**[개별일지]**

* ‘일지\_조각들’ 파일의 내용을 참고하여 해당 날짜에 본인이 무엇을 했는지 작성한다.
* 작성된 개별일지는 추후에 일지 하나로 통합할 예정.
* ‘일지\_조각들’은 슬랙, 카톡, 공유 드라이브 등을 참고하여 짜깁기한 자료이다.
* ‘일지\_조각들’에 빠진 내용이 있다면 그 파일을 수정하지 말고 이 개별일지 파일에 함께 작성한다.
* 기억이 안나면 일단 비워 두고 넘어가기!
* 코딩 수정에 대해서는 되도록이면 말로 풀어서 설명하기
* 사진 등의 참고자료를 함께 넣어주면 좋습니다!
* 어투를 다듬어서 작성할 필요는 없음 (예: ~함. ~했다. ~한..듯? 등 혼합하여 막 써도 됨)
* 어투는 나중에 합치면서 다듬을 예정! (어차피 셋의 어투도 기본적으로 다르니까…)
* 8월 21일 월요일에 한번 중간점검 겸 크로스체크 할 예정 (서로의 기억을 더듬어 빈칸을 채우고, 그 날짜에 그 일을 한 게 정말 맞는 지 점검할 예정)
* 8월 25일 금요일까지 각자 개별일지를 작성한 후 업로드하여 공유할 예정

|  |  |
| --- | --- |
| **날짜** | **내용** |
| 7월 3일 월요일 | (미니 프로젝트 결성, 회의 진행) |
| 7월 4일 화요일 | - 필요한 헬멧 데이터셋 다운로드  - retinaNet 공부 |
| 7월 5일 수요일 | - 헬멧 데이터 전처리 실행, 모델 학습  - 테스트용 동영상 데이터 다운로드 |
| 7월 6일 목요일 | - retinaNet 필요 메소드들 모듈화 |
| 7월 7일 금요일 | - 미니 프로젝트 발표자료 정리 |
| 7월 10일 월요일 | (미니 프로젝트 발표) |
| 7월 11일 화요일 | X |
| 7월 12일 수요일 | X |
| 7월 13일 목요일 | X |
| 7월 14일 금요일 | X |
| 7월 17일 월요일 | X |
| 7월 18일 화요일 | X |
| 7월 19일 수요일 | X |
| 7월 20일 목요일 | X |
| 7월 21일 금요일 | - OCR 광학문자인식 공부, 모델 알아보기  - Tesseract OCR, Kraken, OCRopus > tesseract OCR 선택 |
| 7월 24일 월요일 | - 프로젝트 흐름도 초안 작성  - 장고 수업&공부 |
| 7월 25일 화요일 | - 장고 수업&공부 |
| 7월 31일 월요일 | - 장고 수업&공부 |
| 8월 1일 화요일 | - 데이터 레이크(테이블) 설계, erd 작성  - 프로젝트 아키텍처(흐름도) 수정 |
| 8월 2일 수요일 | - erd 수정  - 고글, 안전조끼, 일반 모자들 데이터셋 다운로드, augmentation 실행  - css framework 알아보기 > tailwindcss 선택 |
| 8월 3일 목요일 | - 프로젝트 아키텍처(흐름도) 수정 |
| 8월 4일 금요일 | - tailwindcss 공부 |
| 8월 5일 토요일 | X |
| 8월 7일 월요일 | - tailwindcss + 장고로 로그인 화면 구현 |
| 8월 8일 화요일 | - tesseract OCR 사용법 자세히 공부 |
| 8월 9일 수요일 | - tesseract OCR 모델과 웹캠을 연결해 실시간 문자 인식 코드 작성  (화면 왼쪽 상단에 인식한 문자가 출력되도록 구현)  - 쓸데없는 문자 인식한 경우 제외하는 정규표현식 코드 작성  (숫자 4자리만 결과로 출력하도록) |
| 8월 10일 목요일 | - ocr 이미지 전처리 공부  - tesseract OCR 성능 올리기 위한 이미지 전처리 코드 작성  (grayscale, 대비 올리기, 노이즈 제거) |
| 8월 11일 금요일 | X |
| 8월 12일 토요일 | - IP webcam 앱 이용 핸드폰 카메라 연결 코드 작성  - tesseractOCR 성능 이슈로 easyOCR 실행 > 모델이 무거워 기각  - 경량화 OCR 모델 paddleOCR 실행 > 성능 나쁘지 않아 이걸로 결정 |
| 8월 14일 월요일 | - PaddleOCR 사용법 자세히 공부  - 글씨체별(궁서체, d2coding, 맑은 고딕) 문자인식 테스트 > 궁서체 결정  - 깃허브 공유 프로젝트 생성, documentation 레파지토리 생성 |
| 8월 15일 화요일 | X |
| 8월 16일 수요일 | - paddleOCR 클래스 파이썬 파일 작성  - 장고 프로젝트 생성해서 4개(access, attend, login, visualization) 앱 생성, 디비 테이블 생성  - 깃허브 레파지토리(ocr) 생성, 버전 업로드, documentation 레파지토리에 문서 추가 |
| 8월 17일 목요일 | - paddleOCR 클래스 파이썬 파일 return 값 수정  (text list, confidence list로 반환)  - 장고 디비 + mySQL 연결  - 코드 테이블, 작업자 테이블 사전 데이터 입력  - DB 데이터 웹에 뿌리기 성공  - 장고 form을 이용해 웹에 입력한 데이터를 DB에 저장 |
| 8월 18일 금요일 | - 장고 + OCR 모델 연결  - OCR 모델 결과값 웹에 출력  - OCR 모델 결과값을 DB에 INSERT |
| 8월 21일 월요일 | - workers, attendance, safety\_check 테이블 외래키 연결  - visualization/ : 코드 정보 확인 기능 구현  - OCR 모델 결과 w\_id와 workers 테이블 w\_id 비교한 후 디비 insert  attend/ : 출근부 입력 기능 구현  - attendance 테이블에 같은 날짜에 w\_id가 이미 있을 경우 not insert 처리  - attend/create\_worker : 새로운 작업자 등록 기능 구현  - attend/leave : 퇴근 찍는 페이지&함수 구현 |
| 8월 22일 화요일 | - 최초 detection 결과에 한해서 safety\_check 테이블 입력 구현  - access, slack, place 테이블 외래키 연결  - access/place\_info : 위험 지역 정보 출력 기능 구현  - access/ : 입장 기능 구현 |
| 8월 23일 수요일 | - access/ : 퇴장 기능, 재입장 기능 구현  - access/place\_info : 실시간 구역별 인원 수 확인 기능 구현  - visualization/statistics : 월별 최초 통과 보호구 수 그래프 그릴 뷰 |
| 8월 24일 목요일 | - 수연언니 dash랑 내 DB랑 연결  - html 연결 |
| 8월 25일 금요일 | - access 시간차 두기 (출근, 퇴근 동적 작용)  - html 서로 동적 연결  - nas에 DB 생성, 장고에 연결  - leave.html (퇴근 페이지) 생성 |
| 8월 26일 토요일 | - 개별일지 정리, 사진들 캡처  - 월, 일별 보호구 정상/비정상 착용 인원수 조회 뷰(normal\_sf\_view) 생성  - 통계 대시보드 구성 생각  - 헬멧, 고글, 조끼 세 컬럼으로 나눠 조회하는 뷰(hgv\_cnt\_view) 생성 |
| 8월 28일 월요일 | - ip\_webcam 연결해서 웹페이지에 송출 성공  - 실시간 ocr, detection 모델 연결. DB에 데이터 저장 성공 |
| 8월 29일 화요일 | - 30분 간격으로 slack 테이블에 경고 저장하기  - access 실시간  - 프로젝트 경량화 |
| 8월 30일 수요일 | - DB에 경고가 두 개씩 실행되는 문제점 발생 > 스케줄러 인스턴스가 프로젝트가 reload됨에 따라 중복 실행되는 걸로 파악 > --noreload로 실행하면 문제없이 한번만 데이터 입력됨 |
| 8월 31일 목요일 | - 로그인/레지스터 화면 수정  - 유리언니가 전체적인 웹페이지 다듬기 |
| 9월 1일 금요일 | - 각자 수정했던 것들 합치기  - 카메라 연결 안됐을 때 오류 페이지 처리  - 출퇴근 스크롤바 없애기  - 통계 차트 더 다듬기/가로 스크롤 없애기/예쁘게 만들기 |
| 9월 4일 월요일 | - access 페이지 다듬기  - 로그아웃 버튼 모든 페이지에 띄우기 (지금은 대시보드에만 있음)  - attend/access 합쳐서 ocr\_model 하나만 남기기 (import로)  - logo 이미지 수정본으로 변경 (workArea에 원래 ver 주석처리함)  - mediapipe 사용법 공부 |
| 9월 5일 화요일 | - 대시보드 꾸미기  - 추가학습 모델 성능 테스트 |
| 9월 6일 수요일 | - 웹 화면 마무리  - attend 알고리즘 정의서 작성 |
| 9월 7일 목요일 | - access 알고리즘 정의서 작성 |
| 9월 8일 금요일 | [문제]  webcam.html이 로딩되면 check\_attendance 함수 실행, check\_attendance 내의 safety\_chk가 실행되는 동안 webcam.html에는 ‘확인중’이라는 메시지를 띄우고 싶음. 하지만 메시지를 띄우기 위해 웹 페이지를 reloading하면 check\_attendance 함수가 재실행되기 때문에 원하던 처리가 제대로 되지 않음. 이걸 해결하려면 어떻게 해야 할까?  [해결]  AJAX를 사용하면 페이지 전체를 새로고침하지 않고도 서버에 데이터를 요청하거나 서버로부터 데이터를 받아올 수 있음. 따라서 페이지의 일부분만 업데이트하므로 check\_attendance 함수가 재실행되지 않는다.  AJAX 요청이 성공적으로 완료되면 check\_attendance 함수의 return값인 render가 실행되고, 그게 아니라면 메시지 부분만 ‘확인 중..’으로 업데이트. 요청이 실패하면 error 콜백 함수 실행  따라서 웹 페이지의 일부분만 동적으로 업데이트하는 데 이 방법을 사용할 수 있으며, 이렇게 하면 사용자 경험을 향상시키고 서버 리소스를 효율적으로 활용할 수 있다 !! |