



캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
팀 명	DRM
문서 제목	결과보고서

Version	1.5
Date	2024-05-20

팀원	이동국 (20192282) (팀장)
	이강희 (20181653)
	장수미 (20193216)
	허수빈 (20201320)
	황솔희 (20203168)
	최지원 (20213091)

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “시선 추적을 통한 유동인구의 광고 관심도 분석 대시보드”를 수행하는 팀 “DRM”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “DRM”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	결과보고서-지하철 탑승객 행동 기반 광고 관심도 분석 대시보드.doc
원안작성자	이동국, 이강희, 장수미, 최지원, 허수빈, 황솔희
수정작성자	최지원, 허수빈

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-05-15	장수미	1.0	최초 작성	프로젝트 개요 작성 및 유저 인터페이스 작성
2024-05-16	이동국	1.1	내용 추가	모델 서버 및 아키텍처 파트 작성
2024-05-17	이강희	1.2	내용 추가	AI 파트 작성 및 라벨링 예시 추가
2024-05-18	최지원	1.3	내용 추가	클라이언트 및 관리자 메뉴얼(GUI) 추가
2024-05-19	황솔희	1.4	내용 수정	모델 서버 수정된 내용 작성
2024-05-20	허수빈	1.5	내용 수정	AI 파트 수정 및 모델 결과값 추가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

목 차

- 1 개요
 - 1.1 프로젝트 소개
 - 1.2 추진 배경 및 필요성
 - 1.2.1 서비스 소개
 - 1.2.2 서비스 목표
 - 1.3 팀 소개
- 2 개발 내용 및 결과물
 - 2.1 목표
 - 2.2 협업 방법
 - 2.3 연구/개발 내용 및 결과물
 - 2.3.1 모델
 - 2.3.2 유저 인터페이스
 - 2.3.2.1 Figma Styles 지정
 - 2.3.2.2 UI Description 작성
 - 2.3.2.3 UXUI 디자인
 - 2.3.3 클라이언트
 - 2.3.4 서버
 - 2.4 기대효과 및 활용방안
- 3 자기평가
- 4 참고 문헌
- 5 부록
 - 5.1 사용자 매뉴얼
 - 5.2 운영자 매뉴얼
 - 5.3 배포 가이드
 - 5.4 XXX 매뉴얼
 - 5.5 테스트 케이스

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

1 개요


1.1 프로젝트 소개

광고주는 자사 광고가 실제로 어떤 효과를 내는지 알고 싶어 한다. 또한 소비자들이 정말로 자신의 광고를 관심 있게 보는지에 대한 의문이 있다. 온라인 광고처럼 오프라인 옥외 광고도 분석이 가능하다면 광고주는 더욱 효율적으로 광고 집행이 가능할 것이다. 이러한 니즈를 충족시키기 위해 오프라인 광고 분석 서비스 *Do you READ me*의 개발을 시작하였다.

팀 DRM은 다음 세대의 광고 대행사를 만들고자 한다. 기존의 광고대행사들은 단순히 디스플레이에 광고를 띄우는 데 그치는데, 이는 광고의 효과를 정확히 파악하는 것이 불가능하다. 우리는 이러한 한계를 극복하고, 광고 송출에 광고 분석 서비스를 추가하여 오프라인 광고의 실제 효과를 정확히 측정하고자 한다.

서비스명 *Do you READ me?*에서 사용되는 *read*는 사전적 의미로, 단어나 상징을 보고 그 의미를 이해하는 것(*to look at words or symbols and understand what they mean*)을 의미한다. 서비스에서 *read*는 두 가지 의미를 갖는다. 첫째, 서비스는 소비자가 광고를 실제로 인지하고 주목하는지를 알려준다. 둘째, 서비스는 광고주가 광고의 효과를 제대로 이해하도록 대시보드를 제공한다. *Do you READ me?*를 통해 사용자는 온라인 광고뿐만 아니라 오프라인 옥외 광고도 광고 분석 결과를 제공받을 수 있다. 서비스의 가치는 *read*의 의미와 일치한다고 말할 수 있다. 프로젝트의 소스 코드와 개발 진행 상황은 해당 레포지토리([kookmin-sw/capstone-2024-04](https://github.com/kookmin-sw/capstone-2024-04))에서 확인할 수 있다.

1.2 추진 배경 및 필요성

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

1.2.1 서비스 소개

*Do you READ me*는 데이터 분석을 통해 옥외 광고의 효율성을 극대화하는 B2B 서비스이다. 해당 서비스는 디스플레이 앞을 지나가는 유동 인구와 타겟층 정보를 분석하여 광고주가 시각적으로 쉽게 이해할 수 있는 대시보드를 제공한다. 광고주는 대시보드를 통해 사람들의 관심도와 반응을 분석하고 이를 바탕으로 최적의 광고 전략을 수립할 수 있다. 또한 관심을 가진 대상의 나이대, 성비 등과 같은 정보를 제공받아, 이를 활용하여 예상 타겟층과 실제 관심을 가진 대상을 비교함으로써 광고의 적절성을 파악할 수 있다. 서비스는 광고주에게 데이터 기반의 솔루션을 제공함으로써 광고의 효과를 극대화하고 광고 캠페인의 ROI를 향상시키는 데 기여한다.

1.2.2 서비스 목표

*Do you READ me*는 서울 지하철 2호선 신형 열차 출입문 위에 존재하는 광고 디스플레이를 대상으로 광고 효과 분석효과를 제공한다. 디스플레이 앞을 지나가는 보행자들이 해당 광고를 실제로 관심 있게 보는지에 대한 데이터를 수집하고, 광고의 효과를 평가한다. 이를 통해 광고 송출과 동시에 광고 분석이 가능한 시스템을 구축한다. 객체 분석 기술을 활용하여 광고에 노출된 보행자의 데이터를 정확하게 분석하고 시각화하여 대시보드 형태로 제공한다. 이 시스템을 통해 광고주들이 온라인 광고와 유사하게 오프라인 광고 또한 분석할 수 있도록 지원한다. 광고 분석 결과를 통해 광고 집행을 최적화할 수 있도록 돕는다.

1.3 팀 소개

팀 DRM은 ‘시선 추적을 통한 유동인구의 광고 관심도 분석 대시보드’ 프로젝트를 진행하는 팀으로, 백엔드 개발자 2명, AI 전문가 2명, 프론트엔드 엔지니어 1명, 그리고 기획 및 디자이너 1명으로 구성되어 있다. 서로 다른 배경과 기술을 가진 팀원들이 협력하여 프로젝트를 이끌어가고 있으며 새로운 기술과 아이디어를 탐구하며 최선의 해결책을 제시하기 위해 노력하고 있다.

팀의 목표는 협업을 통해 최상의 사용자 경험을 가진 서비스를 제공하는 것이다. 각자가 자신의 분야에서 전문가가 되도록 노력하며, 사용자들에게 가치 있는 서비스를 제공하기 위해 노력하고 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

2 개발 내용 및 결과물

2.1 목표


디지털 사이니지(옥외 광고)에 영상 광고가 송출된다는 것을 가정했다. 디지털 사이니지에서도 영상 광고가 집행되지만 모바일 광고의 특성과 차이가 있다. 모바일에서 집행되는 광고들의 경우, 사용자의 반응, 관심, 이탈 정보를 모두 수집하기에 이 데이터 기반의 광고 집행이 가능하다. 하지만 옥외 디스플레이용 광고의 경우, 유동인구 및 지나가는 사람에 대한 데이터 분석이 없어, 정량적인 광고 집행이 어렵다.

해당 프로젝트는 서울 지하철 2호선 신형 열차 출입문 위에 존재하는 광고 디스플레이 위에 카메라가 부착됐을 때를 가정하며, 카메라가 찍어온 유동 인구의 데이터를 분석해서 다음의 가치 창출을 목표로 했다.

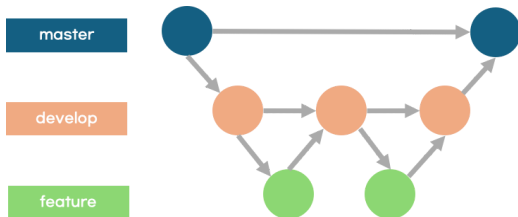
- 1) 디스플레이가 위치한 열차 내에 존재하는 유동 인구의 정보를 수집한다.
- 2) 보행자의 광고에 대한 관심도 정보를 통해 열차 내부 디스플레이 광고의 최적화 의사결정용 데이터 제공한다.

2.2 협업 방법

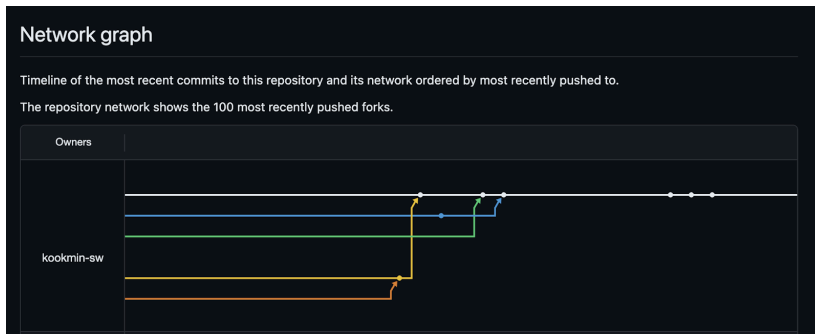
팀원 간 효율적인 협업을 위하여 브랜치 전략으로 GitHub Flow를 개발 방법론으로는 애자일 방법론을 적용하였다. 이러한 접근으로 프로젝트의 유연성을 높이고 빠른 개발 주기를 가지는 것으로 프로젝트의 빠른 완성에 중점을 두었다. 추가로 GitHub Issue Template과 PR Template을 도입하여 작업의 표준화와 효율성을 증진에 노력하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

1. GitHub Flow¹




master 브랜치를 기반으로 새로운 기능이나 수정 사항을 위한 브랜치를 생성하고, 작업 완료 후 master 브랜치로 병합하도록 하였다. DRM의 작업 영역을 크게 client, server, model로 구분됨을 고려하여 각각 별도의 develop 브랜치와 feature 브랜치를 생성하도록 하여 프로젝트의 유지 관리를 간소화하였다.



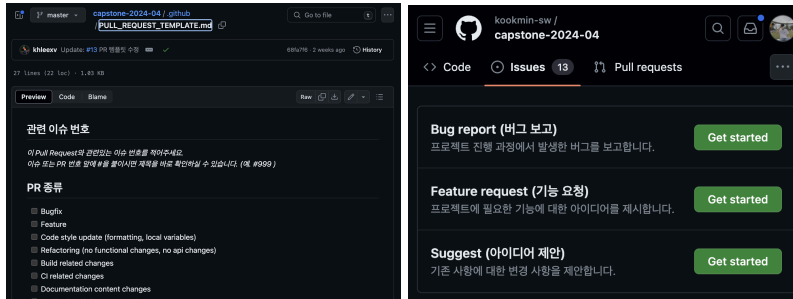
2024.03.30 기준으로 실제 프로젝트에 적용된 모습은 위와 같으며, 저장소 내부 네트워크 그래프를 통하여 확인할 수 있다.

2. 애자일 방법론


세부 기획 및 디자인 등의 변화에 유연하게 대응하고 지속적인 요구사항을 반영할 수 있도록 애자일 방법론을 채택하였다. 프로젝트 내 개발 요구 사항은 하나의 태스크(이슈) 단위로 생성하고 각 팀원에게 할당하여 프로젝트 개발을 진행하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

3. Issue template², PR template



프로젝트 참여 팀원들에게 일관된 형식의 Issue와 Pull request 제공 목적으로 도입하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

2.3 연구/개발 내용 및 결과물

2.3.1 모델

2.3.1.1. 모델 개발에서의 목표

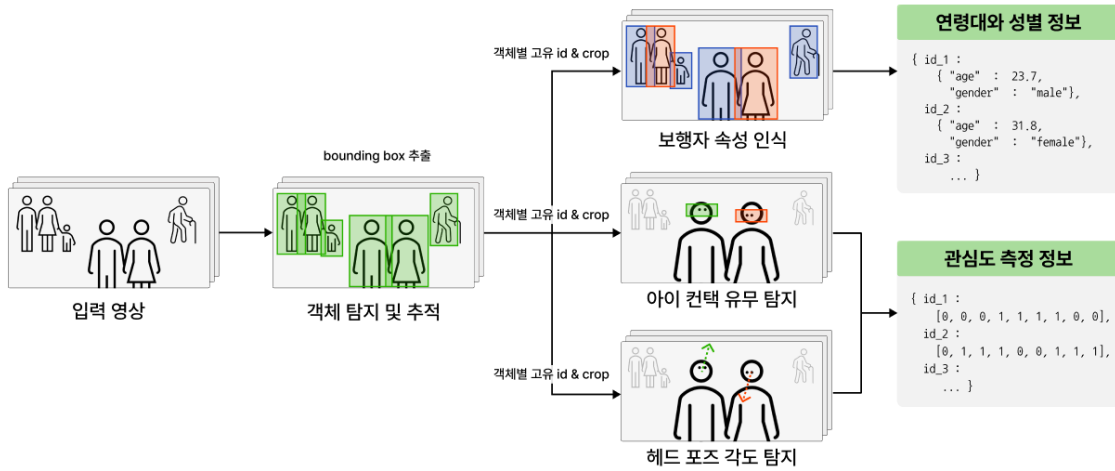
디스플레이에서 송출되는 광고에 대한 사람들의 관심도를 파악하고, 동시에 유동인구의 연령대와 성별 정보를 수집하고자 한다. 이를 위해서 지하철 출입문 상단의 디스플레이 위쪽에 위치한 카메라로 유동인구의 영상을 수집하여 모델의 입력으로 사용하고, 유동인구의 연령대 및 성별 정보와 관심도를 파악하기 위한 모델 파이프라인을 구성하였다.

광고에 대한 지하철 탑승객들의 관심도를 측정하고, 이들의 속성을 파악하기 위해서 크게 세 가지 모델을 사용한다. 첫번째는 영상에 등장하는 유동인구를 추적할 수 있는 객체 추적 기술이다. 두번째는 영상의 프레임에 등장하는 각 사람의 연령대와 성별 정보를 수집하기 위한 보행자 속성 인식 모델이다. 마지막으로 지하철 탑승객들이 실제로 디스플레이를 응시하고 있는지를 판단하기 위해서 아이 컨택 모델과 헤드 포즈 모델을 사용하고자 한다.

각 모델을 거치며 수집된 데이터는 사람 단위, 즉 지하철 탑승객 각각을 단위로 가공되며 최종적으로는 시각적으로 쉽게 이해할 수 있는 대시보드 형태로 광고주에게 제공된다. 이를 통해 광고주들은 지하철 내부의 광고에 노출되는 유동인구에 대한 정보와 이들의 관심도를 수치적으로 파악하여 효과적으로 광고 전략을 수립함으로써 광고 집행을 위한 예산을 효율적으로 운용할 수 있다.

2.3.1.2. 모델 워크플로우

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05




디스플레이 상단에 위치한 카메라로부터 입력 영상이 들어오면, 먼저 사람 객체를 인식하여 bounding box를 추출한다. 영상에서 각각의 사람들을 시간에 따라 추적하고, 각 객체에 고유한 id를 부여하기 위하여 객체 추적 알고리즘을 사용한다. 이 단계에서 인식된 객체의 영역을 자르고, 잘라낸 이미지는 이후 모델들의 입력으로 사용된다.

먼저 객체의 속성을 인식하고 분류하기 위해서 보행자 속성 인식 모델을 사용한다. 이 단계에서는 유동인구의 연령대와 성별을 추론한다. 다음으로는 객체의 관심도 측정 정보를 수집하기 위하여 아이 컨택 모델과 헤드 포즈 모델을 함께 사용한다. 우선 아이 컨택 모델을 사용하여 사람의 자세가 정면을 향하고 있는지를 추론한다. 하지만 자세가 정면을 향하고 있더라도 머리의 방향이 아래를 향하고 있는 경우에는 지하철 출입문 상단의 디스플레이를 응시했다고 간주하기 어렵다. 따라서 헤드포즈 모델을 함께 사용하여 관심도 유무를 측정하였다. 이를 위해, 객체의 영역에서 얼굴을 인식하는 모델을 사용하여 얼굴에 해당하는 bounding box를 추출하고, 헤드 포즈 모델의 입력으로 사용한다. 해당 모델에서는 객체의 헤드 포즈 각도를 추론하며, 디스플레이가 출입문 상단에 위치한 만큼, 탑승객이 고개를 올려 디스플레이를 응시했는지를 추론하여 관심도 유무를 판단하고자 한다.

2.3.1.3. 모델 상세 내용 (이름, 변경 또는 개선사항)

1. 객체 추적 모델

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

- 목적

지하철 출입문 상단에 위치한 디스플레이 근처의 각 유동인구를 추적하여, 각 사람의 연령대와 성별 정보를 수집하고 관심도 유무를 판단하고자 한다. 이 정보는 디스플레이에서 송출되는 광고가 얼마나 효과적으로 유동인구의 관심을 끄는지 평가하는 데 핵심적인 기능이 된다. 또한 광고가 타겟 층에게 얼마나 많이 노출되었는지와 타겟층 중 광고에 실제로 관심을 보인 사람들의 비율을 계산하는 데 중요한 역할을 한다. 해당 모델에서의 입력은 디스플레이 상단의 카메라로 촬영한 유동인구의 영상이며, 출력은 탐지한 사람의 프레임 별 위치 정보와 각 객체에 할당된 고유 id이다.


- 세부 설명

- 사용 모델 : 탐지기(yolov8n-pose), 추적기(OCSORT)
- **BoxMOT - 알고리즘이라고 해야 하는가,,**
BoxMOT에 대한 간략한 설명 - box...
- 데이터 처리 : 입력된 영상에서 객체를 감지하고, 각 객체에 고유한 id를 할당한 후, 이들의 움직임을 추적한다. 이 때, 다음 프레임에서 같은 사람이 등장하는 경우 동일한 id를 부여하는 재식별 기능을 사용하여 각 사람에 대한 정확한 정보를 수집하는 동시에, 객체 탐지의 정확도를 높이고자 한다.
- 객체 탐지 모델은 추후 아이콘택 모델에 활용하기 위해 단순 bounding box 탐지 모델이 아닌 키포인트 모델 yolov8n-pose를 사용했으며, 객체 추적 모델은 자체적인 평가를 통해 가장 높은 성능을 달성한 OCSORT를 사용했다.
- 객체 추적 모델에서 탐지된 사람들 중 일정 시간(20 프레임) 이상을 체류한 사람들을 선별하여 해당 사람들을 박스 형태로 잘라낸다. 이 정보는 보행자 속성 모델 및 아이 컨택 모델, 헤드 포즈 모델과 결합하여 관심도를 측정하는 데 사용된다.

2. 보행자 속성 인식 모델

- 목적

각 사람들의 연령대와 성별을 식별하여 광고주가 타겟 층에게 더욱 적합한 광고를 제공할 수 있게 돕는다. 이 정보는 타겟층에 대한 정보를 수치적으로 확인하여, 광고를 송출할 시간대에 대한 전략을 세우거나, 더 나아가 광고를 송출하고자 하는 지하철 호선을 선정하는 데 도움을 줄 수 있다. 해당 모델에서의 입력은

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

객체 추적 모델에서 잘라낸 박스 형태의 이미지와 고유 id 값이며, 출력은 유동인구의 연령대와 성별의 보행자 속성 정보이다.


- 세부 설명
- 사용 모델 : MiVOLO
- 보행자 속성 인식 데이터셋인 IMDB-Clean에서 사전 학습된 VOLO 기반의 모델을 사용하고 있다. 해당 모델은 보행자의 외형 이미지를 기반으로 성별과 나이대 정보를 추론할 수 있다. 유동인구의 나이대는 실수 형태로 추론한 뒤 10대 단위로 가공하여 제공하며, 유동인구의 성별은 남성 또는 여성으로 추론한다.

3. 관심도 측정 모델

- 목적

지하철 호차 내부 광고에 대한 실제 유동인구의 관심도를 측정하고자 한다. 관심도 측정을 위해서는 두 가지 모델을 사용한다. 첫번째로 아이 컨택 모델을 사용하여 화면 응시 여부를 추론하며, 두번째로 헤드 포즈 모델을 사용하여 지하철 탑승객이 고개를 들어올려 출입문 상단의 광고를 응시했는지를 함께 추론하고자 한다. 해당 모델에서의 입력은 객체 추적 모델에서 잘라낸 한 사람의 박스 형태의 이미지이다. 아이컨택 모델의 출력은 화면을 응시했다고 추론하는 확률값이며, 헤드 포즈 모델의 출력은 머리를 위아래로 움직이는 각도이다.

- 세부설명
- 사용 모델 : 1. 아이 컨택 모델 : **LOOK(정확한 명칭 확인 필요)** 2. 헤드 포즈 모델: 6DRepNet
- 아이 컨택 모델은 사람이 카메라를 응시하는지 여부를 측정한다. 인체의 특정 부위의 좌표를 나타내는 17개의 keypoint를 기반으로 화면 응시 여부를 측정하며, 자세가 정면을 향하는 경우 사람의 머리가 아래를 향하고 있더라도 화면을 응시한다고 추론하는 한계가 있다. 직접 수집한 데이터셋을 이용하여 아이 컨택 모델 추론 결과와 데이터셋 영상 프레임 이미지를 시각화하여, 화면을 응시했다고 추론하는 확률이 0.5 이상인 경우에 화면을 응시했다고 판단하였다.
- 헤드 포즈 모델은 사람의 머리가 어떤 3D 자세를 가지고 있는지를 측정하여, 머리의 3D 좌표와 회전 정보를 바탕으로 사람이 광고 디스플레이에 관심을 가지고 있는지 판단할 수 있다. 광고 디스플레이가 지하철 출입문 상단에 위치하기 때문에, 헤드 포즈 모델에서 제공하는 세가지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5


각도값 중, 머리 앞 뒤 기울기를 나타내는 Pitch 값을 사용하였다. 이는 고개를 들거나 숙이는 동작을 반영하며, 고개를 위로 들 때 양수 값을 가지고 고개를 아래로 숙일 때 음수 값을 가진다. 직접 수집한 데이터셋을 이용하여 헤드 포즈 모델 추론 결과와 데이터셋 영상 프레임 이미지를 시각화하여, 카메라가 위쪽에 위치해 있는 경우에 Pitch 값이 -20보다 큰 경우에 위를 올려다 본 것으로 판단하였다.

2.3.1.4. 데이터 수집 및 라벨링

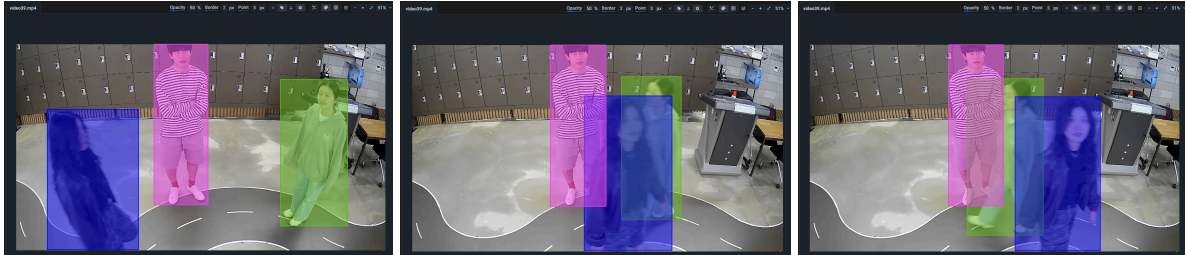
미래관 신관 4층에 모니터와 웹캠을 설치하여 시선 추적 데이터를 수집하는 것을 계획했으나 학교 총무팀에 문의한 결과 웹캠을 설치하는 것은 매우 긴 절차가 필요했다. 총무팀은 웹캠 설치를 위해 다음과 같은 절차를 제안했다. 먼저, 설치 계획을 이메일로 제출해야 하며 이때 포함되어야 할 내용은 프로젝트 목적, 설치 목적, 수집하는 데이터 내용, 해당 데이터 활용 방안, 그리고 설치할 기간과 시간대이다. 이후 학교 구성원들에게 2주 이상 동안 설문조사를 실시하여 설치에 대한 의견을 수렴해야 했다. 설문 결과 설치에 반대 의견이 없을 경우, 통행이 방해되지 않는 선에서 설치를 진행할 수 있었다.

따라서 불특정 다수에 대해 실험을 진행하는 대신 데이터 수집 및 활용에 동의하는 학우들을 모아 촬영을 진행하기로 했다. 실험 진행은 5월 9일, 5월 10일 총 2일간 미래관 4층 복도 혹은 자율주행스튜디오에서 소프트웨어학부생 대상으로 데이터셋 수집을 진행했다. 구두로 참가 의사를 밝힌 학우에게 개인정보 수집 목적, 수집하는 정보, 활용 범위, 데이터 폐기 일자를 충분히 설명 후 동의 서명을 받아 촬영을 진행했다. 총 38명의 학우가 해당 실험에 참여하였다. 총 50여개의 영상을 촬영하였으며, 그 중 41개의 영상을 모델 성능 평가 및 학습에 사용하였다.

수집한 데이터에 대하여 객체 추적 데이터셋, 그리고 보행자 속성 인식과 관심도 측정 데이터셋을 별개로 구축하였다. 라벨링은 팀원들이 역할을 분담하여 진행하였으며, 라벨링용 구글 계정을 생성하여 라벨링 진행 상황을 슬랙을 통해 공유하며 진행하였다. 우선, 객체 추적 데이터셋은 Supervisely라는 도구를 사용하여 라벨링을 진행하였다. Supervisely의 경우 20프레임 단위로 객체를 자동으로 추적하여 라벨링해주는 기능을 제공하기 때문에, 해당 툴로 객체 추적 데이터셋을 구축하였다. 각 영상에 대해서 5fps 단위로 객체의 id를 라벨링하였으며, 각 사람에 대해서 프레임이 연속되도록 라벨링을 진행하였다. 이렇게 구축한 데이터셋을 사용하여 객체 추적 성능을 평가하였으며, 평가 지표와 평가 결과에 대한 수치는

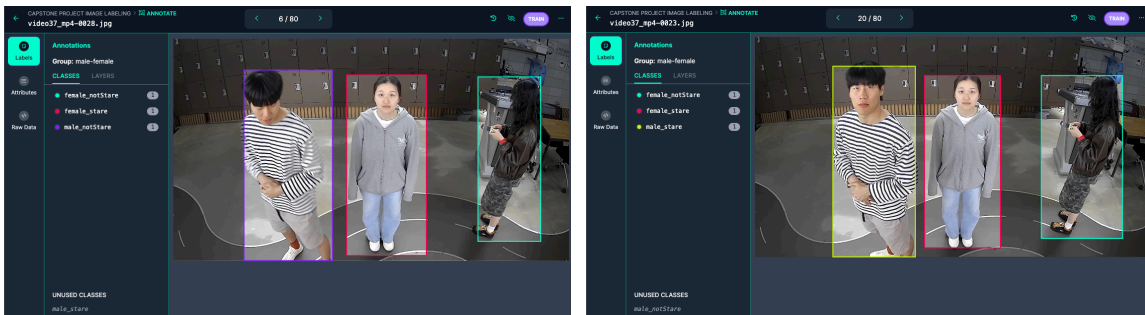
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

아래에서 확인할 수 있다.




Supervisely에서 라벨링을 진행한 화면

보행자 속성 인식과 관심도 측정 데이터셋은 Roboflow라는 도구를 사용하여 라벨링을 진행하였다. Roboflow에 mp4 영상을 그대로 업로드하고 Roboflow의 자체적인 툴로 영상을 5fps 단위로 자르는 방식을 사용하였다. 참여자들의 나이대가 전부 20대인 관계로, 나이대 라벨링은 생략하였으며, 성별과 화면 응시 여부를 라벨링 하기 위해 male_stare, male_notStare, female_stare, female_notStare 총 4개의 클래스를 구성하였다. 이렇게 구축한 데이터셋을 사용하여 보행자 속성 인식 모델과 관심도 측정 모델의 성능을 평가하였으며, 평가 지표와 평가 결과에 대한 수치는 아래에서 확인할 수 있다.



Roboflow에서 라벨링을 진행한 화면

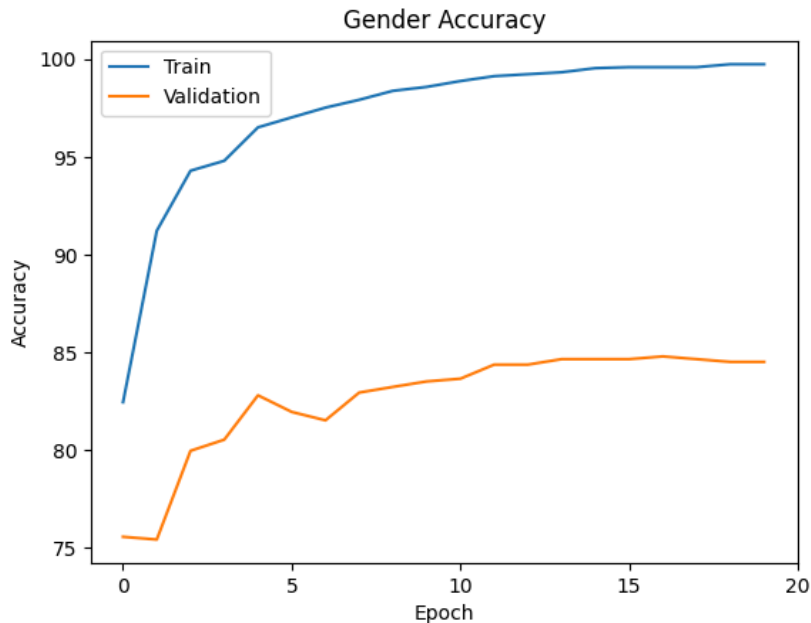
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5


2.3.1.5. 보행자 속성 인식 모델 성능 측정 결과

중간 발표 이전에 사용하던 ResNet 계열의 보행자 속성 인식 모델은 모델이 학습된 데이터셋의 크기가 작고, 모델의 크기 또한 작아서 성별, 연령대 정확도 모두 저조한 성능을 보여주었다. 또한 우리 프로젝트에 불필요한 복장 착용, 헤어스타일 등의 정보를 추론하고 있었다. 우리 프로젝트에 더 적합한 모델을 추가적으로 조사했고, 조사 결과 MiVOLO 모델을 사용하기로 결정했다. MiVOLO 모델은 프로젝트에 필요한 정보인 성별, 연령대 정보만을 출력한다.

직접 수집한 데이터셋에서 사전 학습된 MiVOLO 모델을 평가했을 때 성별 정확도는 70%였다. 벤치마크 성능에 비해 많이 낮은 성능을 보여주는데, 이는 카메라가 높은 위치에 설치되어 있다는 특수성과 모델이 한국인 데이터로 학습되지 않았기 때문인 것으로 보인다.

성능 향상을 위해 직접 수집한 데이터셋을 학습용 70%, 검증용 30%로 분할한 뒤 추가 학습을 진행했고, 검증 성능 85%를 달성했다. 더 많은 데이터를 확보한다면 추가적인 성능 향상을 달성할 수 있을 것으로 예상된다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

2.3.1.6. 관심도 측정 모델 성능 측정 결과

초기에는 일반적인 옥외광고 상황에 사용될 수 있는 아이컨택 모델만을 사용했다. 아이컨택 모델은 사람 외형 이미지가 아닌 키포인트 정보를 사용하여 관심 여부를 판단하기 때문에, 모델의 크기가 작다는 장점이 있다. 시선을 보낸 것이 광고에 대한 관심을 보였다는 게 아니라는 문제점을 해결하기 위해 추후 프로젝트 기획이 바뀌었고, 카메라가 높은 위치에 설치되는 상황으로 바뀌었다. 바뀐 상황에서 아이컨택 모델의 성능이 떨어졌는데, 몸이 카메라를 향해있지만, 고개를 숙인 경우를 잘 처리하지 못하는 것이 원인으로 추정된다.


관심도 측정 정확도를 향상시키기 위해 헤드포즈 추정 모델을 추가적으로 사용했다. 헤드포즈 정보를 통해 고개를 든 것을 감지했고, 아이 컨택 모델과 고개를 든 것 모두 감지된 경우 관심을 보였다고 판단했을 때, 아이 컨택 모델 단독 사용에 비해 10%의 정확도 향상이 있었다.

2.3.1.7. 객체 추적 성능 측정 결과

기존에 공개되어 있는 트래커들은 벤치마크 데이터셋에서 최고 성능을 달성하도록 학습 및 튜닝되었기 때문에, 우리 프로젝트 데이터와 특성이 다르다. 지하철이라는 좁은 공간에서 수집되는 영상 데이터 특성상, 카메라와 사람이 3m 이내 근접한 상황으로 제한되는데, 이로 인해 사람이 영상에서 큰 크기로 등장하고, 작은 움직임에도 박스의 크기와 변화도가 큰 편이다. 이 외에도 지하철 내부라는 협소한 공간 특성상 사람들이 겹치는 등 트래킹에 도전적인 상황이 많이 발생한다는 특징이 있다.

프로젝트 상황에 적합한 트래커를 선정하기 위해 오픈소스로 공개된 트래커 6종을 선정하여 별도의 튜닝 없이 평가했을때 다음과 같은 성능을 얻었다. 벤치마크 데이터셋에서의 성능과 다르게 트래커별로 성능에 상당한 차이가 있음을 확인할 수 있다. 이후 파라미터 튜닝을 통해 +2% 정도의 추가적인 성능 향상을 이뤘다.

Tracker	HOTA ↑	MOTA ↑	IDF1 ↑
ByteTrack	62.82	62.16	72.68
OCSORT	68.89	71.07	79.67
StrongSORT	53.62	-7	59.48


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

DeepOCSORT	66.22	70.36	76.14
BoTSORT	65.92	64.35	71.95
HybridSORT	68.72	70.31	79.57

2.3.2 유저 인터페이스

2.3.2.1. Figma Styles 지정

유저 인터페이스는 사용자 경험을 최우선으로 하여 작성되었다. 캡스톤 디자인을 위해 수행한 모든 디자인 작업은 Figma를 통해 작성했다. 디자인 일관성 유지 및 개발의 편의성을 위해 Text styles 와 Color styles 를 지정하여 작성하였다. Text styles는 다양한 텍스트 요소들을 일관되게 사용하기 위해 정의되었다. 이를 통해 각 텍스트의 크기, 두께, 행간 등을 미리 설정하여 디자인 요소들 간의 통일성을 유지할 수 있었다. 예를 들어, 제목, 본문, 캡션 등 다양한 텍스트 스타일을 구분하여 사용함으로써 UI의 가독성을 높이고, 사용자가 각 요소의 중요도를 쉽게 파악할 수 있도록 했다. 또한, 이러한 텍스트 스타일의 정의는 프론트 개발에서의 효율성을 높일 수 있었다. Color styles 또한 중요한 요소로, 전체 디자인의 색상 팔레트를 미리 정의하여 일관된 사용자 경험을 제공할 수 있도록 했다. 주요 색상, 보조 색상, 강조 색상 등을 명확히 구분하여 사용함으로써 UI 디자인의 통일성을 유지하였다. 예를 들어, 브랜드의 대표 색상을 일관되게 사용하여, 사용자들이 주요 인터페이스 요소를 직관적으로 인식할 수 있도록 하였다. 이와 같은 색상 스타일의 지정은 디자인 시 일관성을 유지할 뿐만 아니라, 프론트 개발에서 색상 코드를 재사용할 수 있게 하여 개발 작업의 효율성을 높였다. Figma를 사용하여 작성된 이번 서비스 UI 디자인은 Text styles와 Color styles를 체계적으로 지정함으로써 디자인 일관성을 유지하기위해 노력했다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

Text styles		Color styles
Pretendard/Normal/18	Pretendard/Normal/14	● Main
Ag 400-Regular · 18/Auto	Ag 400-Regular · 14/Auto	○ White
Ag 500-Medium · 18/Auto	Ag 500-Medium · 14/Auto	● White Sub
Ag 600-SemiBold · 18/Auto		● Black Sub
Pretendard/Normal/16	Pretendard/Normal/12	● Black
Ag 400-Regular · 16/Auto	Ag 400-Regular · 12/Auto	● Red
Ag 500-Medium · 16/Auto	Ag 500-Medium · 12/Auto	● Gray Btn
Ag 600-SemiBold · 16/Auto		● Gray Line

Figma Text styles 와 Color Styles

```

src > client > JS tailwind.config.js > ...
You, 지남달 | 1 author (You)
1  /** @type {import('tailwindcss').Config} */
2  export default {
3    content: ["/index.html", "./src/**/*.{js,ts,jsx,tsx}"],
4    theme: {
5      extend: {
6        colors: {
7          main: "#4200ff",
8          white: "#ffffff",
9          white_sub: "#a9c3eb",
10         black: "#191919",
11         black_sub: "#54657b",
12         red: "#ff0000",
13         placeholder: "#6b6b6b",
14         gray1: "#f0f0f0",
15         gray2: "#d9d9d9",
16       },
17     },
18   },
19   plugins: [],
20 };
21


```

Tailwind의 테마 기능

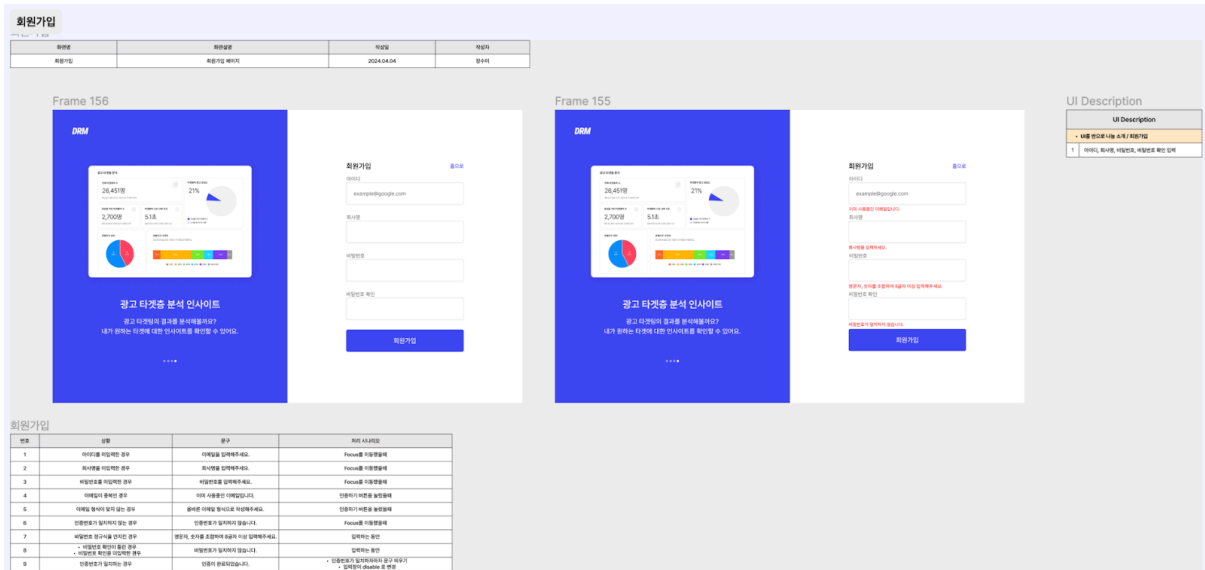
2.3.2.2. UI Description 작성

프론트엔드 개발 및 서버 개발팀과의 원활한 소통을 위해, 모든 페이지에 대한 UI Description를 작성하였다. 각 페이지의 디자인이 완료된 후, UI Description를 상세히 작성하여 개발 과정에서 참고할 수 있도록 하였다. 이를 통해 개발자들이 디자인 의도를 명확히 이해하고, 일관성 있는 구현이 가능하도록 하였다.

또한, Figma Comments 기능을 활용하여 개발자들이 추가적인 질문을 하거나 변경된 사항을 반영할 수

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

있도록 하였다. 실시간으로 피드백을 주고받으며, 변경 사항이 즉각적으로 문서에 반영될 수 있도록 하였다. 이러한 과정을 통해, 프론트엔드 개발과 서버 개발 간의 협업을 더욱 효율적으로 진행할 수 있었다.



Figma UI Description

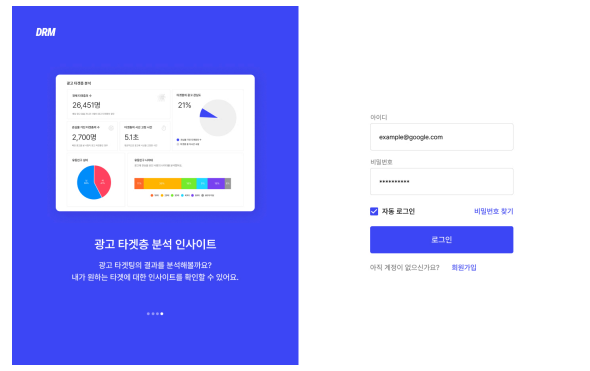
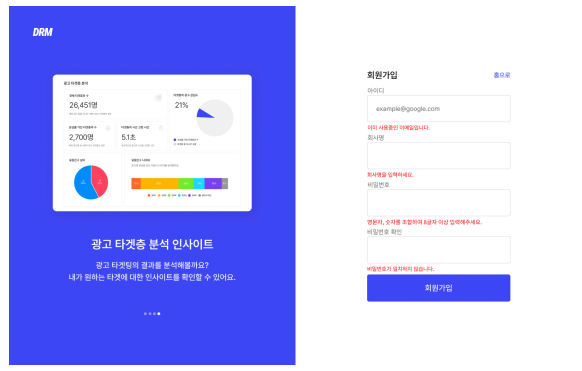
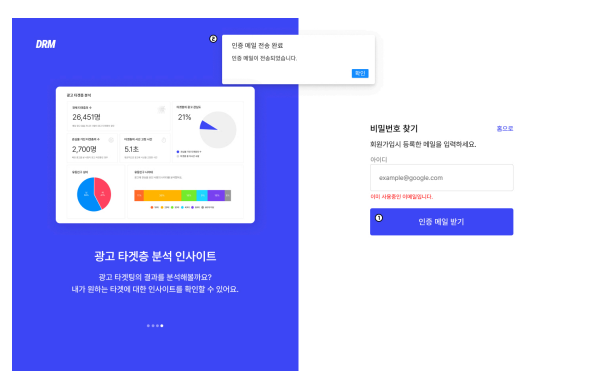
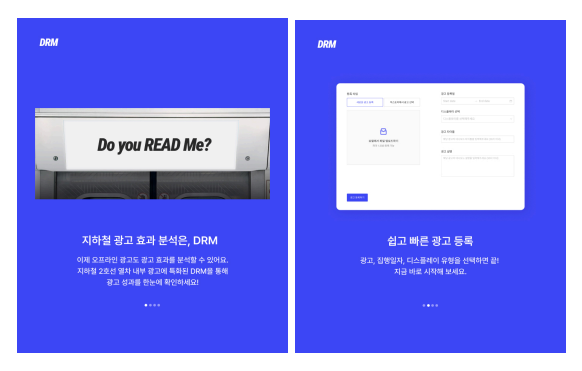
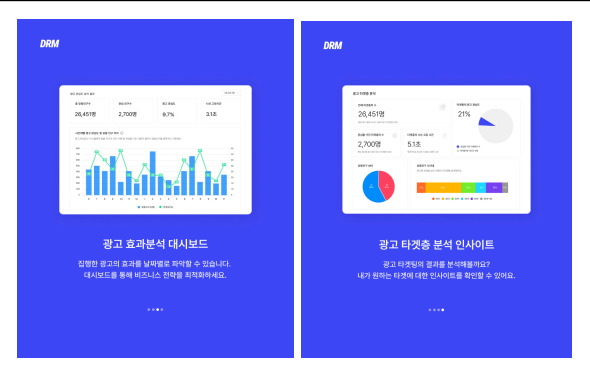
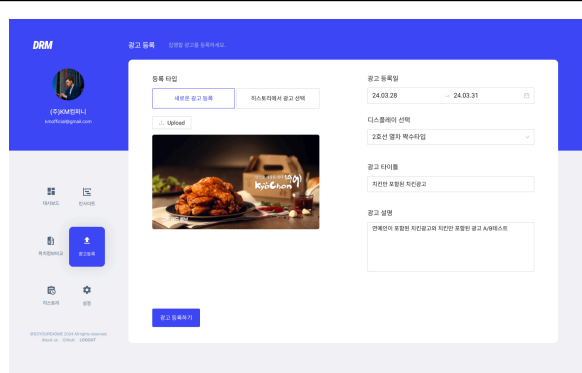
2.3.2.3. UXUI 디자인

전체 UI 디자인 작업을 진행하면서 고객 페이지와 관리자 페이지를 명확하게 구분하여 디자인하였다. 고객 페이지는 사용자 경험을 최적화하는 데 중점을 두고 설계되었으며, 직관적이고 접근성이 높은 인터페이스를 제공하도록 하였다. 고객들이 쉽게 서비스 이용에 적응할 수 있으면서도 필요한 정보를 신속하게 찾을 수 있도록 서비스 플로우를 간단하게 설계하였다.

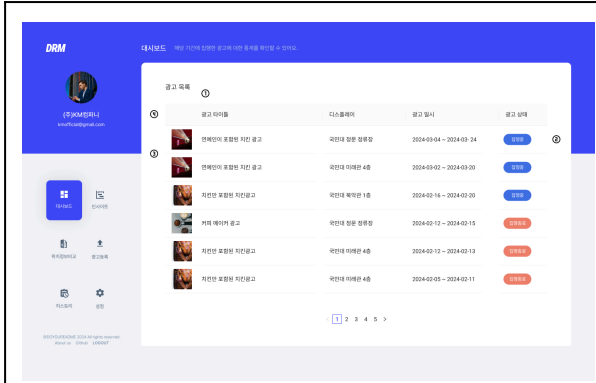
관리자 페이지의 경우 효율적인 관리와 운영을 할 수 있도록 필수적인 기능만을 설계하였다. 관리자가 다양한 데이터를 한눈에 파악할 수 있도록 전체 고객에 대한 광고목록을 볼 수 있도록 구성하였다. 관리자 페이지 인터페이스 제공을 통해 관리자는 직접 데이터베이스에 접근하는 위험 없이 업무를 수행할 수 있으며 시스템 운영에 필요한 모든 기능을 직관적으로 접근할 수 있다.

모든 페이지에 대해 각각의 용도와 사용자 그룹의 특성을 고려하여 디자인하였다. 디자인 과정에서 개발팀의 지속적인 피드백을 받아 수정 과정을 거쳤다. 최종 결과물이 사용자 요구사항을 충족하고, 시스템 운영에 최적화될 수 있도록 하였다.

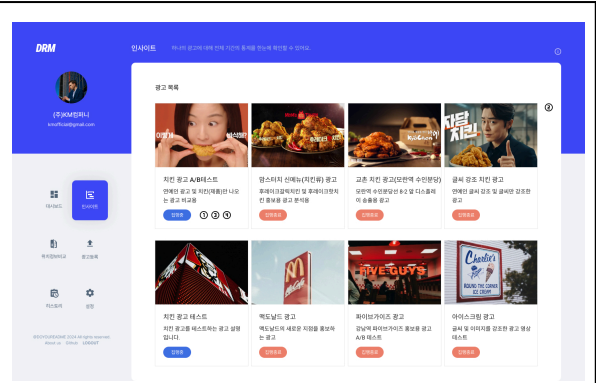
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

	
로그인	회원가입
	
비밀번호 찾기	서비스 소개
	
서비스 소개	광고등록

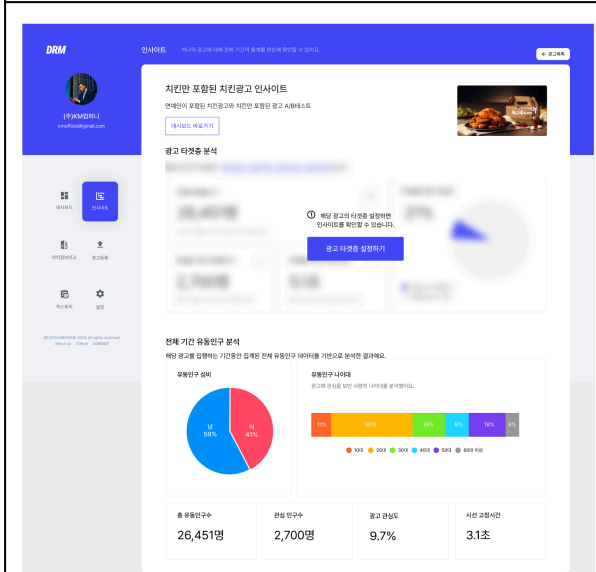
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05



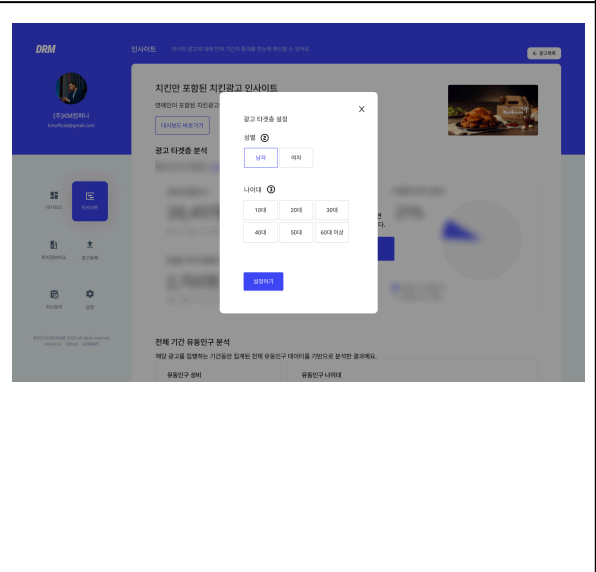
대시보드 - 광고목록




인사이트 - 광고목록

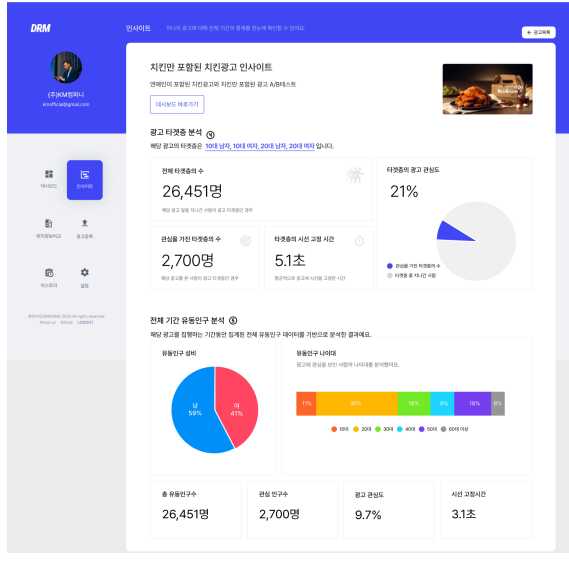


인사이트 - 광고 타겟 설정 전



인사이트 - 광고 타겟 설정 모달

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05



지하철 포함된 지킨광고 인사이트

광고 타겟 분석

전체 타겟층의 수: 26,451명

타겟층의 광고 관심도: 21%

관심도 기반 타겟층의 수: 2,700명

타겟층의 시연 구성 시간: 5.1초

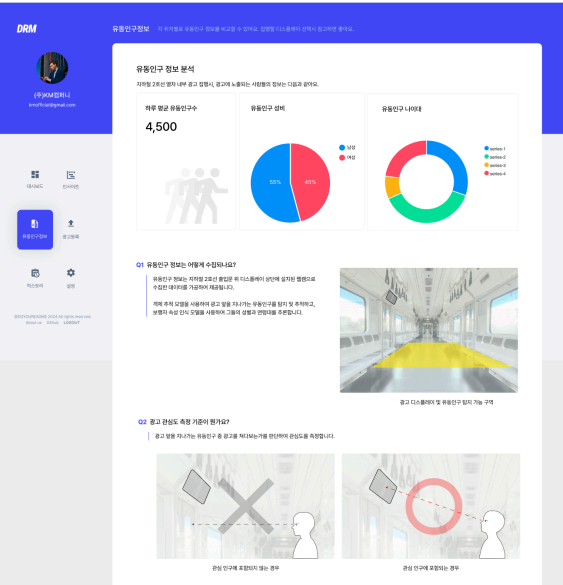
전체 기간 유동인구 분석

유동인구 성별: 남 54%, 여 46%

유동인구 나이에 따른 관심도 분포: 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%

총 유동인구 수: 26,451명 | 관심 인구 수: 2,700명 | 광고 관심도: 9.7% | 시연 구성시간: 3.1초

인사이트 - 광고 타겟 설정 후



유동인구 정보 분석

총 유동인구 수: 4,500

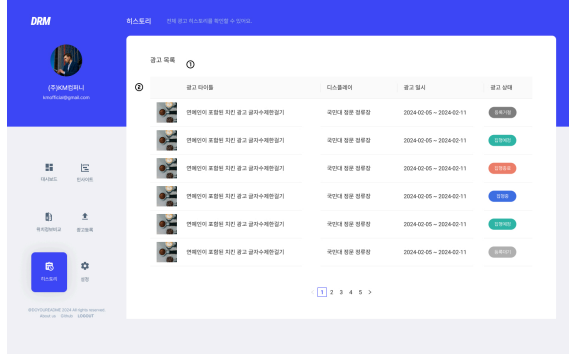
유동인구 성별: 남 50%, 여 50%

유동인구 나이에 따른 관심도 분포: 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%

01 유동인구 정보는 어떻게 수집되나요?

02 광고 관심도 측정 기준이 변경될까?

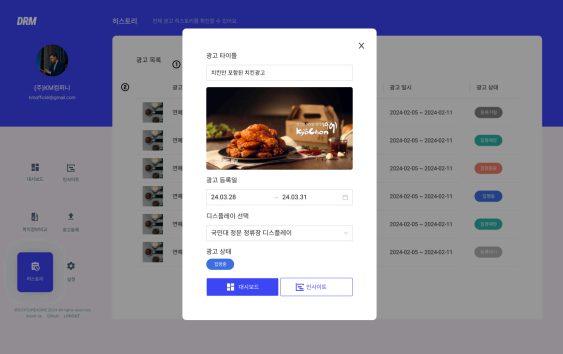
유동인구정보



광고 목록

광고 이름	디스플레이	광고 일시	광고 상태
전체인구 포함된 지킨 광고 공개수행중입니다	국민대 홍보 영상	2024-02-05 ~ 2024-02-11	실행중
전체인구 포함된 지킨 광고 공개수행중입니다	국민대 홍보 영상	2024-02-05 ~ 2024-02-11	실행중
전체인구 포함된 지킨 광고 공개수행중입니다	국민대 홍보 영상	2024-02-05 ~ 2024-02-11	실행중
전체인구 포함된 지킨 광고 공개수행중입니다	국민대 홍보 영상	2024-02-05 ~ 2024-02-11	실행중
전체인구 포함된 지킨 광고 공개수행중입니다	국민대 홍보 영상	2024-02-05 ~ 2024-02-11	실행중
전체인구 포함된 지킨 광고 공개수행중입니다	국민대 홍보 영상	2024-02-05 ~ 2024-02-11	실행중
전체인구 포함된 지킨 광고 공개수행중입니다	국민대 홍보 영상	2024-02-05 ~ 2024-02-11	실행중

히스토리 - 광고목록



광고 디테일

광고 이름: 지킨 포함된 지킨광고

광고 일시: 2024-02-05 ~ 2024-02-11

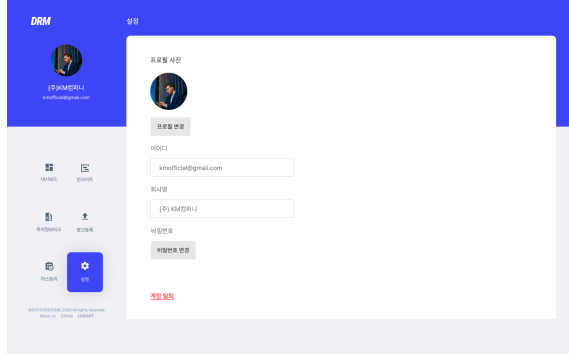
광고 상태: 실행중

광고 유형: 24.03.28 - 24.03.31

디스플레이 선택: 국민대 홍보 영상 디스플레이

광고 상태: 실행중

히스토리 - 광고 세부사항 모달



유저 관리

프로필 변경

이름: [이름 입력]

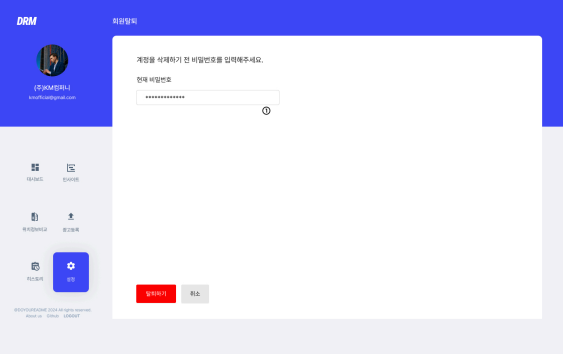
이메일: kmooc@kmu.ac.kr

비밀번호 변경

비밀번호: [비밀번호 입력]

비밀번호 확인: [비밀번호 입력]

저장



유저 관리

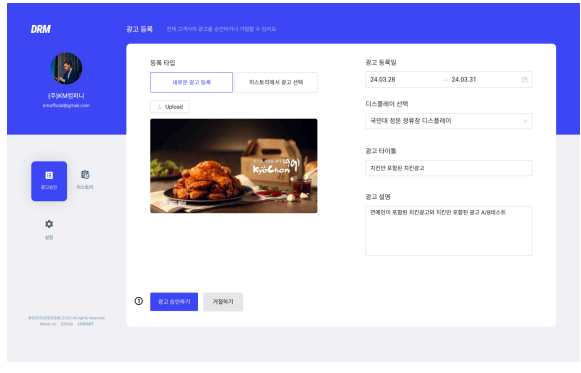
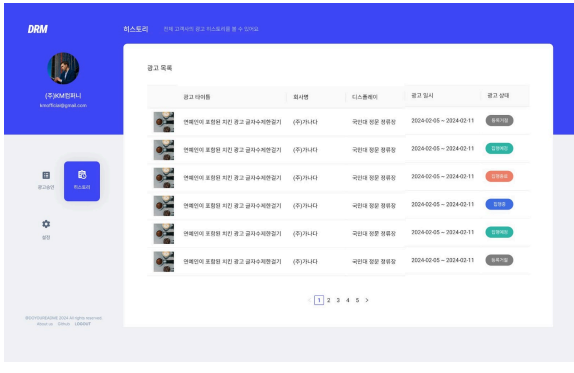
계정을 삭제하기 전 비밀번호를 입력해주세요.


현재 비밀번호: [비밀번호 입력]

비밀번호 확인: [비밀번호 입력]

삭제

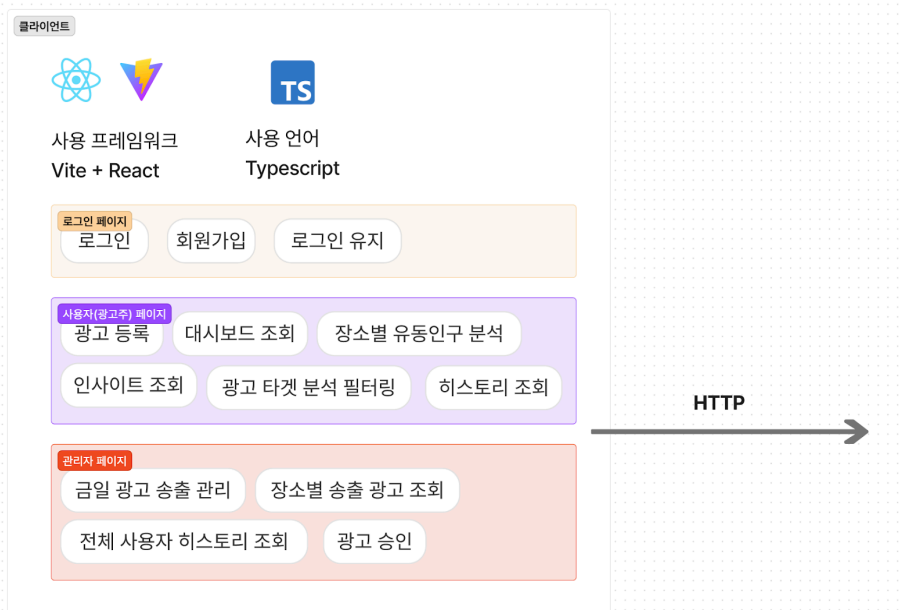
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

설정	탈퇴
	
관리자 - 광고 승인	관리자 - 광고 히스토리

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

2.3.3 클라이언트

2.3.3.1. 클라이언트 아키텍처 및 사용 기술 스택 선정



본 프로젝트는 광고주에게 대시보드 형태의 솔루션을 제공하는 것이 주된 목표이다. 대시보드는 다량의 데이터를 처리하고 분석 결과를 시각적으로 표현하여 사용자에게 정보를 제공하는 도구이며 이를 개발하기 위하여 다양한 기술 스택을 검토한 결과, 웹 형태로 서비스를 제공하는 것이 가장 적합하다고 판단하였다.


프론트엔드 프레임워크로는 Vite와 React를 사용하였다. React는 컴포넌트 기반 아키텍처를 통하여 사용자 인터페이스(UI)를 구현하는 데 있어 코드의 재사용성과 유지보수성을 높일 수 있었고 대시보드와 같은 복잡한 인터페이스에 적합하였다. Vite를 채택하는 것으로 기존 React보다 빠른 개발 환경을 확보하고 프로덕션 환경에서 최적화된 빌드를 제공할 수 있는 몇 가지 이유는 아래와 같다.

1. 초기 번들링 없이 빠른 개발 서버

Vite는 기존 번들러와 달리 애플리케이션을 번들링하지 않고 개발 서버를 시작한다. ES모듈을 사용하여 필요 모듈만 동적으로 불러오기 때문에 초기 시작 시간을 대폭 단축한다.

2. Hot Module Replacement

HMR을 더 효율적으로 반영하여 변경 사항을 즉시 반영한다. Vite는 모듈 간 의존성을 추적하고 변경된 모듈에 대해서만 리로드를 진행하기 때문에 빠른 피드백 루프를 가진다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

3. 최신 브라우저 기능 활용

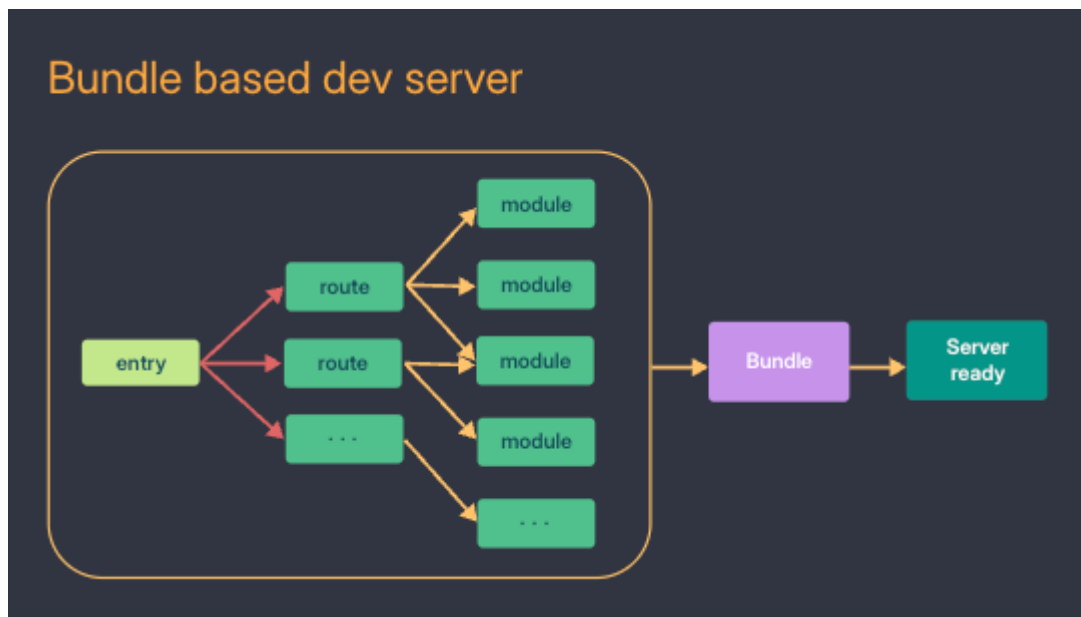
최신 브라우저 기능인 ES모듈을 활용하여 번들링 없이 개발 서버를 실행한다. 불필요한 번들링 작업 생략을 통하여 개발 속도를 향상시킬 수 있다.

4. 빌드 성능 최적화

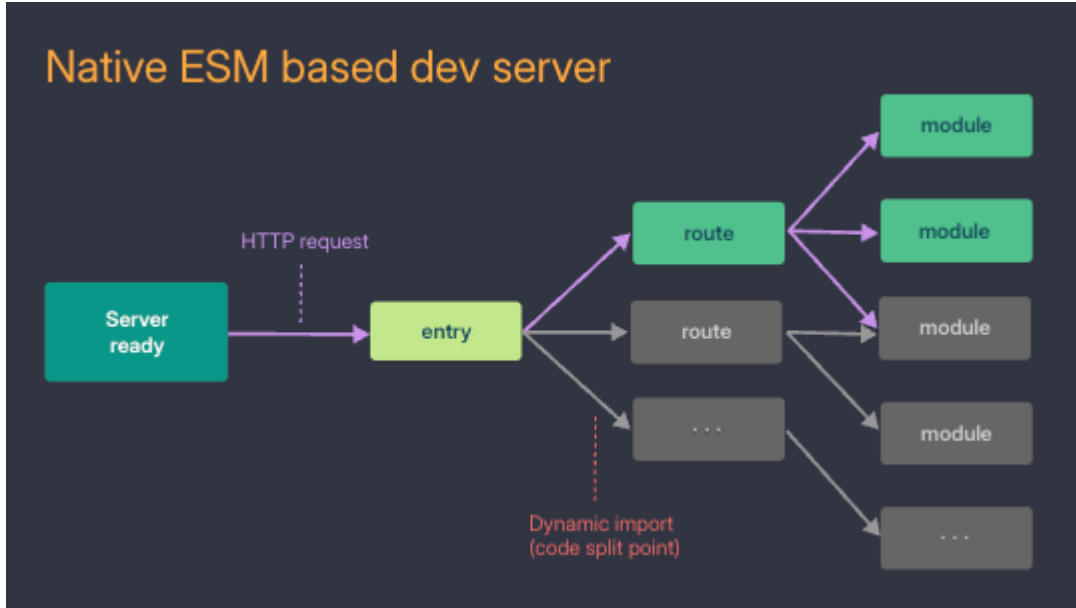
빌드 단계에서 Rollup을 사용하여 최적화된 번들링을 수행한다. 이를 통하여 개발 환경에서는 빠른 속도의 빌드를 제공하며 프로덕션 환경에서는 최적화된 빌드를 제공한다.

5. 코드 분할 및 지연 로딩

Vite는 코드 분할을 기본적으로 지원하며 필요에 따라 지연 로딩을 설정할 수 있어 개발 중에도 빠른 응답성을 유지할 수 있도록 도와준다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05



2.3.3.2. 기본 컴포넌트 및 대시보드 UI 구현

클라이언트의 주 목표는 DRM 서비스를 이용하는 사용자(광고주)에게 유의미한 정보, 분석 결과를 주는 것이다. 모델에서 수치화한 데이터를 서버에서 1차 가공하여 출력해준 값을 클라이언트의 새로운 컴포넌트(혹은 위젯)의 입력으로 주어야 한다.

개발 단계에서의 시간을 줄이고 일관된 UI를 제공하기 위하여 기본 UI 구성에 있어 Ant Design 라이브러리를 사용하였고 대시보드 내부 차트 구현에는 Apex Charts 라이브러리를 사용하였다.

2.3.3.3. 프로젝트 내부 라우터 및 상태(State) 관리

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

```
function App() {
  return (
    <QueryClientProvider client={queryClient} contextSharing={true}>
      <BrowserRouter>
        <Routes>
          <Route path="/" element={<MainPage />} />
          <Route
            path="/home"
            element={
              <ProtectedRoute isAuthenticated={isAuthenticated()}>
                <HomePage />
              </ProtectedRoute>
            }
          />
          <Route
            path="/admin"
            element={
              <ProtectedRoute isAuthenticated={isAdmin()}>
                <AdminPage />
              </ProtectedRoute>
            }
          />
          <Route path="/playlist/:locationId" element={<PlayListScreen />} />
          <Route path="*" element={<Navigate to="/" />} />
        </Routes>
      </BrowserRouter>
    </QueryClientProvider>
  );
}
```

DRM 웹 서비스의 라우팅 구조는 크게 로그인, 관리자, 사용자 페이지 3가지로 나뉘어져 있다. 이를 통하여 기대한 장점은 아래와 같았다.

1. 구조적 명확성

- 각 페이지가 보다 직관적인 URL을 가지기 때문에 웹 애플리케이션의 구조가 단순하고 이해하기 쉽다.
- 사용자와 다른 개발자에게 해당 URL이 어느 사용자를 대상으로 하며, 어떤 기능을 수행하는지 쉽게 파악할 수 있음.

2. SEO(검색 엔진 최적화) 친화적

- 각각의 페이지가 고유 URL을 가지기 때문에 검색 엔진 최적화에 있어 유리함을 가진다.
- 각 페이지를 개별적으로 인덱싱하는 것으로 검색 엔진 결과에서 더 잘 노출될 수 있도록 한다.

3. 코드 분할

- 각 페이지 컴포넌트를 독립적으로 로드할 수 있도록 하여 초기 번들 크기를 줄이고 필요한 순간에만 코드를 로드할 수 있도록 한다.

4. 상태 관리의 용이성

- 각 페이지가 독립적인 상태를 가질 수 있도록 하여 상태 관리의 복잡성을 줄이도록 한다.
- 페이지 별로 상태를 분리하여 잠재적인 디버깅과 유지보수 비용을 줄인다.

하지만 서비스의 플로우 상, 각각의 페이지(사용자 및 관리자 페이지) 내부에서는 버튼을

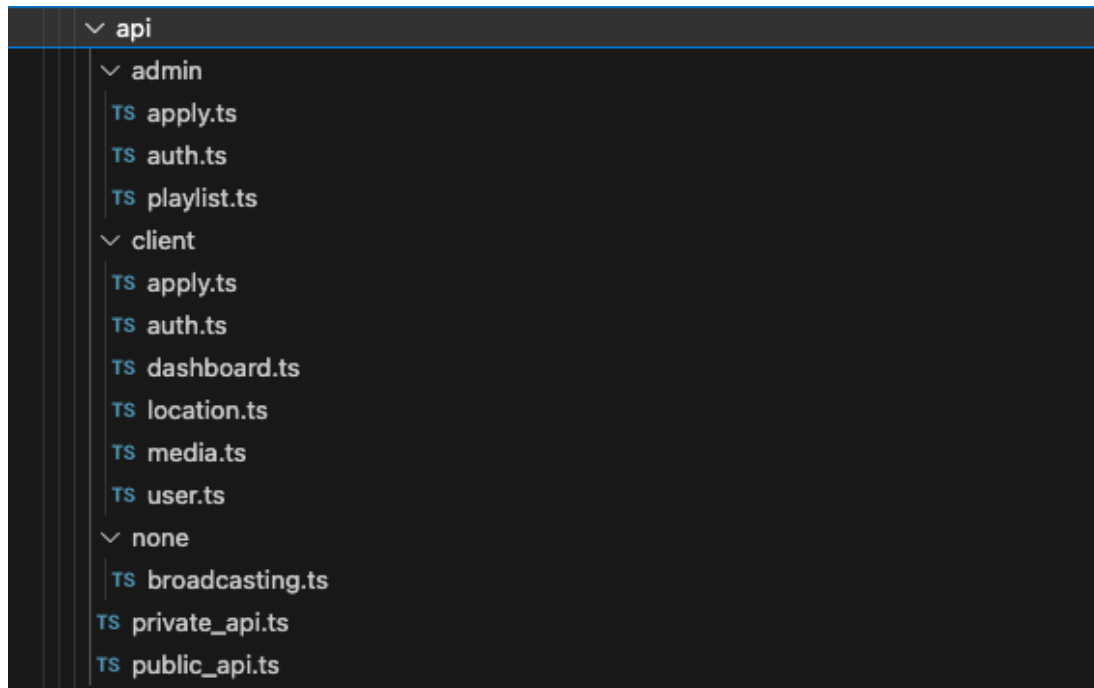
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

이용하여 화면을 전환하는 경우가 잦았고 서로 다른 화면 간 전환 기능이 존재하여 전역으로 상태 관리를 하지 않는 이상 Props drilling 현상을 피할 수 없었다.


또한, 한 페이지에서 state를 변경하여 화면만 전환하는 방식으로 진행하기 때문에 브라우저 내비게이션과 관련한 문제 상황이 발생하였다. 브라우저의 기본 뒤로가기 / 앞으로 가기 기능 구현에 어려움이 있었으며 한 페이지 내에서 URL이 변경되지 않기 때문에 페이지 새로고침 시, 사용자가 보고 있던 상태가 초기화되어 사용자 경험을 해치게 된다는 부분이 있었다.

하지만 이와 같이 하나의 페이지로 구성하여 state로 관리하는 방법으로 서버로부터 받은 응답 결과를 재사용하여 요청 수를 절감할 수 있는 명확한 장점이 존재하기에 적절한 선을 찾는 것이 클라이언트의 최종 목표이자 고민거리라고 볼 수 있다.

2.3.3.4. 서버 API 요청 모듈화



클라이언트에서 서버로의 요청을 관리자용, 사용자용, 토큰이 필요 없는 공용으로 구분하고 재사용하기 위하여 각각 모듈화를 해두었다. 요청 모듈화를 위하여 Axios 라이브러리를 이용하였으며 토큰이 필요한 관리자용이나 사용자용 요청에는 interceptor를 추가하여 request를 보냈을 때 토큰을 추가하여 요청하도록 하고, 토큰 만료 시 refresh token을 이용하여 access token과 refresh token을 자동으로 갱신하도록 하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

2.3.4 서버

2.3.4.1 개발시 서버에 대한 요구사항

현재 중앙 서버는 카메라의 영상을 분석해서 객체 정보를 전달하는 모델 서버와, 대시보드를 보여주는 클라이언트의 사이에 위치한다. 중앙 서버는, 다수의 카메라에서 촬영된 비디오에서 시작되는 다수의 객체 데이터를 송신 받게 된다. 이를 데이터 손실 없이 받고, 시간대, 성별, 나이대, 해당되는 광고 종류 등의 기준에 맞게 적절하게 분류하고, 데이터베이스에 저장하는 역할을 한다. 이후 클라이언트가 대시보드에 사용하기 위해 요청을 하고, 이를 반환해주는 역할을 하게 된다.

이 상황에 특화된 서버의 성능에 대한 요구 사항은 다음과 같다.

1. 데이터 처리의 안정성

다수의 카메라에서 촬영이 이루어지고, 인식된 사람에 대한 대량의 정보가 동시에 서버로 전송될 수 있다. 이때 서버는 최대한 데이터 손실이 적게끔, 주어진 데이터를 처리하여 데이터베이스에 저장해야 한다.

2. 클라이언트 대시보드 API 요구 데이터 반환의 최적화


서버는 카메라 -> 모델 서버를 통해 포착된 "사람당"의 가장 가공되지 않은 형태의 데이터를 받게 된다. 이후 클라이언트는 이러한 데이터로부터, 광고별 집계, 날짜별 집계, 위치별 집계, 성별-나이대별 집계 등의 자세한 분류가 포함된 데이터를 요구하게 된다. 이 요청이 들어올때마다 모두 가공되지 않은 형태의 데이터에서 계산하고 반환해주는 것은 시간이 많이 걸린다는 점에서 비효율적이다. 따라서 클라이언트가 요청하는 상황에 최적화된 형태로 데이터베이스 테이블 형태 및 갱신 서비스 로직을 설계해야 하는 요구사항이 있다.

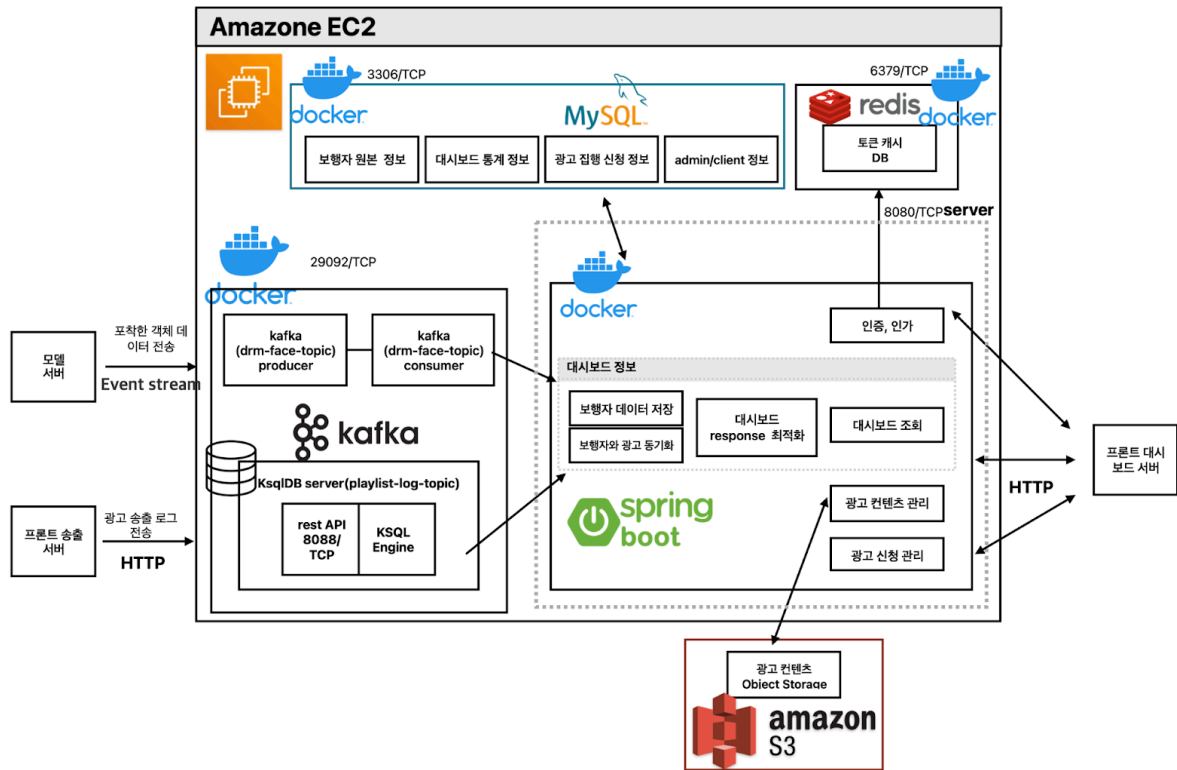
3. 카메라 영상과 송출 광고와의 동기화

모델은 카메라에 사람이 등장한 시점과 나간 시점을 포함한 사람 데이터를 서버로 보내준다. 모델이 보내주는 사람 데이터의 경우 그 사람이 어떤 광고를 보고 있는지 알기 힘들다. 더불어 성능 이슈로 인해 실시간 분석이 어렵기 때문에 송출되고 있는 광고 로고를 따로 저장 후 시간 별로 이를 동기화하여 모델 데이터를 분류하는 작업을 거쳐야 한다.

2.3.4.2 서버 아키텍처 및 ERD

아키텍처

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05




- 카프카 메시지 큐를 이용한 비동기 처리 방식
위에서 언급한 데이터 처리의 안정성을 높이기 위해서는, 한번에 다수의 카메라에서부터 분석된 대량의 데이터가 들어오는 상황에 대비해야 한다. 서버는 요청을 여러 스레드를 통해 관리하나, 제한된 처리 속도를 가지고 있고, 이에 따라, 한번에 많은 요청이 들어왔을때, 이를 처리 속도가 따라가지 못해 데이터가 손실될 수 있다.

따라서, 데이터 손실을 줄이고자, 카메라(모델 서버)로부터 들어오는 모든 데이터는 앞단에 배치하는 카프카 메시지 큐를 통해 보관하도록 설계하였다. 카프카는 Apache 재단에서 관리하는 오픈소스 메시지 큐 툴로, 이벤트 스트리밍 방식으로 들어오는 데이터들을 보관하는 메시지 큐이다.

카프카를 앞단에 배치함으로써, 모델 서버는 서버의 현재 상태와 무관하게 원하는 시간에 데이터를 바로 전송할 수 있게 되었다. 또한 카프카에 보관된 데이터는, 서버의 처리 속도에 맞춰서, 꺼내어 처리하게 된다. 따라서 서버의 처리 속도가 들어오는 데이터의 양을 따라잡지 못하더라도, 데이터 손실 없이 이를 비동기적으로 처리할 수 있게 설계하였다.

또한 카프카를 사용할 경우, 하나의 서버에서 처리할 수 없는 양의 데이터가 들어올 경우, 서버를 증설하고 카프카 내에서 파티션을 서버의 개수만큼 늘리는 방식으로 대응할 수 있어, 전체적인 가용성을 높일 수 있다는 장점이 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

- 클라이언트의 조회에 최적화된 테이블 설계 및 반환 과정 구현
 대시보드를 반환해주는 목적에서, 서버는 다양한 방식으로 통계 데이터를 반환하게 된다. 이때 중요한 것은, 모델 서버로부터 받게 되는 객체에 대한 row 데이터를 어떤 방식으로 데이터베이스에서 관리할 지 설계하는 것이다. 우리는 클라이언트의 요구사항을 분석하여, 이것을 다음 3가지 종류의 데이터 형태로 분류하였다.

1. 원본 row 데이터
2. 일별 갱신 데이터
3. 성별-나이대별 갱신 데이터

1 -> 2 -> 3번 순으로 원시적인 데이터 형태에서, 클라이언트의 요구 데이터 형태로 변화하게 된다. 무한정 클라이언트의 요구 데이터 형태에 가깝게 설계하는 것은 장점만 포함하고 있지는 않다. 데이터베이스의 테이블의 수를 늘리게 되고, 같은 종류의 데이터가, 가공된 형태로 중복해서 존재하며, 데이터베이스의 저장 공간을 점유하게 된다.

이에 따라, 우리는 대시보드의 요청 사항이 크게는 1) 광고별 2) 위치별 3) 나이-성별 세부 타겟팅별로 조회하는 것이라고 판단했고, 이에 따라서 원본, 일별, 성별-나이대별로 분류했다.

이를 통해서, 만약 클라이언트가 광고별 조회를 할 경우에, 이를 모두 row 데이터에서 검색, 합계, 반환해주는 것이 아니라, 갱신되고 있는 일별 데이터를 집계해서 주는 방식으로 설계 했다.

이렇게 클라이언트 요구사항과 가깝지만, 일별, 성별-나이대별 이라는 테이블을 계속해서 갱신함으로써, 클라이언트의 빈번하고, 다양한 대시보드 요청에 빠른 속도로 반환할 수 있게끔 하였다.

- KSQL를 이용한 데이터 조회
 앞서 언급했다시피 우리 모델은 실시간 분석이 어려운 상황이다. 따라서 현재 송출하고있는 광고와 서버에 들어오는 보행자 데이터의 시점이 다르기 때문에 송출 중인 광고 로그를 따로 저장 후에 이를 동기화는 작업을 서버에서 처리 해주어야한다. 하지만 일반적으로 광고는 30초 내외의 짧은 영상이며 이러한 영상이 하루에 몇천번씩 바뀌면서 송출될텐데 이 모든 로그들을 저장하는 방법에 대한 고민이 필요했다.

1. 프론트에서 광고가 전환될때마다 서버로 요청을 보내 mysql 에 저장
 현재 개발환경에 맞춰진 개발이라 개발 비용은 적게 들 것으로 예측된다. 하지만 이미 서버에서는 카프카 이벤트 스트림으로 모델에서 들어오는 데이터 처리, 프론트에서 요청하는 대시보드 로직 처리 등을 맡고 있다. 여기서 로그 요청까지 처리하면 서버에 부하가 올 가능성이 높다. 쿼리를 제공하긴 하지만 아래 두 방법에 비해서는 연산 처리가 느리다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

2. 프론트에서 광고가 전환될 때마다 서버로 요청을 보내 Redis에 저장
 프론트에서 로그 정보를 서버로 요청을 보내는 과정은 1번 방법처럼 서버에 부하가 걸릴 수 있다. 다만 Redis는 mysql과 다르게 in-memory 저장 방식이기 때문에 연산 속도가 빠르며 사용기한을 설정할 수 있기 때문에 .로그와 같이 영구적으로 저장할 필요가 없는 데이터는 삭제를 할 수 있다. 하지만 in-memory 방식의 경우 서버 메모리에 따라 성능이 좌지우지 되기 때문에 우리의 의지와 상관없이 데이터가 소실 될 수도 있다. 모델 서버에서 데이터를 보내기 전에 데이터가 소실 되면 해당 모델 데이터는 광고주에게 전달 할 수 없게 된다.

3. ksqlDB와 restAPI를 이용한 데이터 저장
 KsqlDB에서 제공하는 restAPI로 프론트에서는 광고가 전환될 때마다 ksqlDB에 직접 로그를 전송한다. 서버에서 로그요청을 처리할 필요가 없기 때문에 서버 부하가 줄어든다.

```

ssori
└─┬─> curl -X "POST" "http://0.0.0.0:8088/ksql" \
    -H "Content-Type: application/vnd.ksql.v1+json; charset=utf-8" \
    -d $'{
    "ksql": "INSERT INTO playlistlog_stream (location_id, start_time, end_time, playlist_id) VALUES (2, \'2023-05-16T12:34:56Z\', \'2023-05-16T13:34:56Z\', 1);"
    }'
ssori
  
```

또한 일반적인 카프카 이벤트 스트림으로는 제한적인 명령어로 인해 우리가 원하는 데이터만 필터링 하여 조회할 수 없다고 판단하여 KSQL를 이용하여 쿼리문을 작성하였다. 모델에서 들어오는 데이터 중 location, arrive_at, leave_at 를 가지고 해당 장소, 시간대에 포함되어 있는 송출 로그들을 가져와서 모델 데이터를 광고별로 분류하는 로직을 작성하였다.

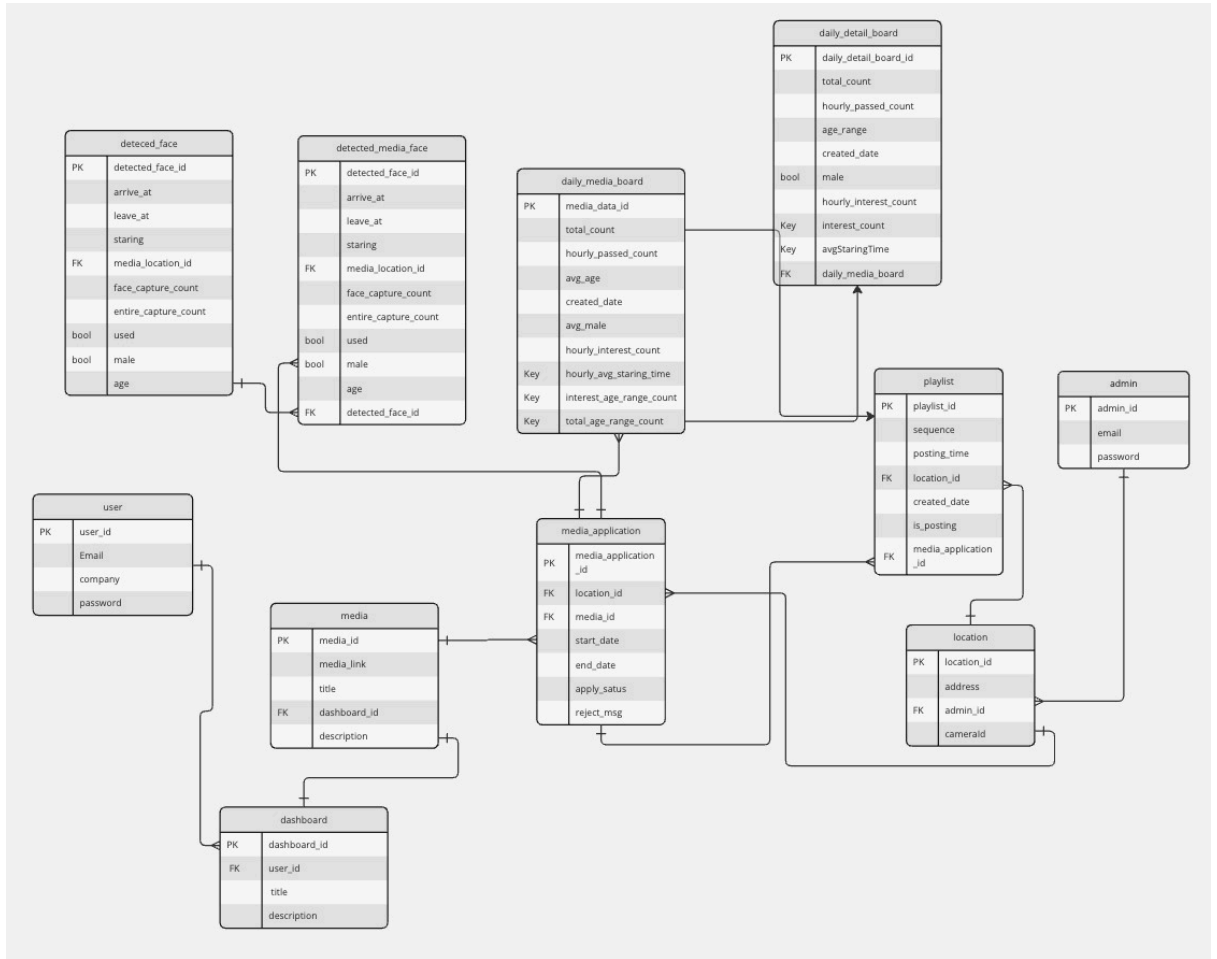
```

ksql> select * from playlistlog_stream
+-----+
|START_TIME|END_TIME|PLAYLIST_ID|LOCATION_ID|
+-----+
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|null|null|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|null|null|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|null|null|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|null|null|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|101|null|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|101|1|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|1|1|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|1|1|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|1|2|
|2023-05-16T12:34:56.000|2023-05-16T13:34:56.000|1|2|
Query Completed
query terminated
  
```

- Redis 이용한 토큰 관리
 redis는 nosql db 로 “유저 id : 리프레시 토큰” 형태로 리프레시 토큰을 관리하는 용도로 사용한다. redis는 in-memory 방법을 사용하기 때문에 하드웨어 단에서 처리하는 다른 데이터베이스들에 비해 연산 속도가 빠르다. 리프레시 토큰의 경우 액세스 토큰을 재발급하기 위해 자주 호출되기 때문에 속도가 빠른 redis를 선정하여 관리하였다. 또한 redis는 유효기간을 지정할 수 있기 때문에 토큰 만료일로 유효기간을 설정할 시 토큰이 자동으로 삭제 되어 토큰 관리에 용이 하다. 또한 토큰의 경우 메모리 용량으로 인해 데이터가 유실되더라도 재발급하여 사용할 수 있기 때문에 토큰 관리에 적합하다고 판단된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

- ERD




2.3.4.3 클라이언트와의 협업 방법 및 제공 API 리스트

개발을 진행할 때, 클라이언트는 서버에 요청하는 API 를 요구하며 개발하게 된다. 하지만, 이런 경우, 클라이언트가 서버의 개발 진행 상황에 맞춰서 동기적으로 작업할 경우 개발 진행이 어렵다고 판단했다.

따라서 서버와 클라이언트의 동시적인 개발을 위해, 서버의 API 를 선제적으로 설계하고, 해당 API 의 요청 데이터 양식, 반환 데이터 양식을 정의해주고 이를 클라이언트에게 제공하는 것이 요구되었다.

이를 위해 고려한 방법은 다음과 같다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

1. postman MockServer 제공

postman 은 지정된 서버로의 API 실행을 웹, 또는 PC 환경에서 대행해주는 테스트 서비스 소프트웨어로 알려져 있다. 여기서 제공하는 기능 중 mock Server 라는 기능이 존재한다. 해당 기능을 통해, 클라이언트에게 전달하길 원하는 API 의 resource url, request parameter, response parameter 를 정의하여 클라이언트와 공유할 수 있다. 클라이언트는 정의된 url 로 요청을 할 시, 정의한 dummy data 를 반환받게 된다.
2. 클라이언트 대상의 개발용 서버 제공

AWS EC2 인스턴스를 사용하여, dummy data 를 리턴하는 컨트롤러 부분만 구현된 백엔드 서버를 배포하여 클라이언트에게 전달하는 방법이다.
3. spring swagger 문서 제공


2의 방법처럼, 백엔드 서버의 컨트롤러를 우선적으로 개발하고, swagger 라는 툴을 gradle 에 포함시켜 컨트롤러의 입출력을 기반으로 클라이언트에게 공유할 수 있는 명세서 문서를 자동 생성되게끔 한다. 해당 swagger 페이지에 대한 port 를 개방할 수 있고, 만들어진 명세서를 PDF 등으로 전달할 수 있다.

우선 시도한 것은 1번 방식이었다. 하지만, postman 서비스에 대해 익숙하지 않은 멤버들이 있었고, 추가적으로 해당 웹 서비스에 자주 들어가며 추가적인 학습 비용이 소모된다고 느껴졌다. 또한 spring backend 서버를 개발하면서, 이와 동일하게 postman mockServer 의 컨트롤러도 지속적으로 별도로 수정해주어야 하는 것은 서버 개발자 입장에서도 부담이 생기는 일이었다. 따라서 해당 방법은 배제했다.

2번 방법인 조기 배포의 경우, 클라우드 자원을 사용하고, 다른 별도의 서비스를 이용하지 않고 가능하다는 장점이 있었다. 하지만 이 경우, 코드의 변경에 따른 CI CD 작업이 먼저 구축되어야지만, 코드 변경에 따라 자동 배포가 가능하여 업무 효율성이 생긴다고 판단했다. 하지만 당시 개발 일정을 고려했을때 CI CD, 배포 관련된 지식을 숙지하는 것보다 서버 설계, 서버 API 개발 등이 우선순위가 높았기에 배제했다.

따라서 3번 방식으로 swagger 를 통해 클라이언트가 요청하는 API에 대한 문서를 생성시키고, 제공하는 것으로 클라이언트와의 협업을 진행했다. 이 방식을 통해 컨트롤러가 개발된 다음에는, 클라이언트가 서버의 작업 속도에 동기화시키지 않아도 되는 장점을 가지며 개발할 수 있었다.

아래는 클라이언트에게 제공하는 swagger API 개발 명세서의 일부이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

auth token 관련	
POST	/api/v1/auth/signup 회원가입
POST	/api/v1/auth/signin 로그인
POST	/api/v1/auth/reissue 토큰 갱신
user user 정보 관련	
POST	/api/v1/user/verify-password 비밀번호 확인
PATCH	/api/v1/user/profile 프로필 변경
PATCH	/api/v1/user/password 비밀번호 변경
Media 광고 등록 관련 api	
GET	/api/v1/media 등록된 광고 리스트
POST	/api/v1/media 광고 등록
DELETE	/api/v1/media/{mediaId} 광고 삭제

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

Dashboard 대시보드 API	
POST	/api/v1/dashboard/{dashboardId}/mediaApplication/{mediaApplicationId} 날짜 별(광고 + 집행기간 + 일(day)) 단위 대시보드
POST	/api/v1/dashboard/{dashboardId}/detail 나이 + 성별 단위로 필터링된 세부 대시보드
POST	/api/v1/dashboard/location/{locationId} Location(디스플레이) 단위 대시보드
POST	/api/v1/dashboard/compare 비교하기
GET	/api/v1/dashboard/{dashboardId} 광고 단위별 대시보드
GET	/api/v1/dashboard/{dashboardId}/board 신청 단위(광고 + 집행 시간) 별 대시보드 리스트
Apply	
POST	/api/v1/media/{mediaId}/apply 광고 신청
GET	/api/v1/media/{mediaId}/apply/{applyId} id 별 신청 데이터 조회
DELETE	/api/v1/media/{mediaId}/apply/{applyId} 신청 취소
GET	/api/v1/media/applies 신청 리스트 조회
Location	
GET	/api/v1/location 디스플레이 장소 조회


2.3.4.4 서버 개발 멤버간 협업 방법, 브랜치 전략

해당 팀에서 서버 개발은 황솔희, 이동국 두 인원이 담당했다. 매주 금요일마다 스크럼을 진행했고, 이에 따라 목표를 설정한 뒤, 주간에는 개별적으로 코드 작업이 진행되었다. 따라서, 코드의 병합시 충돌을 피하기 위해 서버 개발에 대한 결과물을 깃허브 브랜치로의 체계적인 관리가 요구되었다.

서버 코드 개발 브랜치 전략에서는 크게 두 가지로 나눌 수 있었다. 개발용 브랜치 작업물과 배포용 브랜치 작업물이다.

서버 개발 코드는, server/develop 브랜치를 중점으로 하였다. 해당 브랜치에서 매주 개별적인 기능 구현은 server/feat/~기능명으로 분기하여 작업했고, 매주 금요일 전에 추가된 기능을 server/develop 으로 병합하여 협업을 진행했다.

배포용 코드는, 초기 프로토타입 개발이 끝난 뒤, server/deploy 브랜치를 통해 1차적으로 배포했다. 이후 docker compose, docker container 등에 대한 환경 설정을 추가적으로 포함시켜서 server/deploy_test 브랜치를 주 배포 브랜치로 포함시켰다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

해당 브랜치는 배포에 최적화된 도커 세팅 등을 포함시켰고, 코드 변경이 있을시 주된 배포를 위한 절차는 다음과 같다.

1. server/develop 에서 분기되어 feature 작업
2. server/develop 으로의 병합
3. server/develop -> server/deploy_test 로의 병합
4. ec2 instance 에서 git pull 을 통한 최신 코드 업데이트
5. docker build 를 통한 최신 코드 빌드 및 도커 컨테이너에서의 배포

해당 방식으로, 클라이언트의 요구사항이 변경되었거나 버그가 발견되었을 때마다 추가적인 수정, 배포를 진행하였다.


2.3.4.4 클라우드 배포 환경

배포를 위한 클라우드 사용은 학교에서 배스핀글로벌을 통해 지정된 AWS ID 를 통해 진행했다. 배포할때의 목적은 1) 예상할 수 없이 중단되는 상황이 없도록 자원 할당 2) 1을 만족하면서 가장 저렴한 세팅을 하는 것이었다.

위의 1번 목적을 달성하고자 경험적으로 mysql 서버, spring server 를 모두 포함시킬 때 자주 예측할 수 없는 종료 현상이 있다는 것을 느꼈기 때문에, 아래처럼 서버를 3개로 분리하여 설계했다. 3개의 서버에는 모두 가상화된 환경으로의 배포를 지원하는 도커 컨테이너를 이용해 배포를 진행했다.

1. Kafka Server
 - a. zookeeper broker
 - b. kafka
 - c. kafka UI
 - d. ksqldb - server
 - e. ksqldb - cli
2. Backend Server
 - a. spring boot server
 - b. redis cache
3. DB Server
 - a. mysql

이렇게 세개의 서버를 운영한다고 할시, 어떤 스펙의 클라우드 자원을 할당할지 결정해야 한다. 클라우드 자원을 세팅할때의 고려사항은 다음과 같다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

1. 메모리 사이즈
2. Disk volume

초기에 적절한 메모리 사이즈, Disk volume 결정을 위해서, Local docker 에서 해당 서버에서 운영되는 컨테이너들을 임의로 작동시키면서 얼마만큼 메모리를 점유하고, 얼마만큼의 디스크 용량을 차지하는지를 확인했고 이를 기반으로 시작 용량을 결정했다.

하지만 개발 이후에 처리해야 하는 데이터 개수를 높일 경우, (2) 스프링 백엔드 서버의 메모리 점유량이 과하게 높아져서 중단되는 현상, 또는 로그의 양이 방대하게 늘어나면서 디스크가 가득 차 죽어버리는 현상 등이 발견되어, 이것이 용량의 문제라고 판단되었을 경우, 스펙을 업그레이드시켰다.

아래는 최종적인 형태의 서버의 배포 스펙이다.


Aa Name	Instance Type	memory	# disk volume(G)	disk type	Instance IP Address	portNum	# 시간당 요금(USD)
kafka	t2.medium	4G	10	gp3	3.34.47.236	29092, 8989	0.0576
springboot, redis	t2.large	8G	12	gp3	43.200.7.133	8080	0.115215
mysql	t2.medium	4G	8	gp3	3.39.11.158	3306, 8080	0.0576

총 3개의 인스턴스에 3가지 기능적인 분류로 서버를 가동했고, 시간당 요금의 총합은 0.23 USD 한화로 311원이 지출된다. 이 부분에서 시간당 311원일 경우 하루 24시간 가동할 경우 약 7464원 지출된다. 이는 14일 이상 가동시 학교에서 제공하는 100,000원을 초과할 수도 있는 금액이다. 그리고 클라우드를 사용하는 기간은 초기 프로토타입이 개발이 끝난 5월 1주차부터, 5월 4주차(프로젝트 전시회) 일까지 이므로 약 4주이다.

이에 따라 클라우드 사용 자원이, 주어진 자원의 양을 초과한 것이 아쉬운 점이다. 이에 대한 개선 방법으로는, 클라우드 모니터링을 활용하여 팀원들이 잘 사용하지 않는 시간을 확인한 뒤, 해당 시간에서 팀원들이 작업하지 않도록 사전에 공지한 뒤, aws lambda 등의 추가 세팅을 통해 자동 중단 및 일정 시간 후 재가동 등을 이용할 수 있다고 조사했다. 하지만 프로젝트의 주요 로직 및 서버 환경 세팅 등에 시간을 쏟으면서, 클라우드 사용 비용 감축을 위해서 더 시간을 쏟을 수 없다고 느껴서, 해당 부분을 개선하지 못했다.

2.3.4.5 서버 성능 테스트

앞에 언급한 것처럼, 서버의 성능 기준으로, 카프카를 통해 전달 받는 대량의 데이터를 손실 없이 처리하는지를 볼 수 있다. 이를 위해서, 서버에서 동일한 형태의 데이터베이스를 구축하고, 해당 서버로 카프카 Producer 에서 보내는 데이터의 양을 변인으로 두며 점점 양을 늘려가면서 서버가 손실 없이 처리하는지를 관찰했다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

테스트를 위한 백엔드 서버와 데이터베이스 서버는 각각 AWS EC2 instance 를 가동시켜 진행했다.
테스트를 진행한 서버 스펙은 다음과 같다.

aws instance resource				
Aa Name	Instance Type	memory	# disk volume(G)	disk type
kafka	t2.medium	4G	10	gp3
springboot, redis	t2.large	8G	12	gp3
mysql	t2.medium	4G	8	gp3


테스트를 위한 데이터 전송 플로우는 다음과 같다.

- 1) 로컬에서 카프카 Producer Python 코드를 이용하여 MQTT 프로토콜로 Kafka 서버에 전송
- 2) 클라우드에 위치한 Kafka 서버에서 메시지 데이터 송신 및 보관
- 3) 스프링 백엔드 서버가 비동기적으로 처리 속도에 맞춰서 데이터 Consuming
- 4) 처리된 데이터는 클라우드에 위치한 MYSQL DB 에 저장

이 과정에서 다음의 세가지를 판단했다.

- 1) 10만까지의 데이터 전송량을 늘렸을때, 데이터 손실 없이 최종적으로 데이터베이스에 10만개가 저장되는가?
- 2) 10만까지의 데이터 전송량을 늘려가는 과정에서, 백엔드 서버의 비정상 종료가 발생하지 않는가?
- 3) 개별적인 실험에서 카프카로의 데이터 전송 - 최종적인 데이터베이스 데이터 저장까지의 처리 시간은 얼마나 걸리는가

실험 결과 표

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

서버 데이터 처리량 테스트 ...				
# 데이터량	≡ 모델 서버 전송 시작	≡ 마지막 데이터 수정 시간	≡ 소모 시간	≡ 특이사항
10	20:51:07.463	20:51:09.177743	1.714743 seconds	flush 포함
100	20:53:11.942	20:53:27.553697	15.611697 seconds	flush 포함
1000	20:56:33.235	20:59:06.776494	2 minutes 33 seconds	flush 포함
3000	21:44:38.535	21:46:39.211247	2 minutes 1 seconds	flush 제거(loop 외부로)
10000	21:49:32.315	21:56:09.852264	6 minutes 37seconds	flush 제거(loop 외부로)
100000	22:00:34.211	23:06:31.666478	1 hour 5 minutes	flush 제거(loop 외부로)

실험 결과 해석 및 과정 설명

- row 1,2,3 의 경우, 개별적인 데이터 전송 과정에서 송신을 동기적으로 처리하는 flush 를 포함시켜서 전송했다.
- row 4,5,6 의 경우, 개별적인 데이터 전체 전송이 끝난 다음 flush 를 loop 바깥에서 호출하도록 하여 비동기적으로 송신하도록 하였다.

위의 표에서 유의미한 정보는 다음의 두 가지이다.


- 클라이언트의 전송 속도에 비동기, 동기 처리 방식이 송신 속도에 크게 영향을 주는 것을 확인했다.
- 10만개까지 데이터를 큐에 쌓아놓고 Consumer Lag(Consumer 가 앞으로 Consume 해야 하는 데이터의 사이즈)가 크게 높은 경우(작업량이 많이 쌓인 경우)에도, 서버는 본인의 처리 속도에 맞춰서 데이터를 손실 없이 꺼내서 처리한다.

이러한 실험 결과를 통해, 원래의 모델 서버의 데이터 손실을 최소화하는 목표를 비동기적인 카프카 메시지 큐 사용 방법으로 달성했다고 판단했다.


2.4 기대효과 및 활용방안

Do you READ me? 서비스의 기대효과는 다음과 같다.

- 지하철 열차 내 광고에 노출되는 유동인구에 대한 정보와 이들의 관심도를 파악하여 광고 전략을 수립할 수 있다.
- 광고 집행을 위한 예산을 효율적으로 운용할 수 있고, 기존의 오프라인 광고 집행보다 더욱 효과적인 광고 집행이 가능하다.
- 특정 시간대와 노선별로 다양한 유동인구 데이터를 분석함으로써, 광고를 보다 정밀하게 타겟팅할 수 있다. 이를 통해 광고 메시지가 적절한 타겟에게 도달할 확률을 높일 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

- 광고에 대한 소비자 반응과 행동을 분석하여, 제품 및 서비스의 마케팅 전략을 보다 세부적으로 수립할 수 있다.
- 다양한 형식의 광고 (영상, 이미지, 텍스트 등)에 대한 효과를 비교 분석하여, 가장 효과적인 광고 형식을 도출할 수 있다.
- 유동인구와 광고 노출 데이터를 수집하고 분석하여, 광고주들이 광고 전략을 수정하거나 보완할 수 있도록 지원한다.
- 광고 효과 분석을 통해 어떤 광고가 가장 효과적인지에 대한 인사이트를 얻고, 이를 바탕으로 향후 광고 성과를 지속적으로 개선할 수 있다.
- 정확한 광고 효과 분석을 통해 광고 투자 대비 수익(Return on Investment, ROI)을 극대화할 수 있으며, 이를 통해 광고 예산을 보다 효율적으로 배분할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

3 자기평가

(이동국)


크게 팀장으로서의 역할과 서버 개발자, 인프라 개발자로서의 역할 두 가지를 수행했다. 팀장으로서 서버, 모델, 클라이언트 다양한 파트들에 대해 지속적으로 관심을 가지고 물어보면서 진행 상황을 적극적으로 소통했다. 또한 지속적으로 팀의 주간 스크럼에서, 해당 기간의 우선순위가 무엇인지 질문하면서 중요한 것을 우선적으로 처리하도록 한 것이 가장 잘한 점이라고 생각한다. 부족한 점은 중간 발표 준비를 소홀히 하고, 역할 분담 없이 혼자서 모든 것을 처리하는게 더 효율적이라고 처음에 생각했던 점이다. 이 부분은 다른 팀원들의 피드백을 받고, 적절하게 역할을 나눠서 다시 작업하는 형식으로 개선했다. 서버 개발자로서는 서버 대시보드 서비스 로직 개발, 클라우드 자원 관리 및 배포, 깃허브 브랜치 관리 등을 수행했다. 카프카를 도입한 것, 적절한 갱신용 테이블을 도입한 것, 그리고 서버 대시보드 개발에서 TDD 를 작게나마 도입해서 선제적으로 버그를 몇가지 수정하고, 다양한 종류의 대시보드 API 를 개발한 것을 잘했던 점이라고 생각한다. 아쉬운 점은 서버 개발의 다양한 요소들을 다른 경험 있는 팀원에게 많이 맡겼다고 느끼며, 이에 따라 서버 전체의 코드에서 보안, 레디스 캐싱 등에 대해서 이해하지 못하고 사용한 것이다.

(이강희)

캡스톤 프로젝트에서 모델 파이프라인 구현을 담당했다. 이번 프로젝트를 통해 처음으로 동영상 데이터를

(장수미)

캡스톤 프로젝트에서 기획과 UI 디자인을 담당하였다. 평소 일반 사용자를 대상으로 기획하여 모바일 UI을 디자인하는 것을 주로 담당했었다. 이번 캡스톤에서는 모델을 개발하는 팀원이 주축이 되어 평소에 하던 주제와는 색다른 도전적인 주제를 맡을 수 있었다. B2B 서비스를 기획하는 것, 특히 데스크탑을 기반으로 대시보드를 중점적으로 디자인하는 것이 처음이라 어려운 부분이 있었다. 특히 대시보드의 경우 온라인 데이터를 기반으로 하는 경우가 많아, 참고할 레퍼런스가 많지않아 난감했다. 이러한 경험이 부족한 부분을 커버하기 위해 광고홍보학과 교수님의 자문, 팀원들의 조언을 적극적으로 반영해 기획하려고 노력했다. 모든 작업은 Figma를 사용했으며, UI Description과 Figma Comments를 통해 개발자들과 소통하며 서비스 기획을 디벨롭했다. 프로젝트에서 개발을 담당하지 않았기 때문에 팀원들이 최대한 개발에 집중할 수 있도록 돕고자 했다. 전화 및 이메일과 같은 문의, 중요한 일정 리마인드, 회의비 증빙, 웹캠 및 디스플레이 대여와 같은 행정적인 일을 처리했다. 개발 막바지의 경우 최종적으로 제출해야하는 자료를 주도적으로 만들며 다른 팀원들이 최대한 개발 마무리에 힘 쓸수있도록 했다. 일의 분배에도 균형을 맞추어 프로젝트를 진행함이 중요함을 느꼈다. 해당 프로젝트를 통해 배경이 다른 사람과의 소통 방법에 대해 생각하게 되었다. 이전에는 프론트엔드 개발자와 주로 소통하여서 서버 및 모델 개발자와 소통할 일이 많지 않았다. 개발 분야가 다양하니 다른 이해 관계를 가진 팀원들과의 소통을 경험을 할 수 있었다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

(허수빈)


(황솔희)

캡스톤 프로젝트에서 서버 개발 및 아키텍처 설계를 맡았다. 이런 B2B 서비스 및 대시보드 형 서비스는 처음 개발해봐서 처음에는 많이 해맸던 것 같다. 실제로 서비스하기 위해서는 광고 콘텐츠 관리, 광고 송출, 대시보드 데이터 관리 등 고려해야 할 게 많은데 이 모든 걸 다 구현하자니 시간적으로나 실력적으로 무리였고 일부만 구현하자니 데이터에 일관성이 없어지는 등의 용납이 안 되는 부분이 생겨 타협하는 과정이 오래걸렸던 것 같다. 이러한 서비스 확장 정도를 결정하는 과정에서 ERD 및 아키텍처 설계도 느려지고 얼리타임에도 불구하고 개발 시작은 늦어진 점이 아쉬웠다. 다행히도 팀원 모두 적극적인 참여와 소통으로 적절한 타협점이 생기고 난이도 너무 쉬워지지 않는 선에서 서비스가 구체화 되었고 아키텍처와 ERD를 바로 작성하여 조금이라도 개발을 빨리 시작할 수 있게 되었다.

개발 초기에는 전 프로젝트들에서 썼던 기술들과 유사한 서비스가 많아서 순탄하게 진행이 되었다. 중간 발표때까지 광고 콘텐츠 관리, 신청 관리, 유저 등 익숙한 작업들은 빨리 처리하여 프로젝트에 부담이 덜했던 것 같다. 다만 JPA를 사용하는 과정에서 쿼리 사용을 남발하는 감이 없지 않아 있어 서버 성능 개선의 필요성을 느꼈다. 해당 작업을 프로젝트 후반에 진행하려고 했으나 카프카를 도입한 순간부터 할 일이 많아져 신경을 못쓰게 아쉬웠다. 카프카를 도입해서 전반적인 서버 성능을 올리려고 한 건 잘한 점이라고 생각한다. 대규모 데이터를 다루지 않으면 카프카를 쓸 일이 없다고 생각했는데 우리 서비스에 도입하면서 단순히 카프카 스트림에서 그치는 것이 아닌 여러 상황을 가정하여 개발도 해본 것이 도움이 되었다. 그로 인해 모델 데이터와 광고 동기화 문제를 끝까지 해결 못할 것이라고 생각했는데 스트리밍 서비스를 개발하지 않고도 동기화 처리를 할 수 있게 되어 완성도를 높였다고 생각한다.

(최지원)


캡스톤 프로젝트에서 클라이언트 개발을 맡아 사용자(광고주)가 사용하는 사용자 페이지와 관리자가 쉽게 광고 승인 및 송출을 도울 수 있는 관리자 페이지(백 오피스)를 구현하였다. 이번 캡스톤 디자인에서 진행하는 DRM 서비스의 소비 대상은 기업이자 광고주이고 대시보드 형태의 웹 서비스를 제공해야 했다. 서비스 타겟과 형태 모두 처음 접하는 것이었기에 다소 난해하고 생소했던 부분들이 많았다. 또한, 전체적인 BM 수립을 위한 고객 가치가 확립할 수 있도록 기획, 디자인을 담당한 팀원과 Figma 코멘트를 이용하여 소통하거나 추가적인 회의를 진행하며 프로젝트를 진행하였다. 이번 프로젝트는 '사용자 경험(UX)'과 '성능 개선', 2가지 키워드를 중점으로 개발을 시작하였다. 첫 번째로 기존 React보다 더 최적화된 빌드를 가져오고 초기 렌더링 속도를 줄일 수 있도록 Vite를 사용하였으며 사용자가 런타임(run-time) 환경에서의 비정상적인 종료 및 오류 발생을 막기 위하여 동적 타입을 지원하는 Typescript를 사용 언어로 지정하였다. 두 번째로는 일관된 사용자 인터페이스를 제공하고자 재사용 가능한 단위를 기준으로 컴포넌트를 만들고 관리하였으며 기본 UI Component는 Ant Design 라이브러리를,

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

대시보드와 인사이트 내부 차트 형태의 UI Component는 Apex Charts 라이브러리를 사용하였다. 이미 만들어져 있는 라이브러리를 사용하는 것으로 간단한 윤곽을 잡는 것은 빠르게 진행하였지만 기획 단계에서 정한 세세한 사용자 인터렉션이나 디자인의 변경을 반영하기 위하여 커스터마이징 하는 것에는 생각보다 더 오랜 시간이 걸렸고 이 부분에서는 만족스러운 결과물을 얻지 못한 것 같아 아쉬움이 있다. 마지막으로 Axios와 State 활용으로 서버 측에 중복되거나 불필요한 데이터 요청을 최소화하였다. 크게 관리자용, 사용자용, 공용 API가 존재하였는데 이를 각각 axios instance로 분리하여 재사용 가능하도록 모듈화하였다. 인증이 필요한 요청에 대한 토큰 추가, 토큰 만료 시 재발급 등은 axios interceptor를 이용하여 진행하여 일관된 예외 처리와 간편한 재사용성을 확보할 수 있었다. 다만, 프로젝트에서 라우팅 구조를 최대한 단순하게 가져가고 내부 스크린 전환을 state로 관리하여 요청 최소화, 초기 렌더링 속도 확보는 성공하였지만 가장 기본적인 브라우저 네비게이션인 뒤로가기, 앞으로 가기에 대한 처리를 하지 못하여 이 부분에서 사용자 경험을 해치게 되어 아쉬움이 남았다. 프로젝트를 진행하며 가장 어려웠고 또 가장 중요하다고 느꼈던 것은 적당한 지점을 찾는 것이다. 사용자 경험과 성능 개선은 둘다 중요한 문제이지만 어느 쪽에 더 초점을 맞출 지, 얼마만큼의 리스크를 허용할 것인지에 대하여 보다 세밀하게 검토할 필요가 있다 느꼈다. 그리고 서버 개발자와의 요청 및 응답 구조에 대하여 논의하고, 자료에 대한 처리 영역을 나누며 전체적인 프로젝트의 성능을 개선해나가는 경험을 할 수 있었다.

4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	공식문서	ksqlDB 공식문서	https://docs.ksqldb.io/en/latest/operate-and-deploy/			
2	공식문서	apache kafka 공식문서	https://kafka.apache.org/documentation/			
3	공식문서	spring 공식문서	https://spring.io/projects/spring-boot			
4	공식문서	Vite 공식문서	https://ko.vitejs.dev/			

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

5 부록

5.1 사용자 매뉴얼

5.1.1 공용

1. 프로젝트 작업 저장소 받아오기

`git clone` <https://github.com/kookmin-sw/capstone-2024-04.git>

`git clone` 명령어를 통하여 클라이언트, 서버, 모델의 작업 저장소를 로컬 환경에 가져온다.

5.1.2 클라이언트

1. 클라이언트 디렉토리로 이동

`cd src/client`

현재 작업 디렉토리를 `src/client`로 변경한다. 프로젝트의 클라이언트 관련 소스 코드가 존재하는 장소이다.

2. 의존성 설치

`yarn`

`yarn` 명령어를 통하여 프로젝트에 필요한 모든 의존성을 설치한다. 포함된 패키지는 `package.json`에 모두 정의되어 있어 확인이 가능하다.

3. 개발 서버 실행

`yarn run dev`

```
VITE v5.1.6 ready in 187 ms
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

`yarn run dev` 명령어를 통하여 개발 서버를 시작한다. 개발 서버는 5173 포트를 사용하며 서버와의 원활한 통신을 위하여 해당 포트를 점유하고 있는 다른 프로그램이 있을 시에는 해당 프로그램을 종료해야 한다.

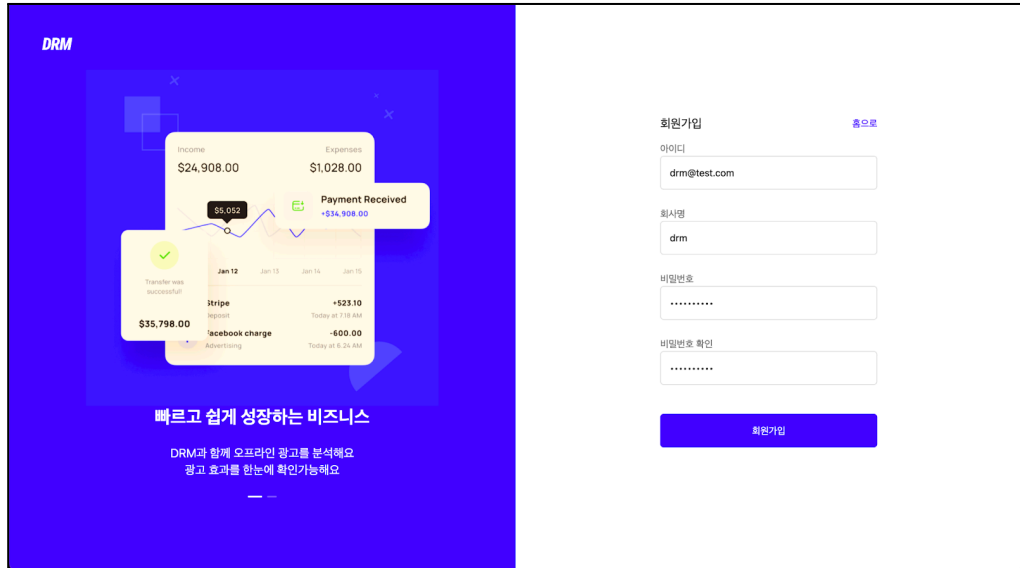
4. 사용자(광고주) 계정으로 웹 서비스 이용하기

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

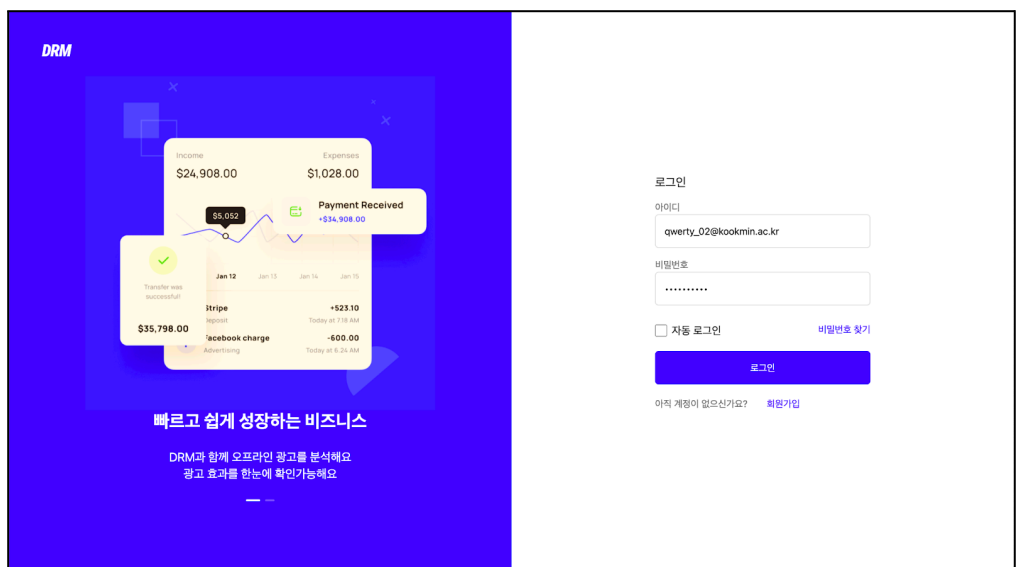
위 명령어를 통하여 시작된 개발 서버(<http://localhost:5173/>)로 접속한다.

4.1. 회원가입 및 로그인


DRM 서비스를 최초 이용하는 사용자는 회원가입이 필요하다.



회원가입 시에는 아이디, 회사명, 비밀번호에 대한 정보를 기입해야 하며 비밀번호는 영문자, 숫자 조합 8글자 이상으로 설정해야 한다. 모든 정보를 기입하면 회원가입 버튼(CTA버튼)이 활성화되며 성공적으로 가입이 완료된 경우, 로그인 페이지로 돌아간다.

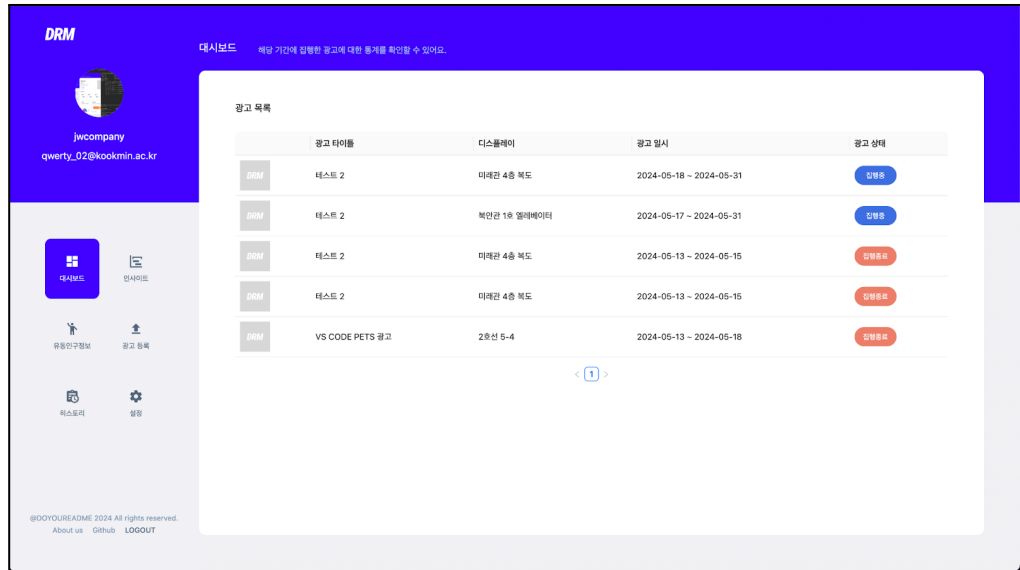


아이디와 비밀번호를 통하여 서비스에 로그인할 수 있다. 자동 로그인을 활성화한 경우,

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

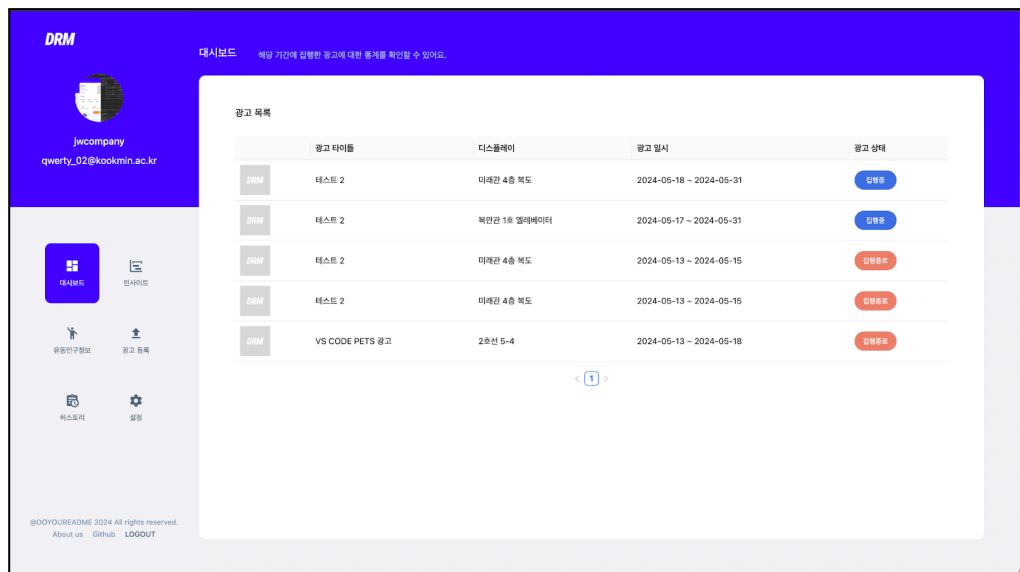
이후 브라우저 접속 시에 웹 브라우저 쿠키에 정보를 저장된 정보를 통하여 자동으로 로그인을 시도한다.


4.2. 사용자 `Home` 페이지



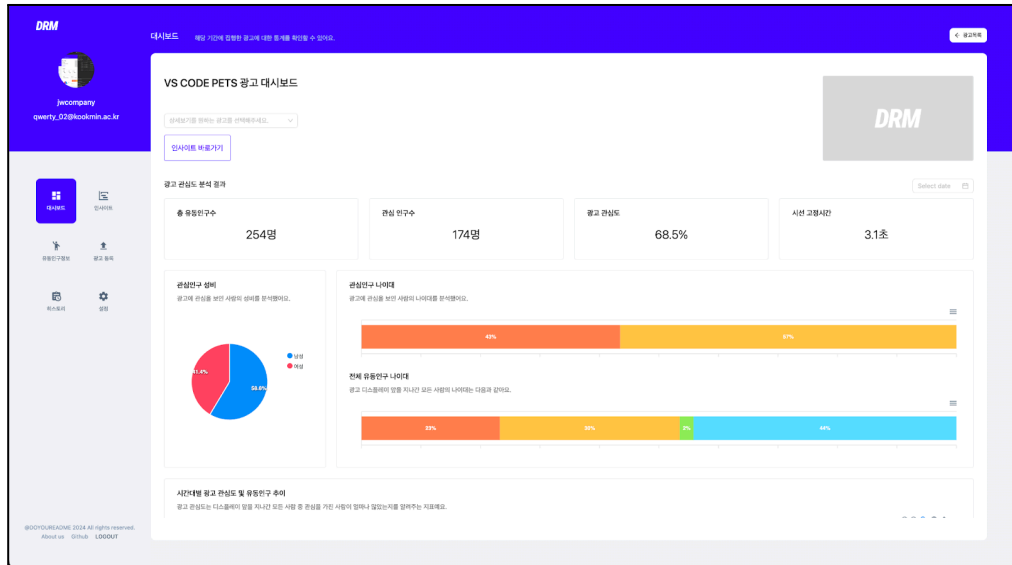
로그인 이후, 사용자가 가장 먼저 진입하는 페이지이다. 사용자는 좌측 6개의 메뉴 버튼을 통하여 ‘광고 대시보드 조회’, ‘광고 인사이트 조회’, ‘유동인구 정보 조회’, ‘광고 등록’, ‘광고 히스토리 조회’, ‘개인 정보 설정’ 등의 기능을 이용할 수 있으며 좌측 하단 ‘LOGOUT’ 텍스트 버튼을 통하여 로그아웃이 가능하다.

4.2.1. 대시보드 페이지

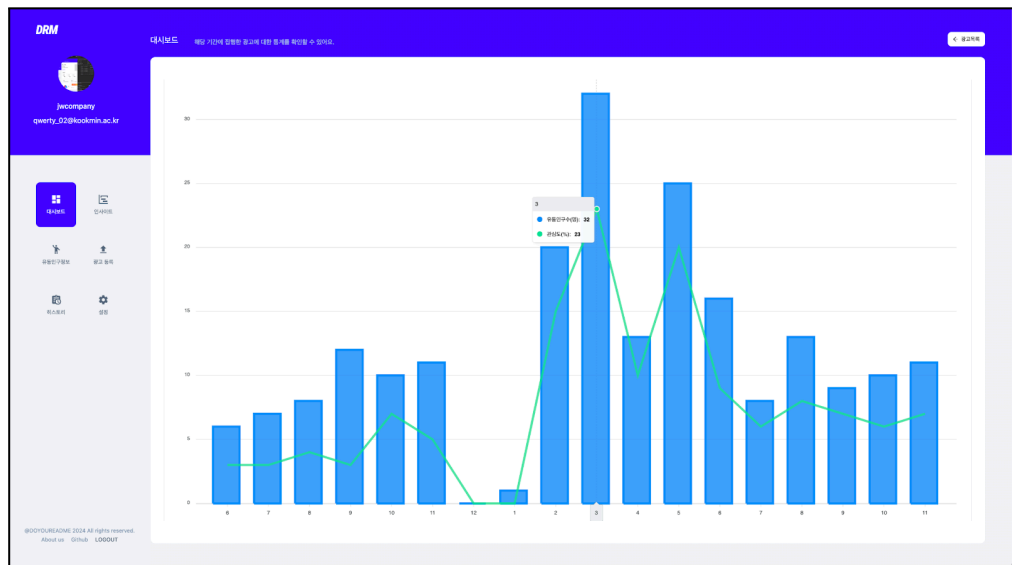


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05


대시보드 페이지에서는 사용자가 등록한 광고 중 승인된 광고(대시보드가 생성된 광고)들에 대한 목록이 나온다. 광고 타이틀, 송출 디스플레이, 광고 집행 일시와 광고 상태에 대한 정보를 간략하게 확인 가능하다.



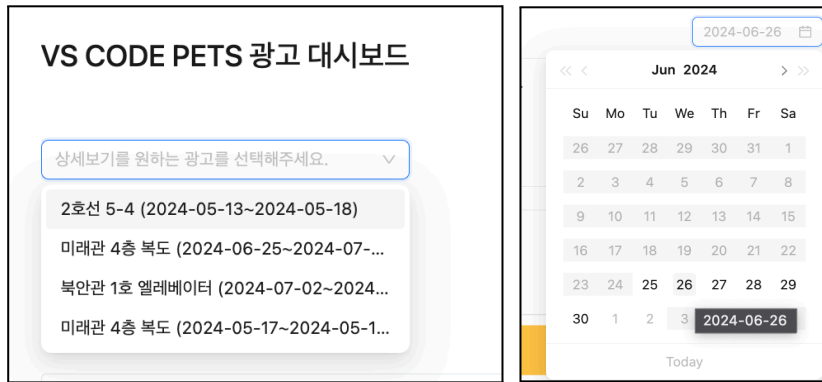
대시보드 상세 페이지에서는 선택한 광고에 대한 분석 정보를 제공받을 수 있다. 제공하는 주요 정보는 ‘총 유동인구수’, ‘관심 인구수’, ‘광고 관심도’, ‘시선 고정시간’, ‘관심인구 성비’, ‘관심인구 나이대’, ‘전체 유동인구 나이대’가 있다.



추가로 시간대별 광고 관심도와 유동인구 추이를 확인할 수 있어 광고주가 추후 광고 집행 시간 및 타겟 변경에 있어 유의미한 기준으로 사용할 수 있다.

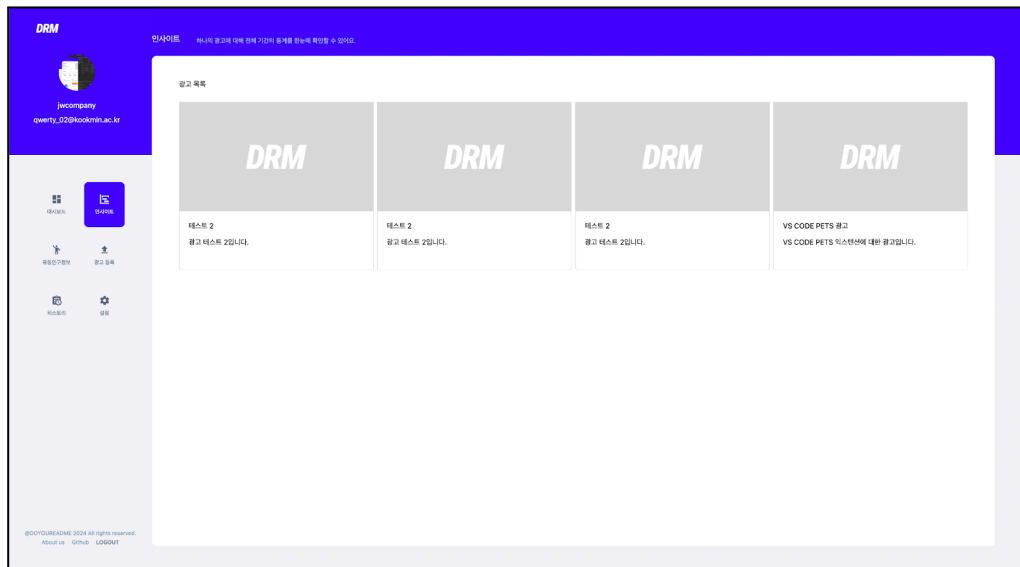
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

각 그래프의 막대에 커서를 갖다대면 정확한 유동인구수와 관심도를 확인할 수 있다.




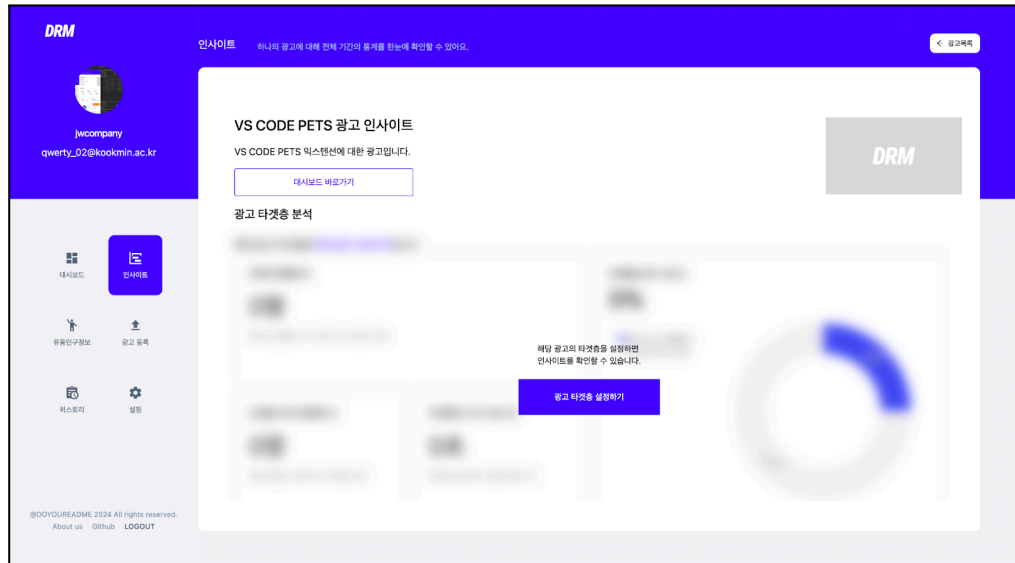
추가로 광고 타이틀, 광고 설명 하단에 있는 Select 컴포넌트를 클릭하여 광고 집행 단위별 상세보기 또한 가능하다. 상세보기는 기본적으로 집행 기간동안의 데이터를 종합하여 분석한 결과를 제공하며, 사용자는 우측에 있는 DatePicker 컴포넌트를 통하여 특정 일자의 데이터만을 분석한 결과를 제공받을 수 있다.

4.2.2. 인사이트 페이지

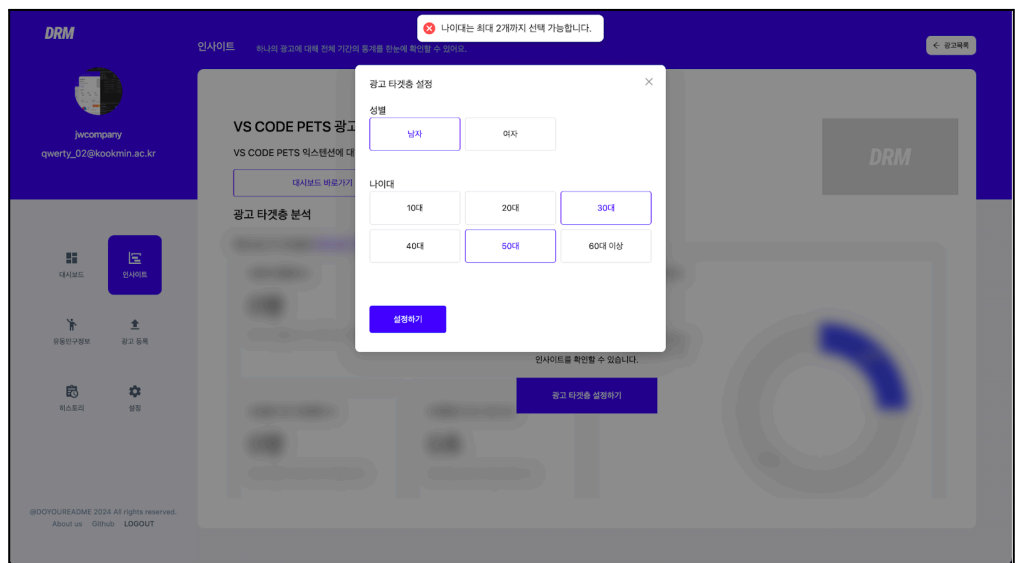


사용자는 인사이트 페이지를 통하여 자신이 등록한 광고 중, 대시보드가 만들어진 광고(집행 중인 광고, 집행 종료된 광고)에 대한 인사이트를 확인할 수 있다. 인사이트 목록에서 광고 썸네일, 광고 타이틀, 설명 등을 간략하게 확인할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

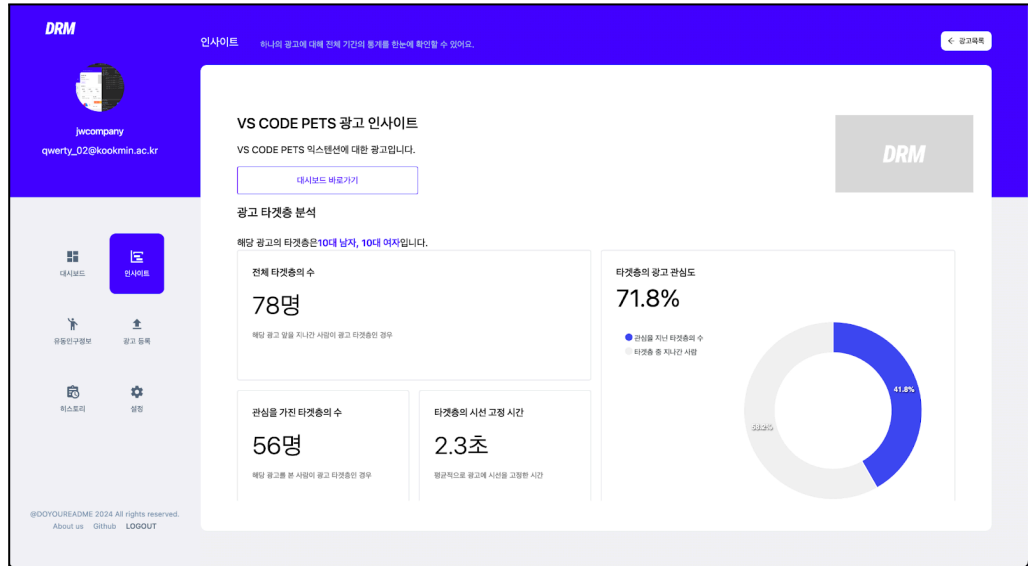


인사이트 상세 페이지 초기 진입 화면으로 광고 타겟층을 설정하기 전에는 분석 결과를 제공하지 않는다.

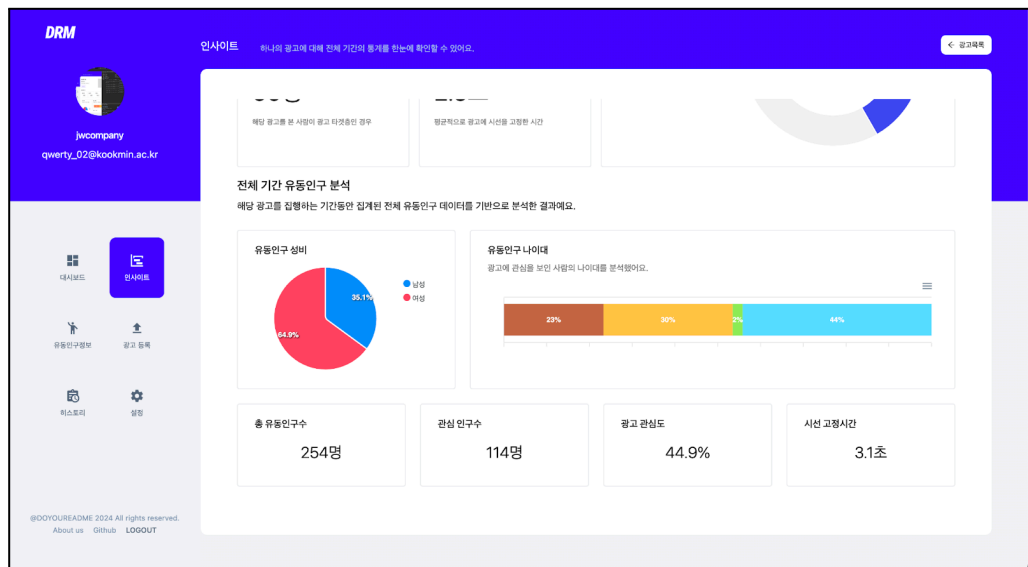


사용자는 광고 타겟층을 라디오 버튼 형태로 선택할 수 있다. 성별 키워드로는 남자, 여자, 성별무관 3가지로 필터링이 가능하며 나이대 키워드는 최대 2개까지 선택 가능하다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05



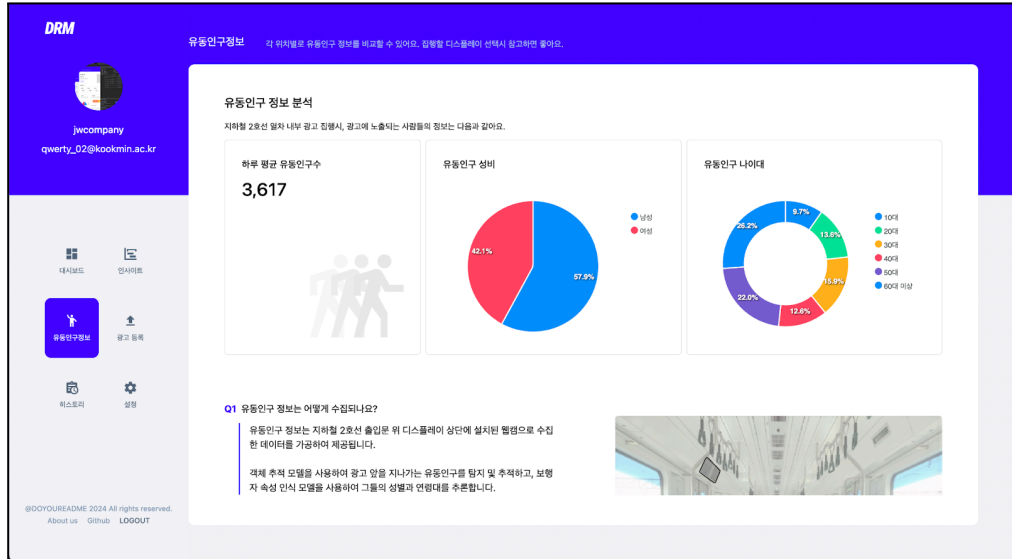
광고 타겟층이 설정 완료된 이후에는 분석 결과를 제공받을 수 있다. 주요 제공 정보로는 ‘전체 타겟층의 수’, ‘관심을 가진 타겟층의 수’, ‘타겟층의 시선 고정 시간’, ‘타겟층의 광고 관심도’가 있다.



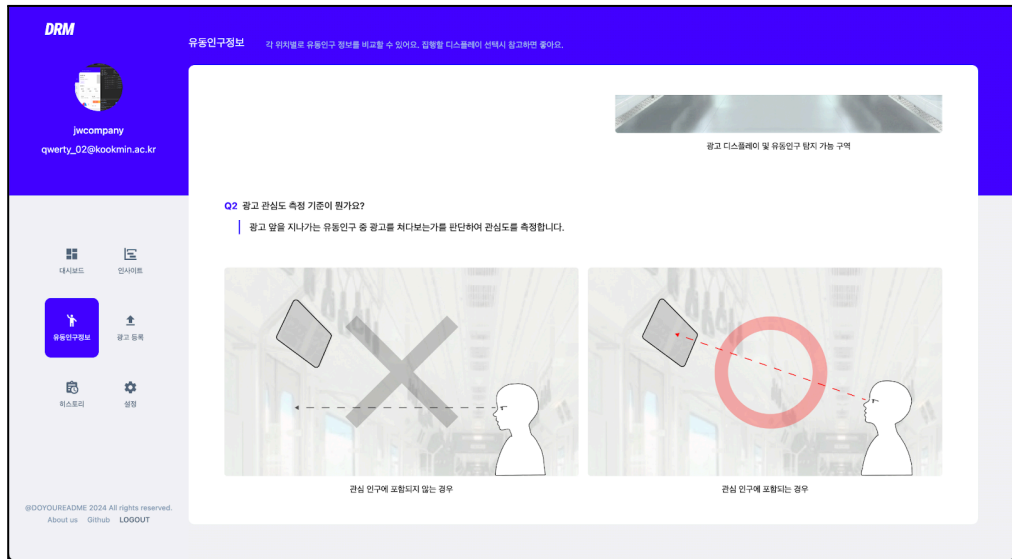
인사이트 상세 페이지 하단에는 전체 유동인구에 대한 정보가 제공되기 때문에 광고주는 광고의 타겟층 설정이 제대로 되었는지 등을 쉽고 빠르게 확인할 수 있다.

4.2.3. 유동인구 정보 페이지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05




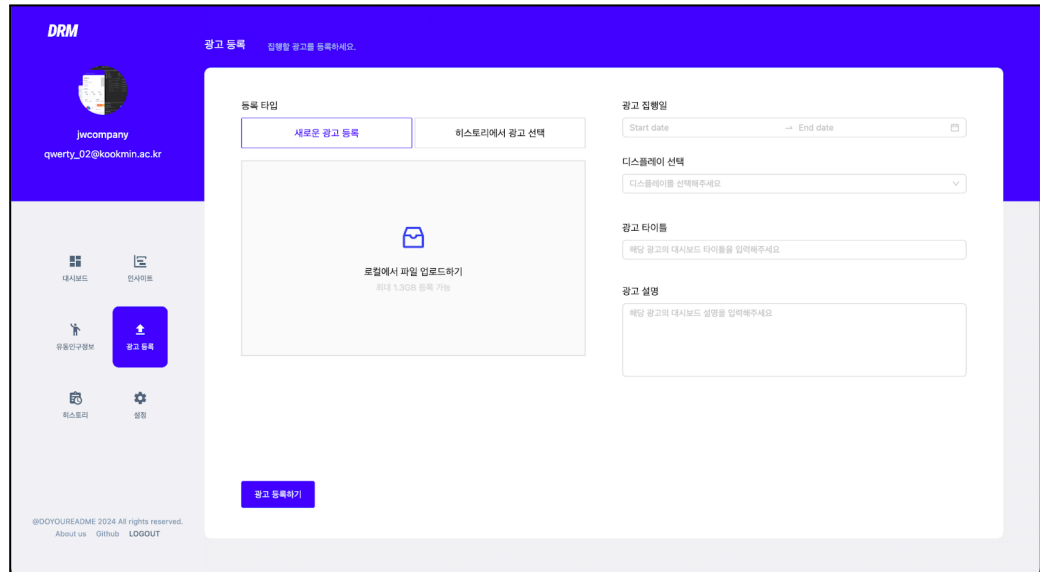
사용자는 유동인구정보 페이지에서 지하철 2호선 열차 내부의 ‘일 평균 유동인구수’, ‘유동인구 성비’, ‘유동인구 나이대’에 대한 정보를 확인할 수 있다.



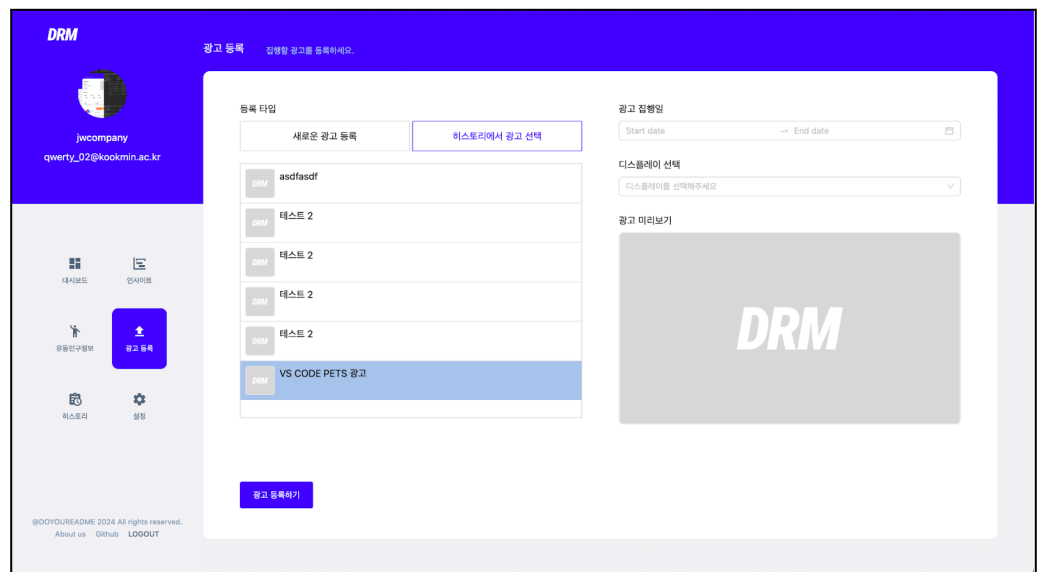
또한, DRM 서비스에서 유동인구 정보 수집 및 광고 관심도 측정 방법을 공개하는 것으로 사용자가 자주 묻는 정보에 대한 답을 추가로 제공한다.

4.2.4. 광고 등록 페이지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05



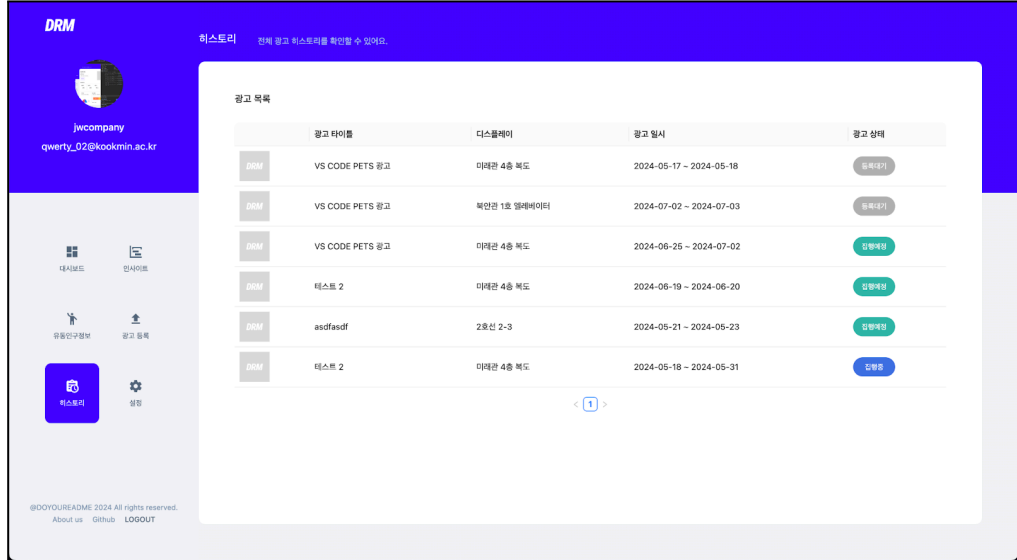
사용자는 광고 등록 페이지에서 송출 및 분석을 원하는 광고를 신청할 수 있다. 로컬에서 업로드한 광고 영상 파일, 광고 집행일, 송출을 원하는 디스플레이, 광고 타이틀, 설명을 기입하여 신규 등록을 할 수 있다.



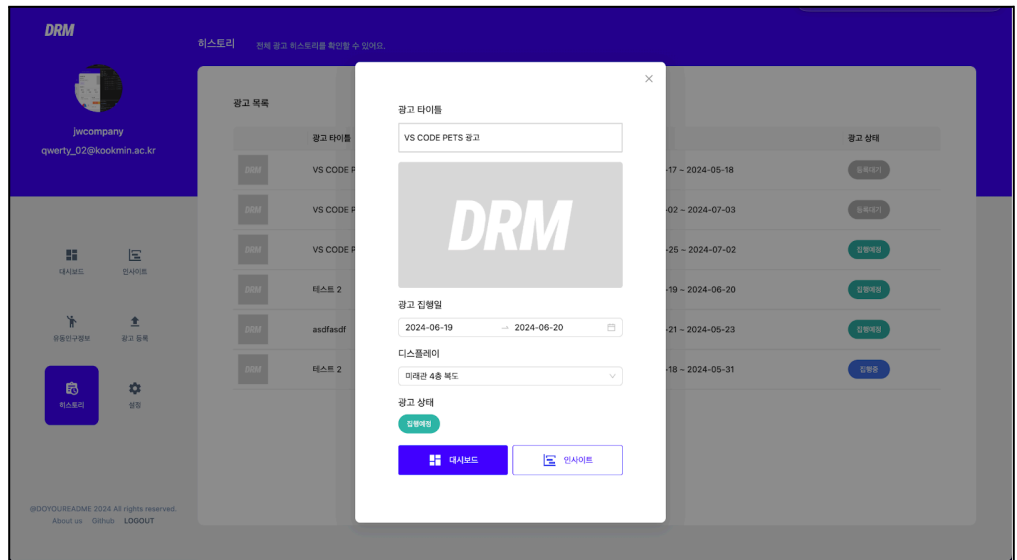
사용자는 '히스토리에서 광고 선택' 기능을 통하여 기존 등록된 광고에서 광고 집행일이나 송출 디스플레이만 변경하여 추가할 수 있다. 해당 기능을 통하여 장소별 A/B 테스트 진행 및 광고 집행일 연장 등을 할 수 있다.

4.2.5. 히스토리 페이지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05




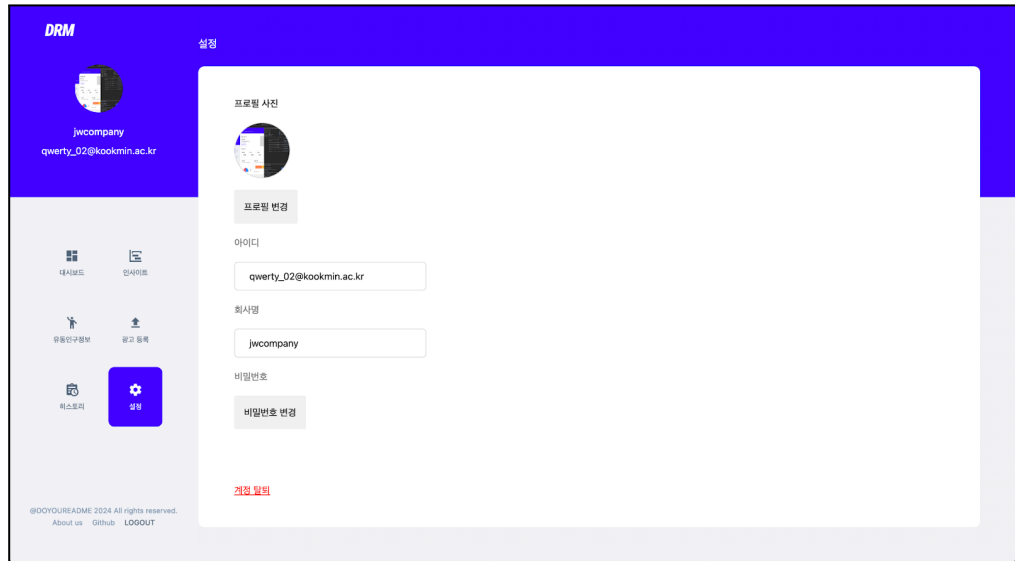
사용자가 등록한 모든 광고를 조회할 수 있습니다. 관리자의 승인을 대기 중인 광고는 '등록대기', 승인된 광고는 광고 집행 일자에 따라 '집행예정', '집행중', '집행완료'로 구분됩니다.



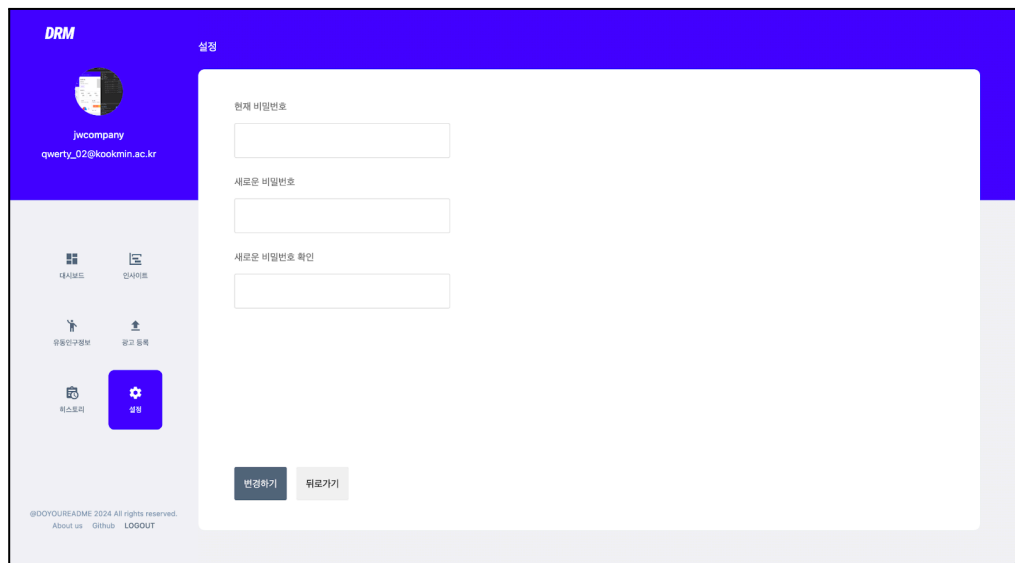
광고를 클릭하면 목록에서 제공되는 정보를 보다 자세하게 확인할 수 있으며 대시보드와 인사이트가 생성된 경우, 해당 광고의 대시보드와 인사이트 상세 페이지로 이동할 수 있다.

4.2.6. 설정 페이지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5




설정 페이지에서는 회원가입 시 설정한 아이디, 회사명을 확인할 수 있으며 프로필, 비밀번호 변경, 회원탈퇴를 진행할 수 있다.



비밀번호 변경 시에는 현재 비밀번호를 더블 체크를 진행하고 있으며, 현재 비밀번호를 잊어버린 경우에는 로그인 화면에서 찾을 수 있다.

5.1.3 서버

5.1.4 모델

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

5.2 운영자 매뉴얼

5.2.1 클라이언트

1. 클라이언트 디렉토리로 이동

```
cd src/client
```

현재 작업 디렉토리를 `src/client`로 변경한다. 프로젝트의 클라이언트 관련 소스 코드가 존재하는 장소이다.

2. 의존성 설치

```
yarn
```

`yarn` 명령어를 통하여 프로젝트에 필요한 모든 의존성을 설치한다. 포함된 패키지는 `package.json`에 모두 정의되어 있어 확인이 가능하다.

3. 개발 서버 실행

```
yarn run dev
```


```
VITE v5.1.6 ready in 187 ms
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

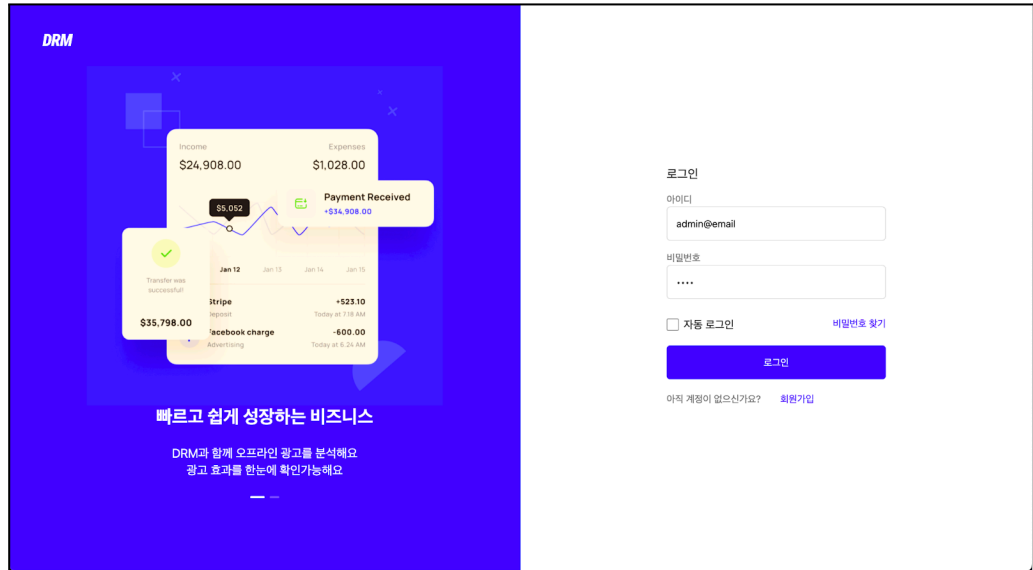
`yarn run dev` 명령어를 통하여 개발 서버를 시작한다. 개발 서버는 5173 포트를 사용하며 서버와의 원활한 통신을 위하여 해당 포트를 점유하고 있는 다른 프로그램이 있을 시에는 해당 프로그램을 종료해야 한다.

4. 관리자 계정으로 웹 서비스 이용하기

위 명령어를 통하여 시작된 개발 서버(<http://localhost:5173/>)로 접속한다.

4.1. 로그인

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

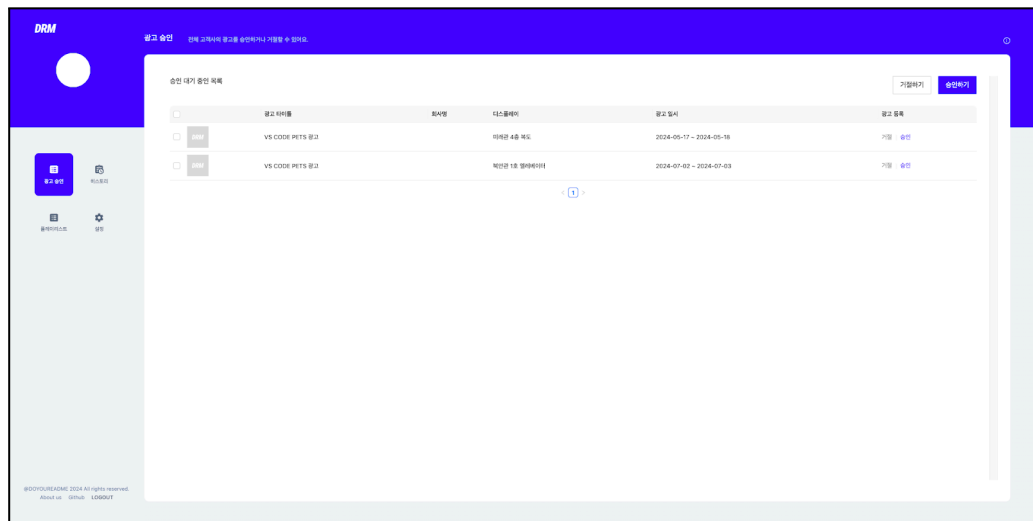


사전에 발급받은 관리자 계정을 통하여 로그인을 진행한다. 로그인이 완료되면 관리자 페이지로 진입하게 된다.

4.2. 관리자 기능

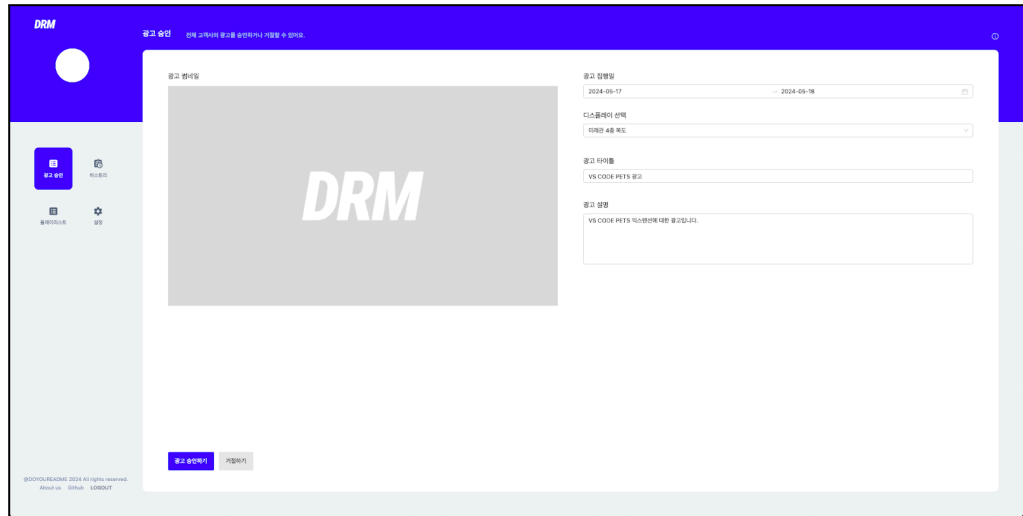
관리자는 관리자 페이지에서 ‘광고 승인’, ‘전체 사용자 히스토리 조회’, ‘플레이리스트 등록 및 장소별 플레이리스트 확인’, ‘관리자 정보 변경’ 등의 기능을 수행할 수 있다.

4.2.1 광고 승인



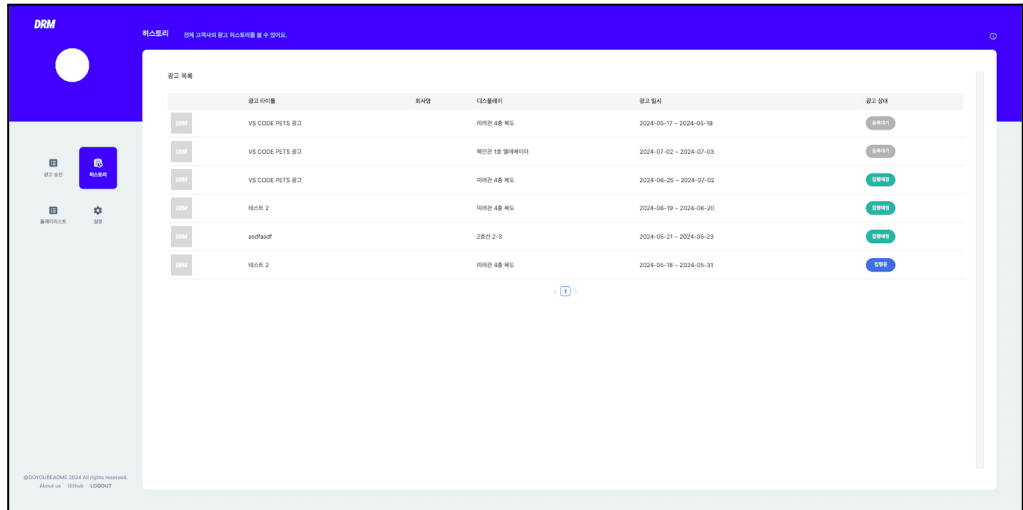
관리자는 사용자(광고주)들이 등록한 광고를 승인하거나 거절할 수 있다. 승인 및 거절을 원하는 광고를 모두 체크한 뒤 우측 버튼을 통하여 일괄 처리할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05




또한, 직접 광고의 세부 정보를 확인하고 개별적으로 승인 및 거절 처리를 진행할 수 있다.

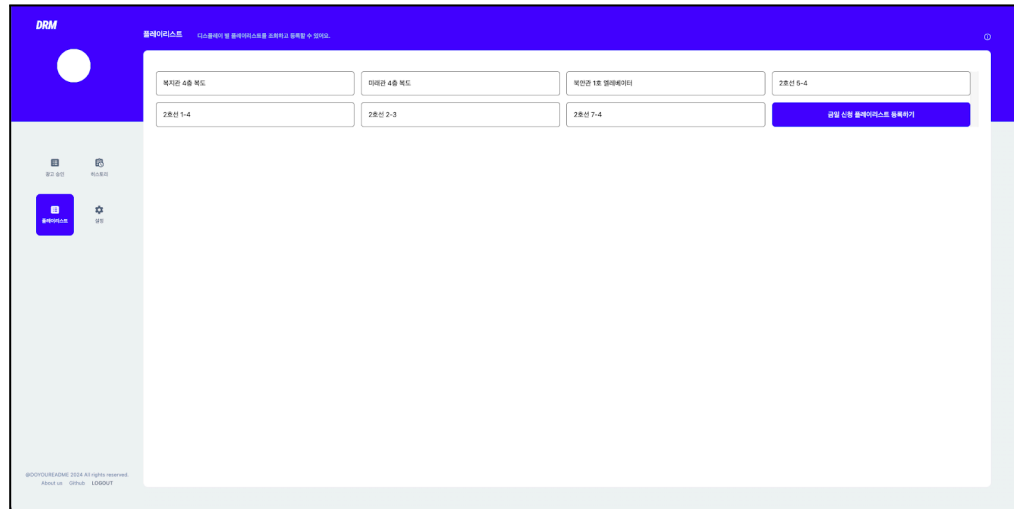
4.2.2 히스토리



모든 사용자(광고주)들이 등록한 광고에 대하여 조회가 가능하다.

4.2.3 플레이리스트

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05



관리자는 플레이리스트 페이지에서 ‘금일 플레이리스트 수동 등록’과 ‘디스플레이 별 송출 중 광고 확인하기’ 두 가지 작업을 수행할 수 있다.

플레이리스트 등록 버튼(파란색 CTA버튼)을 클릭하면 해당 일자에 광고 집행 예정이 되어 있는 광고들을 플레이리스트로 만들어 등록시키고, 각 디스플레이에 송출을 시작한다.

각 디스플레이 이름들이 적힌 텍스트 버튼을 클릭하게 되면, 현재 송출 중인 광고들(플레이리스트)를 확인할 수 있다.

4.2.4. 설정 페이지

사용자용과 마찬가지로 관리자도 설정 페이지 내에서 프로필, 비밀번호 변경, 회원탈퇴를 진행할 수 있다.

5.3 배포 가이드

운영 가이드에 있는 모델이랑 서버 실행방법을 여기다가 적는게 더 적절한거같아요,!

5.3.1 서버

1. 로컬에서 실행

1.1 클라이언트 디렉토리로 이동


```
cd src/server
```

현재 작업 디렉토리를 `src/server`로 변경한다. 프로젝트의 서버 관련 소스 코드가 존재하는 장소이다.

1.2 배포 브랜치로 이동

```
git checkout
```

1.3 도커 컴포즈 파일 빌드 및 실행

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드
	팀 명	DRM
	Confidential Restricted	Version 1.5

`docker-compose up -d`

프로젝트를 빌드하고 실행하는 명령어이다. 해당 명령어 실행시 localhost:8080 으로 서버가 실행 된다.

2. 배포 서버 접근

<http://43.200.7.133:8080/swagger-ui/index.html?urls.primaryName=client>
 현재 서버는 ec2에 배포 되어 위에 Url로 접근이 가능하다.

5.3.2

모델

1. 모델 디렉토리로 이동

`cd src/model`

현재 작업 디렉토리를 `src/model`로 변경한다. 프로젝트의 모델 관련 소스 코드가 존재하는 장소이다.

2. 패키지 설치

`pip3 install -r requirements.txt`

모델 실행에 필요한 파이썬 패키지를 설치한다. 필요한 패키지 목록은 `requirements.txt` 파일에 명시되어 있다.

3. 환경 변수 설정

모델 실행에 필요한 가중치 경로, 백엔드 서버 주소 등의 정보를 설정한다. 필요한 변수 목록과 설명은 `.env` 파일에 기록되어 있으며, 해당 파일에 값을 쓰거나 환경 변수를 직접 설정하면 된다.

4. 모델 서버 실행

`python3 pipeline/main.py`

모델 서버를 실행한다. 카메라로부터 값을 계속 읽어들이며 사람이 등장하는 이벤트를 대기한다. 사람이 등장하는 경우 동영상을 모델로 분석하여 Kafka 메시지 큐를 통해 백엔드 서버로 분석 결과를 전송한다.

5.4 XXX 매뉴얼

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

5.5 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
사용자	로그인	사용자 및 관리자 계정 로그인	<p>초기 진입 페이지에서 아이디와 비밀번호를 기입한다.</p> <ol style="list-style-type: none"> 기입한 아이디로 가입된 이력이 없거나 비밀번호가 틀린 경우 아이디 및 비밀번호에 대한 정규식을 만족하지 못한 경우 기입한 아이디로 가입된 이력이 있고 비밀번호가 일치하는 경우 <p>위 3가지 방법으로 테스트를 진행한다.</p>	<p>1), 2)에 해당하는 경우, 각 상황에 대한 에러 메시지가 출력된다.</p> <p>3)에 해당하는 경우, 사용자 정보가 브라우저 쿠키에 저장되고 사용자는 '/home'으로, 관리자는 '/admin'으로 이동한다.</p>	성공
		자동 로그인	<p>초기 진입 페이지에서 '자동 로그인' 체크박스를 클릭한 뒤 로그인을 수행한다.</p>	<p>브라우저 종료 후, 재접속 시,</p> <ol style="list-style-type: none"> 사용자는 '/home'으로 이동한다. 관리자는 '/admin'으로 이동한다. 	성공
	회원가입	신규 사용자 계정 생성	<ol style="list-style-type: none"> 초기 진입 페이지에서 '회원가입' 텍스트 버튼을 클릭한다. 아이디, 회사명, 비밀번호, 비밀번호 확인을 모두 기입한다. 활성화된 회원가입 버튼을 클릭한다. 	<ol style="list-style-type: none"> 아이디, 회사명, 비밀번호, 비밀번호 확인 미입력 시 에러 텍스트를 출력한다. 기입한 아이디로 가입된 이력이 있는 경우, 에러 	성공



국민대학교
컴퓨터공학부
캡스톤
디자인 I

결과보고서

프로젝트 명

지하철 탑승객 행동 기반 광고 관심도 분석 대시보드

팀 명

DRM

Confidential Restricted

Version 1.5

2024-JUN-05

				텍스트를 출력한다. 3) 1, 2에 해당하지 않는 경우 새로운 계정이 생성된다.	
	광고등록	신규 광고 등록	<ol style="list-style-type: none"> 1) 등록을 원하는 광고 영상을 업로드한다. 2) 광고 집행일, 송출을 원하는 디스플레이, 식별을 위한 광고 타이틀과 설명을 기입한다. 3) 광고 등록 버튼을 누른다. 	승인 대기 상태의 새로운 광고가 생성된다.	성공
		기존 광고 등록	<ol style="list-style-type: none"> 1) 사용자의 히스토리 목록을 조회하며 등록을 원하는 광고를 선택한다. 2) 광고 집행일, 송출을 원하는 디스플레이를 기입한다. 3) 광고 등록 버튼을 누른다. 	승인 대기 상태의 새로운 광고가 생성된다.	성공
	유동인구 정보조회	유동인구 정보 및 유동인구 정보 수집 방법	좌측 '유동인구정보' 메뉴 버튼을 클릭한다.	지하철 2호선의 일평균 유동인구수, 유동인구 성비, 나이대 정보를 화면에 출력한다.	성공
	대시보드	대시보드 목록 조회	좌측 '대시보드' 메뉴 버튼을 클릭한다.	사용자가 등록한 광고 중 관리자에게 승인된 광고를 리스트 형태로 화면에 출력한다.	성공
		대시보드 상세 조회(통합)	<ol style="list-style-type: none"> 1) 대시보드 목록 중 하나의 광고를 선택한다. 	선택한 광고에 대한 대시보드를 제고	
		대시보드 상세 조회(광고단위별)	<ol style="list-style-type: none"> 1) 대시보드 통합 조회 페이지에서 Select 컴포넌트를 통하여 조회 가능한 광고 단위 리스트를 확인한다. 2) 조회하고자 하는 광고 단위 리스트를 클릭한다. 		

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	지하철 탑승객 행동 기반 광고 관심도 분석 대시보드	
	팀 명	DRM	
	Confidential Restricted	Version 1.5	2024-JUN-05

		대시보드 상세 조회(날짜별)			
--	--	-----------------------	--	--	--