2025 정보보호프로젝트실습

AI 기반 허위 매물 방지 중고거래 서비스 '라온'

› 2학기 최종발표

PIO 팀

Agenda

Team member

Problem

2024 Highlight

번개장터

새 주인을 찾아
등록된 물건 수

4,100 만 개

총 가입자 2,300만명 중 MZ 세대 비중

78%

상품이 가장 많이 판매되는 시간
TUE/WED

9-11PM

23년 대비
거래건수 성장률

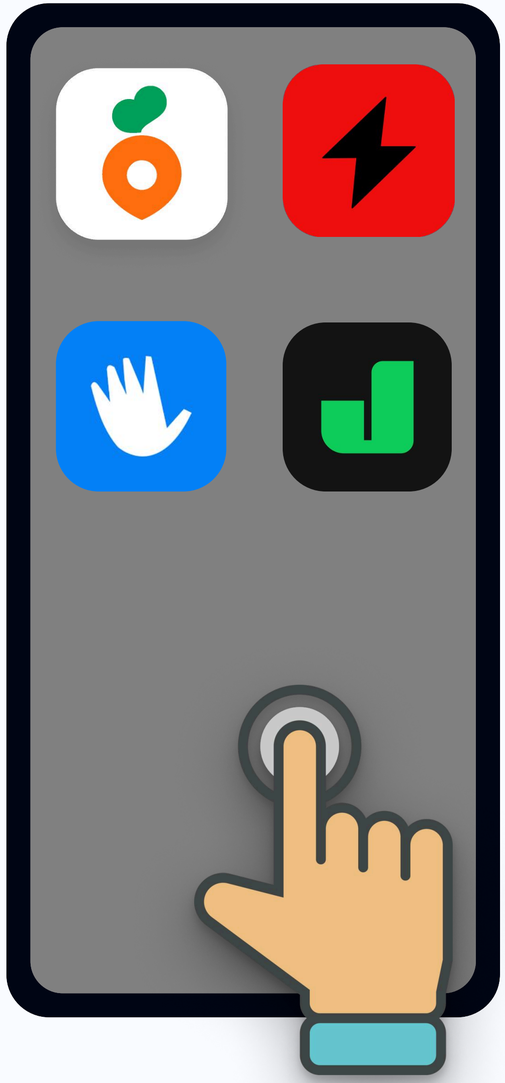
63%

최대 상품 이동거리

890km

가장 빠른 거래시간

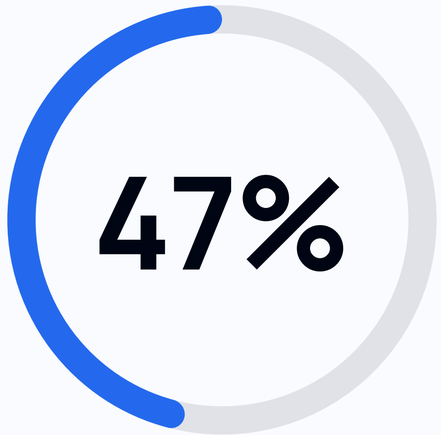
12.7 초



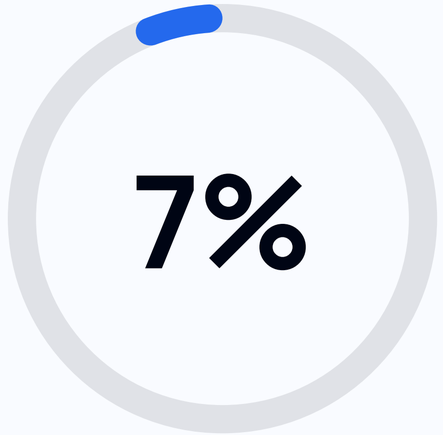
The PainPoint

이용자 입장에서의 불편함

사기거래에 대한 불안감



거래물품 품질·상태 확인

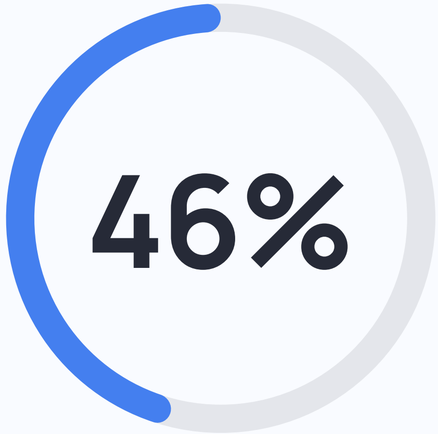


판매자와의 연락·가격 조율
및 기타

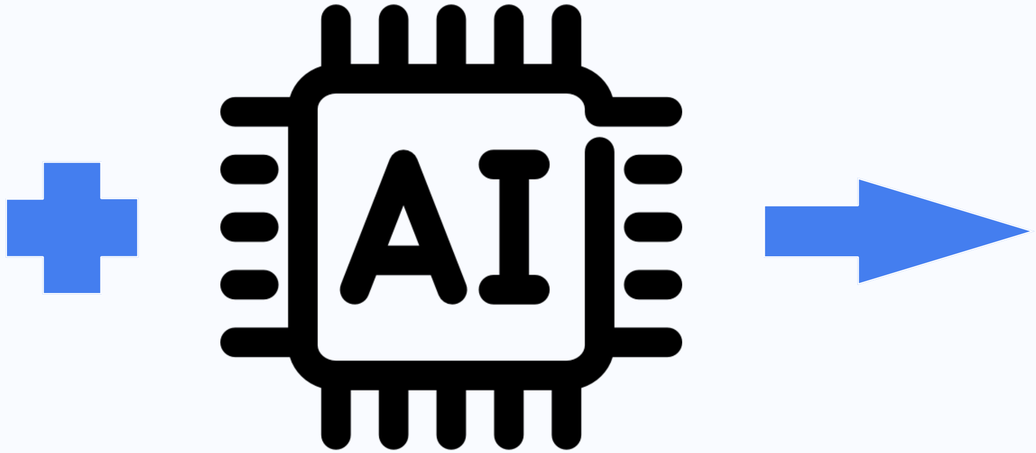
출처 : 세이프타임즈 뉴스 (https://www.safetimes.co.kr/news/articleView.html?idxno=204507)

The PainPoint

이용자 입장에서의 불편함



사기거래에 대한 불안감



PIO 팀

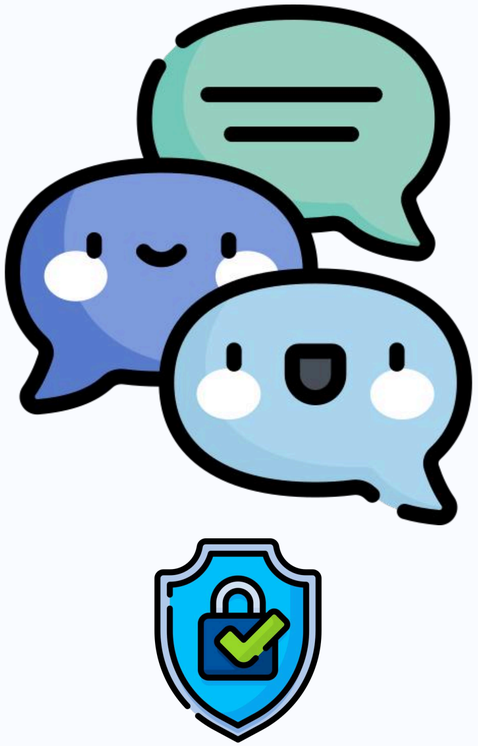
Solution

AI 기반 허위 매물 방지 중고거래 서비스

Improvements

AI 기반
허위 매물 방지

판매자



구매자

사기 방지
AI 챗봇

Develop

Tech Spec

Frontend (app)

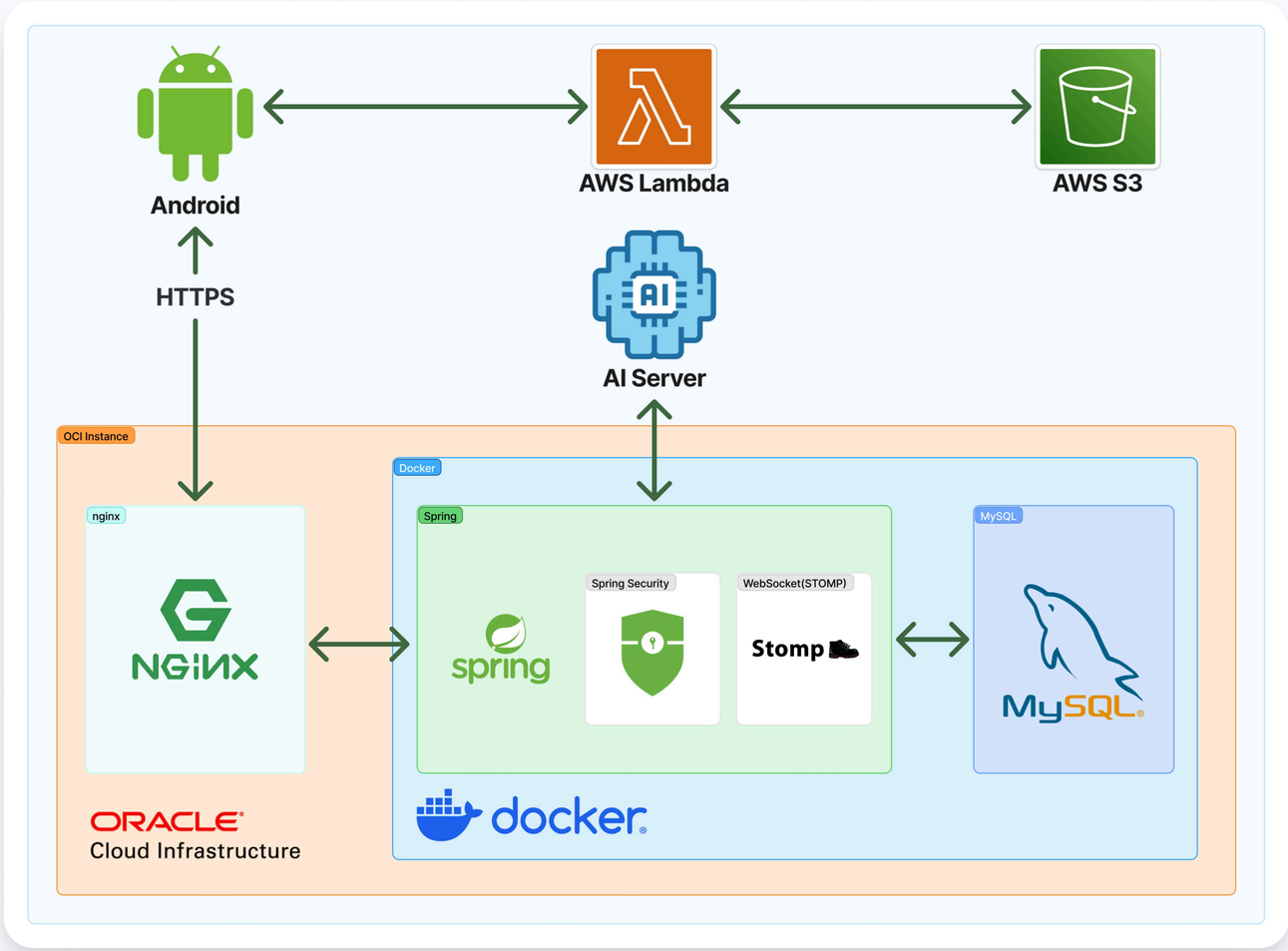
Kotlin, Stomp, Amazon s3...

Backend

Spring, MySQL, Docker...

AI

Fast api, Ollama, DINOv2...



Progress

Backend

Progress - Backend

핵심 프레임워크 및 기술

Spring

JWT

MySQL

Spring Security

STOMP Protocol

Progress - Backend

백엔드 기능 - Auth

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
회원가입	POST, /api/v1/auth/sign-up	False	• 입력값 검증(이메일 형식, 비밀번호 및 닉네임 글자수, 지역 선택 여부) • 이메일/닉네임 중복 검사 • 비밀번호 해시화(Bcrypt) • JWT(Access, Refresh) 발급
로그인	POST, /api/v1/auth/sign-in	False	• 입력값 검증(이메일 형식) • JWT(Access, Refresh) 발급
액세스 토큰 재발급	POST, /api/v1/auth/refresh	False	• 리프레시 토큰 쿠키(httpOnly, Secure) 사용
로그아웃	POST, /api/v1/auth/sign-out	False	• 리프레시 토큰 쿠키 사용 • 서버 측 리프레시 토큰 무효화

Progress - Backend

백엔드 기능 - Locations / Categories

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
지역 목록 조회 및 검색	GET, /api/v1/locations	False	• 페이지네이션 적용
카테고리 목록 조회	GET, /api/v1/categories	False	• 캐시 적용

Progress - Backend

백엔드 기능 - Products

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
상품 목록 조회 및 검색	GET, /api/v1/products	False	• 특정 지역(동네) 기준 반경 10km 이내 상품 목록 조회 기능 • 필터링 기능(카테고리, 최소 및 최대 가격, 검색 키워드, 상품 컨디션, 거래 방식) • 정렬 기능 • 페이지네이션 적용
상품 상세 조회	GET, /api/v1/products/{productId}	False	
내 찜 여부 조회	GET, /api/v1/products/{productId}/favorites/me	True	
상품 관련 채팅방 조회	GET, /api/v1/products/{productId}/chats	True	• 특정 상품 판매자와 나와의 채팅방 정보 조회
상품 구매 희망자(채팅) 목록 조회	GET, /api/v1/products/{productId}/buyers	True	
상품 등록	POST, /api/v1/products	True	• 입력값 검증(제목 및 설명 글자수, 가격, 이미지 주소 형식)

Progress - Backend

백엔드 기능 - Products(2)

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
상품 채팅방 생성	POST, /api/v1/products/{productId}/chats	True	
상품 정보 수정	PUT, /api/v1/products/{productId}	True	• 입력값 검증(제목 및 설명 글자수, 가격, 이미지 주소 형식)
찜 추가 및 해제	PUT, /api/v1/products/{productId}/favorites	True	
상품 상태 변경	PATCH, /api/v1/products/{productId}/status	True	• 판매중/예약됨/판매완료 中 선택 • 판매 완료 시 구매자 선택(옵션) 후 거래 내역 생성
조회수 증가	PATCH, /api/v1/products/{productId}/views	False	
상품 삭제	DELETE, /api/v1/products/{productId}	True	• 소프트 삭제

Progress - Backend

백엔드 기능 - Chats

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
내 채팅방 목록 조회	GET, /api/v1/chats	True	• 정렬 기능 • 페이지네이션 적용
채팅방 상세 조회	GET, /api/v1/chats/{chatId}	True	
메시지 목록 조회	GET, /api/v1/chats/{chatId}/messages	True	• 정렬 기능 • 페이지네이션 적용
메시지 전송	POST, /api/v1/chats/{chatId}/messages	True	• 입력값 검증(이미지 주소 형식) • 상대방에게 웹소켓을 통해 메시지 실시간 전송
사기 탐지 실행	POST, /api/v1/chats/{chatId}/fraud-detection	True	• 최근 n개 메시지 기반 AI 분석
상품 이미지 분석 실행	POST, /api/v1/chats/{chatId}/image-analysis	True	• 해당 상품의 이미지 분석 후 허위매물 판단
메시지 읽음 처리	PUT, /api/v1/chats/{chatId}/messages/read	True	

Progress - Backend

백엔드 기능 - Users

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
특정 사용자 프로필 조회	GET, /api/v1/users/{userId}	False	
특정 사용자의 상품 목록 조회	GET, /api/v1/users/{userId}/products	False	• 정렬 기능 • 페이지네이션 적용

Progress - Backend

백엔드 기능 - Me

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
내 프로필 조회	GET, /api/v1/me	True	
내가 등록한 상품 목록 조회	GET, /api/v1/me/products	True	• 정렬 기능 • 페이지네이션 적용
내가 찜한 상품 목록 조회	GET, /api/v1/me/favorites	True	• 정렬 기능 • 페이지네이션 적용
내 거래 내역 조회	GET, /api/v1/me/trades	True	• 정렬 기능 • 페이지네이션 적용 • 구매 내역/판매 내역/전체 내역 中 선택

Progress - Backend

백엔드 기능 - Me(2)

기능	HTTP 메소드 및 엔드포인트	인증 필요 여부	비고
닉네임 변경	PATCH, /api/v1/me/nickname	True	• 입력값 검증(닉네임 글자수) • 닉네임 중복 검사
프로필 이미지 변경	PATCH, /api/v1/me/profile-image	True	• 입력값 검증(이미지 주소 형식)
내 지역 변경	PATCH, /api/v1/me/location	True	• 입력값 검증(지역 선택 여부)
회원 탈퇴	DELETE, /api/v1/me	True	• 소프트 삭제 • '닉네임(탈퇴)' 표시
프로필 이미지 삭제	DELETE, /api/v1/me/profile-image	True	

Progress - Backend

백엔드 기능 요약

회원가입, 로그인

상품 CRUD 기능

거래 및 거래 내역 조회

사용자 지역 조회, 검색

채팅

Progress - Backend

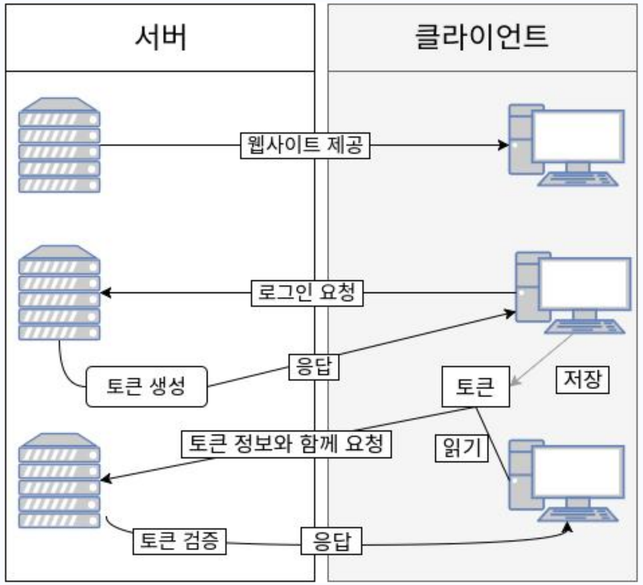
인증 방식

세션 방식의 한계(Scalability 문제)

- 사용자가 많아질수록 세션 저장소의 메모리 부하 증가
- 확장성 문제 - 서버를 Scale-out할 경우, 세션을 모든 서버가 공유해야 하는 문제 발생(eg. Sticky Session, Session Clustering)

대안: 토큰(Token) 기반 인증

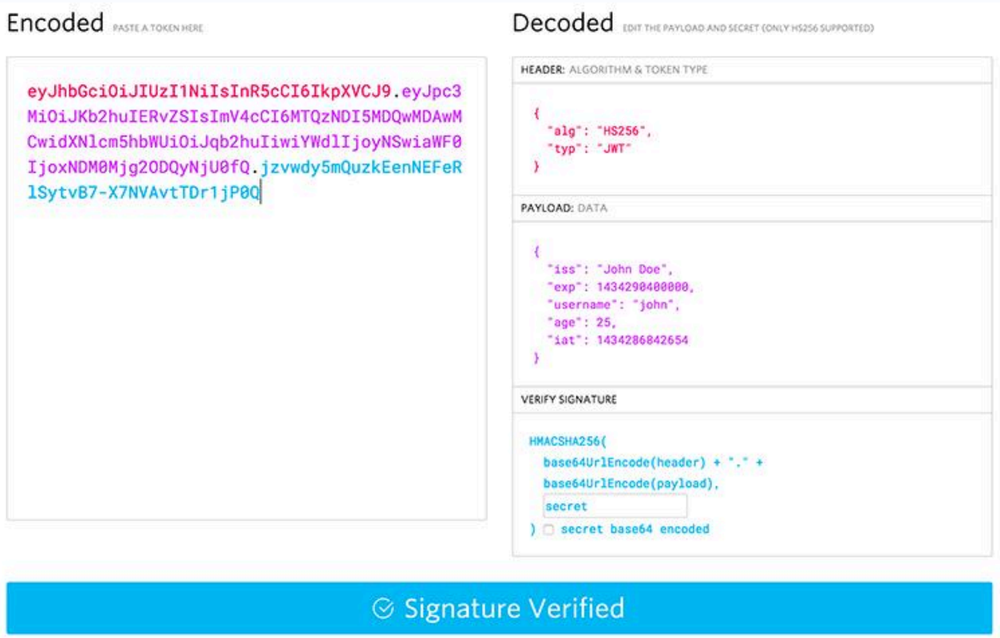
- Stateless(무상태) 인증
- 서버가 인증 상태를 저장하지 않음
- 인증 정보가 담긴 'Token'을 클라이언트가 직접 관리



Progress - Backend
인증 방식

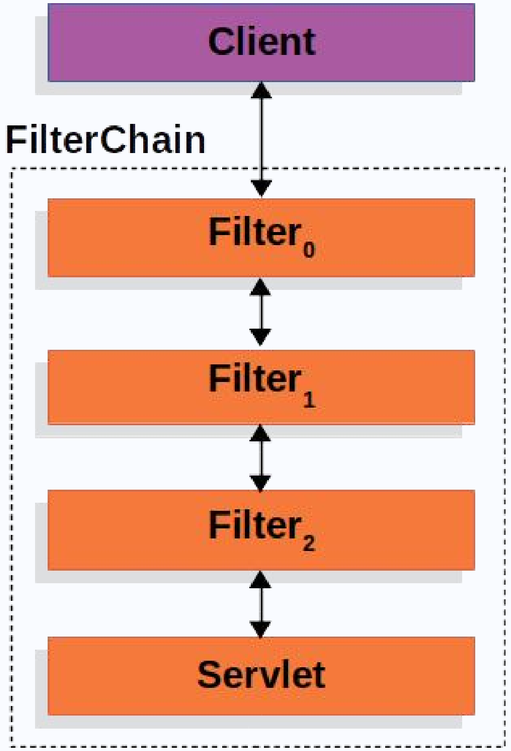
JWT(JSON Web Token)

- Header, Payload, Signature로 구성되어 있으며 각 부분은 Base64로 인코딩되어 있고 '.'으로 구분



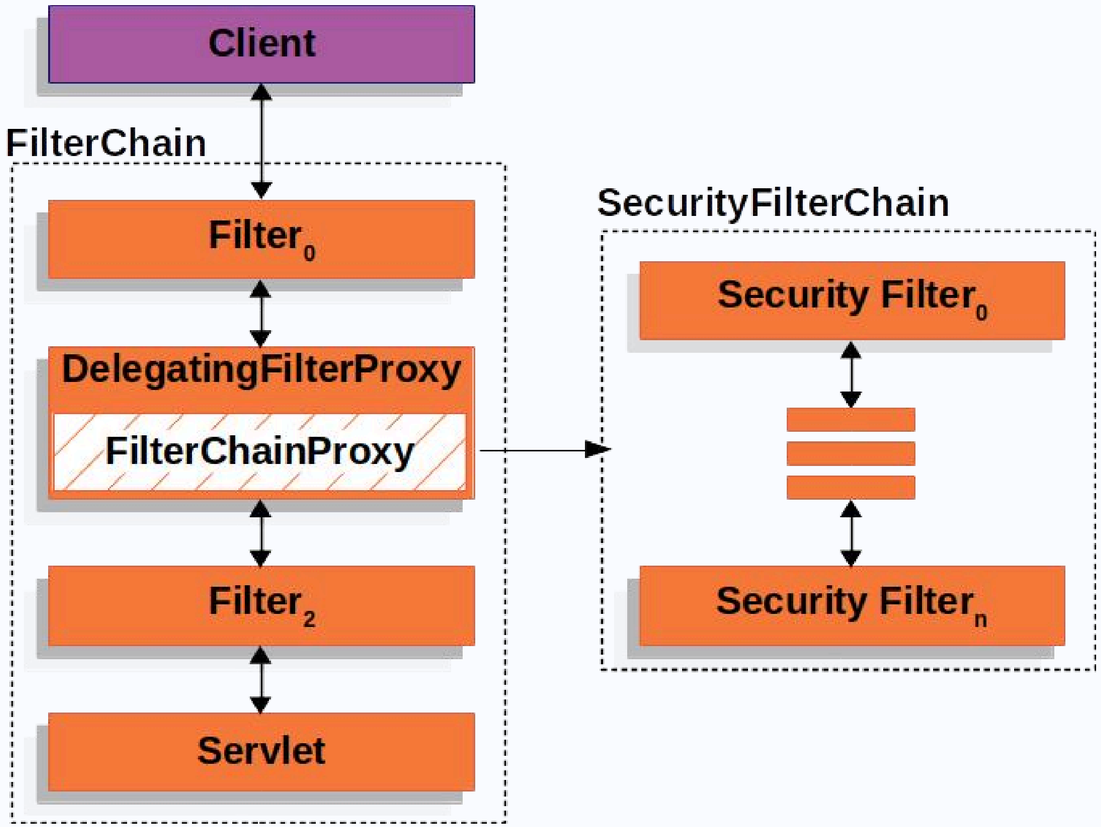
Progress - Backend

Spring Security



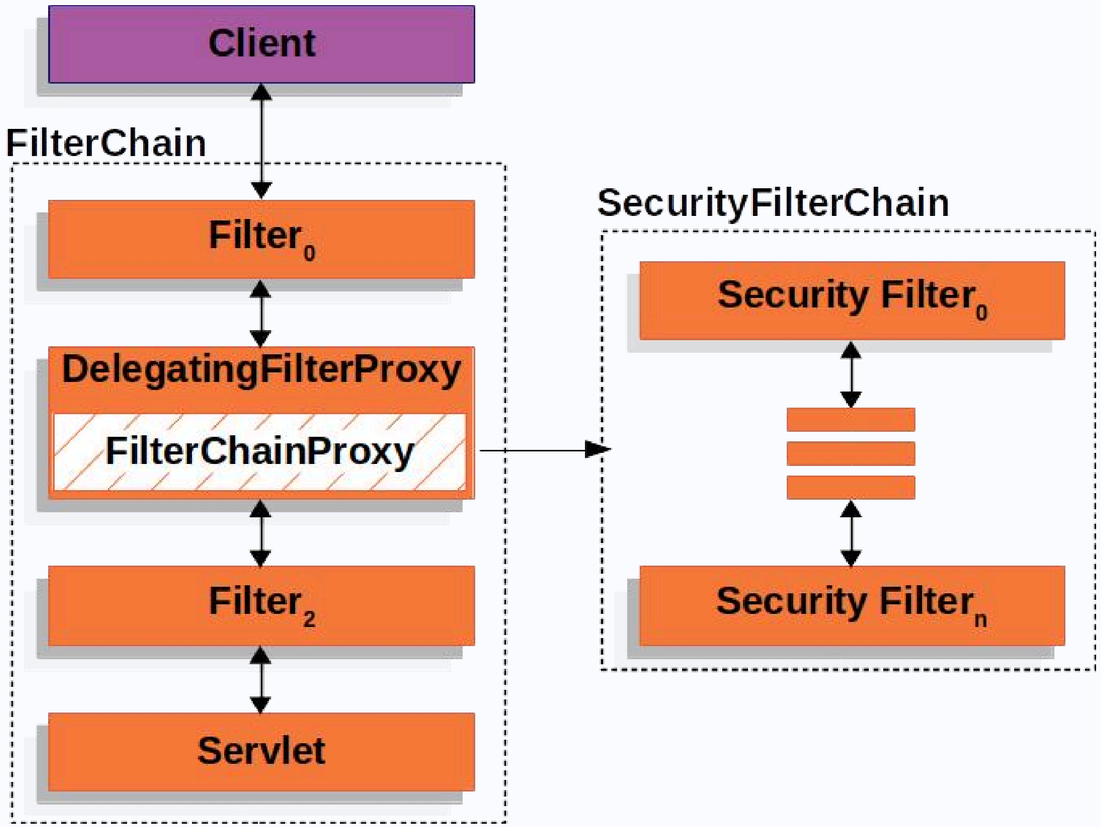
Progress - Backend

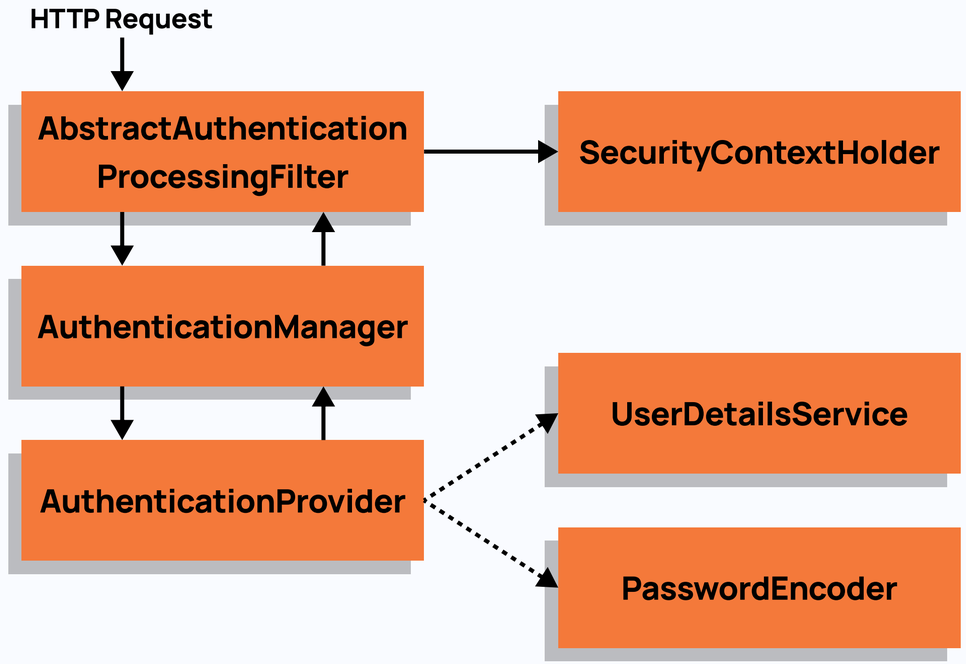
Spring Security



Progress - Backend

Spring Security

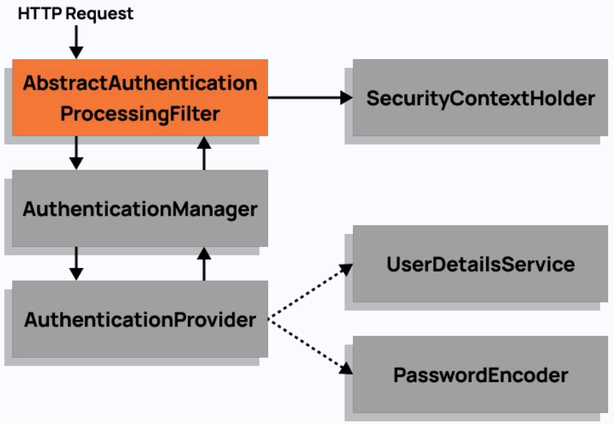




Progress - Backend

Spring Security

HTTP 요청에서 JWT를 추출하여 인증을 시도, 인증에 성공하면 SecurityContextHolder에 유저 정보 저장

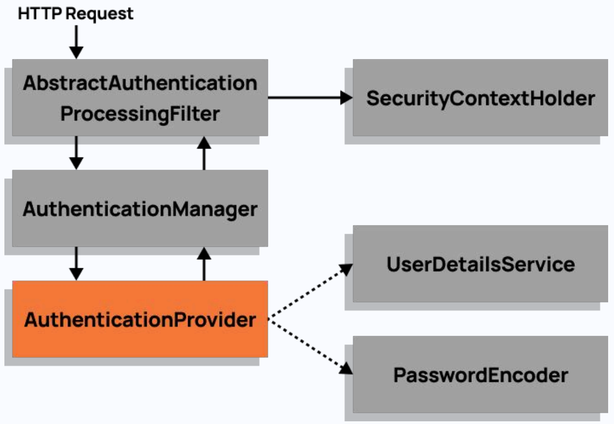



```java
public class JwtAuthenticationFilter extends AbstractAuthenticationProcessingFilter {

  // ...

  @Override
  public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response)
      throws AuthenticationException, IOException, ServletException {
    String token = obtainToken(request);
    JwtAuthenticationToken authRequest = new JwtAuthenticationToken(token);
    setDetails(request, authRequest);
    return this.getAuthenticationManager().authenticate(authRequest);
  }

  @Override
  protected void successfulAuthentication(
      HttpServletRequest request, HttpServletResponse response, FilterChain chain, Authentication authResult
  ) throws IOException, ServletException {
    SecurityContext context = SecurityContextHolder.createEmptyContext();
    context.setAuthentication(authResult);
    SecurityContextHolder.setContext(context);
    chain.doFilter(request, response);
  }

  // ...

```

Progress - Backend

Spring Security

전달받은 JWT 토큰을 검증하고 `UserDetailsService`를 호출하여 사용자 확인 후 최종 인증을 처리

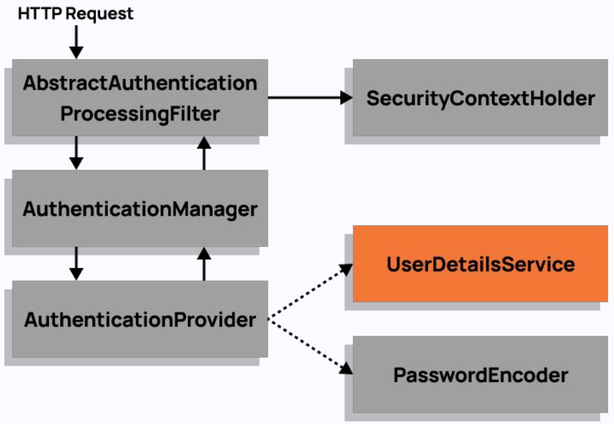



```java
public class JwtAuthenticationProvider implements AuthenticationProvider {

  // ...

  @Override
  public Authentication authenticate(Authentication authentication) throws AuthenticationException {
    String token = ((JwtAuthenticationToken) authentication).getToken();
    if (token == null || token.isEmpty()) throw new BadCredentialsException("토큰이 없습니다.");
    try {
      String email = jwtUtil.validateAccessToken(token);
      RaonUser principal = (RaonUser) userDetailsService.loadUserByUsername(email);
      return new RaonAuthenticationToken(principal);
    } catch (io.jsonwebtoken.ExpiredJwtException exception) {
      if (exception.getClaims().get("type", String.class).equals("access")) {
        throw new ExpiredJwtException("만료된 토큰입니다.");
      } else {
        throw new BadCredentialsException("유효하지 않은 토큰입니다.");
      }
    } catch (JwtException exception) {
      System.out.println(exception.getMessage());
      throw new BadCredentialsException("유효하지 않은 토큰입니다.");
    }
  }

  // ...

}
```

Progress - Backend

Spring Security

인증에 필요한 사용자 정보를 이메일을 통해 데이터베이스에서 조회




```java
public class RaonUserService implements UserDetailsService {

  // ...

  @Override
  public UserDetails loadUserByUsername(String email) throws UsernameNotFoundException {
    Optional<User> optionalUser;
    try {
      optionalUser = userRepository.findByEmailAndIsDeletedFalse(email);
    } catch (Exception exception) {
      throw new UsernameCannotFoundException(exception.getMessage());
    }
    if (optionalUser.isEmpty()) throw new UsernameNotFoundException("User not found with email: " + email);
    User user = optionalUser.get();
    return new RaonUser(user);
  }

}
```

Progress

AI

Progress - AI

이미지 유사도 검사

가볍고 빠른 python
기반 서버

PyTorch

Transformers

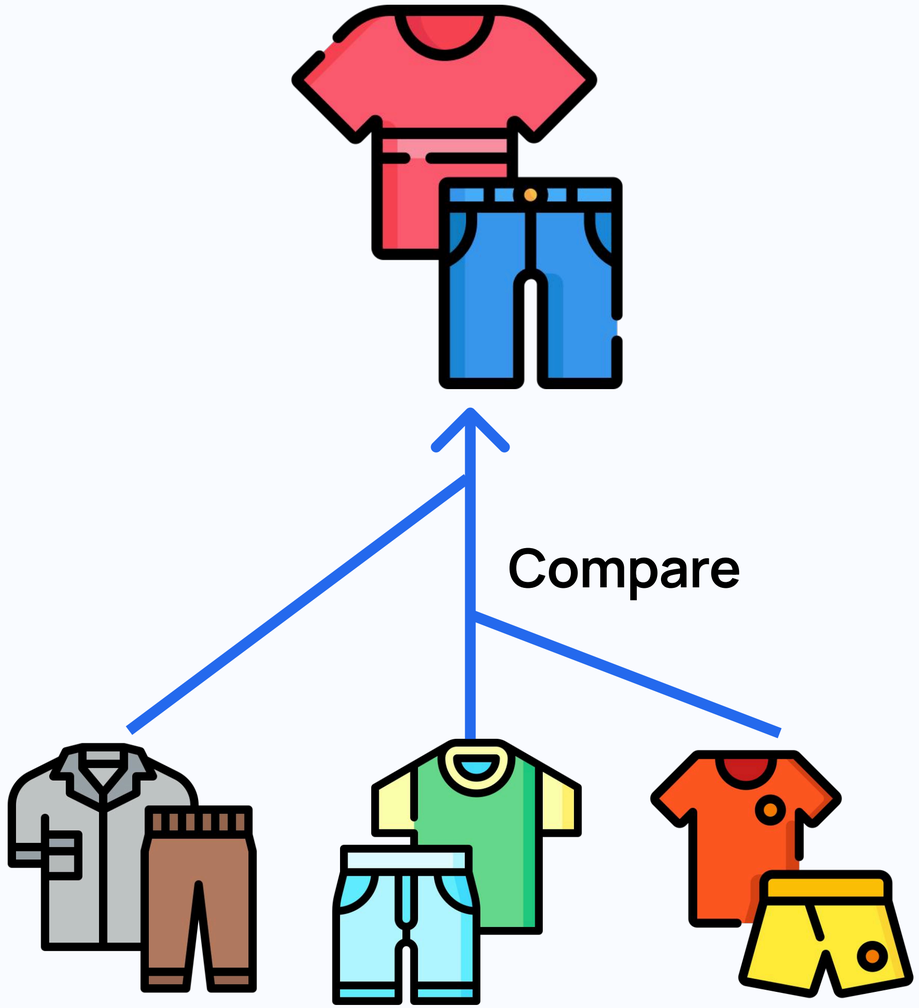
사전 학습된 임베딩 재사용

Meta
DINOv2
SigLIP{**}

이미지 정합성 및 유사도 판별

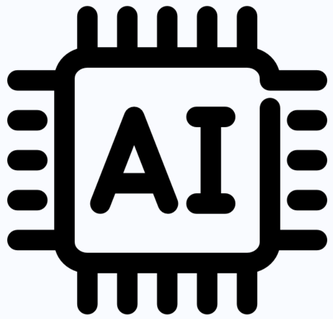
SerpApi

역이미지검색으로
유사 이미지 수집



Progress - AI

이미지 유사도 검사




```
{
 "images": [
  {
   "imageId": 1,
   "imageUrl":"https://...",
  }...
```

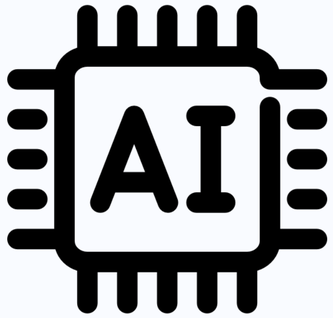
Amazon
S3

Compare

SerpApi

Progress - AI

이미지 유사도 검사




```
{
 "images": [
  {
   "imageId": 1,
   "imageUrl":"https://...",
  }...
```

```
[CATEGORY DETECT] 1위=신발 | 후보=['신발', '자동차/오토바이', '취미/장난감'] | 점수=[-0.06, -0.072, -0.082]
[CATEGORY] 신발
[INFO] Loading DINOv2 (vit_small_patch14_dinov2.lvd142m) ...
[MATCH] https://serpapi.com/searches/68fab93a73bf595c4519e14a/images/64930b72b0a6684727831c53f3fd3c6a10c923adf240a71646c2cf3f3e58fce2.jpeg | score=0.896
[MATCH] https://serpapi.com/searches/68fab93a73bf595c4519e14a/images/64930b72b0a66847737c47fea11812c70d9aebcd207d42a726be3d1d0b2a6ff0.jpeg | score=0.896
[MATCH] https://serpapi.com/searches/68fab93a73bf595c4519e14a/images/64930b72b0a6684753aba439708805289fe41ceac6f7efd2c6471c2a36063f58.jpeg | score=0.896
[TOP SCORE] 0.896
```

```
{
  "results": [
    {
      "imageId": 1,
      "imageUrl": "https://img2.joongna.com/media/original/2025/08/06/1754435492273m3z_fC
      "result": "WARNING",
    }
  ]
}
```

Progress - AI

이미지 유사도 검사


```
{
  "results": [
    {
      "imageId": 1,
      "imageUrl": "https://img2.joongna.com/media/original/2025/08/06/1754435492273m3z_fC
      "result": "WARNING",
    }
  ]
}
```

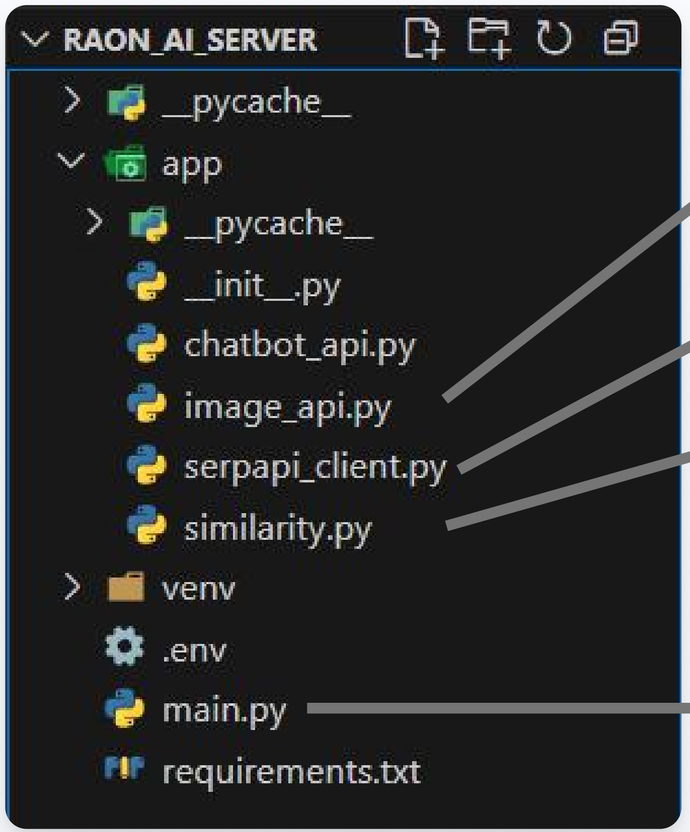
Progress - AI

이미지 유사도 검사


```
{
  "results": [
    {
      "imageId": 1,
      "imageUrl": "https://img2.joongna.com/media/original/2025/08/06/1754435492273m3z_fC
      "result": "WARNING",
    }
  ]
}
```

Progress - AI

이미지 유사도 검사 - 서버 구조



이미지 유사도 검사 REST 엔드포인트

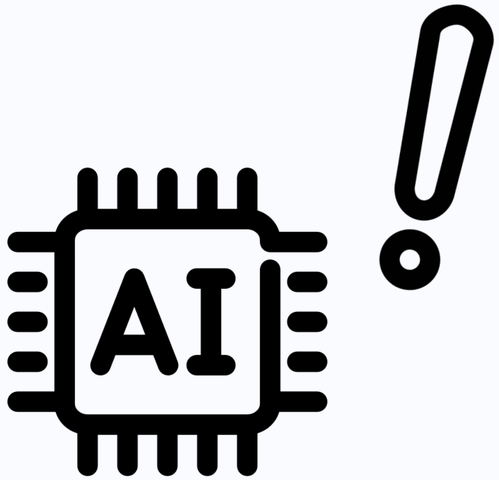
SerpAPI를 이용한 Google Reverse Image Search

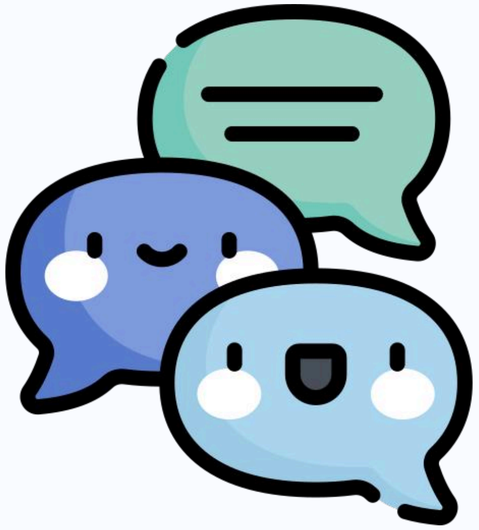
이미지 임베딩과 유사도 계산

Fast api 서버

Progress - AI

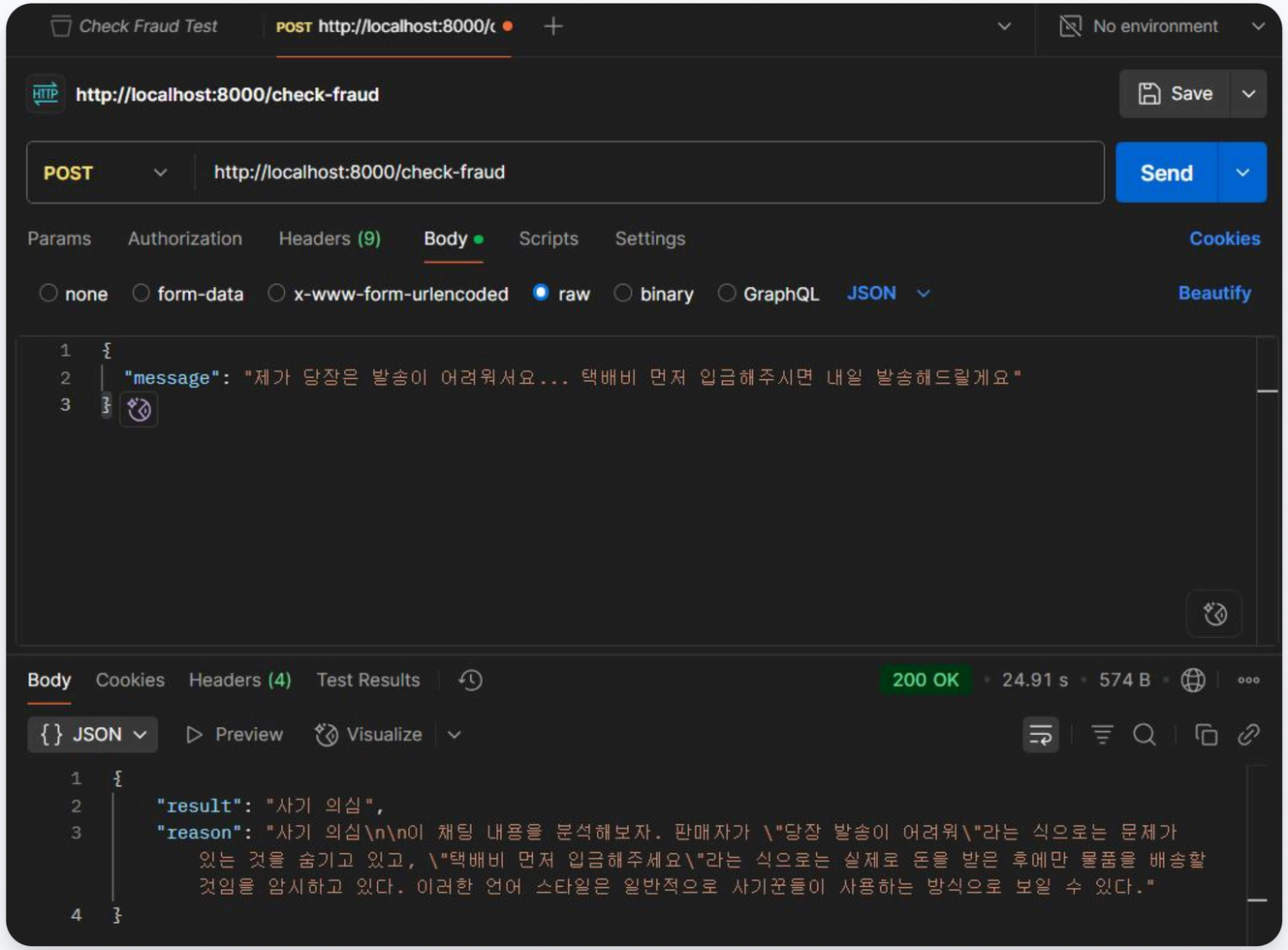
사기 경고 챗봇





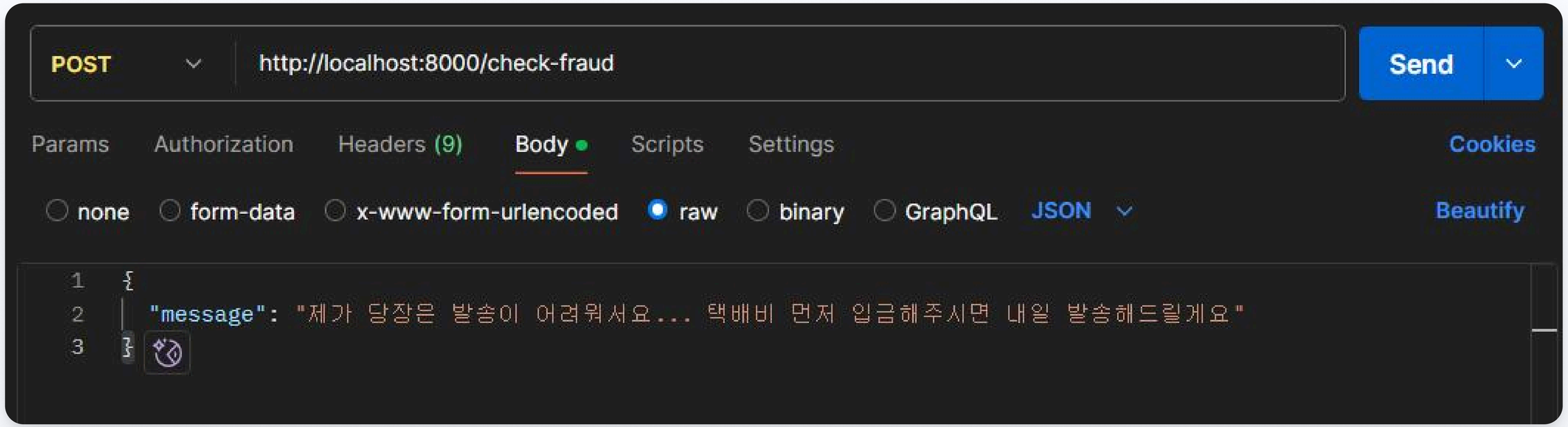
- Ollama 모델 기반 핵심 사기 유형 감시 프롬프트
  - 직접 다루기 쉬운 LLM
  - API 사용에 과금이 없음
- Fast api 프레임워크를 이용해 서버 구동
  - LLM 처리 속도 중시

Progress - AI
사기 경고 챗봇



Progress - AI

사기 경고 챗봇



Progress - AI

사기 경고 챗봇

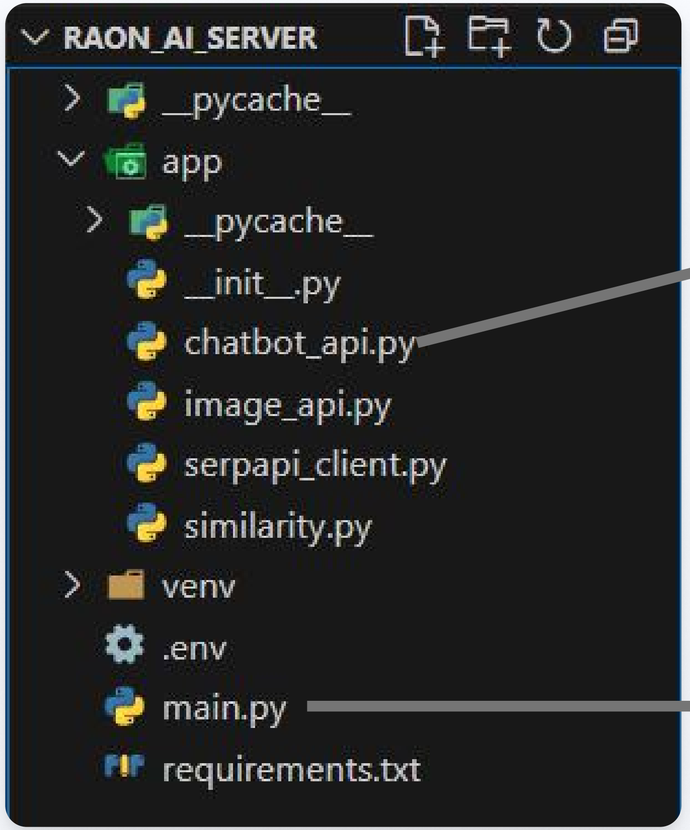
Body Cookies Headers (4) Test Results 200 OK 24.91 s 574 B

{ } JSON Preview Visualize


```
{
    "result": "사기 의심",
    "reason": "사기 의심\n\n이 채팅 내용을 분석해보자. 판매자가 \"당장 발송이 어려워\"라는 식으로는 문제가 있는 것을 숨기고 있고, \"택배비 먼저 입금해주세요\"라는 식으로는 실제로 돈을 받은 후에만 물품을 배송할 것임을 암시하고 있다. 이러한 언어 스타일은 일반적으로 사기꾼들이 사용하는 방식으로 보일 수 있다."
}
```

Progress - AI

사기 경고 챗봇 - 서버 구조



채팅 사기 경고 REST 엔드포인트

- LLM 로드
- 프롬프트
- LLM 응답 전달

Fast api 서버

Progress

APP

Progress - App
기술 스택

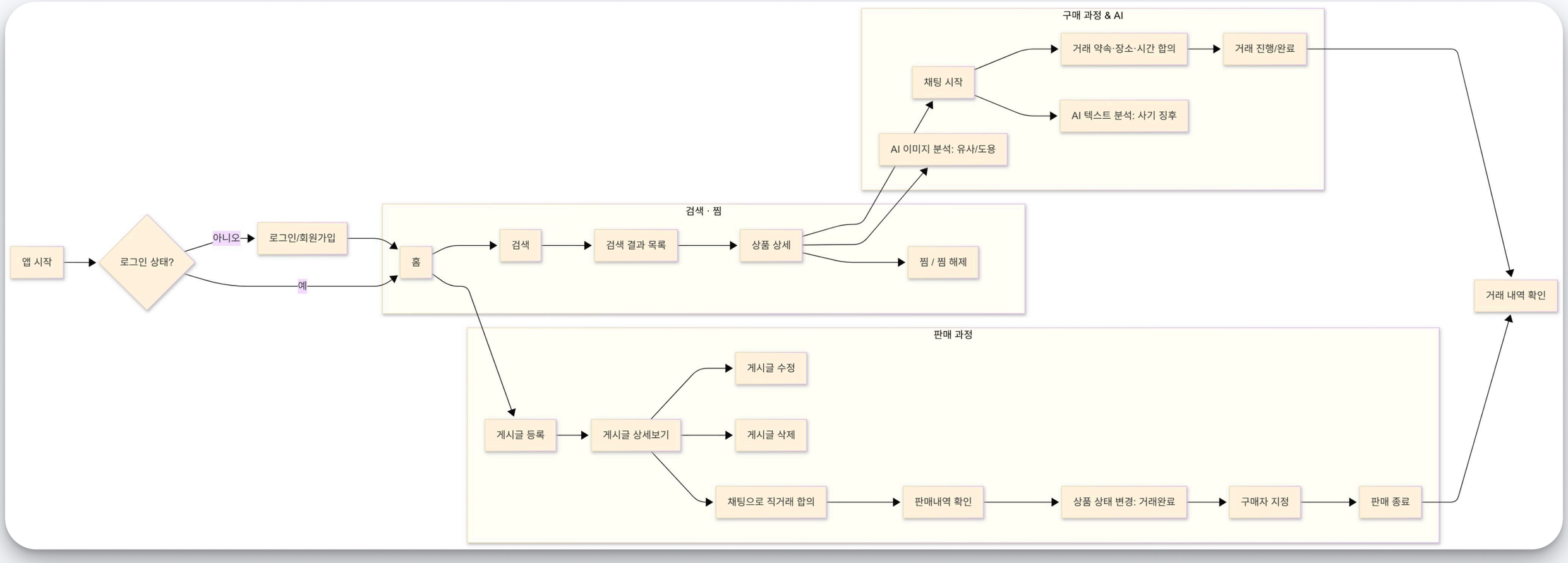
Kotlin

Jetpack Compose

Retrofit

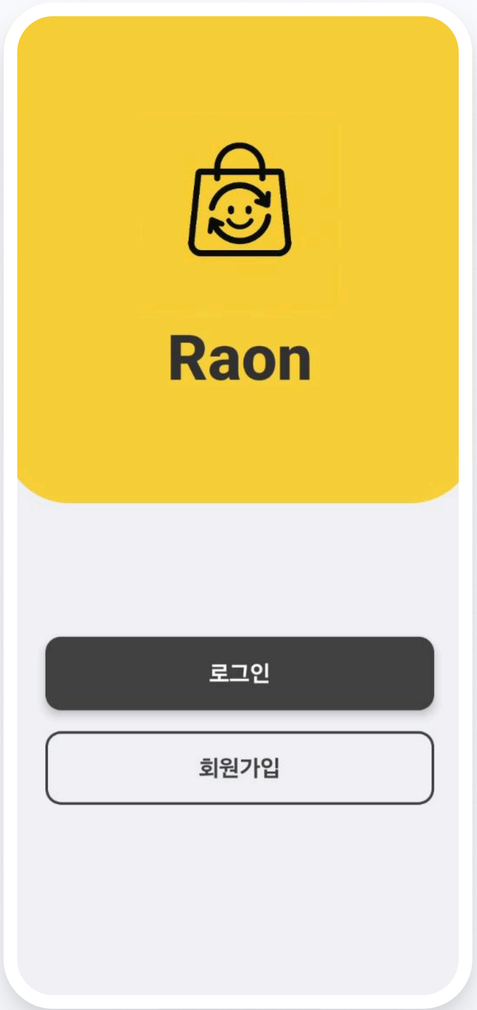
Krossbow

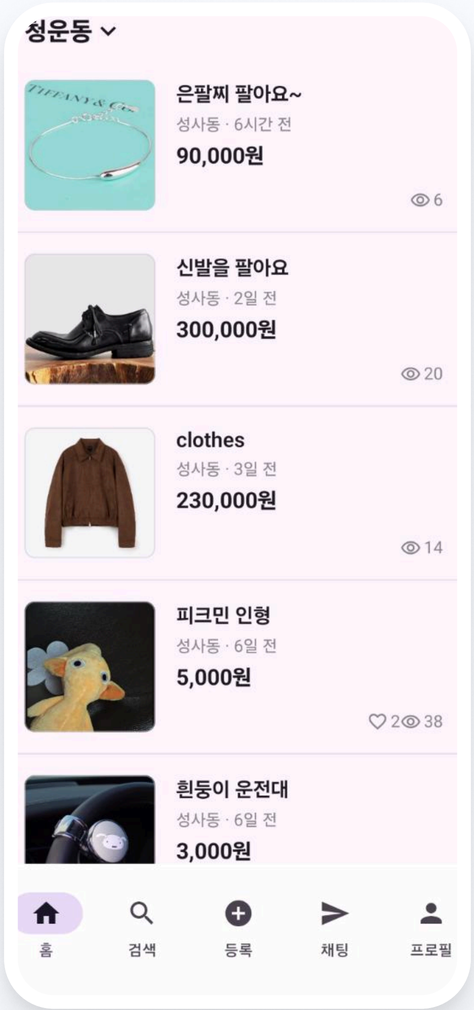
Progress - App
기술 스택

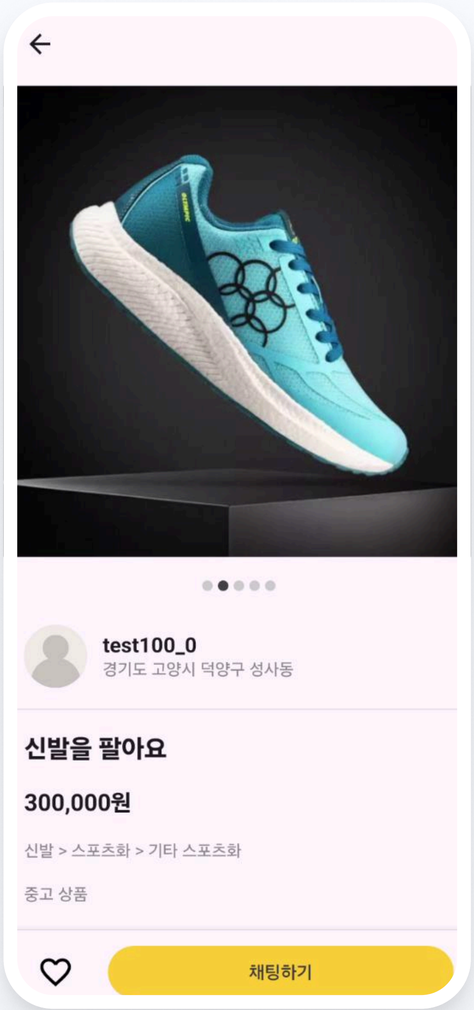


Progress - App

주요 화면

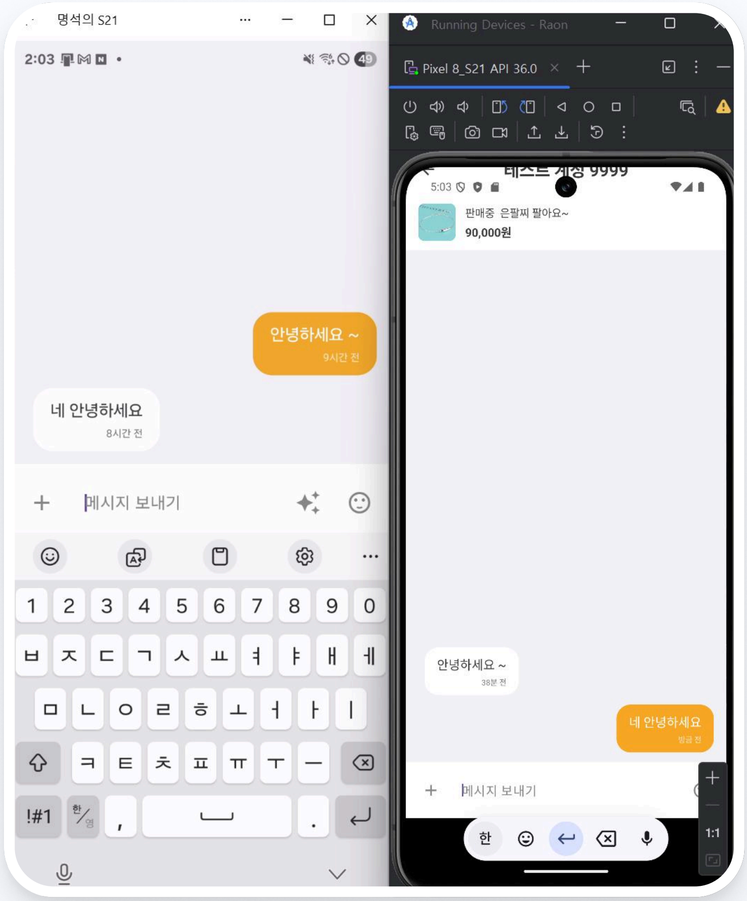






Progress - App

채팅 및 AI 분석 화면



Progress - App

데모 영상

› 데모 영상 보기

Progress - App

다운로드 링크

Try me!

Expected Effect

구매자 - 안전한 거래 경험

- 금전적/심리적 사기 피해 원천 차단
- 거래 피로도 감소 및 만족도 극대화

판매자 - 공정한 판매 환경

- 허위 매물 제거로 정상 판매자 노출 증대
- 빠른 구매 결정으로 판매 속도 향상

플랫폼 - 차별화된 경쟁력

- '안전' 브랜딩을 통한 신규 유입 / 유지
- CS 운영 비용 절감
- 거래 활성화(GMV 증대)

Future Plan

종단간 암호화 채팅

에스크로 결제 방식

AI 모델 성능 향상

Thank You