**장고개발폴더명: DCustoMountain**

**프로젝트명: DCM**

**앱명:**

**1. 등산 정보 탭 - mountains**

**2. 커뮤니티 탭 - community**

**3. 마이페이지 – users**

**<등산 정보 탭>**

- 앱이름: mountains

**기능**

1. **유저 등산 레벨 및 취향에 따른 검색**

* 난이도(초, 중, 고) , 지역, 높이, 평균 산행시간, 산이름으로 검색 가능
* 모든 항목 미선택시 전체 조회

1. **100대 명산 정보 제공**

* 산림청 홈페이지 크롤링
* 사진, 난이도, 지역, 높이, 관련 설명 등 정보 제공

**<마이페이지 탭>**

- 앱이름: users

**기능**

1. 인증

* 사용자 인증 기능

Form: (ModelForm 사용)

|  |  |  |  |
| --- | --- | --- | --- |
| 이름 | 모델 | 필드 | 위젯 |
| LoginForm | User | username |  |
|  |  | password |  |
| SignupForm | User | Username |  |
|  |  | Password1 |  |
|  |  | Password2 |  |
|  |  | Profile\_image |  |
|  |  | Short\_description |  |

Model:

|  |  |  |
| --- | --- | --- |
| 이름 | 필드 | 내용 및 특징 |
| User | Profile\_image |  |
| Short\_description |  |
| Like\_posts |  |
| Comment\_posts |  |
| Bookmark\_posts |  |
| Following |  |
| Block\_flag |  |
| Ability |  |

View:

* Login\_view :
  + 로그인 되어 있으면 feed 페이지로 redirect
  + 로그인 안되어 있으면 login.html 로 render
* Logout\_view :
  + Login.html 로 render
* Signup :
  + 회원가입 완료하면 feed 페이지로 redirect
  + 회원가입 하려면 signup.html 로 render
  + Make\_style 기능 SignupForm 에 같이 표시

Template:

* Login.html
* Signup.html

url:

* users/login/
* users/logout/
* Users/signup/

1. 팔로우

* 사용자들 간의 팔로우 기능

Models:

|  |  |  |
| --- | --- | --- |
| 이름 | 필드 | 내용 및 특징 |
| Relationship | From\_user |  |
| To\_user |  |
| Created |  |

View:

* Follow :
  + 팔로우 되어 있으면 언팔로우
  + 팔로우 안되어 있으면 팔로우
  + url\_next 로 redirect
* followers :
  + followers.html 로 render -> 팔로워 조회
* following :
  + following.html 로 render -> 팔로잉 조회

Template:

* followers.html
* following.html

Url:

* users/<int:user\_id>/follow/
* users/<int:user\_id>/followers/
* users/<int:user\_id>/following/

1. 프로필

* 개인 사용자 프로필 화면

Model:

* users/User
* community/Post
* community/Comment

View:

* mypage :
  + mypage.html 로 render -> 사용자 메인 마이 페이지 화면 조회
* profile :
  + profile.html 로 render -> 사용자 개인 프로필 조회
* edit\_profile :
  + SignupForm 불러와서 signup.html 로 render
* Post\_like\_detail :
  + Post\_like.html 로 render -> 좋아요한 게시물 목록 조회
* Post\_comment\_detail :
  + Post\_comment.html 로 render -> 댓글단 게시물 목록 조회
* Post\_bookmark\_detail :
  + Post\_bookmark.html 로 render -> 저장한 게시물 목록 조회

Template:

* Mypage.html
* Profile.html
* Post\_like\_detail.html
* Post\_comment\_detail.html
* Post\_bookmark\_detail.html

Url:

* Users/mypage/
* Users/profile/
* Users/edit\_profile/
* Users/post\_like\_detail/
* Users/post\_comment\_detail/
* Users/post\_bookmark\_detail/

1. 사용자 난이도 설정

* 사용자가 자신에게 맞는 난이도를 설정할 수 있도록 라디오 버튼 생성

Model:

|  |  |  |
| --- | --- | --- |
| 이름 | 필드 | 내용 및 특징 |
| MyMountain | User |  |
| Mountain |  |
| Date |  |
| Duration |  |
| Distance |  |
| Complete |  |
| Wish |  |

View:

* Make\_style :
  + 사용자가 자신의 난이도 측정을 위한 라디오 버튼 클릭
  + 작성 완료하면 Style\_detail 로 redirect
  + 작성 하려면 make\_style.html 로 render
* Edit\_style :
  + 난이도 측정 라디오 버튼 수정
  + 수정 완료하면 style\_detail 로 redirect
  + 수정하려면 데이터 담아서 Make\_style.html 로 render
* Style\_detail :
  + 난이도 상세 페이지
  + Style\_detail.html 로 render
* Complete :
  + 완등한 산 complete=True 로 save
  + Complete 누른 페이지로 redirect
* Incomplete :
  + 완등하지 않은 산 complete=False 로 save
  + Incomplete 누른 페이지로 redirect
* Complete\_detail :
  + 완등한 산 조회
  + Complete\_detail.html 로 render
* Wish :
  + 위시리스트 wish=True 로 save
  + Wish 누른 페이지로 redirect
* Unwish :
  + 위시리스트 wish=False 로 save
  + Unwish 누른 페이지로 redirect
* Wish\_detail :
  + 위시리스트 산 조회
  + Wish\_detail.html 로 render
* Mtn\_Search :
  + Complete, wish 버튼이 달린 산 목록 검색해서 조회

Template:

* Make\_style.html
* Style\_detail.html
* Complete\_detail.html
* Wish\_detail.html
* Mtn\_search.html

url:

* Users/make\_style/
* Users/edit\_style/
* Users/style\_detail/
* Users/complete/
* Users/incomplete/
* Users/complete\_detail/
* Users/wish/
* Users/unwish/
* Users/wish\_detail/
* Users/mtn\_search/

1. 캘린더에 기록

* 사용자가 등반한 산 기록을 캘린더에 등록하여 한눈에 확인하게 쉽도록 하는 기능

Form:

* Calender ModelForm 으로 폼 생성

Model:

|  |  |  |
| --- | --- | --- |
| 이름 | 필드 | 내용 및 특징 |
| Calender | User |  |
| Mountain |  |
| Mymountain |  |

View:

* Add\_calender :
  + 캘린더에 등록할 폼 작성
  + 작성 완료하면 mypage 로 redirect
  + 작성 하려면 add\_calender.html 로 render
* Calender\_detail :
  + 캘린더 세부 조회
  + Calender\_detail.html 로 render
* Edit\_calender :
  + 수정 완료하면 calender\_detail 로 redirect
  + 수정하려면 데이터 담아서 add\_calender.html 로 render
* Delete\_calender :
  + Mypage 로 redirect

Template:

* Add\_calender.html
* Calender\_detail.html

url:

* Users/add\_calender/
* Users/calender\_detail/
* Users/edit\_calender/
* Users/delete\_calender/

**<커뮤니티 탭>**

- 앱이름: community

- 짭스타그램 플젝 인용하지만, 몇몇 기능 추가 및 수정으로 우리 웹사이트에 맞춰 구현

- 아래 계획의 기능 설명은 짭스타그램과 다른 부분 위주로 작성

**기능**

\* 커뮤니티 탭 이용 권한은 회원가입 및 로그인한 유저에게만

1. 게시글

* 기본 포맷은 짭스타그램과 유사
* 피드 페이지에서 게시글은 2줄만 보이게 css 적용 댓글은 3개까지만! 전체 내용은 세부페이지에서 확인

Form: (ModelForm 사용)

|  |  |  |  |
| --- | --- | --- | --- |
| 이름 | 모델 | 필드 | 위젯 |
| PostForm | Post | content | placeholder |

Model:

|  |  |  |
| --- | --- | --- |
| 이름 | 필드 | 내용 및 특징 |
| 1. Post | user | users.User 를 외래키로 |
| content | 게시글 내용 |
| created | 날짜, 시간 포함된 작성일시  (auto\_now\_add) |
| tags | community.HashTag와 ManyToManyField 설정 |
| 1. PostImage | post | users.User 를 외래키로 |
| photo | 게시할 사진, upload\_to 속성 |

View:

* Feeds: post 객체 리스트와 comment\_form 렌더링
* Add\_post: post 요청 온 정보를 postform 에 넣고 이미지와 해시태그 객체 만들어 해당 게시글이 보이도록 redirect/ get 요청이면 빈 폼
* post\_detail: url 로 전달받은 post\_id 로 post 객체 찾아서 빈 comment\_form 과 함께 렌더링

Template:

* feeds.html
* add\_post.html
* post\_detail.html

url:

* /community/feeds/ (“feeds”),
* /community/add\_post/ (“add\_post”),
* /community/<int:post\_id>/ (“post\_detail”)

--------------------------------------------------------------------------------------------------------------

1. 댓글

* 게시글 목록 페이지에선 댓글 3개만 볼 수 있도록
* 세부 목록 페이지에선 모든 댓글 볼 수 있도록

Form: (ModelForm 사용)

|  |  |  |  |
| --- | --- | --- | --- |
| 이름 | 모델 | 필드 | 위젯 |
| CommentForm | Post | post |  |
| content | placeholder |

Model:

|  |  |  |
| --- | --- | --- |
| 이름 | 필드 | 내용 및 특징 |
| Comment | user | users.User 를 외래키로 |
| post | Post 를 외래키로 |
| content | 댓글 내용 |
| created | 날짜, 시간 포함된 작성일시 (auto\_now\_add) |

View:

* add\_comment: post 요청이 온 정보를 commentform 에 넣어 유효성 검사 후 해당 게시글이 보이도록 redirect
* delete\_comment: url 로 comment\_id 받아서 본인이 쓴 댓글에 대해 삭제

Template:

* post\_detail.html
* feeds.html

url:

* /community/add\_comment/ (“add\_comment”)
* /community/delete\_comment/ (“delete\_comment”)

--------------------------------------------------------------------------------------------------------------

1. 좋아요

* 본인 게시글 이외의 다른 게시물들에 좋아요 가능

View:

* Like\_post: url로 post\_id 를 전달 받아, 타겟 포스트에 좋아요 여부에 따라 토글방식으로 구현

Template:

* feeds.html
* post\_detail.html 사용

url:

* /community/<int:post\_id>/like/ (“like\_post”)

--------------------------------------------------------------------------------------------------------------

1. 팔로우

* 짭스타 플젝과 유사하게 팔로우 관계 형성 (users 앱에서 모델 생성)
* 팔로우할 수 있도록 템플릿에 버튼만 구현 (조건문으로 팔로우 여부에 따라 버튼 생성)

Template:

* feeds.html
* post\_detail.html 사용

--------------------------------------------------------------------------------------------------------------

1. 인기글 (시간대별 Top 10)

* 1시간 단위로 좋아요 개수에 따라 순위를 정해 인기글 top 10 제시

View:

* Trending: 당일의 인기 게시글 목록. 매 정각 새로고침. 좋아요 개수로 내림차순하여 10개 골라 보여줌

Template:

* trending.html

url:

* /community/trending/ (“trending”)

--------------------------------------------------------------------------------------------------------------

1. 게시글 저장

* 본인이 업로드하지 않은 게시글은 저장 버튼을 만들어. 마이 페이지 (users) 에서 저장한 목록이 보이도록 함.

View:

* Save\_post: 저장하고 싶은 게시글의 세부 목록 페이지에서 저장 버튼을 만들어 저장

Template:

* Post\_detail.html

url:

* /community/<int:post\_id>/save\_post/ (“save\_post”)

--------------------------------------------------------------------------------------------------------------

1. 해시태그

* 우리 웹사이트에서 제공하는 100대 명산의 해시태그 미리 생성

Model:

|  |  |  |
| --- | --- | --- |
| 이름 | 속성 | 내용 및 특징 |
| HashTag | name | 태그명 |

View:

* tags: url 로 전달받은 tag\_name 을 가지는 게시글 리스트 렌더링

Template:

* tags.html

url:

* /community/tags/<str:tag\_name>/ (tags)

--------------------------------------------------------------------------------------------------------------

1. 검색기능 (게시글, 계정명, 해시태그)

* 검색바를 생성하여 게시글, 계정명, 해시태그 검색 가능하도록 함
* 템플릿에서 검색바 구현

View:

* search: url 로 들어온 문자열로 게시글, @계정명, #해시태그 구분 (조건문 사용)
* recommendation: 단위 시간당 게시글, 계정명, 해시태그 검색 횟수를 순위 매겨 5개 제안

Template:

* feeds.html
* search\_post.html

url:

* /community/search/ (“search”)
* /community/recommendation/ (“recommendation”)

--------------------------------------------------------------------------------------------------------------

1. 신고 (부적절 게시글)

* 신고 버튼을 만들어 특정 횟수 이상 신고 보고가 되면 계정 정지
* 게시글, 댓글에 모두 적용

View:

* report\_post: url로 post\_id 를 받으며, 템플릿에서 신고 횟수를 세고 이 값이 특정 횟수 이상이면 계정 중지.

Template:

* feeds.html
* post\_detail.html

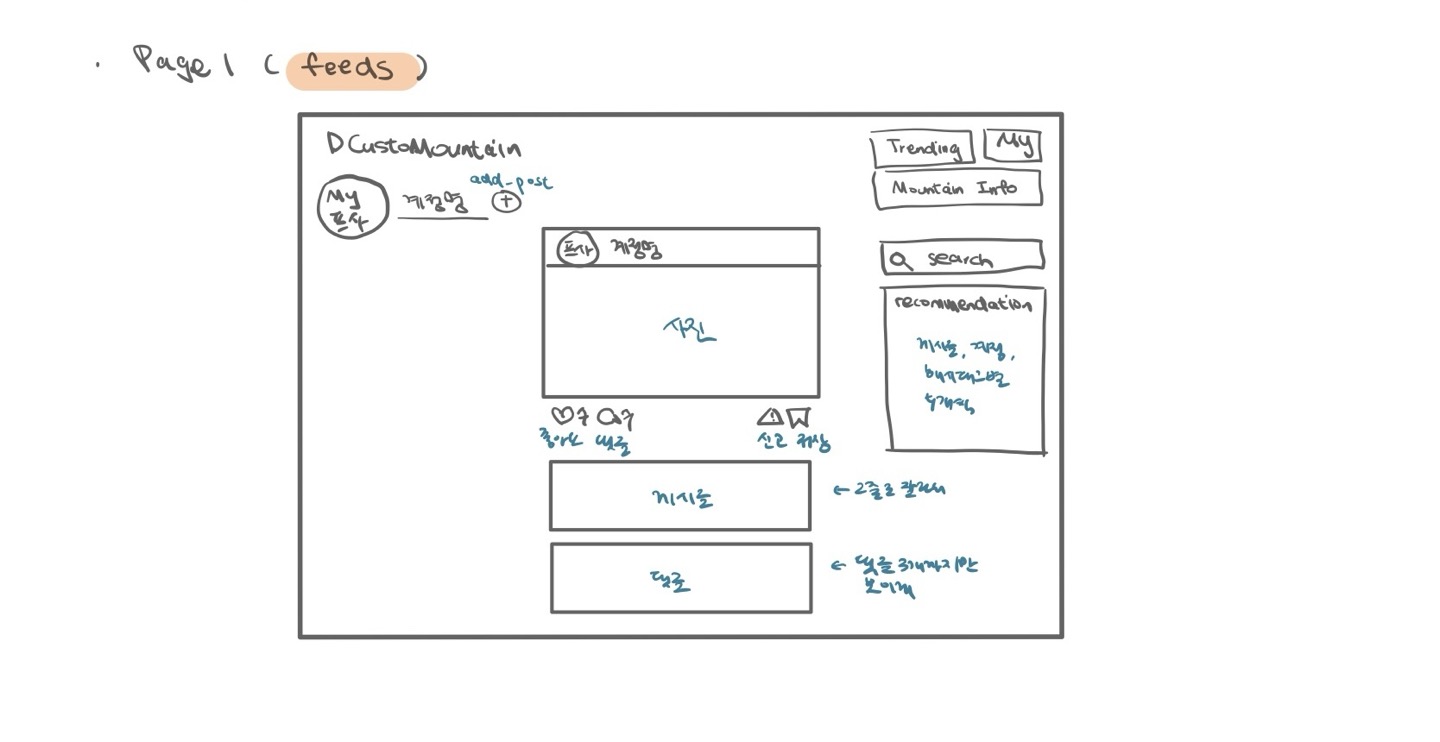
url:

* /community/<int:post\_id>/report\_post/ (“report\_post”)

**웹페이지 구성**

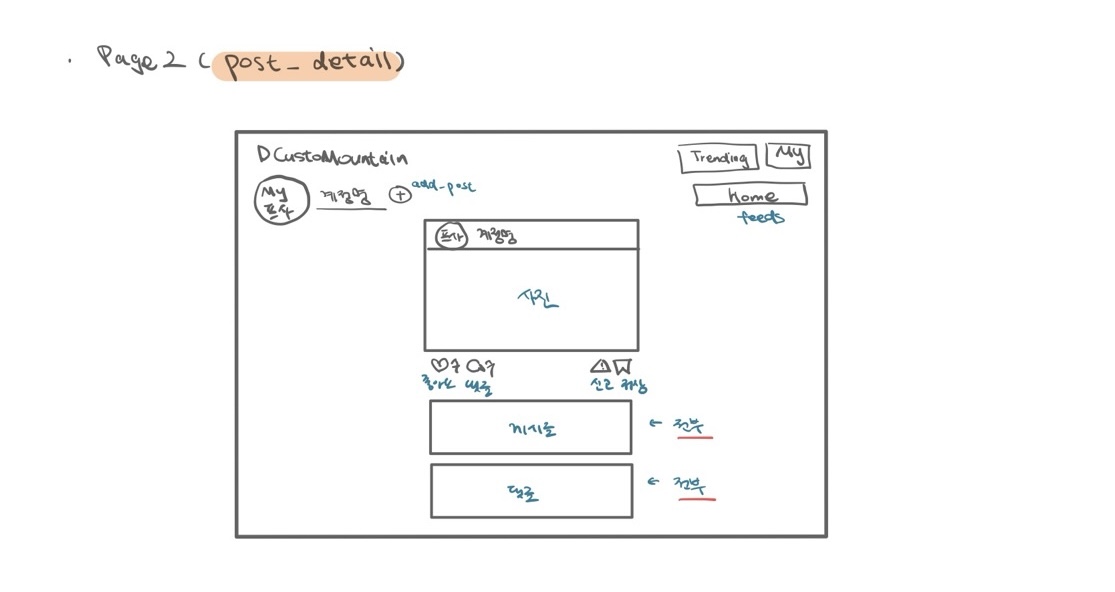
**Page1. feeds:**

게시글의 전체 목록을 둘러 볼 수 있는 페이지로, 게시글 일부 내용 및 사진, 댓글 일부, 좋아요, 저장, 신고, 상단에는 검색바를 위치



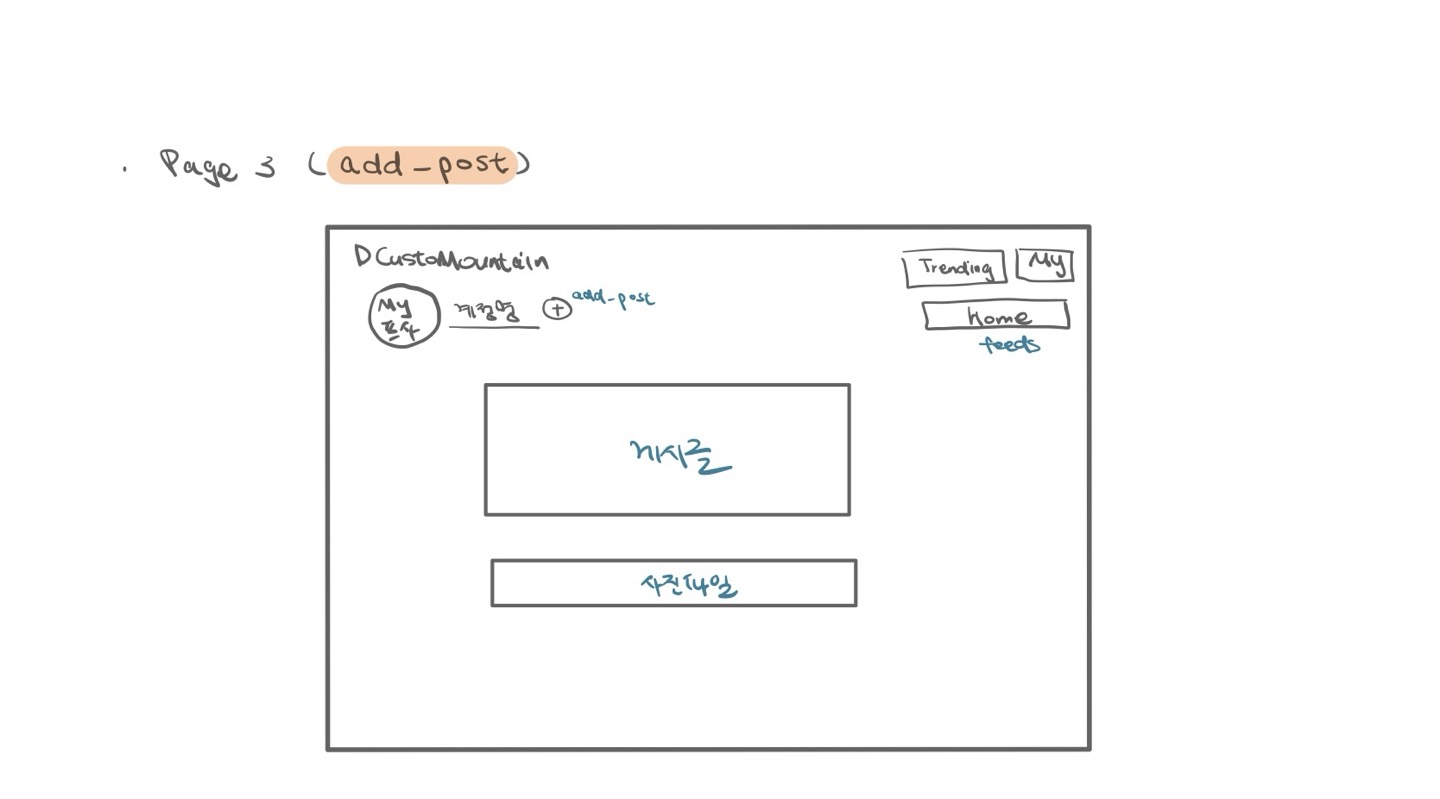
**Page2. post\_detail:**

page1 에서 게시글의 사진을 클릭하면 게시글의 전체 내용과 댓글 전체를 확인할 수 있도록 안내



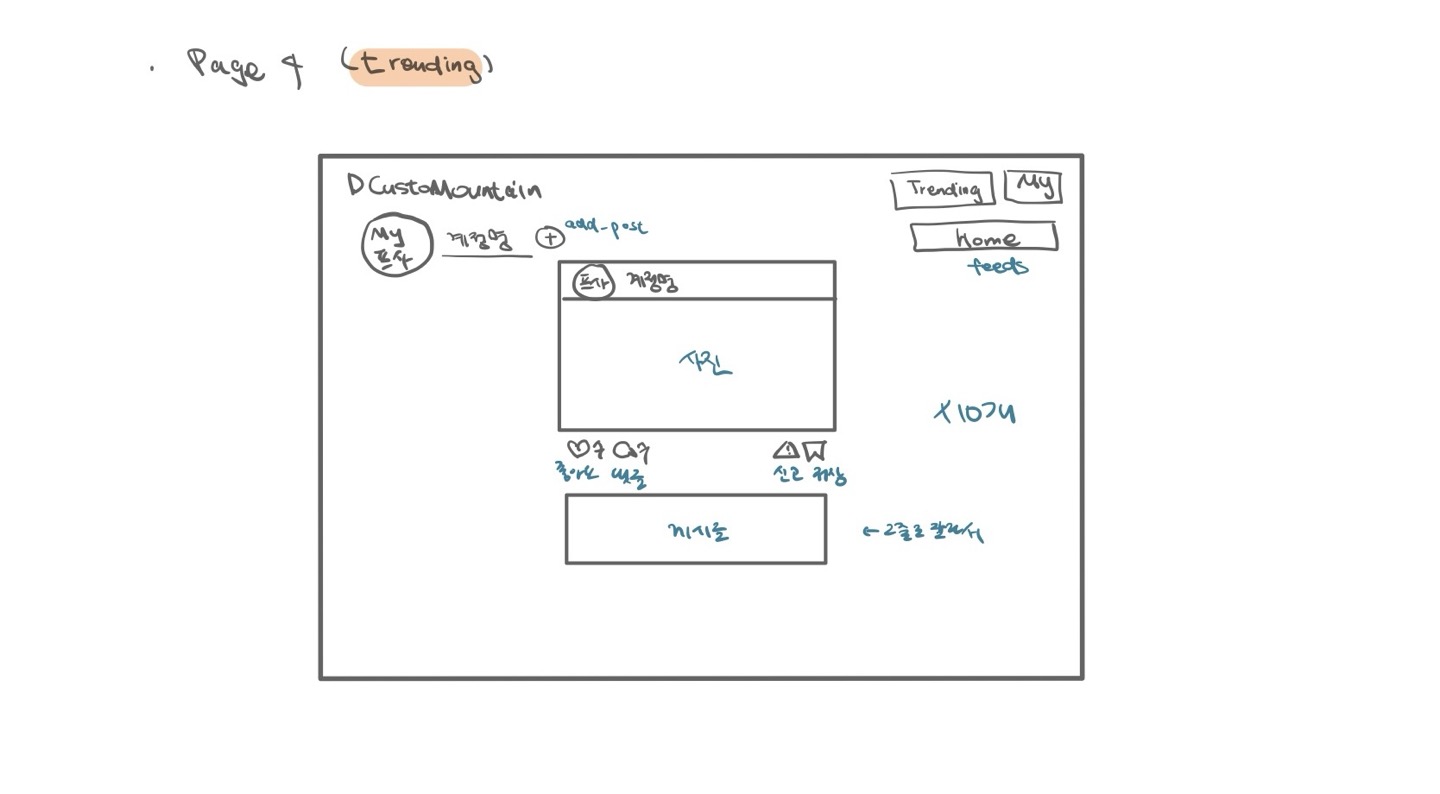
**page3. add\_post:**

새 게시글과 이미지를 업로드할 수 있도록 폼 제공



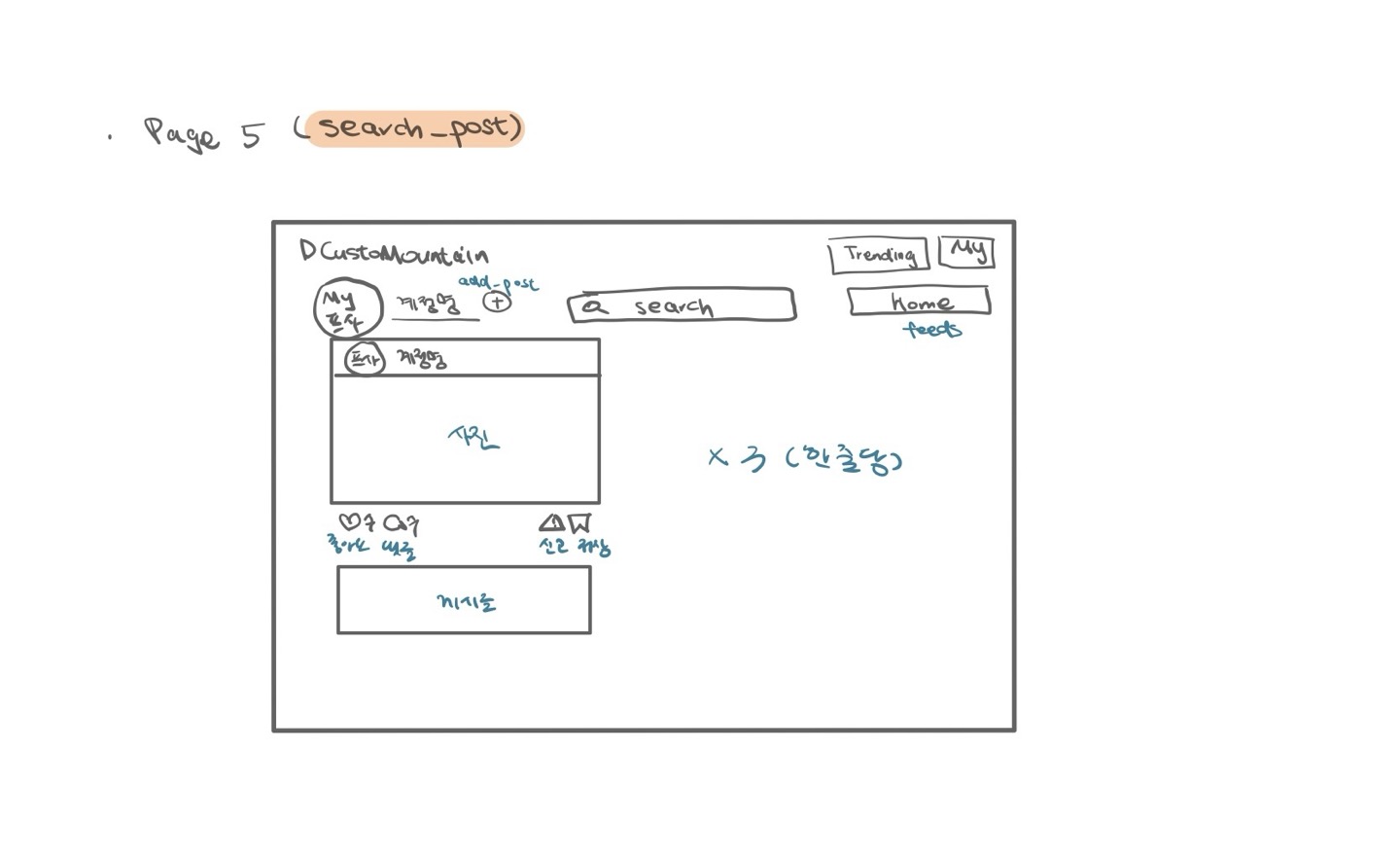
**Page4. trending:**

시간 단위별 게시글의 좋아요 개수에 따라 순위를 정해 TOP 10 게시글 리스트를 제공



**Page5. search\_result:**

page1 상단에 위치한 검색바를 통해 검색을 하면 결과를 보여줌.

****