

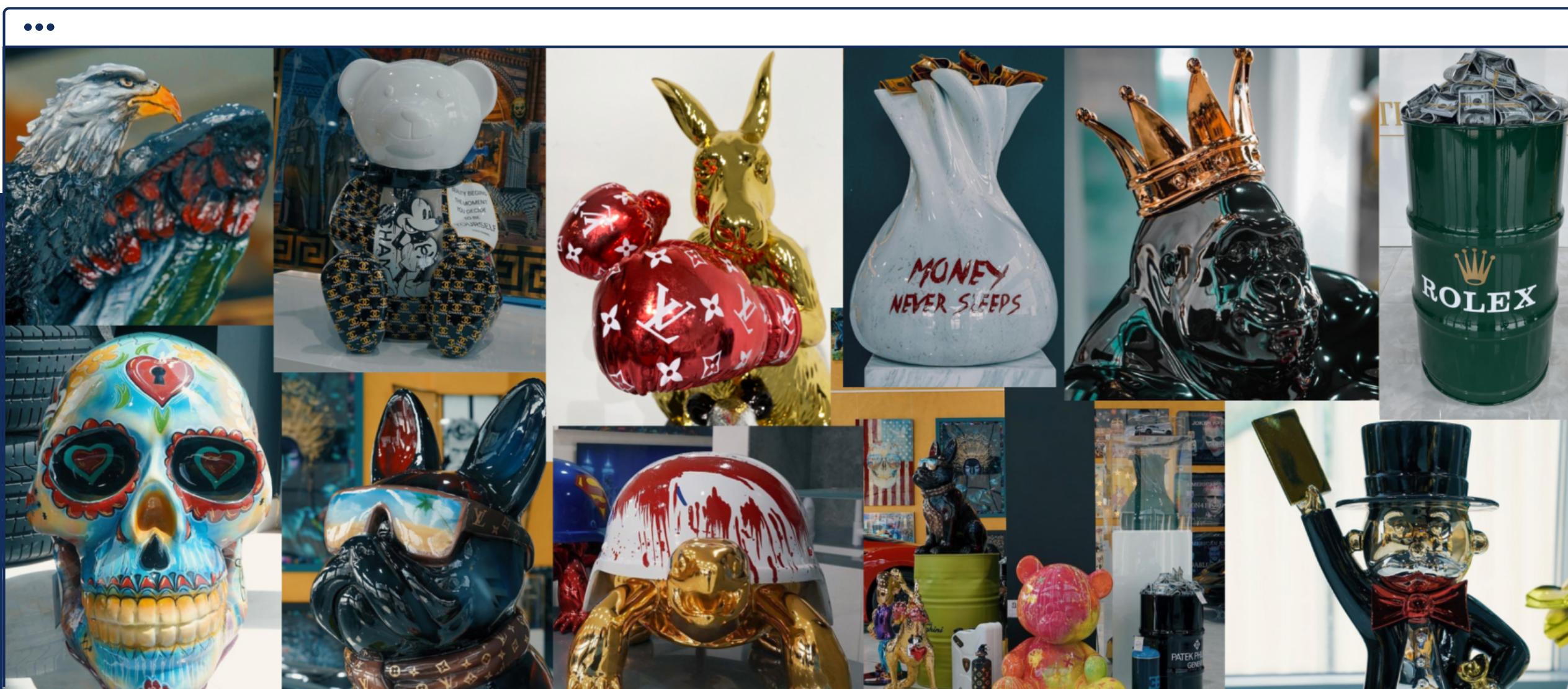
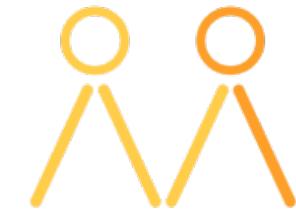
<https://github.com/myj1217/nimonemo>



2024

# 니모내모

네가 만든 모든 것, 내가 만든 모든 것



>>>

:01



...

# 목차

Contents

## 01 프로젝트 배경

- 01 프로젝트 배경
- 02 핵심 기능
- 03 개발 환경
- 04 프로젝트 구조

## 02 팀 구성 및 역할

- 01 팀 구성 및 역할

## 03 수행 절차 및 방법

## 04 수행 결과

- 01 프로젝트 범위 (구현 기능)
- 02 데이터베이스 설계
- 03 웹페이지 계층 구조
- 04 페이지별 상세 기능

## 05 기대효과 및 느낀점

- 01 기대효과
- 02 느낀점

# 프로젝트 배경

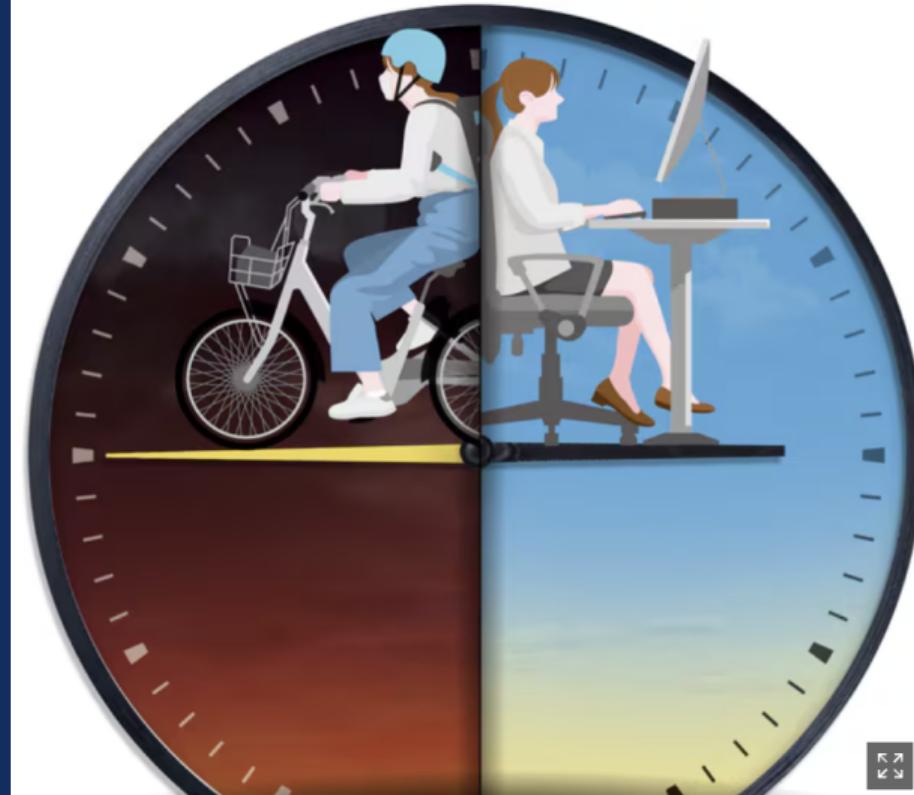
## 직장인들의 부업 긍정도 증가

### 삼성·현대차 직원들도 투잡 뛴다... 4대그룹 4610명 부업 전선

부업 중개 플랫폼 '커리어데이' 살펴보니... 4대 그룹 4610명

석남준 기자  
업데이트 2024.03.13. 10:51

▷ 가 □



일러스트=백형선

" 삼성전자 직원 971명을 포함해 삼성그룹 1784명, 현대차그룹 1373명, SK그룹 743명이 회원으로 등록돼 있습니다. " 직장인들의 부업 자리를 중개해주는 플랫폼 '커리어데이' 강경민 대표의 말이다. 이 업체에 회원으로 가입하기 위해서는 주민등록번호를 입력하고, 건강보험공단 이력 조회에 동의해야 한다. 이 과정을 거쳐 업체는 회원의 과거부터 현재까지 재직 이력을 확인하게 된다.

### 직장인 89.1% '재능살려 N잡 희망'...부업 경험도

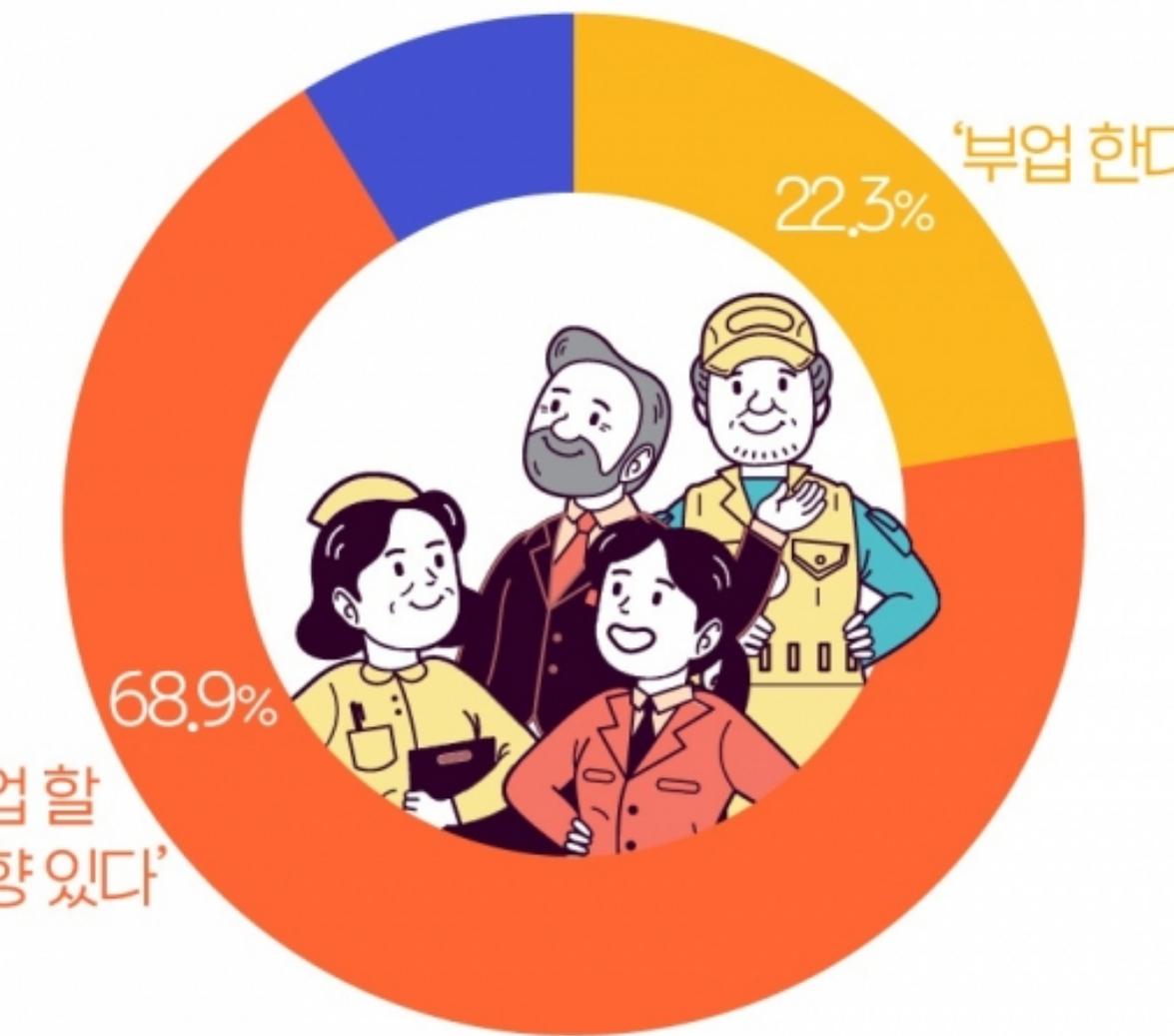
김윤경 기자 / 기사승인 : 2022-08-19 11:33:01

본 직장인 864명 대상 '직장인 N잡려 경험과 인식' 설문조사

에 대한 관심이 높은 것으로 나타났다. 직장인 10명 중 6명이 부업 경험이 있고 8명이 부업하고 싶다고 답했다.

### 직장인 91.2% 'N잡려 꿈꾼다'

| 자료제공 : 알바몬X긱몬 (직장인 1,324명 조사)



포털 알바몬과 재능거래 앱 긱몬이 직장인 864명을 대상으로 '직장인 N잡려(여러 경험과 인식)'에 대한 설문조사에서 응답자의 89.1%가 '재능을 살려 부업을 해보고 싶다'고 답했다. 20대 직장인 90.6%와 30대 89.6%, 40대이상 86.2%가 이같이 답

았다. 응답자의 61.3%가 '부업 경험이 있다'고 답했다.

한 부업은 '오프라인 아르바이트'(28.5%)였다. 응답자(복수응답)의 18.1%는 손그림 '핸드메이드 제작'을 해봤다고 했다. 이어 △문서작업(14.5%) △과외/악기레슨(13.0%)

하는 N잡(여러직업) 아이템은 20대 응답자의 21.8%가 블로그, 유튜브 등 'SNS'에 '문서작업(21.0%)'이나 '핸드메이드 제작(19.0%)' 순이었다.

>>>

:03

# 프로젝트 배경

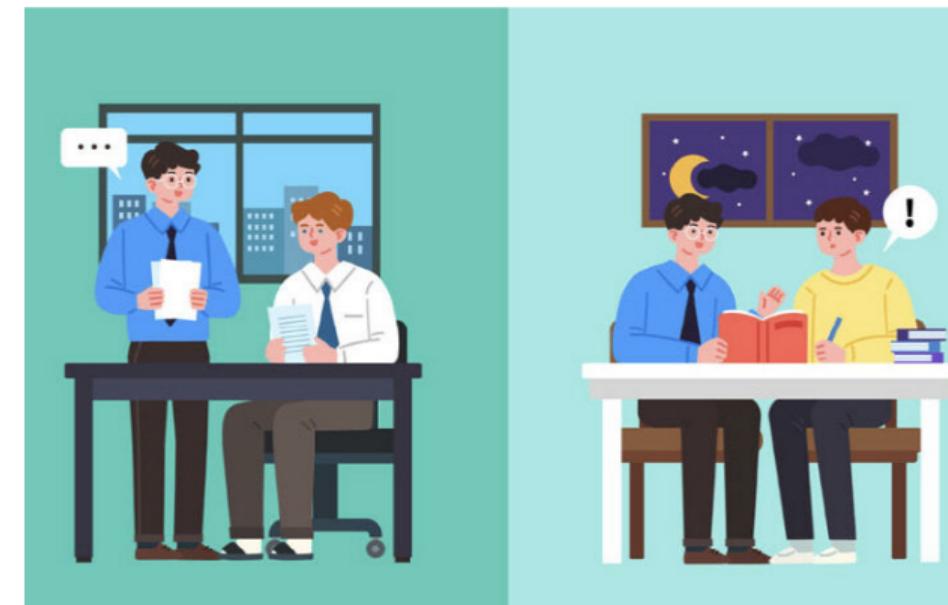
## 이전과 달라진 부업의 종류

### 직장인 10명 중 4명 취미·특기 살려 알바·부업하는 'N잡러'

잡코리아 '직장인 N잡 현황' 설문조사  
응답자 중 38.5% 본업 외 다른 일 하고 있어

승인 2021-11-21 13:50

박병주 기자



[이미지=게티이미지뱅크]

자신의 취미와 특기를 살려 아르바이트나 부업 등으로 'N잡'을 하는 직장인이 10명 중 4명에 이르는 것으로 나타났다.

주 5일제 근무제 도입으로 투잡 열풍이 시작됐고, 최근 주 52시간제가 시행되면서 미래를 위해 투자하는 환경이 조성되면서다.

특히 고용 불안과 더불어 평생직장 개념이 모호해지면서 자기계발을 통해 활동 범위를 넓혀가는 분위기도 이어지고 있다.

### [N잡해볼까] '덕질'로 오픈마켓 입점해서 돈 버는 직장인 가능?

김세원 | 승인 2024.02.22 17:27 | 댓글 0

인구축 방법부터 직장인 부업 주의점까지 '한 번에'

#### \* 연령대별 희망하는 N잡 아이템 (복수응답)



하는 젊은 직장인들이 늘어나고 있다.

6명을 대상으로 조사한 결과, 전체 응답자 중 34.5%가 '현 직장생활과 병행'을 주로 하는 알바 업종으로는 '음식점 서빙·주방일'(18.1%), '인터넷 커뮤니케이션'(17.7%), 이벤트·행사 스텝(14.8%), 매장관리·판매(14.5%), 문서작성(12.5%) 등이다.

소한 '돈 버는 취미'가 생각보다 꽤 많다. 핸드메이드 액세서리를 맞춤 제작하거나 전자책을 출간해도 된다. 개인 SNS를 열심히 하고 있다면 체험단이나 수입이 창출되는 파이프라인이다.

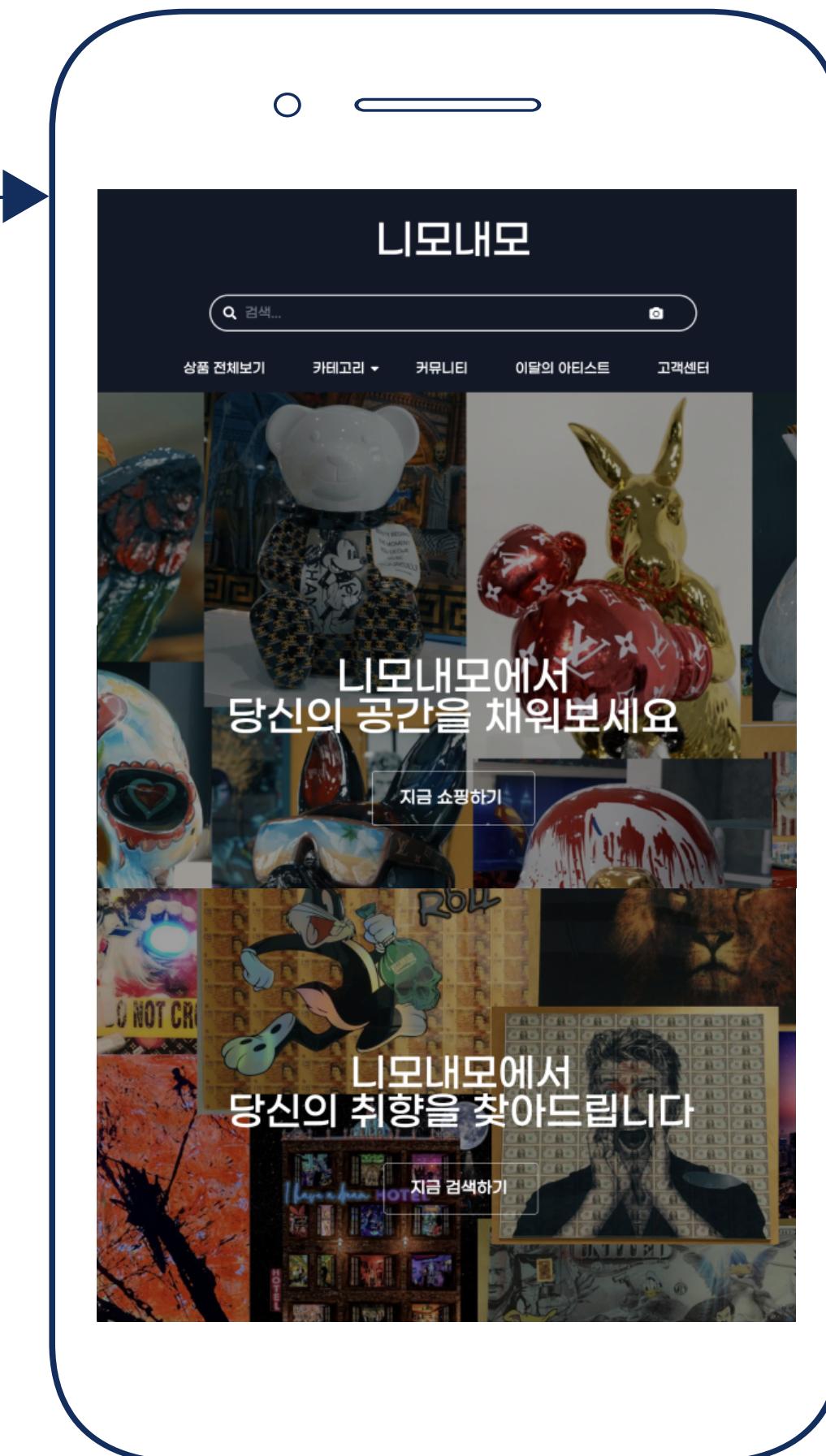
직장인은 어떻게 구축할 수 있을까.

>>>

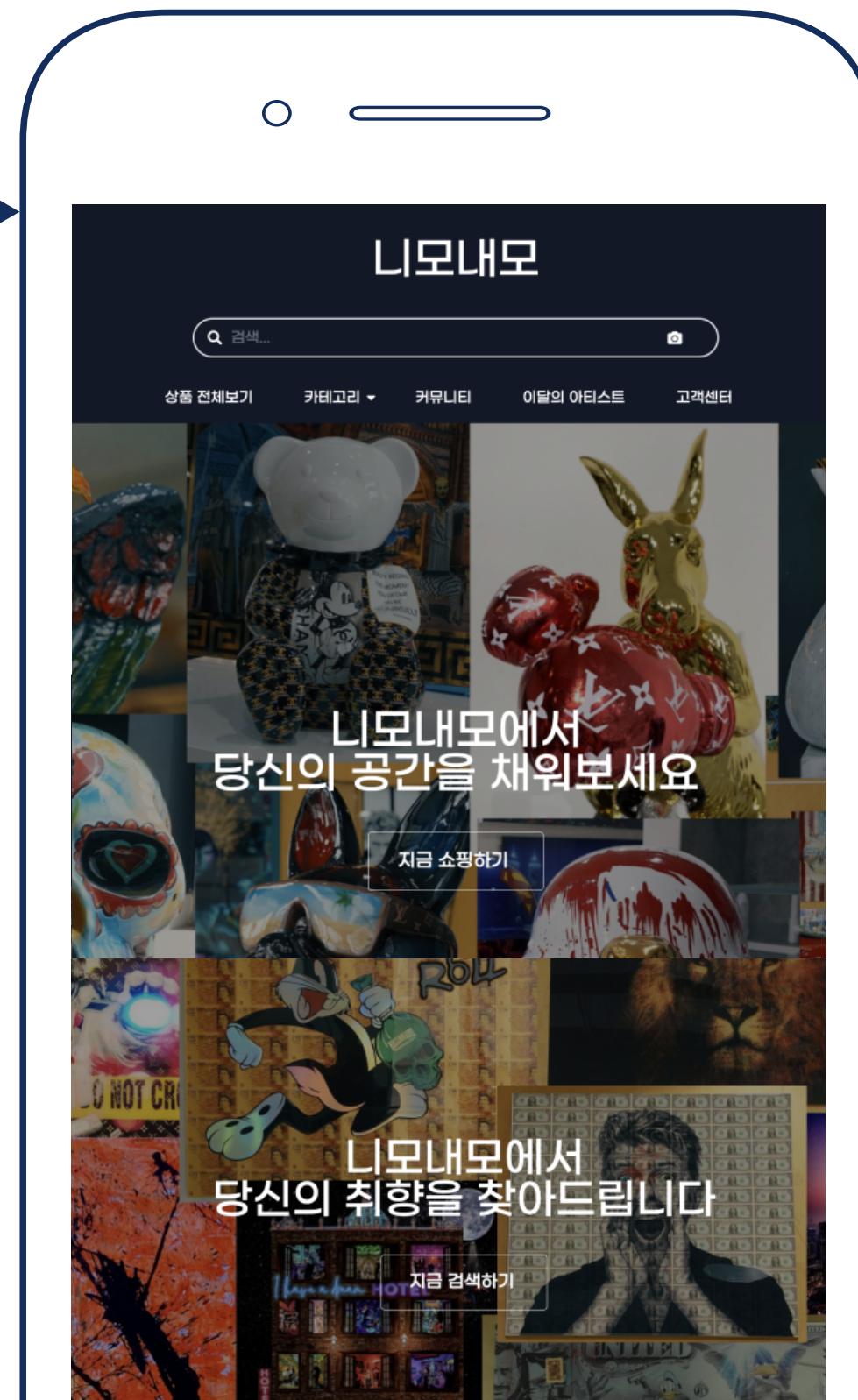
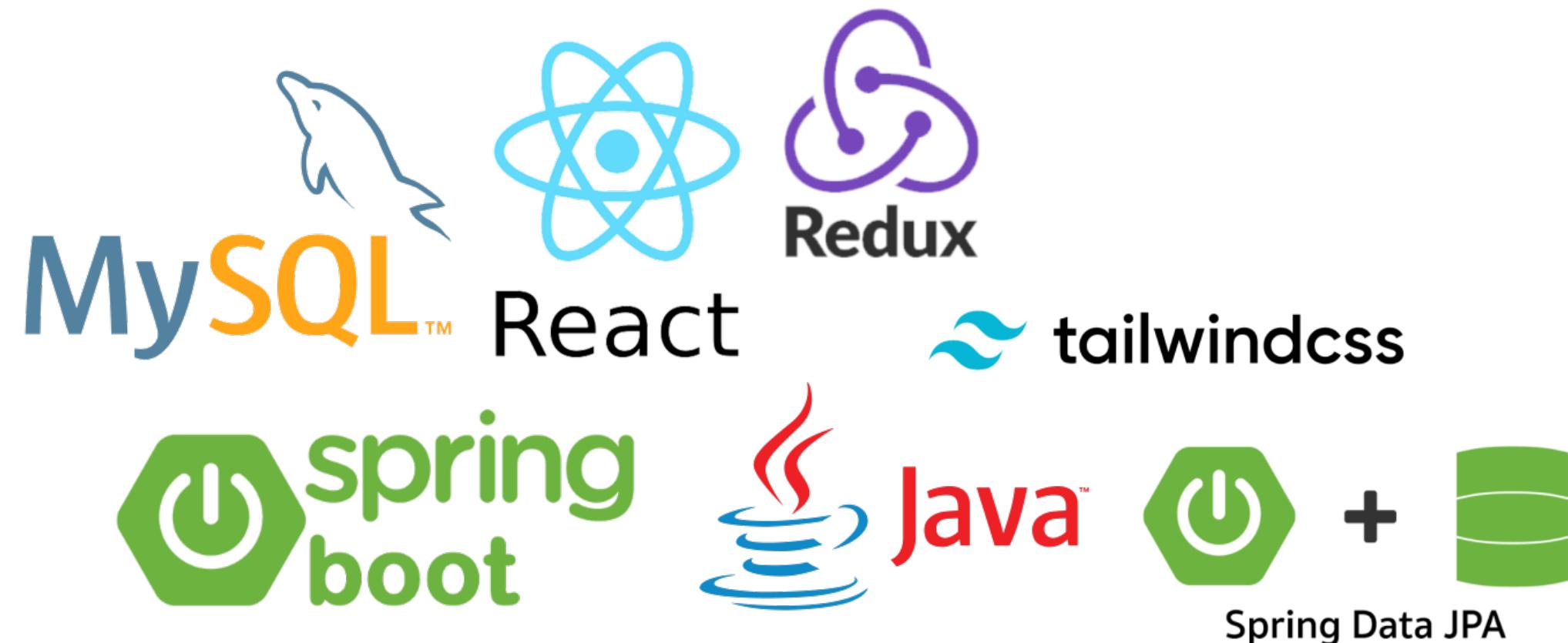
:04

# 핵심 기능

핵심 기능	상세 설명
Tensorflow API를 이용한 이미지 분석	<ul style="list-style-type: none"><li>- 검색창에 카메라 아이콘을 통해 <b>이미지를 업로드</b>하면 <b>Tensorflow API</b>를 이용해 <b>분석</b>을 진행한다.</li><li>- 분석 후 관련 <b>키워드 추출</b>한다.</li><li>- 키워드를 통한 검색을 진행한다.</li><li>- 키워드에 해당되는 제품을 검색결과로 추천한다.</li></ul>
이메일 인증을 통한 비밀번호 찾기	<ul style="list-style-type: none"><li>- 비밀번호 찾기 진행 시 가입 시 입력한 이메일 주소로 <b>인증코드</b>를 전송하여 인증 후 <b>새 비밀번호</b>를 입력하여 변경한다.</li></ul>
챗봇	<ul style="list-style-type: none"><li>- 사용자가 원하는 내용의 <b>키워드</b>를 입력하면 (eg. 배송, 비밀번호 등) 관련 <b>답변을 제공</b>한다.</li></ul>

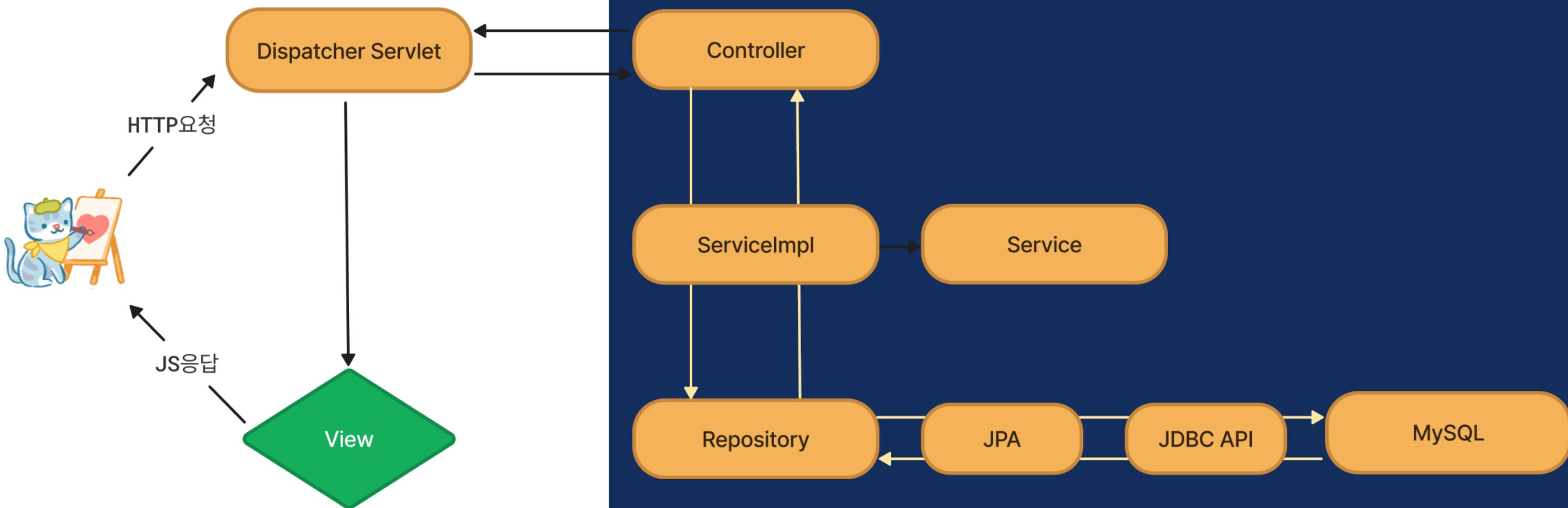


# 개발 환경



:06

# 프로젝트 구조



# 팀 구성 역할



나 모 내 모  
조민영

(Front)

- 메인페이지 구현
- 상품 CRUD 페이지 및 모달 구현
- 장바구니 CRUD 페이지 및 모달 구현
- 장바구니 체크박스 기능 및 하이라이트 처리 구현
- 페이지 간 라우터 처리 구현

(Back)

- 상품 CRUD 및 페이징 처리 구현
- 상품 관련 파일 처리 구현
- 상품 이미지 썸네일 자동 생성 구현
- 장바구니 CRUD 구현
- 장바구니 목록 카운트 처리 구현



나 모 내 모  
김경목

(Front)

- 로그인 페이지 구현
- 회원가입 페이지 구현
- 마이페이지, 회원정보 수정 페이지 구현
- 상품 검색 및 결과창 구현
- 페이지 정보 상단바 구현
- 챗봇 버튼, 스크롤업 버튼 구현
- 404 ERROR 페이지 구현
- 권한 확인 페이지 구현

(Back)

- 계정 데이터베이스(토큰 검증) 구현
- 로그인, 로그아웃, 소셜로그인 기능 구현
- 회원가입, 회원 탈퇴 기능 구현
- 마이페이지, 회원정보 수정, 비밀번호 찾기 기능 구현
- 상품 검색 기능 구현

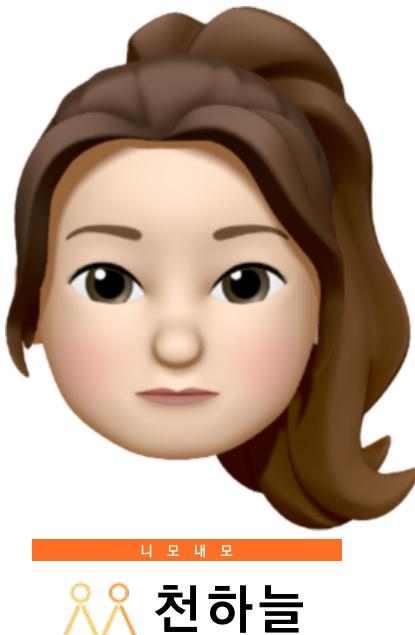
# 팀 구성 역할



(Front)

- BasicMenu 페이지 구현
- 장바구니 페이지 구현
- 로그인 페이지 구현
- 마이페이지 페이지 구현
- 회원가입 페이지 구현

- 상품 이미지 분석 기능 구현
- 챗봇 기능 구현
- 로딩페이지 구현
- 고객센터 페이지 구현
- 월간 아티스트 구현



(Front)

- 게시판 리스트 페이지 구현
- 게시글 CRUD 페이지 구현
- 파일 CRUD 페이지 구현
- 결제창, 주문확인 모달 창 페이지 구현

(Back)

- 게시판 리스트, 페이징 구현
- 게시글 CRUD 구현
- 게시글 파일CRUD 구현
- 파일 미리보기, 썸네일 구현
- 아임포트API를 통한 카카오페이지 결제 기능 구현

# 수행 절차 방법



## 사전 기획

3/5(화)~3/7(목)

- 아이디어 브레인 스토밍
- 프로젝트 기획 및 주제 토의
- 역할 분담 및 각 파트별 선수학습



## 레퍼런스 조사

3/8(금)~3/11(월)

- 모토기업 지정
- 레퍼런스 조사  
(아이디어스, 엣시, 럭스아트,  
이너팩토리 등)



## 프로젝트 설계

3/12(화)~3/14(목)

- UI설계, 핵심 기능 개발
- ERD 설계, WBS 초안 작성



## 최종 구현

3/26(화)~3/27(수)

- 추가기능 (비밀번호 찾기, 챗봇 등) 구현
- UI 최종 작업
- 각 코드 리뷰, 주석 및 리팩토링
- 프로젝트 정리 및 문서화



## 프로젝트 구현

3/15(금)~3/25(월)

- CRUD 형태의 필수 기능 구현
- 메인페이지 UI화면 픽스
- 파트 병합 후 에러 처리



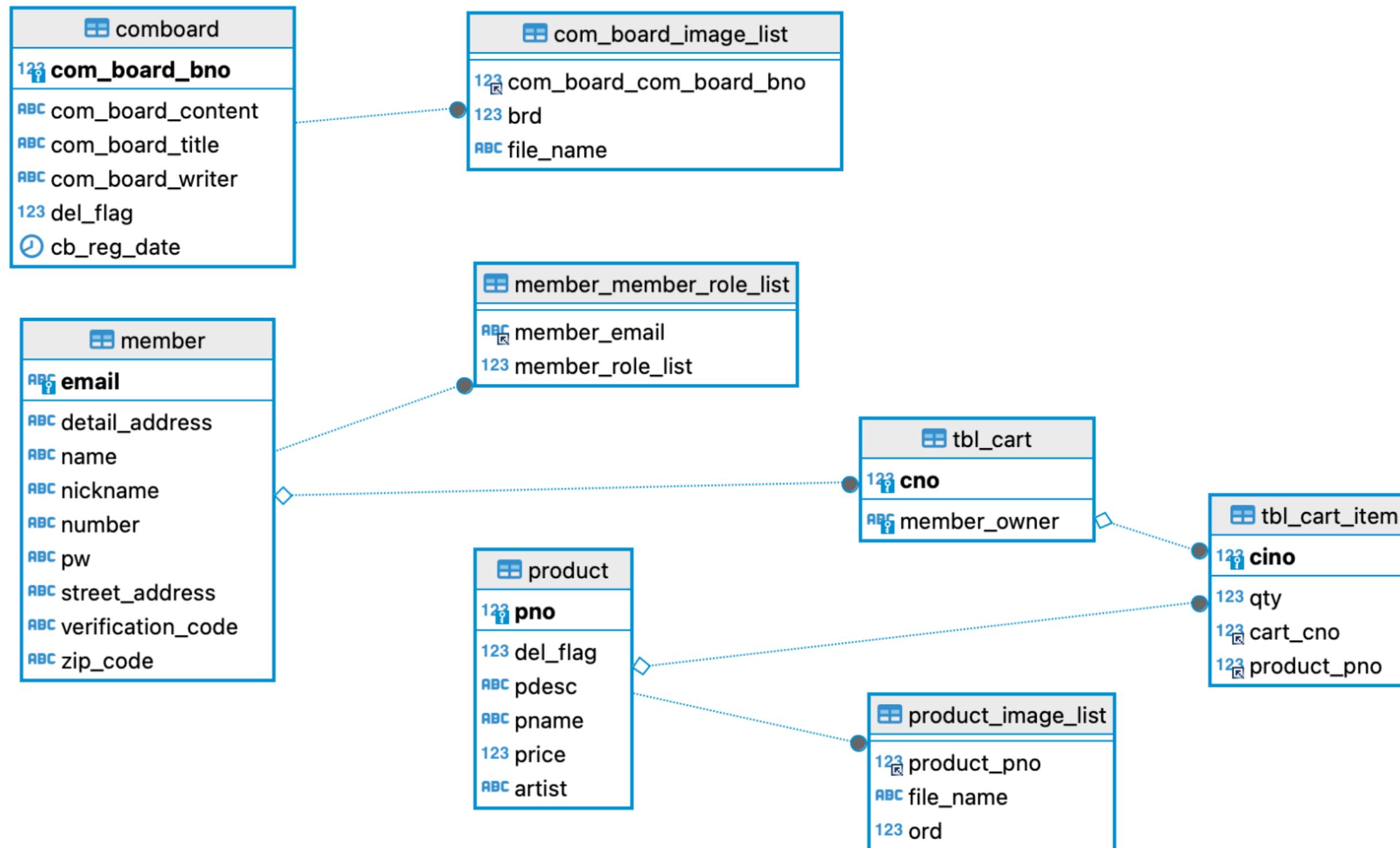
# 수행 결과

## 1. 프로젝트 범위(구현 기능)

구현 기능	상세 내용
회원	<ul style="list-style-type: none"><li>- 로그인, 로그아웃, 소셜로그인</li><li>- 회원가입, 탈퇴</li><li>- 회원정보수정, 비밀번호찾기</li></ul>
상품	<ul style="list-style-type: none"><li>- 이미지분석 후 키워드 추출</li><li>- 키워드를 통한 상품 검색 결과 제공</li><li>- 상품 리스트 전체 조회, 상세보기, 등록, 수정, 삭제</li><li>- 상품리스트페이지</li><li>- 장바구니 자동 합산</li><li>- 결제 페이지에서 배송지 변경</li><li>- 카카오페이지 결제</li></ul>
게시판	<ul style="list-style-type: none"><li>- 전체 게시글 조회, 게시글 상세보기, 등록, 수정, 삭제</li><li>- 게시글에 이미지 등록, 이미지 미리보기</li><li>- 게시글 페이지</li></ul>
챗봇	<ul style="list-style-type: none"><li>- 키워드 검색을 통한 문의 답변 제공</li></ul>

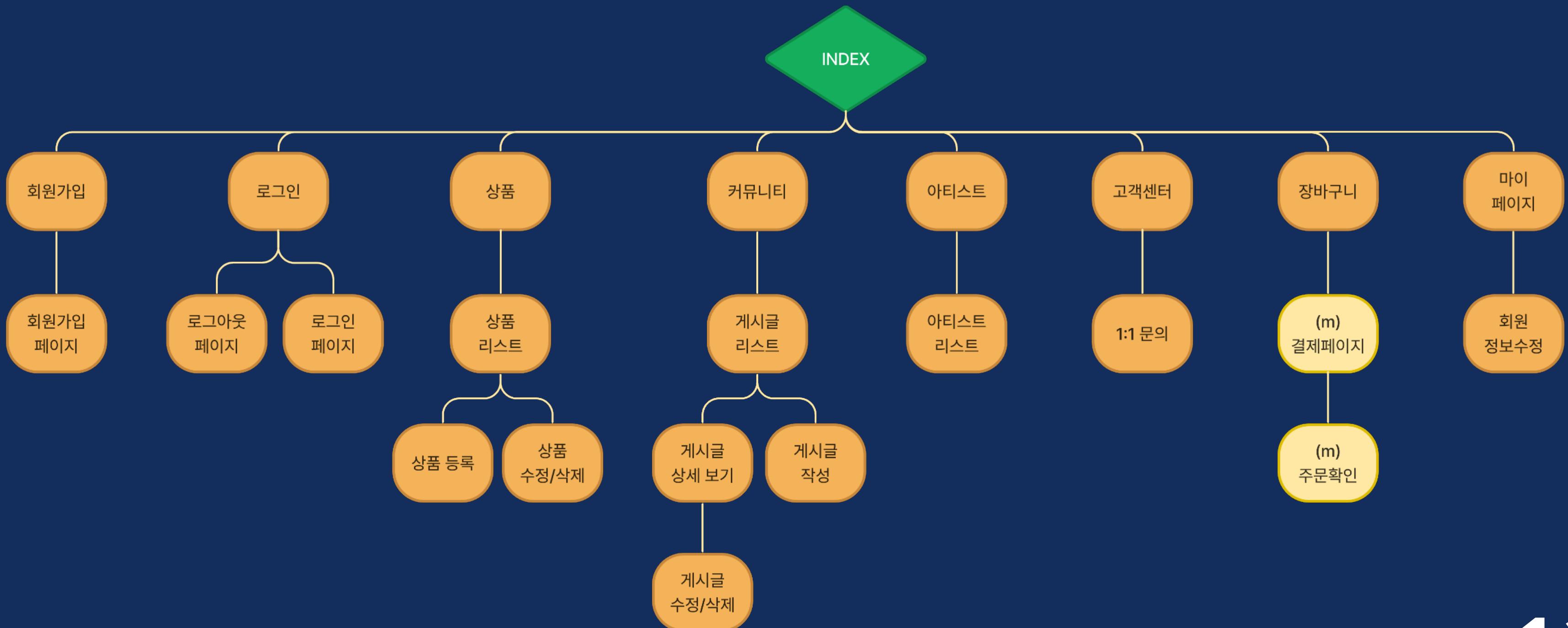
# 수행 결과

## 2. 데이터베이스 설계 (ERD)



# 수행 결과

## 3. 웹 페이지 계층 구조



# 수행 결과

## 4. 페이지 별 상세 기능



```
const determineCategory = async (predictions) => {
  //이미지 분석결과 중 가장 일치할 확률이 높은것 선택
  const firstPrediction = predictions[0];
  const label = firstPrediction.className.toLowerCase();

  // 이미지 분석결과를 한글로 번역
  const translatedLabel = await translateLabel(label, "en", "ko");

  // 이미지분석한것의 검색결과창으로 리다이렉트
  setError("");
  navigate(`/products/search?pname=${translatedLabel}`);
  setLoading(true);

  try {
    const img = document.createElement("img");
    img.src = URL.createObjectURL(file);
    img.onload = async () => {
      const predictions = await model.classify(img);
      console.log(predictions);

      setIsModalOpen(true); // 여기서 모달을 엽니다
      setTimeout(() => {
        setIsModalOpen(false); // 2초 후 모달을 닫습니다
        determineCategory(predictions);
      }, 2000);
    };
  } catch (error) {
    console.error("이미지 분류 중 오류 발생:", error);
    setError("이미지 분류 중 오류 발생.");
  } finally {
    // ...
  }
};
```

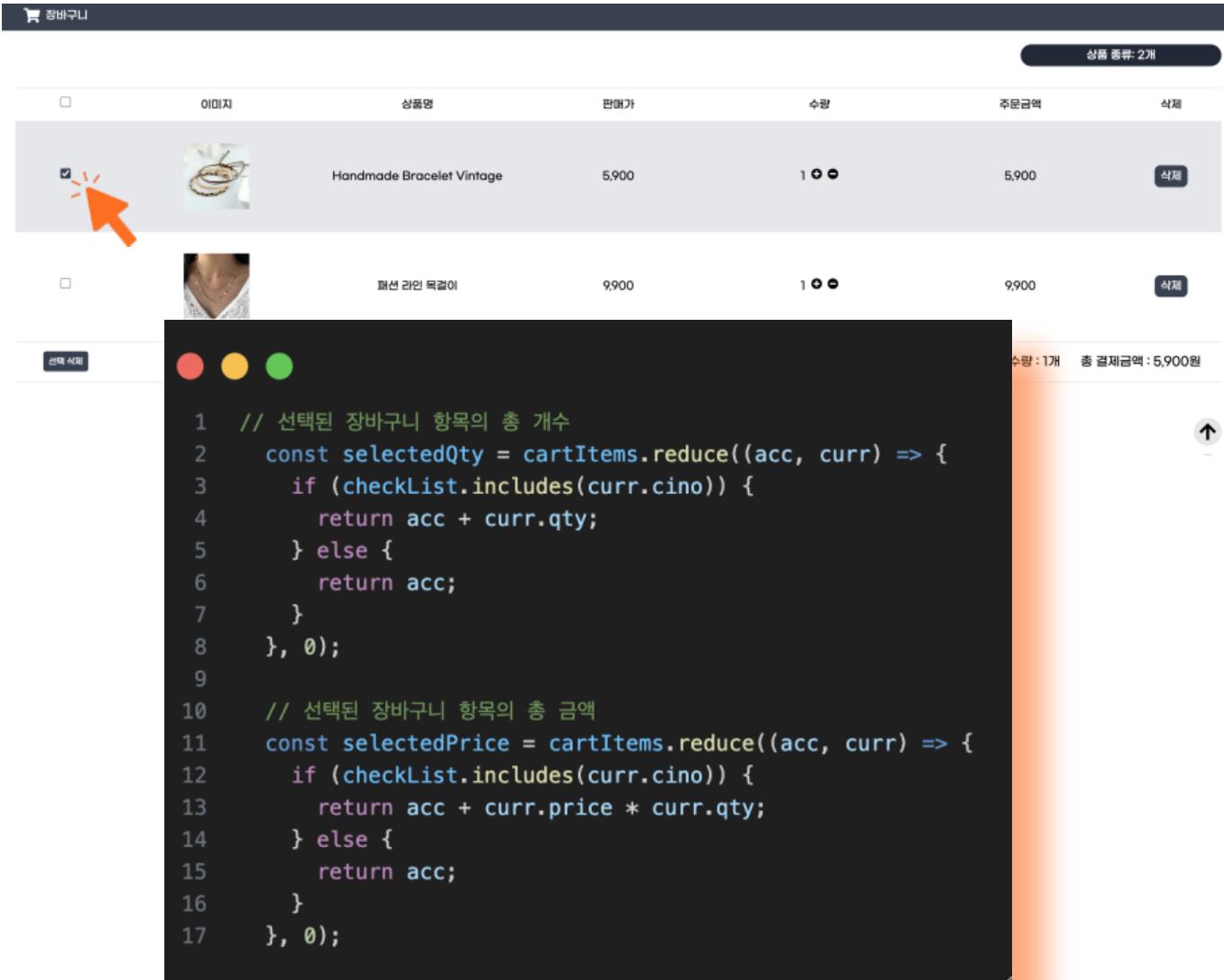
- 1. 카메라 아이콘을 통해**
  - 이미지 업로드
  - 키워드를 추출
- 2. 추출된 키워드로 검색 실행**  
검색 결과를 유저에게 제공

>>>

: 14

# 수행 결과

## 4. 페이지 별 상세 기능



```
1 const ClickChargeBtn = (pg_method) => {
2   const { IMP } = window;
3   IMP.init('개인 impid');
4   IMP.request_pay(
5     {
6       pg: "kakaopay",
7       pay_method: "card",
8       merchant_uid: `mid_${new Date().getTime()}`,
9       name: "테스트 상품",
10      amount: 100,
11      buyer_email: buyerInfo.email,
12      buyer_name: buyerInfo.name,
13      buyer_tel: buyerInfo.tel,
14      buyer_postcode: buyerInfo.postcode,
15      buyer_addr: buyerInfo.address,
16      m_redirect_url: "http://localhost:3000/cart",
17    },
18    function (rsp) {
19      // callback
20      if (rsp.success) {
21        alert("결제 성공");
22        closeModal(); // 결제 성공 시 모달을 닫습니다.
23      } else {
24        alert("결제 실패");
25        console.log(rsp);
26      }
27    });
28  };
29};
```

- 선택한 부분만 totalqty, totalprice 제공
- IMPORT API를 이용한 결제 진행

>>>

: 15

# 수행 결과

## 4. 페이지 별 상세 기능



1. 자동으로 주문 완료한 상품은 장바구니에서 삭제

The screenshot shows the '장바구니' (Cart) page with the following details:

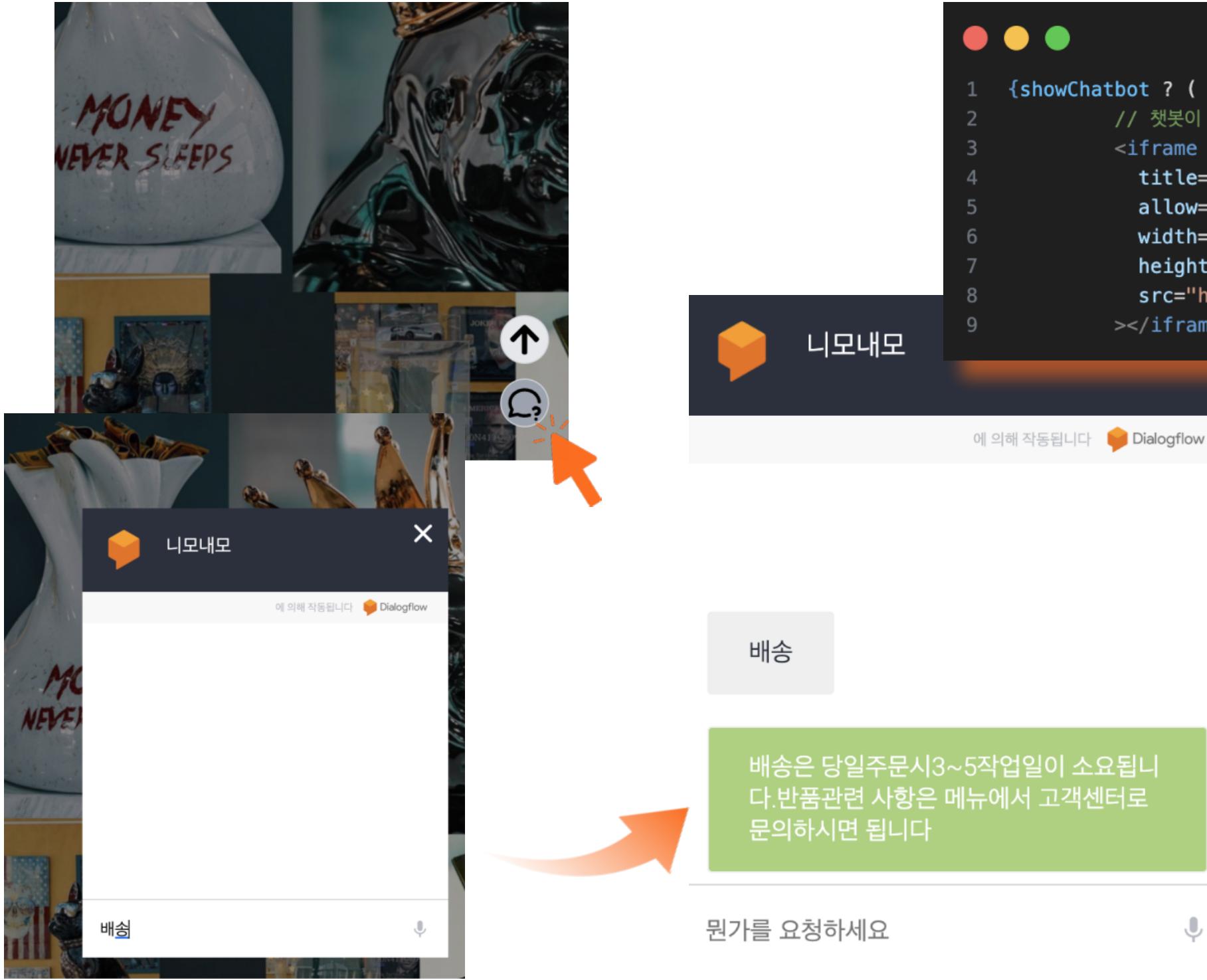
□	이미지	상품명	판매가	수량	주문금액	삭제
		패션 라인 목걸이	9,900	1 ⚡ 0	9,900	<button>삭제</button>

Below the table, there is a modal window titled '선택 상품 주문하기' (Select Product to Order) containing the following code:

```
1 // 주문완료 후 장바구니 비우기
2 const clearCart = () => {
3   cartItems.forEach((item) => {
4     if (checkList.includes(item.cino)) {
5       return changeCart({
6         email: loginState.email,
7         cino: item.cino,
8         pno: item.pno,
9         qty: 0,
10      });
11    }
12    return item;
13  });
14}
```

# 수행 결과

## 4. 페이지 별 상세 기능



1. 구글 클라우드를 이용한 챗봇 러닝

```
1 {showChatbot ? (
2     // 챗봇이 표시될 때는 챗봇 iframe이 보여짐
3     <iframe
4         title="nimonemo"
5         allow="microphone;"
6         width="350"
7         height="430"
8         src="https://console.dialogflow.com/api-client/demo/embedded/fcbf4403-2103-4a95-9b76-e8ec1cff9901"
9     ></iframe>
```

Action and parameters

Responses

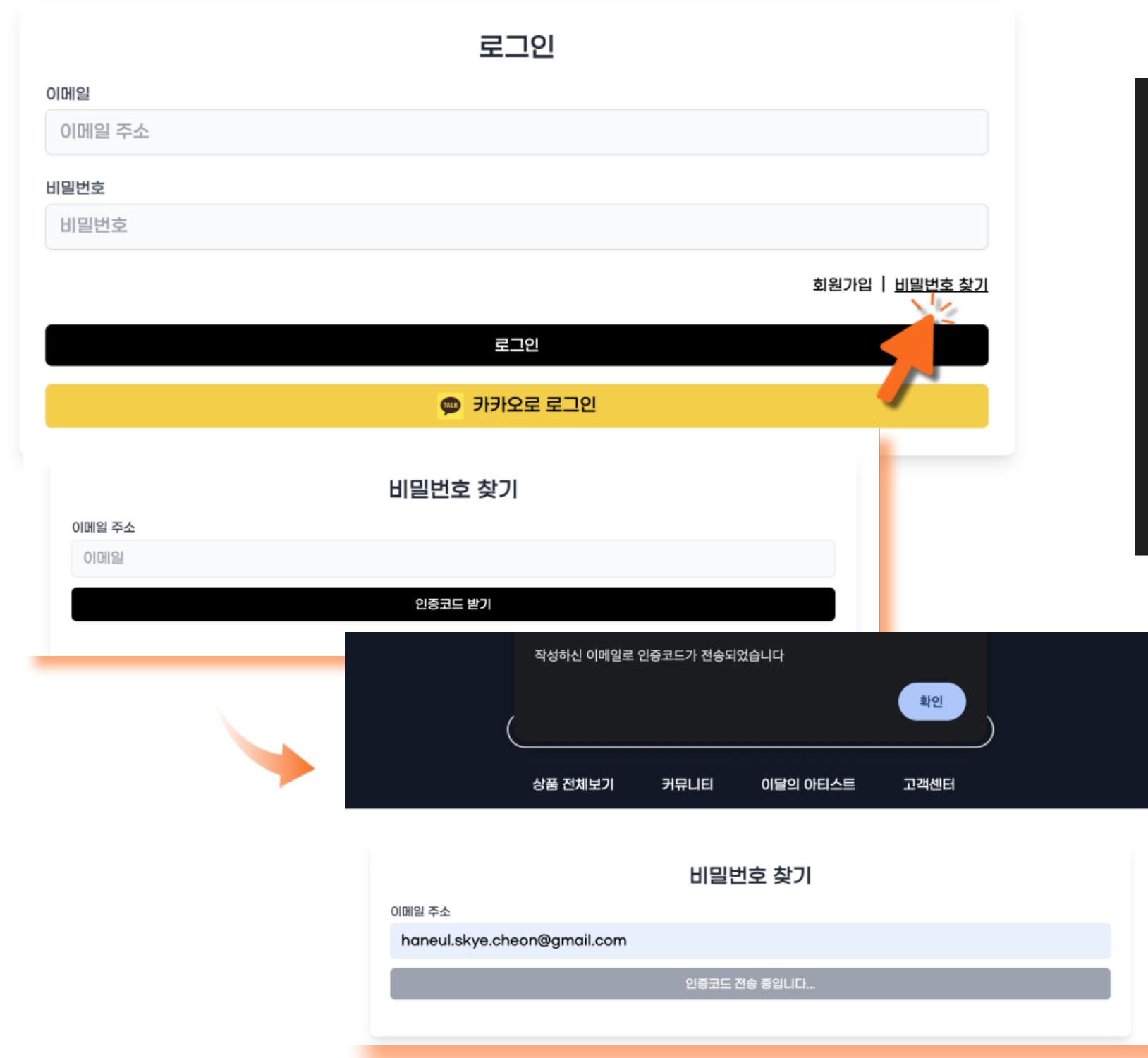
Text Response

:17

# 수행 결과

## 4. 페이지 별 상세 기능

### 로그인



```
1  @PostMapping("/sendCode")
2  public ResponseEntity<String> checkMember(@RequestParam("email") String email) {
3      boolean memberExists = mailService.existsByEmail(email); // 이메일을 확인한다
4      if (memberExists) {//이메일이 존재할경우 인증코드 전송.
5          MailDTO dto = mailService.createMail(email);
6          mailService.mailSend(dto);
7          return ResponseEntity.ok("이메일로 인증코드가 발급되었습니다.");
8      } else {//이메일이 없다면 안함
9          return ResponseEntity.status(HttpStatus.NOT_FOUND).body("입력하신 이메일에 해당하는 회원이 없습니다.");
10     }
11 }
12 }
13 }
```

1. 비밀번호 찾기 진행 시 메일주소 검증 진행
2. 가입 메일주소로 인증코드 발송

# 수행 결과

## 4. 페이지 별 상세 기능

The screenshot shows a web application interface. On the left, there's an inbox with a search bar at the top. Below it are three tabs: '기본' (Basic), '프로모션' (Promotion), and '소셜' (Social). The '기본' tab is selected. A list of emails is shown, with the first email from 'nmmntest6@gmail.com' highlighted. An orange arrow points to the subject line '니모내모 인증코드 안내 이메일 입니다.' and the recipient's name 'nmmntest6@gmail.com'. The email content is partially visible, showing a verification code 'ZYVKZUJHHU'. On the right, there's a large code editor window displaying Java code for generating and sending emails. The code uses a template string to construct the email body with HTML and CSS. Below the code editor, two large numbered steps are displayed: 1. 이메일로 받은 인증코드 입력 and 2. 비밀번호 변경.

```
1 // 메일 내용을 생성하고 인증번호를 저장시키기
2 @Override
3 public MailDTO createMail(String email) {
4     String code = randomCode();
5     MailDTO dto = new MailDTO();
6     dto.setAddress(email);
7     dto.setTitle("니모내모 인증코드 안내 이메일 입니다.");
8     dto.setMessage(
9         "<html><div align='center' style='border:1px solid grey; font-family:verdana; height: 200px;'><h3 style='color:grey;'>안녕하세요 니모내모입니다.
10 회원님의 비밀번호 재설정을 위한 인증코드를 보내드립니다.</h3><br/><br/>" +
11         + "<div style='font-size:30px'><strong>" +
12         + code +
13         + "</strong></div></div></html>");
14     updateCode(code, email);
15     return dto;
16 }
17
18 //데이터베이스에 인증번호를 저장.
19 @Override
20 public void updateCode(String code, String email) {
21     Member member = memberRepository.findByEmail(email);
22     member.changeVerificationCode(code);
23     memberRepository.save(member);
24 }
25 }
```

1. 이메일로 받은 인증코드 입력
2. 비밀번호 변경

>>>

: 19

# 기대 효과

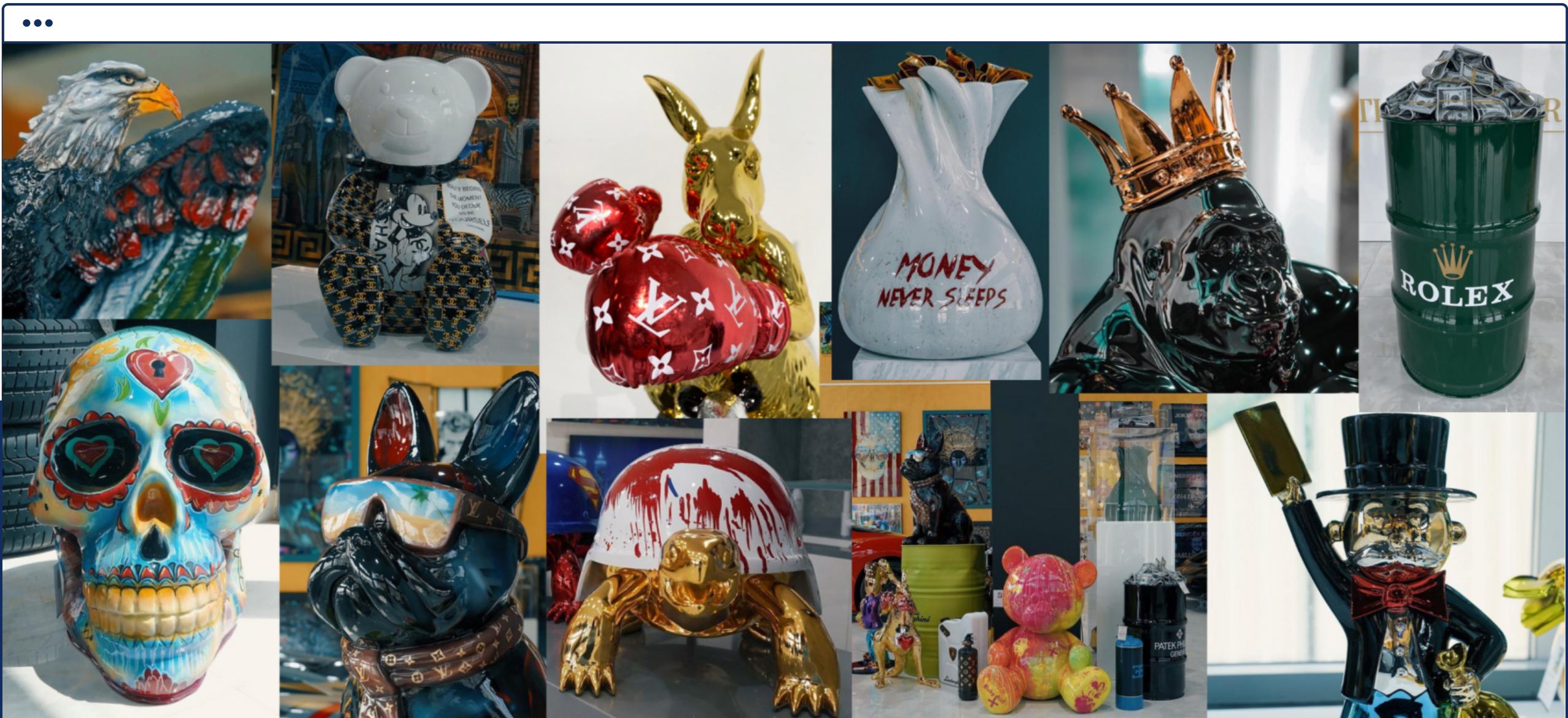
핵심 기능	기대 효과
Tensorflow API를 이용한 이미지 분석과 키워드 추출	- 사용자가 원하는 상품을 찾기 위해 <b>직접 이미지를 업로드</b> 를 하고 <b>추출된 키워드</b> 를 통해 관련도 높은 <b>상품을 추천</b> 받음으로서 능동적인 경험에 비례한 만족도를 얻게 된다.
게시판을 통한 사용자간 커뮤니케이션	- 검색 결과에도 원하는 상품을 찾지 못했을 경우에 <b>커뮤니티를 통해 사용자간 정보 공유</b> 할 수 있도록 한다.
챗봇	- 배송 및 계정 등 <b>키워드를 검색하면 관련도 높은 순으로 답변함</b> 으로서 관리자와 사용자의 편의를 모두 고려할 수 있다.

# 느낀 점

구분	내용
프로젝트 수행 중 어려움 극복 사례	<ul style="list-style-type: none"><li>매일 같은 <b>마감시간을 지정 후 진행상황을 공유</b>하여 <b>코드 변경으로 인한 혼란을 극복</b>하였다.</li><li>팀 노션을 통한 네트워크와 <b>코드 리뷰를 공유</b>하며 팀원들 간 어려움을 극복하였다.</li></ul>
프로젝트에서 잘한 부분	<ul style="list-style-type: none"><li>모든 팀원들이 담당한 파트에 대하여 <b>깊은 책임감을 가져 맡은 부분을 모두 완료</b>하였다.</li><li>프로젝트에 필요한 JPA 등의 새로운 <b>핵심 기술을 팀 스터디를 통해 습득</b>하여 프로젝트에 적용하였다.</li><li><b>Git을 이용한 프로젝트</b> 병합하는 과정에서 잦은 코드 충돌이 있어 난관을 겪었지만 <b>활발한 의사소통</b> 통해 Git을 능숙하게 다룰 수 있게 되었다.</li></ul>
프로젝트에서 아쉬운 부분	<ul style="list-style-type: none"><li>초기 셋업 시 BasicMenu에 의존성을 높게 두어 <b>코드 가독성이 떨어지는 점</b>이 아쉬웠다.</li></ul>



# Thank you



<<< :END