

# **Instagram Clone Coding(Instacorn)**

김동연 김종민 양정윤 오승빈 이채은 정은정 한다솔

**instacorn**

# Contents

---

- 01. Subject**
- 02. Introduction**
- 03. Development Environment**
- 04. Configuration**
- 05. ERD design**
- 06. Main Function**
- 07. Testing**
- 08. Page Explanation**
- 09. Improvements**

# Subject

---

Django를 더 깊이 이해하기 위해 프론트엔드, 백엔드, 디자인을 모두 포함한 인스타그램 클론 코딩 프로젝트를 선택했습니다. 다양한 기능과 많은 사람들이 사용하는 SNS를 직접 구현함으로써, 웹 개발의 전반적인 기술 향상을 목표를 했습니다. 이 프로젝트를 통해 Django의 여러 기능을 실습하고, 실제로 작동하는 소셜 미디어 플랫폼을 구축해보며 개발자로서의 실력을 한층 더 높이고자 합니다.



# Introduction



김동연

ERD 설계, 검색, 업로드



김종민

ERD 설계, 마이페이지, 메인페이지



양정윤

화면 설계, 상세페이지



오승빈

화면 설계, 업로드



이채은

관리자 페이지, 코드병합



정은정

화면 설계, 회원가입, 로그인, 회원정보 수정



한다솔

DB 서버 구축

## Development Environment

---

HTML



CSS



JS



django



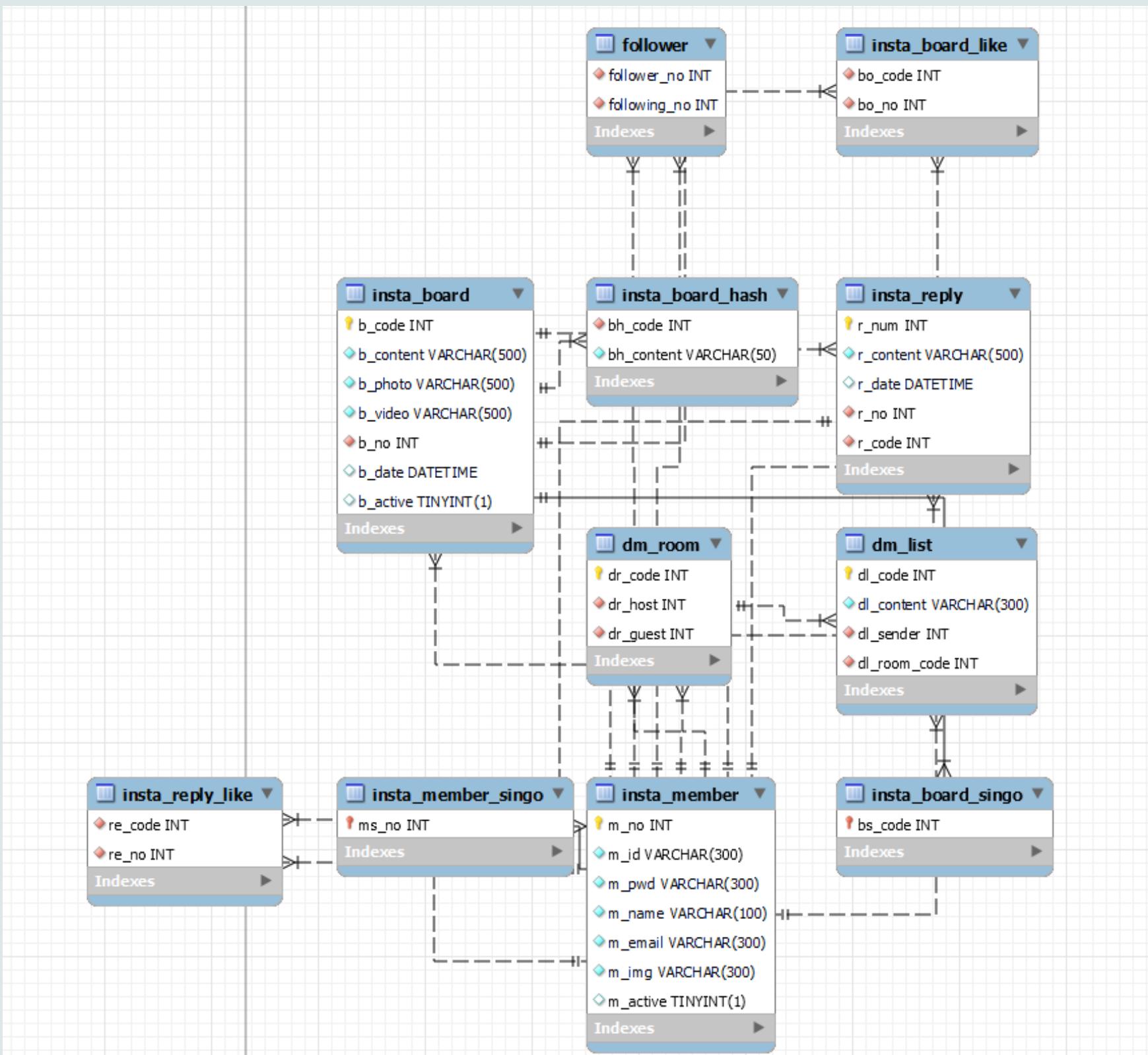
python™

MySQL®

# Configuration

환경설정	db 서버 구축	다솔	완료됨 ▼	2024. 6. 17	2024. 6. 18	마이페이지	프론트	종민	완료됨 ▼	2024. 6. 19	2024. 6. 20
	프로젝트 생성?, 공유	동연, 채은	완료됨 ▼	2024. 6. 17	2024. 6. 18		팔로우/팔로잉		완료됨 ▼	2024. 6. 19	2024. 6. 24
ERD 설계		동연, 종민	완료됨 ▼	2024. 6. 15	2024. 6. 18		좋아요 보관함		완료됨 ▼	2024. 6. 20	2024. 6. 20
화면설계	전체적인 스타일 잡기?	승빈, 정윤, 은정	완료됨 ▼	2024. 6. 15	2024. 6. 17		회원신고		완료됨 ▼	2024. 6. 24	2024. 6. 24
로그인	프론트	은정	진행 중 ▼	2024. 6. 18	yyyy. m. d	검색	프론트	동연	완료됨 ▼	2024. 6. 19	yyyy. m. d
	로그인 기능		완료됨 ▼	2024. 6. 18	2024. 6. 19		검색 기능		완료됨 ▼	2024. 6. 17	2024. 6. 19
	로그아웃		진행 중 ▼	2024. 6. 18	yyyy. m. d	상세페이지	프론트	정윤	진행 중 ▼	2024. 6. 18	yyyy. m. d
	이용제한 회원 처리	채은	완료됨 ▼	2024. 6. 20	2024. 6. 20		게시물 수정/삭제		완료됨 ▼	2024. 6. 18	2024. 6. 20
회원가입	프론트	은정	진행 중 ▼	2024. 6. 18	yyyy. m. d		댓글		완료됨 ▼	2024. 6. 19	2024. 6. 20
	회원가입		완료됨 ▼	2024. 6. 18	2024. 6. 18	채팅	채팅	다솔	완료됨 ▼	2024. 6. 18	2024. 6. 19
	이메일, 아이디 중복 X		완료됨 ▼	2024. 6. 20	2024. 6. 20	관리자	관리자 페이지 네이밍, 다큐기	채은	진행 중 ▼	2024. 6. 24	yyyy. m. d
	이메일 인증		진행 중 ▼	2024. 6. 18	yyyy. m. d		관리자 계정 로그인/로그아웃		완료됨 ▼	2024. 6. 19	2024. 6. 20
회원정보수정	프론트	은정	시작 전 ▼	yyyy. m. d	yyyy. m. d		신고내역 리스트		완료됨 ▼	2024. 6. 20	2024. 6. 23
	수정 기능		완료됨 ▼	2024. 6. 19	2024. 6. 20		회원 이용제한/복구		완료됨 ▼	2024. 6. 19	2024. 6. 23
	비밀번호 변경		진행 중 ▼	2024. 6. 19	yyyy. m. d		포스트 신고/복구		완료됨 ▼	2024. 6. 23	yyyy. m. d
파일업로드	페이지 레이아웃	승빈	완료됨 ▼	2024. 6. 19	2024. 6. 20		관리자 추가/권한		시작 전 ▼	yyyy. m. d	yyyy. m. d
	업로드 기능 구현	승빈, 동연	완료됨 ▼	2024. 6. 17	yyyy. m. d	통합		채은	진행 중 ▼	2024. 6. 24	yyyy. m. d
메인페이지	프론트	종민	완료됨 ▼	2024. 6. 18	2024. 6. 19		검색오류/사이드바		완료됨 ▼	2024. 6. 24	yyyy. m. d
	전체 포스트 출력		완료됨 ▼	2024. 6. 18	2024. 6. 18		상세페이지 스타일		완료됨 ▼	2024. 6. 24	yyyy. m. d
	신고 포스트 블러		완료됨 ▼	2024. 6. 18	2024. 6. 19		로그인/회원가입 스타일		완료됨 ▼	2024. 6. 24	yyyy. m. d
	좋아요		시작 전 ▼	yyyy. m. d	yyyy. m. d						
프로필 사진 -> 회원페이지(마이페이지) 이동			완료됨 ▼	2024. 6. 20	2024. 6. 20						

# ERD design



- | dm\_room
- | follower
- | insta\_board
- | insta\_board\_hash
- | insta\_board\_like
- | insta\_board\_singo
- | insta\_member
- | insta\_member\_singo
- | insta\_reply
- | insta\_reply\_like
- | messages

instacorn

# Main Function

---

member join	login	member modify	manager	main_page	my_page	detail_page
회원가입	로그인	정보 수정	신고 관리	업로드	프로필 사진변경	게시물 상세
비밀번호재설정			계정 관리	좋아요	팔로우, 팔로워	댓글
				검색	게시물	
					좋아요	

# Testing



instacorn

페이스북으로 로그인

OR

이메일 주소

성명

사용자 아이디

비밀번호

이미 존재하는 이메일입니다. 다른 이메일을 사용해주세요.

가입하기

People who use our service may have uploaded your contact information to Instagram. [Learn More](#)

By signing up, you agree to our [Terms](#), [Privacy Policy](#) and [Cookies Policy](#)

계정이 있으신가요? [로그인](#)

# Page Explanation - member join

```

with connection.cursor() as cursor:
    msg="select * from insta_member where m_id = (%s)"
    cursor.execute(msg, (m_id, ))
    data_id = cursor.fetchall()
    print('data_id', data_id)

    if len(data_id) != 0:
        messages.error(request, '이미 존재하는 아이디입니다. 다른 아이디를 사용해주세요.')
        return redirect('insjoin.do')

    msg="select * from insta_member where m_email = (%s)"
    cursor.execute(msg, (m_email, ))
    data_email = cursor.fetchall()
    print('data_email', data_email)

    if len(data_email) != 0:
        messages.error(request, '이미 존재하는 이메일입니다. 다른 이메일을 사용해주세요.')
        return redirect('insjoin.do')

    m_salt, hashedpw = hashing_password(m_pwd)
    m_img = "profile02.png"
    msg="insert into insta_member(m_id, m_pwd, m_name, m_email, m_salt, m_img) values(%s, %s, %s, %s, %s, %s)"
    cursor.execute(msg, (m_id, hashedpw, m_name, m_email, m_salt, m_img))

    if '@' not in m_email:
        messages.error(request, '유효한 이메일 주소를 입력해주세요.')
        return render(request, 'instacorn/insjoin.html')

```

# instacorn

 페이스북으로 로그인

OR

이메일을 입력해주세요

비밀번호

success! 회원가입 완료, 로그인을 해주세요.

로그인

[비밀번호 찾기](#)

People who use our service may have uploaded  
your contact information to Instagram. [Learn More](#)

By signing up, you agree to our [Terms](#), [Privacy Policy](#) and [Cookies Policy](#)

계정이 없으신가요? [가입하기](#)

## Page Explanation - login

```
# auth_user 테이블에 없는 경우 insta_member 테이블에서 확인
cursor = connection.cursor()
msg = "SELECT m_no, m_email, m_pwd, m_salt, m_active FROM insta_member WHERE m_email = %s"
cursor.execute(msg, [m_email])
data = cursor.fetchone()

# 이메일이 틀렸을 때
if not data:
    messages.error(request, '이메일이 존재하지 않습니다')
    return redirect('inslogin.do')

m_no, m_email, regpw, m_salt, m_active = data

# MD5와 저장된 솔트를 사용하여 비밀번호를 검증
hashedpw = hashlib.md5(str(m_pwd + m_salt).encode('utf-8')).hexdigest()

if hashedpw != regpw:
    messages.error(request, '비밀번호가 일치하지 않습니다')
    return redirect('inslogin.do')
```

# Page Explanation - pwd

## 비밀번호 재설정

비밀번호를 잊어버리셨나요?  
가입하신 이메일을 입력해 주세요.  
잠시 후 해당 이메일로 비밀번호 재설정 메일이 발송됩니다.

이메일:

## 비밀번호 재설정 메일 발송 완료

비밀번호 재설정을 위한 메일이 발송되었습니다.  
30분 내에 비밀번호를 재설정 해주세요.  
만약 메일이 오지 않는다면, 이메일 주소를 다시 한번 확인해 주시고, 스팸 폴더를 확인해 주세요.

## 비밀번호 재설정



나에게 ▾

안녕하세요!

다음 링크를 누르시면 본인 계정의 비밀번호를 재설정 할 수 있는 화면으로 이동합니다.

<http://127.0.0.1:8000/reset/Mg/cab72k-42704f264f42b0f2f4a32226e265265e/>

비밀번호 재설정을 요청하지 않았다면 이 이메일을 무시하셔도 됩니다.  
감사합니다.

## 비밀번호 재설정

새 비밀번호를 입력해 주세요.

새 비밀번호:

- 다른 개인 정보와 유사한 비밀번호는 사용할 수 없습니다.
- 비밀번호는 최소 8자 이상이어야 합니다.
- 통상적으로 자주 사용되는 비밀번호는 사용할 수 없습니다.
- 숫자로만 이루어진 비밀번호는 사용할 수 없습니다.

새 비밀번호 (확인):

# Page Explanation - pwd

## 비밀번호 재설정

비밀번호를 잊어버리셨나요?

가입하신 이메일을 입력해 주세요.

잠시 후 해당 이메일로 비밀번호 재설정 메일이

이메일:

전송

## 비밀번호 재설정 메일 발송 완료

비밀번호 재설정을 위한 메일이 발송되었습니다.

30분 내에 비밀번호를 재설정 해주세요.

만약 메일이 오지 않는다면, 이메일 주소를 다시 한번 확인하

```
mysql> select * from auth_user;
```

id	password	
is_active	date_joined	
1	pbkdf2_sha256\$720000\$50I8yu7E5067ImupJo8tqa\$mZZsjU4hNVx5ew6YJqbqY3FcG/s98iC2qMkftetCJ5Q=	
	1   2024-06-19 07:57:31.646226	
2	pbkdf2_sha256\$720000\$QzI4ooNijUaDtUsqpcmbII\$LqJV+5Kp1TMJxVUI8QZUYvpb3Enq1aY6C8BiHzeusq8=	
	1   2024-07-17 03:34:30.774965	

```
2 rows in set (0.00 sec)
```

```
mysql> select * from auth_user;
```

id	password	
is_active	date_joined	
1	pbkdf2_sha256\$720000\$50I8yu7E5067ImupJo8tqa\$mZZsjU4hNVx5ew6YJqbqY3FcG/s98iC2qMkftetCJ5Q=	
	1   2024-06-19 07:57:31.646226	
2	pbkdf2_sha256\$720000\$dZ1ktb5MODPqwGfmxTQJW9\$AkZT6Cd0rF4LFFB6/t+LoBWGu5Xae/RatII++iEGCZo=	
	1   2024-07-17 03:34:30.774965	

세요.

사용한 비밀번호는 사용할 수 없습니다.

가 이상이어야 합니다.

용되는 비밀번호는 사용할 수 없습니다.

비밀번호는 사용할 수 없습니다.

이메일 :

jungyn@gmail.com

아이디 :

jungyn

이름 :

이름수정

수정하기

## 비밀번호 변경

현재 비밀번호 :

현재 비밀번호를 입력하세요.

새 비밀번호 :

변경할 비밀번호를 입력하세요.

새 비밀번호 확인 :

비밀번호를 한 번 더 입력해주세요.

변경하기

Page Explanation -  
member modify

```
if request.method=="GET":  
    try:  
        m_no = request.session.get('m_no')  
    except KeyError:  
        messages.error(request, '로그인 정보가 없습니다.')  
        return redirect('inslogin.do')  
  
msg = "select * from insta_member where m_no = %s"  
  
try:  
    with connection.cursor() as cursor:  
        cursor.execute(msg, [m_no])  
        data = cursor.fetchone()  
  
    if data:  
        m_email = data[5]  
        m_id = data[1]  
        m_name = data[4]  
        return render(request, 'instacorn/insmember_modify.html', {  
            'm_email': m_email, 'm_id': m_id, 'm_name': m_name,  
        })
```

관리

신고된 게시물	<a href="#">+ 추가</a>	<a href="#">변경</a>
신고된 회원	<a href="#">+ 추가</a>	<a href="#">변경</a>
전체 게시물	<a href="#">+ 추가</a>	<a href="#">변경</a>
전체 회원	<a href="#">+ 추가</a>	<a href="#">변경</a>
인증 및 권한		
그룹	<a href="#">+ 추가</a>	<a href="#">변경</a>
사용자(들)	<a href="#">+ 추가</a>	<a href="#">변경</a>

최근 활동

나의 활동

- ✖ 26 신고된 게시물
- ✍ User 99 전체 회원
- ✍ content111; 전체 게시물
- ✍ User 99 전체 회원
- ✍ User Five 전체 회원
- ✍ User 99 전체 회원

```
#신고회원(비활성 처리 후 신고리스트에서 삭제)
@admin.register(InstaMemberSingo)
class InstaMemberSingoAdmin(admin.ModelAdmin):
    list_display = ('ms_no_display', 'reported_user', 'm_active_status')
    readonly_fields = ('ms_no',)
    actions = ['set_inactive']

    def ms_no_display(self, obj):
        return obj.ms_no.m_no

    def reported_user(self, obj):
        return obj.ms_no.m_id

    def m_active_status(self, obj):
        return obj.ms_no.m_active

    m_active_status.short_description = 'User Active Status'

    def set_inactive(self, request, queryset):
        updated_count = 0
        for singo in queryset:
            singo.ms_no.m_active = False
            singo.ms_no.save()
            updated_count += 1

        singo.delete()

        self.message_user(request, f'{updated_count}건이 계정 비활성 처리 되었습니다.')

    set_inactive.short_description = '선택된 계정들을 비활성 처리 합니다.'
```

# Page Explanation - manager

필터에 타이핑 시작...	
관리	
신고된 게시물	+ 추가
신고된 회원	+ 추가
전체 게시물	+ 추가
전체 회원	+ 추가
인증 및 권한	
그룹	+ 추가
사용자(들)	+ 추가

변경할 신고된 게시물 선택

선택한 항목들을 삭제합니다.

선택된 게시물을 습김 처리 합니다.

REPORTED CONTENT	B ACTIVE STATUS
testing2	True
archi3	True
fu	True

신고된 게시물

instacorn

```
#신고된 게시물
@admin.register(InstaBoardSingo)
class InstaBoardSingoAdmin(admin.ModelAdmin):
    list_display = ('bs_code_display', 'reported_content', 'b_active_status')
    readonly_fields = ('bs_code',)
    actions = ['set_inactive']

    def bs_code_display(self, obj):
        return obj.bs_code.b_code

    def reported_content(self, obj):
        return obj.bs_code.b_content

    def b_active_status(self, obj):
        return obj.bs_code.b_active

    #b_active_status.short_description = '게시물 상태'

    def set_inactive(self, request, queryset):
        updated_count = 0
        for singo in queryset:
            singo.bs_code.b_active = False
            singo.bs_code.save()
            updated_count += 1

        singo.delete()

    self.message_user(request, f'{updated_count}건이 게시물 습김 처리 되었습니다.')
```

# Page Explanation - manager

새 게시물

close

파일 선택 선택된 파일 없음

127.0.0.1:8000 내용:  
업로드 성공

확인

새 게시물

share



# Page Explanation - main\_page : upload

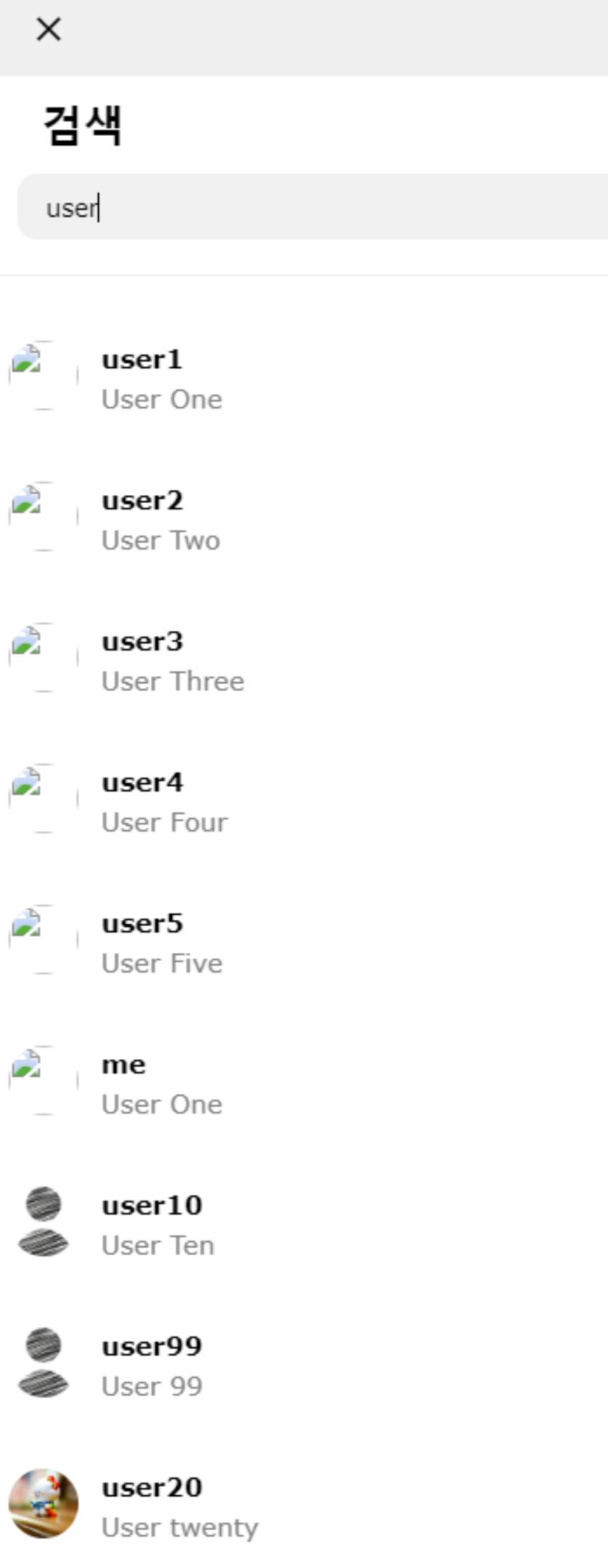
```

function upload(file){
  $(document).ready(function(){
    $('#submit_modal').on('click', function(){
      var csrftoken = document.querySelector("meta[name=csrf_token]").content
      var form = document.getElementById('uploadForm');
      var formData = new FormData(form);

      // 폼 밖에 있는 데이터 추가
      var fileInput = file
      formData.append('b_photo', file);

      // alert(' b_content = ' + b_content + ' b_photo = ' + b_photo + ' b_no = ' + b_no)
      $.ajax({
        url: '/uploadsave.do/',
        type: 'POST',
        data: formData,
        processData: false,
        contentType: false,
        headers: {"X-CSRFToken": csrftoken},
        success: function(data) {
          alert('업로드 성공')
          modal.style.display = "none";
        },
        error: function() {
          alert('업로드 실패')
        }
      })
    })
  })
}

```



# Page Explanation - main\_page : search



UHAN

신고

취소



UHAN

...



# Page Explanation - main\_page : blur

```
#게시물 신고
def board_singo(request):
    b_code = request.GET.get('b_code')

    cursor = connection.cursor()

    msg_sel = f"select count(*) from insta_board_singo where bs_code='{b_code}'"
    cursor.execute(msg_sel)
    b_singo_cnt = cursor.fetchone()[0]

    if b_singo_cnt==0:
        msg_insert = f"insert into insta_board_singo values({b_code})"
        cursor.execute(msg_insert)
        connection.commit()

        messages.success(request, '신고되었습니다.')
    else:
        messages.error(request, '이미 신고된 개시물입니다.')

    connection.close()
    return redirect('/home.do/')
```



팔로워

X



cco14

ccomi

삭제



silverjjj

은정

삭제

팔로잉

X



laftelaniking

김동연

팔로잉



silverjjj

은정

팔로잉

instacorn

# Page Explanation - my\_page : follow

```
function del_following(follower_no, following_no){
    $.ajax({
        type:"POST",
        url:"/del_following.do/",
        data:{'follower_no':follower_no, 'following_no':following_no, 'csrfmiddlewaretoken': '{{ csrf_token }}'},
        dataType:"json",

        success: function(response){
            $('#following_btn_'+following_no).hide();
            $('#follow_btn_'+following_no).show();
        },
        error: function(request, status, error){
            alert('실패했습니다.');
        }
    })
} //end

function follow(follower_no, following_no){
    $.ajax({
        type:"POST",
        url:"/follow.do/",
        data:{'follower_no':follower_no, 'following_no':following_no, 'csrfmiddlewaretoken': '{{ csrf_token }}'},
        dataType:"json",

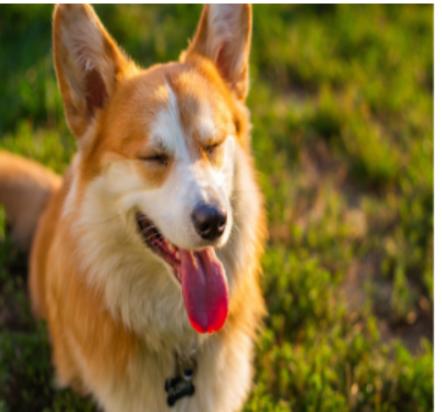
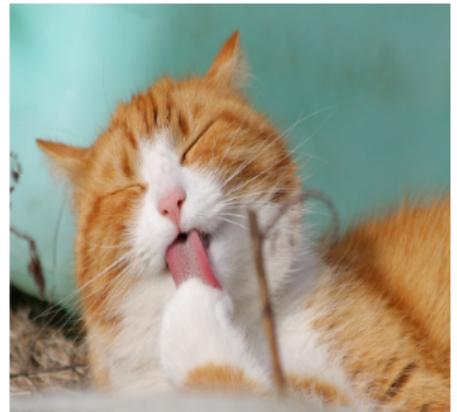
        success: function(response){
            $('#follow_btn_'+following_no).hide();
            $('#following_btn_'+following_no).show();
        },
        error: function(request, status, error){
            alert('실패했습니다.');
        }
    })
} //end
```



jungynnee [프로필 편집](#)

게시물 2 팔로워 2 팔로우 2

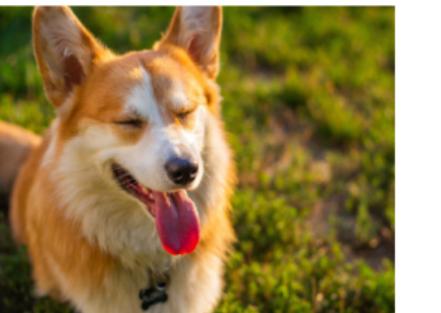
게시물 저장됨 태그됨 좋아요



jungynnee [프로필 편집](#)

게시물 2 팔로워 2 팔로우 2

게시물 저장됨 태그됨 좋아요



Instacorn

홈

검색

메시지

알림

만들기

프로필

더 보기



cco14 [팔로잉](#) [메시지 보내기](#)

게시물 2 팔로워 3 팔로우 3

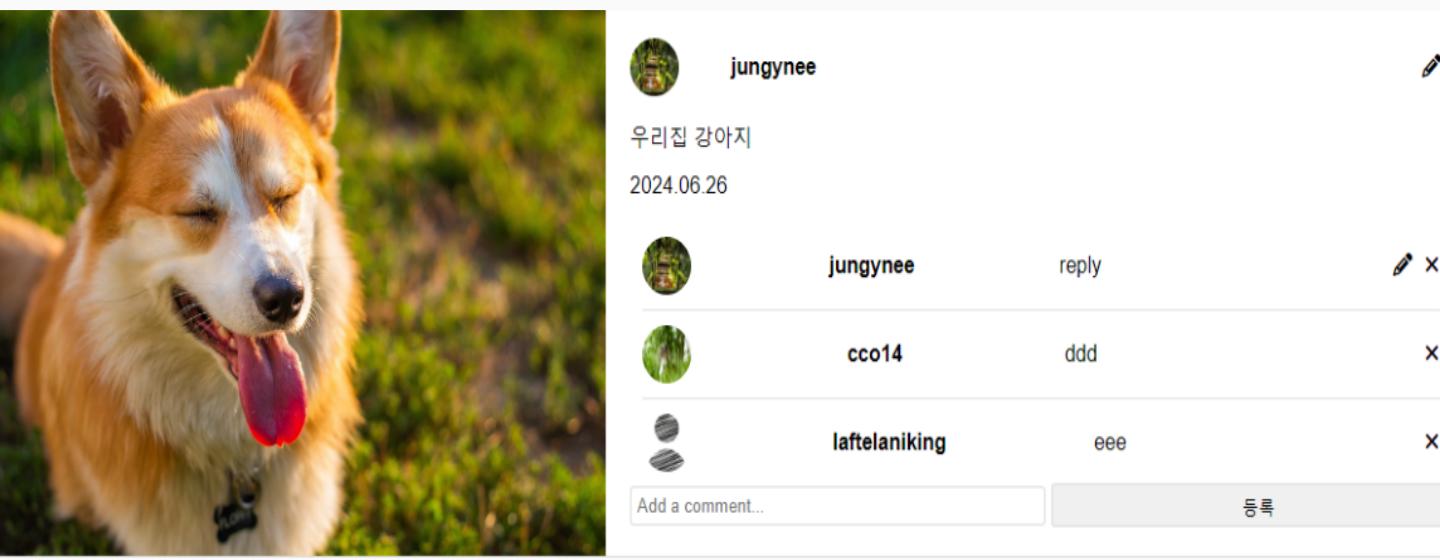
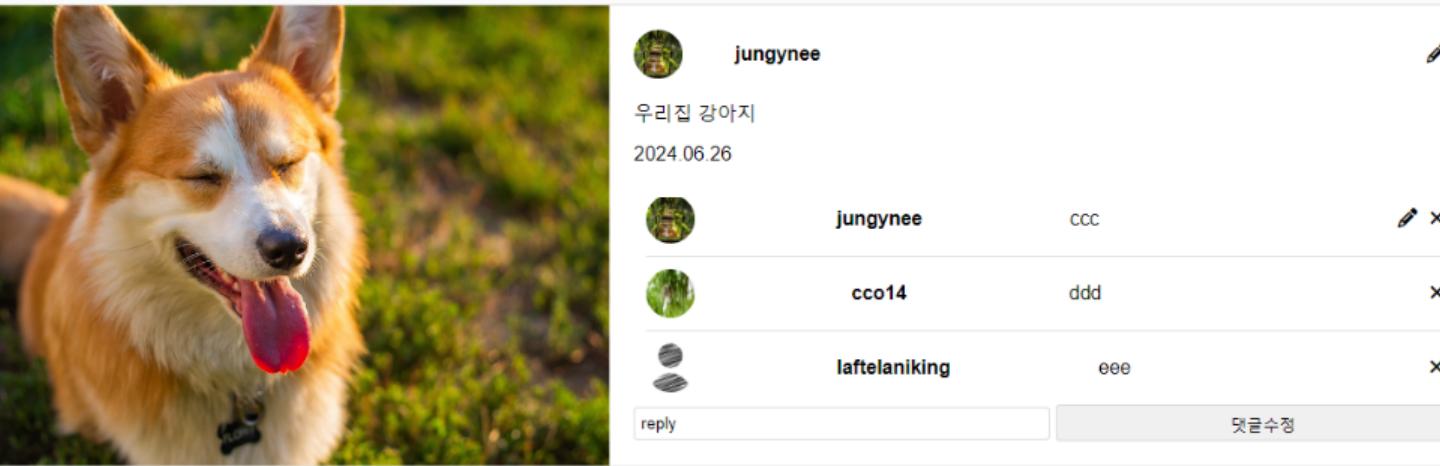
게시물 저장됨 태그됨 좋아요



# Page Explanation - my\_page : tab

```
</div>
</div>
<!-- 본인 게시물 --&gt;
<div class="imagechild">
    {% for item in result_board %}
        <a href="/insdetail.do?idx={{item.b_code}}">
            
        </a>
    {% endfor %}
</div>
<!-- 좋아요 누른 게시물 --&gt;
<div class="imagechild_like" style="display: none;">
    {% for item_like in result_like %}
        
    {% endfor %}
</div>
```

# Page Explanation - detail\_page



```
# 상세페이지
def insdetail(request):
    m_no = request.session.get('m_no')
    print('m_no', m_no)

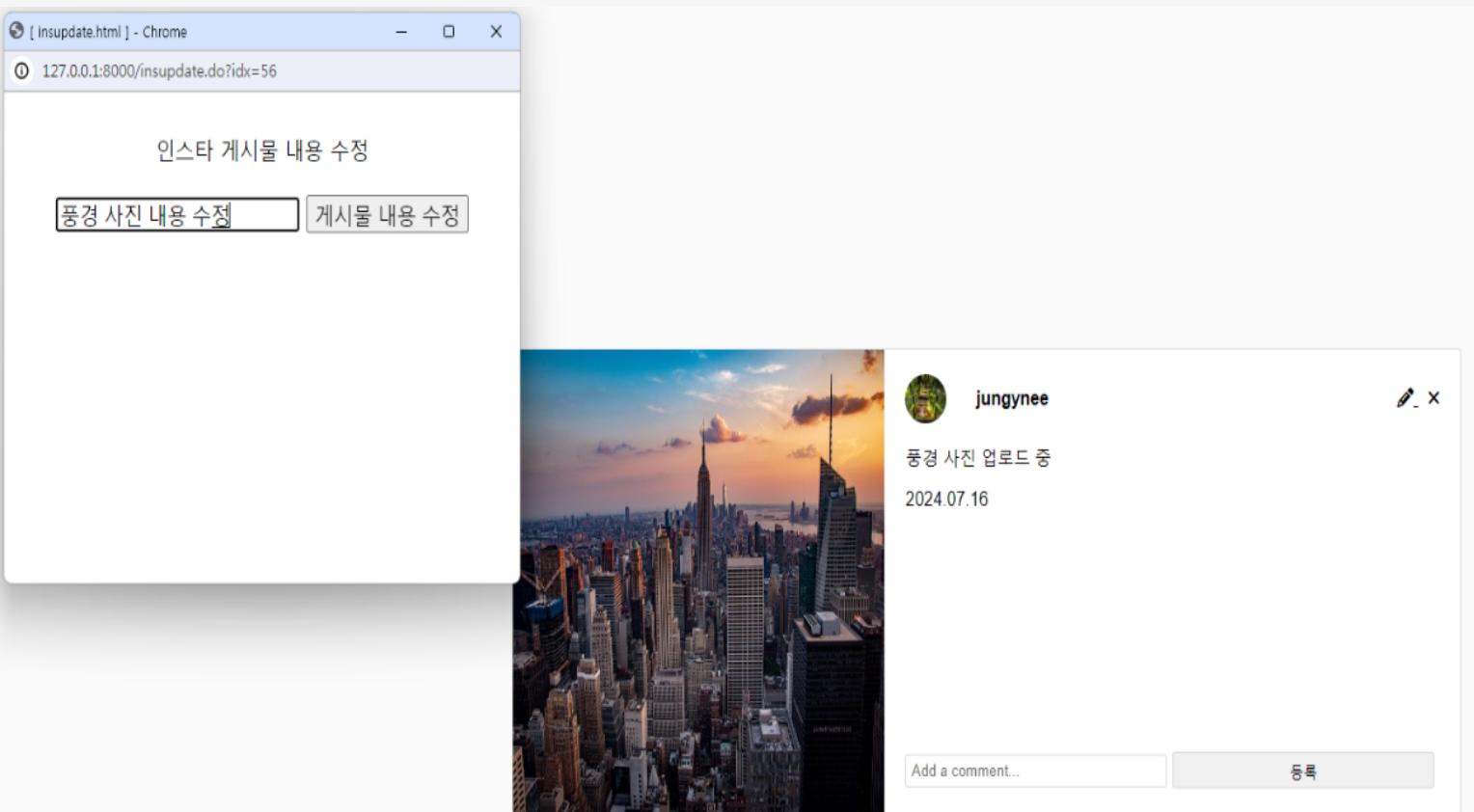
    idx = request.GET.get('idx')
    print('idx =', idx)
    msg = "select b.*, m.* from insta_board b inner join insta_member m on b.b_no = m.m_no where b_code = %s"
    cursor = connection.cursor()
    cursor.execute(msg, [idx])
    rows = cursor.fetchall()

    result = []
    for data in rows:
        row = {'b_code':data[0], 'b_content':data[1], 'b_photo':data[2], 'b_no':data[3], 'b_date':data[4],
               'b_active':data[5], 'm_id': data[7], 'm_name': data[10], 'm_img': data[12]}
        result.append(row)

    rmmsg = "select r.*, m.* from insta_reply r inner join insta_member m on r.r_no = m.m_no where r_code = " + idx
    cursor = connection.cursor()
    cursor.execute(rmmsg)
    rows = cursor.fetchall()
    connection.commit()

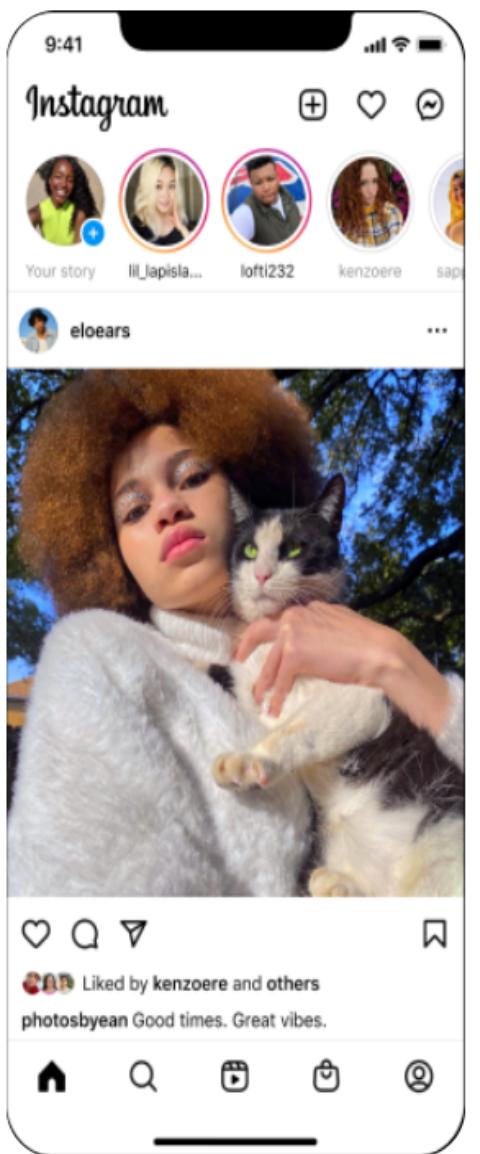
    msg_singo = f"select count(*) from insta_board_singo where bs_code={idx}"
    cursor.execute(msg_singo)
    b_singo_cnt = cursor.fetchone()[0]

connection.close()
```



# Page Explanation - detail\_page2

```
<div class="post-header">
    
    <div class="user-info">
        <span class="username"> {{ i.m_id }} </span>
    </div>
    <div align="right">
        {% if i.b_no == m_no %}
            <a href="javascript:void(0);"
                onclick="openPopup('insupdate.do?idx={{ idx }}')"
            >
                <i class="fas fa-pencil-alt"></i>
            </a> &ampnbsp
            <a href="javascript:void(0);"
                onclick="confirmDelete('{{ idx }}')"
            >
                <i class="fas fa-times"></i>
            </a>
        {% endif %}
    </div>
</div>
```



instacorn

[페이스북으로 로그인](#)

OR

이메일을 입력해주세요

비밀번호

success! 회원가입 완료, 로그인을 해주세요.

[로그인](#)

[비밀번호 찾기](#)

People who use our service may have uploaded your contact information to Instagram. [Learn More](#)

By signing up, you agree to our [Terms](#), [Privacy Policy](#) and [Cookies Policy](#)

[계정이 없으신가요? 가입하기](#)

© 2024 Instacorn

# Improvements

instacorn

# Thank You

---