



Spring Web Project

CONTENTS



Part 1 프로젝트 개요

1. 기획 의도
2. 개발 환경

Part 2 프로젝트 설계

1. 화면설계
2. ERD
3. 기능정의서

Part 3 프로젝트 구현

Part 1

프로젝트 개요



기획 의도

**SHARE
YOUR
OWN
TRIP**

당신이 꿈꿔왔던,
당신을 꿈꾸게 했던,
장소를 공유하세요.
그곳이 우리의 다음 여행지입니다.

**MAKE
YOUR
TRAVEL
MATES**

아무리 오랜 친구라도
여행스타일까지 같을 수는 없겠죠.
나와 꼭 맞는 여행스타일의
새로운 친구들을 만나보세요.
당신의 여행이 더욱 즐거워질 것입니다.

**EVERYTHING
IS A
JOURNEY**

거창한 여행코스가 아니어도 좋습니다.
일상이 여행처럼 소중하다는 것,
모두가 알고 있으니깐요.
소소한 일상을 자유롭게 공유해주세요.

개발 환경



데이터베이스 및 서버



개발 도구 및 언어



형상관리 및 협업 툴

Part 2

프로젝트 설계

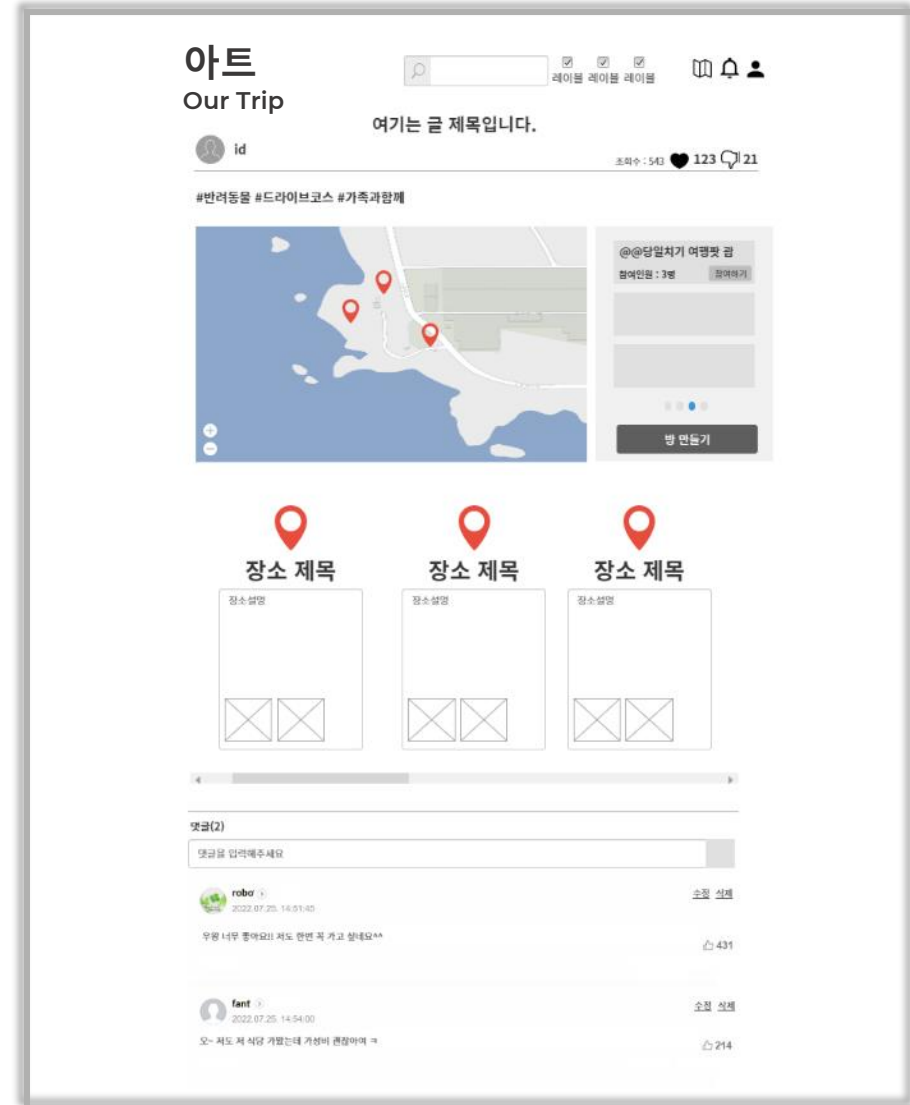


Part 2

화면설계 kakao oven



메인 화면

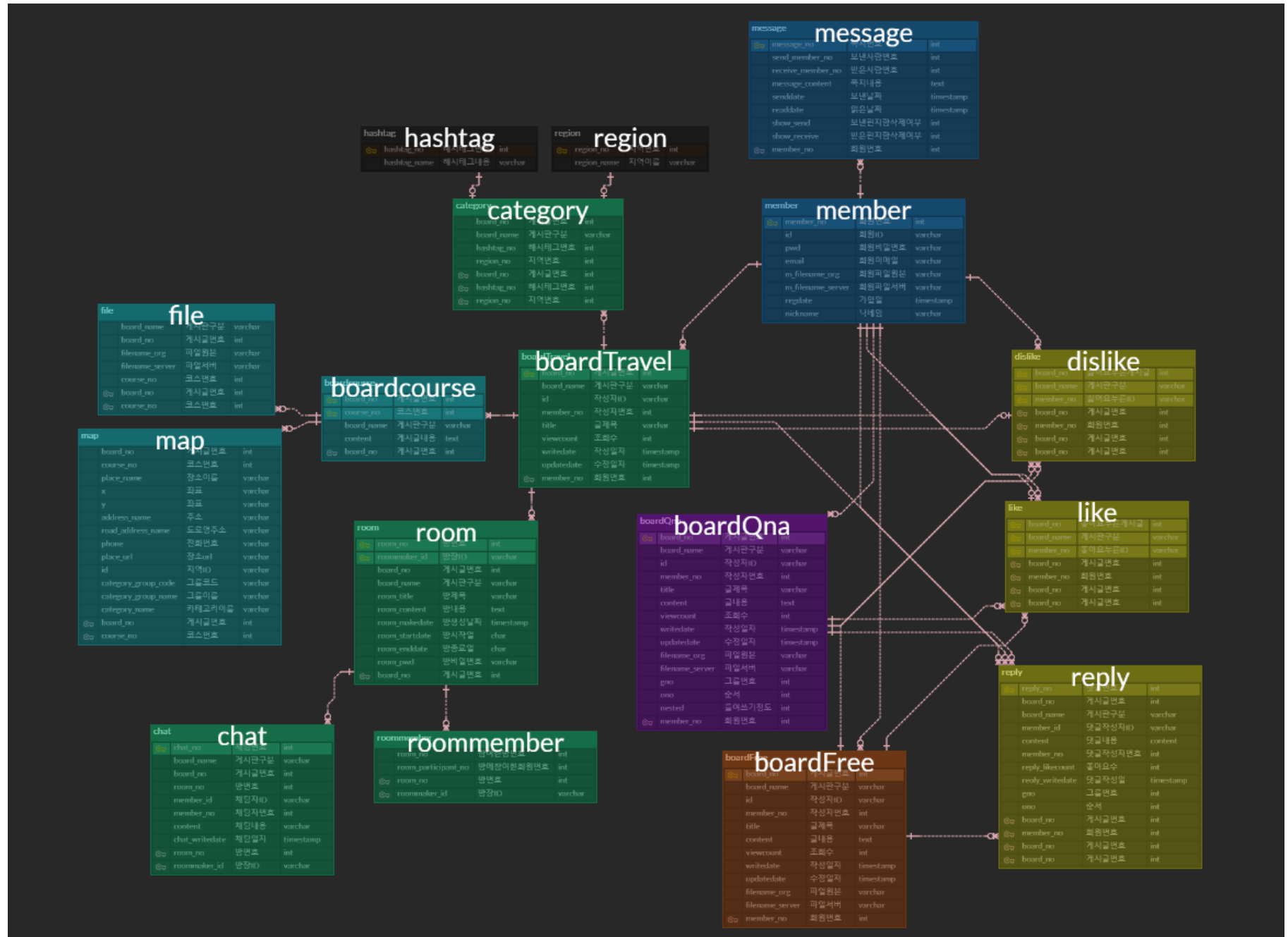


글 상세 화면

Part 2

ERD

ERDCloud



기능정의서

화면	기능	기능 정의
통합검색	엘라스틱서치	<ul style="list-style-type: none"> • nori_tokenizer 사용하여 인덱싱 • mysql에서 elasticsearch로 데이터 이관 • multi match query 사용하여 여러 필드에서 검색 • 게시판 구분 없이 지역/장소/제목/작성자/내용으로 키워드 검색 가능
상단 메뉴바	안읽은 쪽지 확인	<ul style="list-style-type: none"> • 쪽지 아이콘 마우스오버시 안읽은 쪽지 개수 확인 • 쪽지 아이콘 클릭시 안읽은 쪽지 리스트를 모달로 출력 • 안읽은 쪽지가 있을 때/없을 때 아이콘 다르게 표시
	실시간 쪽지 알람	<ul style="list-style-type: none"> • ajax를 일정 주기로 호출하여 새로운 쪽지 도착시 [new] 표시
	쪽지 전송	<ul style="list-style-type: none"> • 게시글/댓글 작성자 id 클릭 시 모달로 쪽지 전송
메인화면 (여행게시판)	무한스크롤	<ul style="list-style-type: none"> • 게시물 9개씩 출력, 무한스크롤
	검색	<ul style="list-style-type: none"> • 검색어/해시태그/지역선택으로 검색
	목록	<ul style="list-style-type: none"> • 게시글마다 좋아요수/댓글수/조회수 출력 • 선택한 코스 리스트로 출력
여행게시판 상세화면	카카오 지도 api	<ul style="list-style-type: none"> • 선택한 코스를 순서대로 지도위에 마커로 표시한 후 직선으로 연결 (거리 및 소요시간 출력)
	이미지 크게보기	<ul style="list-style-type: none"> • 코스에 추가한 이미지 클릭시 모달로 이미지 크게 띄움
	채팅방 목록	<ul style="list-style-type: none"> • 방만들기 클릭시 채팅방 생성을 위한 모달 띄움 • 비밀번호 입력시 자물쇠 아이콘으로 비밀방 구분 • 참여인원 표시 • session 로그인 정보로 채팅방 참여 여부 확인 후 [참여중]/[참여하기] 표시 • 채팅방 클릭시 해당 채팅방으로 이동
채팅화면	채팅하기	<ul style="list-style-type: none"> • ajax를 일정 주기로 호출하여 채팅내역 출력 • 새로운 채팅내역 있으면 스크롤 맨 아래로 이동

기능정의서

여행게시판 작성화면	카카오 지도 api	<ul style="list-style-type: none"> 지도 표시(이동, 확대축소가능) 및 검색 검색시 해당 위치정보를 리스트로 왼쪽에 표시하고 지도에 마커로 위치 표시 리스트에서 위치정보 클릭시 해당 마커로 지도 중심좌표 이동 리스트와 마커에 마우스오버시 마커 위에 장소 이름 표시 마커 클릭시 '코스 추가?' alert 띄움 (추가는 최대 5개로 제한함) 추가시 하단에 위치 상세정보를 보여주고 작성자가 내용 및 파일을 첨부할 수 있음 추가한 코스는 [코스 보기]를 통해 한눈에 볼 수 있도록 지도를 축소 혹은 확대하고 추가한 순서대로 선을 연결하고 거리, 소요시간(도보, 자전거)을 보여줌
	지역선택	<ul style="list-style-type: none"> 글쓰기 화면 로드시 ajax로 우리나라 지역 이름 가져오고 안보이게 숨겨둠 광역단위 지역 선택시 해당 자치구 선택가능하게 보여줌 클릭시 선택한 자치구 이름을 따로 표시하고, 지도에 자치구 검색 선택했던 자치구 이름 클릭시 체크박스 해제하고 표시했던 글자 지움
	내용 및 파일 첨부	<ul style="list-style-type: none"> 추가한 코스마다 내용과 사진(최대 2장)을 첨부 추가했던 코스 삭제 가능
자유게시판 & QnA	글 작성	<ul style="list-style-type: none"> smart editor 사용하여 글 작성 파일 첨부
	글 목록 페이지징	<ul style="list-style-type: none"> 한 페이지 당 게시물 10개씩 출력, (페이지) 1~10, 이전/다음 페이지
	글 수정	<ul style="list-style-type: none"> 게시글 수정 완료시 수정날짜로 업데이트
	글 목록	<ul style="list-style-type: none"> 게시물에 댓글 있으면 제목 옆 댓글 수 출력 첨부파일 있으면 사진 아이콘 출력
	검색	<ul style="list-style-type: none"> 제목/회원ID로 검색
	파일 다운로드	<ul style="list-style-type: none"> 첨부파일 클릭 시 다운로드 처리
QnA	글 답변	<ul style="list-style-type: none"> 계층형 게시판(gno, ono, nested로 그룹, 순서, 들여쓰기 정도 계산하여 출력)

기능정의서

게시판 공통	댓글/답글	<ul style="list-style-type: none"> • 해당 댓글에 달린 답글 출력 (ajax) • 답글 누르면 답글 보였다 숨기기 (toggle)
	좋아요/싫어요	<ul style="list-style-type: none"> • 좋아요/싫어요 클릭시 db에 저장 후 아이콘 변경 • 게시글 로드시 session 로그인 정보로 좋아요/싫어요 클릭 여부 확인
	사진파일 확장자 체크	<ul style="list-style-type: none"> • 정규식으로 확장자 체크 (gif, jpg, jpeg, png만 가능)
	게시글 수정/삭제	<ul style="list-style-type: none"> • 본인이 작성한 게시글, 댓글만 수정, 삭제 가능
	interceptor	<ul style="list-style-type: none"> • 게시글 작성, 쪽지, 채팅방 url 호출 시 로그인 여부 체크하여 미로그인시 로그인 화면으로 이동
회원가입	아이디/이메일 중복확인	<ul style="list-style-type: none"> • focus out시 ajax로 중복 확인
	비밀번호 유효성 검사	<ul style="list-style-type: none"> • 숫자, 영문, 특수문자 조합으로 8자 이상으로 입력 (정규식)
	비밀번호 입력 확인	<ul style="list-style-type: none"> • focus out시 두개의 비밀번호 일치 여부 확인
로그인	로그인	<ul style="list-style-type: none"> • 로그인 완료시 메인페이지로 이동, 실패시 알림창 띄움 • session에 로그인 정보 저장
	아이디/비밀번호 찾기	<ul style="list-style-type: none"> • 가입시 사용한 정보(이메일, 아이디)로 아이디/비밀번호 찾기
마이페이지	회원정보 수정	<ul style="list-style-type: none"> • 회원정보 수정 완료시 로그아웃 되어 메인페이지로 이동
	로그아웃	<ul style="list-style-type: none"> • session에 저장된 정보 삭제 후 메인페이지로 이동
	내가 쓴 게시물/댓글 확인	<ul style="list-style-type: none"> • session에 저장된 회원정보 ajax로 전달하여 리스트 출력
	좋아요 누른 게시물 확인	<ul style="list-style-type: none"> • 클릭 시 해당 게시물로 이동
	내가 참여한 방 확인	
	받은 쪽지함/보낸 쪽지함	<ul style="list-style-type: none"> • 받은 쪽지 클릭시 읽음처리 후 답장 가능하도록 모달 띄움 • 쪽지 삭제(상대방 쪽지함에서는 삭제되지 않음)

Part 3

프로젝트 구현



Part 3

ElasticSearch

ourtrip 인덱스 생성과 mapping

```
PUT ourtrip
{
  "settings": {
    "analysis": {
      "analyzer": {
        "korean": {
          "tokenizer": "nori_tokenizer"
        },
        "standard": {
          "tokenizer": "standard"
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "board_no": {"type": "long"},
      "board_name": {"type": "text"},
      "id": {"type": "text", "analyzer": "korean"},
      "title": {"type": "text", "analyzer": "korean"},
      "writedate": {"type": "date", "format": "yyyy-MM-dd HH:mm:ss.S"},
      "conLst": {
        "properties": {
          "content": {"type": "text", "analyzer": "korean"}
        }
      },
      "content": {"type": "text", "analyzer": "korean"},
      "hashLst": {
        "properties": {
          "hashtag_name": {"type": "text", "analyzer": "korean"}
        }
      },
      "mapLst": {
        "properties": {
          "place_name": {"type": "text", "analyzer": "korean"},
          "address_name": {"type": "text", "analyzer": "korean"}
        }
      }
    }
  }
}
```

① 한글 검색을 위해
nori_tokenizer 사용

hashtag	hashtag_no	해시태그번호	int
	hashtag_name	해시태그내용	varchar

③ hashtag, category
두 테이블을 조인해서
hashtag_name으로 put

② 여행게시판은 하나의 게시물에
여러 개의 내용, 해시태그, 지도(장소정보)가
존재하기 때문에 list로 저장

boardTravel	board_no	게시글번호	int
	board_name	게시판구분	varchar
	id	작성자ID	varchar
	member_no	작성자번호	int
	title	글제목	varchar
	viewcount	조회수	int
	writedate	작성일자	timestamp
	update date	수정일자	timestamp

boardcourse	board_no	게시글번호	int
	course_no	코스번호	int
	board_name	게시판구분	varchar
	content	게시글내용	text

category	board_no	게시글번호	int
	board_name	게시판구분	varchar
	hashtag_no	해시태그번호	int
	region_no	지역번호	int

map	board_no	게시글번호	int
	course_no	코스번호	int
	place_name	장소이름	varchar
	x	좌표	varchar
	y	좌표	varchar
	address_name	주소	varchar
	road_address_name	도로명주소	varchar
	phone	전화번호	varchar
	place_url	장소url	varchar
	id	지역ID	varchar
	category_group_code	그룹코드	varchar
	category_group_name	그룹이름	varchar
	category_name	카테고리이름	varchar

boardQna	board_no	게시글번호	int
	board_name	게시판구분	varchar
	id	작성자ID	varchar
	member_no	작성자번호	int
	title	글제목	varchar
	content	글내용	text
	viewcount	조회수	int
	writedate	작성일자	timestamp
	update date	수정일자	timestamp
	filename_org	파일원본	varchar
	filename_server	파일서버	varchar
	gno	그룹번호	int
	ono	순서	int
	nested	들여쓰기정도	int

boardFree	board_no	게시글번호	int
	board_name	게시판구분	varchar
	id	작성자ID	varchar
	member_no	작성자번호	int
	title	글제목	varchar
	content	글내용	text
	viewcount	조회수	int
	writedate	작성일자	timestamp
	update date	수정일자	timestamp
	filename_org	파일원본	varchar
	filename_server	파일서버	varchar

Part 3

ElasticSearch

Mysql에서
Elasticsearch로
데이터 이관

```
// put document
@GetMapping("/put.do")
@ResponseBody
public List<Map<String, Object>> put() {
    List<Map<String, Object>> list = new ArrayList<Map<String, Object>>();
    int j = 1;
    for (int i=1; i<3; i++) {
        Map<String, Object> res = elaMapper.tb(i);
        System.out.println(res);
        List<Map<String, Object>> conLst = elaMapper.tb_content(i);
        List<Map<String, Object>> hashLst = elaMapper.tb_hash(i);
        List<Map<String, Object>> mapLst = elaMapper.tb_map(i);

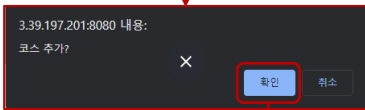
        Map<String, Object> doc = new HashMap<String, Object>();
        doc.put("board_no", res.get("board_no"));
        doc.put("board_name", res.get("board_name"));
        doc.put("id", res.get("id")); doc.put("title", res.get("title"));
        doc.put("writedate", res.get("writedate").toString());
        doc.put("conLst", conLst);
        doc.put("hashLst", hashLst);
        doc.put("mapLst", mapLst);
        list.add(doc);

        try {
            ElasticSearch.putDocument("ourtrip", Integer.toString(j), doc);
        } catch (Exception e) {
            e.printStackTrace();
        }
        j++;
    }
}
```

ElasticsearchController.java

GET ourtrip/_search 결과

```
1 {
2   "took" : 4,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 3,
13      "relation" : "eq"
14    },
15    "max_score" : 1.2377291,
16    "hits" : [
17      {
18        "_index" : "ourtrip",
19        "_type" : "_doc",
20        "_id" : "6",
21        "_score" : 1.2377291,
22        "_source" : {
23          "board_name" : "질문게시판",
24          "board_no" : 21,
25          "writedate" : "2022-11-28 05:48:31.0",
26          "id" : "test",
27          "title" : "질문 아무내용입니다.",
28          "content" : "<p>아무내용 썼다</p>"
29        }
30      },
31      {
32        "_index" : "ourtrip",
33        "_type" : "_doc",
34        "_id" : "1",
35        "_score" : 0.22920428,
36        "_source" : {
37          "board_name" : "여행게시판",
38          "conLst" : [
39            {
40              "content" : "아무내용~1"
41            },
42            {
43              "content" : "아무내용~2"
44            }
45          ]
46        }
47      }
48    ]
49  }
50 }
```

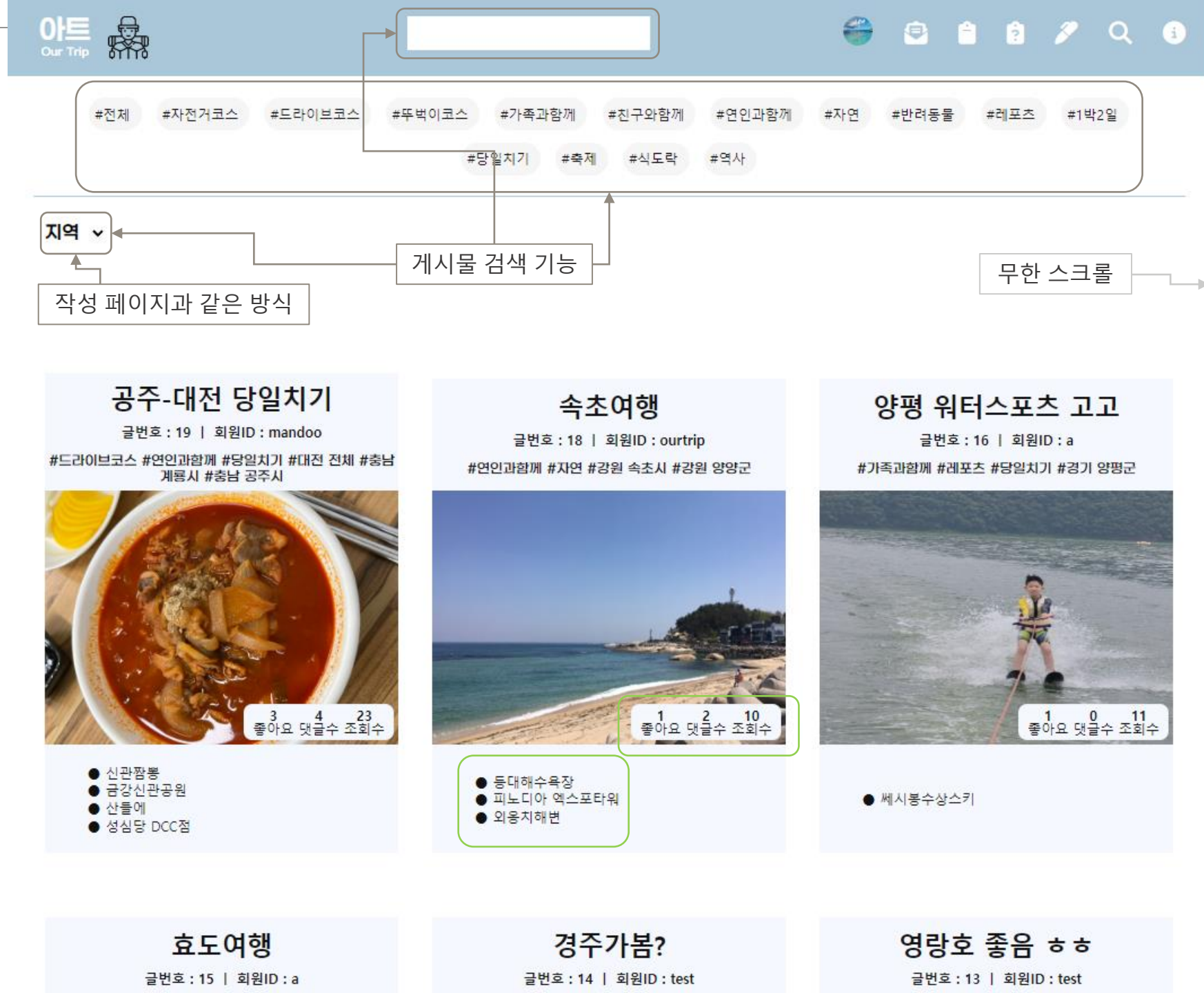



마커 클릭으로
코스 추가

코스별로
사진 최대 **2장** 가능
(미리보기, 확장자 검사)

Part 3

메인 페이지



Part 3

상세보기



속초여행

03 13 ♥️ 🗑️

03 12 ♥️ 🗑️

#연인과함께 #자연 #강원 속초시 #강원 양양군

해당 페이지 좋아요 싫어요 표시



방번호	방제목	방장	시작일	종료일	참여인원
1	이 코스로 같이 여행가실분?	ourtrip	2022-12-14	2022-12-16	2명 참여하기
2	고성-속초-강릉 자차 있으신분!!	mandoo	2022-12-26	2022-12-29	1명 참여하기
3	비밀 방~~~🔒	ourtrip	2023-01-16	2023-01-19	1명 참여하기

참여여부, 참여인원, 비밀번호 있는 방(아이콘) 표시

방번호

방제목

1 이 코스로 같이 여행가실분?

2 고성-속초-강릉 자차 있으신

3 비밀 방~~~🔒

Make Room

*방 제목

*방 내용

[여행 시작날짜와 종료날짜를 선택해주세요]

*시작날짜:

*종료날짜:

비밀번호:

방만들기

방 만들기 모달

3 비밀 방~~~🔒

Send Message

To: ourtrip

From: ourtrip

보낼 메시지를 입력하세요.

Send

쪽지 보내기 모달

등대해수욕장

☎ 033-639-2027

📍 강원 속초시 영랑동 1-9

해수욕장~~~ 🌊

피노디아 익스포트타워

☎ 033-633-5083

📍 강원 속초시 조양동 1544-1

올라갈 수 있음

외옹치해변

☎

📍 강원 속초시 대포동 594-2

산책로 있음



사진 보기 모달

mandoo 🗨️ 연말에 친구들이랑 속초 여행하려고 하는데.. 맛집 추천좀 해주세욤!! [답글] (1)

답글을 작성해주세요.

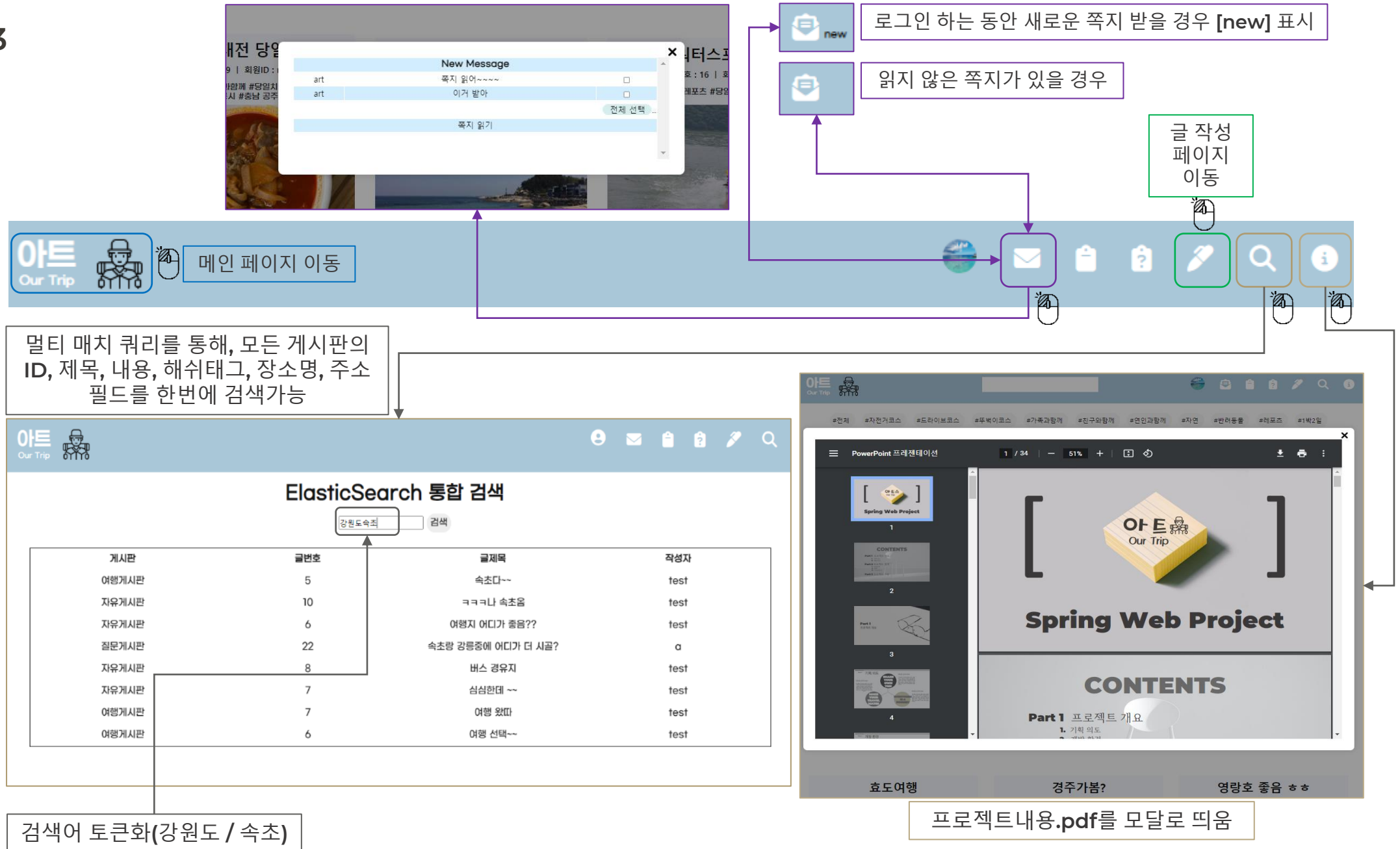
banana 🗨️ 버스타고 가시나요 아님 차로 가시나요?? 22-12-06 09:03:24

수정 삭제

댓글 답글창 열기

Part 3

헤더



A close-up, low-angle shot of a Go board. The board is made of light-colored wood with dark lines forming a grid. Numerous black and white Go stones are scattered across the board, some in the foreground and others further back, creating a sense of depth. The lighting is warm, casting soft shadows from the stones.

OMOK

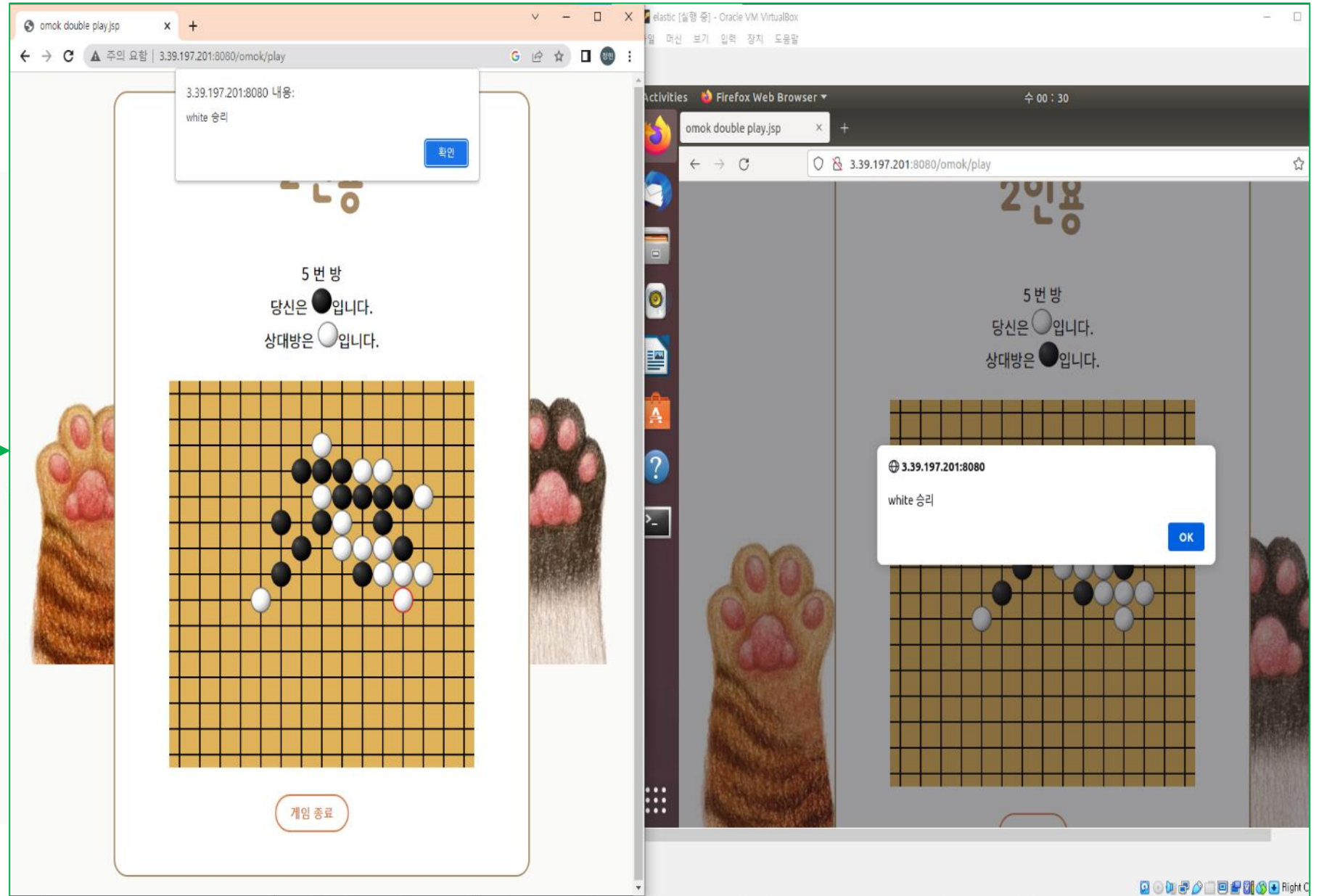
Mini Web Project

Mini Web Project OMOK

온라인에서 2인용 가능.

빨간색 테두리로 상대방이 둔 마지막 돌 표시.

select
game mode



프로젝트

<http://3.39.197.201:8080>

Github

https://github.com/aqwsde321/pj_public_test