



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

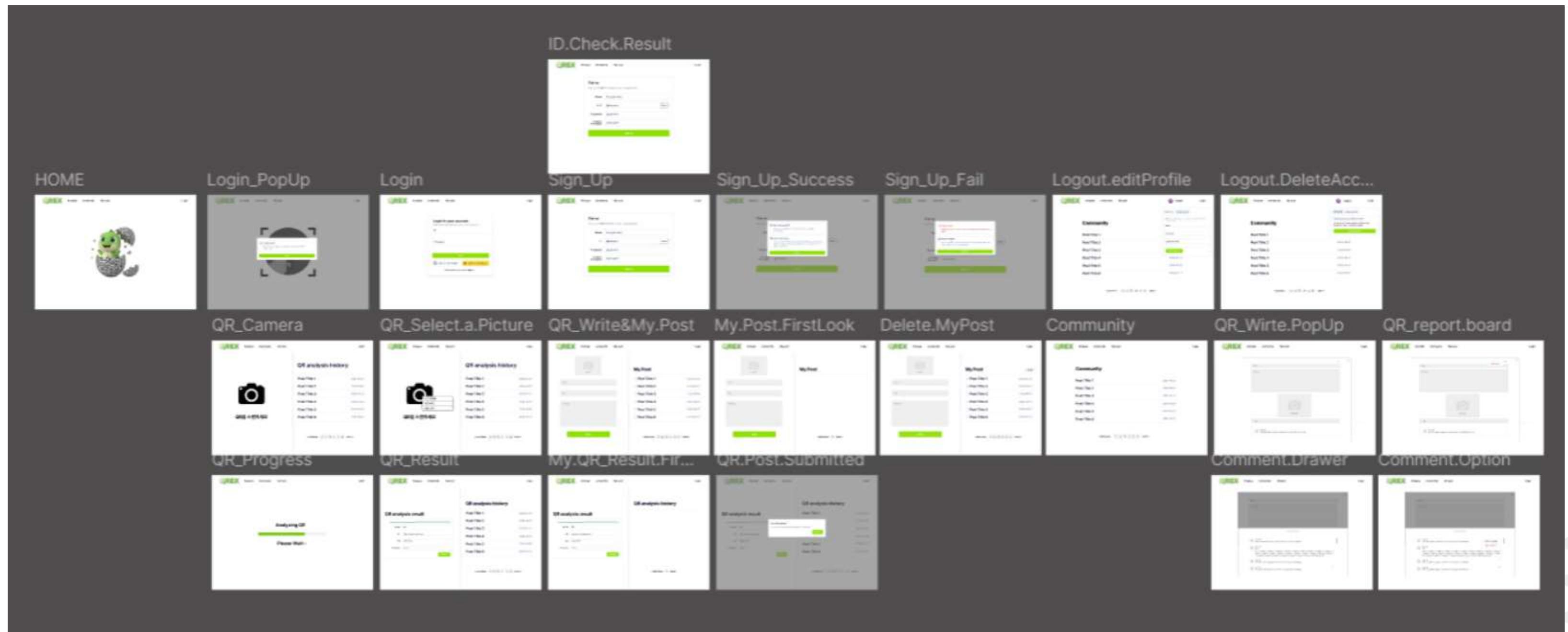
04 개인 맞춤형 기록 관리

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

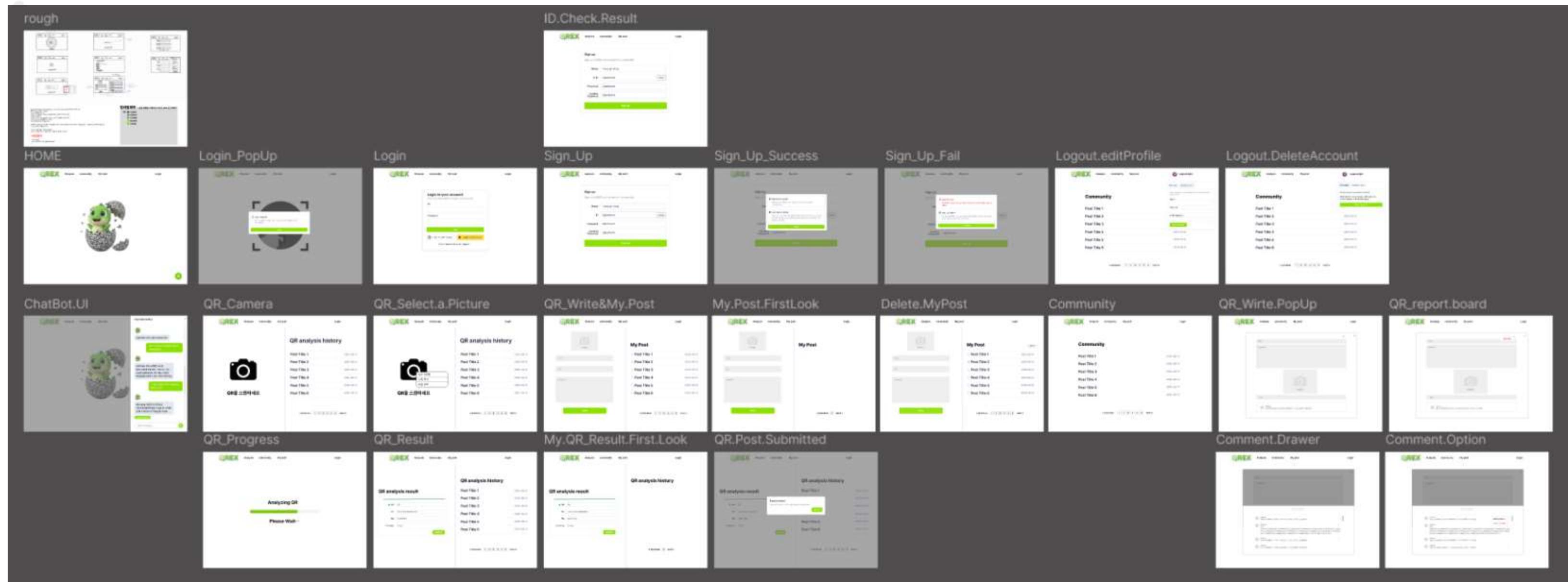


충북대학교 소프트웨어학과 4학년 배00

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



주변 피드백 수용 후



협업 방식 Notion



QRex Frontend Client

Key Features

- QR Code & URL Analysis (Code)
- Clean UI Rendering: 깨끗한 디자인과 사용자 친화적인 인터페이스를 제공합니다.
- Client-side Processing: 클라이언트에서 데이터를 처리하고 이를 서버로 전송하는 데 있어 최적화된 처리를 합니다.
- Visual Report: AI의 분석 결과(SAFE/DANGEROUS)를 기반으로 다양한 차트와 표로 결과를 시각화합니다.
- AI Security Agent
- Context Awareness: 사용자의 행동 패턴을 기반으로 개인화된 경험을 제공합니다.
- Screening UI: 사용자에게 위험한 링크나 파일을 차단하는 기능입니다.
- Authentication & Security
- JWT & API Interceptor
- Request: QR 코드를 통해 접속한 사용자를 인증합니다.
- Response: QR 코드를 통해 접속한 사용자를 인증합니다.
- OAuth 2.0: Google, Kakao, Naver 등의 계정을 지원합니다.

Tech Stack

- Frontend: React
- Build Tools: Vite (고속도, 편리한 설정)
- Styling: Tailwind CSS v1.5, Shadow UI (Dark UI 지원)
- Motion: Framer Motion (액션과 함께 흐름)
- Unit Testing: JEST, Mocha, Node.js

Project Structure



Frontend Development Progress

1. 초기 설계 및 세팅 (Initialization)

- Framework 선택: 써드파티 라이브러리 선택하고 React + Vite 투입
- Styling 선택: Tailwind CSS v1.5, Shadow UI (Dark UI 지원)
- Directory Structure: 위치별로 구조화하고, lib, components, pages, store, utils 등으로 분리

2. API 스크립트 최적화 (Optimization)

- API Endpoint: API를 적극 활용하여 데이터를 전송하는 방식으로 API를 활용해 성능을 향상
- [Challenge] API를 적극 활용하여 데이터를 전송하는 방식으로 API를 활용해 성능을 향상
- [Solution] API를 적극 활용하여 데이터를 전송하는 방식으로 API를 활용해 성능을 향상

3. API 호환성 테스트 (Compatibility)

- [Issue] API 호환성을 확보하기 위해 API를 활용해 데이터를 전송하는 방식으로 API를 활용해 성능을 향상
- [Challenge] API를 활용하여 데이터를 전송하는 방식으로 API를 활용해 성능을 향상
- [Solution] API를 활용하여 데이터를 전송하는 방식으로 API를 활용해 성능을 향상

4. 보안 및 인증 헤더 구현 (Security)

- API 헤더를 활용하여 보안을 강화합니다.
- [Challenge] API 헤더를 활용하여 보안을 강화합니다.
- [Solution] API 헤더를 활용하여 보안을 강화합니다.

5. 배포 및 배포 관리 (Deployment)

- Runtime Config
- Network Config
- [Challenge] API 헤더를 활용하여 보안을 강화합니다.
- [Solution] API 헤더를 활용하여 보안을 강화합니다.



백엔드 (Backend)

Deployment Architecture (배포 환경)

백엔드 서비스는 단일적인 서비스 제공을 위한 멀티 리스너 환경에서 구동되며, 형식 표준화를 통한 서비스 간의 통합을 목표로 합니다.

- Server Environment: Linux (Ubuntu/CentOS)
 - Domain Connection: Docker Container: 도메인과 연결되어 있으며, DNS 설정을 통해 호스팅을 제공합니다.
 - Process Management: Docker 을 사용하여 애플리케이션을 배포하고, 서비스를 슬로우 아웃 서비스로 충당하지 않도록 구현하였습니다.
- CORS Policy:** CORS 설정은 API 요청에 대한 접근 제한을 위해 CORS 설정을 적용하였습니다.

Directory Structure (상세 구조)

비즈니스 로직과 보안 관련 표시물을 명확하게 분리하여 심각했습니다.



Backend Development Progress

1. DB 설계 및 API 명세 (Architecture)

- DB 설계: MySQL, PostgreSQL, Oracle 등 다양한 DB를 지원합니다.
- API Docs: API 문서를 통해 API 명세와 동작 원리를 설명합니다.

2. 인증 시스템 고도화 (Auth System)

- OAuth 2.0: OAuth 2.0을 활용하여 사용자 인증을 지원합니다.
- JWT 토큰: JWT 토큰을 활용하여 사용자 인증 및 권한 관리를 합니다.

3. AI 서버와의 연동 설계 (Integration)

- API Endpoint: API를 활용하여 AI 서버와의 통신을 지원합니다.
- [Challenge] AI 서버와의 통신을 지원하는 API를 구현합니다.
- [Solution] AI 서버와의 통신을 지원하는 API를 구현합니다.

4. 외부 API 및 웹밀리터 통합

- Geolocation API: 위치 정보를 활용하여 서비스를 더 나은 방향으로 개선합니다.
- Safe Browsing API: 웹사이트의 안전성을 평가하는 API를 활용합니다.

5. 배포 (Deployment)

- Server: Linux 환경 (PC Server / Cloud)
- Domain: AWS Route 53: 도메인 구성을 관리합니다.
- Process: Docker 을 활용한 배포를 실행 및 보고 관리합니다.



RAG & Agent(Spring ai)

Core Algorithms

1. 베이지 데미네이트스 기반 RAG

- Embedding: 브론 카드, 마션 세레, 퍼마리스터스 대미네이트를 고지션 베이지로 비하하여 Chroma DB에 저장합니다.
- Semantic Search: “친구 미상호”라는 모호한 말로 “친구 미상호”라는 단어의 뜻을 찾습니다.
- Hybrid Analysis (Dual Path)
- Fact Path (AI-Based): 대로지 세션 히스토리/플랫폼으로 인증된 내 속성 판단.
- Deep Path (AI Reasoning): 단어 풍자, Gemini 2.5가 URL 구조, 디아프로세서, RAG 컨버스를 통한 주제 추정.
- AI Agent & Function Calling
- 단순 단면 정보를 넘어서 사용자의 행동 패턴을 파악하고 Java 코드를 직접 실행합니다.
- Example: “방금 친구님에게” → “제가 알았습니다.” → “방금 친구님에게”

Tech Stack

- Language: Java 21 LTS
- Frontend: Spring Boot 3.3.5 → Spring AI (1.0.1-M1)
- Database: Chroma DB (Local/Cloud)
- LLM Model: Google Gemini 1.5 툐킷 (Agent), 2.5 툐킷 (문서 분석)

Key API Logic

- 최상위 헤더 설정 (API 헤더)
- Rule Check - Vector Search (혹시 문제 있습니까?) → LLM Reasoning (혹시 문제 있습니까?)
- AI 예측 및 대화 (API 헤더)
- Insert Classification - Context Retrieval (Static RAG) - Function Calling (Action)



UI/UX Design

반응형 웹 디자인

UX Highlights

1. Landing Page & Animations

- Video Background: 고화질 멀티 배경과 Framed Buttons을 활용한 스그룹 매니페이션으로 보안 서비스의 신뢰감과 불안감을 제거합니다.
- Scroll Lock: 모바일 환경에서의 텍스트는 경험을 위해 브라우저 화면/혹은 브라우저 스크롤 헤더 헤더 (Header Lock)을 적용합니다.

2. Split View Dashboard (PC)

- Resizable Panel: PC 환경에서는 분석(Analysis)과 커뮤니티(Community) 헤더에서 좌우 패널 크기를 마우스로 자유롭게 조정할 수 있습니다. 이를 통해 사용자는 정보를 동시에 비교하여 생산성을 높일 수 있습니다.

3. Responsive Design

- Mobile Optimized: 모바일에서는 터치(Tap) 방식의 대로 자동 전환되어 습관화에서도 편리한 사용성을 제공합니다.

UI/UX Design Process

1. 컨셉 도출 및 아이디어 선정

- Keywords: 신뢰(Trust), 경고(Alert), 속도(Speed).
- Color Palette:
 - Main Deep Navy (#001E2E) : 보안과 신뢰감 향상.
 - Point Vibrant Red/Orange : 위험 요소에 대한 직관적 경고.
- Logo Design: QR 코드와 콘텐츠(T-Rex)를 통합하여 “강력한 팀” 이미지 시각화.

2. 웹사이트 프레임 및 레이아웃 구조화

- PC View: 정보 단색의 흐름상을 위해 Split View (좌: 백국 / 우: 상세) 차택.
- Mobile View: 출판 헤더에서의 사용상을 위해 Bottom Navigation & Tab 구조 차택.

3. 반응형 웹(RWD) 구현 전략

- Figma Auto Layout: 디자인 단계부터 반응형을 고려하여 모바일 미리보기 헤더 활용.
- Breakpoint 설정: Tailwind 기준 (675px), (lg:1024px) 를 기반으로 UI 구조가 드라마틱하게 변환되도록 설계 (행마다 대체로 48px → 96px).

4. 인터랙션 고도화

- Framer Motion: 단순화 헤더 디자인을 통해 AI 헤더를 지원하는 행성을 확장하고 사용자 경험 향상.
- 프론트엔드 “내가 도구를 사용할 수 있는 예상보다”하고 명시.
- 사용자 헤더 버튼에 AI 헤더를 활용한 헤더 헤더를 포함하는 헤더 헤더.
- Loading UI: 본래 대기 시간(약 3초) 동안 차운하지 않도록 스크립트는 UI와 진행률 바(Progress Bar) 디자인 적용.

5. 배포 및 배포 관리

• Environment

• Networking

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 

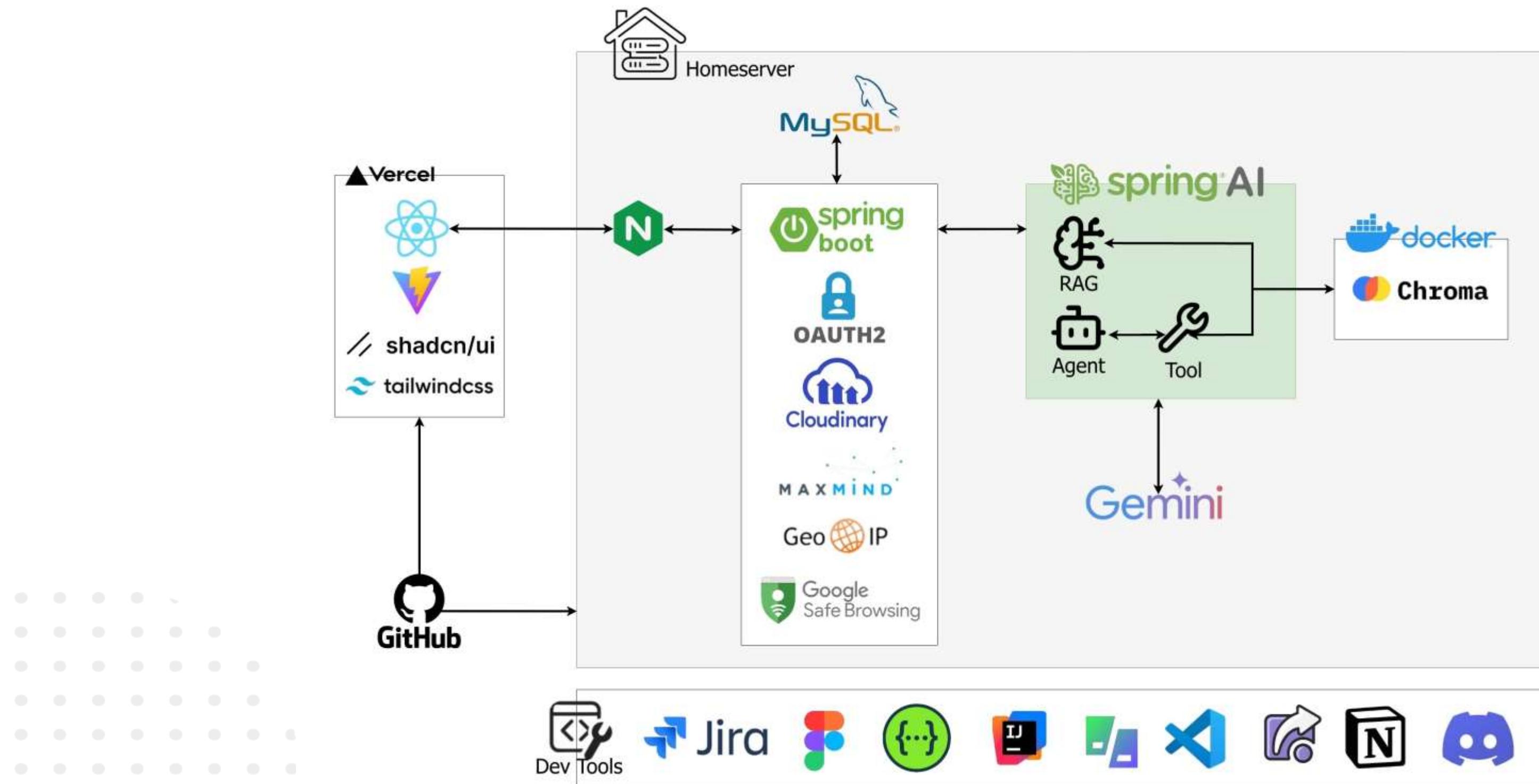
REVIEW 1

- 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 

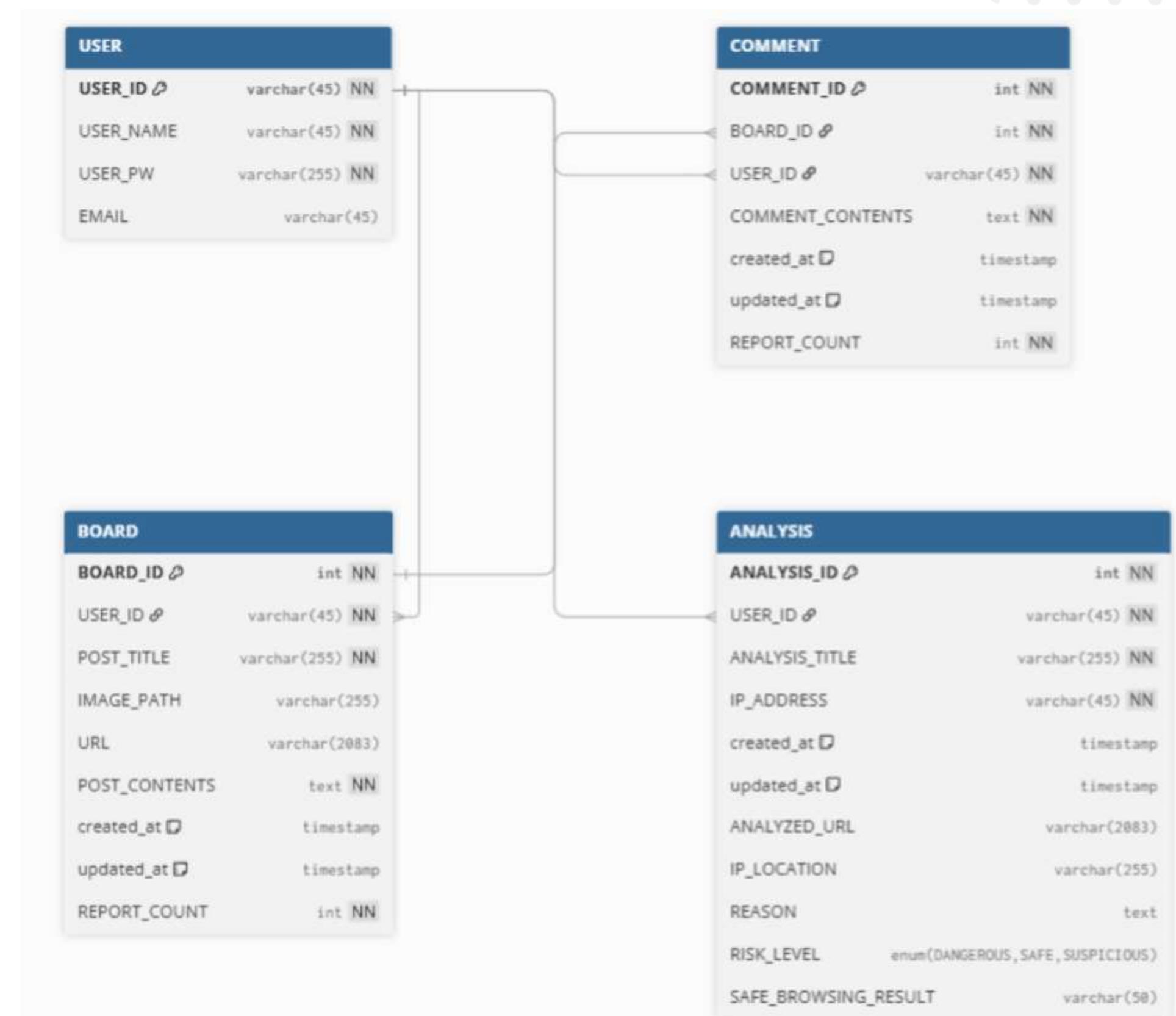
DONE 38 ✓

- 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