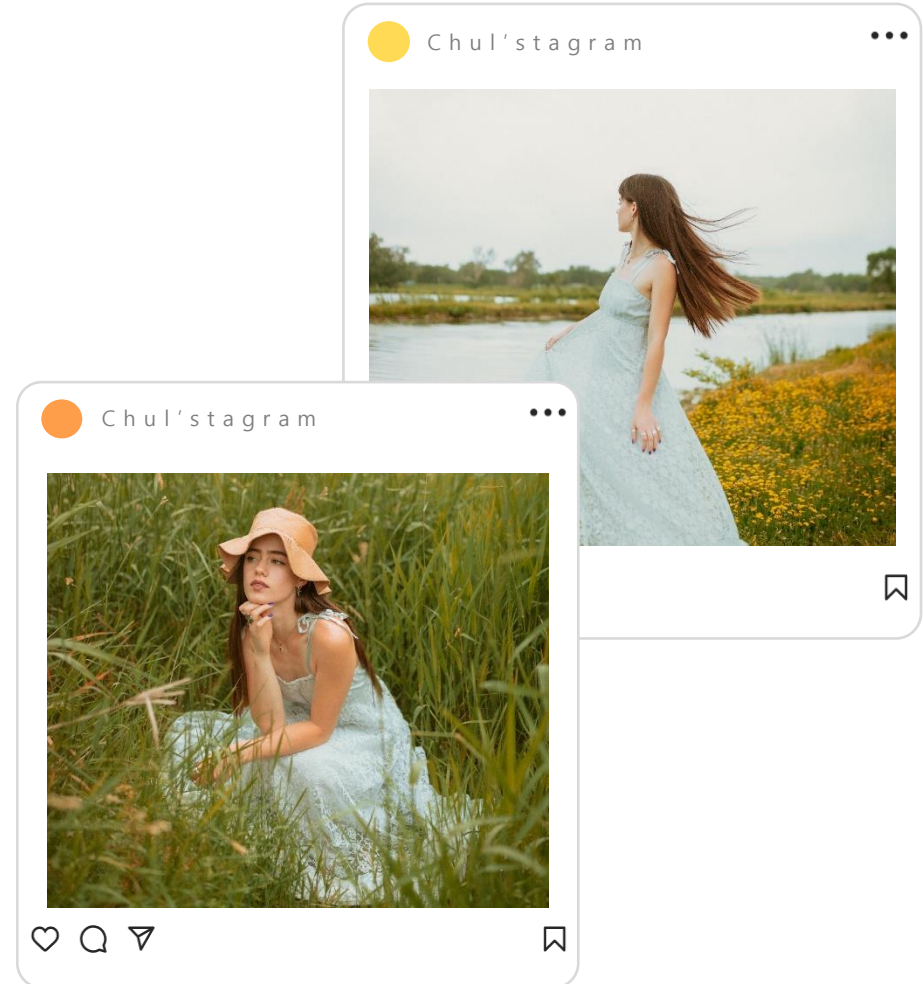




# SpringBoot 개인프로젝트

*Chul'stagram*

이철민





## 프로젝트 개요

- 개발일정
- 개발환경
- DB 구성



## 기술 상세

- 서비스 및 기능



## 시연

- 영상 시연

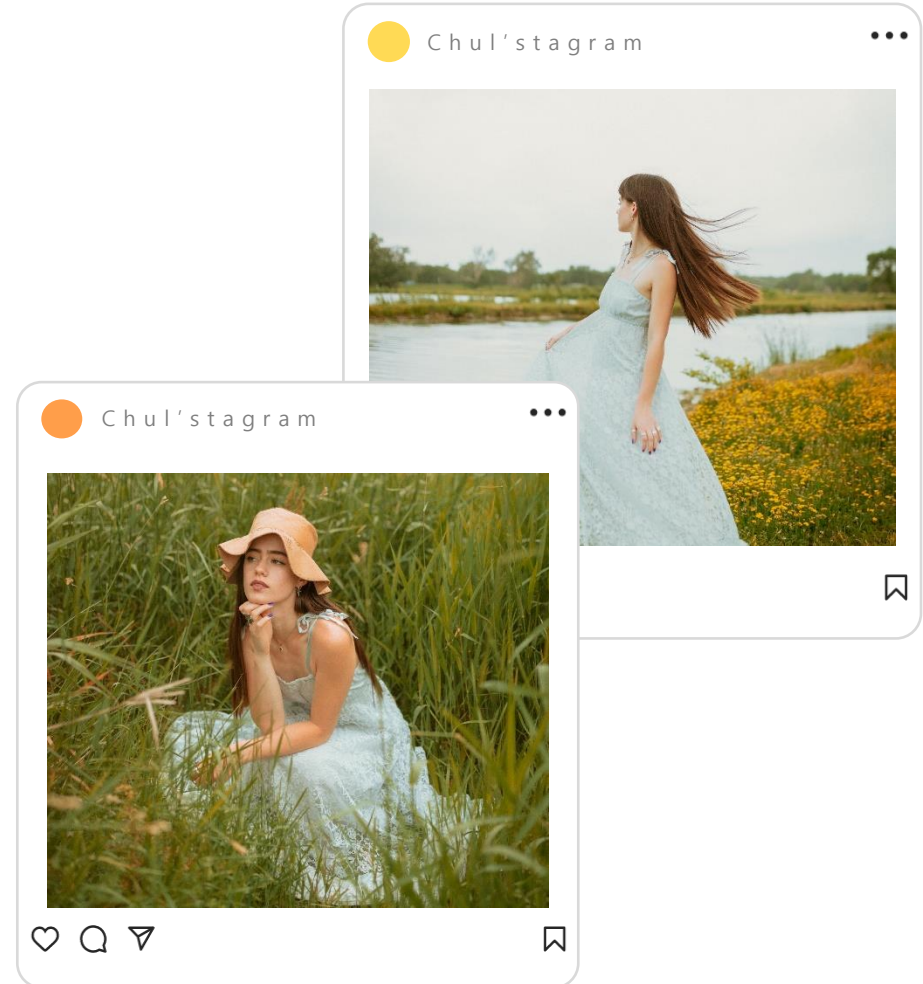


## 자체평가



# 프로젝트 개요

*Chul'stagram*





## 프로젝트 기획

2024.08.19

## 프로젝트 분석 및 화면 설계

2024.08.20 – 2024.08.23

## Programming

2024.08.24 – 2024.09.05

## 테스트 및 버그 수정

2024.09.05 – 2024.09.09



OS - windows 10



개발환경 - IntelliJ IDEA



JDK22



SpringBoot 3.2.6



DB - MySQL 8.1.0



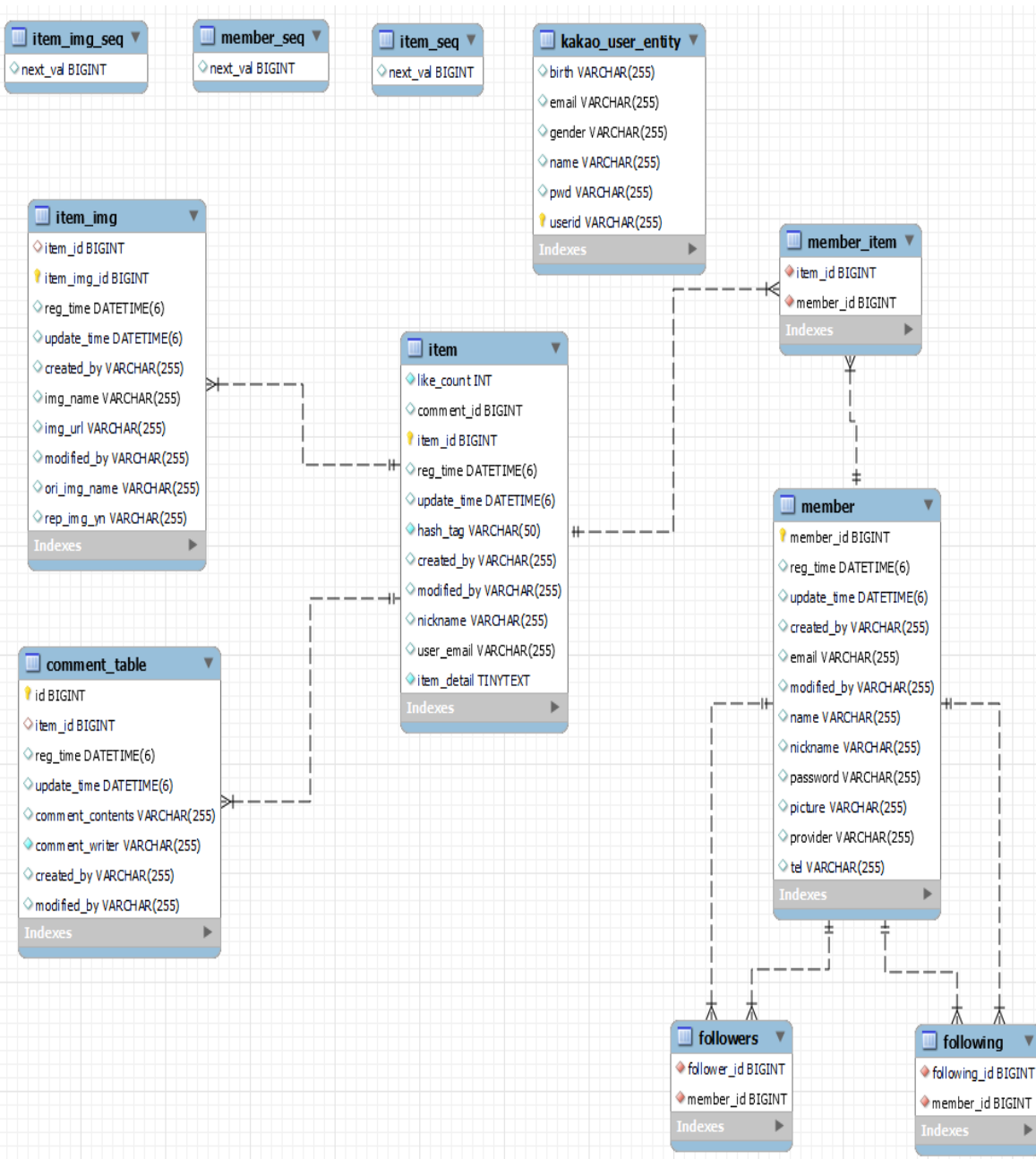
Front-end - HTML, CSS, JavaScript, Thymeleaf, Ajax, JQuery, Bootstrap



Back-end - Java, SpringBoot, Lombok, Hibernate







## DB 주요구성

### 1. Item

- Like\_count -> 좋아요 수
- Comment -> 댓글
- hashtag -> 해시태그
- Nickname -> 닉네임

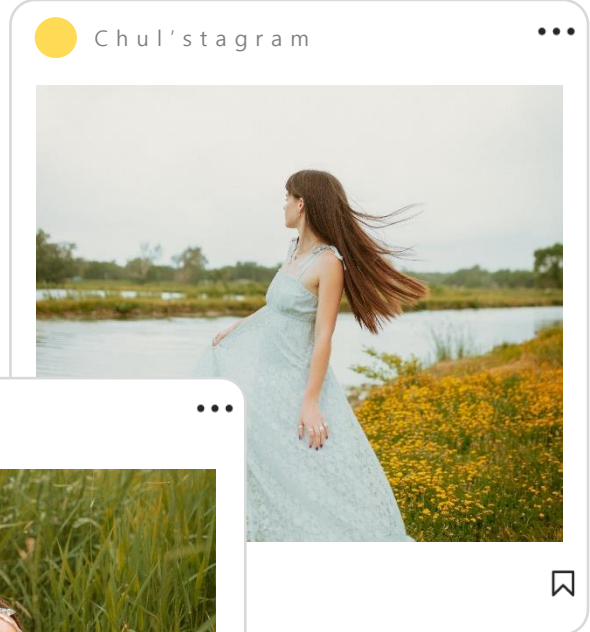
### 2. Member

- Followers -> 나를 팔로우 하는 사람
- Following -> 내가 팔로우 하는 사람
- Provider -> 소셜 로그인 (Google, Kakao, Naver)



# 기술상세

## *Chul'stagram*





로그인

*Chul'stagram*

이메일

비밀번호

로그인

가입하기

or

구글 로그인

카카오톡 로그인

네이버 로그인



## 로그인 화면

### 1. 로그인

- API를 이용해 편의성을 높인 회원가입 및 로그인 (Google, Kakao, Naver)

### 2. 회원가입

- 별도의 회원가입 가능

### 3. 로직 설명

- 로그인 버튼을 누르면 form action을 통해 Post형식으로 Controller로 입력한 데이터 전송
- 소셜 로그인 버튼을 누르면 설정 해놓은 a href를 통해 HttpSecurity의 OAuth2에서 provider로 소셜을 구분하여 로그인





## 회원가입



## 회원 가입

Chul'stagram

이름

이름을 입력해주세요.

이메일 주소

이메일을 입력해주세요.

인증 코드 요청

인증 코드

인증 코드 입력 후 인증 확인 버튼을 꼭 눌러주세요.

인증 코드 확인

비밀번호

비밀번호 입력해주세요.

닉네임

사용하실 닉네임을 입력해주세요.

전화번호

전화번호를 입력해주세요.

가입하기



## 회원가입 화면

## 1. API

- API를 이용한 메일인증 기능

## 2. 닉네임

- 활동 시 회원가입시 입력한 이름이 아닌 닉네임으로 출력

## 3. 로직 설명

- 입력한 값을 Controller에서 Dto의 양식을 활용해 Member 객체를 생성 후 데이터 전송
- MemberService에서 유효성 검사 후 DB 저장



## 게시글 등록



Chul'stagram

게시글 등록

게시글 관리

마이페이지

로그아웃

### 새 게시물

일반로그인 게시물

#일반로그인



공유



### 게시글

Follow



일반로그인 게시물

#일반로그인

좋아요 수: 0

## 게시글 등록 화면

### 1. 게시물 등록

- 각 입력 폼을 입력하지 않으면 상품이 등록되지 않음
- 이미지는 최소 1개 이상 등록해야 함

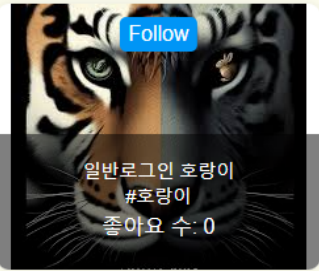
## 해시태그 검색

게시글



이전 1 다음

게시글  
호랑이 검색 결과



해시태그 검색

검색

## 해시태그 검색

### 1. 해시태그 검색

- 검색 input에 원하는 해시태그 검색 시 전체 게시글List에서 해당 게시글 List만 출력
- ItemSearchDto의 SearchQuery를 통해 입력한 HashTag의 게시글만 출력



## 게시글 관리



Chul'stagram

게시글 등록

게시글 관리

마이페이지

로그아웃

내 게시글



localhost 내용:  
정말로 삭제하시겠습니까?

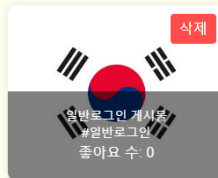
확인

취소

localhost 내용:  
삭제되었습니다.

확인

내 게시글



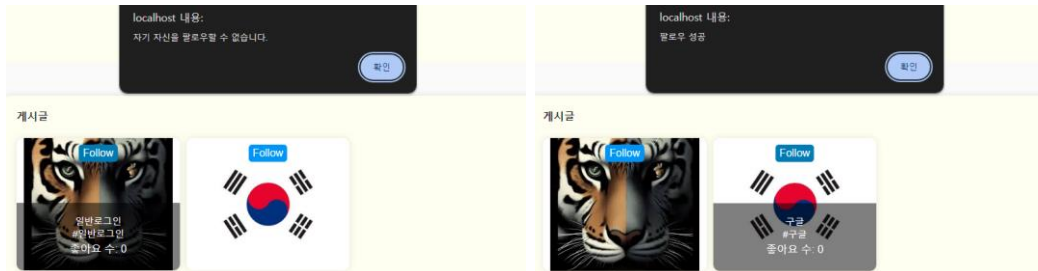
## 게시글 관리 화면

### 1. 게시글 삭제

- 각 게시글의 삭제버튼을 누르면 해당 게시글 삭제
- Ajax를 활용해 해당 ItemId 조회 후 삭제



## 팔로우 / 마이페이지



Chul'stagram    게시물 등록    게시물 관리    **마이페이지**    로그아웃

팔로워 : 1

일반로그인  
leechulmin1@naver.com

Follow

팔로잉 : 0

팔로워 : 0

팔로잉 : 1

최애의이경영  
leechulmin1@gmail.com

Unfollow



## 팔로우 / 마이페이지

### 1. 팔로우

- 각 게시글의 팔로우버튼을 누르면 팔로우 가능
- 본인 게시글의 팔로우버튼을 누르면 팔로우 불가능
- Ajax를 활용해 해당 Item을 공유한 member의 nickname과 email을 조회 후 마이페이지에 출력



연팔로우 / 마이페이지



localhost 내용:

연팔로우 성공

확인

팔로워

팔로잉 : 1

최애의이경영

leechulmin1@gmail.com

Unfollow

일반로그인 마이페이지

팔로워 : 0

팔로잉 : 0

구글로그인 마이페이지

팔로워 : 0

팔로잉 : 0



## 연팔로우 / 마이페이지

### 1. 팔로우

- 마이페이지의 연팔로우버튼을 누르면 연팔로우 가능
- Ajax를 활용해 해당 Item을 공유한 member의 nickname과 email을 조회 후 List에서 삭제 후 마이페이지에 출력



 좋아요

...

최애의이경영

#구글  
구글

Follow



좋아요 수: 0

최애의이경영

#구글  
구글

Follow



좋아요 수: 1



## 좋아요

### 1. 좋아요

- Ajax를 활용해 해당 Item의 id를 조회하고 현재 로그인한 Member의 email과 Item의 ItemId를 서로 연결시킨 후 LikeCount를 +1
- 좋아요 취소 -> 좋아요를 누른 사람의 email이 중복되면 좋아요 취소 후 LikeCount -1



댓글



댓글

최애의이경영

Korea

작성

작성자

내용

작성시간

삭제

댓글

최애의이경영

내용

작성

작성자

내용

작성시간

최애의이경영

Korea

2024-09-10 12:01:27

삭제



## 댓글

### 1. 댓글

- 각 게시글의 아래 댓글내용에 원하는 내용을 적은 후 작성 버튼 클릭 -> 댓글 등록
- 작성자의 email이 아닌 nickname, 내용, 작성시간 출력
- Ajax를 활용해 해당 작성자의 nickname과 내용을 가져온 데이터를 itemId에 전송



## 댓글



## 댓글

최애의이경영

내용

작성

작성자

내용

작성시간

최애의이경영

Korea

2024-09-10 12:01:27

삭제

localhost 내용:

댓글을 삭제하시겠습니까?

확인

취소

localhost 내용:

삭제했습니다!!

확인

## 댓글

최애의이경영

내용

작성

작성자

내용

작성시간

삭제

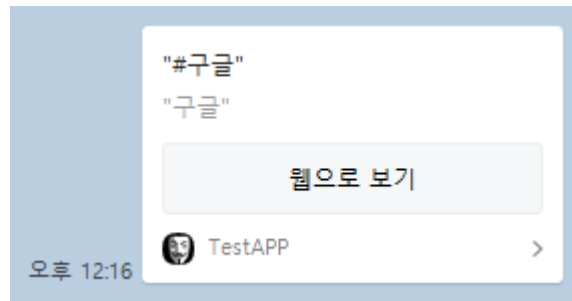
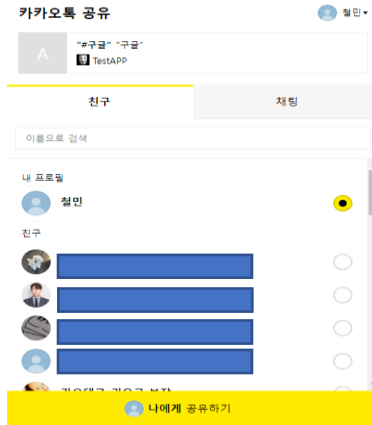
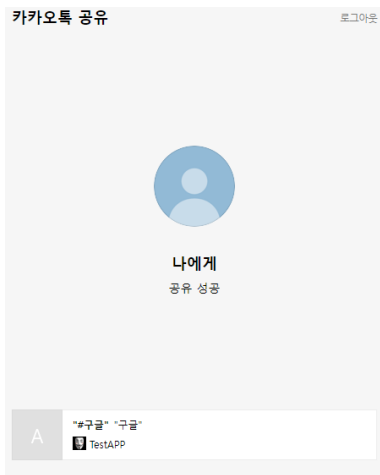
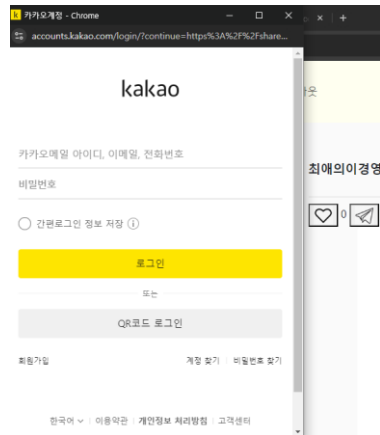


## 댓글 삭제

## 1. 댓글 삭제

- 각 게시글의 아래 댓글내용에 삭제버튼 클릭 -> 댓글 삭제
- Ajax를 활용해 해당 댓글 작성자인지 CommentWriter와 비교 확인 후 삭제

## 공유하기



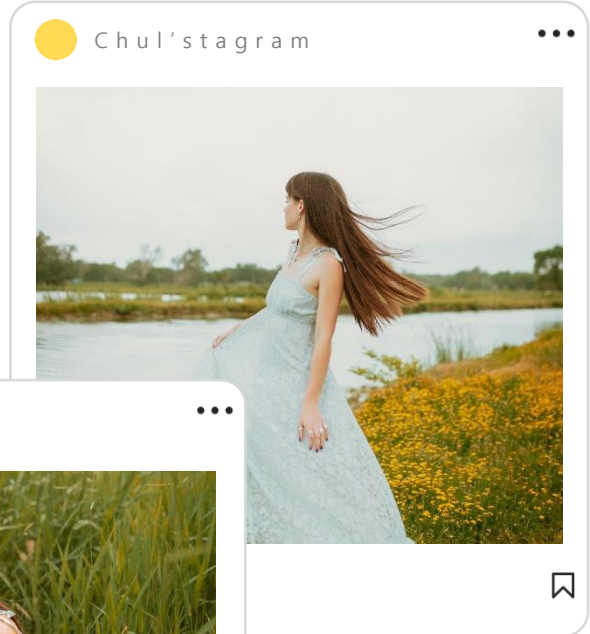
## 공유하기

- 공유하기
  - Kakao API를 활용해 게시물 공유가능



# 자체평가

*Chul'stagram*





자체평가



## 자체평가

Q. 사전 기획의 관점에서 **프로젝트 결과물에 대한 완성도 평가**(10점 만점)

A. 6점

Q. **잘한 부분과 아쉬운 점**

A.

잘한 점 – 처음 구상했던 기능들은 구현했다.

아쉬운 점 – 기능은 구현했지만 디테일하게 완성도를 높이진 못했다.  
계속 보완해야 할 것 같다.

Q. 프로젝트 결과물의 **추후 개선점이나 보완할 점** 등 내용 정리

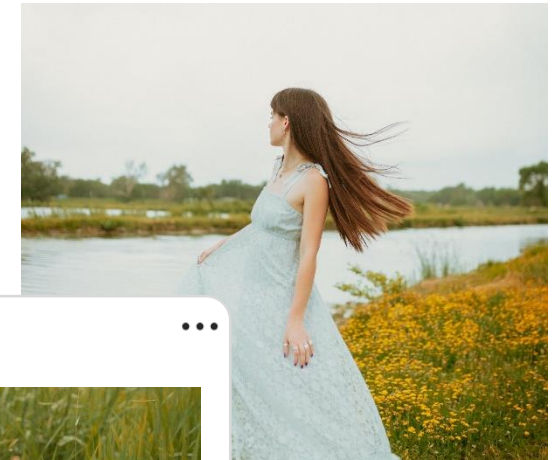
A. 각 기능과 css를 디테일하게 개선, 보완이 필요하다. 그래도 팀 프로젝트와 이번 개인 프로젝트를 준비하고 개발하는 과정에서 스스로 많이 성장했음을 느꼈다.





# Thank you

Chul'stagram



Chul'stagram

