

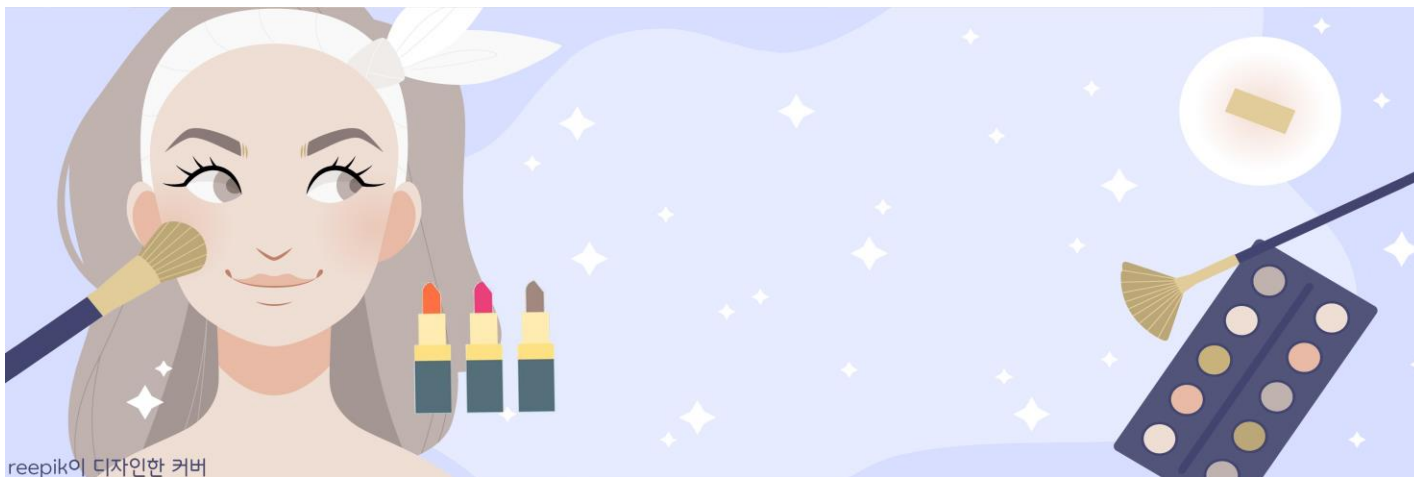


Smart MakeUp

AR 기술을 활용한 화장 미리보기 서비스를 제공하는 웹 서버



최준원
한기윤
현호성



목차

- 프로젝트 개요
- 서비스 배경
- 목표 및 목표 달성 방법
- 기술 스택 및 설계
- 프로젝트 기능
- 회고
- Q&A

목차

1. 프로젝트 개요

2. 서비스 배경

- 비대면/온라인 시장 발전

3. 스택 및 설계

- 사용 기술 스택
- 프로젝트 관리 방식 : 애자일 방법론 응용
- USECASE 다이어그램, UI 프로토타입
- DATABASE 설계

4. 프로젝트 기능

- 데이터베이스 연동
- 회원 기능
- 화장하기 기능 구현
- 화장하기 기능 상세 (화장 적용할 영역 찾기)
- 화장하기 기능 상세 (프레임을 처리하여 화장을 구현)
- 화장하기 기능 상세 (서버 간 통신)
- 화장 추천하기 기능
- 게시판 기능 (화장품 톡톡)

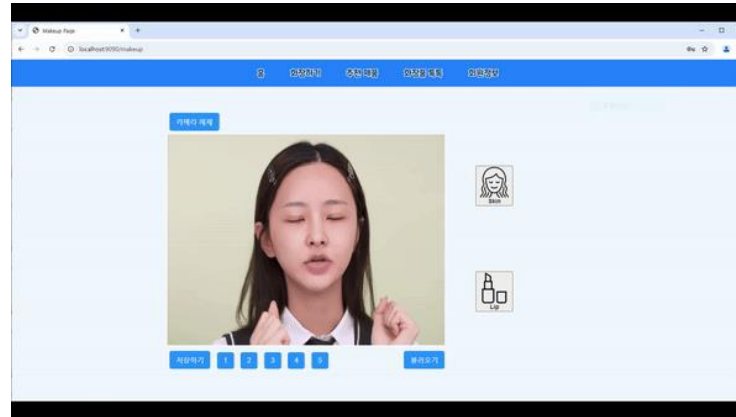
5. 회고

- 기술적 개선사항 : CUDA 사용
- 기술적 개선사항 : Python으로 작성된 영상처리를 C++로 변경
- 기술적 개선사항 : Java process의 FastAPI 동작 문제
- 비즈니스 개선사항

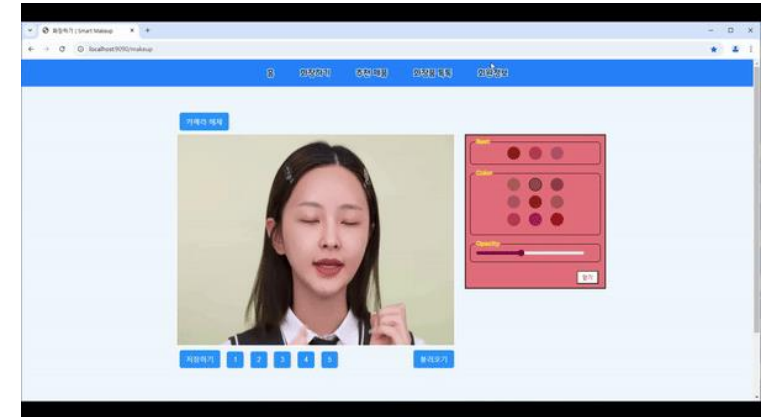
프로젝트 개요

Smart MakeUp

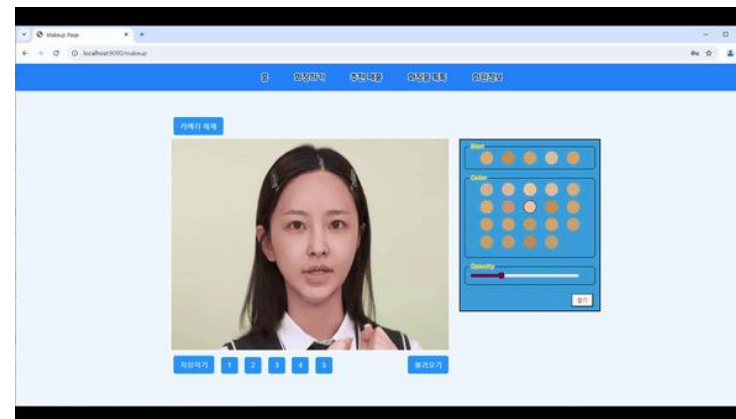
웹 서버로
AR기술을 활용하여
화장 미리보기 기능을 구현



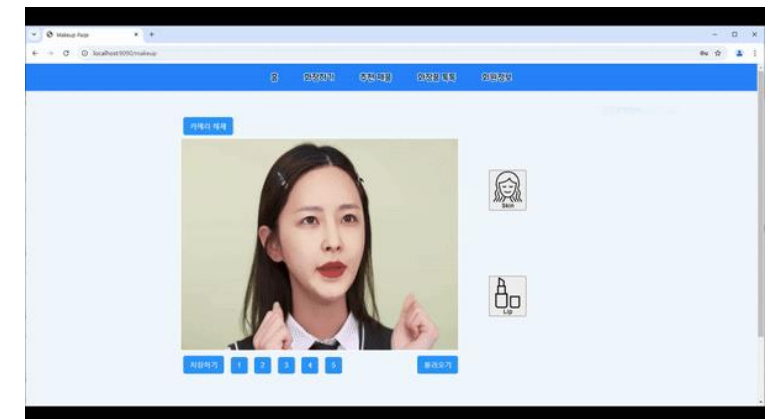
화장 적용 이전



입술 화장 적용



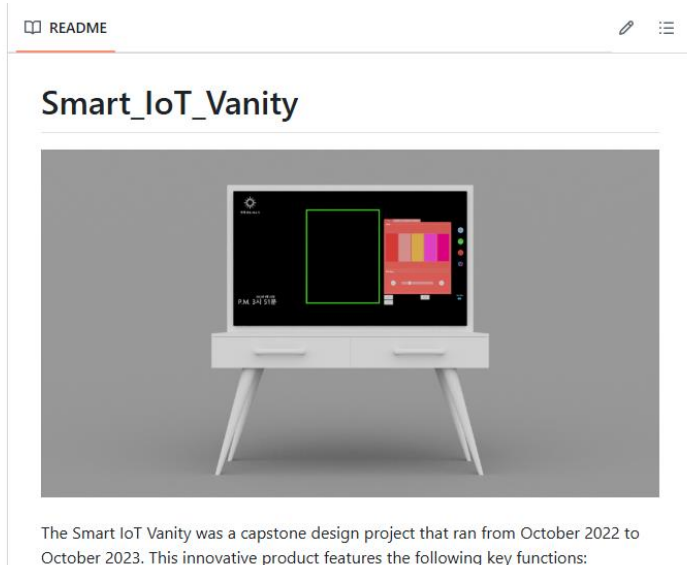
스킨 화장 적용



전체 화장 적용

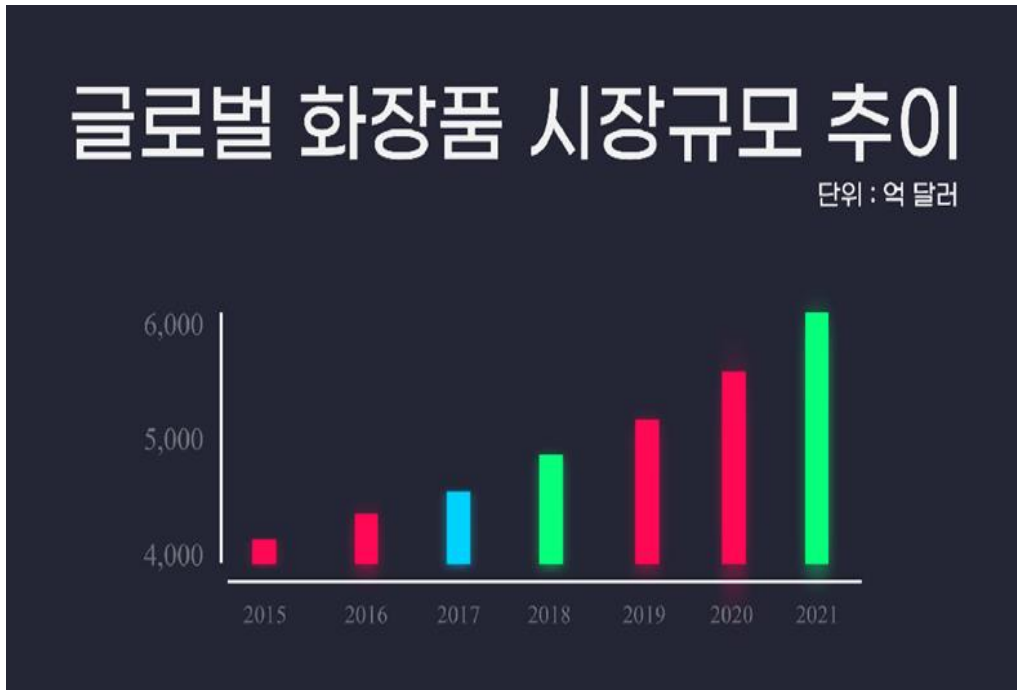
프로젝트 개요

이전 UI로 개발한 Smart IoT Vanity를 웹 서버로 이식



서비스 배경

비대면/온라인 시장 발전



화장품 산업은
온라인을 중심으로 급성장하고 있다.

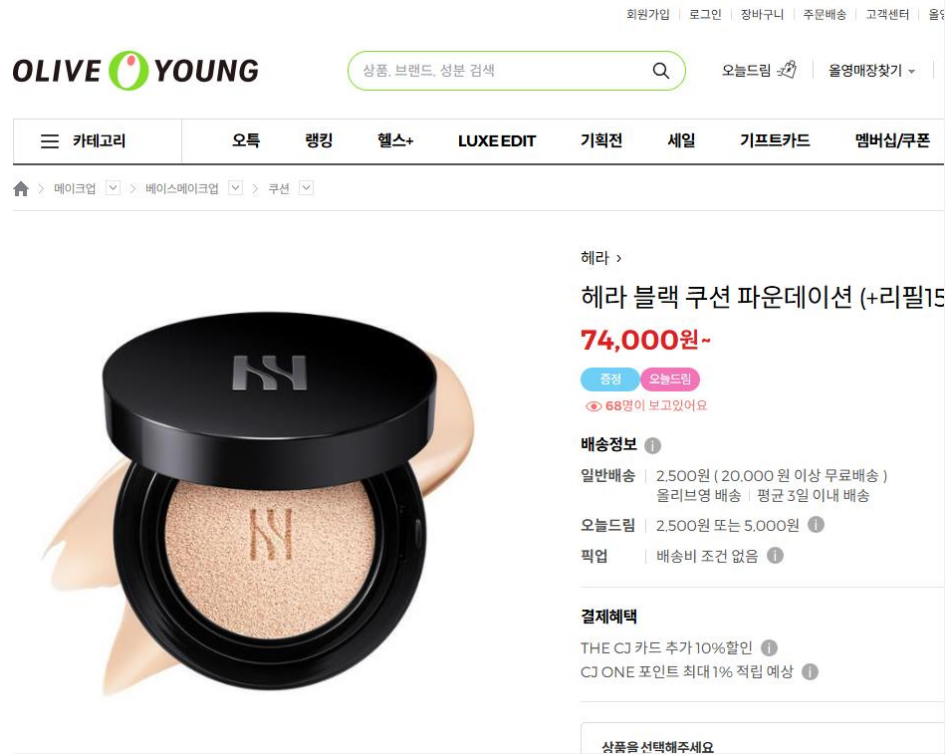


서양에서는 이미
증강 현실(AR) 기술과 AI를 활용한
맞춤형 뷰티 서비스가 주요 트렌드로 떠오르고 있다.

서비스 배경

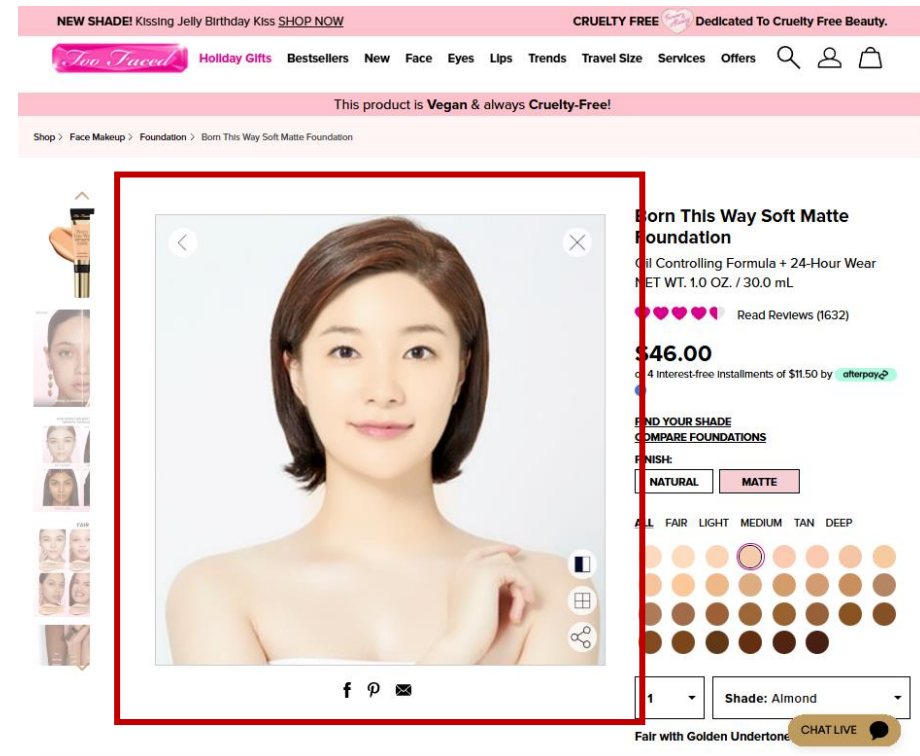
국내외 온라인 시장 비교

국내 쇼핑 플랫폼 "올리브영"



AR 화장 서비스 **미제공**

미국의 뷰티 사이트인 "Too Faced"



AR 화장 서비스를 **제공**

기술 스택 및 설계

사용 기술 스택

Front-End



Back-End



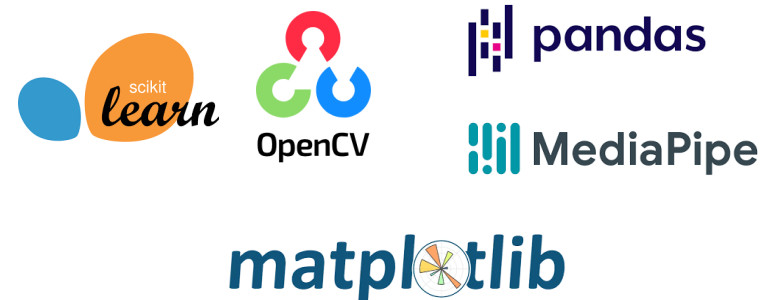
Collaboration



Database

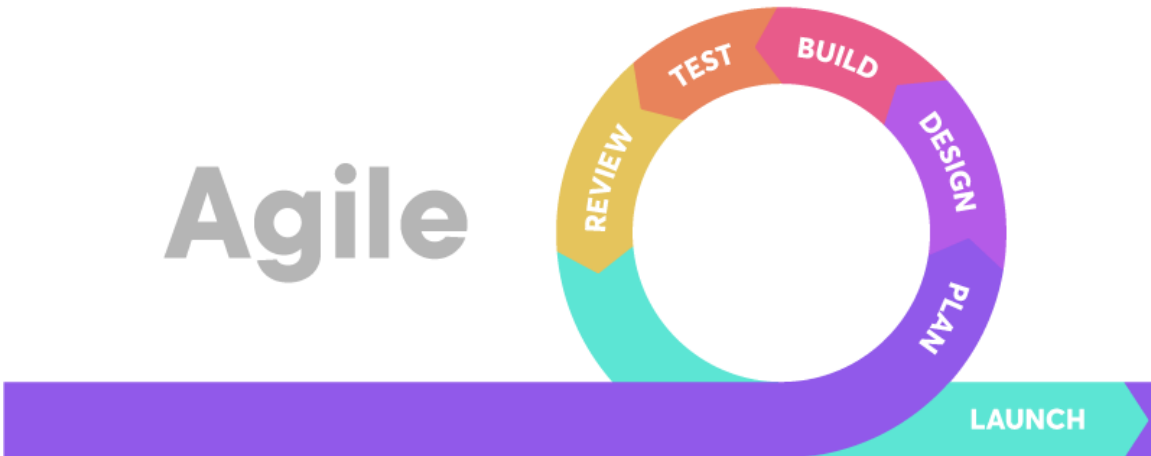


Video

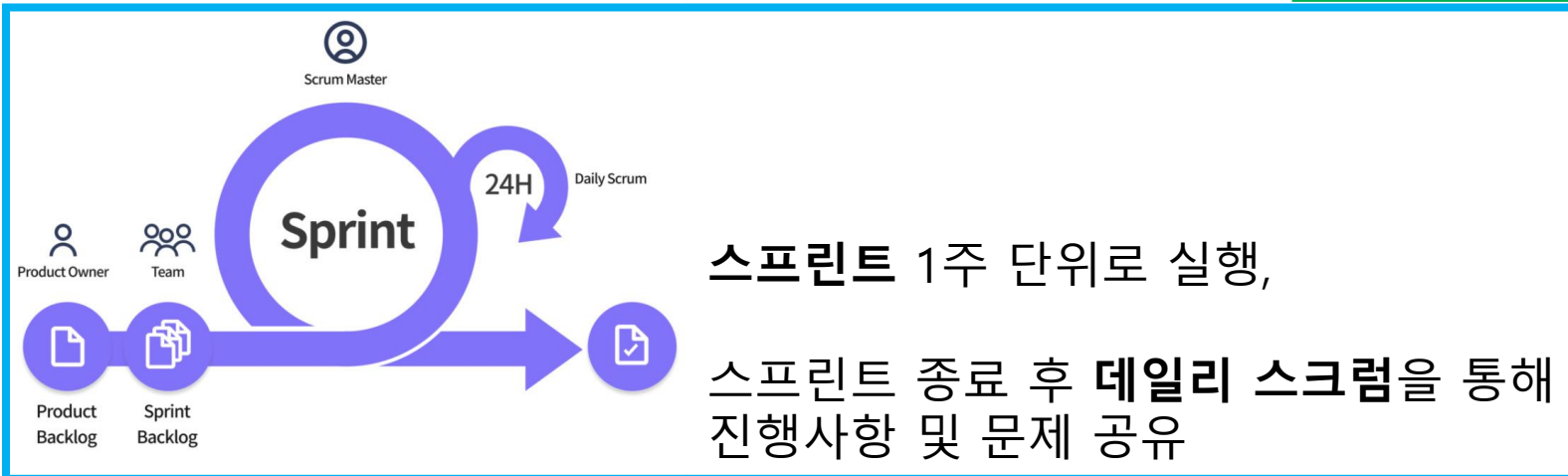


기술 스택 및 설계

프로젝트 관리 방식: 애자일 방법론을 응용



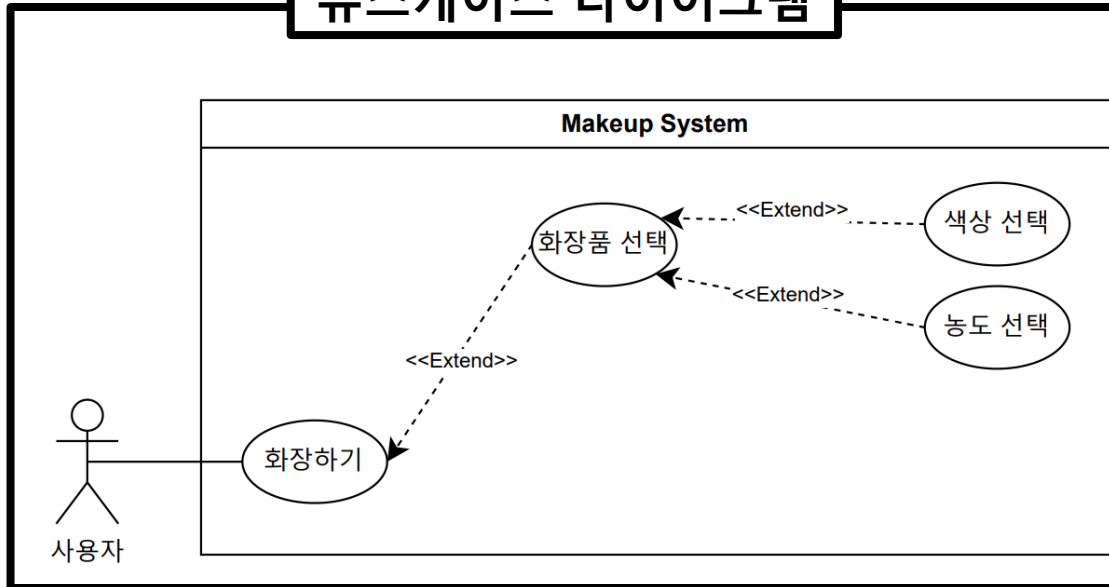
절차	추진 일정				기간
	9월	10월	11월	12월	
프로젝트 주제 선정					09/19-09/19
프로젝트 계획서 작성					09/20-09/23
업무 기능도					09/24-09/26
단위업무 기술서					09/27-09/30
업무 흐름도					10/01-10/03
구현					10/04-11/15
버그 수정					11/16-12/04



기술 스택 및 설계

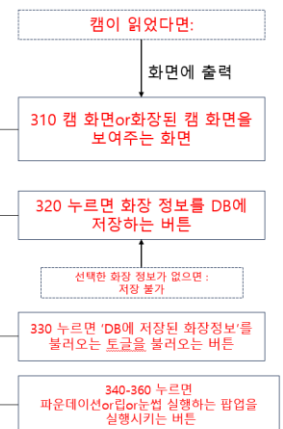
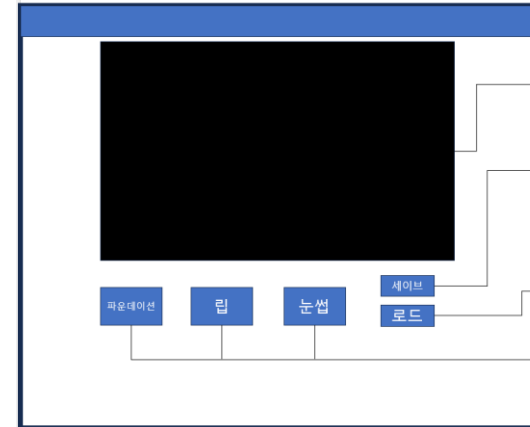
USECASE 다이어그램, UI 프로토타입 설계

유스케이스 다이어그램

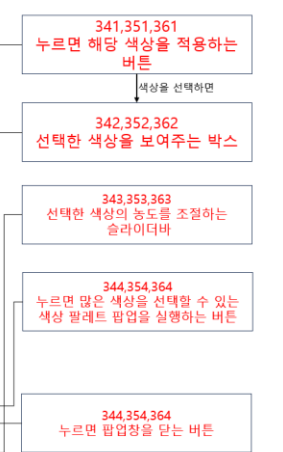
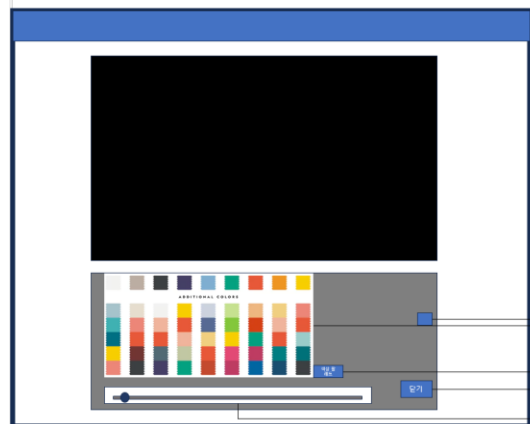


UI 프로토타입

300 "화장하는법" 으로 가는 버튼



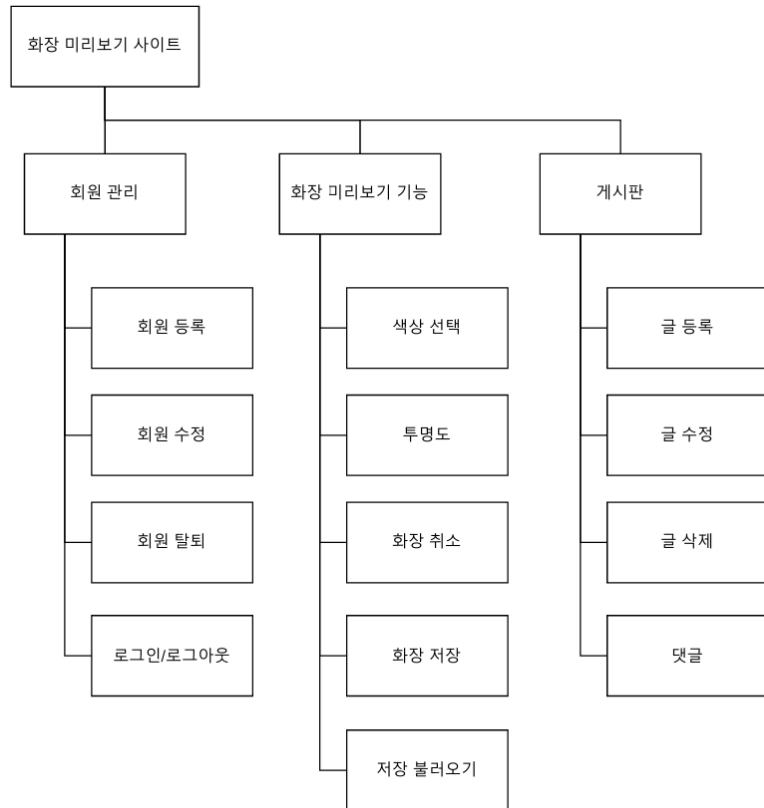
340,350,360 파운데이션or립or눈썹 실행하는 팝업



기술 스택 및 설계

Database 설계

논리적 데이터베이스 설계



물리적 데이터베이스 설계

member	
member_id	varchar(50)
member_password	varchar(100)
email	varchar(50)
phone	varchar(20)

comment	
comment_id	bigint
board_id	bigint
member_id	varchar(50)
comment_content	text

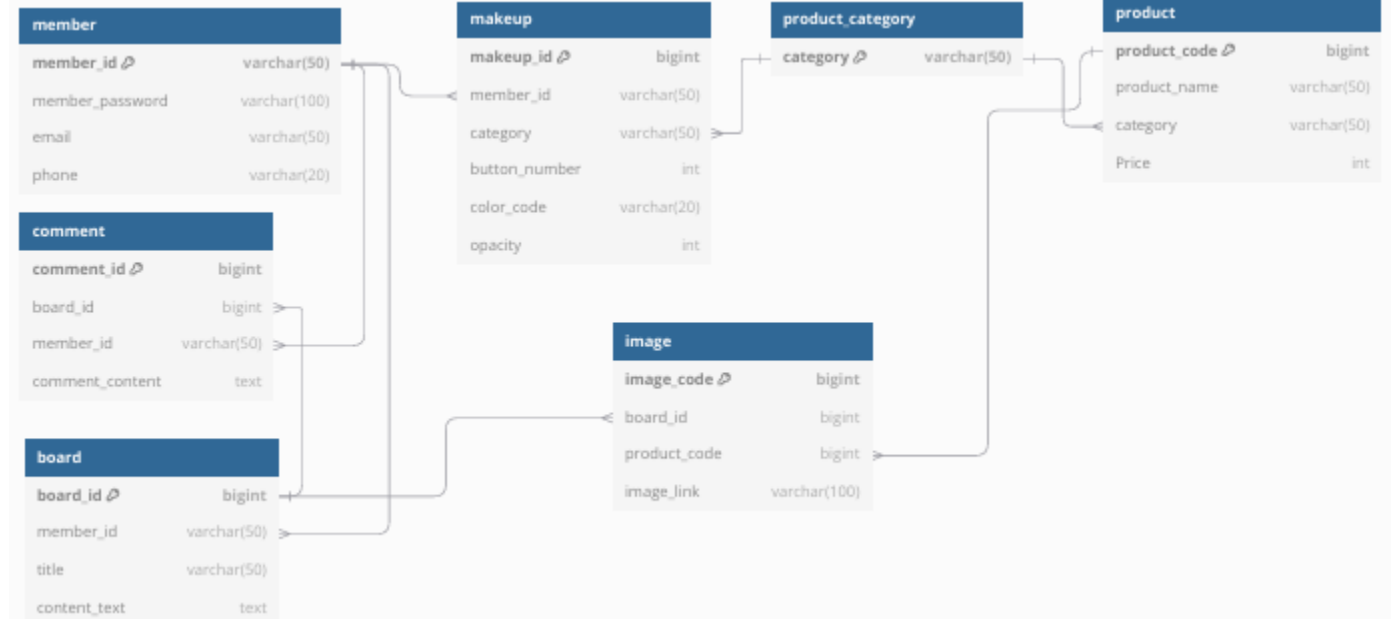
board	
board_id	bigint
member_id	varchar(50)
title	varchar(50)
content_text	text

makeup	
makeup_id	bigint
member_id	varchar(50)
category	varchar(50)
button_number	int
color_code	varchar(20)
opacity	int

product_category	
category	varchar(50)

product	
product_code	bigint
product_name	varchar(50)
category	varchar(50)
Price	int

image	
image_code	bigint
board_id	bigint
product_code	bigint
image_link	varchar(100)



프로젝트 기능

데이터베이스 연동

Spring Boot

```
// 회원 테이블
@Entity
@Table(name = "member")
@Getter
@Setter
public class Member {
    @Id
    @Column(name = "member_id", updatable = false, length = 50)
    private String member_id;    // 사용자 아이디

    @Column(name = "Member_password", nullable = false, length = 100) // not null 추가
    private String Member_password;    // 사용자 비밀번호

    @Column(name = "email", nullable = false, length = 50)
    private String email;

    @Column(name = "phone", length = 20)
    private String phone;

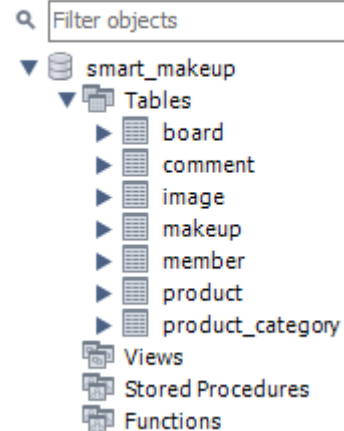
    @OneToMany(mappedBy = "member", cascade = CascadeType.ALL)
    private Set<MakeUp> makeups;

    // 생성자
    public Member() {}
    public Member(String member_id, String Member_password, String email, String phone) {
        this.member_id = member_id;
        this.Member_password = Member_password;
        this.email = email;
        this.phone = phone;
    }

    public void setMember(MemberDTO userDTO) {
        this.member_id = userDTO.getMember_id();
        this.Member_password = userDTO.getMember_password();
        this.email = userDTO.getEmail();
        this.phone = userDTO.getPhone();
    }
}
```

MySQL

SCHEMAS



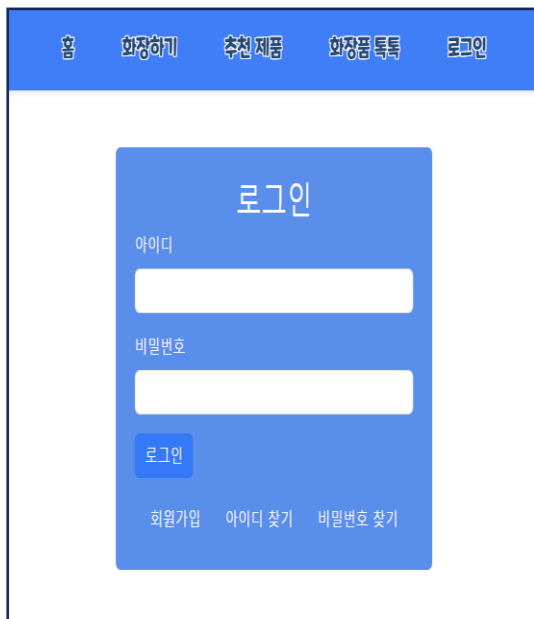
```
Query 1 x  makeup  product  member  product  member
Limit to 1000 rows

1 • CREATE DATABASE smart_makeup;
2 • USE smart_makeup;
3
4 • CREATE TABLE member ( -- 사용자 테이블
5     member_id VARCHAR(50) NOT NULL UNIQUE,
6     member_password VARCHAR(100) NOT NULL,
7     email VARCHAR(50) NOT NULL,
8     phone VARCHAR(20),
9     PRIMARY KEY(member_id)
10 );
11
12 • CREATE TABLE product_category (
13     category VARCHAR(50) NOT NULL UNIQUE,
14     PRIMARY KEY(category)
15 );
16
17 • CREATE TABLE makeup ( -- 화장 정보를 저장하기 위한 테이블
18     makeup_id BIGINT AUTO_INCREMENT PRIMARY KEY,
19     member_id VARCHAR(50),
20     category VARCHAR(50),
21     button_number INT,
22     color_code VARCHAR(20) NOT NULL,
23     opacity INT NOT NULL,
24     FOREIGN KEY (member_id) REFERENCES member (member_id) ON DELETE CASCADE,
25     FOREIGN KEY (category) REFERENCES product_category (category) ON DELETE CASCADE
26 );
```

Spring Boot와 MySQL 연동

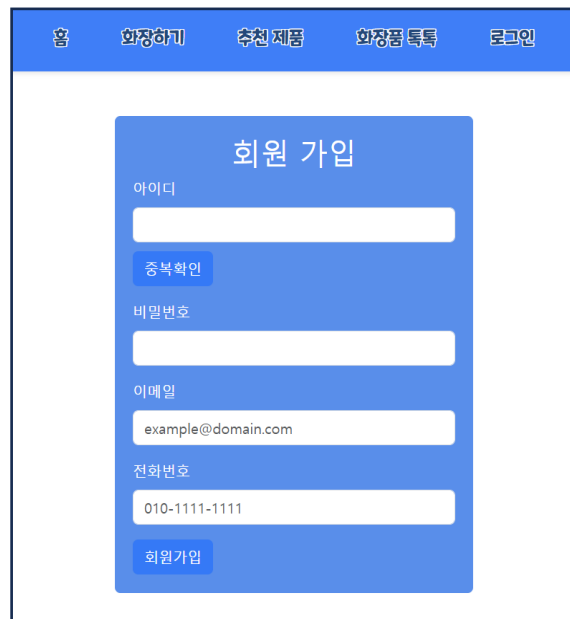
프로젝트 기능

회원 기능



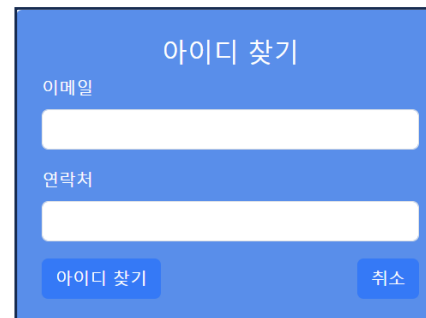
The login form is a white rectangular box with a blue header. The header contains a navigation bar with links: '홈' (Home), '화장하기' (Makeup), '추천 제품' (Recommended products), '화장품 목록' (Cosmetics list), and '로그인' (Login). The main content area has a title '로그인' (Login) in blue. Below the title are two input fields: '아이디' (ID) and '비밀번호' (Password). A blue '로그인' (Login) button is positioned below the password field. At the bottom, there are three links: '회원가입' (Sign up), '아이디 찾기' (Find ID), and '비밀번호 찾기' (Find password).

로그인 기능

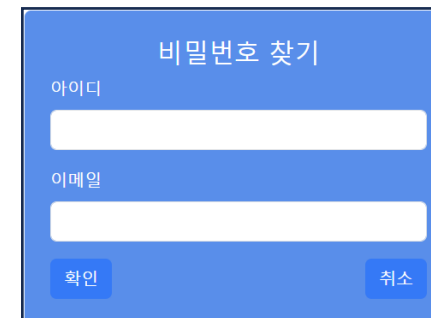


The sign-up form is a white rectangular box with a blue header. The header contains a navigation bar with links: '홈' (Home), '화장하기' (Makeup), '추천 제품' (Recommended products), '화장품 목록' (Cosmetics list), and '로그인' (Login). The main content area has a title '회원 가입' (Sign up) in blue. Below the title are four input fields: '아이디' (ID), '비밀번호' (Password), '이메일' (Email), and '전화번호' (Phone number). A blue '회원가입' (Sign up) button is positioned below the phone number field. A blue '중복확인' (Check duplicate) button is positioned below the ID field.

회원가입 기능



The find ID form is a blue rectangular box with a title '아이디 찾기' (Find ID) in white. Below the title are two input fields: '이메일' (Email) and '연락처' (Contact). A blue '아이디 찾기' (Find ID) button is positioned below the contact field. A blue '취소' (Cancel) button is positioned below the email field.



The find password form is a blue rectangular box with a title '비밀번호 찾기' (Find password) in white. Below the title are two input fields: '아이디' (ID) and '이메일' (Email). A blue '확인' (Confirm) button is positioned below the email field. A blue '취소' (Cancel) button is positioned below the ID field.

아이디 / 비밀번호 찾기 기능

프로젝트 기능

화장하기 기능 구현



화장 적용 이전



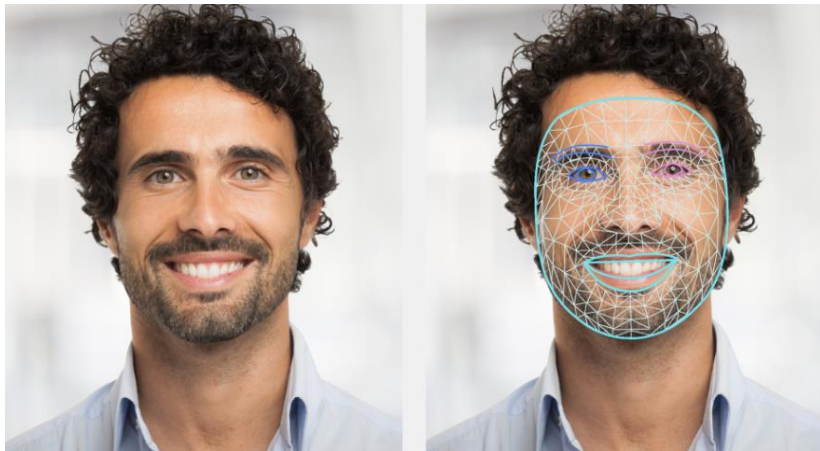
화장하기 적용



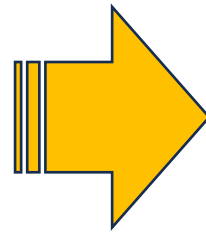
화장 적용 이후

프로젝트 기능

화장하기 기능 상세 (화장 적용할 영역 찾기)



 **MediaPipe**



Mediapipe의 머신러닝 (ML) 모델을 사용하여
사용자 얼굴의 대칭점을 찾아
영역 처리 (area Processing)을 실행하여
화장할 얼굴 영역을 찾기

프로젝트 기능

화장하기 기능 상세 (프레임을 처리하여 화장을 구현)

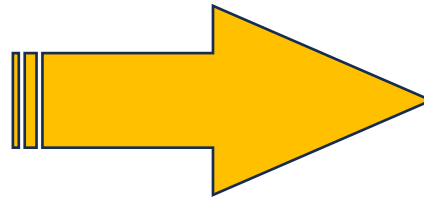
 MediaPipe



Scikit-learn의 컬러 매칭을 위한
이미지로 화장을 구현



컬러 매칭을 위한 레퍼런스 이미지
(사용자 얼굴에 맞게끔 블러를 적용하기 위해 필요한 이미지)



Scikit-learn을 활용한 컬러 매칭을 통해
사용자의 얼굴에 맞게 블러를 적용하고,

화장된 모습을
프레임 처리하여 구현

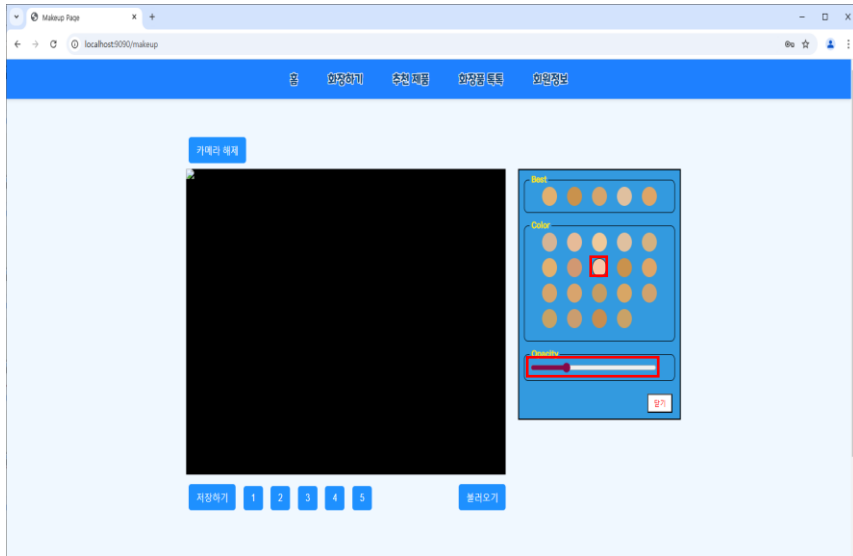
프로젝트 기능

화장하기 기능 상세 (서버 간 통신)



프로젝트 기능

화장 추천하기 기능

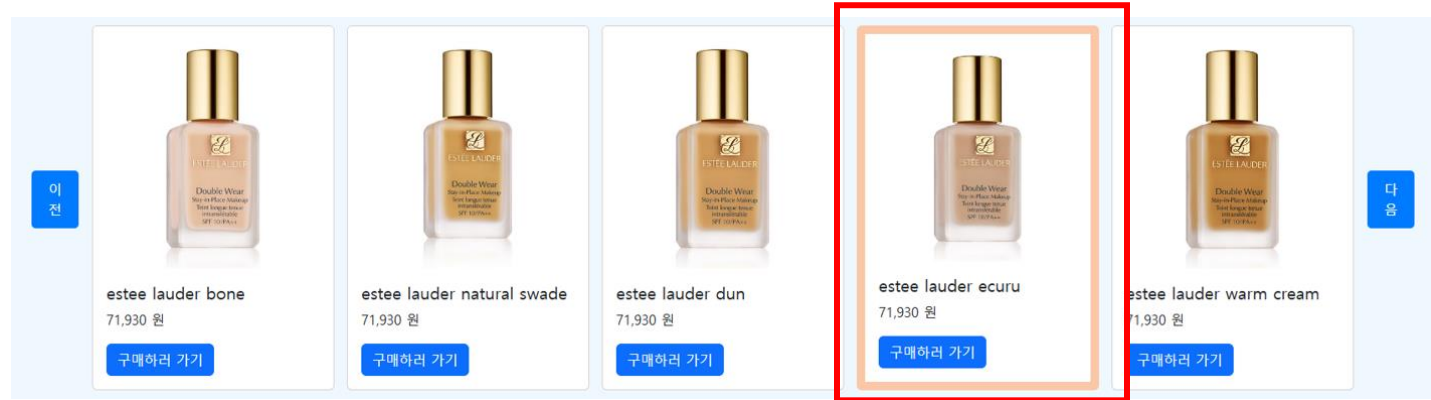


색상과 진하기를 선택

데이터베이스에 색상과 진하기 저장

makeup_id	member_id	category	button_number	color_code	opacity
1	qwer	foundation	1	#e2b06f	38
2	qwer	lipstick	1	#942a19	54
NULL	NULL	NULL	NULL	NULL	NULL

저장된 색상을 바탕으로 화장품 추천



프로젝트 기능

게시판 기능 (화장품 특특)

[홈](#) [화장하기](#) [추천 제품](#) [화장품 특특](#) [회원정보](#)

새 게시글 작성

제목

이미지

파일 선택

hs04321_1732280628882_...r_1731654850616_face.jpg

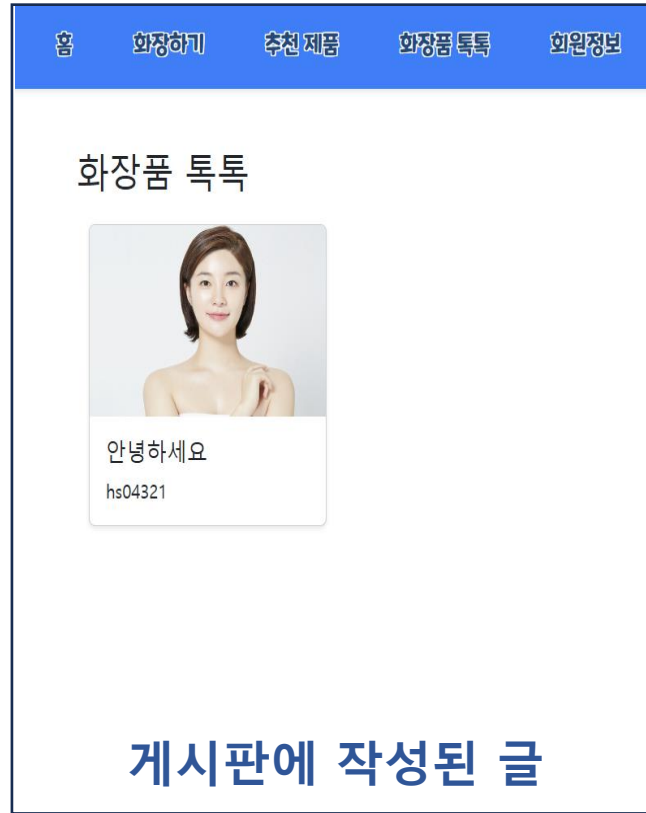
Ctrl 또는 Shift 키를 사용하여 여러 이미지를 선택할 수 있습니다.

본문

반갑습니다.


저장

게시판에 글 작성



[홈](#) [화장하기](#) [추천 제품](#) [화장품 특특](#) [로그인](#)

안녕하세요



반갑습니다.

수정 삭제

댓글

댓글을 입력하세요

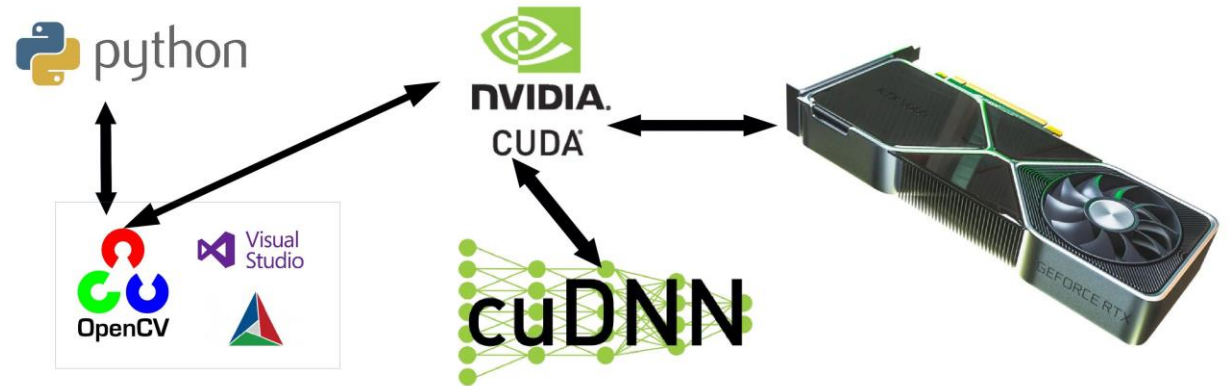
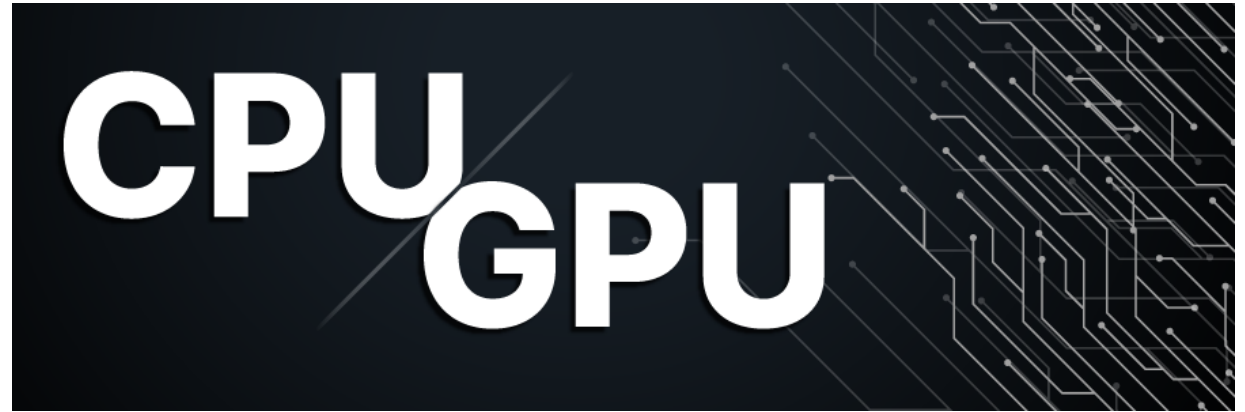
댓글 저장

게시판 글 자세히 보기

회고

기술적 개선 사항 : Cuda 사용

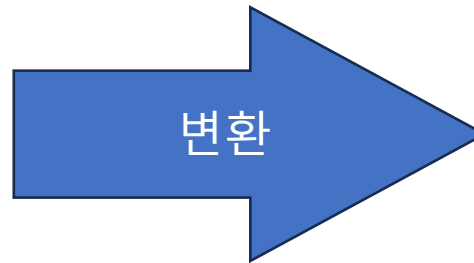
CPU로 동작하던 이미지 연산을
더 빠른 속도를 지원할 수 있는 GPU로 변경
(NVIDIA에서 Cuda를 지원)



회고

기술적 개선 사항 : Python으로 작성된 영상처리를 C++로 변경

Python으로 작성된 서버를 C++로 변경



CMake



회고

기술적 개선 사항 : Java Process의 Fast API 동작 문제



Java로 작성된 Spring Boot 서버에서
Java에서 Python Process를 동작시키는 방식으로
Python으로 작성된 Fast API 서버 동작이 가능했지만,
TensorFlow를 추가한 Fast API 서버는 해당 동작이 불가능

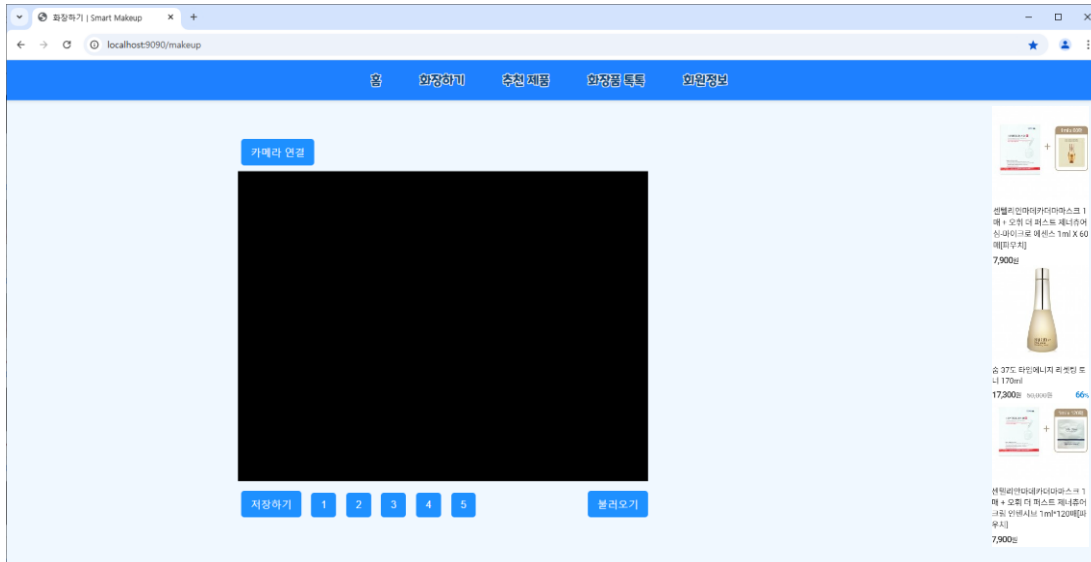


Fast API 서버를
따로 실행 가능하도록 변경

회고

비즈니스 개선 사항

비즈니스 개선 사항



웹 사이트에 광고를 추가

비즈니스 개선 사항



estee lauder bone

~~71,930 원~~ 69,990원

구매하러 가기

협찬을 통한 할인 기능 추가

마무리

End