



2025 오픈소스 개발 프로젝트



RAG 기반 QR코드 피싱 여부 조회 시스템

RAG Based QR Phishing Checker Service

- 일시 2025년 12월 12일
- 팀명 13팀 - 404 FOUND
- 팀원 2024042053 김여민,
2024042050 심연우,
2024042016 최수연

CONTENTS

01

팀 소개

404 FOUND

02

서비스 개요

배경 및 문제 정의

03

서비스 주요 기능

분석, RAG, 챗봇

04

개발 과정

개선 점, 협업 방법

05

시스템 구조

아키텍쳐, db

06

프로젝트 시연

QRex

07

후기 & 향후 발전 방향

느낀점, 향후 계획

~~NOT~~

404 FOUND

겉보기에는 오류와 한계로 보이는 상황 속에서도 해결의 실마리를 찾아내고,
존재하지 않는 것처럼 보였던 가능성을 발굴하겠다는 의지



김여민

2024042053



심연우

2024042050



최수연

2024042016

각 팀원이 전문성을 발휘하는 파트에 집중하면서도
기획, 개발, 디자인 등 전 과정에 협력하여 진행



- 서비스 이름 선정 이유

'QR' 코드와 최강의 포식자 공룡인 T-Rex'의 결합으로, 교모하게 숨어있는 QR 피싱 링크를 찾아내어 사용자에게 가장 안전한 디지털 경험을 제공하겠다는 의미

• 서비스 선정 배경



Image 1: Number of QR code phishing cases reported in June 2023

These **quishing statistics** show that from June to August 2023, a startling surge in QR code phishing emails was observed. Systems detected 8,878 such incidents, indicating a worrying shift in cybercriminal tactics. June witnessed the peak of this trend, **with 5,063 reported cases in QR code phishing statistics.**

큐싱(Qshing) 해킹 대응 방안 연구

이동건*, 황규준**, 김동오***, 황진석****

Research on Qshing hacking response measures

Dong-Geon Lee*, Gyu-Jun Hwang**, Dong-Oh Kim*** Jin-Suk Hwang****

최근 스마트폰의 보급이 과거에 비해 빠르게 확대되고 있고, 남녀노소 구분없이 다양하게 사용되고 있다. 또한 IT 시대라는 말에 걸맞게 여러 가지 편리한 방법을 사용해 정보에 접근할 수 있게 되었다. 예를 들어 지금 근처에서 많이 보이는 **QR 코드를 활용하여 정보에 접근하는 방법이** 있다. QR 코드가 많이 상용화되고 있는 이 시점에 QR 코드를 악용하여 타인의 정보를 빼앗으려는 큐싱이라는 해킹 기법과 그 대응 방안에 대해 풀어보려고 한다.

● 서비스 선정 배경

“QR코드 찍었다가 털린다” 신종 피싱 수법 기승

유창수 기자 | 입력 2025.09.04 16:13 | 댓글 0

최근 QR코드와 안내문을 악용한 신종 사기 수법이 전국적으로 잇따르면서 경찰이 주의를 당부했다. 단순 문자 스미싱을 넘어 실제 벌금 고지서나 우편을 안내문처럼 꾸며 시민들을 속이는 방식이다.

가장 흔한 수법은 차량에 불법주차 단속 스티커처럼 보이는 종이를 붙여놓고, ‘벌금 납부’를 명목으로 QR코드 접속을 유도하는 것이다. 하지만 해당 QR코드를 스캔하는 순간 악성 앱이 설치돼 개인정보와 금융정보가 고스란히 빠져나간다. 벌금이나 과태료는 반드시 지자체 공식 앱이나 홈페이지에서만 확인해야 한다.

문 앞에 ‘우편물 도착 안내서’를 붙여두는 경우도 늘고 있다. 안내서에 찍힌 QR코드를 스캔하면 역시 피싱 앱이 설치된다. 실제 우편 여부는 우체국 공식 앱이나 콜센터(1588-1300)를 통해 확인 가능하다.

보이스피싱도 한층 교묘해졌다. 법원·검찰 수사관을 사칭해 “시간이 없다”며 IP 접속을 요구하고, 경찰청 홈페이지와 유사한 가짜 화면을 띄워 개인정보 입력을 유도하는 방식이다. 하지만 실제 수사기관은 개인 전화로 사건을 안내하거나 개인정보를 요구하지 않는다.

최근에는 음식물 처리 위반 통보 문자, 공유 키보드에 덧씌운 가짜 QR코드까지 등장해 피해가 확산되고 있다. 경찰 관계자는 “조금이라도 수상한 문자나 안내문을 받으면 절대 개인정보를 입력하지 말고 즉시 112 또는 118(사이버 수사대)에 신고해야 한다”며 “특히 QR코드 접속은 각별히 주의해야 한다”고 강조했다.

전문가들은 신종 사기 피해를 막기 위해 △벌금·과태료는 반드시 지자체·공식 홈페이지 확인 △우편·등기 안내는 우체국 콜센터 확인 △수사기관은 절대 개인전화로 사건을 안내하지 않음 △의심 문자는 즉시 신고할 것 등을 필수 예방 수칙으로 꼽았다.

바이낸스, 얼굴 인증 피싱 사기 경고...QR 코드 통한 계정 탈취 시도 급증

손정환 기자 | 2025.06.16 (월) 18:53

1 2

글로벌 암호화폐 거래소 바이낸스가 사용자들을 향한 새로운 유형의 피싱 사기 수법에 대해 경고하고 나섰다. 사기범들은 바이낸스 고객으로 위장해 영상 촬영이나 QR 코드 스캔을 유도하면서 이용자의 얼굴 데이터를 탈취하려는 시도를 벌이고 있다. 회사 측은 얼굴 인증 절차를 빌미로 한 영상 요청이 실제로는 계정 탈취를 위한 함정일 수 있다며 주의를 당부했다.

경고에 따르면, 피싱 사기범들은 바이낸스 고객센터나 기술지원 직원을 사칭해 “보안 검증”을 명목으로 피해자에게 접근한다. 이후 보안 점검을 이유로 QR 코드 스캔이나 얼굴 영상 제출을 요구하며, 해당 데이터를 악용해 계정 접근 권한을 확보하려는 시도다. 이는 바이낸스의 얼굴 인식 시스템을 우회하려는 목적을 갖고 있으며, 영상 인증을 단순한 절차로 오인하도록 유도한다.

특히 이들은 왓츠앱(WhatsApp) 등의 메신저 앱을 통해 위조된 QR 코드를 전달하거나, 개인 정보 입력을 유도하는 메시지를 보낸 뒤 악성 행위를 이어가는 전략을 구사한다. 이에 바이낸스는 공식 채널 외에 의심스러운 요청이나 보안 프로세스를 받은 경우, 절대 응하지 말고 즉시 신고하라고 권고했다.



주요 기능

01 실시간 QR 코드 분석

카메라나 갤러리 이미지를 통해 QR 코드를 인식하고, URL의 피싱 여부를 즉시 판별

02 RAG 기반의 정확한 판별

RAG 모델을 활용해 최신 피싱 사례 및 데이터와 결합 분석 단순 블랙리스트가 아닌 근거 기반 정교한 판단 제공

03 커뮤니티를 통한 피싱 예방

사용자들이 직접 피싱 사례를 공유하는 게시판 제공
댓글 신고 기능으로 건전한 커뮤니티 유지

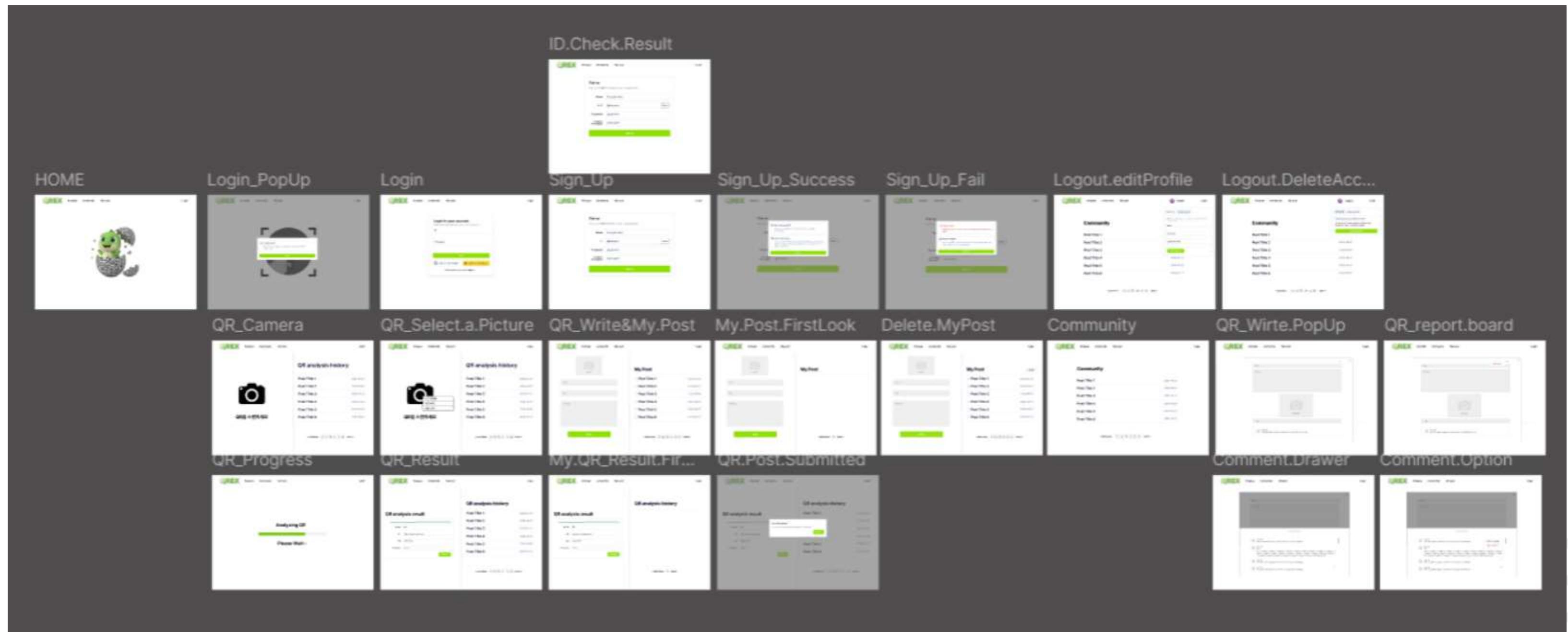
04 개인 맞춤형 기록 관리

사용자가 검사한 QR 코드 기록을 저장하고 언제든지 다시 확인할 수 있도록 관리 기능 제공

05 사용자 경험 강화를 위한 챗봇 Agent

ChatBot Agent는 서비스 사용법 안내, 게시글 작성·삭제, 분석 기록 제목 변경 등의 기능을 통해 사용자와 상호작용하며 서비스 편의성 향상

개발 초기



주변 사람들의 조언

초기 개발 직후 비공개 배타 테스트 진행

충북대학교 강재구 교수님

서비스에 대한 사용 방법 등을
설명해주는 챗봇 agent가
있으면 좋겠다



서울여자대학교 첨단미디어디자인학부 2학년 강00

서비스 설명이 아래 있는지
모르겠어서 표시같은거
있으면 좋겠다



충북대학교 소프트웨어학과 4학년 박00

서비스에 대한 설명이 없어서
메인화면에 이걸 확인할 수 있게
추가했으면 좋겠다

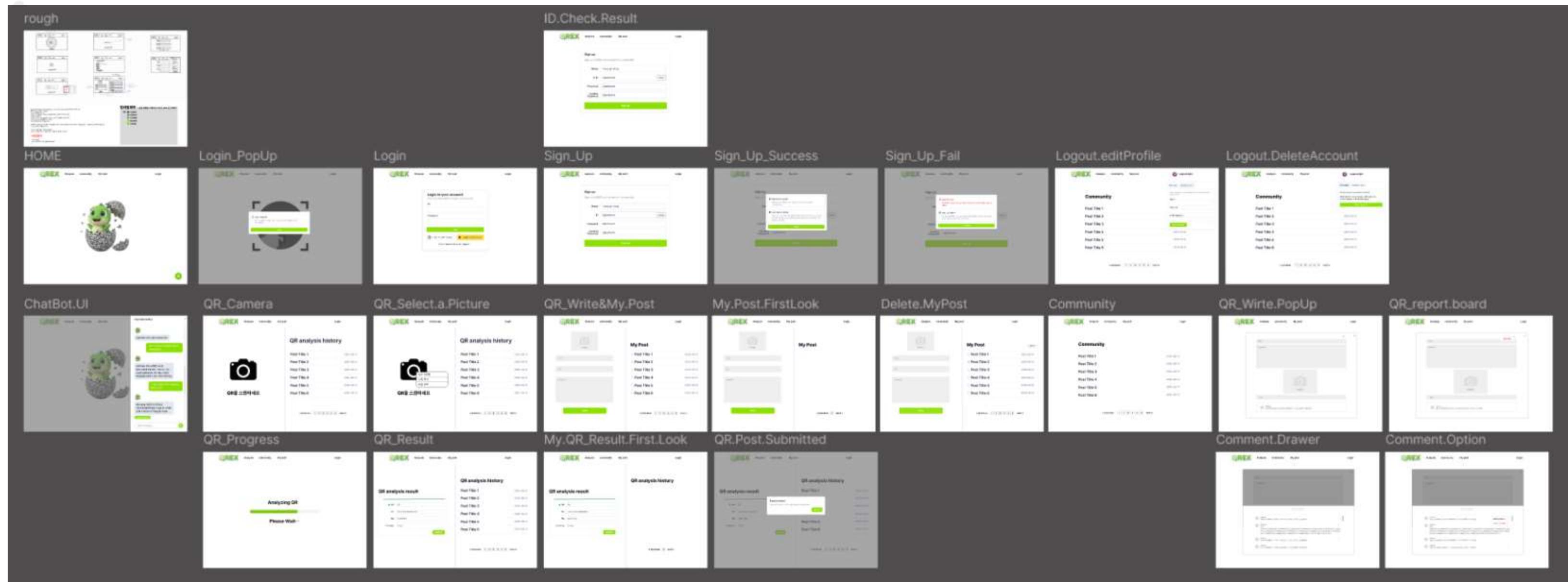


충북대학교 소프트웨어학과 4학년 배00

https 프로토콜 사용해서 보안을
강화했으면 좋겠다.
글 작성할 때 존재하지 않는 url
넣으면 알림해줬으면 좋겠다.



주변 피드백 수용 후



협업 방식 Notion

The logo for QRex Frontend Client, featuring a blue square icon with a white QR code and the text "QRex Frontend Client" in a bold, sans-serif font.

백엔드 (Backend)

1. Deployment Architecture (배포 환경)

백엔드 서비스는 안전적인 서비스 제공을 위해 멀티-tier 아키텍처 환경에서 구동되며, 정식 배포만을 염두하여 고려합니다.

- Server Environment: Linux (Ubuntu/CentOS)
- Domain Connection: <https://www.letsencrypt.org> 도메인과 연결되어 있으며, DNS 설정을 통해 트래픽을 프록시로 관리합니다.
- Process Management: `Node.js` 사용하여 애플리케이션을 백그라운드에서 실행하고, 비안이 솔루션으로 서비스가 올바르게 멀티-tier 구성했습니다.

```
# 환경 설정 (로그 경로 설정 포함)
root@jaws-jar:~/src/backend.jar# java -jar backend.jar --logging=INFO
```

- CORS Policy: 프론트엔드 (React) 도메인에서의 API 요청을 허용하기 위해 `cors_ORIGIN` 메시지 `allowCredentials`를 허용하도록 설정하여 보안 위협을 차단했습니다.

2. Directory Structure (상세 구조)

프로젝트 구조는 안전한 표준을 명기해 분명하게 설계했습니다.

```
src/main/java/com/taeho/api
+ controller/          # 컨트롤러 폴더 (Auth, Community, Analytics, Alipay)
+ service/             # 서비스 폴더 (BoardService, AuthService, ...)
+ repository/          # 리ポ지토리 폴더 (User, Board, Analytics)
+ domain/              # 도메인 폴더 (User, Board, Analytics)
+ security/            # 보안 폴더 (Jwt, Bcrypt, OAuth ...)
+ dto/                 # DTO 폴더 (Request, Response)
+ config/               # Swagger, WebConfig 폴더 구조
```

▶ 자세 설명

3. Backend Development Progress

1. DB 설계 및 API 명세 (Architecture)

- ERD 설계: [ERD](#) ([MySQL](#), [Oracle](#)) 간의 관계 설정. 초기에는 단순했으나 [Spring](#) ([MongoDB](#)) 가능으로 전환 관계를 확장합니다.
- API Docs: 양한 흐름을 위해 [Swagger](#) ([OpenAPI](#)) 을 초기부터 도입하여 트렌드에도 쉽게 맞설 수 있습니다.

2. 인증 시스템 구조화 (Auth System)

- OAuth 2.0 페일: 카카오/구글/로그인 연동 ([OAuth2Client](#)) 구현 클래스를 가상으로 DB 적용 조작 구현.
- JWT 보안 강화: 단순히 토큰을 발급하는 것을 넘어, 토큰에 처리에 대한 고민.
 - [Solution] [InMemoryTicketService](#) 주제로 토큰을 요청한 토큰을 Redis 또는 In-memory Set에 저장하고, 토큰에 대해서 해당 토큰의 접근을 차단.

3. AI 서버와의 연동 설계 (Integration)

- [Problem] 프론트엔드에서 AI 서비스 요청 표정 시, API 기기 노출 및 CORS 문제 발생.
- [Solution] Backend as a Proxy (Gateway) 패턴 적용.
 - [KtorProxyController](#) 를 생성하여 모든 AI 요청을 빅뱅드가 중재.
 - 미 개발에서 사용자의 `requestID` 를 AI 서비스로 함께 전달하여 대화 커넥션을 유지 가능 구현.

4. 외부 API 및 유저리티 통합

- Geoplugin 해제: 전세계 IP를 분석해 국가/도시 정보를 DB에 할당. ([GeoIP2](#) ([GeoLite2](#)) 활용).
- Safe Browsing: AI 분석 API 구글 API를 통해 멀티적으로 검색된 맵별 URL 밀다운 분석 추가.

5. 배포 (Deployment)

- Server: Linux 환경 ([PC Server](#) / [Cloud](#))
- Domain: <https://www.letsencrypt.org> 도메인 구매 및 연결.
- Process: `Node.js` 을 이용한 백그라운드 실행 및 보고 관리.

RAG & Agent(Spring ai)

단순 풋풋의 마린, 행동하는 에이전트

기존 Python API를 넘어서, Spring AI를 통해 샐리프라이즈급 인형성을 갖춘 자동형 보안 서비스입니다.
Chroma DB를 활용한 벡터 검색과 Function Calling을 결합하여 강력한 능동형 보안을 제공합니다.

💡 Core Algorithms

1. 벡터 대비베이스 기반 RAG

- Embedding: 브랜드 가이드, 미션 세부, 헤미즈리스트 데이터를 고차원 벡터로 변환하여 Chroma DB에 저장
- Semantic Search: “본인 이 마침 예술”라는 모호한 말도 “제한 적용 분야” 전반의 시사 제정학의 철학과 습무선을 찾음

2. Hybrid Analysis (Dual-Path)

- Fast Path (Rule-Based): 규칙적 계산된 퀘리스/플레이스로 QDRNN 내 속시 판단
- Deep Path (AI Reasoning): 판단 불가 시, Gemini 2.5가 LLM 구조, 디어포스팅, RAG 컨버스를 통합하여 실행 후 판

3. AI Agent & Function Calling

- 단순 단답 형식을 넘어, 사용자의 의도를 파악하고 Java 메소드를 직접 실행합니다.
- Examples: “당신은 금지 지역에?” → `isAreaForbidden()`: 실행 → “당해 발표했습니다” 응답

🛠 Tech Stack

- Language: Java 21 (JTS)
- Framework: Spring Boot 3.3.5 + Spring AI (1.1.0-M4)
- Vector DB: Chroma DB (Local/Docker)
- LLM Model: Google Gemini 1.5 Hash lbs (Agent), 2.5 hash (금증 분석)

⚡ Key API Logic

1. 최상 위정도 행정 문서 ([GET /docs/drive](#))

Rule Check → Vector Search (문서 사이의 경색) → LLM Reasoning (작용 전략)

2. AI 에이전트 대화 ([GET /api/agents/{chat}](#))

Intent Classification → Context Retrieval (Static RAG) → Function Calling (Action)

◆ RAG Server (Spring AI) Development Log

1. 🚧 Migration Decision (Python → Java)

프로젝트 Python(FastAPI) 버전 업을 원. 벽면도 원전설치 주인이 Java에게 기술 스펙 종합 필요성 대우.

- [Goal] Spring의 강력한 생태계(DL, Security)를 AI 퍼미트라인에도 적용하자.
- [Action] [Spring AI](#) 리액티비티를 스피드업하고, 기존 Python LangChain 코드를 Java 코드로 1:1 호환 시키.

2. 💡 Vector Store & RAG Pipeline 구축

- Data Preprocessing: 파싱 URL 데이터, 브랜드 가이드 등 미방형 데이터를 로드하고 헛된 Chaining이라는 ITI [https://tinyurl.com/5yqzjw3t](#) 구조.
- Vector DB: 드론 경계 이미지를 위해 [Chroma DB](#) 커스텀화 활용. 범위 및 보강을 통해 벡터스토어 퍼미트라인 지원.
- Search Logic: 단순 키워드 큐레이션 [Searcher / Search](#); 구현 차단에 “본인 이 마침 예술” → “제한 적용” 문서 큐레이션.

3. 💬 AI Agent & Function Calling 구현 (핵심)

- [Challenge] AI가 단순히 답변만 하는 것이 아니라, DB를 조회하거나 사용하는 행위를 적어 만들고 실행.
- [Solution] Spring AI의 [Function Calling](#) 기능 활용.
 1. [Spring AI](#)에 `def __init__(self, get_llm, get_vectorstore)` 높은 패스드를 [https://tinyurl.com/2k44x64c](#)로 등록.
 2. 프론트엔드 “너는 도구를 사용할 수 있는 에이전트다”라고 명시.
 3. 사용자 입력의 의도를 파악하여 최적 스코프 Java 메소드를 호출하는 조건 체크.

4. 🛠️ 배포 환경 (Infrastructure)

- Environment: 고성능 연산의 필요한 AI 모델 특성상, 글라우드 조각화의 대신 고사양 PC Server를 활용.
- Networking: 포드워드 및 멘션에 설정을 통해 매번 캐리온 디바이스에서 다른 접근 가능한 대로 노출된 대량원 구동.

UI/UX Design

반응형 웹 디자인

UX Highlights

1. Landing Page & Animations

- Video Background: 고화질 영상 배경과 [Framer Motion](#)을 활용한 스크롤 애니메이션으로 보안 서비스의 신뢰감과 불안감을 제공합니다.
- Scroll Lock: 모바일 환경에서의 퍼그리운 경험을 위해 브라우저 하단/혹은 맨 위 스크롤 헤더 코딩 ([scroll-lock.js](#))을 적용했습니다.

2. Split View Dashboard (PC)

- Restactable Panel: PC 환경에서는 분석(Analysis)과 커뮤니티(Community) 페이지에서 자유 확장 크기를 마우스로 자유롭게 조정할 수 있습니다. 이를 통해 사용자는 정보를 동시에 비교하여 생산성을 높일 수 있습니다.

3. Responsive Design

- Mobile Optimized: 모바일에서는 탭(Tab) 방식의 대신 자동 전환되어 익숙한 환경에서도 패st적인 사용성을 제공합니다.

UI/UX Design Process

1. 💡 컨셉 도출 및 아이디에이션

- Keywords: 신뢰(Trust), 경고(Alert), 속도(Speed).
- Color Palette:
 - Main: Deep Navy ([#0E172A](#)) : 보안과 신뢰감 형성.
 - Point: Vibrant Red/Orange : 위험 표시에 대한 직관적 경고.
- Logo Design: QR 코드와 콩크(T-Rex)를 형상화하여 "강력한 팀지" 이미지 시각화.

2. 📐 와이어프레임 및 레이아웃 구조화

- PC View: 정보 흐름의 흐름성을 위해 Split View (좌: 뉴스 / 우: 상세) 차별.
- Mobile View: 익숙한 환경에서의 사용성을 위해 Bottom Navigation & Tab 구조 차별.

3. 🖼 반응형 웹(RWD) 구현 전략

- Figma Auto Layout: 디자인 단계부터 반응형을 고려하여 모바일 미리보기 할뿐.
- Breakpoint 설정: Tailwind 기준 ([320px](#)), ([1024px](#)) 를 기점으로 UI 구조가 드라마틱하게 변경되도록 설계 (햄버거 메뉴 -- GNB).

4. ✨ 인터랙션 고도화

- Framer Motion: 단순한 페이지 전환을 넘어, 스크롤에 따른 표시 등장 애니메이션 등을 추가하여 고급스러운 사용자 경험 제공.
- Loading UX: 분석 대기 시간(약 3초) 동안 지루하지 않도록 스냅백은 UI와 진행률 바(Progress Bar) 디자인 적용.

- 🚀 QREX: RAG 기반 QR코드 피싱 방지 시스템
 - 💡 Service Identity: QREX'QR Code' + 'T-Rex'
 - ⌚ Tech Migration: FastAPI → Spring AI
 - ❓ Why 'RAG (Retrieval-Augmented Generation)'?
- 🌟 Key Features
- 👤 Team & Tech Stack
- 👉 Team Name: 404 FOUND
- 👩 Role
- 🛠 System Architecture
 - 💻 프로젝트 상세 자료
 - 📅 회의록
 - 📎 시스템 정의서
 - 📎 요구사항 정의서
 - 📎 기능적 요구사항
 - 📎 Figma UI/UX 디자인
 - 📎 데이터베이스(DB) 스키마
 - ⚙️ 기술 아키텍처 및 상세 구현

협업 방식 Jira

프로젝트 QRex ...

요약 목록 보드 캘린더 타임라인

목록 검색  필터 ▾

| | 유형 | 키 | 요약 |
|--------------------------|---|--------|----------|
| <input type="checkbox"/> | >  | KAN-2 | UI |
| <input type="checkbox"/> | >  | KAN-3 | Backend |
| <input type="checkbox"/> | >  | KAN-4 | Frontend |
| <input type="checkbox"/> | >  | KAN-5 | RAG |
| <input type="checkbox"/> | >  | KAN-6 | DB |
| <input type="checkbox"/> | >  | KAN-8 | Study |
| <input type="checkbox"/> |  | KAN-36 | AI |

+ 만들기

프로젝트 QRex ...

요약 목록 보드 캘린더 타임라인 페이지 양식 코드 +

보드 검색  필터 그룹 ▾

TO-DO 10

- ChromaDB 알아보기(벡터 db) RAG KAN-48 
- qr 찍는 부분 사진이랑 사진 촬영한 거 어떻게 넣는지 알아보기 FRONTEND KAN-50 
- white list url dataset 찾기 RAG KAN-56 
- spring fastAPI 연동 

PROGRESS 3

- 2차 figma 디자인 수정, RAG 공부 STUDY 
- KAN-49 
- UI 1차 수정 및 리뷰와 수정사항 정리, RAG 공부한 내용 서로 공유 및 토의, IntelliJ 환경 설정 STUDY 
- KAN-52 
- spring 추가 2차 수정 BACKEND KAN-72 

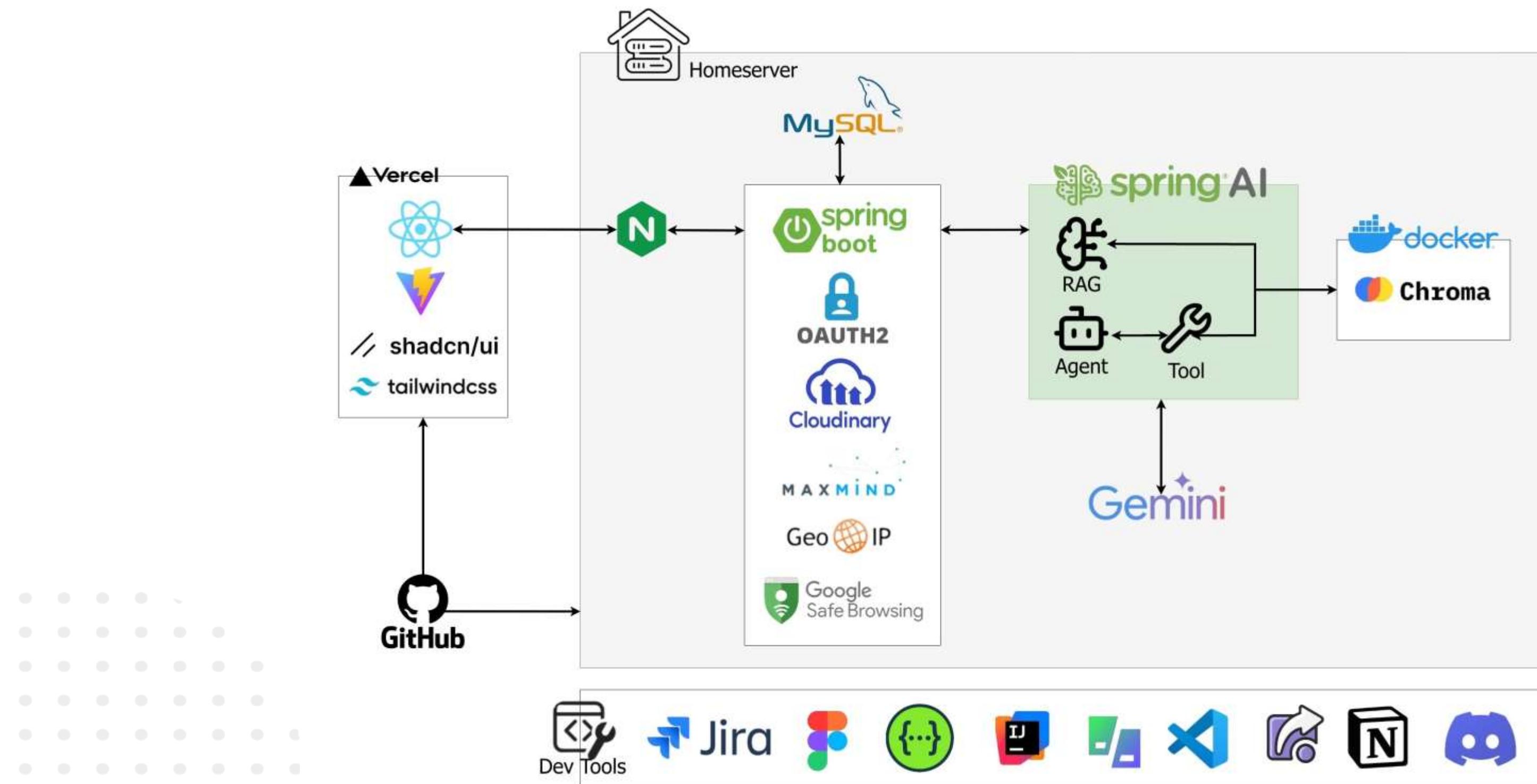
REVIEW 1

- IntelliJ IDEA 설치 예정, 피그마 UI 디자인 기간 설정(~09.21.(일)), UI 하위 항목 작성(최수연), UI 디자인 완료 후 DB 설계, RAG에 대해 간단히 공부, BACKEND 기능 정리 STUDY 
- KAN-10 

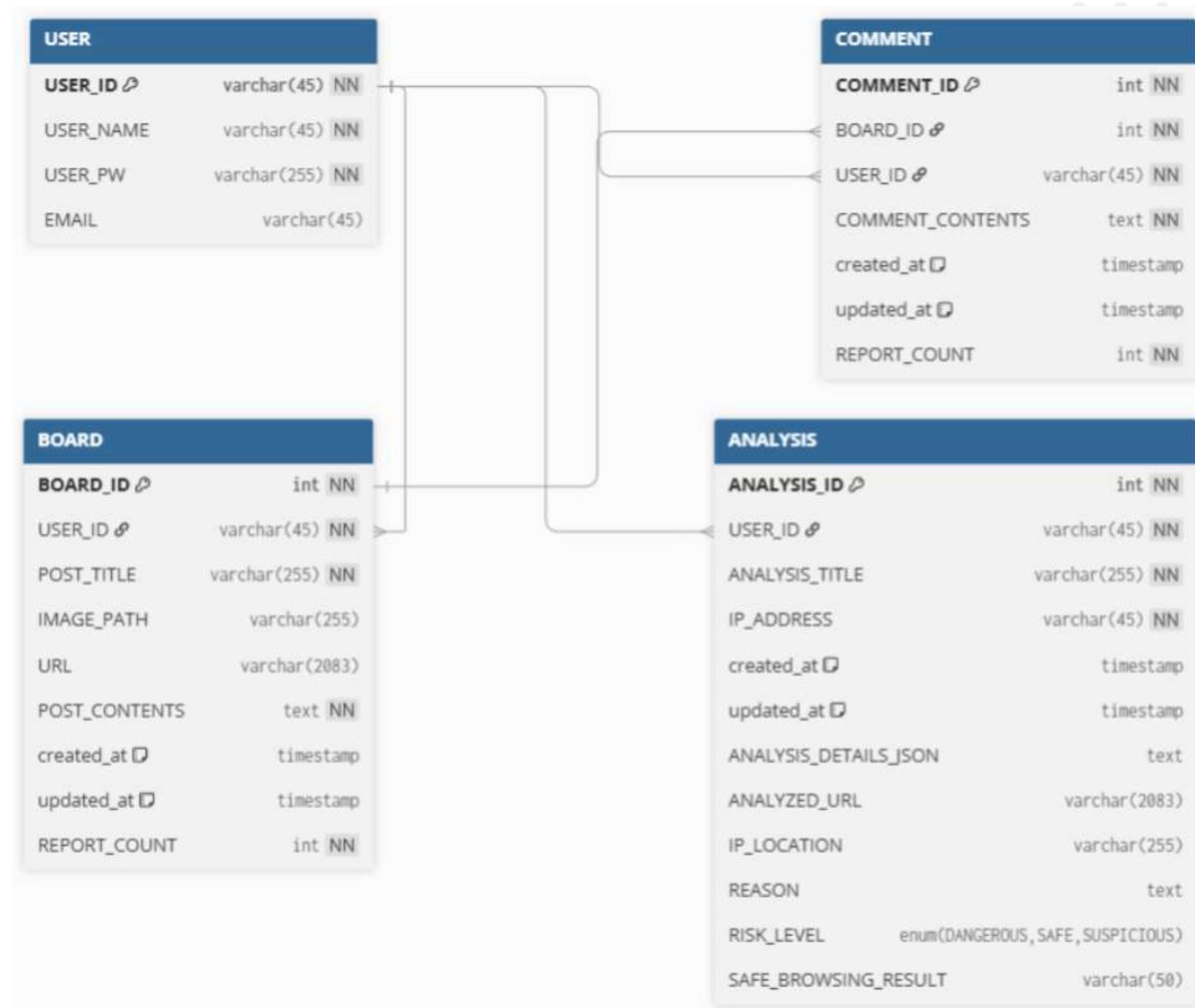
DONE 38 ✓

- github 링크 제출 STUDY 
- KAN-46 
- 3주차 과제 제출 STUDY 
- KAN-45 
- Github organization 생성 STUDY 
- KAN-38 
- dbdiagram 사용법 공부 

시스템 구성 및 아키텍처



DataBase





Analysis

Community

My post



15p

프로젝트 후기

- QR피싱이라는 사회적 문제를 기술적으로 해결하고자 하며 기획 · 개발 전 과정을 주도적으로 경험함
- 프론트, 백엔드, AI, DB, 서버까지 통합하면서 서비스 전체 아키텍처를 바라보는 시각과 협업 능력을 키움
- 직접 환경 구축과 문제 해결 과정을 통해 기술 선택과 소통 방식, 개발자로서의 성장 방향을 고민할 수 있었음

향후 발전 방향

- QR 간편결제 확산에 대응하기 위해 결제 과정에서의 QR 위변조 탐지 기술에 적용하여 범위 확대
- QR 이미지 분석·처리 기술을 통해 실물 위치 여부 판단과 보안 검증 기능을 고도화
- 통신사/금융기관 연동을 강화하여 의심 QR을 통한 결제·송금 시도를 실시간 차단하는 구조로 확장
- 스미싱·보이스피싱 등 다른 피싱 유형까지 대응하는 종합 모바일 보안 솔루션으로 발전 가능
- RAG 기반 데이터 확장을 지속하여 새로운 피싱 패턴에 대응하는 AI 분석 성능 향상
- RAG 기반 위험 분석을 수행한 뒤, AI 에이전트가 직접 웹페이지를 탐색하여 악성 동작을 검증하는 Active Analysis를 적용

2025 오픈소스 개발 프로젝트

THANK
YOU

2024042053 김여민,
2024042050 심연우,
2024042016 최수연