

얼굴인식을 활용한 전자 출결 시스템

2020. 10. 23

중부대학교 정보보호학과

지도교수 : 양환석 교수님

4 조 - Forward

김 수 윤

우 다 혜

이 재 현

- 조원 편성
- 주제 선정
- 구 상 도
- 추진 경과
- 개발 환경 및 개발 내용
- 개발 시스템 운영
- 결론 및 기대 효과

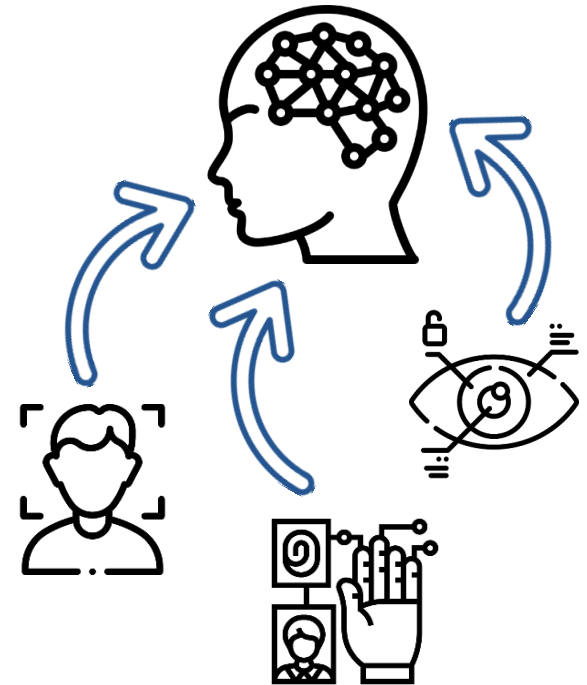
이름	역할
김수운(조장)	얼굴인식기술 개발, 작품 전반 개발
우다혜	Bootstrap을 이용한 WEB Design, PPT 제작
이재현	백엔드 서버, DB 개발

AI 시스템 → 인간 지능을 인공적으로 부가한 컴퓨터 시스템

AI 기술은 많은 양의 데이터와 빠른 처리능력,
강력한 알고리즘이 결합된 떠오르는 기술

과거에 없던 방법으로 일을 수행하는 AI는 용도
또한 광범위하여 잠재적 비즈니스 기회를 확대

특히 AI 기술을 활용한 인증 및 확인은 인증
서비스에 새로운 패러다임을 제시



□ 전자출결 시스템

- 블루투스, 와이파이, RFID 등의 통신방식이나 지문, 홍채, 얼굴을 인식하는 생체인식을 결합한 출결 시스템

□ 전자출결 시스템의 문제점

- 블루투스와 와이파이는 신호 간섭이나 감쇄로 인한 불안정 문제
- 지문, 홍채 등의 생체인식은 고가 장비가 필요하여 보편적 사용이 제한
- RFID는 물리적으로 짧은 인식 거리문제가 제기

AI를 이용한 얼굴인식 전자출결 시스템

- 시공간에 구애 받지 않는 자동화 출결 시스템
- 데이터가 누적되며 인식률을 지속적으로 높이는 반면 추가 비용이 거의 발생하지 않음

"ICT·科技로 확실한 도약"...문재인 정부 'AI 1등 국가' 선포

문재인 정부가 인공지능

정부는 연내 AI 인재 100
세대(5G) 통신 인프라에
공지능(DNA) 플랫폼을

문재인 대통령은 16일
터 2020년도 첫 번째 업

문 대통령은 정보통신기
는 강력한 의지에 따라
했다.

최기영 과기정통부 장관은 이 자리에서 "새해 AI
으로 삼는 것을 핵심 목표로 하겠다"고 보고했다

코로나19 안정성·보

신종 코로나바이러스감
변화 키워드는 온라인
고3을 제외한 나머지 학

새로운 교육 현장 패러
을 안정적으로 운영하
인 데이터를 통합해서

[칼럼] 패러다임 전환 시기 필요한 보안 전문가

김선애 기자 | 승인 2020.03.31 10:15 | 댓글 0

[데이터넷] 본격적인 4차 산업혁명 시대가 열리면서 보안
시장에도 급격한 기술적 패러다임 변화를 이루고 있다.
보안기업은 클라우드, 빅데이터, AI를 활용한 다양한 보안
솔루션을 개발, 공급하고 있다. 클라우드 도입이 가속화
되고, 마이크로서비스 아키텍처(MSA)-애자일 방식의 서
비스 개발 환경도 확산되고 있다.

그러나 여전히 많은 기업은 레거시(Legacy)라고 불리는
전통적인 방식을 고수하면서 패러다임의 변화에 빠르게
대응하지 못하고 있다. 보안 컨설팅이나 보안솔루션을 다
루는 관리자들은 전통적인 방법에서 벗어나지 못하고 있
다.

보안 인재 육성 위해 기업 스스로 노력해야

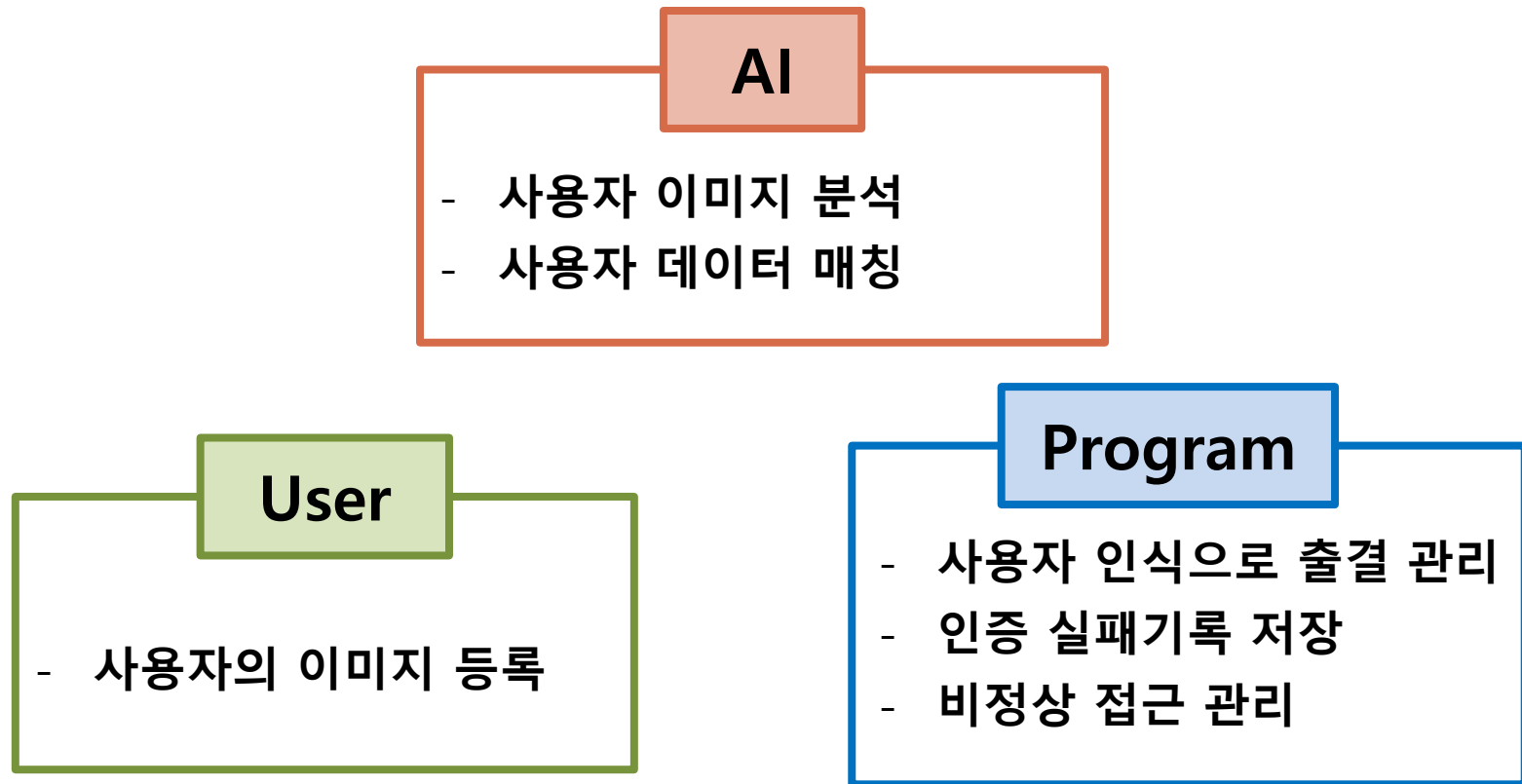
4차 산업혁명 트렌드에 맞추기 위해 많은 기업이 관련 기술을 도입했거나, 도입을 고려하고 있지
만, 이를 운영하는 사람에 대한 투자는 소극적인 상황이다. 당장 드러나는 문제로 클라우드 보안,
IoT 보안, AI 보안 분야 등은 인력 부족이 심각하다. 보안에서 가장 중요한 요소인 사람, 즉 전문가
에 대한 투자의 중요성이 점점 더 부각되고 있다. 시급한 많은 기업들이 클라우드를 도입해서 활용
할 뿐 명확



<김석 노브레이크 대표이사>

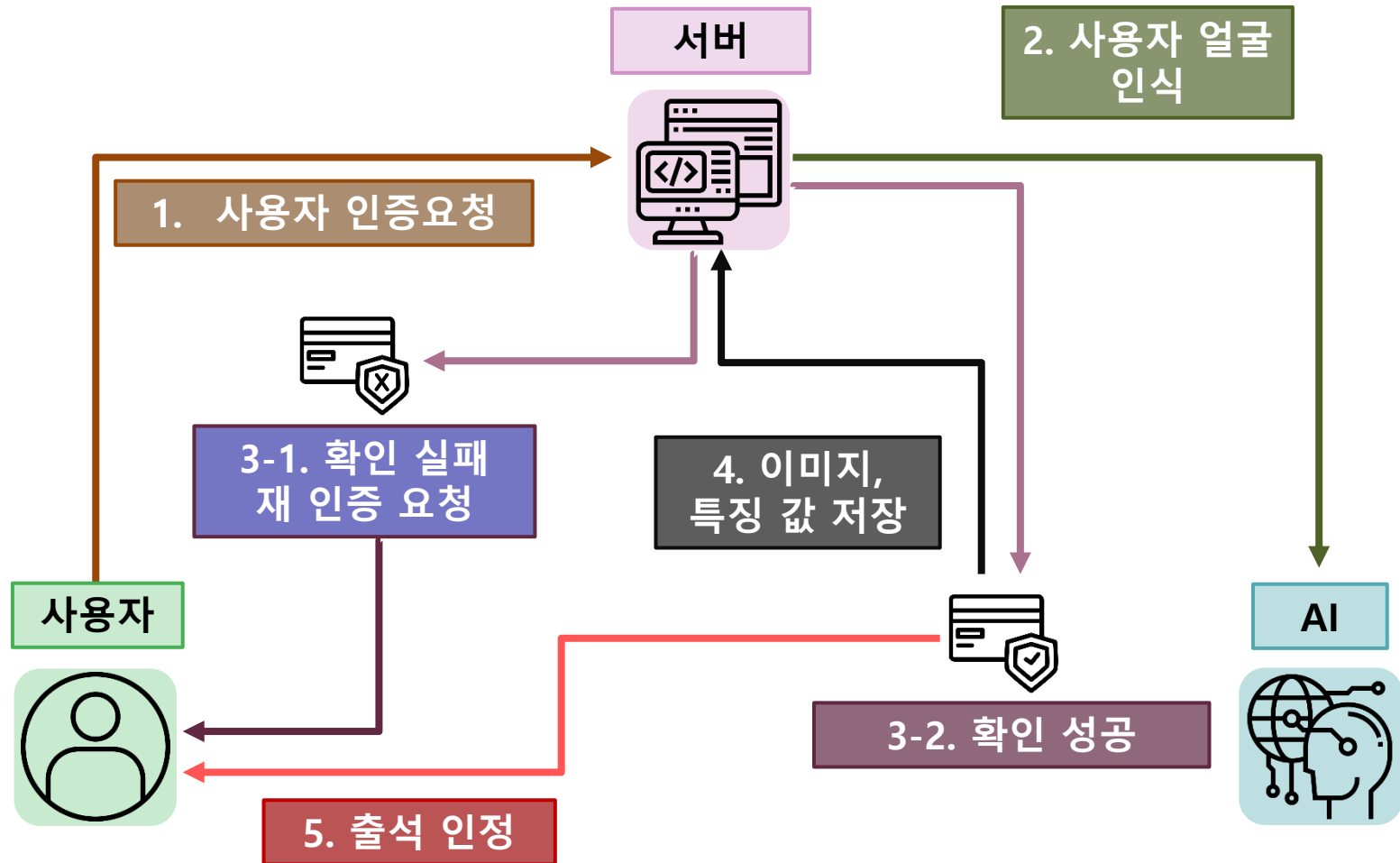
AI를 활용한 얼굴 인식 출결 시스템 개발

기능적 측면



사용자 얼굴을 매칭하여 사용자 검증 및 출결 관리

동작 원리



수행 업무 \ 추진기간 (2020년)	3월	4월	5월	6월	7월	8월	9월	10월
자료 조사 및 연구								
얼굴 인식 기술 개발 (AI 적용)								
출결 프로그램 개발								
web 개발								
서버 개발								
시스템 검증 및 보완								

개발 환경

Backend

Flask

Frontend



OS



Windows 10

DB



Development
Language



python

개발 내용 (1/10)

❖ Phone 영역 제외 구현

```
def apply_brightness_contrast(self, input_image):  
    if brightness != 0:  
        if brightness > 0:  
            face_landmarks_list = face_recognition.face_landmarks(input_image)  
            shadow = 1.0 - brightness  
            highlight = 1.0 + brightness  
            if face_landmarks_list:  
                filtered_image = self.apply_brightness_contrast(cv2image, -127, 127)  
                hsv = cv2.cvtColor(filtered_image, cv2.COLOR_BGR2HSV)  
  
                lower_blue = np.array([0, 90, 55])  
                upper_blue = np.array([20, 255, 200])  
  
                hsvmask = cv2.inRange(hsv, lower_blue, upper_blue)  
                kernel = np.ones((10, 10), np.uint16)  
                hsvmask = cv2.morphologyEx(hsvmask, cv2.MORPH_OPEN, kernel)  
                hsvmask = cv2.morphologyEx(hsvmask, cv2.MORPH_CLOSE, kernel)  
  
                res = cv2.bitwise_xor(cv2image, filtered_image, mask=hsvmask)  
            else:  
                shadow = 1.0 - brightness  
                highlight = 1.0 + brightness  
                alpha_b = (highlight - shadow) / 2  
                gamma_b = shadow + alpha_b  
                buf = cv2.addWeighted(input_image, alpha_b, filtered_image, gamma_b, 0)  
        else:  
            buf = input_image  
    if contrast != 0:  
        f = 131 * (contrast + 128) / (255 * (contrast + 1))  
        alpha_c = f  
        gamma_c = 128 - f  
        buf = cv2.addWeighted(buf, alpha_c, input_image, gamma_c, 0)
```

이미지 밝기, 대비조절

이미지 색상 추출

Phone 영역 제외를 위한 캠 이미지 처리

개발 내용 (2/10)

❖ 얼굴인식 기술 구현

```
if (os.path.isfile(csvfiledir)):
```

```
    countA = 0
```

```
    countB = 0
```

```
    with open(csvfiledir, 'r') as csvfile:
```

```
        reader = csv.reader(csvfile, delimiter=',')
```

```
        for row in reader:
```

```
            countA = countA+1
```

```
            face_feature = list(map(float, row[1:]))
```

```
            face_distances = face_recognizer.distance(face_feature)
```

```
            print('저장된 데이터와 ', 100 - face_distances*100, '%')
```

```
            if face_distances<0.2:
```

```
                df.to_csv(csvfiledir, mode='a', index=False)
```

```
                break
```

```
            else:
```

```
                countB = countB + 1
```

얼굴 인식 및 저장

```
ff = face_function()
```

```
if ff.check_face_num(cv2_image) == True:
```

```
    if (os.path.isdir('students/' + str(stu_num) + '_' + name)):
```

```
        pass
```

```
    else:
```

```
        os.mkdir('students/' + str(stu_num) + '_' + name)
```

```
check = ff.registerimage(cv2_image, stu_num, name)
```

```
time_today = str(datetime.now().year) + "." + \
```

```
    str(datetime.now().month) + "." + str(datetime.now().day)
```

```
imagedir = 'students/' + str(stu_num) + \
```

```
    '_' + name + '/' + time_today + '.jpg'
```

```
if check == True:
```

```
    im.save(imagedir, 'png')
```

```
    result = {
```

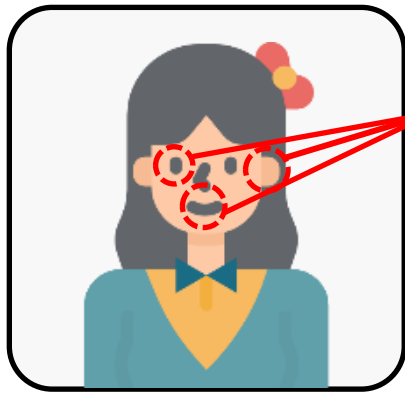
```
        "success": True,
```

이미지 저장

캠 이미지에 인식된 얼굴과 저장된 특징 값을 비교 후
특징 값/이미지 저장과 결과 전송

개발 내용 (3/10)

❖ 특징 데이터 저장과 매칭



0.011368	0.012032	0.011949	-0.1173	특징 데이터	
-0.03276	0.024633	0.016156	-0.1239		
-0.00772	0.054429	0.00731	-0.09373	-0.05735	-0.00
-0.01521	0.051318	-0.00864	-0.09791	-0.07106	0.002
-0.0088	0.056095	-0.00861	-0.08949	-0.08018	-0.00
-0.01031	0.066893	-0.0032	-0.09484	-0.06989	-0.01
0.011691	0.040069	0.007666	-0.06395	-0.07254	-0.00
0.0020	0.066826	0.00474	0.00622	0.07051	0.00



이미지 얼굴 좌표 : [(268, 397, 397, 545)]
저장된 데이터와 [74.75796047] % 일치합니다.
저장된 데이터와 [71.57260198] % 일치합니다.
저장된 데이터와 [77.54124585] % 일치합니다.
저장된 데이터와 [80.94089831] % 일치합니다.

Back

저장된 사용자의 얼굴 특징 데이터와 매칭

개발 내용 (4/10)

❖ 웹 개발

```
<body class="alert alert-secondary">
  <main role="main">
    <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h3>Forward</h3>
        <p>third slide test</p>
      </div>
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
```

메인 페이지 구성

Carousel 사용

부트스트랩을 사용하여 웹페이지 디자인

개발 내용 (5/10)

❖ DB 연동 및 설계

DB 연결

동작 설계

```
1  import pymysql
2  import os
3
4  class Database:
5      def __init__(self, host, user, password, database):
6          self.host = host
7          self.user = user
8          self.password = password
9          self.database = database
10         self.db = pymysql.connect(host=self.host, user=self.user, password=self.password, database=self.database)
11
12         self.cursor = self.db.cursor()
13
14         self.execute("CREATE TABLE IF NOT EXISTS users (id INT(11) PRIMARY KEY, name VARCHAR(255), email VARCHAR(255), password VARCHAR(255))")
15
16         self.commit()
17
18         self.close()
19
20     def execute(self, query, args={}):
21         self.cursor.execute(query, args)
22
23     def executeOne(self, query, args={}):
24         self.cursor.execute(query, args)
25         row = self.cursor.fetchone()
26         return row
27
28     def executeAll(self, query, args={}):
29         self.cursor.execute(query, args)
30         row = self.cursor.fetchall()
31         return row
32
33     def commit(self):
34         self.db.commit()
35
36     def close(self):
37         self.cursor.close()
38         self.db.close()
```

데이터 송수신을 위한 함수 구현

개발 내용 (6/10)

❖ CRUD API 설계

```
def register(self, User):
    sql = "INSERT INTO users(stu_num, name, email, password)"
    sql += " VALUES('{'', '{', '{', '{')";".format(User.get('stu_num'), User.get('name'),
    User.get('email'), User.get('password'))
    try:
        self.cursor.execute(sql)
        self.db.commit()
        result = True
    except Exception:
        result = {"error": "register failed"}
    return result
```

Create 설계

```
def login(self, stu_num, password):
    sql = "SELECT stu_num, password "
    sql += "FROM users "
    sql += "WHERE stu_num='{'';".format(stu_num)
    try:
        saved_pass = self.executeOne(sql)
        if bcrypt.check_password_hash(saved_pass.get('password'), password):
            return True
    except Exception as e:
        return {"error": "{''.format(e)}
```

Read 설계

회원가입, 로그인을 위한 DB 동작 설계

개발 내용 (7/10)

❖ DB 데이터 확인

Table 목록					
이름 ^	행	크기	생성됨	업데이트됨	
attend	0	16.0 KIB	2020-06-01 17:1...		InnoDB
info	4	16.0 KIB	2020-06-12 16:5...		InnoDB
professo					
student					
users					

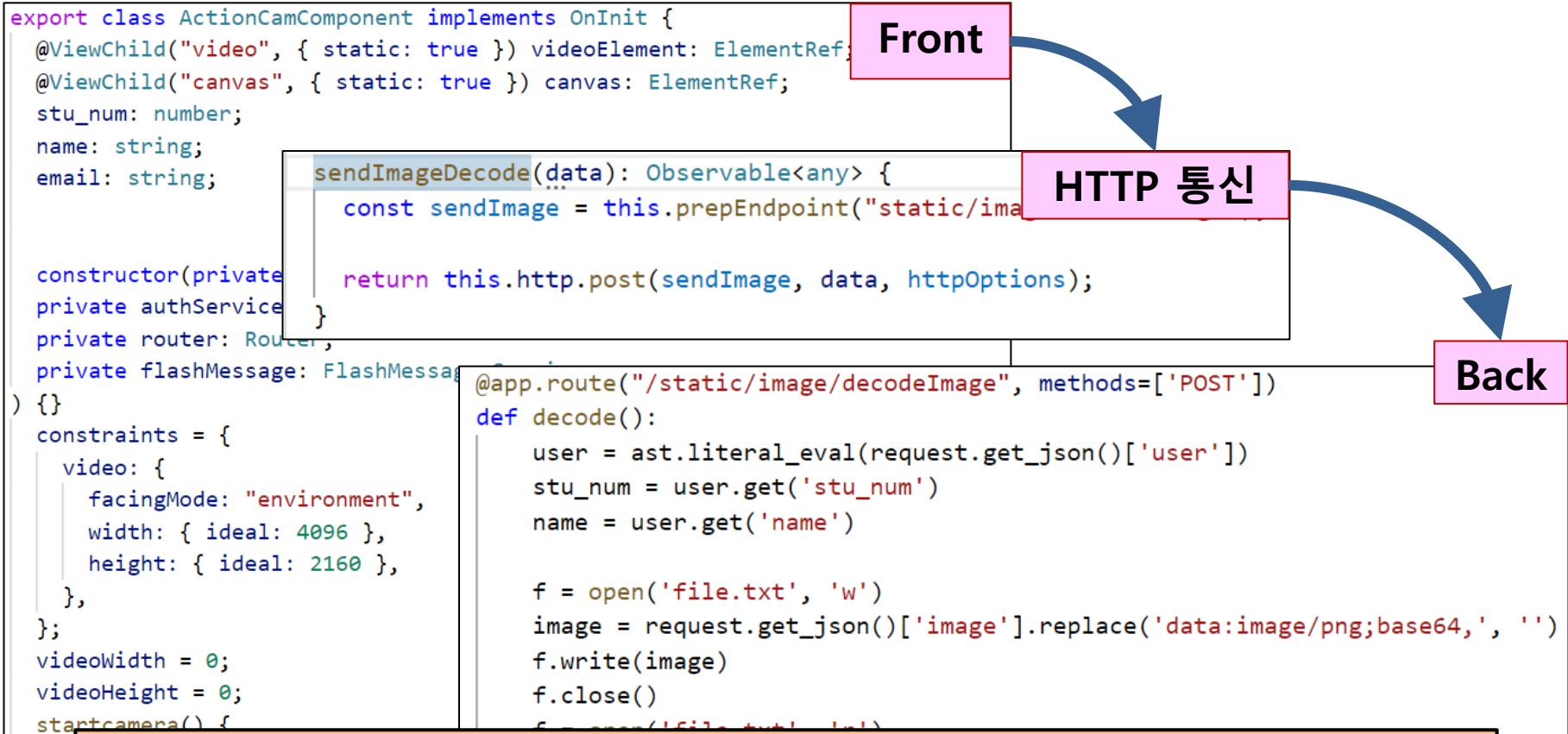
stu_num	name	password	
123	bcbc	\$2b\$12\$HoFy4T8bxts7eCydZ/kjvuRNQcnX3plrVA...	bcbc@bcbc.com
22,323	bcbc2	\$2b\$12\$tt6nZNBz7HVTP3NcTkB2DeBSzKonHEKU...	bcbc2@bcbc.com
3,334	af	\$2b\$12\$uT3rvat7lcu.w7qlonUzcO13DZdYdCle/TLE...	af@af.af
22	ak@ak	\$2b\$12\$9SyozLukPkeoRU0Uk51nJev44tIXhlUfoR7...	ak@ak.ak
22	ak@ak	\$2b\$12\$xhMxsX0AJusyVKSioRgyruITsfZHLBRMLr.n...	ak@ak.ak
223	2323	\$2b\$12\$vKmRhde2henNYZX/29SotOkXoRw.EgXN...	asdf@df.com
1,234	Lee	\$2b\$12\$uaEOXN3vSJSZC1AtWcDpYutiHt.EoCtGeH...	swaa23@gmail.com
123	bcbc	\$2b\$12\$7OKmISVuo7eUaerUsII7R.Q3QcGIJySKyB5...	2020.9.1
424	aa	\$2b\$12\$C5LaXk3SvFuC1LgnnC6Mj.HSujg2InMJEa...	aa@aa.com
4,243	4243	\$2b\$12\$tV9EUqolrBNALwG9T44Kjeq2yAjkY5xYnL...	423@234.com
4,455	33	\$2b\$12\$ezlrsXqO3J9ECKauFFzG2.Sgm0JXTc/myRV...	23@32.com
4,455	33	\$2b\$12\$8EIR4IO5Fx3FDdfIZV2CI.eH1LjW6gvcMR/...	23@32.com
424	JaeHyun Lee	\$2b\$12\$liLJ3AUOJ1ZrBHj6egwvQ/VLyxcudCiKFM...	swaa23@gmail.com
555	1234	\$2b\$12\$HsmK/UmvUKAKlMq/vLiOdep5UE0La5C...	swaa23@gmail.com

저장된 Data

저장된 mySQL DataBase 확인

개발 내용 (8/10)

❖ 웹캠 캡처 기능



웹캠으로 캡처한 이미지를 서버로 전송하여 처리

개발 내용 (9/10)

❖ 출결 시스템 구현

```
ngOnInit() {  
  var msg;  
  this.authService  
    .getSubjectData(localStorage.getItem("user"))  
    .subscribe((data) => {  
      msg = data["msg"];  
      for (var i = 0; i < msg.length; i++) {  
        var date_form = new Date(msg[i].atten_date);  
        var date_str =  
          date_form.getFullYear().toString() +  
          "." +  
          (date_form.getMonth() + 1).toString() +  
          "." +  
          date_form.getDate().toString();  
        this.Attentionlist.push(  
          new Attention(msg[i].stu_num, msg[i].week, date_str, msg[i].atten)  
        );  
      }  
    });  
}
```

출석 체크 - 학생

출석기록 조회 기능 구현

개발 내용 (10/10)

❖ 출결 시스템 구현

```
ngOnInit() {  
  this.authService.checkadm  
    (token) => {  
      this.token = token;  
      this.admin @app.route("/static/admin/check  
      this.admin @jwt_required  
      def checkadmin():  
        token data = get jwt identity()  
    },  
    selectWeek(event: any){  
      this.adminAttenlist = [];  
      var atten_data  
      this.authService.getAttenData(event.target.value).subscribe(data => {  
        atten_data = data["data"];  
        for (var i = 0; i < atten_data.length; i++){  
          this.adminAttenlist.push(  
            new adminAtten(i+1, atten_data[i].name, atten_data[i].stu_num, atten_data[i].atten)  
          )  
        }  
      })  
    }  
  }  
}
```

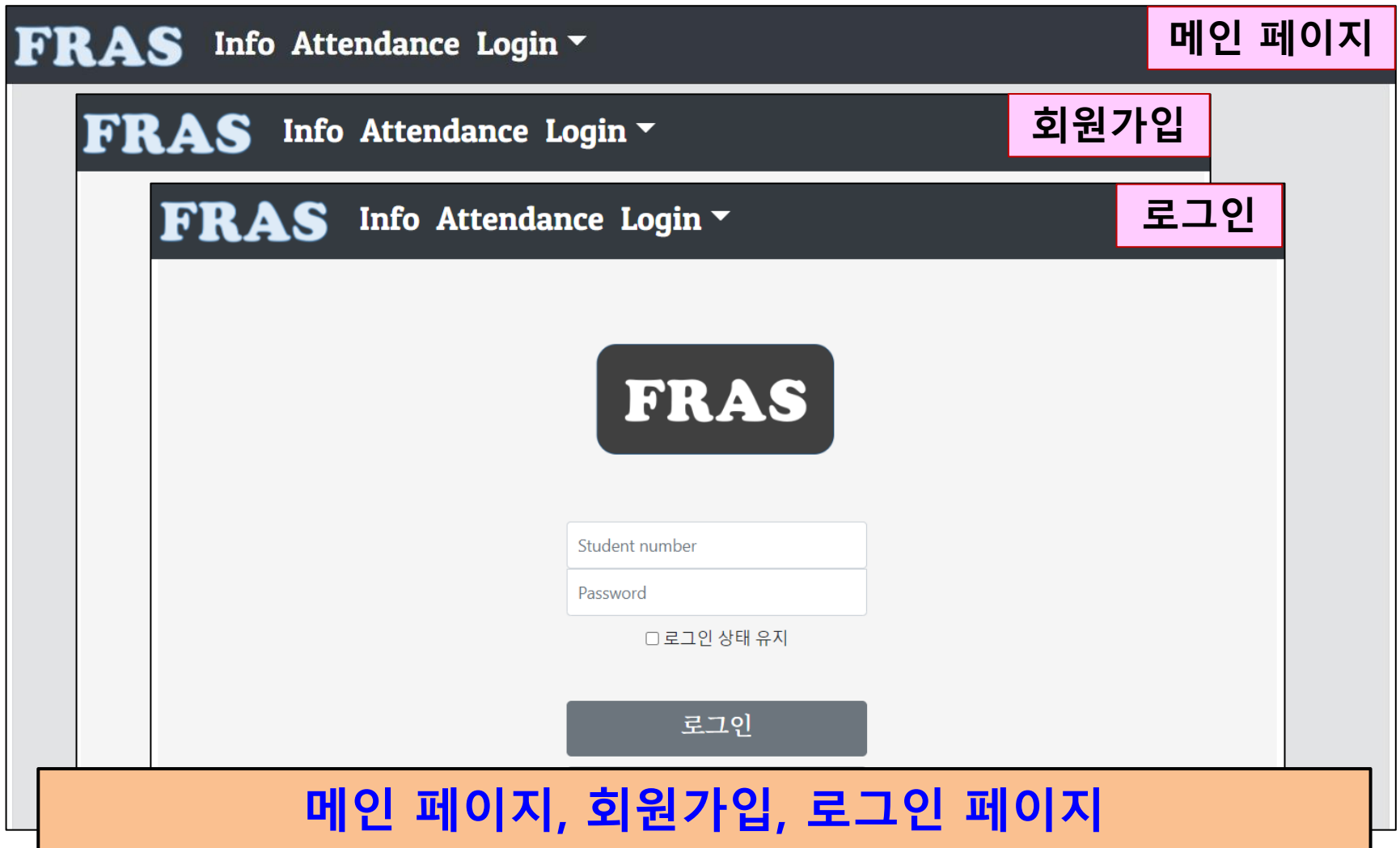
JWT 인증요청

JWT 결과반환

출석확인

관리자 인증 / 출석기록 조회 기능 구현

초기 화면



The screenshot shows the FRAS Info Attendance Login page. The page has a dark header with the text "FRAS Info Attendance Login" and a dropdown arrow. The main content area is white and contains the FRAS logo, input fields for "Student number" and "Password", a checkbox for "로그인 상태 유지" (Keep login state), and a "로그인" (Login) button. Annotations in pink boxes point to the "메인 페이지" (Main page), "회원가입" (Sign up), and "로그인" (Login) links. A large orange box at the bottom contains the text "메인 페이지, 회원가입, 로그인 페이지" (Main page, Sign up, Login page).

FRAS Info Attendance Login ▾

메인 페이지

FRAS Info Attendance Login ▾

회원가입

FRAS Info Attendance Login ▾

로그인

FRAS

Student number

Password

☐ 로그인 상태 유지

로그인

메인 페이지, 회원가입, 로그인 페이지

출결 시스템 구현 (1/3)

FRAS Info Attendance Login ▾

출결 현황

강의명	담당교수	학점	강의기간 및 시간
정보보호프로그래밍	홍길동	3	수요일 3교

출석일: 2020-09-26

출결 사진



닫기

주차	마감일	출결 현황	출결사진
1주차	2020.8.25	지각	보기
2주차	2020.9.1	출석	보기
3주차	2020.9.8	출석	보기
4주차	2020.9.15	출석	보기

학생 출석 / 출결 현황 페이지

출결 시스템 구현 (2/3)

FRAS Info Attendance Login ▾

웹캠 캡처



Capture Image

웹 카메라 얼굴인식 페이지

출결 시스템 구현 (3/3)

FRAS Info Attend_Admin Login ▾

관리자 출결 확인

FRAS Info Attend_Admin Login ▾

학생 출결 목록

강의명		담당교수	학점	강의기간 및 시간	
정보보호프로그래밍		홍길동	3	수요일 3교시 ~ 6교시	
주차	마감일	출결 현황	출결사진	출석수정	
1주차	2020.8.25	출석	보기	출결사항수정 ▾	
2주차	2020.9.1	지각	보기	출결사항수정 ▾	
3주차	2020.9.8	지각	보기	출결사항수정 ▾	
4주차	2020.9.15	지각	보기	출결사항수정 ▾	
5주차	2020.9.22	지각	보기	출결사항수정 ▾	
6주차	2020.9.29	출석	보기	출결사항수정 ▾	

관리자 출결 확인, 학생 출결 목록 페이지

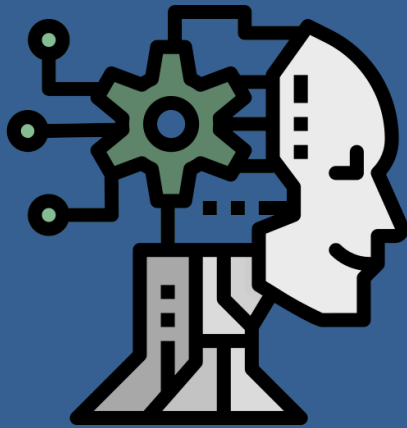
❖ 결 론

- 최근 활용도가 급증하고 적용분야가 확대되고 있는 AI 기술을 적용한 전자 출결 시스템을 개발하는데 성공
- Full Stack 프로젝트의 임무를 적절히 분담하고 팀워크를 이루어 난이도 높은 시스템을 구현

❖ 기대 효과

- 사용자 부정출석 방지를 위한 캠 이미지 인식기술을 개발하여 출석체크의 효율성과 관리자의 편의성을 향상
- COVID-19로 인해 필수가 된 비대면 강의를 보조하는 얼굴인식 기술을 활용한 출석 시스템을 개발하여 사회적 거리두기 확립에 기여. 끝.

Q&A



감사합니다