

인공지능을 활용한 물리적 보안 시스템

팀 명 : 하나~둘~찰칵!
지도 교수 : 이병천 교수님
팀 장 : 김혜정
팀 원 : 신수민
김건희
김승언
윤정목

2025. 10. 27

중부대학교 정보보호학과

목 차

1. 서론	
1.1. 연구 배경 및 필요성	4
1.2. 연구 목적	4
2. 관련연구	
2.1. 컴퓨터 비전(Computer Vision)	5
2.2. 인공지능(AI) 및 딥러닝 기술	5
2.3. 얼굴 인증 유사도 판단	5
2.4. 출입 통제 시스템 이론 및 동향	6
3. 본론	
3.1. 서비스 개요	7
3.1.1 주요 기능	7
3.1.2 시나리오	12
3.2. 개발 환경	12
3.2.1 운영체제	12
3.2.2 IDE 및 텍스트 편집기	13
3.3. 사용 기술	13
3.3.1 언어	13
3.3.2 라이브러리	14
3.3.3 데이터베이스	15
3.3.4 외부 서비스	16
3.3.5 협업 도구	17
4. 결론	
4.1. 프로젝트 결과	18
4.2. 기대 효과	18
5. 참고 문헌	19
6. 별첨	
6.1. 소개 자료	20
6.2. 프로젝트 소개	21
6.2.1 시연 영상	21
6.2.2 웹 서비스 주소	21
6.3. 팀 소개	22
6.4. 발표 자료	23

그 림 목 차

<그림 1> 서비스 동작 구조	7
<그림 2> 등록 / 조회	7
<그림 3> AI 모델 클라우드 저장소	8
<그림 4> AI 연동 코드	8
<그림 5> 중앙 관리자 페이지	9
<그림 6> 관리자 페이지 코드	9
<그림 7> neonDB 서버	10
<그림 8> QR 생성 및 sms 전송 요청 페이지	11
<그림 9> sms 전송	11
<그림 10> QR / sms 서버 코드	11

1. 서론

1.1. 연구 배경

현대 사회에서 보안은 개인의 프라이버시와 자산을 보호하기 위한 핵심 요소로, 특히 사무실, 연구소, 주거 공간 등과 같은 폐쇄적 공간에서는 출입 관리의 중요성이 높아지고 있다. 기존의 비밀번호 기반 출입 통제 시스템은 간편하다는 장점이 있지만, 비밀번호의 유출·도용 가능성이 높고 사용자가 직접 입력해야 한다는 번거로움이 존재한다.

또한, 비밀번호나 카드 키를 분실하거나 공유할 경우, 제3자가 쉽게 접근할 수 있어 보안상 취약점을 초래한다. 이처럼 편의성과 보안성 간의 균형을 유지하기 어려운 구조는 전통적인 출입 통제 시스템의 한계로 지적되어 왔다.

‘펜타시큐리티시스템이 발표한 인증보안 솔루션 리포트(2025)’에 따르면, 최근 몇 년간 비밀번호 유출 및 계정 도용으로 인한 침입 사고가 급증하고 있으며, 기존 비밀번호 기반 인증 체계는 이미 구조적 한계에 도달했다고 지적하였다. 이에 따라 업계 전반에서는 ‘패스워드리스>Password-less’와 생체 인식 기반 인증 기술을 차세대 표준으로 전환하려는 흐름이 가속화되고 있다.

한편, 인공지능(AI)과 컴퓨터 비전 기술의 발전으로 사람의 얼굴, 음성, 홍채 등 고유한 생체 정보를 기반으로 한 비접촉식 인증 기술이 빠르게 확산되고 있다. 특히 얼굴 인식 기술은 비접촉 방식으로 빠르고 정확한 인증이 가능해, 감염 예방과 사용자 편의성을 동시에 확보할 수 있다는 점에서 최근 다양한 산업 분야에 적용되고 있다. 이러한 기술적 흐름 속에서, 기존 출입 방식의 한계를 극복하고 AI 기반의 안전하고 효율적인 얼굴 인식 출입 통제 시스템을 개발하려는 필요성이 대두되고 있다.

1.2 연구 목적

본 연구의 목적은 인공지능 기술을 활용하여 사용자의 얼굴을 인식함으로써 출입을 통제하는 지능형 얼굴 인식 출입 통제 시스템을 개발하는 것이다. 이를 통해 비밀번호나 출입 카드에 의존하지 않고, 사용자의 생체 정보를 기반으로 한 고신뢰도 인증 방식을 구현하고자 한다.

또한, 딥러닝 기반 얼굴 인식 알고리즘을 적용하여 다양한 환경(조명 변화, 각도, 표정, 마스크 착용 등)에서도 안정적으로 동작할 수 있는 시스템을 구축하는 것을 목표로 한다. 더불어, 출입 기록을 자동으로 저장하고 관리자가 중앙에서 실시간으로 모니터링할 수 있는 기능을 포함함으로써, 보안성과 관리 효율성을 동시에 향상시키는 것을 지향한다.

궁극적으로 본 연구는 AI 기술을 적용한 스마트 보안 시스템 모델을 제시함으로써, 향후 주거·사무 공간뿐만 아니라 공공기관 및 산업 현장에서도 적용 가능한 차세대 출입 통제 시스템의 기초 연구로서의 의의를 가진다.

2. 관련 연구

2.1. 컴퓨터 비전(Computer Vision)

컴퓨터 비전은 인간의 시각적 인지 능력을 컴퓨터가 모방하도록 하는 인공지능의 한 분야로, 이미지나 영상을 통해 물체를 인식하고 이해하는 기술을 말한다. 초기에는 단순한 패턴 인식과 경계 검출 알고리즘에 기반하였으나, 최근에는 딥러닝(Deep Learning) 기술의 발전으로 인해 객체 탐지(Object Detection), 얼굴 인식(Face Recognition), 행동 분석(Action Recognition) 등 다양한 응용 분야로 확장되고 있다.

특히 합성곱 신경망(CNN, Convolutional Neural Network)의 도입은 이미지에서 특징을 자동으로 추출하고 학습할 수 있게 함으로써, 기존 방식보다 높은 정확도와 효율성을 보장한다. 이러한 컴퓨터 비전 기술은 얼굴 인식 출입 통제 시스템의 핵심 요소로, 사용자의 얼굴을 정확히 식별하고 실시간으로 인증하는 데 중요한 역할을 수행한다.

2.2. 인공지능(AI) 및 딥러닝 기술

인공지능은 인간의 사고와 학습 과정을 모방하여 스스로 판단하거나 문제를 해결할 수 있는 기술을 의미한다. 그중에서도 머신러닝(Machine Learning)과 딥러닝(Deep Learning)은 대규모 데이터를 기반으로 스스로 규칙을 학습하고, 이를 토대로 예측이나 분류를 수행한다.

얼굴 인식 분야에서는 주로 CNN(Convolutional Neural Network) 구조를 기반으로 한 모델들이 사용되며, 대표적인 알고리즘으로는 FaceNet, VGGFace, ResNet, ArcFace 등이 있다. 이러한 모델들은 얼굴 이미지로부터 고유한 임베딩 벡터를 추출하고, 두 얼굴 간의 유사도를 계산함으로써 동일 인물 여부를 판단한다.

또한, 최근에는 경량화된 AI 모델을 적용하여 라즈베리파이(Raspberry Pi), NVIDIA Jetson, 엣지 디바이스(Edge Device) 등에서도 실시간 얼굴 인식이 가능하도록 구현하는 연구가 활발히 이루어지고 있다. 이는 출입 통제 시스템과 같은 IoT 기반 환경에 매우 적합하다.

2.3. 얼굴 인증 유사도 판단

얼굴 인식 시스템에서는 단순히 이미지를 비교하는 것이 아니라, 각 얼굴의 특징을 고차원 공간 상의 벡터로 표현한 후 이들 간의 유사도(Similarity)를 측정하여 동일 인물 여부를 결정한다. 일반적으로 사용되는 주요 유사도 계산 방식에는 유클리드 거리(Euclidean Distance), 맨해튼 거리(Manhattan Distance), 그리고 코사인 유사도(Cosine Similarity)가 있다.

- 1 - 유클리드 거리(Euclidean Distance) 는 두 벡터 간의 직선 거리를 계산하여, 거리가 짧을수록 유사도가 높다고 판단한다.
- 2 - 맨해튼 거리(Manhattan Distance) 는 축을 따라 이동하는 거리의 합을 계산하는 방식으로, 노이즈에 강하다는 장점이 있다.
- 3 - 코사인 유사도(Cosine Similarity) 는 벡터의 크기보다는 방향(각도)에 초점을 맞추어 두 벡터가 얼마나 같은 방향을 가리키는지를 측정한다.

본 연구에서는 코사인 유사도(Cosine Similarity) 를 이용하여 얼굴 간 유사도를 계산하였다. 이는 벡터의 크기보다 특징 벡터의 패턴(방향성) 을 더 중요하게 평가하기 때문에, 조명 변화나 거리 차이 등 외부 요인에 의한 얼굴 크기 변화에도 비교적 강인한 성능을 보인다. 즉, 두 얼굴 임베딩 벡터의 코사인 유사도 값이 1에 가까울수록 동일 인물일 가능성이 높다고 판단할 수 있다.

2.4. 출입 통제 시스템 이론 및 동향

출입 통제 시스템(Access Control System)은 특정 공간이나 자원에 대한 접근을 허가된 사용자에게만 부여하기 위한 보안 기술이다. 초기의 출입 통제 방식은 기계적 키나 비밀번호 입력형 키패드를 사용하는 단순한 형태였으나, 이러한 방법은 분실·유출·복제의 위험이 크고 사용자 편의성이 떨어진다는 문제가 있었다.

이에 따라, RFID 카드, 지문 인식, 홍채 인식, 얼굴 인식 등 생체인식 기반(비밀번호 비의존적) 접근 제어 기술이 발전하게 되었다. 특히, 코로나19 팬데믹 이후 비접촉식 인증 기술에 대한 수요가 급증하면서 얼굴 인식 시스템이 각광받고 있다.

최근 연구에서는 단순히 얼굴을 인식하는 수준을 넘어, 인공지능을 활용하여 사용자의 표정 변화, 조명, 마스크 착용 여부에도 강인한 인식률을 보장하는 기술 개발이 이루어지고 있다. 또한, 클라우드 서버나 엣지 컴퓨팅과 연계하여 출입 기록을 자동 저장하고, 관리자에게 실시간으로 통계 데이터를 제공하는 지능형 출입 통제 시스템으로 발전하고 있다.

3. 본론

3.1. 서비스 개요

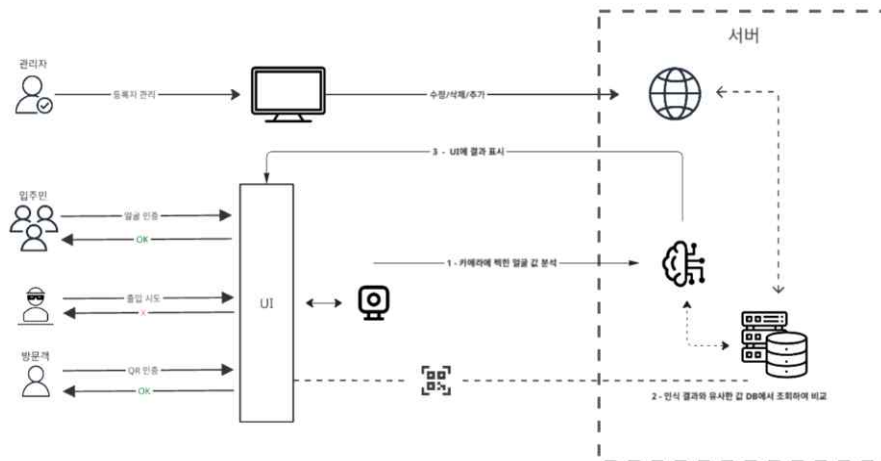


그림 1 서비스 동작 구조

본 시스템은 얼굴 인식 기반 출입 통제를 통해 거주지 및 사무 공간의 보안을 강화하고, 사용자 편의성을 제공하는 것을 목표로 한다. 기존 비밀번호나 카드 키 기반 출입 방식의 한계를 극복하고, 비접촉 인증 방식을 통해 안전하고 효율적인 출입 환경을 제공한다. 다음은 본 연구에서 구현한 서비스에 대한 간략한 설명이다.

3.1.1 주요 기능

```
[INFO] 다운로드 시작 : 등록 모델
[DOWNLOAD] 진행률 : 100%
[INFO] 다운로드 완료 : 등록 모델 (크기: 170296 KB)
[INFO] 다운로드 시작 : 탐지 모델
[DOWNLOAD] 진행률 : 100%
[INFO] 다운로드 완료 : 탐지 모델 (크기: 16527 KB)
[INFO] 모델 메모리에 로드 중...
[INFO] 모델 로드 완료 : 등록 모델
[INFO] 모델 로드 완료 : 탐지 모델
[INFO] 모든 모델 로드 완료!
[INFO] R2 업로드 성공 (SDK): https://pub-8060-1172-010765-010211-01011000.r2.dev/images/faces/user_170070440.jpg
[ERROR] embedding 값에 NaN 포함, 등록 실패
[INFO] R2 업로드 성공 (SDK): https://pub-8060-1172-010765-010211-01011000.r2.dev/images/faces/user_170070440.jpg
[INFO] ? 등록 완료: user_170070440 -> https://pub-8060-1172-010765-010211-01011000.r2.dev/images/faces/user_170070440.jpg
[INFO] 상태: 사용자 등록 완료
[INFO] R2 객체 목록 조회 성공.
[INFO] 유사도 최고: 0.93541 -> ? 인증 성공 (ID: user_170070440)
```

그림 2 등록 / 조회

- 얼굴 등록 및 조회: 사용자는 건물 외부에 있는 카메라를 통해 자신의 얼굴을 등록하고, 최초 등록 후, 출입 시 실시간 얼굴 인증을 통해 등록 여부를 검증하여 조회가 될 시 건물 내로 출입할 수 있게 된다.

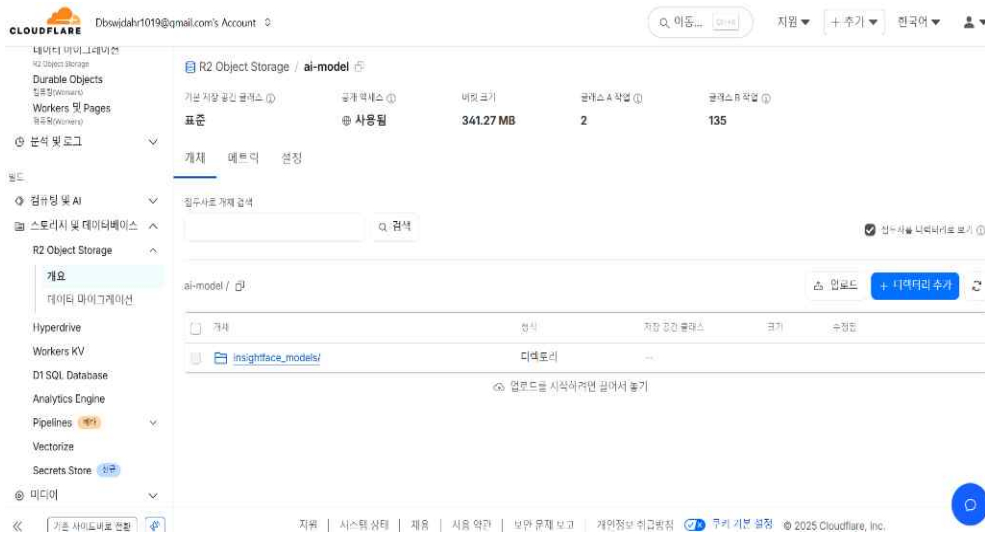


그림 3 AI 모델 클라우드 저장소

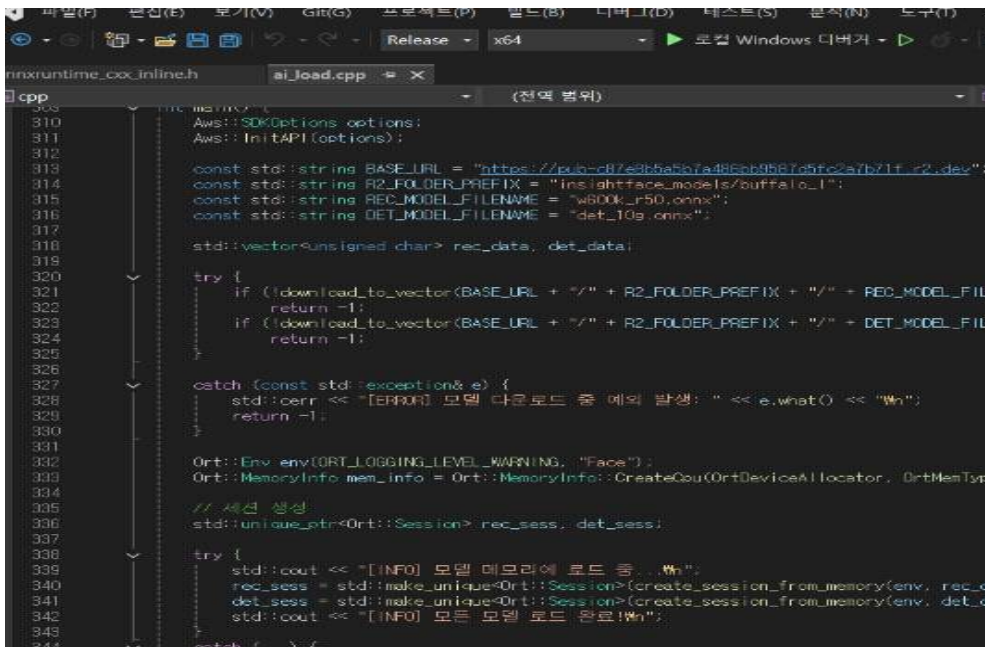


그림 4 AI 연동 코드

등록 과정에서 사용자의 얼굴 이미지 데이터를 서버에 저장하게 된다. 이후 카메라에서 출입 인증 요청을 보내면 DB의 얼굴 목록과 코사인 유사도를 기준으로 유사한 값이 있는지 비교하게 된다.

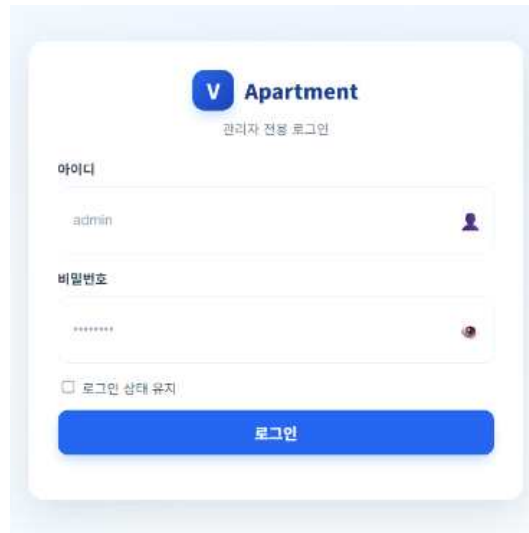


그림 5 중앙 관리자 페이지

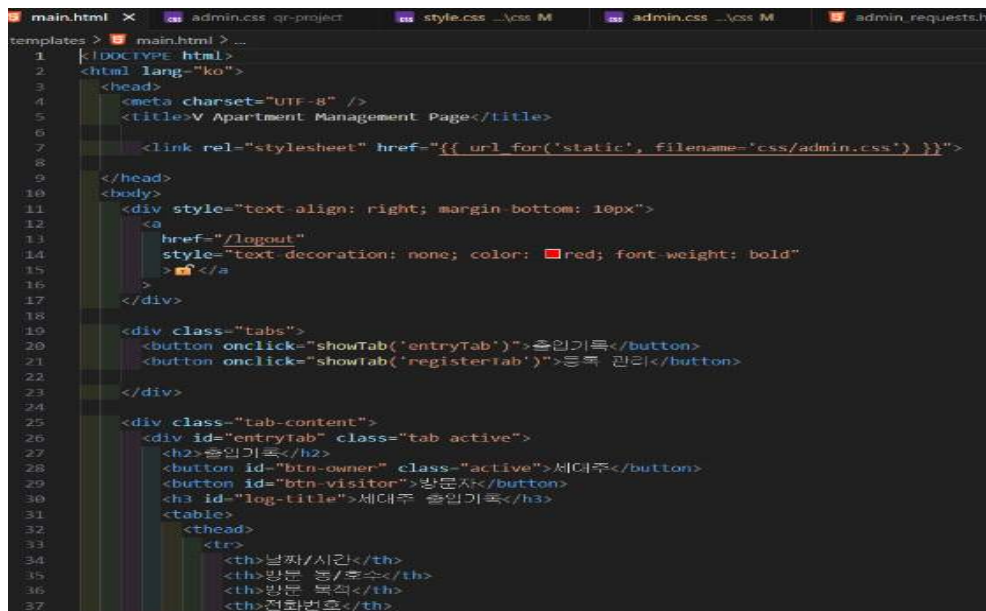


그림 6 관리자 페이지 코드

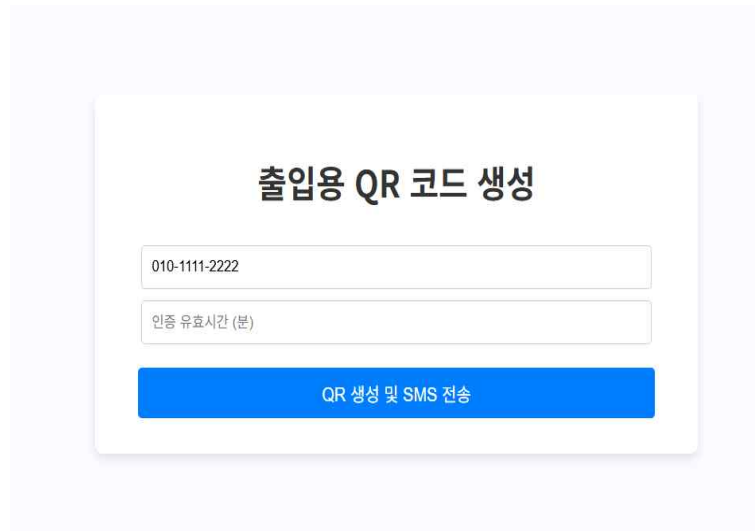
- 관리자 기능: 시스템 관리자는 중앙 서버에 등록된 사용자(거주자)들의 정보를 관리하고 필요 시 등록된 데이터를 수동으로 삭제하거나 수정하여 관리할 수 있다.

Access Control API (Full Cloud Native) 6.1.0 ONS 3.1
/openapi.json

default		^
GET	/healthz Healthz	▼
GET	/registrations 모든 등록 이벤트 목록 조회	▼
POST	/registrations Registrations	▼
GET	/access-events 모든 출입 이벤트 목록 조회	▼
POST	/access-events Access Events	▼
GET	/qr-events 모든 QR 이벤트 목록 조회	▼
POST	/qr-events Qr Events	▼
PATCH	/registrations/{registration_id}/approve 등록 요청 승인	▼

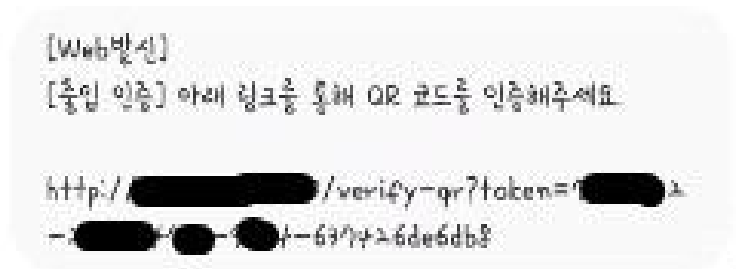
그림 7 neonDB 서버

- 출입 기록 관리: 건물 출입 시도 시, 출입 시도 기록과 인증 결과를 자동으로 저장하여 관리자 화면에서 실시간으로 확인할 수 있다. 이후, 범죄나 기타 특이사항이 발생했을 경우, 추적이 가능하도록 한다.



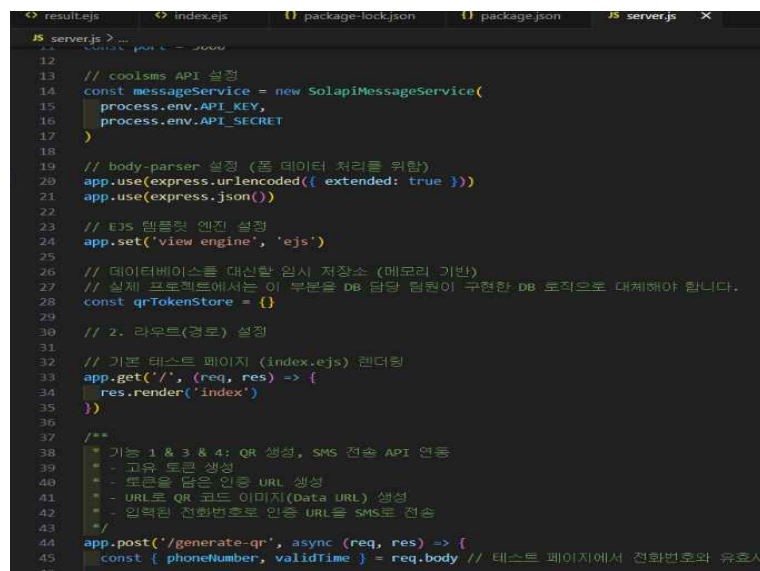
The image shows a web form titled "출입용 QR 코드 생성" (QR Code Generation for Entry). It contains two input fields: the first is for a phone number, with "010-1111-2222" entered; the second is for an expiration time in minutes, labeled "인증 유효시간 (분)". Below these fields is a blue button labeled "QR 생성 및 SMS 전송" (Generate QR and Send SMS).

그림 8 qr 생성 및 sms 전송 요청 페이지



The image shows a screenshot of an SMS message. The text reads: "[Web발신] [출입 인증] 아래 링크를 통해 QR 코드를 인증해주세요" (SMS from Web [Entry Authentication] Please authenticate the QR code through the link below). Below the text is a URL: "http://[redacted]/verify-qr?token=[redacted]-[redacted]-[redacted]-637926de6db8".

그림 9 sms 전송



The image shows a snippet of JavaScript code for an Express.js server. It includes comments in Korean and code for setting up the server, including API keys, body parsing, and a POST endpoint for generating QR codes and sending SMS. The code is as follows:

```

12
13 // coolsms API 설정
14 const messageService = new SolapiMessageService(
15   process.env.API_KEY,
16   process.env.API_SECRET
17 )
18
19 // body-parser 설정 (폼 데이터 처리를 위한)
20 app.use(express.urlencoded({ extended: true }))
21 app.use(express.json())
22
23 // EJS 템플릿 엔진 설정
24 app.set('view engine', 'ejs')
25
26 // 데이터베이스를 대신할 임시 저장소 (메모리 기반)
27 // 실제 프로젝트에서는 이 부분을 DB 담당 팀원이 구현한 DB 로직으로 대체해야 합니다.
28 const qrTokenStore = {}
29
30 // 2. 라우트(경로) 설정
31
32 // 기본 테스트 페이지 (index.ejs) 렌더링
33 app.get('/', (req, res) => {
34   res.render('index')
35 })
36
37 /**
38  * 기능 1 & 3 & 4: QR 생성, SMS 전송 API 연동
39  * - 고유 토큰 생성
40  * - 토큰을 담은 인증 URL 생성
41  * - URL로 QR 코드 이미지(Data URL) 생성
42  * - 입력된 전화번호로 인증 URL을 SMS로 전송
43  */
44 app.post('/generate-qr', async (req, res) => {
45   const { phoneNumber, validTime } = req.body // 테스트 페이지에서 전화번호와 유효시간
46

```

그림 10 QR / sms 서버 코드

- 예외 상황 처리: 배달, 지인 방문 등 얼굴 인증을 통한 접근이 어려운 경우에는 QR 코드 기반 임시 발급 및 인증 기능을 통해 출입을 허용한다.

3.1.2. 시나리오

- 거주자 출입 흐름

- 거주자가 건물 현관에 접근하면, 설치된 카메라가 자동으로 얼굴을 감지한다.

시스템은 등록된 얼굴 데이터와 실시간 촬영 이미지를 비교하고, 인증이 완료되면 출입문이 자동으로 개방된다.

- 인증 과정에서 사용자는 별도의 입력 없이 자연스럽게 출입할 수 있어 편의성을 높인다.

- 방문객 출입 흐름

- 등록되지 않은 사용자가 출입을 시도할 경우, 시스템은 QR코드 기반 임시 인증 절차를 안내한다.

- 방문객은 사전에 발급받은 QR코드를 스캔하거나 단말기에 제시하여 출입이 가능하며, 모든 기록은 자동으로 저장된다.

- 예외 상황 대응

- 얼굴 인식이 실패하거나 마스크 착용 등으로 인증이 어려운 경우, 관리자 승인 또는 QR코드 인증으로 임시 출입이 가능하다.

- 이러한 예외 상황은 모두 기록으로 남아 추후 출입 내역 검토 및 보안 관리에 활용된다.

- 관리자 모니터링

- 관리자는 중앙 서버를 통해 출입 시도와 인증 결과를 실시간으로 확인하고, 이상 패턴이 감지되면 즉시 대응할 수 있다.

- 이를 통해 시스템의 안전성과 신뢰성을 확보하며, 사용자의 출입 편의성과 보안을 동시에 유지한다.

3.2. 개발 환경

본 프로젝트의 개발 환경은 로컬 Windows 11 환경을 기반으로 구성되었으며, 다양한 개발 도구와 언어를 활용하여 효율적인 개발을 수행할 수 있도록 설계하였다. 프로젝트의 특성과 역할에 따라 개발 환경을 다음과 같이 세부적으로 분류할 수 있다.

3.2.1 운영체제

프로젝트 개발과 테스트는 Windows 11 환경에서 진행되었다. Windows는 다양한 개발 도구와 IDE를 안정적으로 지원하며, 로컬에서의 테스트와 디버깅에 유리하다. 본 프로젝트에서는 C++ 기반 모듈 개발이 포함되어 있어, Windows에서 기본적으로 제공하는 MSVC(Microsoft Visual C++) 컴파일러, Visual Studio 디버깅 및 프로파일링 도구, 풍부한 라이브러리 지원을 활용할 수 있는 점이 이점이기때문에 windows 11를 채택하였다.

3.2.2 개발 도구 (IDE 및 텍스트 편집기)

효율적인 코드 작성과 디버깅을 위해 세 가지 주요 개발 도구를 활용하였다.

Visual Studio 2022

Microsoft에서 개발한 IDE로 주로 C++ 기반 모듈과 네이티브 애플리케이션 개발에 사용되었다. Windows 환경에서 최적화된 MSVC 컴파일러와 강력한 디버깅 및 프로파일링 도구를 제공한다.

대규모 프로젝트 관리와 다양한 OpenCV, QT 등 외부 라이브러리 연동이 용이하여 C++ 모듈 개발에 유리하고 메모리와 스레드 등의 디버깅이 가능하여 상세한 분석으로 최적화된 개발이 가능하다.

PyCharm

JetBrains에서 제작한 IDE, Python 기반 데이터 처리와 다차원 배열 연산, API를 통한 외부 저장소에서의 AI 모델 연동, 테스트 스크립트 작성 등에 활용하였다. Python 개발 시 가장 완성도가 높고 강력한 기능을 제공한다.

Python 특화 IDE로 전문적인 지원 기능과 코드 완성, 디버깅, 가상환경 관리, 패키지 설치 자동화 기능을 제공하여 Python 개발 효율성과 편리성을 극대화하였다.

Visual Studio Code (VS Code)

Microsoft에서 개발한 텍스트 편집기. 풍부한 플러그인과 확장 기능을 통해 다양한 언어와 환경을 통합적으로 관리할 수 있으며, 사용하기 쉽다는 장점을 가지고 있어 CSS, Python, DB 및 웹 관련 코드 작성 등 여러 방면으로 활용되었다.

프로젝트 전반을 전체적으로 관리하고, IDE 대비 초기설정이 간편한 경량 편집기로 빠른 테스트와 수정이 가능하고 크로스 플랫폼을 지원하여 가장 유연한 개발 환경을 제공한다.

3.3. 사용 기술

3.3.1. 언어

C++

C++은 C 언어를 기반으로 객체지향 프로그래밍(OOP) 기능을 추가한 범용 프로그래밍 언어로, 고성능 애플리케이션 개발에 적합하다. 메모리 직접 제어, 스레드와 병렬 처리 지원 등 시스템 수준의 제어가 가능하며, 이로 인해 게임, 시스템, 임베디드 등 분야에 자주 활용된다. 오랜 역사만큼 방대한 커뮤니티와 라이브러리를 보유하고 있고 OpenCV, Qt 등의 라이브러리가 C++로 작성되어 연동이 용이한 점을 고려하여 채택하였다.

본 프로젝트에서는 C++를 활용하여 메모리 효율성을 고려한 고성능 모듈 개발과 네이티브 애플리케이션 구현을 수행하였다. 특히, 로컬에서 Windows 기반 환경에서 진행되는 개발의 최적화와 수월한 디버깅과 빌드를 위해 C++를 선택함으로써 프로젝트의 효율성과 안정성을 높이하고자 하였다.

Python

Python은 간결하고 가독성이 높은 문법을 특징으로 하는 범용 스크립트 언어로, 프로그래밍 입문자가 가장 쉽게 접할 수 있는 언어이다. 데이터 처리, AI/머신러닝, 웹 개발 등 다양한 분야에서 폭넓게 활용되며, 풍부한 라이브러리 생태계를 갖추고 있다. Numpy, Pandas, Scikit-learn 등을 통해 복잡한 연산과 데이터 처리를 효율적으로 구현할 수 있으며, 최근 빅데이터와 AI에 대한 수요가 계속해서 증가하고 있어 Python을 통한 활발한 연구가 진행 중이다.

프로젝트에선 Python으로 작성된 코드를 통해 AI 모델을 연동, 데이터 처리, 테스트 스크립트 작성 등 다양한 작업에 수행하였다. 또한, Python의 가상환경과 패키지 관리 기능을 활용하여 프로젝트 종속성 관리와 개발 편의성을 확보하였다.

Javascript

JavaScript는 웹 환경에서 동작하는 대표적인 스크립트 언어로, 동적 웹 페이지 제작과 사용자 인터페이스(UI) 구현에 널리 사용된다. 비동기 처리(Asynchronous)와 이벤트 기반 구조를 통해 빠른 응답성과 실시간 상호작용을 제공하며, Node.js 환경에서는 서버 사이드 개발에도 활용이 가능하다.

관리자 페이지 구현과 QR코드 기반 출입 인증 기능을 위해 JavaScript를 사용하였다. 특히 Node.js 서버를 통해 QR 생성 및 인증 요청 처리를 수행하였으며, 이를 통해 외부 방문자(배달, 지인 등)에 대한 임시 출입 관리 기능을 구현하였다.

3.3.2. 라이브러리

OpenCV

OpenCV는 컴퓨터 비전 및 이미지 처리 분야에서 널리 사용되는 오픈소스 라이브러리이다. 이미지와 영상 데이터를 처리, 분석, 변환하는 다양한 함수와 알고리즘을 제공하기 때문에 AI 모델과 연동한 연산이 가능하며, 실시간 이미지 및 영상 처리에도 최적화되어 있다.

얼굴 인식 및 이미지 전처리 기능을 구현하기 위해 OpenCV를 활용하였다. 특히, C++ 기반 모듈과 연동하여 고성능 영상 처리와 효율적인 메모리 관리를 가능하게 하였다.

Qt

Qt는 C++ 기반의 크로스 플랫폼 GUI 프레임워크로, 윈도우, 리눅스, macOS 등 다양

한 운영체제에서 GUI 애플리케이션을 개발할 수 있도록 지원한다.

스크린을 통해 보이는 사용자 인터페이스의 구성과 네이티브 애플리케이션 개발을 위해 Qt를 사용하였으며, C++ 모듈과의 호환성으로 효율적인 UI 개발과 이벤트 처리 구현이 가능하다.

ONNX Runtime

ONNX Runtime은 다양한 머신러닝 프레임워크에서 학습된 모델을 효율적으로 실행할 수 있도록 지원하는 런타임 라이브러리이다. 용량이 거대하여 부담스러운 AI 모델을 직접 저장하지 않고 메모리로 불러와 실행 시 사용이 가능하게 해준다.

진행 과정 중, Python과 C++ 환경에서 AI 모델을 연동하고 추론 속도를 최적화하기 위해 ONNX Runtime을 활용하였다. 이를 통해 Python을 통해 로컬 환경에서 간단한 테스트를 진행하고 C++로 빠른 실행과 연산을 수행해 다양한 플랫폼 호환성을 확보할 수 있었다.

FastAPI

FastAPI는 Python 기반의 비동기 웹 프레임워크로, 빠른 실행 속도와 간결한 코드 구조를 강점으로 한다. Starlette과 Pydantic을 기반으로 하여 고성능 비동기 처리를 지원하며, 자동 문서화(Swagger UI, ReDoc) 기능을 통해 개발 및 테스트 효율을 극대화할 수 있다.

관리자 단에서 데이터베이스와 대시보드 UI 간 연동을 위해 FastAPI를 사용하였다. 특히, HTML 템플릿 렌더링과 NeonDB와의 연동이 용이하여 관리자 페이지 구현 및 QR 생성 시 Node.js 서버와의 통신 구조를 효율적으로 구성할 수 있다. 또한, FastAPI의 비동기 요청 처리 특성 덕분에 사용자 요청에 대한 빠른 응답과 안정적인 서버 성능을 확보가 가능하다.

3.3.3. 데이터베이스

MySQL

MySQL은 널리 사용되는 오픈소스 관계형 데이터베이스로, 안정성과 성능이 검증된 시스템이다. 테이블 기반 구조와 SQL 표준을 준수하여 데이터 관리와 조회가 용이하며, 다양한 운영체제와 개발 환경에서 폭넓게 지원된다. 또한, 방대한 커뮤니티와 문서, 라이브러리를 갖추고 있어 유지보수와 확장에 유리하다. MySQL이 오라클에 인수된 이후, MySQL이 상용화되어 MariaDB로의 전환이 늘어나는 추세이다.

본 프로젝트에서는 전통적인 MySQL을 사용해 사용자 정보, 시스템 로그, AI 모델 연동 데이터 등 구조화된 데이터를 안정적으로 관리하고자 하였다. 로컬 개발 환경과 서버 간 호환성이 뛰어나고, MySQL Workbench 등과 같은 기존 RDBMS 도구를 활용할 수 있어 편리하다는 점에서 선택하였다.

3.3.4 외부 서비스

Cloudflare R2

서버리스 객체 스토리지 서비스로, AWS의 S3 API와도 호환이 되어 API 이미지, 로그, 백업 등 대용량 데이터를 안정적으로 저장하고 빠르게 접근할 수 있고 데이터의 이전도 쉽게 가능한 장점이 있다. 별도의 트래픽 비용 청구가 없어 부담없이 무료 사용이 가능하다. CDN과 연동이 가능하여 웹 서비스 성능 향상에도 유리하며, 별도의 인프라 운영 부담 없이 데이터 관리를 효율화하기 위해 사용하였다.

용량이 큰 얼굴인식 AI 모델을 R2 오브젝트 스토리지를 통해 보관할 버킷을 생성하고 트리 구조 형태로 저장한 후, ONNX 런타임을 통해 C++로 메모리에 로드하여 구동하였다. 매월 10GB의 저장 공간을 무료 플랜으로 제공하여 저장되는 사용자의 데이터도 안정적으로 저장할 수 있다.

NeonDB

NeonDB는 PostgreSQL을 기반으로 한 클라우드 데이터베이스 플랫폼으로, 서버 관리 없이 데이터베이스를 운영할 수 있는 서버리스 환경을 제공한다. 클라우드 기반으로 제공되기 때문에 사용자가 직접 로컬 서버 설정이나 인프라 관리를 해야 하는 부담이 줄어들고, 자동 스케일링 및 고가용성 기능을 활용할 수 있어 운영 효율성이 높다. 또한, PostgreSQL 호환성을 유지하면서도 백업, 복제, 복구 등의 기능이 자동화되어 있어 유지보수와 데이터 안정성 측면에서 장점이 있다.

본 프로젝트에서는 NeonDB를 사용하여 클라우드 환경에서 AI 모델 연동 데이터와 웹 서비스 관련 데이터를 관리하였다. 클라우드 기반 데이터베이스로서 로컬 환경과 별도로 효율적인 DB 운영으로 사용자 데이터 관리가 가능하며, 서버리스 특성을 활용해 프로젝트 초기 설정과 운영 부담을 최소화하고자 선택하였다.

Render

클라우드 기반 애플리케이션 호스팅 플랫폼으로, 웹 페이지 서버와 백엔드 API를 배포하고 관리하는 데 사용되었다. 자동 배포와 스케일링 기능을 제공하여 서비스 운영을 간소화하고, 개발 환경과 배포 환경을 통합적으로 관리할 수 있다는 장점이 있다.

실제 사용자는 Render에서 카메라를 통해 서버에 요청하고, 요청이 정상적으로 처리 되면 다시 결과를 반환받게 된다.

CoolSMS (Solapi SDK)

주소록 기반의 카카오톡 알림톡 및 문자메세지 웹발송 지원 서비스로, 자사 API인 SOLAPI SDK를 통해 간단한 API 연동 및 관리가 가능하다.

3.3.5. 협업 도구

이외, 프로젝트 수행 과정에서 팀 내 효율적인 소통과 협업, 실시간 진행 상황 관리를 위해 다양한 협업 도구를 활용하였다. 각 도구는 접근성과 기능 측면에서 장점을 고려하여 선정되었다.

GitHub

팀 단위 개발 시 코드 버전 관리와 변경 사항 추적을 위해 전세계적으로 많이 활용되는 형상관리 서비스로 Git의 push, pull, commit 등의 명령어로 팀원 간 코드 병합과 리뷰가 용이하며, 변경 이전 버전의 상태를 확인이 가능하여 협업 개발의 효율성을 높였다.

Zoom

비즈니스용으로 많이 사용되는 통합 커뮤니케이션 서비스로, 프로젝트 진행 간 주기적인 회의를 진행하기 위해 사용되었다. 화면 공유 기능을 통해 관련 자료와 진행 상황을 시각적으로 공유할 수 있어, 프로젝트 일정 조율과 의사결정에 효율적으로 사용이 가능한 협업 도구이다.

카카오톡

빠른 메시지 전달과 간단한 공지 공유를 위해 활용되었다. 국내 최대 규모의 메신저 서비스로 접근 편의성이 매우 높고 실시간의 빠른 소통과 일정 계획을 정할 수 있으며, 전달 사항 확인 등이 용이하다.

4. 결론

4.1. 프로젝트 결과

본 프로젝트에서는 얼굴 인식 기반 출입 통제 시스템의 기능인 얼굴 등록, 실시간 인식, 관리자 관리, QR코드 인증 기능을 구현하였다. 사용자의 얼굴 데이터가 등록 및 조회되어 출입 결과가 서버 데이터베이스에 반영되는 과정을 확인할 수 있었다.

그러나 실제 통합/배포 과정에서는 몇 가지 기술적인 어려움이 있었다, AI 모델 연동 과정에서 Python과 C++ 간 환경 차이로 인한 모델 로드 및 추론 과정에서 예외 발생과 성능 이슈가 생겼다. AI 모델의 경우, 다양한 조명 환경과 얼굴 각도 변화에 따라 인식 정확도가 일정하지 않은 문제가 있었으며, 이는 추후 모델 최적화나 데이터셋 확장을 통해 개선이 필요하다.

4.2. 기대 효과

결론적으로, 본 프로젝트로 인해 얻을 수 있는 기대효과로는 얼굴 인식을 통한 개인 인증으로 보안이 강화되며, 얼굴만으로 인증이 이루어지기 때문에 물리적인 불편함 없이 편리하게 출입이 가능하다. 전통적인 패스워드 방식 인증 대비 시간 절약이 가능하며, 편의성이 증대된다.

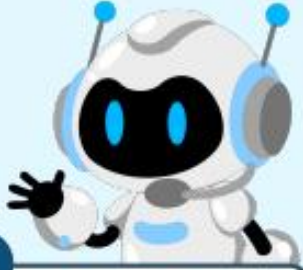
외부인 출입 시 단기 출입 인증을 별도로 두어 수행하기 때문에 보안을 더욱 강화할 수 있다. 이는 출입 카드 발급 및 관리, 분실 시 재발급 등과 같은 작업에 소모되는 노동력 감소와 자원 낭비를 예방하여 최종적으로 비용 절감 효과도 기대할 수 있다.

5. 참고 문헌

1. 김종기, 강다연. (2008). 패스워드의 정보시스템 보안효과에 영향을 미치는 요인에 관한 연구. *Asia Pacific Journal of Information Systems*, 18(4), 1-26.
2. 허석렬, 김강민, 이완직. (2021). 딥러닝 얼굴인식 기술을 활용한 방문자 출입관리 시스템 설계와 구현. *디지털융복합연구*, 19(2), 245-251.
3. 류성관, 박요한, 김창균, 박영호. (2016-06-22). 출입통제 시스템의 취약성 분석 및 대응방안. *한국통신학회 학술대회논문집*, 제주.
4. 손선경, 이윤경, 김정녀. (2024). IoT 디바이스의 코사인 유사도 분석을 통한 IoT 보안 위협 확산 방지 방법. *정보보호학회논문지*, 34(6), 1471-1483.
5. 천상훈, 최지훈, 김예진, 강성관. (2021). 젯슨 나노 기반의 OpenCV 및 딥러닝을 사용한 스마트 도어 구현. *한국통신학회논문지*, 46(2), 380-387.
10.7840/kics.2021.46.2.380
6. 김강철, Hai tong Wei. (2017). 라즈베리파이를 이용한 얼굴검출 및 인식 시스템 개발. *한국전자통신학회 논문지*, 12(5), 859-864.
7. 펜타시큐리티, 2025 인증보안 솔루션 리포트: iSIGN Password-less, 펜타시큐리티 공식 블로그, 2025. [온라인]
<https://www.pentasecurity.co.kr/in-the-news/2025-authentication-solutions-reports-penta-security-isign-password-less/>

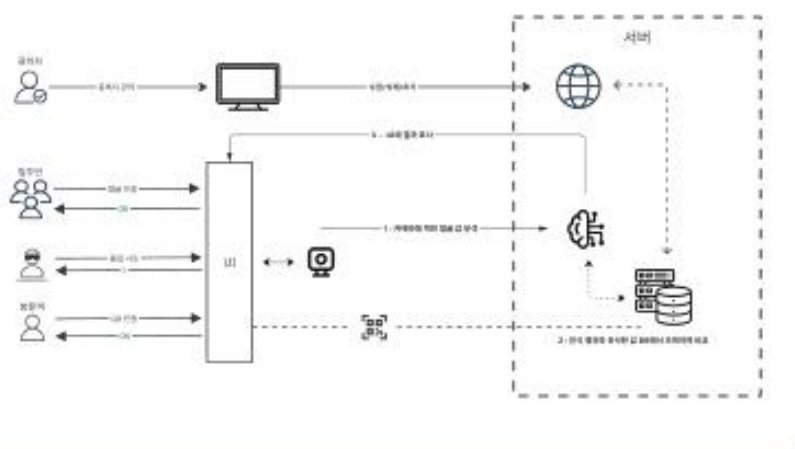
6. 별첨

6.1. 소개자료



인공지능을 활용한 물리적 보안 시스템

구상도



개요

본 연구의 목적은 인공지능 기술을 활용하여 사용자의 얼굴을 인식함으로써 출입을 통제하는 지능형 얼굴 인식 출입 통제 시스템을 개발하는 것이다. 이를 통해 비밀번호나 출입 카드에 의존하지 않고, 사용자의 생체 정보를 기반으로 한 고신뢰도 인증 방식을 구현하고자 한다.

기대효과

- ✓ 얼굴 인식을 통한 개인 인증으로 보안 강화
- ✓ 외부인 출입 시 단기 출입 인증을 통한 더욱 강화된 보안
- ✓ 물리적인 불편함 없이 편리한 출입
- ✓ 출입 카드 발급 및 관리, 분실 시 재발급 비용 절감 효과

팀 구성

팀장 - 김혜정, 팀원 - 신수민, 윤정목, 김건희, 김승언

2025 정보보호학과 졸업작품 전시회 @하나~ 둘~ 찰칵!

6.2. 프로젝트 소개

6.2.1 시연 영상

<https://www.youtube.com/watch?v=irGDTQZwbSw&feature=youtu.be>

6.2.2 웹 서비스 주소

<http://ai.pssuai.com/>

<http://api.pssuai.com/>

<http://qr.pssuai.com/>

<https://jjapt.onrender.com/login>

6.3. 팀원소개



팀장: 김혜정 (프로젝트 총괄 관리 및 관리자 페이지 개발)

<https://github.com/hyejeong22/jjapt>



팀원: 신수민 (클라이언트 및 백엔드 기반의 출입 시스템 개발, 통합 및 배포)

<https://github.com/LoadNim/AccessControlServer>

<https://github.com/LoadNim/AccessControl>



팀원: 김건희 (QR 전체 총괄 및 메시지 프로토콜 담당)

<https://github.com/000712>



팀원: 윤정목 (AI 모델 데이터셋 학습 및 검증)

<https://github.com/Yun-Jeong-Mok/ai-load-server/tree/main>



팀원: 김승언 (총괄 디자인)

<https://github.com/anvoanvo>

6.4. 발표자료

줄작PPT최종.ppt