

프로젝트 요약서

프로젝트 제목: 모락모락: 개인 맞춤형 헤어스타일을 추천받고 다양한 조건을 비교하여 본인에게 맞는 헤어 디자이너를 찾을 수 있도록 매칭해주는 서비스

날짜	팀명	팀원	학번	서명	팀지도교수
2024. 05. 17	A.A				

요구사항 정리

많은 사람들이 자신만의 개성을 찾아내고자 하는 욕구가 더욱 증가하고 있습니다. 우리 팀은 사람들의 이러한 개성 표현의 한 수단으로 헤어스타일이 큰 역할을 한다고 생각했습니다.

하지만 자신에게 어울리는 헤어스타일이 무엇인지 모르는 사람들이 많아 헤어스타일 선택을 쉽게 하지 못하고, 따라서 헤어스타일을 추천받고 싶어하는 사람들도 많습니다. 실제로 많은 사람들은 어떤 헤어스타일이 어울리는지 스스로 판단하기 어려우며 일반적인 사용자들은 이에 대한 충분한 지식이 없어 헤어스타일 선택에 많은 시간을 소요하게 됩니다. 게다가 헤어스타일을 직접 변경하기 전에는 결과가 눈에 보이지 않고 직접 시도해보는 것에 대한 두려움을 가질 수 있습니다.

헤어 디자이너는 고객들이 자신의 개성과 스타일을 표현할 수 있도록 도와주는 전문가라고 할 수 있습니다. 그러나 헤어 디자이너들의 뛰어난 실력을 알리는 홍보의 장이 부족함을 느꼈습니다. 요즘 인스타그램에서 본인의 작업물을 사진/영상으로 올려 DM(인스타그램에서 제공하는 1:1메세지)으로 예약을 잡는 헤어디자이너들을 자주 볼 수 있습니다. 인스타그램은 사진 및 동영상 공유 소셜 네트워크 서비스라는 광범위한 기능을 제공하기 때문에, 유저들은 본인의 목적에 맞게 가공하여 서비스를 이용하고 있는 것입니다. 하지만 사용법을 잘 모르는 헤어디자이너들은 서비스 활용률이 떨어진다는 점과 가격 안내 및 작업물 정리 프로세스가 분명하지 않아 중간에 헤어 스타일링을 받고자 하는 사람들은 불편함을 느끼고 프로세스에서 이탈할 확률이 높다는 단점이 존재합니다. 또한, 헤어 스타일링을 받고자 하는 사람들은 원하는 디자이너의 작업물(헤어스타일링)이 본인에게 어울릴지 예상해볼 수 없다는 단점이 있습니다.

목표 사용자 :

- 자신에게 어울리는 헤어스타일이 무엇인지 궁금한,
어떤 헤어스타일을 해야 할 지 고민인 20대 여성들
- 자신을 포트폴리오를 브랜딩하고 홍보하고자 하는 헤어디자이너들

관련 연구/서비스/시스템 조사 결과 및 한계점

1. 유사 서비스 비교

1) 헤어핏

- 가상 체험을 통해 다양한 헤어스타일을 체험할 수 있음.
 - 별도 기능으로 주변 디자이너들에게 시술 예약을 할 수 있음.
 - 그러나, 전체적으로 합성 과정이 어색하며 결과물이 자연스럽지 못함.
 - '개인 맞춤 스타일'을 제공해주지는 않고, 정해진 머리스타일을 여러 개 합성해봐야 하는 번거로움이 있음.
 - 또한 합성을 했다고 해서 그게 그 헤어스타일을 잘 하는 디자이너와 바로 연결되지 않음.
- 따라서 등록된 디자이너들이 많아도 디자이너들에 대한 홍보 효과는 부족함.

2) 카카오헤어샵

- 헤어 디자이너들의 포트폴리오를 확인하기에는 유용한 서비스.
- 스토리북 기능이 있어 마음에 드는 헤어스타일의 사진을 클릭 시 어떤 헤어 디자이너가 그 머리를 했는지 확인할 수 있음.
- 하지만 개인 헤어스타일 추천 시스템은 존재하지 않으며, 오직 디자이너들의 홍보에 초점을 맞춘 느낌임.

2. 결론:

- 사용자 맞춤 헤어스타일을 추천해주는 서비스는 존재하지 않는다.
- 특히, 헤어디자인에 가장 핵심적인 요소인 얼굴형을 고려한 서비스는 더더욱 존재하지 않는다.
- 헤어스타일을 가상으로 체험해봐도 해당 헤어를 잘하는 디자이너를 추천받을 수 있는 앱은 없다. 즉, 사용자들은 가상 헤어 기능까지만 체험하고 맞춤 디자이너까지 찾아보지는 않으므로, 현존하는 앱들을 이용해 디자이너들은 고객 유치가 힘들다.

따라서, 개인에게는 맞춤 헤어스타일을 추천하고, 그에 기반한 헤어디자이너 추천 기능을 모두 제공하는 차별화된 서비스, ‘모락모락(毛樂毛樂)’을 제안함.

제안 내용
헤어스타일에 고민이 많은 20대 여성 사용자를 대상으로 5가지 얼굴형(heart, oblong, oval, round, square)에 따라 어울리는 헤어스타일과 어울리지 않는 헤어스타일을 각 3개씩 정의한 후, 서비스 내 자체 헤어 합성 AI 기술을 이용하여 스타일링에 참고할만한 결과물을 제공합니다. 잘 어울리는 헤어스타일링 3가지와 헤어 디자이너의 전문분야가 일치하는 헤어 디자이너의 포트폴리오를 리스트 형태로 추천받습니다. 헤어샵 또는 자신의 작업물을 어떻게 홍보해야 할지 어려움을 겪는 헤어 디자이너들에게 포트폴리오를 서비스에 제출하여 홍보할 기회를 제공합니다. 신규 헤어디자이너들에게는 광고 신청 기회를 제공함으로써 사용자에게 노출빈도를 올릴 수 있도록 합니다. 전체 디자이너들은 작성한 포트폴리오를 공유 링크로 만들어 예비 서비스 사용자가 디자이너의 포트폴리오에 쉽게 접근할 수 있게 함으로써 고객 유치에 도움을 받을 수 있습니다.

최종 기능

1. 사용자 얼굴형 판단 후 어울리는/어울리지 않는 헤어 합성 사진 제공 : 사용자의 얼굴 이미지를 통해 얼굴형(하트형, 긴형, 계란형, 사각형, 둥근형)을 분석한 후, 분석된 얼굴형에 어울리는 헤어스타일을 합성한 사진 3장과 어울리지 않는 헤어스타일을 합성한 사진 3장을 가시적으로 보여주는 기능이다.
2. 사용자에게 추천된 헤어스타일 관련 디자이너 목록 제공 : 위의 기능에서 추천된 헤어스타일을 기반으로 그 헤어스타일을 전문 헤어로 등록한 디자이너들을 리스트 형태로 제공하는 기능이다.
3. 디자이너 포트폴리오 생성 : 디자이너가 자신의 정보, 근무지, 자격인증서, 전문헤어, 헤어 이미지, 가격 등을 넣어 자신만의 포트폴리오를 생성하는 기능이다.
4. 디자이너 리스트 조회 및 검색 : 자신에게 맞는 디자이너들을 리스트 형태로 모아보고 검색할 수 있는 기능이다.
5. 디자이너 세부 정보 확인 : 디자이너 리스트에서 원하는 디자이너를 눌러 디자이너가 생성한 포트폴리오 내용을 상세하게 확인할 수 있는 기능이다.
6. 디자이너 포트폴리오 찜하기 : 마음에 드는 디자이너 포트폴리오를 찜할 수 있는 기능이다.

개발 진척 과정 중 발생 이슈

1. 효율성 문제 해결 :

초기에는 얼굴형 분석 및 얼굴 합성을 위하여 **facial landmarks** 모델과 **stargan-v2**를 사용하였습니다. 하지만 **facial landmarks** 모델을 사용할 때 사용자 얼굴의 특징점 판단이 메인인 모델을 이용함과 동시에 모락모락 팀 측에서 정한 수치를 이용하여 얼굴형을 판단하도록 하였기 때문에 정확도가 좋지 못했습니다.

또한 **Stargan-v2**를 사용할 때 너무 오랜 빌드 시간이 걸리고, 얼굴형을 합성하여 결과 사진을 보았을 때 매우 어색하다는 단점을 발견하였습니다. 따라서 얼굴형을 분석을 약 5000장의 사진으로 학습시킨 **EfficientNet**활용하도록 변경하였습니다. 또한 얼굴 합성 방법도 **FaceSwap**으로 변경하여 시간을 단축시켰습니다.

2. 비용 발생 문제 해결 :

1) 백엔드:

2024년 2월부터 **IPv4** 주소 고갈에 대한 이슈가 떠오르며 **AWS**에서는 **IPv4** 1개에 대해서만 무료로 지원을 합니다. 따라서, 작업을 하며 비용적인 부분에 부담을 느끼고 배포 아키텍처를 새롭게 구성해보며 비용을 절약해보고자 합니다. 퍼블릭 액세스 없이 **RDS**를 **EC2**에서 **SSH**방법으로 연결함으로써 배포환경에서 월마다 추가될 **Rds IPv4**에 대한 비용(대략 5000원)을 줄이는 것을 목표로 세웠습니다. 도메인 서비스의 프록싱을 이용한 직접 연결 방법으로 **Route53**과 **Loadbalancer**를 이어 배포함으로써 **IPv4**에 대한 비용(대략 5000원)을 줄이는 것을 목표로 세웠습니다.

2) AI :

AWS EC2와 **github action**, **Docker**를 사용하여 **model**을 이용하는 것들을 모두 배포하고자 하였습니다. 하지만 **EC2**를 이용하여 배포할 시에 **gpu**를 사용하면 학생 신분에서 너무 과한 비용이 청구되는 것을

인지하였습니다. 따라서 **google cloud platform**에서 제공하는 크레딧 40만과 GPU자원을 이용하여 배포하는 것을 목표를 세웠습니다.다만 **gpu**를 사용할 시 많은 크레딧이 사용되기 때문에 중간발표 전후로는 **google cloud platform**을 사용할 것이고 이후로는 학교에서 지원해주고 있는 **tencent cloud**를 사용하여 진행할 예정입니다.

구현 방법

1. AI

Kaggle 에서 얼굴형을 5가지 **Heart, Oblong, Oval, Round, Square** 로 분류한 **dataset**을 이용하여 **colab**을 통해 얼굴형 판단 **EfficientNet** 모델을 파인튜닝합니다.

그 후, 학습시킨 **EfficientNet-B4** 모델을 이용하여 사용자의 얼굴형 예측을 진행하고자 하였습니다. 우선 사용자로부터 이미지를 입력받아 이를 전처리하여 모델에 적합한 형식으로 전환합니다. 그 다음 전처리된 이미지를 입력으로 받아 얼굴형을 예측하는 함수를 정의합니다. 예측된 얼굴형과 가장 낮은 확률로 예측된 얼굴형에 대해 사용자 얼굴 사진과 미리 데이터를 넣어놓은 헤어스타일 사진을 합성하여 이미지를 생성하고, 이 결과 이미지를 **AWS S3**에 업로드 하고 **presigned URL** 기능을 통해 **URL**만 다운로드 하여 사용할 수 있게 합니다. 이는 전부 **python** 베이스의 코드로 작성되었고 **flask**를 사용하였습니다. 이를 **github**에 업로드 한 후 **google cloud platform**에서 제공하는 **compute engine**의 가상머신 인스턴스를 생성한 후, 가상환경을 생성하여 **github**의 레포지토리를 **clone** 하였습니다. 그 후 **flask** 를 실행하여 외부 **IP**에서 해당 프로젝트에 접속할 수 있도록 하였습니다. 인스턴스에서는 **GPU**를 할당받아 서비스의 속도를 높이하고자 하였습니다.

2. 백엔드

프론트엔드로부터 들어온 **API** 요청 및 유저 인증 로직을 **Spring Boot**를 사용해 코드로 작성하였습니다. 카카오톡 유저인증(회원가입, 로그인)부터 헤어디자이너의 포트폴리오 작성, 필터링, 드롭다운 기능 및 일반 손님의 얼굴형에 어울리는 헤어디자이너 추천, 헤어디자이너 포트폴리오 좋아요 및 조회 등의 **API**에 대한 로직을 **Spring Security** 프레임워크를 통해 인증하였습니다. 이를 통해 데이터보호에 유념하며 작업하였습니다. 전체 코드를 **Docker image**로 만들어 **Docker Hub**로 푸시하도록 구현했으며 **github action**을 통한 **CI/CD** 파이프라인을 구성함으로써, 매 푸시(**push**)마다 자동으로 업데이트된 내용을 포함해 서버 재배포가 되게 하였습니다. (이때 로컬과 배포환경을 구분해야 했기에 **docker-local**과 **docker-compose**를 따로 만들어 작업했습니다.) 서버는 **AWS**의 **EC2**에 도커 이미지를 **pull**해 작업환경과 같은 배포환경을 만들어 사용하였습니다. 데이터베이스로 **RDS**를 **private** 서브넷에 둔 후 **EC2**에서 **SSH**연결을 함으로써 접근하여 사용했으며, 서버 **IP**주소에 **Route53**을 통해 만들어진 도메인 네임을 연결함으로써 보안과 비용적인 측면에서 신경쓰며 프론트와 통신할 수 있었습니다.

3. 프론트엔드

사용자 친화적인 **UI/UX**를 구상해 구현하는 데 초점을 맞췄으며, 적절한 애니메이션 효과를 넣었습니다. 유저 인증 과정에서 빠르고 효과적인 인증 과정을 위해 **Next.js**의 미들웨어 기능을 활용하였습니다. 또한 서버로부터 사진 등의 크기가 큰 리소스를 불러오는 데 지연 시간을 줄이기 위하여 서버사이드 렌더링을 최대한 활용하려 노력하였습니다. 디자이너가 스스로를 홍보할 수 있는 화면(디자이너 포트폴리오 등록, 디자이너 리스트, 디자이너 상세 페이지)와 사용자 플로우(가상 헤어 체험, 디자이너 리스트 필터링 페이지, 좋아요 누르기 등) 기능 및 화면을 구현하였습니다. 마지막 배포는 **vercel**을 이용해서 진행하였습니다.

기대효과 및 의의

이 프로젝트를 통해 사용자들은 자신의 얼굴형에 대한 이해와 헤어스타일 선택에 도움을 받을 수 있으며, 헤어 디자이너들과의 연결까지 제공받을 수 있습니다.

자신의 얼굴형 판단 결과에 대한 호기심을 갖고, 동시에 자신의 얼굴형에 맞는 헤어스타일을 예측하며 추천받고 싶어하는 **20대 여성들이** 얼굴형 예측 결과를 제공받을 수 있습니다. 뿐만 아니라 그에 따른 헤어스타일과 사용자 얼굴 합성 사진을 다양하게 제공받을 수 있습니다. 동시에 자신에게 추천되는 헤어스타일을 주력으로 하는 디자이너를 추천받고 디자이너의 포트폴리오와 관련된 정보를 제공받을 수 있습니다.

디자이너들은 자신의 헤어스타일 작품을 모아 포트폴리오로 생성할 수 있으며 이를 서비스의 다양한 사용자들에게 노출시킴으로써 자신을 홍보할 수 있고 나아가 고객을 확보하는 지름길이 될 수 있습니다.

목표

- 얼굴형 판단 모델의 정확도를 **90%** 이상으로 학습시키도록 하겠습니다.
- 사용자 얼굴과 헤어스타일을 합성시키는 과정이 **1분** 내로 완료되도록 하겠습니다.

프로젝트 기술서

요구사항 정의

주요 기능들에 대한 서비스 요구사항은 다음과 같습니다.

1. 얼굴형 분석 후 헤어스타일 합성 사진 제공

- 필요한 **API**로는 “사용자 얼굴 이미지 업로드 **API**”, “얼굴형을 분석하고 서버로 보내는 **API**”, “합성 사진 생성 후 이를 저장하는 **API**”가 있습니다.
- 입력으로는 사용자가 직접 등록하게 될 사용자 얼굴 이미지가 필요하고 최종 출력으로는 사용자 이미지를 통해 분석된 얼굴형 결과와 합성 사진들이 나오게 됩니다.
- 이때 필요한 모델은 이미지 분류를 위한 **EfficientNet**과 얼굴 합성 사진 생성을 위한 **MobileFaceSwap**이 있습니다.
 - EfficientNet**의 활용을 위해 **Kaggle**의 다섯 종류의 얼굴형 분류 이미지 데이터 약 **5000장**을 이용하여 **Colab**을 통해 이 모델을 학습하는 과정이 필요합니다.
 - 학습된 **EfficientNet** 모델은 사용자의 얼굴형을 분석하여 가장 일치하는 확률이 높은 얼굴형과 가장 일치하는 확률이 낮은 얼굴형을 결과로 도출하기 위해 필요합니다.
- 필요한 데이터는 아래와 같습니다.

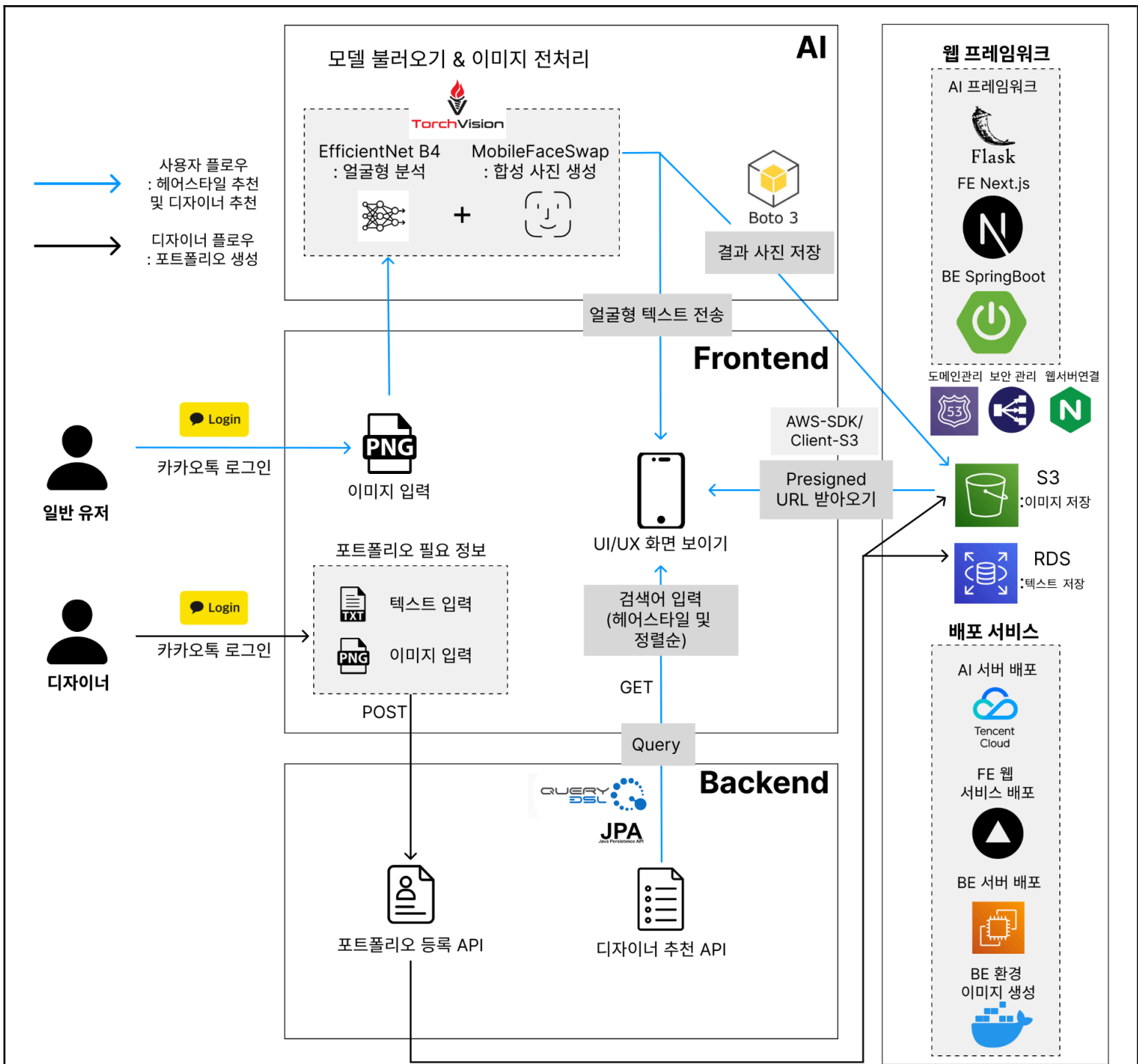
- i. 첫째로 라벨링된 얼굴형 데이터셋이 필요합니다. 얼굴형 라벨은 총 5가지로 분류되었으며, **FACESHape_HEART**로 지정된 하트형, **FACESHape_OBLONG**으로 지정된 긴형, **FACESHape_ROUND**로 지정된 둥근형, **FACESHape_SQUARE**로 지정된 사각형, 그리고 **FACESHape_OVAL**로 지정된 계란형이 있습니다.
- ii. 두번째로는 각 얼굴형에 대해 지정된 헤어스타일 이미지 데이터셋이 필요합니다. 하트형에는 "단발 C컬펌", "보브컷", "아이롱펌"이 있으며, 긴형에는 "레이어드컷", "히피펌", "중단발 레이어드펌"이 있습니다. 둥근형에는 "스트레이트", "숏컷", "태슬컷"이 있으며, 사각형에는 "단발 레이어드펌", "샤밍컷", "허쉬컷"이 있고, 계란형에는 "사이드뱅", "장발 C컬펌", "빌드펌"이 있습니다. 해당 데이터셋 이미지들은 사용자가 등록한 사진을 합성할 때 사용됩니다.
 - 1. 얼굴형에 어울리는 헤어스타일 지정을 위한 기준은 「얼굴형에 따른 헤어스타일 연구」, 「얼굴형의 특성을 보완한 헤어스타일 연구」, 「얼굴형에 따른 헤어스타일 연출 제안」, 「얼굴 유형에 따른 헤어 커트 스타일 선호도 연구」와 같은 논문들이 필요합니다.
- e. 해당 기능을 구현하기 위해 **python** 베이스의 코드 작성이 필요하며, 이때 파이썬 웹 프레임워크인 **flask**가 필요합니다. 그 후 배포를 위해 클라우드 서비스인 **tencent cloud**가 필요합니다.

2. 디자이너 추천 및 디자이너 상세 페이지

- a. 필요한 **API**로는 "포트폴리오 등록 **API**", "포트폴리오 상세 정보 조회 **API**", "디자이너 추천 **API**"가 있습니다.
 - i. "포트폴리오 상세 정보 조회 **API**"에서는 디자이너 별 포트폴리오를 조회하기 위해 포트폴리오 **ID**를 파라미터로 넘기는 것이 필요합니다.
- b. 디자이너가 등록한 포트폴리오 데이터가 필요합니다. 포트폴리오 작성 시, 디자이너는 아래와 같은 정보를 입력하게 됩니다.
 - i. 이름, 근무지, 근무지 전화번호, **SNS** 주소, 한 줄 소개, 프로필 이미지, 작업물 4개, 시술 가격표 정보 작성이 필요합니다.
 - ii. 승인 절차를 위해 '미용사 일반 국가 자격증' 제출이 필요합니다.
 - iii. 디자이너는 전문 헤어스타일을 3가지 선택을 위한 **ENUM**타입 리스트가 필요합니다. 제공되는 리스트는 아래와 같습니다.
 - 1. "단발 C컬펌", "보브컷", "아이롱펌", "히피펌", "레이어드컷", "중단발 레이어드펌", "스트레이트", "숏컷", "태슬컷", "단발 레이어드펌", "샤밍컷", "허쉬컷", "사이드뱅", "장발 C컬펌", "빌드펌"
- c. 얼굴형 분석 후, 가장 유사도가 높다고 판단된 얼굴형에 지정된 헤어스타일 3가지에 대한 이름정의가

필요합니다. 이름정의는 아래와 같이 구성됩니다.

- i. ('FACESHape_HEART', "단발 C컬펌"), ('FACESHape_HEART', "보브컷"), ('FACESHape_HEART', "아이롱펌")
 - ii. ('FACESHape_OBLONG', "레이어드컷"), ('FACESHape_OBLONG', "히피펌"), ('FACESHape_OBLONG', "중단발 레이어드펌")
 - iii. ('FACESHape_ROUND', "스트레이트"), ('FACESHape_ROUND', "숏컷"), ('FACESHape_ROUND', "태슬컷")
 - iv. ('FACESHape_SQUARE', "단발 레이어드펌"), ('FACESHape_SQUARE', "샤밍컷"), ('FACESHape_SQUARE', "허쉬컷")
 - v. ('FACESHape_OVAL', "사이드뱅"), ('FACESHape_OVAL', "장발 C컬펌"), ('FACESHape_OVAL', "빌드펌")
- d. 포트폴리오에 등록된 전문 헤어 분야와 헤어스타일 네이밍을 텍스트를 매칭하는 query문 작성이 필요합니다.
- i. 빠른 탐색을 위해 QueryDSL 라이브러리를 사용합니다.
 - ii. 체계화된 매칭을 위해 데이터베이스를 정규화합니다.
 - 1. 정규화 결과, Portfolio 테이블, Hairstyles 테이블, Portfolio_Hairstyles 테이블(Portfolio 테이블과 Hairstyles 테이블을 1:N 연결을 위한 테이블)이 필요합니다.
- e. 이미지 저장을 위해서는 AWS S3가 필요하며, S3에서 발급받은 presigned URL들은 RDS로 보내는 과정이 필요합니다. 텍스트 데이터 저장 시에는 이를 RDS로 보내는 과정이 필요합니다.
- f. 해당 기능을 구현하기 위해 JAVA 베이스의 코드 작성이 필요하며, 이때 자바 웹 프레임워크인 Spring Boot가 필요합니다. 그 후 배포를 위해 클라우드 서비스인 AWS가 필요합니다.



1. 전체 SW 아키텍처 설명

공통

1. 사용자가 **morakmorak.vercel.app**에 접속하면, 프론트에서 제공하는 UI화면을 보여줍니다.
2. 카카오톡 로그인을 한 후, 일반 사용자와 헤어 디자이너 중 역할을 선택함으로써 이 후 기능이 달라집니다.

사용자 입장

1. 일반 사용자의 경우, 본인의 정면 얼굴 이미지를 업로드 후 얼굴형에 맞는 헤어스타일링을 추천받을 수 있습니다.

이때, 얼굴형 판단 및 추천/비추천 헤어스타일과의 합성 시스템은 **tencent cloud** 에서 제공하는 **compute engine**의 **gpu**를 포함한 인스턴스를 통해 배포한 파이썬 기반의 코드와 **EfficientNet B4**, **FaceSwap** 모델을 통해 진행됩니다. **Efficient B4**를 통해 얼굴형이 분석되며, **MobileFaceSwap**을 통해 합성 사진이 생성됩니다. 이는 **torchvision**을 통해 모델 불러오기와 이미지 전처리를 하면서 진행됩니다.

2. **EfficientNet B4**에서 얼굴형 분석 후, 사용자와 유사도가 가장 높은 얼굴형과 낮은 얼굴형은 프론트엔드로 보냄으로써 사용자의 데이터를 쌓아두고, 어울리는/어울리지 않는 헤어와 본인의 얼굴이 합성된 총 6장의 이미지들은 **Boto3**라는 **Python**용 **AWS SDK**를 사용하여 **AWS S3**로 저장됩니다.

3. 이 후, 프론트엔드에서는 **AWS S3**에 저장된 6장의 이미지를 **AWS-SDK/Client-S3**을 사용해 객체 접근용 **presigned Url**을 받아옵니다. 받아온 이미지들과 **AI** 서버로부터 받아온 얼굴형 텍스트 결과를 이용해 사용자에게 분석 및 합성 결과를 화면에 제공합니다.

4. 그 후, 프론트엔드는 어울리는 헤어스타일과 해당 헤어스타일을 전문으로 하는 디자이너를 텍스트 매칭하여 디자이너 포트폴리오를 추천하는 **API**를 호출합니다.

5. 이 때, **QueryDSL**와 **JPA** 라이브러리를 사용해 작성해둔 쿼리문을 거쳐 해당 데이터를 프론트로 넘겨줍니다.

6. 프론트엔드에서의 화면을 통해 추천된 디자이너들이 리스트 형태로 보여집니다.

헤어디자이너 입장

1. 헤어 디자이너의 경우, 포트폴리오 생성 화면에서 포트폴리오를 작성함으로써 본인의 작업을 홍보할 수 있습니다.

2. 포트폴리오 생성에 필요한 정보를 텍스트와 이미지로 입력하고, 각각 **RDS**와 **S3**에 저장합니다.

3. 작성, 수정, 삭제 기능 등의 **api**는 백엔드에서 **AWS Route53**을 거쳐 만든 도메인으로 **post**, **get**, **delete**를 호출하면 됩니다.

4. 이때 **AWS Loadbalancer**를 거치며 **HTTPS**로 변경하기 위해 **AWS Certificate Manager**를 붙임으로써 **TLS** 보안을 적용하고 라우팅을 분산적으로 가능하게 합니다.

5. **Nginx**를 거쳐 웹서버를 띄우고 이를 도커에 패키징해서 배포(**push/pull**)함으로써, 다른 **OS** 환경에서도 같은 환경으로 돌아갈 수 있게 설정합니다.

2. 주요 모듈

1) **EfficientNet** : **Kaggle**에서 가져온 얼굴형 분류 이미지 데이터 약 5000장을 활용하여 학습된

EfficientNet B4 모델은 입력받은 사용자 얼굴 이미지를 토대로 **FACESHape_HEART**,

FACESHape_OBLONG, **FACESHape_ROUND**, **FACESHape_SQUARE**, 그리고

FACESHape_OVAL 중에서 가장 일치하는 얼굴형과 가장 일치율이 낮은 얼굴형의 결과를 도출할 수

있습니다. 이 모델은 얼굴형 분석 기능을 구현하기 위해 사용되며, **python** 기반 코드를 통해

실행됩니다. 이 모델을 사용하기 위해서는 **python** 코드에서 모델을 불러오는 코드를 통해 모델을

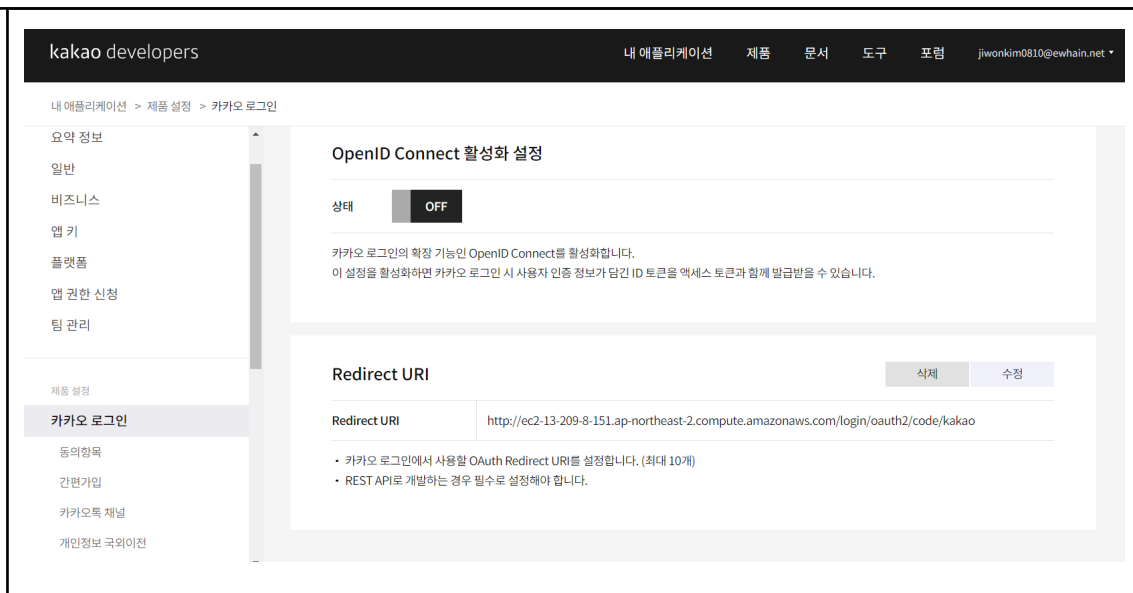
불러온 후, 입력받은 사용자 이미지를 전처리 하는 함수를 거쳐 모델에 전처리된 이미지가 입력으로

들어가고, **predicted_class**와 **least_likely_class_label**로 가장 일치하는 얼굴형과 가장 일치율이 낮은

얼굴형을 텍스트로 출력하도록 코드를 구현하였습니다.

- 2) **MobileFaceSwap** : **MobileFaceSwap**은 비디오에서 얼굴 스와핑을 위하여 만들어진 경량 프레임워크입니다. 이 프레임워크는 서비스에서 얼굴 이미지와 헤어스타일 이미지를 합성하는 기능을 수행하기 위해 필요합니다. 직접 구현한 **python** 코드와 기존에 존재하는 이 프레임워크를 통해 가장 일치하는 얼굴형과 가장 일치율이 낮은 얼굴형과 연관된 헤어스타일 이미지를 사용자 얼굴 이미지와 합성할 수 있도록 하였습니다. 이 모듈은 얼굴형 분석 이후의 단계에서 사용됩니다. 분석 단계에서의 함수를 통해 반환되는 가장 일치하는 얼굴형과 가장 일치율이 낮은 얼굴형 텍스트를 데이터 베이스의 사진 분류 폴더 이름과 비교하는 함수를 통해, 해당하는 폴더에 담긴 헤어스타일 이미지를 **ref_image**로 지정하고 초기에 입력받은 후 전처리 된 사용자 이미지를 **target_image**로 하여 **target_image**의 이목구비와 **ref_image**의 헤어스타일 부분이 분리되어 서로 합성된 이미지를 만들 수 있도록 구현하였습니다.
- 3) **Docker** : **SpringBoot** 환경을 컨테이너에 담아 다른 OS환경에서도 돌아가게 하기 위해 애플리케이션을 실행할 기본환경, 종속성, 명령 등을 포함한 **Dockerfile**을 작성합니다. **Dockerfile**을 사용하여 **Nginx** 웹서버와 **Spring Boot** 애플리케이션 이미지를 빌드합니다. 이때 **Nginx**는 웹서버 소프트웨어로 클라이언트로부터 **HTTP**요청을 **Spring Boot**서버로 전달하기 위해 필요합니다. 준비된 **Docker** 이미지를 **DockerHub**에 **Push** 한 후 **AWS EC2** 가상서버에서 **Docker** 이미지를 **Pull** 하여 **Spring Boot** 애플리케이션을 배포 및 실행합니다.
- 4) **Kakao developers API** : **Kakao Developer** 사이트에 가입하고 새로운 애플리케이션을 등록하여 클라이언트 ID와 인증정보를 제공받습니다. **Spring Security**를 사용하여 애플리케이션의 사용자 인증 및 권한 제어를 실행합니다. 클라이언트가 카카오톡 서버로 인가 요청을 보내면 받은 인가토큰을 서버가 받아 서버내에서 자체적으로 **JWT** 토큰을 생성한 후에 클라이언트 측으로 전송함으로써 세션 내 사용자의 상태를 유지합니다. 이를 통해 간편한 로그인 프로세스를 구현했으며, 일반 사용자와 디자이너의 권한을 제어함으로써 보안적인 측면을 강화했습니다.

Kakao developers API 사이트 실제 화면



3. 필요 기술

기술 검증을 통해 확정된 파트별 기술 스택은 아래와 같습니다.

1) AI

- kaggle face shape 데이터셋
- flask: 파이썬 웹 프레임워크
- torch, torchvision : 딥러닝 모델을 위한 PyTorch 패키지
- AWS S3 : 이미지 저장 및 불러오기
- Boto3 : 파이썬에서 AWS를 이용하기 위해 사용
- EfficientNet : 심층 신경망 아키텍처, 이미지 분류를 위해 사용
- MobileFaceSwap : 이미지 간의 얼굴을 교환하는 모델

2) 프론트

- Next.js : SSR(Server Side Rendering)을 지원하는 웹 프레임워크
- Typescript: Javascript의 확장 언어로 정적 타입을 지원해 안정성과 생산성이 높은 언어
- TailwindCSS: 웹 UI를 디자인하기 위한 CSS 프레임워크
- Vercel: 웹 애플리케이션 및 정적 웹사이트를 배포하고 호스팅할 수 있는 클라우드 배포 서비스
- Yarn Berry: 프로젝트의 종속성을 관리하는 도구로, 프로젝트 실행 시 필요한 라이브러리를 효율적으로 가져올 수 있음
- AWS-SDK/Client-S3: 클라이언트에서 직접 AWS S3의 객체를 접근할 수 있는 라이브러리

3) 백엔드

- **Spring Security**: 인증과 권한 부여 및 보호 기능을 체계적으로 적립하기 위해 사용
- **Lombok**: 어노테이션을 통해 메소드를 자동으로 생성하기 위해 사용
- **Jpa**: 자바에서 **OMR** 기술 표준으로 사용되는 인터페이스를 쓰기 위해 사용
- **Oauth2**: 외부 카카오 **api** 인증 및 권한 부여를 위해 사용
- **Swagger**: **Rest API**들의 목록을 확인하고 자동으로 명세서를 만들기 위해 사용
- **Jsonwebtoken**: 백엔드측 토큰을 만들어 클라이언트에게 제공함으로써 로그인을 유지하기 위해 사용
- **Mustache**: 서버측에서 돌아가는 **api**를 **ui**적으로 테스트해보기 위해 사용
- **Kakao developers API**: 카카오 로그인을 쓰기 위해 사용

주요 엔진(Core Engine, Main Part등)의 기능 및 설계

1. 사용자 얼굴형 판단 및 이미지 합성

해당 기능에서 사용되는 모듈들은 **EfficientNet B4** 모델, **MobileFaceSwap** 프레임워크, **flask** 웹 프레임워크, **PIL** 라이브러리, **torchvision.models**, **torchvision.transforms**, **AWS-SDK/Client-S3** 등이 있습니다.

모든 과정에 앞서, 얼굴형을 계란형, 둥근형, 긴형, 하트형, 사각형으로 라벨링하고 각각 **FACESHape_OVAL**, **FACESHape_ROUND**, **FACESHape_OBLONG**, **FACESHape_HEART**, **FACESHape_SQUARE** 로 지정합니다. 이후 5가지 얼굴형에 어울리는 헤어스타일을 각 3가지씩 지정합니다. 얼굴형에 적절한 헤어스타일은 「얼굴형에 따른 헤어스타일 연구」, 「얼굴형의 특성을 보완한 헤어스타일 연구」, 「얼굴형에 따른 헤어스타일 연출 제안」, 「얼굴 유형에 따른 헤어 커트 스타일 선호도 연구」와 같은 논문을 기반으로 선정하였습니다. 이 각각의 헤어스타일에 맞는 사진을 데이터베이스에 각각 저장하도록 합니다. 이 기능에 대한 대부분의 코드는 **python** 언어로 **flask** 웹 프레임워크 상에서 작성되었습니다. 이렇게 코드 구현 전에 필요한 데이터 처리 과정이 마무리 됩니다.

코드 구현은 다음과 같이 진행되었습니다. **torchvision.models** 모듈을 사용하여 미리 **colab**에서 파인튜닝한 **EfficientNet B4** 모델을 불러옵니다. 사용자의 이미지를 업로드할 수 있는 함수를 작성하여 사용자로부터 이미지를 입력으로 받아오도록 구현합니다. **torchvision.transforms**를 통해 사용자가 업로드한 이미지를 전처리합니다. 그 후, **EfficientNet B4** 모델을 사용하여 사용자의 얼굴형을 예측한 결과를 반환하고, 가장 확률이 낮게 예측된 얼굴형 결과 또한 반환하도록 하는 함수를 작성합니다. 직접 정의한 **post_json_data** 함수를 이용하여 이 결과를 백엔드 서버측으로 **json** 형태로 전송합니다. 이때 전송된 결과는 데이터베이스에 텍스트 형태로 저장됩니다. **AI** 서버에서는 그 결과를 이용하여 데이터베이스에 저장되어있는 헤어스타일 이미지를 불러온 후 **MobileFaceSwap** 에서 제공하는 **pretrained model**과 **image_test.py** 코드를 통해 앞서 전처리된 사용자 얼굴 이미지와, 불러온 헤어스타일 이미지를 합성합니다. 이때 가장 확률이 높게 예측된 얼굴형으로 분류된 헤어스타일 이미지와 합성을 해서 결과를 만들어내고, 가장 확률이 낮게 예측된 얼굴형으로 분류되어있는 헤어스타일 이미지와도 합성을 해서 결과를 만들어낸다. 이렇게 총 6장의 이미지가 생성되면, **python**에서 **AWS S3**를 사용할 수 있는 라이브러리를 이용하여 작성한 코드를 통해 이미지들을 **S3** 버킷에 저장합니다. **AWS S3**에 저장된 이미지들은 프론트엔드에서 **AWS-SDK/Client-S3** 라이브러리를 통해 직접 가져와 사용자에게 제공합니다.

2. 디자이너 추천 및 디자이너 상세페이지

해당 기능에서 사용되는 모듈은 **queryDSL**와 **JPA**가 있습니다. 기능 구현을 위해 우선 프론트엔드에서 사용자의 얼굴과 가장 일치하는 얼굴형 텍스트 결과를 **URL**의 쿼리로 **AI** 서버로부터 넘겨받습니다. 이후, 프론트엔드 측에서 해당 사용자 얼굴형에 어울리는 헤어스타일 이름 3가지를 다중값으로 백엔드 서버로 넘깁니다. 백엔드 서버는 넘겨받은 정보를 콤마(,)를 기준으로 파싱한 후, **Hairstyle** 테이블에서 **hair_name**부분을 탐색합니다. **hair_name**에 해당하는 **hair_style_id** 리스트를 받아, **Portfolio_Hairstyles** 테이블에서 해당 **hair_style_id**를 가진 **portfolio_id** 리스트를 받아옵니다. 앞 로직을 구현하는 쿼리문을 실행시키는 함수를 **queryDSL** 라이브러리를 사용하여 직접 구현함으로써 **portfolio_id**를 가진 **portfolio** 객체를 가져와 디자이너 포트폴리오 리스트로 사용자에게 제공합니다. 이 과정에서, **queryDSL**을 사용하여 **JPA** 엔티티에 대한 쿼리를 쉽게 작성하기 위해 **JPA** 라이브러리를 사용합니다. 이로써 사용자는 자신에게 어울리는 헤어스타일링을 잘하는 디자이너를 추천 받아볼 수 있습니다. 사용자가 디자이너 포트폴리오 리스트에서 특정 포트폴리오를 클릭하면 “포트폴리오 상세 정보 조회 **API**”와 함께 해당 포트폴리오 **ID**가 백엔드 서버로 **GET** 메소드가 요청되어 포트폴리오 세부내용을 조회할 수 있습니다.

Portfolio_Hairstyles 테이블

portfolio_hair_style_id	hair_style_id	portfolio_id
4	3	2
5	12	2
6	15	2
19	1	3
20	10	3
21	13	3

Hairstyle 테이블

hair_style_id	face_shape	hair_name
1	FACESHape_HEART	단발 C컬펌
2	FACESHape_HEART	보브컷
3	FACESHape_HEART	아이롱펌
4	FACESHape_OBLONG	레이어드컷
5	FACESHape_OBLONG	히피펌
6	FACESHape_OBLONG	중단발 레이어드펌
7	FACESHape_ROUND	스트레이트
8	FACESHape_ROUND	숏컷
9	FACESHape_ROUND	태슬컷
10	FACESHape_SQUARE	단발 레이어드펌
11	FACESHape_SQUARE	샤밍컷
12	FACESHape_SQUARE	허쉬컷
13	FACESHape_OVAL	사이드뱅
14	FACESHape_OVAL	장발 C컬펌
15	FACESHape_OVAL	발드펌

평가 항목	평가 기준	평가 내용
기능 완성도	모듈 실행 시간 1분 이내	헤어스타일 합성 시 시간이 1분 소요되어 기존 3분에 비해 약 66% 감소했습니다.
	모델 정확도 90% 이상	얼굴형을 판단하는 모델의 정확도가 훈련을 통해 약 5% 상승했습니다.
	API 속도 1ms 이내	검색 / 헤어디자이너 추천 API 의 경우 QueryDSL 라이브러리를 통해 1초 소요되어 기존 5초에 비해 약 80% 감소했습니다.
	이미지 캐싱 속도 30ms 미만	웹 배포 프레임워크(Next.js)에서 제공하는 이미지 캐시 기능을 이용해 첫 시도 이후 이미지 로드 시간을 약 95% 감소했습니다.
개인정보 및 데이터 보안	사용자별 인증 및 데이터 보호 여부	카카오 외부 API 에서 제공하는 OAuth 2.0 프로토콜을 사용해 사용자를 인증하고 접근자원을 제어했습니다. 스프링 시큐리티에서 제공하는 HTTPS 와 BCrypt 해시 함수를 사용하여 데이터를 안전하게 전송 및 저장했습니다.
프로젝트 전체 완성도	목표 달성 여부	1차 보고서에서 제안하였던 주요 6 가지 기능을 전부 구현하여 연결하였고 서비스 배포까지 완료하였습니다.


결론
<div> 모락모락: 개인 맞춤형 헤어스타일 추천 및 헤어 디자이너 매칭 서비스 </div> <p> 많은 사람들이 자신만의 개성을 표현하고자 하는 욕구가 증가하고 있는 가운데, 헤어스타일은 큰 역할을 한다고 판단했습니다. 그러나 많은 이들은 자신에게 맞는 헤어스타일을 찾는데 어려움을 겪고있지만 현재 시장에서는 사용자의 얼굴형을 고려한 맞춤형 헤어스타일 추천 서비스가 부족한 상황입니다. 또한, 디자이너의 홍보와 자신의 작업물을 모아보기 위한 서비스도 존재하지 않았습니다. 따라서 저희 팀은 개인 맞춤형 헤어스타일 추천과 헤어 디자이너 매칭 서비스 '모락모락'을 개발했습니다. </p>

모락모락은 자신에게 어울리는 헤어스타일에 고민이 많은 **20대 여성**과, 포트폴리오를 만들어서 자신을 홍보하고자 하는 헤어디자이너들을 타겟으로 합니다. 모락모락은 해당 타겟층에게 아래와 같은 주요 기능들을 제공합니다. 사용자 얼굴형 판단 후 어울리지/어울리지 않는 헤어 합성 사진 제공, 사용자에게 추천된 헤어스타일 관련 디자이너 목록 제공, 디자이너 포트폴리오 생성, 디자이너 리스트 조회 및 검색, 디자이너 세부 정보 확인 그리고 디자이너 포트폴리오 찜하기 기능입니다. 이러한 기능들의 구현을 위해 프론트엔드의 경우 서비스 화면의 **UI/UX 디자인** 및 서비스 화면 구현을 담당하였으며, 백엔드의 경우 **ERD 설계, API제작, 명세서 작성, 서버 배포, 데이터베이스 관리** 등의 역할을 담당했습니다. 또한 **AI** 파트의 경우 모델 훈련, 학습 및 데이터 수집과 모델을 배포하여 서비스에 적용하는 역할을 담당하였습니다.

저희 '모락모락' 서비스는 현재 **20대 여성** 유저를 타겟팅 하고있지만, 점차 연령대와 성별을 늘려 서비스를 확장할 것입니다. 또한 예약 기능과 커뮤니티 기능을 추가해 고객들의 니즈를 잡을 예정입니다.

최종 산출물

모락모락은 헤어스타일 선택이 어려운 사용자에게 후보를 추천해주고 관련 합성 이미지를 제공하며, 동시에 헤어 디자이너 추천을 제공하는 웹서비스로, 아래와 같은 기능과 화면을 제공할 예정입니다.



나만의 헤어스타일을 정해보는 **즐거움**
나만의 포트폴리오를 만들어보는 **즐거움**

카카오로 시작하기

모락모락이 처음이시군요!
먼저 **회원가입**을 진행해주세요.

닉네임

성별

여성

약관 동의

필수 서비스 이용약관

필수 개인정보 수집 및 이용동의

선택 제 3자 정보 제공 동의

사용자입장

모자람없는인생님은
무엇이든 찰떡인 **계란형**



정방형의 얼굴 사진을 중앙에
위치하게 해주세요!

인생 머리를 찾을 준비가 되셨나요?

네!

모락모락이 추천하는
BEST 헤어스타일

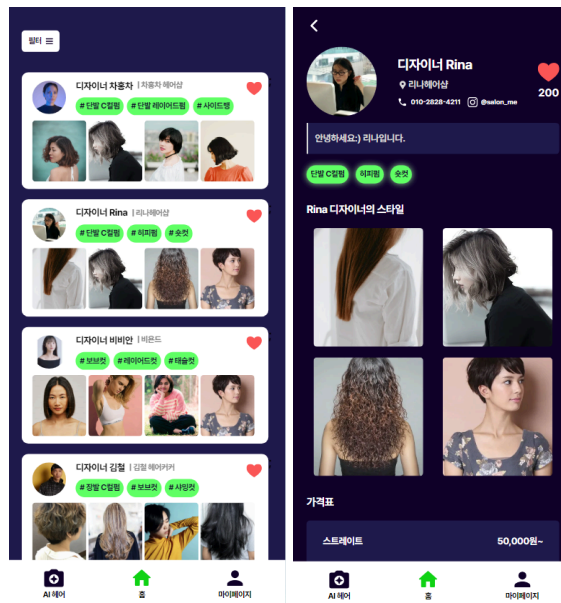


나에게 꼭 맞는 헤어 유형

직선의 짧은 길이의 헤어 혹은
볼 스트레이트 헤어가 잘 어울려요!

추천하는 헤어스타일

<일탈> 볼드형 사이드뱅



디자이너 입장

환영합니다, 디자이너님!
포트폴리오를 등록해주세요.

+

이름

다음으로

디자이너님을 표현할 수 있는
한 줄 소개가 있을까요?

한 줄 소개

SNS(선택)

모락미용실 모락모락점

다음으로

현재 디자이너님이 계신
근무지에 대한 정보를 알려주세요.

근무지

정확한 지명명까지 포함해주세요.

전화번호

근무지점의 전화번호를 입력해주세요.

다음으로

평소 디자이너님의 작업이 궁금해요!
전문 스타일, 작업물을 공유해주세요.

전문 헤어
평소 자신있는 전문 헤어스타일을 3개 골라주세요

사이드컷

마이크로링

직자컷

스트레이트

단발 C컷

장발 C컷

단발 웨이퍼드컷

중단발 웨이퍼드컷

보브컷

숏컷

레이어드컷

사랑컷

사이드백

태슬컷

포트폴리오 등록
디자이너님만의 헤어스타일 작품들을 4장 공유해주세요!

다음으로

고객 분들이 참고할 수 있는
가격표가 있다면, 작성해주세요.

가격 입력

디자이너님만의 헤어스타일 가격대를 4개까지 작성해주세요.

일반형

20,000원 ~

+

다음으로

거의 다 작성해주셨어요!
마지막으로 미용 자격증을 올려주세요!

+

다음으로

축하합니다!
무사히 포트폴리오를 등록하셨습니다!
잠시 후 메인 화면으로 이동합니다.