

게시판번호	제목	조회수	작성자	등록일자
bd202412050001	공지사항1	5	dc4638	2024-12-05 13:46:30
bd202412060001	공지사항2	4	dc4638	2024-12-06 12:15:06
bd202412060002	세번쨰공지	1	dc4638	2024-12-06 12:15:25
bd202412060004	두번쨰 공지사항	1	dc4638	2024-12-06 12:18:03
bd202412200008	이건 신기한 공지사항	0	dc4638	2024-12-20 15:57:25
bd202412300001	필독 무학드립니다.	0	admin123	2024-12-30 17:22:03
bd202412300002	공지사항입니다.	0	admin123	2024-12-30 17:23:04
bd202412300004	판매에 대한 공지	0	admin123	2024-12-30 17:23:52
bd202412300005	계속되는 공지	0	admin123	2024-12-30 17:24:06
bd202412300007	공지사항1	0	admin123	2024-12-30 17:24:40

< 1 >

↓

01

공지사항

본문

자 이제 시작합니다!

목록으로

Description List

01

전체 후기와 동일

24. 게시물 - 상품문의

Description List

상품문의

고객센터

01

조회기간
오늘 | 7일 | 15일 | 1개월 | 3개월 | 1년 | 2024-12-01 ~ 2024-12-31 | 조회

02

게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412020002	의자가 원래 없나요?	5	ml523	2024-12-02 12:13:52	답변완료
bd202412020008	원단이 이상한데?	3	dc4638	2024-12-02 12:15:11	답변필요
bd202412240005	1	2	admintest1	2024-12-24 18:03:17	답변필요
bd202412260007	3	1	admintest1	2024-12-26 17:49:52	답변필요
bd202412260008	656	0	admintest1	2024-12-26 17:50:03	답변필요

03

04

글쓰기

< | 1 | >

01 검색 필터
- 조회기간

02 검색된 전체후기 리스트
- 게시판번호
- 제목
- 조회수
- 작성자
- 등록일자
- 답변여부

03 페이지

04 글쓰기

24. 게시물 - 상품문의 글쓰기

고객센터

상품문의

조회기간 오늘 7일 15일 1개월 3개월 1년 2024-12-01 ~ 2024-12-31 조회

게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412020002	의자가 원래 없나요?	5	mi1523	2024-12-02 12:13:52	답변완료
bd202412020008	원단이 이상한데?	3	dc4638	2024-12-02 12:15:11	답변필요
bd202412240005	1	2	admintest1	2024-12-24 18:03:17	답변필요
bd202412260007	3	1	admintest1	2024-12-26 17:49:52	답변필요
bd202412260008	656	0	admintest1	2024-12-26 17:50:03	답변필요

상품문의

작성자 저서유

제목

본문

목록으로 저장

Description List

01

전체 후기와 동일

글쓰기

글쓰기

24. 게시물 - 상품문의 - 상세보기

고객센터

상품문의

조회기간 오늘이전 7일 15일 1개월 3개월 1년 2024-12-01 ~ 2024-12-31 조회

게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412020002	의자가 원래 없나요?	5	mi1523	2024-12-02 12:13:52	답변완료
bd202412020008	원단이 이상한데?	3	dc4638	2024-12-02 12:15:11	답변필요
bd202412240005	1	2	admintest1	2024-12-24 18:03:17	답변필요
bd202412260007	3	1	admintest1	2024-12-26 17:49:52	답변필요
bd202412260008	656	0	admintest1	2024-12-26 17:50:03	답변필요

< >

글쓰기

상품문의

01 본문 내용

02 답변 내용

본문

책상에 의자포함하려했는데 이 테이블은 의자가 원래 없나요?3

답변

test12

수정 삭제 목록으로

Description List

01 본문 내용

02 답변 내용

24. 게시물 - FAQ

Description List

01

전체 후기와 동일

01

FAQ

고객센터

제목	작성자	등록일자
Q&A1	dc4638	2024-12-05 13:46:30
두번째 faq도?	dc4638	2024-12-06 12:16:51
1	dc4638	2024-12-20 14:48:31
1	dc4638	2024-12-20 14:49:34
1	dc4638	2024-12-20 14:50:39
2	dc4638	2024-12-20 14:50:47
안녕하세요	dc4638	2024-12-20 15:00:01

< 1 >

24. 게시물 - FAQ - 상세보기

고객센터

FAQ												
조회기간	오늘	7일	15일	1개월	3개월	1년	2024-12-01	~	2024-12-31	조회	검색	
게시판번호	제목			조회수	작성자	등록일자						
bd202412050002	Q&A1			2	dc4638	2024-12-05 13:46:30						
bd202412060003	두번쨰 faq도?			1	dc4638	2024-12-06 12:16:51						
bd202412200001	1			2	dc4638	2024-12-20 14:48:31						
bd202412200002	1			0	dc4638	2024-12-20 14:49:34						
bd202412200003	1			1	dc4638	2024-12-20 14:50:39						
bd202412200004	2			0	dc4638	2024-12-20 14:50:47						
bd202412200005	안녕하세요			0	dc4638	2024-12-20 15:00:01						

< 1 >

01

Description List

01

공지사항과 동일

FAQ

본문

이건 큐엔에이입니다. 이렇게 답변 드립니다.

수정본추가

목록으로

24. 게시물 - 1:1문의

Description List

01

상품 문의와 동일

01

고객센터

1:1문의

조회기간 오늘 | 7일 | 15일 | 1개월 | 3개월 | 1년 | 2024-12-01 ~ 2024-12-31 조회

게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412310001	1대1문의 드립니다.	0	admintest123	2024-12-31 16:44:42	답변필요

1

글쓰기

24. 게시물 - 1:1문의 - 상세보기

고객센터

1:1문의

조회기간 오늘 7일 15일 1개월 3개월 1년 2024-12-01 ~ 2024-12-31 조회

게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412310001	1대1문의 드립니다.	0	admintest123	2024-12-31 16:44:42	답변필요

01

1:1문의

본문

내용은 이렇습니다.

답변

답변 대기중...

수정 삭제 목록으로

A red arrow points from the '답변필요' status in the list view to the '답변 대기중...' status in the detailed view, indicating a transition or relationship between the two states.

Description List

01

상품 문의와 동일

25. 관리자 페이지 로그인

The screenshot illustrates the login process for an administrator page. It consists of two main parts: a login form on the left and a developer tools panel on the right.

01 The login form has a red border and contains the following fields:

- Username: admin123
- Password: (redacted)
- Login button: 로그인

02 A pink callout box highlights the login result message:

localhost:3000 내용:
로그인 성공! 환영합니다, admin123님!

03 A pink callout box highlights the session storage data in the developer tools Application tab:

Key	Value
user	admin123
auth	true
role	관리자

Description List

01 관리자 아이디 및 비밀번호 입력

02 로그인 성공 시 alert 메시지 출력

03 로그인과 동시에 세션 스토리지에 정보 저장

26. 관리자 상품 등록

상품등록

▪ 상품명	[]	
▪ 사진	[파일 선택]	선택된 파일 없음
상세이미지	[파일 선택]	선택된 파일 없음
▪ 등록 재고 수량	[] 개	
옵션	[옵션을 선택하세요 ▼]	
상세정보 키워드 설정	[옵션을 선택하세요 ▼]	
▪ 사용 여부	<input checked="" type="radio"/> ON <input type="radio"/> OFF	
혜택	[]	
▪ 상품 가격	0	원
할인율	0	%
▪ 최종 가격	0	원

Description List

01

옵션/키워드 설정

- 옵션/키워드 선택
- 하위 옵션/키워드값 체크

02

가격 계산

- 상품 가격
- 할인율
- 최종 가격

26. 관리자 상품 등록 - 옵션/키워드 설정 ○1

01

```
const [option01, setOption01] = useState([]); //옵션 목록(배열)
const [option02, setOption02] = useState(""); //상위 옵션 선택
const [keyword01, setKeyword01] = useState([]); //키워드 목록(배열)
const [keyword02, setKeyword02] = useState(""); //상위 키워드 선택
const [opVal, setOpVal] = useState([]); // 하위 옵션 값
const [keyVal, setKeyVal] = useState([]); // 하위 키워드 값
const [selectedOption, setSelectedOption] = useState([]);
const [selectedKeyword, setSelectedKeyword] = useState([]);
```

02

```
//옵션 목록 불러오기
useEffect(() => {
  selectOption(myData)
    .then((res) => {
      setOption01(res.data);
    })
    .catch((err) => {
      console.error("옵션 리스트 백엔드 연결 실패", err);
    });
}, [myData]);
```

```
const optionClick = (e) => {
  setOption02(e.target.value); // 선택 옵션 저장
};
```

Description List

01

초기값 설정

02

등록 되어있는 옵션 목록 불러오기

- 셀렉트 된 옵션 저장(키워드는 함수명만 다르고 코드 동일)

26. 관리자 상품 등록 - 옵션/키워드 설정 ○2

Description List

03

```
//옵션값 불러오기
useEffect(() => {
  if (option02) {
    // 상위 옵션이 선택된 경우에만 실행
    // 옵션 리스트에서 선택된 option02의 code 값을 찾아서 전달
    const selectedOption = option01.find(function (opt) {
      // option02와 일치하는 name 찾기
      return opt.name === option02;
    });

    if (selectedOption) {
      const { code } = selectedOption; // 선택된 옵션의 code 값 추출
      selectOptionVal(code)
        .then((res) => {
          setOpVal(res.data); // 옵션값 저장
        })
        .catch((err) => {
          console.error("옵션값 불러오기 실패", err);
        });
    }else{
      setSelectedOption([]);
    }
  }
}, [option02, option01]);
```

03

선택된 옵션/ 키워드의 등록된 하위 값들 불러오기

- 선택된 체크박스의 값이 배열에 저장
- 새로운 상위 옵션/키워드 선택 시 저장됐던 배열 리셋
(키워드는 함수명만 다르고 코드 동일)

26. 관리자 상품 등록 - 가격 계산

01

```
const [orPrice, setOrPrice] = useState(0); //원가
const [disV, setDisV] = useState(0); //할인율
const [fnPrice, setFnPrice] = useState(0); //최종가
```

02

```
function fnPriceChange() {
  //소수점 제거
  let newFnPrice = Math.round(orPrice - orPrice * (disV / 100));
  //1원단위 남지 않게 10으로 나눠서 소수점 올림 후 다시 10 곱하기
  newFnPrice = Math.round(newFnPrice / 10) * 10;

  if (orPrice && disV > 0) {
    //원가 할인율 둘 다 있을 시
    setFnPrice(newFnPrice); //계산된 가격
    if (newFnPrice < 0 || disV > 100) {
      //최종가가 마이너스거나 할인율이 100퍼 넘을 때
      alert("할인율을 다시 입력하세요.");
      return setDisV(0);
    } else if (disV == 100) {
      //할인율 100퍼 맞는지 체크
      let checConfirm = confirm("할인율 100%를 입력하신게 맞나요?");
      if (!checConfirm) {
        return setDisV(0);
      }
    }
  } else if (!disV || disV == 0) {
    //할인율 없거나 0일 시
    let newOrPrice = Math.ceil(orPrice / 10) * 10;
    if (orPrice % 10 <= 4) {
      newOrPrice = Math.round(orPrice / 10) * 10;
    }
    setFnPrice(newOrPrice); //원가 가격
  }
}
```

Description List

01

초기값 설정

02

인풋에 원가 입력 시

- 할인율 없을 시 1원 단위 남지 않게 반올림 자동 계산
- 할인율 있을 시 할인율로 가격 할인 계산 후 1원 반올림

27. 관리자 상품 목록

상품2 상세정보

상품명	상품2
사진	<input type="button" value="파일 선택"/> 선택된 파일 없음
상세이미지	<input type="button" value="파일 선택"/> 선택된 파일 없음
등록 재고 수량	300000 개
옵션	옵션1 색상
옵션값	대,중,소
키워드	키워드2 배송
키워드값	배송가능
사용 여부	<input checked="" type="radio"/> ON <input type="radio"/> OFF
혜택	
상품 가격	300000 원
할인율	0 %
최종 가격	300000 원

01

설정한 옵션/키워드 가져오기

27. 관리자 상품 목록

01

```
const [proList, setProlist] = useState({}); //초기값
// 로딩 화면(처음에 값 바로 안들려와서 설정정)
const [loading, setLoading] = useState(true);
// const [imgShow, setImgShow] = useState([]); //이미지 들고오기
const [dtList, setDtList] = useState([]); //상품디테일 들고오기
const [opList, setOpList] = useState([]); //옵션목록 들고오기
const [keyList, setKeyList] = useState([]); //키워드목록 들고오기
const [keyValList, setKeyvalList] = useState([]); //키워드값 들고오기
const [mainCode, setmainCode] = useState([]); //선택옵션
const [dtcode, setdtcode] = useState([]); //선택옵션값
const [dtBack, setDtBack] = useState({ mainCode: [], dtCode: [] }); // 상품 상세 정보
const [option01, setOption01] = useState([]); //옵션 목록(배열)
const [option02, setOption02] = useState(""); //상위 옵션 선택
const [opVal, setOpVal] = useState([]); // 하위 옵션 값
const [selectedOptions, setSelectedOptions] = useState([]); // 선택된 하위 옵션들
```

01

초기값 설정

02

상품 – 옵션/키워드와 관련된 상품 디테일 정보
가져오기

02

```
//상품 디테일 들고오기
selectDetail(main_code)
.then((res) => {
  setDtList(res.data || []);
})
.catch((err) => {
  console.error("상품 상세 백엔드 연결 실패", err);
});
```

27. 관리자 상품 목록

03

```
//옵션목록 들고오기
selectOption(myData)
  .then((res) => {
    setOpList(res.data || []);
    // console.log(res.data)
  })
  .catch((err) => {
    console.error("옵션목록 백엔드 연결 실패", err);
  }, []);
}

//옵션값 들고오기
selectOptionVal(code)
  .then((res) => {
    // console.log('옵션값 API 응답:', res); // 응답 확인
    if (res.data) {
      setOpVal(res.data); // 정상적으로 데이터가 오면 상태 업데이트
      // setOpvalList(res.data); // 정상적으로 데이터가 오면 상태 업데이트
      // console.log(res.data)
    } else {
      console.error("옵션값 데이터가 비어 있습니다.");
    }
  })
  .catch((err) => {
    console.error("옵션값 API 호출 실패:", err); // 에러 로그
  });
}
```

03

옵션목록/옵션값 들고오기

(키워드도 함수명만 다르고 코드 동일)

28. 옵션 관리(1)

Description List

The screenshot shows the 'Option Management (1)' interface. At the top, there are dropdown menus for '필수여부' (Required), '전체' (All), '사용여부' (Usage), and '전체' (All). A search bar labeled '옵션명' (Option Name) with a magnifying glass icon is followed by a '조회' (Search) button. Below the search area is a table with columns: 번호 (Number), 옵션명 (Option Name), 필수선택 (Required Selection), and 사용여부 (Usage). The table contains several rows of data. To the right of this table is another table with columns: 번호 (Number), 옵션값 명 (Option Value Name), 가격 (Price), 사용여부 (Usage), 수정자 (Modifier), and 수정일시 (Modification Date). This second table also contains six rows of data. At the bottom left, there is a navigation bar with page numbers 1, 2, 3, 4, and a next arrow.

번호	옵션명	필수선택	사용여부
2	색상	필수O	사용
6	질감	필수X	사용
35	고급	필수X	사용
36	신규	필수X	미사용
37	현색	필수X	미사용

번호	옵션값 명	가격	사용여부	수정자	수정일시
1	노광	10000	사용	dc4638	2024-12-12
2	빨강	20000	사용	dc4638	2024-12-12
3	초록	30000	사용	dc4638	2024-12-12
4	노랑	40000	사용	dc4638	2024-12-12
5	파랑	50000	사용	dc4638	2024-12-12
6	현색	15000	사용	dc4638	2024-12-12

01 검색 필터

- 필수여부 : 전체, 필수O, 필수X
- 사용여부 : 전체, 사용, 미사용
- 옵션명

02 검색된 옵션 리스트

- 번호
- 옵션명
- 필수선택
- 사용여부

03 상위 리스트 행추가, 저장

04 페이지

28. 옵션 관리(2)

Description List

The screenshot shows a web-based application for managing options. At the top, there are dropdown filters for '필수여부' (Mandatory), '전체' (All), '사용여부' (Usage), and '전체' (All). Below these are search fields for '옵션명' (Option Name) and a search button '조회' (Search). A pink circle labeled '01' points to the '옵션명' search field.

Below the search bar, there are two buttons: '행 추가' (Add Row) and '저장' (Save). A pink circle labeled '03' points to the '행 추가' button. To the right of the buttons, a pink circle labeled '06' points to the '행 추가' and '저장' buttons.

The main area contains two tables. The left table, labeled '02', has columns: 번호 (Number), 옵션명 (Option Name), 필수선택 (Mandatory Selection), and 사용여부 (Usage). It lists items 2, 6, 35, 36, and 37. The right table, labeled '05', has columns: 번호 (Number), 옵션값 명 (Option Value Name), 가격 (Price), 사용여부 (Usage), 수정자 (Modifier), and 수정일시 (Modification Date). It lists items 1 through 6. Both tables include a '조회' (Search) button at the top right and a set of navigation buttons at the bottom.

번호	옵션명	필수선택	사용여부
2	색상	필수O	사용
6	질감	필수X	사용
35	고급	필수X	사용
36	신규	필수X	미사용
37	기타	필수X	미사용

번호	옵션값 명	가격	사용여부	수정자	수정일시
1	노痈	10000	사용	dc4638	2024-12-12
2	빨강	20000	사용	dc4638	2024-12-12
3	초록	30000	사용	dc4638	2024-12-12
4	노랑	40000	사용	dc4638	2024-12-12
5	파랑	50000	사용	dc4638	2024-12-12
6	현색	15000	사용	dc4638	2024-12-12

05 선택된 옵션 리스트 하위 값 (스크롤 처리)

- 번호
- 옵션값 명
- 가격
- 사용여부
- 수정자
- 수정일시

06 하위 리스트 행추가, 저장

28. 옵션 페이지 컴포넌트화(1)

Description List

01

옵션 관리

필수여부 전체 사용여부 전체 옵션명 조회

02

번호	옵션명	필수선택	사용여부	행 추가	저장
2	색상	필수O	사용		
6	질감	필수X	사용		
35	고급	필수X	사용		
36	신규	필수X	미사용		
37	아이고	필수X	미사용		

03

번호	옵션값 명	가격	사용여부	수정자	수정일시
1	노랑	10000	사용	dc4638	2024-12-12
2	빨강	20000	사용	dc4638	2024-12-12
3	초록	30000	사용	dc4638	2024-12-12
4	노랑	40000	사용	dc4638	2024-12-12
5	파랑	50000	사용	dc4638	2024-12-12
6	현색	15000	사용	dc4638	2024-12-12

1 2 3 4 >

01 전체 소스

02 옵션 관리 화면 컴포넌트화

03 옵션값 관리 화면 컴포넌트화

28. 옵션 페이지 컴포넌트화(2)

Description List

01

01

전체 소스

```
1 <div className="inquiry-section">
2   <div className="filter-header">
3     <h1>옵션 관리</h1>
4   </div>
5   <form name="myFrm">
6     <div className="mypage-date-check">
7       <div>
8         <span className="mypage-period-text">필수여부</span>
9         <select
10           name="reqe"
11           className="color-option"
12           value={selectedReq}
13           onChange={(e) => setSelectedReq(e.target.value)}>
14           <option value="">전체</option>
15           <option value="1">필수O</option>
16           <option value="0">필수X</option>
17         </select>
18       </div>
19     </div>
20     <div>
21       <span className="mypage-period-text">사용여부</span>
22       <select
23         name="status"
24         className="color-option"
25         value={selectedStatus}
26         onChange={(e) => setSelectedStatus(e.target.value)}>
27           <option value="">전체</option>
28           <option value="1">사용</option>
29           <option value="0">미사용</option>
30         </select>
31       </div>
32     </div>
33     <div>
34       <span className="mypage-period-text">옵션명</span>
35       <input type="text" name="name" />
36     </div>
37     <div className="mypage-date-search">
38       <button className="mypage-date-search-btn" onClick={searchOptions}>
39         조회
40       </button>
41     </div>
42   </form>
43   <div className={styles.table_container}>
44     <div className={styles.head_container}>
45       <HeadContainer
46         options={options}
47         searchOptions={searchOptions}
48         setOptionValues={setOptionValues}
49         setOptions={setOptions}
50         setSelectedHeaderCode={setSelectedHeaderCode}>
51       </HeadContainer>
52     </div>
53     <div className={styles.detail_container}>
54       <DetailContainer
55         optionValues={optionValues}
56         setOptionValues={setOptionValues}
57         selectedHeaderCode={selectedHeaderCode}>
58       </DetailContainer>
59     </div>
60   </div>
61 </div>
62 </div>
63 </div>
```

28. 옵션 페이지 컴포넌트화(3)

02

```
1 currentItems.map((item, index) => (
2   <tr
3     key={index}
4     onClick={() => handleRowClick(item.code, index)}
5     className={getRowClassName(index)}
6   >
7     <td>{item.code}</td>
8     <td>
9       <input
10         type="text"
11         value={item.name}
12         onChange={(e) => handleReqChange(e, item, "name")}
13         required
14         placeholder="옵션 명"
15       />
16     </td>
17     <td>
18       <select
19         value={item.req}
20         onChange={(e) => handleReqChange(e, item, "req")}
21       >
22         <option value="0">필수X</option>
23         <option value="1">필수O</option>
24       </select>
25     </td>
26     <td>
27       <select
28         value={item.status}
29         onChange={(e) => handleReqChange(e, item, "status")}
30       >
31         <option value="0">미사용</option>
32         <option value="1">사용</option>
33       </select>
34     </td>
35   </tr>
36 ))
```

Description List

02

옵션 관리 화면 컴포넌트화

28. 옵션 페이지 컴포넌트화(4)

Description List

03

```
1 <tr key={index}>
2   <td>{item.code}</td>
3   <td>
4     <input
5       type="text"
6       value={item.name}
7       onChange={(e) => handleReqChange(e, item, "name")}
8       placeholder="옵션값 명"
9     />
10    </td>
11    <td>
12      <input
13        type="number"
14        value={item.opVal_price}
15        onChange={(e) => handleReqChange(e, item, "opVal_price")}
16        placeholder="가격"
17      />
18    </td>
19    <td>
20      <select
21        value={item.status}
22        onChange={(e) => handleReqChange(e, item, "status")}
23      >
24        <option value="0">미사용</option>
25        <option value="1">사용</option>
26      </select>
27    </td>
28    <td>{item.upt_id}</td>
29    <td>{item.upt_date}</td>
30  </tr>
```

03

옵션값 관리 화면 컴포넌트화

29. 키워드 관리(1)

Description List

The screenshot shows a keyword management interface with the following elements:

- Top Filter Bar:** Includes dropdowns for 필수여부 (필수O), 사용여부 (전체), 키워드명, and a search button (조회).
- Left Panel:** Shows a list of keywords with columns: 번호, 키워드명, 필수선택, and 사용여부. Each row has dropdowns for 키워드명, 필수여부, and 사용여부.
- Right Panel:** Shows a list of keyword values with columns: 번호, 키워드값 명, 가격, 사용여부, 수정자, and 수정일시. Each row has dropdowns for 키워드값 명, 가격, 사용여부, and 수정자.
- Bottom:** Includes a page number indicator (1) and navigation buttons (prev, next).

01 검색 필터

- 필수여부 : 전체, 필수O, 필수X
- 사용여부 : 전체, 사용, 미사용
- 키워드명

02 검색된 옵션 리스트

- 번호
- 키워드 명
- 필수선택
- 사용여부

03 상위 리스트 행추가, 저장

04 페이지

29. 키워드 관리(2)

Description List

The screenshot shows the 'Keyword Management (2)' interface. At the top, there are search filters for '필수여부' (Mandatory), '전체' (All), '사용여부' (Usage), '전체' (All), '키워드명' (Keyword Name), and a search button '조회' (Search). Below the filters is a table with three rows of keyword data. The first row (45) has '1' in the keyword field, '필수X' in the mandatory dropdown, and '미사용' in the usage dropdown. The second row (46) has '의자' in the keyword field, '필수X' in the mandatory dropdown, and '사용' in the usage dropdown. The third row (47) has '책상' in the keyword field, '필수X' in the mandatory dropdown, and '사용' in the usage dropdown. A pink circle labeled '04' points to the page navigation buttons at the bottom left. To the right, another table lists four keyword values. Each row contains: 번호 (Number), 키워드값 명 (Value Name), 가격 (Price), 사용여부 (Usage), 수정자 (Modifier), and 수정일시 (Modification Date). The values are: 1 (스툴의자), 10000, 사용, dc4638, 2024-12-12; 2 (의자없음), 0, 사용, dc4638, 2024-12-12; 3 (이건테스트), 50000, 사용, dc4638, 2024-12-12; 4 (테스트2), 200, 사용, dc4638, 2024-12-12.

번호	키워드명	필수선택	사용여부
45	1	필수X	미사용
46	의자	필수X	사용
47	책상	필수X	사용

번호	키워드값 명	가격	사용여부	수정자	수정일시
1	스툴의자	10000	사용	dc4638	2024-12-12
2	의자없음	0	사용	dc4638	2024-12-12
3	이건테스트	50000	사용	dc4638	2024-12-12
4	테스트2	200	사용	dc4638	2024-12-12

- 05 선택된 옵션 리스트 하위 값 (스크롤 처리)
- 번호
 - 키워드값 명
 - 가격
 - 사용여부
 - 수정자
 - 수정일시

06 하위 리스트 행추가, 저장

30. 주문 관리

Description List

주문 관리 페이지

The screenshot shows the Order Management Page with the following components:

- 01**: Search filters at the top: Date range (연도-월-일), Order status (주문 상태: 전체), Payment method (결제 방법: 전체), Purchase confirmation (구매 확정: 전체), Search button (조회), and Reset button (초기화).
- 02**: Order list table with columns: 주문 ID, 회원 ID, 상태, 결제방법, 구매확정, 총 금액, 날짜, and 상태 변경 (Status Change). The table contains five rows of order data.
- 03**: Pagination at the bottom left with page numbers 1, 2, 3.
- 04**: Status change buttons for each row: 배송시작 (Delivery Start) and 배송완료 (Delivery Complete).

주문 ID	회원 ID	상태	결제방법	구매확정	총 금액	날짜	상태 변경
1	user07	배송준비	신용카드	N	202,224원	2024-12-29T06:18:45.000Z	<button>배송시작</button>
11	user07	배송완료	신용카드	Y	120,000원	2023-12-07T07:00:00.000Z	
7	user07	배송완료	신용카드	N	100,000원	2023-12-03T03:00:00.000Z	
5	user07	배송완료	신용카드	Y	50,000원	2023-12-01T01:00:00.000Z	
9	user07	배송중	신용카드	Y	75,000원	2023-09-05T05:00:00.000Z	<button>배송완료</button>

01 검색 필터: 조회 기간

주문상태: 배송준비, 배송중, 배송완료

결제방법: 신용카드, 휴대폰

구매확정: 확정, 미확정

02 검색된 주문리스트

- 주문ID

- 회원ID

- 주문상태

- 결제방법

- 구매확정

- 총금액

- 날짜

- 상태변경

03 페이징처리

04 주문상태에 따른 변경버튼 생성

배송준비 상태일시 배송시작버튼 생성

배송중 상태일시 배송완료버튼 생성

30. 주문 관리

01

```
const [orders, setOrders] = useState([]);  
const [totalOrders, setTotalOrders] = useState(0);  
const [selectedStatus, setSelectedStatus] = useState("전체");  
const [selectedPayMethod, setSelectedPayMethod] = useState("전체");  
const [selectedConfirmation, setSelectedConfirmation] = useState("전체");  
const [searchTerm, setSearchTerm] = useState("");  
const [currentPage, setCurrentPage] = useState(1);  
const [startDate, setStartDate] = useState("");  
const [endDate, setEndDate] = useState("");  
const [filters, setFilters] = useState(null); // 조회 버튼 눌렀을 때의 필터 저장
```

02

```
{order.status === "배송중" && (  
  <button  
    onClick={() =>  
      handleStatusChange(order.order_id, "배송완료")  
    }  
  >  
    배송완료  
  </button>  
)}
```

Description List

01

검색필터 초기값 설정

02

주문상태에 따른 변경버튼생성

30. 주문 관리

01

```
{orders.map((order) => (
  <tr key={order.order_id}>
    <td>{order.order_id || "N/A"}</td>
    <td>{order.member_id || "N/A"}</td>
    <td>{order.status || "N/A"}</td>
    <td>{order.payMethod || "N/A"}</td>
    <td>{order.isConfirmed || "N/A"}</td>
    <td>
      {order.total ? `${order.total.toLocaleString()}원` : "N/A"}
    </td>
    <td>{order.order_date || "N/A"}</td>
```

02

```
const statuses = ["전체", "배송준비", "배송중", "배송완료"];
const payMethods = ["전체", "휴대폰", "신용카드"];
const confirmations = ["전체", "구매확정", "구매미확정"];
```

Description List

01

주문정보 출력

02

검색필터 설정

31. 환불 관리

Description List

환불 관리 페이지

The screenshot shows a user interface for managing refunds. At the top, there are search filters for '기간' (Date Range) and '상태' (Status), a search button, and a reset button. Below the filters is a table listing five refund requests. Each row contains columns for '환불 ID', '회원 ID', '상태', '총금액', '날짜', and '액션'. The '액션' column for each row contains two buttons: '승인' (Approve) and '거절' (Reject). At the bottom of the table is a pagination bar with pages 1 and 2.

환불 ID	회원 ID	상태	총금액	날짜	액션
28	user07	취소완료	304,000원	2024-12-30T02:39:49.000Z	04 승인 거절
26	user07	반품요청	534,000원	2024-12-29T18:23:19.000Z	04 승인 거절
16	user06	반품요청	32,000원	2023-12-12T02:30:00.000Z	04 승인 거절
14	user06	취소완료	40,000원	2023-12-10T10:00:00.000Z	
8	user07	반품완료	45,000원	2023-11-04T04:00:00.000Z	

01 검색 필터: 조회 기간

주문상태: 취소요청, 취소완료, 반품요청, 반품완료

02 검색된 주문리스트

- 환불ID(주문ID)
- 회원ID
- 주문상태
- 총금액
- 날짜
- 상태변경

03 페이징처리

04 주문상태에 따른 변경버튼 생성

승인 버튼 클릭시 해당 요청이 완료되며
저장된 데이터값이 변경됨

27. 환불 관리

Description List



거절버튼 클릭시 거절사유 모달창 뜨게됨



거절사유 입력후 확인버튼 클릭시
상태값이 거절됨으로 변경됨

31. 환불 관리

01

```
const [modalIsOpen, setModalIsOpen] = useState(false); // 모달 상태
const [selectedRefundId, setSelectedRefundId] = useState(null); // 선택한 환불 ID
const [rejectionReason, setRejectionReason] = useState(""); // 거절 사유
const [rejectedRefunds, setRejectedRefunds] = useState({}); // 거절된 환불 저장

const itemsPerPage = 5; // 한 페이지당 데이터 개수
const statuses = ["전체", "취소요청", "취소완료", "반품요청", "반품완료"];
const rejectionReasons = ["재고 부족", "상품 손상", "기타"];
```

02

```
const [selectedStatus, setSelectedStatus] = useState("전체");
const [currentPage, setCurrentPage] = useState(1); // 현재 페이지
const [startDate, setStartDate] = useState(""); // 시작일
const [endDate, setEndDate] = useState(""); // 종료일
const [filters, setFilters] = useState(null); // 필터 저장
```

Description List

01

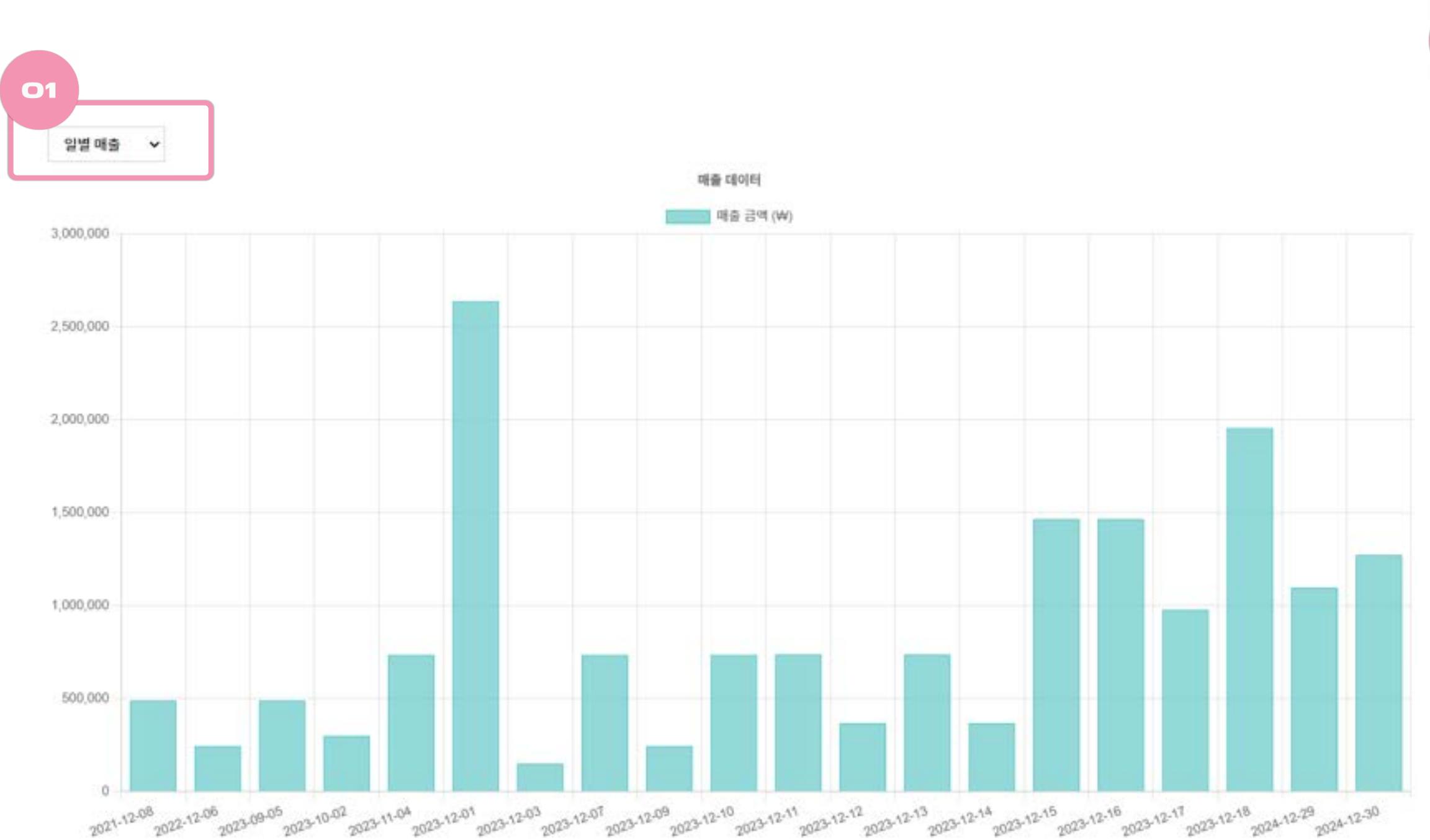
거절사유모달과 거절사유와 관련된 코드

02

필터에 대한 코드

32. 정산 관리

Description List



01

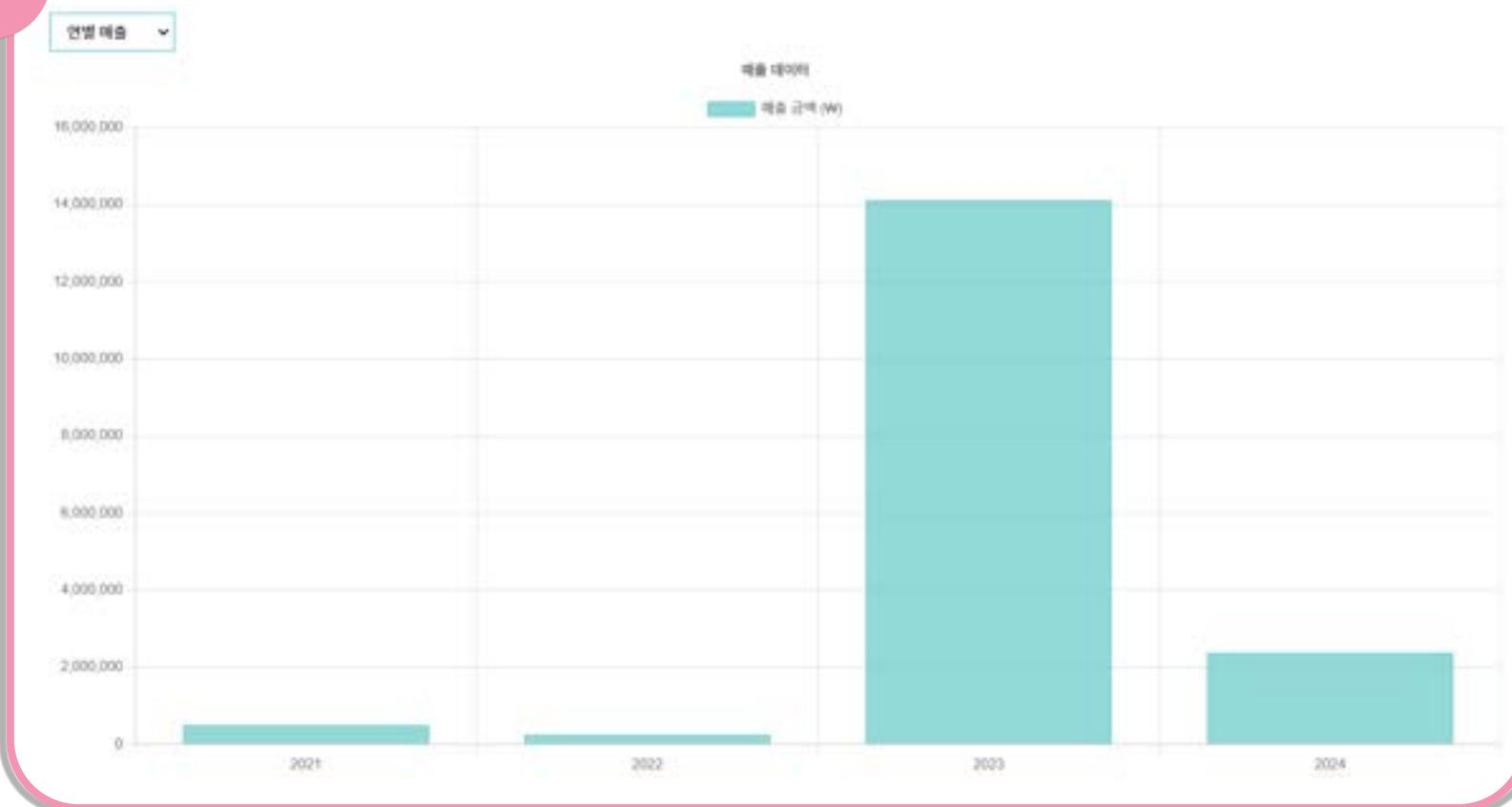
필터 : 일별, 월별, 연별, 상품별

02

검색필터 설정

32. 정산 관리

01



Description List

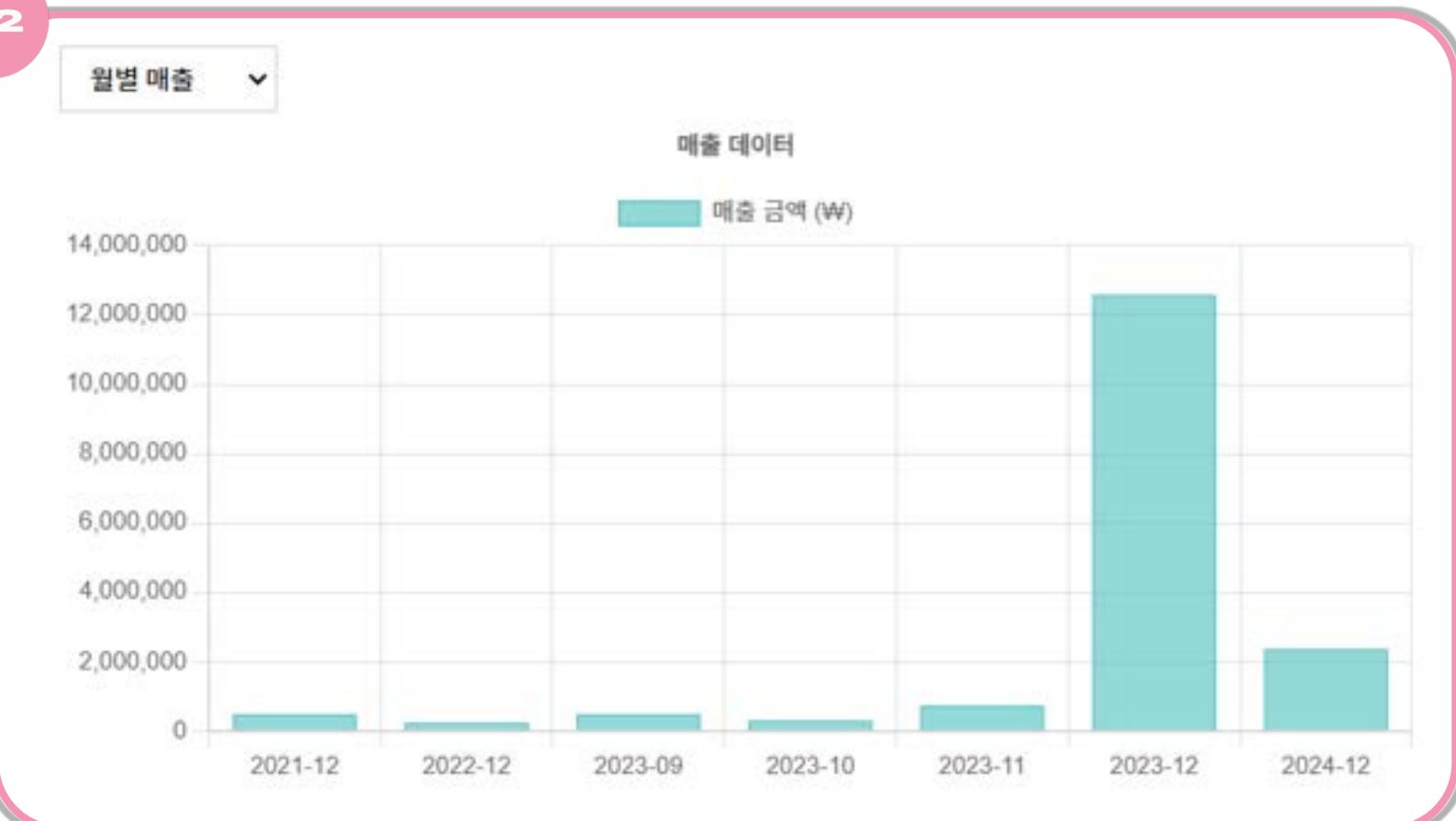
01

월별 정산조회

02

연별 정산조회

02



32. 정산 관리

Description List

01

```
if (type === 'daily') {
  query = `
    SELECT
      DATE_FORMAT(o.order_date, '%Y-%m-%d') AS date,
      SUM(op.quantity * op.unit_price) AS sales
    FROM \`order\` o
    JOIN order_product op ON o.order_id = op.order_id
    GROUP BY DATE_FORMAT(o.order_date, '%Y-%m-%d')
    ORDER BY date ASC;
  `;
} else if (type === 'monthly') {
  query = `
    SELECT
      DATE_FORMAT(o.order_date, '%Y-%m') AS date,
      SUM(op.quantity * op.unit_price) AS sales
    FROM \`order\` o
    JOIN order_product op ON o.order_id = op.order_id
    GROUP BY DATE_FORMAT(o.order_date, '%Y-%m')
    ORDER BY date ASC;
  `;
}
```

01

기간별 필터에 대한 쿼리문

02

필터에 대한 코드

02

```
<div>
  <select value={type} onChange={(e) => setType(e.target.value)}>
    <option value="daily">일별 매출</option>
    <option value="monthly">월별 매출</option>
    <option value="yearly">연별 매출</option>
    <option value="product">상품별 매출</option>
  </select>
  <Bar data={chartData} options={chartOptions} />
</div>
```

33. 게시판 답변 관리

Description List

게시판 답변 관리					
게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412240001	내가쓴 1:1문의	4	admintest1	2024-12-24 18:02:39	답변필요
bd202412240002	내가쓴 1:1문의	2	admintest1	2024-12-24 18:02:40	답변필요
bd202412240003	에잉?	1	admintest1	2024-12-24 18:02:50	답변필요
bd202412240004	▲ㄷㄴㅅ1	0	admintest1	2024-12-24 18:03:01	답변완료
bd202412240006	▲ㄷㄴㅅ1	0	admintest1	2024-12-24 18:04:40	답변필요
bd202412240007	1111111111	1	admintest1	2024-12-24 18:04:49	답변필요
bd202412260001	test1	1	admintest1	2024-12-26 17:02:47	답변완료
bd202412260002	test1	0	admintest1	2024-12-26 17:02:59	답변필요
bd202412260003	tteesstt11	0	admintest1	2024-12-26 17:07:44	답변필요
bd202412260004	1	0	admintest1	2024-12-26 17:33:51	답변필요

01 검색 필터

- 조회 기간
- 구분 : 전체후기, 포토후기, 상품문의, 1:1문의

02 검색된 등록 게시판 리스트

- 게시판 번호
- 제목
- 조회수
- 작성자
- 등록일자
- 답변여부

03 페이징

34. 게시판 등록 관리

Description List

게시판번호	제목	조회수	작성자	등록일자	수정자	수정일자	삭제
bd202412300001	필독 부탁드립니다.	0	admin123	2024-12-30 17:22:03	admin123	2024-12-30 17:22:03	활성
bd202412300002	공지사항입니다.	0	admin123	2024-12-30 17:23:04	admin123	2024-12-30 17:23:04	활성
bd202412300003	또 공지사항입니다.	0	admin123	2024-12-30 17:23:26	admin123	2024-12-30 17:23:26	삭제
bd202412300004	판매에 대한 공지	0	admin123	2024-12-30 17:23:52	admin123	2024-12-30 17:23:52	활성
bd202412300005	계속되는 공지	0	admin123	2024-12-30 17:24:06	admin123	2024-12-30 17:24:06	활성
bd202412300006	신기한 공지	0	admin123	2024-12-30 17:24:32	admin123	2024-12-30 17:24:32	삭제
bd202412300007	공지사항1	0	admin123	2024-12-30 17:24:40	admin123	2024-12-30 17:24:40	활성
bd202412300008	공지사항2	0	admin123	2024-12-30 17:24:44	admin123	2024-12-30 17:24:44	활성
bd202412300009	공지사항3	0	admin123	2024-12-30 17:24:47	admin123	2024-12-30 17:24:47	활성

01 검색 필터

- 조회 기간
- 구분 : 공지사항, FAQ
- 제목
- 작성자

02 검색된 등록 게시판 리스트

- 게시판 번호
- 제목
- 조회수
- 작성자
- 등록일자
- 수정자
- 수정일자
- 삭제여부

03 신규 글 작성

35. 회원 정보 수정

The screenshot shows the ILOKE app interface. At the top, there is a banner with a dark background and some blurred text. Below the banner, the navigation bar includes '전체상품', '고객후기', and '고객센터'. On the right side of the header, there is a pink circular badge with the text '김수훈님, 환영합니다.' and a small profile icon. A pink callout bubble labeled '01' points to this profile icon. To the right of the profile icon is a shopping cart icon with a red notification dot showing '0'.

In the middle section, there is a large blue diamond-shaped logo with the text 'ILOKE' in white. Below the logo, the same navigation bar ('전체상품', '고객후기', '고객센터') is repeated. On the right side of this section, there is a small profile icon and a shopping cart icon with a red notification dot showing '0'.

On the left side of the main content area, there is a message: '김수훈님의 회원 등급은 일반입니다.' (Kim Soohyun's member level is General). In the center, there is a box containing a blue button labeled '회원정보수정' (Member Information Modification) which is highlighted with a pink border. To its right is a red button labeled '회원탈퇴' (Member Logout).

At the bottom, there is a section titled '진행 중인 주문 최근 30일 내에 진행중인 주문정보입니다.' (Recent orders placed within the last 30 days). This section shows four items: 배송준비 (Delivery Preparation) with value '0', 배송중 (In Transit) with value '0', 배송완료 (Delivery Completed) with value '0', and 구매확정 (Purchase Confirmation) with value '0'. To the right of these items is a table with two rows: '취소' (Cancel) with value '0건' (0 items) and '변경' (Change) with value '0건' (0 items).

Description List

- 01 사람 아이콘 클릭 시 마이 페이지로 이동
- 02 마이 페이지
- 03 회원 정보 수정 버튼

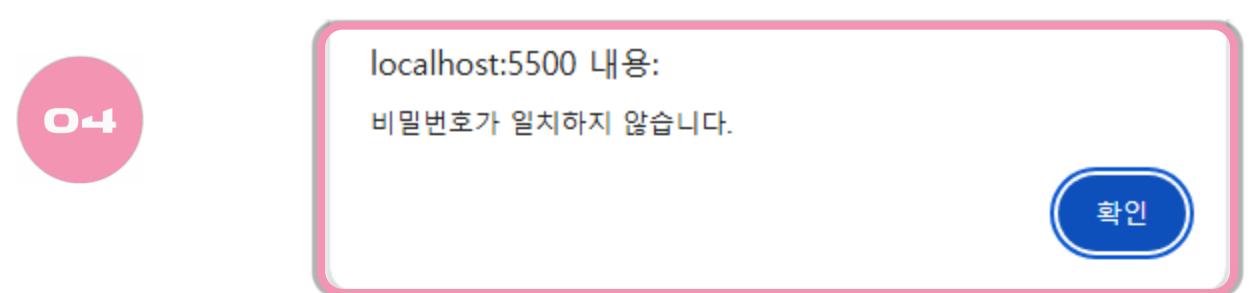
35. 회원 정보 수정

Description List

회원정보 수정

회원님의 정보를 안전하게 보호하기 위해 비밀번호를 다시 한번 확인해 주세요.

A screenshot of a web page titled "회원정보 수정". It shows a user ID "아이디 ksh1234" and a password input field labeled "비밀번호". Below the input field are two buttons: "취소" (Cancel) and "인증하기" (Verify). A pink callout bubble labeled "01" points to the password input field, and another labeled "02" points to the "취소" button.



회원정보 수정

회원님의 정보를 안전하게 보호하기 위해 비밀번호를 다시 한번 확인해 주세요.

A screenshot of a web page titled "회원정보 수정". It shows a user ID "아이디 ksh1234" and a password input field containing "....". Below the input field are two buttons: "취소" (Cancel) and "인증하기" (Verify).

01 비밀번호 입력

02 취소 버튼(메인페이지로 이동)

03 인증하기 버튼
일치 시 회원 정보 수정 화면으로 이동

04 비밀번호 일치하지 않을 시 alert 메시지 출력

35. 회원 정보 수정

회원정보 수정

01	아이디	ksh1234
02	비밀번호 수정	
03	이름	김수훈
04	닉네임	훈수김
05	이메일	<input type="text" value="ksh1234@naver.com"/> 직접 입력 <input type="button" value="▼"/>
<p>아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요.</p> <p><input type="checkbox"/> 정보/이벤트 메일 수신에 동의합니다.</p>		
06	휴대전화	<input type="text" value="010-1234-1234"/> <p>고객님과 연락에 꼭 필요한 정보이므로 정확하게 입력해주세요. xxx-xxxx-xxxx 형태로 입력해주시기 바랍니다.</p> <p><input type="checkbox"/> 정보/이벤트 SMS 수신에 동의합니다.</p>
07	주소	<input type="text" value="서울특별시 서초구 강남대로 331"/>
08	생년월일	<input type="text" value="1993-04-19"/> <input type="button" value="▼"/>

Description List

- 01 아이디(변경 불가)
- 02 비밀번호 수정(버튼 클릭 시 수정 가능)
- 03 이름(변경 불가)
- 04 닉네임(선택사항)
- 05 이메일
- 06 휴대전화
- 07 주소
- 08 생년월일(선택사항)

35. 회원 정보 수정

01

비밀번호 수정 취소

새 비밀번호

새 비밀번호 확인

02

새 비밀번호

새 비밀번호 확인

비밀번호는 영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합으로 10자 이상이어야 합니다.

비밀번호가 일치합니다.

03

새 비밀번호

새 비밀번호 확인

사용 가능한 비밀번호입니다.

비밀번호가 일치하지 않습니다.

04

새 비밀번호

새 비밀번호 확인

사용 가능한 비밀번호입니다.

비밀번호가 일치합니다.

Description List

- 01 비밀 번호 수정 창(취소 버튼 클릭 시 닫힘)
- 02 새 비밀번호가 일치하지만 유효성 충족하지 못할 시
메시지 출력
유효성은 기존 회원가입과 동일
(영문, 숫자, 특수문자 중 2가지 이상 조합 + 10자 이상)

- 03 유효성은 충족하지만 일치하지 않을 시
메시지 출력

- 04 유효성 충족 및 비밀번호 일치 시

35. 회원 정보 수정

새 비밀번호 확인

.....

비밀번호가 일치합니다.

이름

김수훈

닉네임

01

홍민소니

중복되는 닉네임입니다.

	* id varchar(50)	* pw varchar(255)	* name varchar(100)	nick varchar(50)
asdff1134	tnttn1234	니난뇨뇨	후루루루	
bbb123	12341234	김루피는바보	강아지	
ccc123	tngns0328@	손흥민	홍민소니	

이름

김수훈

닉네임

02

훈수@@

닉네임은 한글, 영문, 숫자 포함 2자 이상 10자 이하이어야 합니다.

이름

김수훈

닉네임

03

훈수

사용가능한 닉네임입니다.

Description List

01

닉네임은 선택사항이지만 입력 시 중복 확인
중복 시 사용 불가 및 메시지 출력

02

유효성 충족하지 못할 시 메시지 출력
(한글, 영문, 숫자 포함 2글자 이상 10자 이하. 특수문자 불

03

중복 닉네임 없을 시 사용 가능 및 메시지 출력

35. 회원 정보 수정

01

ksh1234@naver.com

이메일

02

직접 입력

아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요.

정보/이벤트 메일 수신에 동의합니다.

03

수훈1234@naver.com

직접 입력

이메일

이메일 형식이 올바르지 않습니다.

아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요.

정보/이벤트 메일 수신에 동의합니다.

04

ksh1234@gmail.com

gmail.com

이메일

사용 가능한 이메일입니다.

아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요.

정보/이벤트 메일 수신에 동의합니다.

Description List

01

이메일 수정 (유효성 조건은 회원가입과 동일)
로컬파트(영문자, 숫자, ._%+- 만 가능)
@ 도메인(영문자와 숫자, .- 만 가능).
최상위도메인(영문자만 허용, 최소 2글자 이상)
충족 못할 시 메시지 출력

02

Select의 도메인 선택 시 자동으로 이메일 아이디 뒤에 붙

03

유효성 충족하지 못할 시 메시지 출력

04

유효성 충족 시

35. 회원 정보 수정

Description List

01

localhost:5500 내용:

휴대전화 번호는 xxx-xxxx-xxxx 형식으로 입력해주세요.

확인

휴대전화

01043214321

고객님과 연락에 꼭 필요한 정보이므로 정확하게 입력해주세요.
xxx-xxxx-xxxx 형태로 입력해주시기 바랍니다.

정보/이벤트 SMS 수신에 동의합니다.

주소

서울특별시 서초구 강남대로 331

생년월일

1993-04-19



취소

수정하기

01

휴대전화 번호 수정 (유효성 조건은 회원가입과 동일)
(숫자만 허용. xxx-xxxx-xxxx의 형태만 허용)
충족 못할 시 alert 메시지 출력.

35. 회원 정보 수정

03

localhost:5500 내용:
회원정보가 성공적으로 수정되었습니다.

확인

새 비밀번호

사용 가능한 비밀번호입니다.

새 비밀번호 확인

비밀번호가 일치합니다.

이름
김수훈

닉네임
훈수
사용 가능한 닉네임입니다.

이메일
ksh1234@gmail.com
gmail.com ▾
사용 가능한 이메일입니다.
아이디/비밀번호 찾기에 활용되므로 정확하게 입력해주세요.
 정보/이벤트 메일 수신에 동의합니다.

휴대전화
010-4321-4321
고객님과 연락에 꼭 필요한 정보이므로 정확하게 입력해주세요.
xxx-xxxx-xxxx 형태로 입력해주시기 바랍니다.
 정보/이벤트 SMS 수신에 동의합니다.

주소
경기도 남양주시 다산중앙로 123-4

생년월일
1993-04-19

01

취소

수정하기

02

Description List

01 취소 버튼(메인 페이지로 이동)

02 수정하기 버튼

03 수정 할 정보 중 복 확인 및 유효성 통과 시
Alert 메시지와 함께 수정 완료
(데이터베이스에 수정된 데이터로 저장)

35. 회원 정보 수정

The image consists of three vertically stacked screenshots of a mobile application interface. The top screenshot shows a header with '김수훈님, 환영합니다.' and a profile icon. Below it are navigation tabs: '전체상품', '고객후기', and '고객센터'. The middle screenshot shows a similar header and navigation, with a large 'ILOKE' logo on the left. The bottom screenshot shows a detailed view with sections for '김수훈님의 회원 등급은 일반입니다.', '마일리지 7,000점', and a summary of current orders: 배송준비 (0), 배송중 (0), 배송완료 (0), 구매확정 (0). A red box highlights the '회원탈퇴' button in the top right corner of the middle section.

Description List

- 01 사람 아이콘 클릭 시 마이 페이지로 이동
- 02 마이 페이지
- 03 회원 탈퇴 버튼

36. 회원정보 수정 핵심코드

Description List

01

회원정보 수정을 위해 사용자 정보를 조회하는 API 라우터

```
// 사용자 정보 불러오기
router.get('/:id', async (req, res) => {
  const userId = req.params.id; // URL 경로에서 사용자 ID를 추출
  const query = 'SELECT id, name, nick, email, phone_num, addr, DATE_FORMAT(birth_date, "%Y-%m-%d") AS birth_date FROM mem_info WHERE id = ?';

  try {
    // 데이터베이스에서 사용자 정보를 조회
    const [results] = await conn.query(query, [userId]);
    if (results.length > 0) {
      // 사용자 정보가 존재하면 성공 응답 반환
      res.json({ success: true, userData: results[0] });
    } else {
      res.json({ success: false, message: '사용자를 찾을 수 없습니다.' });
    }
  } catch (err) {
    console.error('사용자 정보 불러오기 오류:', err);
    res.status(500).json({ success: false, message: '서버 오류가 발생했습니다.' });
  }
});
```

36. 회원정보 수정 핵심코드

01

```
// 세션 스토리지에서 아이디 가져오기
useEffect(() => {
  const storedUserId = sessionStorage.getItem("userId");
  if (storedUserId) {
    fetchUserData(storedUserId); // 사용자 정보 가져오기
  } else {
    alert("로그인이 필요합니다.");
    navigate("/login");
  }
}, [navigate]);

// 사용자 정보 불러오기
const fetchUserData = async (userId) => {
  try {
    const response = await fetchData(userId); // 서버에서 사용자 정보 요청
    if (response.data.success) {
      setFormData(response.data.userData); // 사용자 정보를 상태에 저장
      setOgNick(response.data.userData.nick); // 기존 닉네임 저장
    } else {
      alert("사용자 정보를 불러오는데 실패했습니다.");
    }
  } catch (error) {
    alert("사용자 정보를 불러오는 중 오류가 발생했습니다.");
  }
};
```

Description List

01

로그인 상태를 확인 후 서버에서 사용자 정보를 가져와 저장하여 화면에 표시

36. 회원정보 수정 핵심코드

```
// 사용자 정보 수정
router.put('/update', async (req, res) => [
  // 클라이언트로부터 요청된 회원정보를 가져옴
  const { userId, pw, name, nick, email, phone_num, addr, birth_date } = req.body;
  // console.log(userId)

  try {
    let query; // SQL 쿼리 변수
    let params; // 쿼리 매개변수 배열

    // 비밀번호가 제공된 경우 비밀번호도 업데이트
    if (pw) {
      query = `
        UPDATE mem_info
        SET pw = ?, name = ?, nick = ?, email = ?, phone_num = ?, addr = ?, birth_date = ?
        WHERE id = ?
      `;
      params = [pw, name, nick, email, phone_num, addr, birth_date, userId];
    } else {
      // 비밀번호가 제공되지 않은 경우 비밀번호를 제외하고 업데이트
      query = `
        UPDATE mem_info
        SET name = ?, nick = ?, email = ?, phone_num = ?, addr = ?, birth_date = ?
        WHERE id = ?
      `;
      params = [name, nick, email, phone_num, addr, birth_date, userId];
    }

    // 데이터베이스 업데이트 실행
    await conn.query(query, params);
    res.json({ success: true, message: '회원정보가 수정되었습니다.' });
  } catch (err) {
    console.error('회원정보 수정 오류:', err);
    res.status(500).json({ success: false, message: '회원정보 수정 중 오류가 발생했습니다.' });
  }
]);

module.exports = router;
```

Description List

01

01 사용자 정보 수정 API 라우터

36. 회원정보 수정 핵심코드

```
// 폼 제출 시 정보 설정
const handleSubmit = async (e) => {
  e.preventDefault();

  if (showPwFields) {
    if (!checkPwValid(formData.pw)) {
      alert("비밀번호는 영문 대/소문자, 숫자, 특수문자 중 2가지 이상 조합되어야 합니다.");
      return;
    }

    if (formData.pw !== formData.pw2) {
      alert("비밀번호가 일치하지 않습니다.");
      return;
    }
  }

  const dataToSend = {
    userId: formData.id,
    pw: showPwFields ? formData.pw : undefined, // 비밀번호 필드가 보일 때만 새 비밀번호 전송
    name: formData.name,
    nick: formData.nick,
    email: formData.email,
    phone_num: formData.phone_num,
    addr: formData.addr,
    birth_date: formData.birth_date,
  };

  try {
    const response = await updateUser(dataToSend);
    if (response.data.success) {
      alert("회원정보가 성공적으로 수정되었습니다.");
      navigate("/");
    } else {
      alert("회원정보 수정 실패: " + response.data.message);
    }
  } catch (error) {
    alert("회원정보 수정 중 오류가 발생했습니다.");
  }
};
```

Description List

- 01 수정된 회원 정보 제출 함수

36. 회원정보 수정 핵심코드

```
// 닉네임 중복 확인
router.post("/", async (req, res) => {
  const { nick } = req.body; // 요청 본문에서 닉네임 추출

  try {
    // 데이터베이스에서 입력된 닉네임이 존재하는지 확인
    const [results] = await conn.query("SELECT COUNT(*) AS count FROM mem_info WHERE nick = ?", [nick]);

    // 닉네임이 존재하면 true, 없으면 false
    const isDuplicate = results[0].count > 0;
    res.json({ isDuplicate }); // 중복 여부를 클라이언트에 반환
  } catch (error) {
    res.status(500).json({ isDuplicate: false, message: "서버 오류 발생" });
  }
});
```

Description List

01

닉네임 중복 여부 확인 API 라우터

02

닉네임 중복 여부를 서버에 요청해 유효성을 검증하는
닉네임 중복 확인 함수

```
// 닉네임 중복 확인 함수
const checkNickDuplicate = async (nick) => {
  // 현재 닉네임과 같으면 중복 체크를 하지 않음
  if (nick === ogNick) {
    setNickMsg("");
    setIsNickDuplicate(false);
    return;
  }

  if (!validNick(nick)) {
    setNickMsg("닉네임은 한글, 영문, 숫자 포함 2자 이상 10자");
    setIsNickDuplicate(true);
    return;
  }
```

```
try {
  const response = await checkNickDup(nick);
  if (response.data.isDuplicate) {
    setNickMsg("중복되는 닉네임입니다.");
    setIsNickDuplicate(true)
  } else {
    setNickMsg("사용가능한 닉네임입니다.");
    setIsNickDuplicate(false);
  }
} catch (error) {
  setNickMsg("닉네임 중복 확인 중 오류가 발생했습니다.");
  setIsNickDuplicate(true);
}
```

37. 관리자 페이지 회원정보 관리

Description List

01

관리자 페이지 회원 정보 관리

01

관리자 페이지 회원목록

회원 ID: 회원 이름: 접근 권한: 전체 회원 등급: 전체

가입 시작일: 연도-월-일 연도-월-일 조회

아이디	회원명	핸드폰번호	접근권한	회원등급	가입일시
black1234	블랙리스트야	010-7979-7979	회원	브론즈	2024-12-31 22:40:49
ksh1234	김수현	010-4321-4321	탈퇴회원	브론즈	2024-12-30 13:56:03
aaa123	김덕배	010-1234-1234	회원	브론즈	2024-12-19 17:36:00
qqqq1111	큐큐큐	010-2424-1515	탈퇴회원	브론즈	2024-12-16 17:35:34
shk445255	김루피	010-8888-8888	회원	브론즈	2024-12-16 07:38:42
test123	테스트	010-8888-8888	회원	브론즈	2024-12-11 11:14:14
asdff1134	나난노노	010-1234-1234	회원	브론즈	2024-12-10 16:46:27
xxxx111	히히히	010-1234-1234	회원	브론즈	2024-12-09 15:51:22
zzz123	강동원		회원	브론즈	2024-12-03 19:34:10
admin123	Admin		관리자	브론즈	2024-12-03 16:39:44

1 2

37. 관리자 페이지 회원정보 관리

관리자 페이지 회원목록

회원 ID:	회원 이름:	접근 권한:	회원 등급:
가입 시작일: 연도-월-일 ~ 연도-월-일	조회 초기화	전체	전체

회원 목록 조회 결과

아이디	회원명	핸드폰번호	접근권한
aaa123	김덕배	010-1234-1234	회원

회원 정보 조회 및 초기화

회원 ID:	회원 이름:	접근 권한:	회원 등급:
가입 시작일: 연도-월-일 ~ 연도-월-일	조회 초기화	전체	전체

회원 목록 조회 결과

아이디	회원명	핸드폰번호	접근권한
ksh1234	김수훈	010-4321-4321	탈퇴회원
ronaldo777	김수훈	010-7777-7777	회원

Description List

- 01 아이디로 조회
- 02 회원 이름으로 조회
- 03 접근 권한으로 조회(전체, 회원, 비회원, 탈퇴회원)
- 04 회원 등급으로 조회
- 05 가입 시작일로 조회
- 06 조회 버튼
- 07 초기화(회원 전체목록 보여줌)
- 08 회원 아이디로 조회 결과
- 09 회원 이름으로 조회 결과

37. 관리자 페이지 회원정보 관리

Description List

01

회원 ID:	회원 이름:	접근 권한:	회원 등급:								
가입 시작일: 연도-월-일 ~ 연도-월-일	조회 초기화	회원	전체								
<table border="1"> <tr> <td>전체</td> <td>회원</td> </tr> <tr> <td>비회원</td> <td>접근권한</td> </tr> <tr> <td>관리자</td> <td>회원</td> </tr> <tr> <td>탈퇴회원</td> <td>회원</td> </tr> </table>				전체	회원	비회원	접근권한	관리자	회원	탈퇴회원	회원
전체	회원										
비회원	접근권한										
관리자	회원										
탈퇴회원	회원										
아이디	회원명	핸드폰번호									
black1234	블랙리스트야	010-7979-7979									
aaa123	김덕배	010-1234-1234									
shk445255	김루피	010-8888-8888									

접근 권한으로 조회 결과

02

회원 ID:	회원 이름:	접근 권한:	회원 등급:								
가입 시작일: 연도-월-일 ~ 연도-월-일	조회 초기화	전체	골드								
<table border="1"> <tr> <td>골드</td> <td>회원등급</td> </tr> <tr> <td>전체</td> <td>골드</td> </tr> <tr> <td>브론즈</td> <td>VIP</td> </tr> <tr> <td>실버</td> <td>골드</td> </tr> </table>				골드	회원등급	전체	골드	브론즈	VIP	실버	골드
골드	회원등급										
전체	골드										
브론즈	VIP										
실버	골드										
아이디	회원명	핸드폰번호	접근권한								
ronaldo777	김수훈	010-7777-7777	회원								

회원 등급으로 조회 결과

03

회원 ID:	회원 이름:	접근 권한:	회원 등급:		
가입 시작일: 2024-12-09 ~ 2024-12-11	조회 초기화	전체	전체		
<table border="1"> <tr> <td>가입일시</td> </tr> <tr> <td>2024-12-11 11:14:14</td> </tr> </table>				가입일시	2024-12-11 11:14:14
가입일시					
2024-12-11 11:14:14					
아이디	회원명	핸드폰번호	접근권한		
test123	테스트	010-8888-8888	회원		
asdffff1134	니난뇨뇨	010-1234-1234	회원		
xxxx111	히히히	010-1234-1234	회원		

가입 일시로 조회 결과

37. 관리자 페이지 회원정보 관리

Description List

관리자 페이지 회원목록

회원 ID: 회원 이름: 접근 권한: 전체

회원 등급: 전체

가입 시작일: 연도-월-일 □ - 연도-월-일 □ 조회 초기화

ID	회원명	핸드폰번호	접근권한	회원등급	가입일시
black1234	블랙리스트야	010-7979-7979	회원	브론즈	2024-12-31 22:40:49
ksh1234	김수훈	010-4321-4321	탈퇴회원	브론즈	2024-12-30 13:56:03
aaa123	김덕배	010-1234-1234	회원	브론즈	2024-12-19 17:36:00
qqqq1111	큐큐큐	010-2424-1515	탈퇴회원	브론즈	2024-12-16 17:35:34
shk445255	김루피	010-8888-8888	회원	브론즈	2024-12-16 07:38:42
test123	테스트	010-8888-8888	회원	브론즈	2024-12-11 11:11:11
asdff1134	나난노노	010-1234-1234	회원	브론즈	2024-12-10 10:10:10
xxxx1111	하하하	010-1234-1234	회원	브론즈	2024-12-09 15:51:22
zzz123	김동원		회원	브론즈	2024-12-03 19:34:10
admin123	Admin		관리자	브론즈	2024-12-03 16:39:44

[회원등록](#)

01 회원 목록

02 회원 상세 정보

03 회원 마일리지 내역

회원 상세 정보

(회원 목록에서 조회 하고자 하는 회원 클릭 시 출력)

회원 상세 정보 내용:

- 이름: 김수훈
- 아이디: ksh1234
- 비밀번호: tngns1234!!
- 닉네임: 훈수
- 생년월일: 1993-04-19
- 휴대전화: 010-4321-4321
- 이메일: ksh1234@gmail.com
- 주소: 경기도 남양주시 다산중앙로 123-4
- 접근권한: 탈퇴회원
- 회원구분: 일반
- 회원등급: 브론즈
- 가입일시: 2024-12-30 13:56:03
- 최근접속일: 2024-12-31 23:55:56

마일리지 내역

순서	적립 날짜	적립 내용	유효 날짜	변동 내역
11	2024-12-31 19:53:55	구매 적립 - 소파	2025-12-31 00:00:00	50000
12	2024-12-31 19:53:55	구매 적립 - 의자	2025-12-31 00:00:00	30000
13	2024-12-31 19:53:55	마일리지 사용 - 의자	2025-12-31 00:00:00	-20000

01 회원 목록

02 회원 상세 정보

03 회원 마일리지 내역

37. 관리자 페이지 회원정보 관리

Description List

회원 상세 정보

이름 : 김수훈

아이디 : ksh1234

비밀번호 : tngns1234!!

닉네임 : 훈수

생년월일 : 1993-04-19

휴대전화 : 010-4321-4321

이메일 : ksh1234@gmail.com

주소 : 경기도 남양주시 다산중앙로 123-4

접근권한 : 탈퇴회원

회원구분 : 일반

회원등급 : 브론즈

가입일시 : 2024-12-30 13:56:03

최근접속일 : 2024-12-31 23:55:56

01

목록으로

(클릭 시 상세정보 창 사라지고 회원 목록만 보여짐)

02

수정

(클릭 시 회원 정보 수정 창 출력)

03

삭제

(클릭 시 회원 정보 삭제)

목록으로

수정

삭제

01

02

03

37. 관리자 페이지 회원정보 관리

Description List

전체

회원등급 가입일시

브론즈	2024-12-31 22:40:49
브론즈	2024-12-30 13:56:03
브론즈	2024-12-19 17:36:00
브론즈	2024-12-16 17:35:34
브론즈	2024-12-16 07:38:42
브론즈	2024-12-11 11:14:14
브론즈	2024-12-10 16:46:27
브론즈	2024-12-09 15:51:22
브론즈	2024-12-03 19:34:10
브론즈	2024-12-03 16:39:44

[회원등록](#)

01

회원 정보 수정

아이디: ksh1234

비밀번호: *****

이름: 김수훈

닉네임: 훈수

생년월일: 1993-04-19

휴대전화: 010-4321-4321

이메일: ksh1234@gmail.com

주소: 경기도 남양주시 다산중앙로 123-4

접근권한: 회원

회원구분: 일반

회원등급: 브론즈

수정자:

[뒤로](#)

02

03

[수정](#)

- 회원 정보 수정 폼

- 뒤

(회원 정보 수정 폼 사라지고 회원 목록만 출력)

- 수집

37. 관리자 페이지 회원정보 관리

01

회원 정보 수정

아이디:
ksh1234

비밀번호:
.....

이름:
김수훈

닉네임:
훈수

생년월일:
1993-04-19

휴대전화:
010-4321-4321

이메일:
ksh1234@gmail.com

주소:
경기도 남양주시 다산중앙로 123-4

접근권한:
회원

회원구분:
일반

회원등급:
브론즈

수정자:

[뒤로](#) [수정](#)

02

회원 정보 수정

아이디:
ksh1234

비밀번호:
.....

이름:
김수훈

닉네임:
훈수

생년월일:
1993-04-19

휴대전화:
010-1234-1234

이메일:
ksh1234@naver.com

주소:
경기도 남양주시 다산중앙로 777

접근권한:
관리자

회원구분:
일반

회원등급:
브론즈

수정자:

[뒤로](#) [수정](#)

Description List

01 수정 전

02 수정 후

03 수정 후 회원 상세 정보

회원 상세 정보

이름 : 김수훈
아이디 : ksh1234
비밀번호 : tngns1234!!
닉네임 : 훈수
생년월일 : 1993-04-19
휴대전화 : 010-1234-1234
이메일 : ksh1234@naver.com
주소 : 경기도 남양주시 다산중앙로 777
접근권한 : 관리자
회원구분 : 일반
회원등급 : 브론즈
가입일시 : 2024-12-30 13:56:03
최근접속일 : 2025-01-01 00:32:29

[목록으로](#) [수정](#) [삭제](#)

37. 관리자 페이지 회원정보 관리

The screenshot shows a user interface for managing member information. On the left, there is a list of members with their names and joining dates. On the right, there is a detailed form for adding a new member. The '회원등록' (Member Registration) button at the bottom left is highlighted with a red box.

회원 등록

아이디: test1234

비밀번호:*

이름: 테스트

닉네임: 테스트입니다

생년월일: 1993-04-19

휴대전화: 010-3333-3333

이메일: test1234@naver.com

주소: 서울시 강남구 테헤란로 123

접근 권한: 회원

회원 구분: 일반

회원 등급: 브론즈

작성자: 관리자

회원등록

목록으로

등록

01

02

03

04

Description List

- 01** 회원 등록
클릭 시 오른쪽에 회원 등록 폼 출력
- 02** 회원 등록 폼
목록으로
- 03** 목록으로
클릭 시 회원 등록 폼 사라지고 회원 목록만 출력
- 04** 등록
입력한 정보가 등록됨

37. 관리자 페이지 회원정보 관리

Description List

관리자 페이지 회원목록

This screenshot shows the 'Member List' page. At the top, there are search fields for 'Member ID', 'Member Name', and '접근 권한' (Access Permission) set to '전체' (All). Below these are dropdowns for '회원 등급' (Membership Level) set to '전체' and date range selection. Two buttons are present: '조회' (Search) and '초기화' (Reset). The main area displays a table of member data:

아이디	회원명	핸드폰번호	접근권한	회원등급	가입일시
test1234	테스트	010-3333-3333	회원	브론즈	2025-01-01 00:25:08
black1234	블랙리스트야	010-7979-7979	회원	브론즈	2024-12-31 22:40:49
ksh1234	김수훈	010-4321-4321	탈퇴회원	브론즈	2024-12-30 13:56:03
aaa123	김덕배	010-1234-1234	회원	브론즈	2024-12-19 17:36:00
qqqq1111	큐큐큐	010-2424-1515	탈퇴회원	브론즈	2024-12-16 17:35:34
shk445255	김루피	010-8888-8888	회원	브론즈	2024-12-16 07:38:42
test123	테스트	010-8888-8888	회원	브론즈	2024-12-11 11:14:14
asdff1134	니난뇨뇨	010-1234-1234	회원	브론즈	2024-12-10 16:46:27
xxxx1111	히히히	010-1234-1234	회원	브론즈	2024-12-09 15:51:22

회원 상세 정보

This screenshot shows the 'Member Detail' page. It displays detailed information for a member with ID 'test1234'. The information includes:

- 이름 : 테스트
- 아이디 : test1234
- 비밀번호 : tngns1234@@
- 닉네임 : 테스트입니다
- 생년월일 : 1993-04-19
- 휴대전화 : 010-3333-3333
- 이메일 : test1234@naver.com
- 주소 : 서울시 강남구 테헤란로 123
- 접근권한 : 회원
- 회원구분 : 일반
- 회원등급 : 브론즈
- 가입일시 : 2025-01-01 00:25:08
- 최근접속일 : 2025-01-01 00:25:26

At the bottom, there are three buttons: '목록으로' (Back to List), '수정' (Edit), and '삭제' (Delete).

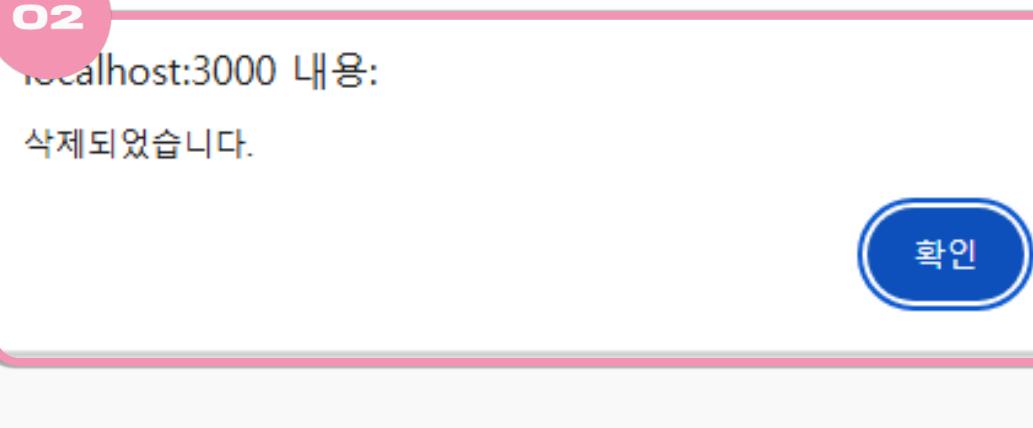
마일리지 내역

순서	적립 날짜	적립 내용	유효 날짜	변동 내역

01 회원 목록에 새로 생성 된 회원 추가

02 회원 등록으로 생성된 회원 상세 정보 조회

37. 관리자 페이지 회원정보 관리



접근권한	회원등급	가입일시
회원	브론즈	2025-01-01 00:25:08
회원	브론즈	2024-12-31 22:40:49
관리자	브론즈	2024-12-30 13:56:03
회원	브론즈	2024-12-19 17:36:00
탈퇴회원	브론즈	2024-12-16 17:35:34
회원	브론즈	2024-12-16 07:38:42

관리자 페이지 회원목록

03

회원 ID:	회원 이름:		
회원 등급:	전체		
작일:	연도-월-일	~	연도-월-일
<input type="button" value="조회"/> <input type="button" value="초기화"/>			
아이디	회원명	핸드폰번호	접근권한
test1234	테스트	010-3333-3333	회원
black1234	블랙리스트야	010-7979-7979	회원
ksh1234	김수훈	010-1234-1234	관리자

관리자 페이지 회원목록

04

회원 ID:	회원 이름:	접근 권한:	
작일:	연도-월-일	~	연도-월-일
<input type="button" value="조회"/> <input type="button" value="초기화"/>			
아이디	회원명	핸드폰번호	접근권한
black1234	블랙리스트야	010-7979-7979	회원
ksh1234	김수훈	010-1234-1234	관리자

Description List

01 삭제(회원 정보 삭제)

02 삭제 버튼 클릭 시 삭제되었다는 alert 출력

03 삭제 전 회원 목록

04 삭제 후 갱신된 회원 목록

37. 관리자 페이지 회원정보 관리 핵심코드

01

```
const [selectedId, setSelectedId] = useState(null); // 선택된 회원 ID
const [showJoin, setShowJoin] = useState(false); // 회원등록 화면 표시 여부
const [showModify, setShowModify] = useState(false); // 회원 수정 화면 표시 여부
```

02

```
import MemberDetail from "./MemberDetail"; // MemberDetail 컴포넌트 임포트
import MemberJoin from "./MemberJoin"; // MemberJoin 컴포넌트 임포트
import MemberModify from "./MemberModify"; // MemberModify 컴포넌트 임포트

/* 선택된 ID가 있고 수정 화면이 아닐 경우, 회원 상세 정보 출력 */
{selectedId && !showModify && (
  <MemberDetail
    id={selectedId}
    fetchMembers={fetchMembers}
    setSelectedId={setSelectedId}
    setShowModify={setShowModify}
  />
)}

/* 회원 수정 화면 출력 */
{showModify && (
  <MemberModify
    id={selectedId}
    setShowModify={setShowModify}
    fetchMembers={fetchMembers}
  />
)}

/* 회원 등록 화면 출력 */
{showJoin && (
  <MemberJoin fetchMembers={fetchMembers} setShowJoin={setShowJoin} />
)}
```

Description List

01

동적인 UI 상태 관리

02

회원 등록, 상세, 수정을 하나의 컴포넌트 내에서 관리 및 세가지 주요 기능을 컴포넌트화

38. 라우팅 동적 할당(1)

Description List

01

```
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import MainPage from './pages/MainPage';

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<MainPage />} />
      </Routes>
    </Router>
  );
}

export default App;
```

01

라우팅 분기

38. 라우팅 동적 할당(2)

Description List

02

```
1 import { useContent } from '../hooks/pageFunction';
2 import Footer from './Footer';
3 import Header from './Header';
4
5 function MainPage(props) {
6     const content = useContent(location.pathname); // useContent 흑호출
7
8     const isMainPage = location.pathname === '/'; // 메인 페이지 확인
9
10    return (
11        <div>
12            <Header />
13            <div id="cont" className={!isMainPage ? 'content' : ''}>
14                {content} /* 동적으로 렌더링될 컴포넌트 */
15            </div>
16            <Footer />
17        </div>
18    );
19 }
20
21 export default MainPage;
```

02

가변적 화면 분배

38. 라우팅 동적 할당(3)

Description List

03

페이지 분기 처리

03

```
1 const [content, setContent] = useState(<Home />);
2     const location = useLocation(); // 현재 경로 가져오기
3
4     useEffect(() => {
5         const path = location.pathname;
6
7         if (!path || path === "/" || path === "/home") {
8             setContent(<Home />);
9         } else if (path === "/goods/myPage") {
10             setContent(<MyPage />);
11     }
```

39. 페이징 처리

Description List

01

```
1 // 현재 페이지 데이터 슬라이싱  
2 const indexOfLastItem = currentPage * itemsPerPage;  
3 const indexOfFirstItem = indexOfLastItem - itemsPerPage;  
4 const currentItems = boards.slice(indexOfFirstItem, indexOfLastItem);
```

02

```
1 // 페이지 변경 처리  
2 const handlePageChange = (page) => {  
3   if (page >= 1 && page <= totalPages) {  
4     setCurrentPage(page);  
5   }  
6};
```

03

```
1 <PaginationControls  
2   currentPage={currentPage}  
3   totalPages={totalPages}  
4   onPageChange={handlePageChange}  
5 />
```

초기값 설정

02 페이징 변경 함수

03 페이징 출력

페이징 출력 화면



카카오 로그인



전체상품 고객후기 고객센터



Description List

회원 로그인

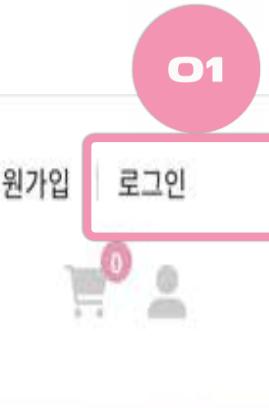
아이디
비밀번호

로그인

회원가입 아이디 찾기 비밀번호 찾기

SNS 간편 로그인

카카오 로그인



로그인 클릭

로그인 페이지 접속

카카오 로그인 클릭

카카오 로그인

01

```
<div className={styles.snsLoginSection}>
  <div className={styles.snsLoginTitle}>SNS 간편 로그인</div>
  <KakaoLoginButton />
</div>
```

02

```
useEffect(() => {
  const urlParams = new URLSearchParams(window.location.search);
  const authCode = urlParams.get('code');

  if (authCode) {
    console.log('인증 코드:', authCode);
    getAccessToken(authCode);
  } else {
    console.error('인증 코드가 없습니다.');
  }
}, []);

const getAccessToken = async (authCode) => {
  const tokenUrl = "https://kauth.kakao.com/oauth/token";
  const data = new URLSearchParams({
    grant_type: "authorization_code",
    client_id: REST_API_KEY,
    redirect_uri: REDIRECT_URI,
    code: authCode,
  });

  console.log("토큰 요청 데이터:", {
    grant_type: "authorization_code",
    client_id: REST_API_KEY,
    redirect_uri: REDIRECT_URI,
    code: authCode,
  });
}
```

Description List

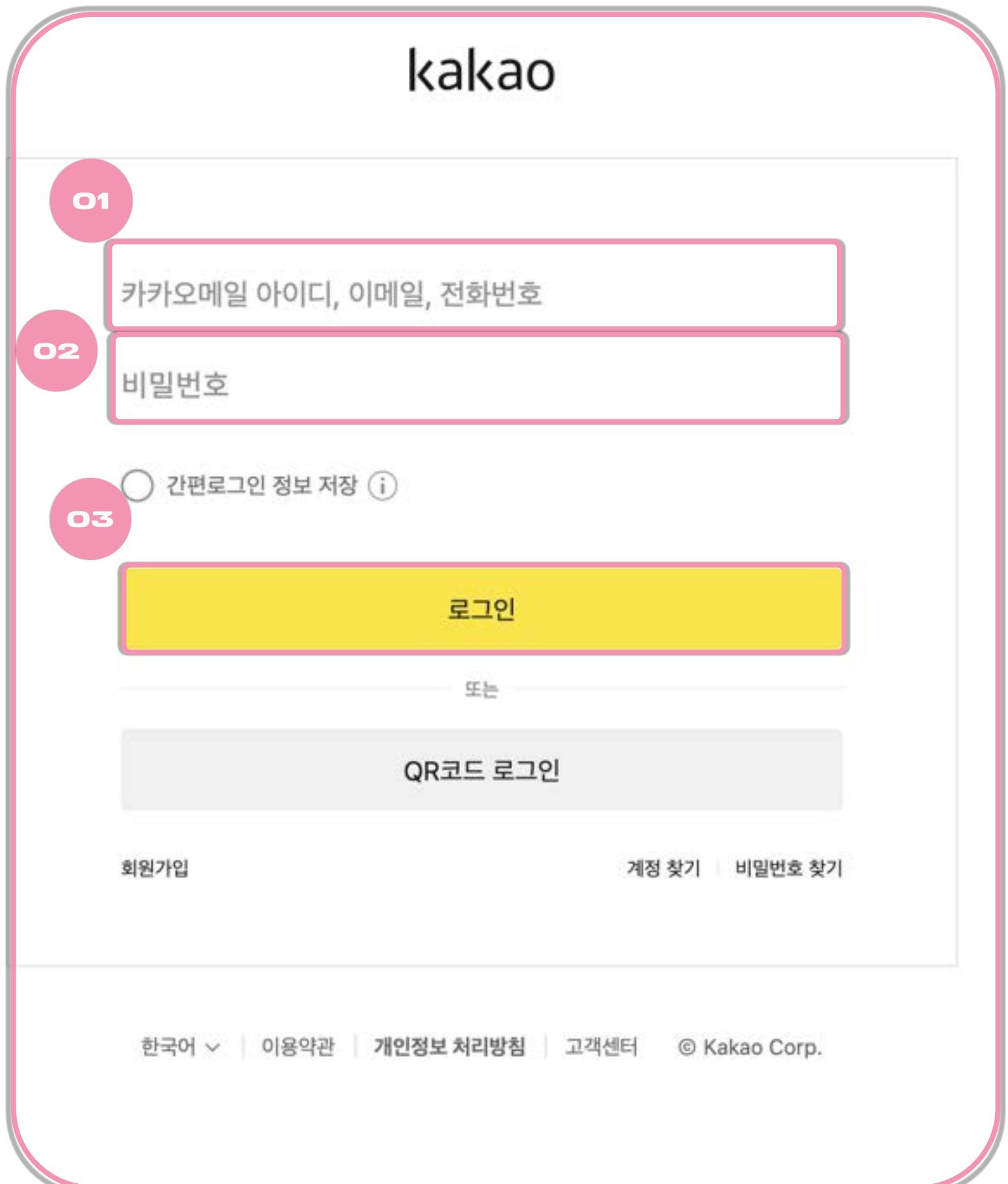
01

KakaoLoginButton 컴포넌트

02

KakaoLoginButton 처리

카카오 로그인



The image shows the Kakao login interface. At the top center is the Kakao logo. Below it are three numbered steps: 01 (Input ID/Email/Phone), 02 (Input Password), and 03 (Check Remember Me and Click Log In). A large yellow 'Log In' button is prominent. Below the button is a separator line with the word '또는' (Or) next to a QR code login button. At the bottom are links for '회원가입' (Sign Up), '계정 찾기' (Find Account), '비밀번호 찾기' (Find Password), and footer links for '한국어' (Korean), '이용약관' (Terms of Service), '개인정보 처리방침' (Privacy Policy), '고객센터' (Customer Service), and '© Kakao Corp.'

kakao

01 카카오메일 아이디, 이메일, 전화번호

02 비밀번호

03 ○ 간편로그인 정보 저장 ⓘ

로그인

또는

QR코드 로그인

회원가입 계정 찾기 비밀번호 찾기

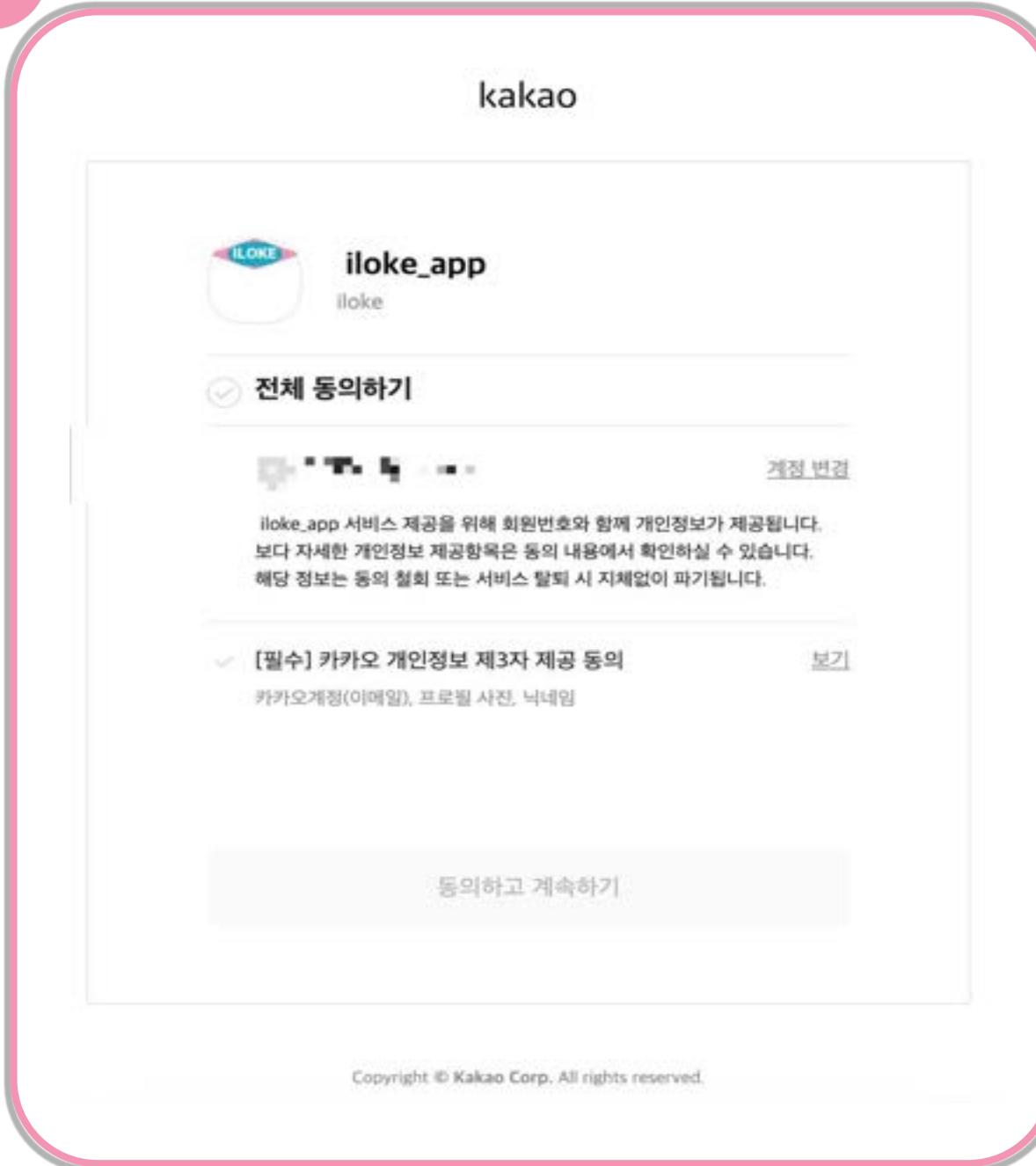
한국어 ⓘ 이용약관 개인정보 처리방침 고객센터 © Kakao Corp.

Description List

- 01 카카오 아이디,이메일,전화번호 입력
- 02 비밀번호 입력
- 03 로그인 클릭

카카오 로그인

01



Copyright © Kakao Corp. All rights reserved.

Description List

01

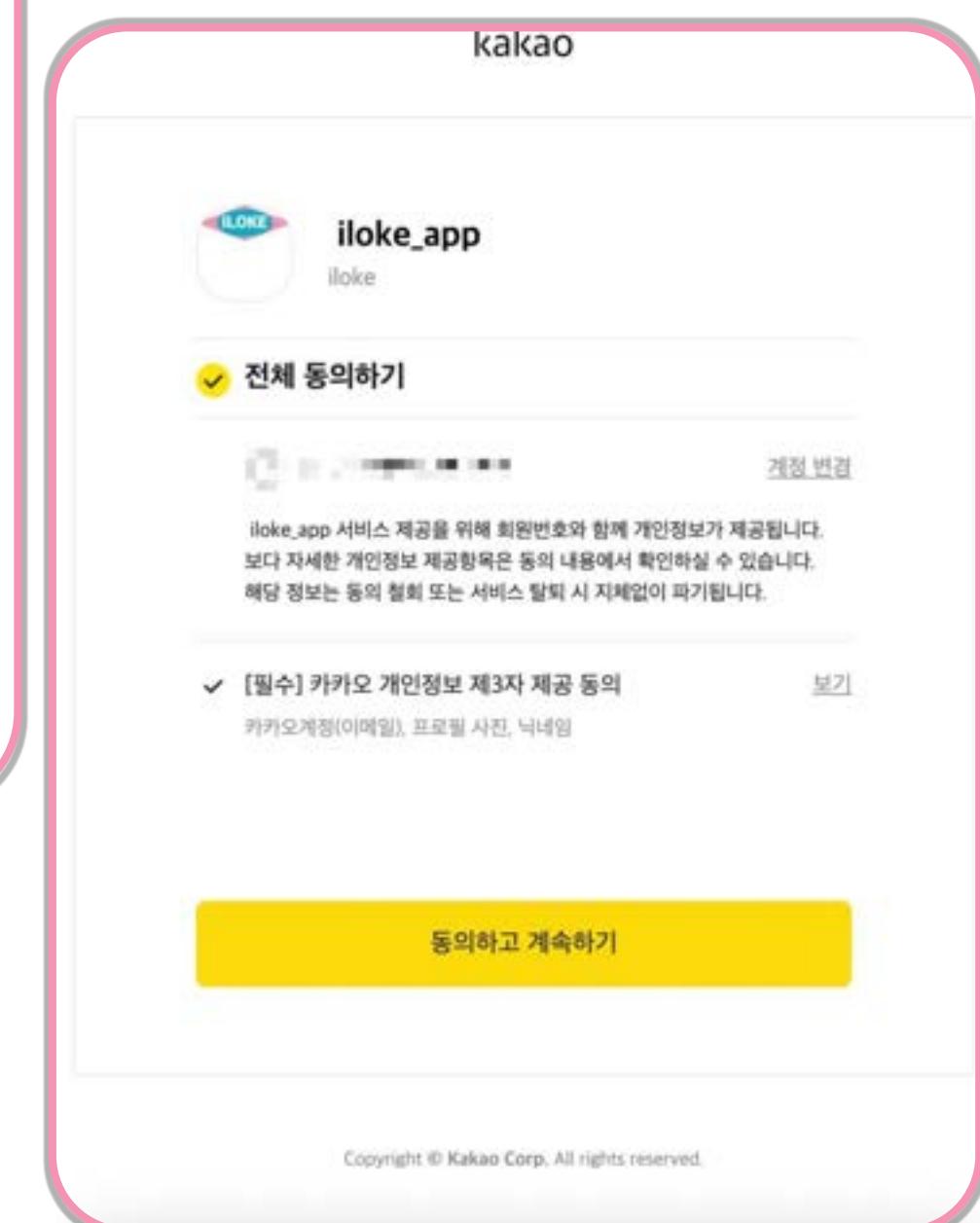
가입되지 않은 회원일 경우:

필수 항목 동의

02

동의하고 계속하기 버튼 클릭

02



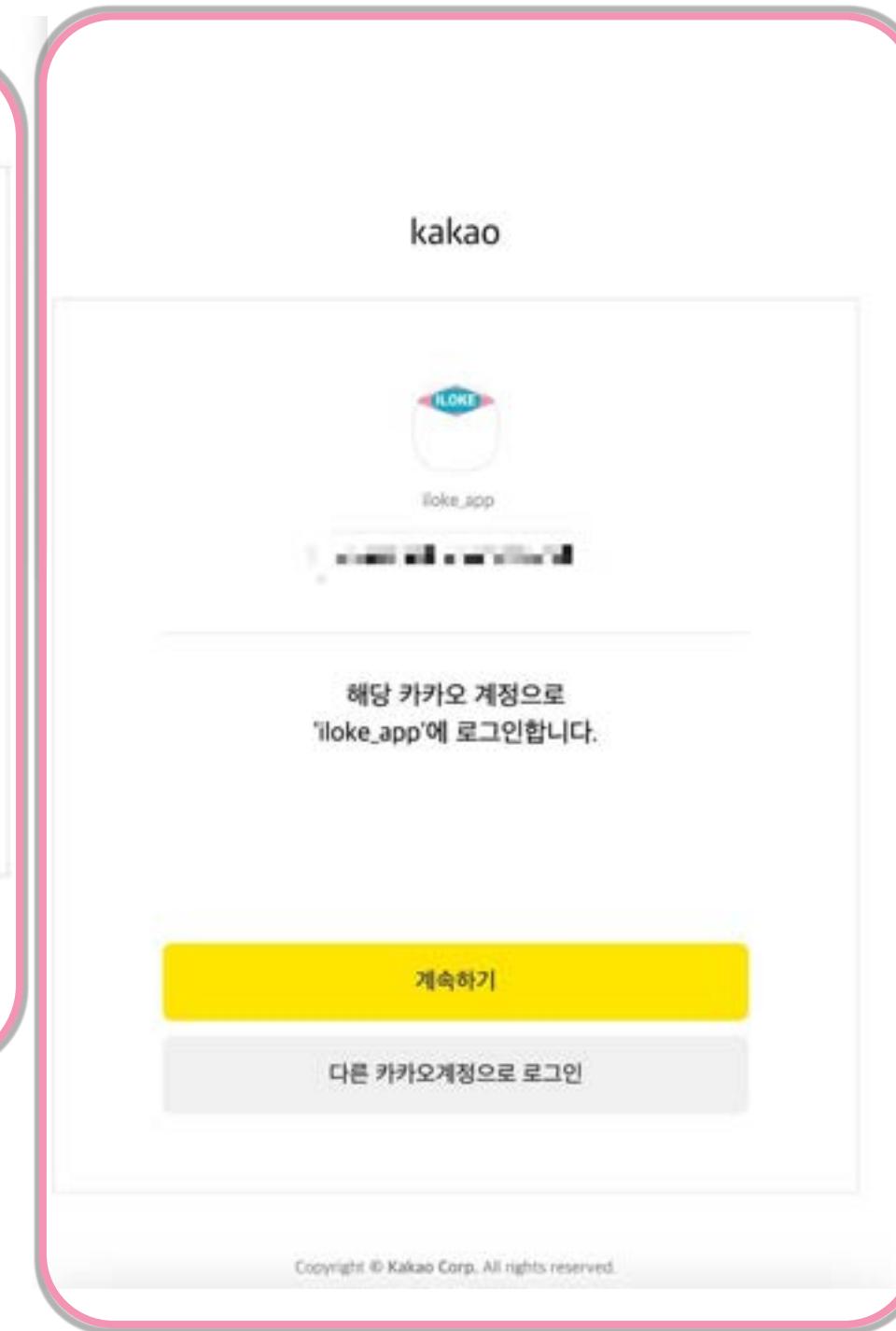
Copyright © Kakao Corp. All rights reserved.

카카오 로그인

01



02



Description List

01

가입된 회원일 경우 : 2단계 인증 (필요시)

02

인증 후 계속하기 버튼 클릭

카카오 로그인



Description List

- 01 카카오 계정 이름으로 alert창 표시
- 02 카카오 계정 이름으로 로그인
- 03 '사용자님, 환영합니다!' 문구 표시



카카오 로그인

01

```
{isLoggedIn ? (  
    <span id="welcomeMessage">{username}님, 환영합니다.</span>  
) : (
```

02

```
try {  
    const response = await fetch(tokenUrl, {  
        method: "POST",  
        headers: {  
            "Content-Type": "application/x-www-form-urlencoded",  
        },  
        body: data,  
    });  
  
    if (response.ok) {  
        const result = await response.json();  
        console.log("액세스 토큰:", result.access_token);  
        getUserInfo(result.access_token);  
    } else {  
        console.error("토큰 요청 실패:", response.statusText);  
        const errorResponse = await response.json();  
        console.error("오류 응답:", errorResponse);  
    }  
} catch (error) {  
    console.error("토큰 요청 중 오류 발생:", error);  
}
```

Description List

01

'Username님, 환영합니다.'

02

토큰 요청 처리

카카오 로그인

01



02

bd202501090002	재입고 문의할게요	1	user07	2025-01-09 12:13:58	답변필요
bd202501100001	상품2 언제나오나요	2	이아람	2025-01-10 14:35:28	답변필요
bd202501100004	상품22222	1	이아람	2025-01-10 14:39:38	답변필요

Description List

01

1:1문의 클릭

02

카카오 계정으로 작성자 이름 표시

카카오 로그인

01

```
{item.reg_id === sessionStorage.getItem('userId') && sessionStorage.getItem('userType') === 'kakao' ? sessionStorage.getItem('nickname') || '알 수 없는 사용자' // 내가 작성한 글이라면 카카오 닉네임 표시 : item.reg_id // 다른 사용자가 작성한 글은 원래 등록된 ID 표시}
```

02

```
const getUserInfo = async (accessToken) => {
  try {
    const response = await fetch("https://kapi.kakao.com/v2/user/me", {
      method: "GET",
      headers: {
        Authorization: `Bearer ${accessToken}`,
      },
    });

    if (response.ok) {
      const userInfo = await response.json();
      console.log("사용자 정보:", userInfo);
      // 로그인 상태를 세션에 저장
      sessionStorage.setItem('userId', userInfo.id); // kakaoid->userid
      sessionStorage.setItem('userType', 'kakao'); // 사용자 유형 저장
      sessionStorage.setItem('email', userInfo.kakao_account?.email || null);
      sessionStorage.setItem('nickname', userInfo.properties?.nickname || 'unknown');
      alert(`환영합니다, ${userInfo.properties?.nickname || '사용자'}님!`);
      // 필요한 경우 리디렉션
      window.location.href = '/home';
    } else {
      console.error("사용자 정보 요청 실패:", response.statusText);
    }
  } catch (error) {
    console.error("사용자 정보 요청 중 오류 발생:", error);
  }
};
```

Description List

01

카카오 로그인 후 작성한 닉네임 표시

02

getUserInfo 함수 사용하여 사용자 닉네임 가져오기

Quill 스마트에디터

Description List

The screenshot shows the 'Customer Center' section of the Quill Smart Editor. At the top, there is a navigation bar with links for '전체상품', '고객후기', and '고객센터'. On the right side of the header, there are links for '손주혜님, 환영합니다.' and '로그아웃', along with a shopping cart icon with a red notification bubble showing '0' and a user profile icon.

The main content area is titled '고객센터' and contains a sub-section titled '상품문의'. Below this, there is a search bar with date range filters from '2024-12-15' to '2025-01-14' and a '조회' button. A table lists five customer inquiries:

게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412290004	ㅋㅋㅋ	2	user07	2024-12-29 20:29:12	답변완료
bd202412300004	1	1	user07	2024-12-30 10:31:26	답변완료
bd202412300008	문의1	1	user07	2024-12-30 11:45:59	답변완료
bd202412300012	문의이벤트	1	user07	2024-12-30 15:49:57	답변완료

At the bottom of the table, there are navigation arrows for pagination, with the number '1' indicating the current page.

01

글쓰기 버튼 클릭

01

글쓰기

Quill 스마트에디터

Description List

01

```
1 <PaginationControls
2   currentPage={currentPage}
3   totalPages={totalPages}
4   onPageChange={handlePageChange}
5 />
6
7 <div>
8   <button type="button" className={styles.write_btn} onClick={() => {
9     navigate(`/notice/itemWrite`, {
10       state: { code: selectedCode, previousPage: '/notice/one_inquiry'}
11     });
12   }}>글쓰기</button>
13 </div>
```

01

글쓰기 버튼 클릭

Quill 스마트에디터

The screenshot shows the Quill Smart Editor interface. At the top, there is a navigation bar with the ILOKE logo, user information (손주혜님, 환영합니다.), and a shopping cart icon. Below the navigation bar, there are three menu items: 전체상품, 고객후기, and 고객센터. The main content area is titled "글쓰기". It includes a "상품문의" section with a "작성자" field containing "손주혜" and a "제목" field with the placeholder "제목을 입력해주세요.". A large text editor area is labeled "본문" and features a toolbar with various styling options like bold, italic, underline, superscript, and subscript. Below the toolbar, there are dropdown menus for font style ("Normal", "Sans Serif") and size ("A", "A"). At the bottom of the editor area, there are two buttons: "목록으로" and a teal-colored "저장" button.

Description List

글쓰기 스마트 에디터

Quill 스마트에디터

01

```
1 <tr>
2   <th>본문</th>
3   <td>
4     <div ref={quillRef} id="editor" className={styles.textfield}></div> /* quillRef를 추가 */
5   </td>
6 </tr>
7 {code === 'pr' ? (
8   <tr>
9     <th>첨부파일</th>
10    <td>
11      <input className={styles.file_upload} type="file" name="upfile" onChange={handleFileChange} />
12    </td>
13  </tr>
14 ) : null}
```

Description List

01

스마트 에디터 코드 장착

Quill 스마트에디터

손주혜님, 환영합니다. | 로그아웃

ILOKE

전체상품 고객후기 고객센터

상품문의

본문

B I U S „ „ % █ █ f_x H₁ H₂ █ █ █ x₁ x² █ █ Normal Heading 1 A A

Sans Serif █ T_x

01

제목

이미지 업로드

2025

답변

답변 대기중...

Description List

01

스마트 에디터로 글쓰기

Quill 스마트에디터

01

```
1 const quillRef = useRef(null);
2
3 useEffect(() => {
4     const toolbarOptions = [
5         ['bold', 'italic', 'underline', 'strike'],
6         ['blockquote', 'code-block'],
7         ['link', 'image', 'video', 'formula'],
8         [{ 'header': 1 }, { 'header': 2 }],
9         [{ 'list': 'ordered'}, { 'list': 'bullet' }, { 'list': 'check' }],
10        [{ 'script': 'sub'}, { 'script': 'super' }],
11        [{ 'indent': '-1'}, { 'indent': '+1' }],
12        [{ 'direction': 'rtl' }],
13        [{ 'size': ['small', false, 'large', 'huge'] }],
14        [{ 'header': [1, 2, 3, 4, 5, 6, false] }],
15        [{ 'color': [] }, { 'background': [] }],
16        [{ 'font': [] }],
17        [{ 'align': [] }],
18        ['clean']
19    ];
20
21    if (quillRef.current) {
22        const quill = new Quill(quillRef.current, {
23            theme: 'snow',
24            modules: {
25                toolbar: toolbarOptions
26            }
27        });
28
29        quill.on('text-change', () => {
30            const html = quill.root.innerHTML; // Quill 에디터의 HTML 콘텐츠를 추출
31            setContents(html);
32        });
33    }
34 }, []);
```

Description List

01

스마트 에디터로 글쓰기

Quill 스마트에디터

손주혜님, 환영합니다. | 로그아웃

ILOKE

전체상품 고객후기 고객센터

01

고객센터

상품문의

조회기간 오늘이전 7일 15일 1개월 3개월 1년 2024-12-15 ~ 2025-01-14 조회

게시판번호	제목	조회수	작성자	등록일자	답변여부
bd202412290004	ㅋㅋㅋ	2	user07	2024-12-29 20:29:12	답변완료
bd202412300004	1	1	user07	2024-12-30 10:31:26	답변완료
bd202412300008	문의1	1	user07	2024-12-30 11:45:59	답변완료
02412300012	문의이된다	1	user07	2024-12-30 15:49:57	답변완료
bd202501140001	스마트 에디터로 글쓰기	1	user09	2025-01-14 13:54:39	답변필요

01

< 1 >

글쓰기

Description List

스마트 에디터로 쓴 글
저장

Quill 스마트에디터

01

```
 1 const [content, setContent] = useState(contents || '');
 2 const [answerValue, setAnswerValue] = useState(comment || '');
 3 const [imageSrc, setImageSrc] = useState(null);
 4 const quillRef = useRef(null);
 5
 6 const userId = sessionStorage.getItem('userId');
 7
 8 useEffect(() => {
 9   if (code !== 'rv') {
10     imageLoad();
11   }
12
13   if (quillRef.current) {
14     const quill = new Quill(quillRef.current, {
15       theme: 'snow',
16       modules: {
17         toolbar: [
18           ['bold', 'italic', 'underline', 'strike'],
19           ['blockquote', 'code-block'],
20           ['link', 'image', 'video', 'formula'],
21           [{ 'header': 1 }, { 'header': 2 }],
22           [{ 'list': 'ordered'}, { 'list': 'bullet' }, { 'list': 'check' }],
23           [{ 'script': 'sub'}, { 'script': 'super' }],
24           [{ 'indent': '-1'}, { 'indent': '+1' }],
25           [{ 'direction': 'rtl' }],
26           [{ 'size': ['small', false, 'large', 'huge'] }],
27           [{ 'header': [1, 2, 3, 4, 5, 6, false] }],
28           [{ 'color': [] }, { 'background': [] }],
29           [{ 'font': [] }],
30           [{ 'align': [] }],
31           ['clean']
32         ]
33       }
34     });
35
36     quill.on('text-change', () => {
37       const html = quill.root.innerHTML;
38       setContent(html);
39     });
40
41     quill.root.innerHTML = contents;
42   }
43 }, [code, contents]);
```

Description List

01

스마트 에디터로 쓴 글
저장 및
글 수정하기에도 코드
추가

주문 관리



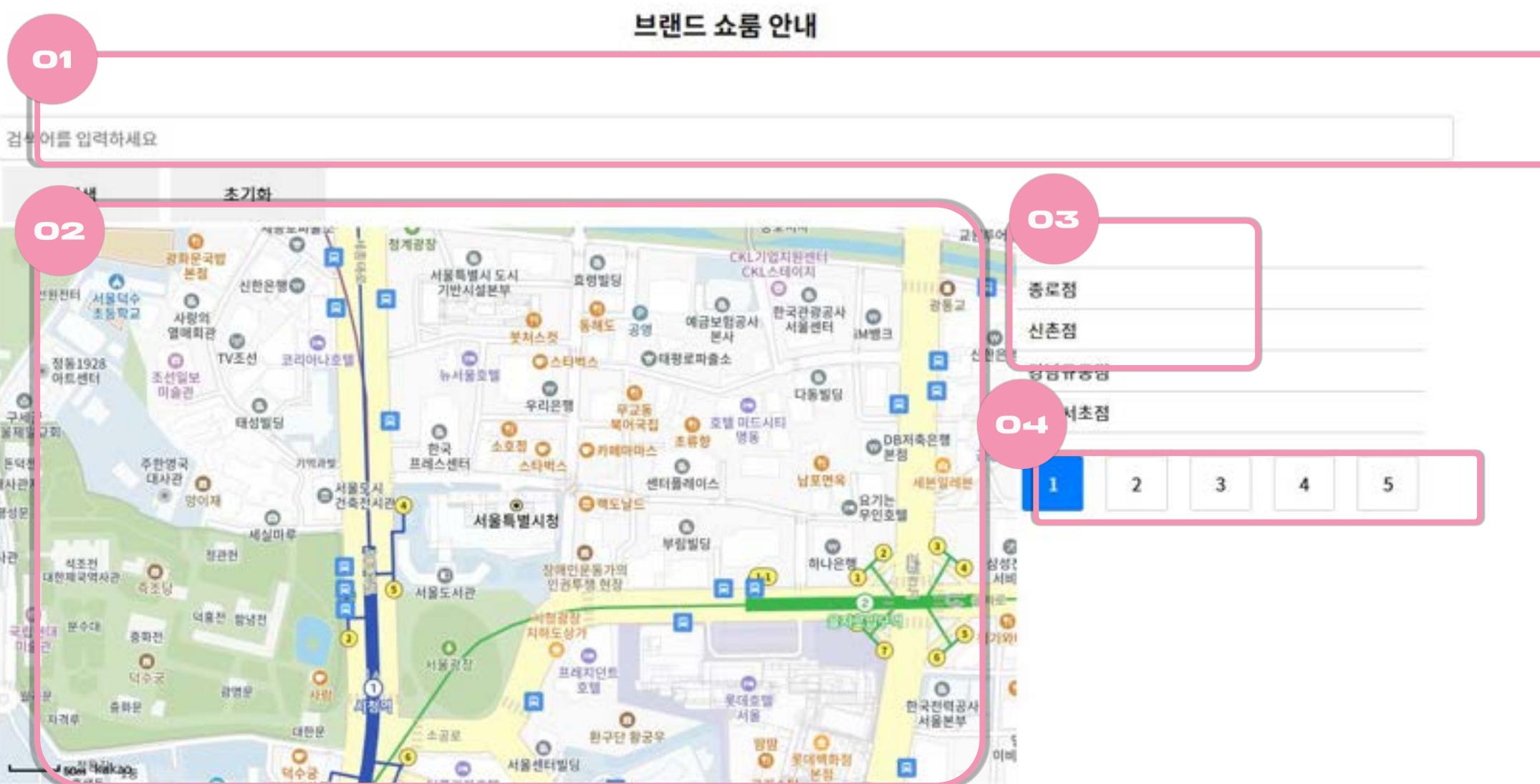
Description List

01

고객센터 하위 메뉴 지점 선택

주문 관리

Description List



01 원하는 지점명을 검색할 수 있는 검색창

02 카카오 api를 통해 연동한 지도

03 기본적으로 데이터베이스에
저장되어있는 지점들을 나열

04 지점들의 목록을 페이지네이션을
통해 나열

주문 관리

Description List

01

검색어 입력 후 검색

02

해당 검색어를 포함하는 지점들이 목록에
표시됨

브랜드 쇼룸 안내



주문 관리

Description List

01

```
useEffect(() => {
  const script = document.createElement('script');
  script.src =
    '//dapi.kakao.com/v2/maps/sdk.js?appkey=e207779030ab91ab36823aa2027266a6&libraries=services&autoload=false';
  script.async = true;
  document.head.appendChild(script);
```

02

```
// 검색 필터 추가
if (search) {
  whereClause += ' AND (name LIKE ? OR address LIKE ?)';
  queryParams.push(`%${search}%`, `%${search}%`);
```

03

```
const dataQuery = `
  SELECT id, name, address, phone, hours, image, lat, lng
  FROM showrooms
  ${whereClause}
`;
```

01

카카오 api지도 연동

02

검색 필터 관련 코드

03

데이터베이스상의 지점 정보 가져오기

주문 관리

Description List

01

해당 지점을 선택하면 저장된 위치로 이동

02

각 위치값이 저장된 곳에 마커로 표시

03

마커 선택시 해당지점으로 이동 및 정보 표시
뒤로가기 버튼을 통해 원래 창으로 이동



주문 관리

배송정보

■ 받으실 분
임세훈

■ 받으실 곳
어딘가

■ 주소 선택

■ 전화번호
02-579-6269

■ 휴대폰 번호
010-5796-6269

Description List

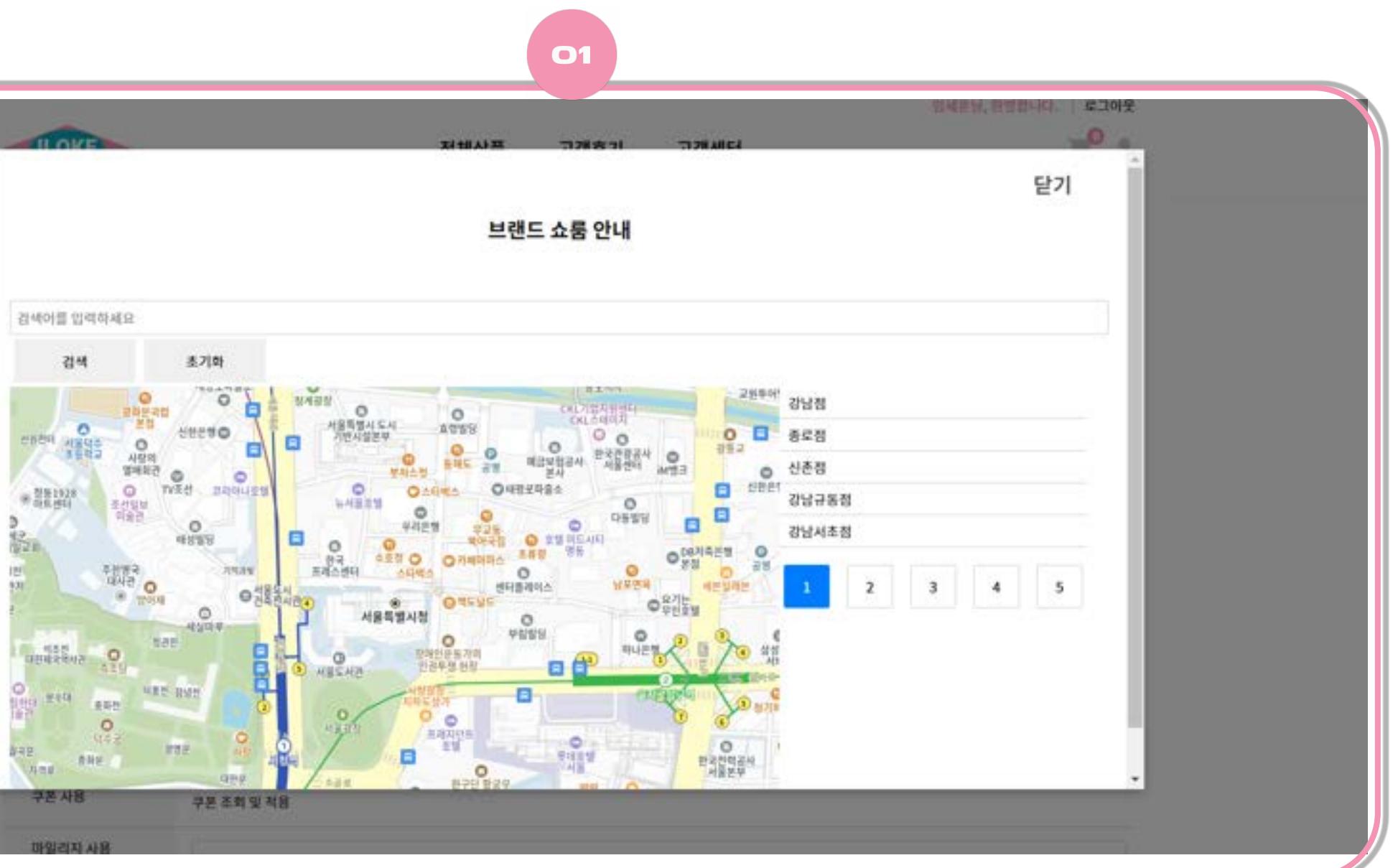
- 01 주소 선택 버튼을 통해 각 지점을 찾아 해당 주소로 주문 진행 가능 (직접 입력도 가능)

- 02 모달 상태일 때는 예, 아니오 버튼이 뜨도록 설정

```
/* 모달 창 */
{showModal && (
  <div className={styles.modal_overlay}>
    <div className={styles.modal_content}>
      <button
        className={styles.close_button}
        onClick={() => setShowModal(false)} // 모달 닫기
      >
        닫기
      </button>
      <ShowRoomInfo
        onSelectShowroom={handleSelectShowroom}
        mode="modal" // 모달에서는 modal 모드로 설정
      />
    </div>
  </div>
)}
```

01

버튼 클릭시 모달창 출력



주문 관리

Description List

01

모달창에서는 지점 클릭시
예, 아니오 버튼을 통해 해당 지점으로 배송
할것인지 여부를 선택가능



주문 관리

배송정보

■ 받으실 분 : 임세훈

■ 받으실 곳 : 서울특별시 신촌구 대흥동 50-1

■ 주소 선택

■ 전화번호 : 02-579-6269

■ 휴대폰 번호 : 010-5796-6269

01

Description List

01

찾은 지점에서 예 버튼을 클릭해
지점의 주소로 등록된 화면