



pet gallery

<https://github.com/HYUN1103/petgallery>

<http://3.37.178.26:5000>

박현진

Contents

개 요

설 계

개 발

기 능

반려동물과 함께 살아가는 사람들과 소통을
통해
동물에 대한 정보 공유 목적으로 만든
펫 갤러리 사이트 입니다.
채팅, 갤러리, 영상, 게시판, 지도 커뮤니티 기능
이 있습니다.

🕒기간

2024.07.15~2024.08.09

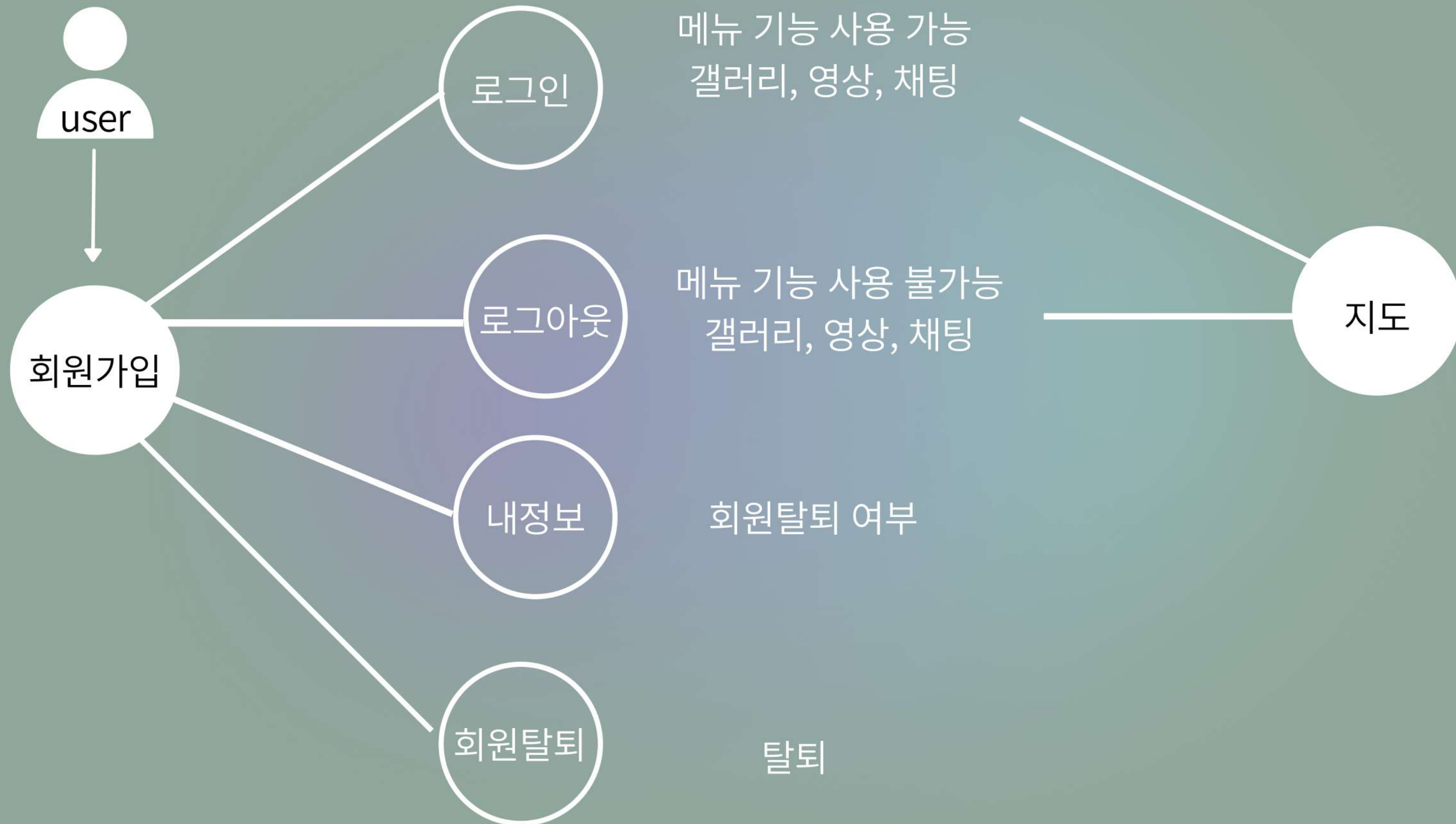
⚙️개발환경

- python 3.10
- vscode
- Flask
- git
- github
- DBbrowser(sqlite3)
- socket
- javascript
- jQuery
- Ajax
- Canva
- AWS

📌기능

- 会員登録
- ログイン
- ログアウト
- ギャラリー
- ビデオ
- 掲示板
- チャット
- 地図

전체 기능구조



- 회원가입 HTML

Signup

User name

password

Check password

email

Create

```
<div class="mb-3">
  <label for="username">User name</label>
  <input type="text" class="form-control" name="username" id="username"
    value="{{ form.username.data or '' }}" />#username.form-control
</div>/.mb-3
<div class="mb-3">
  <label for="password1">password</label>
  <input type="password" class="form-control" name="password1" id="password1"
    value="{{ form.password1.data or '' }}" />#password1.form-control
</div>/.mb-3
<div class="mb-3">
  <label for="password2">Check password</label>
  <input type="password" class="form-control" name="password2" id="password2"
    value="{{ form.password2.data or '' }}" />#password2.form-control
</div>/.mb-3
<div class="mb-3">
  <label for="email">email</label>
  <input type="text" class="form-control" name="email" id="email"
    value="{{ form.email.data or '' }}" />#email.form-control
</div>/.mb-3
<button type="submit" class="btn btn-primary">Create</button>
</form>
```

- 회원가입 views

```
@bp.route('/signup/', methods=('GET', 'POST'))
def signup():
    form = UserCreateForm()
    if request.method == 'POST' and form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if not user:
            try:
                user = User(username=form.username.data,
                             password=generate_password_hash(
                                 form.password1.data),
                             email=form.email.data)
                db.session.add(user)
                db.session.commit()
                return redirect(url_for('main.home'))
            except IntegrityError:
                db.session.rollback()
                flash('이미 존재하는 사용자 이름 또는 이메일입니다.')
        else:
            flash('이미 존재하는 사용자입니다.')
    return render_template('auth/signup.html', form=form)
```


- 로그인 HTML

Login

Username

password

Login

```
{% block content %}
<div class="container-auth">
  <h5 class="my-3 border-bottom pb-2">Login</h5>
  <form method="post">
    {{ form.csrf_token }}
    {% include "form_errors.html" %}
    <div class="mb-3">
      <label for="username">Username</label>
      <input type="text" class="form-control" name="username" id="username"
        value="{{ form.username.data or '' }}">/#username.form-control
    </div>/mb-3
    <div class="mb-3">
      <label for="password">password</label>
      <input type="password" class="form-control" name="password" id="password"
        value="{{ form.password.data or '' }}">/#password.form-control
    </div>/mb-3
    <button type="submit" class="btn btn-primary">Login</button>
  </form>
</div>/container-auth

{% endblock %}
```

- 로그인 views

```
form = UserLoginForm()
# 요청이 POST 방식이고 폼이 유효한 경우
if request.method == 'POST' and form.validate_on_submit():
    error = None # 오류 메시지 변수 초기화
    # 사용자명을 기준으로 사용자 조회
    user = User.query.filter_by(username=form.username.data).first()
    # 사용자가 존재하지 않는 경우
    if not user:
        error = "존재하지 않는 사용자입니다."
    # 비밀번호가 올바르지 않은 경우
    elif not check_password_hash(user.password, form.password.data):
        error = "비밀번호가 올바르지 않습니다."
    # 오류가 없는 경우 (로그인 성공)
    if error is None:
        session.clear() # 세션 초기화
        session['user_id'] = user.id # 사용자 ID를 세션에 저장

        _next = request.args.get('next', '') # 리다이렉트할 URL 가져오기
        if _next:
            return redirect(_next) # 지정된 URL로 리다이렉트
        else:
            return redirect(url_for('main.home')) # 기본 홈 페이지로 리다이렉트

    flash(error) # 오류 메시지를 플래시로 표시
# GET 요청이거나 폼이 유효하지 않은 경우 로그인 템플릿 렌더링
return render_template('auth/login.html', form=form)
```


- 내정보 HTML, views

test1 : information

email: abc@gmail.com

Delete account

```
<div class="container mt-3">
  <li class="list"> {{ g.user.username }} : information </li>
  {{ form.csrf_token }}
  {% include "form_errors.html" %}
  <ul class="list-group">
    <li class="list-group-item"><strong>email:</strong> {{ g.user.email }}</li>
    <!-- 여기에 다른 사용자 정보를 추가 -->
  </ul>/.list-group

  <hr>

  <a class="btn btn-danger" href="{{ url_for('auth.delete_account') }}" \
  onclick="return confirm('Are you sure you want to delete your account?');"> Delete account</a>/.btn.bt
```

```
# 프로필
@bp.route('/profile')
@login_required
def profile():
    form = UserLoginForm()
    return render_template('auth/profile.html', form=form)
```

- 회원탈퇴 HTML, views

Delete account

Are you sure you want to delete the account?
This operation is irreversible

password

Delete Account

close

```
<h5 class="my-3 border-bottom pb-2">Delete account</h5>
<form method="post">
  {{ form.csrf_token }}
  <div class="alert alert-warning" role="alert">
    Are you sure you want to delete the account? <br>This operation is irreversible
  </div>/.alert.alert-warning
  <div class="form-group">
    {{ form.password.label }} <!-- 비밀번호 레이블 -->
    {{ form.password(class="form-control") }} <!-- 비밀번호 입력 필드 -->
  </div>/.form-group
  <button type="submit" class="btn btn-danger">Delete Account</button>
  <a href="{{ url_for('main.home') }}" class="btn btn-secondary">close</a>
```

```
# view
@bp.route('/auth/delete_account', methods=['POST', 'GET'])
@login_required
def delete_account():
    form = DeleteAccountForm()
    if form.validate_on_submit():
        if check_password_hash(g.user.password, form.password.data):
            db.session.delete(g.user)
            db.session.commit()
            flash('계정이 성공적으로 삭제되었습니다.')
            return redirect(url_for('auth.login'))
        else:
            flash('비밀번호가 올바르지 않습니다.')
    return render_template('auth/delete_account.html', form=form)
```


각 기능별 코드

- 갤러리



- HTML, view, CSS
- JavaScript, jQuery, AJAX
- DBbrowser(sqlite3)

Login ID:왕벧살

Logout



뜰이

우리집 귀여운 강아지

Uploaded by: 왕벧살

수정

삭제



아무 사진 업로드 중

사이트 테스트 중

Uploaded by: Gusu

Add

```
@bp.route('/add', methods=['POST'])
```

```
def add_card():
```

```
    data = request.form
```

```
    title = data['title']
```

```
    text = data['text']
```

```
    image = request.files['image']
```

```
    username = data['username']
```

```
    if g.user is None:
```

```
        return jsonify({'error': 'Unauthorized'}), 401
```

```
    image_filename = secure_filename(image.filename)
```

```
    image_path = os.path.join(current_app.root_path, 'static/uploads', image_filename)
```

```
    image.save(image_path)
```

```
    print(username)
```

```
    print(type(username))
```

```
    card = Card(
```

```
        user_id=g.user.id,
```

```
        title=title,
```

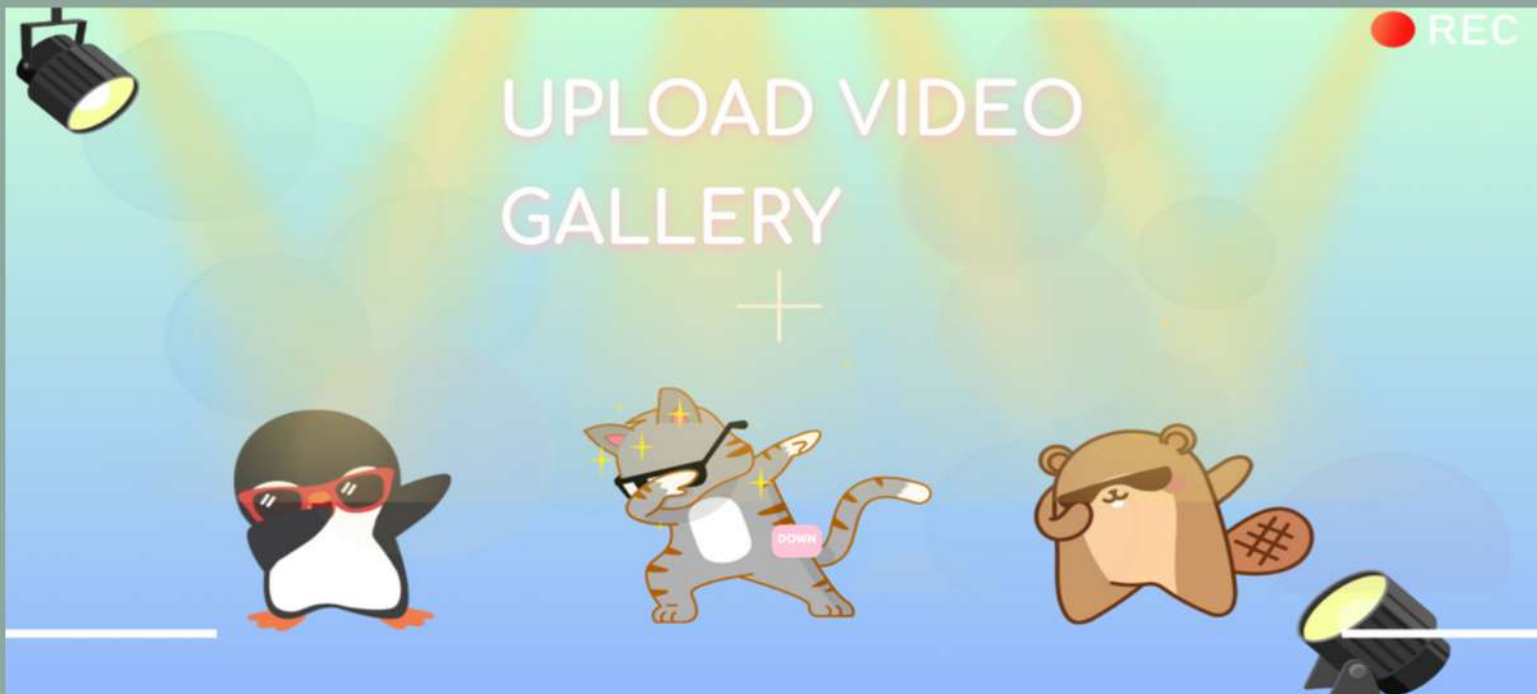
```
        text=text,
```

```
        image_url=f'/static/uploads/{image_filename}', # 수정된 부분
```

```
        username=username
```


각 기능별 코드

- 영상



```
# 동영상 추가 라우트
@bp.route('/add', methods=['POST'])
def add_video():
    data = request.form # 폼 데이터 가져오기
    title = data['title'] # 동영상 제목 가져오기
    content = data['content'] # 동영상 설명 가져오기
    video = request.files['video'] # 동영상 파일 가져오기
    username = data['username']

    if g.user is None: # 사용자가 로그인되어 있지 않으면
        return jsonify({'error': 'Unauthorized'}), 401 # 인증 오류 반환

    video_filename = secure_filename(video.filename) # 안전한 파일 이름 생성
    video_path = os.path.join(current_app.root_path, 'static', 'video_uploads', video_filename) # 파일 경로 생성
    print(f"Video path: {video_path}") # 경로 확인
    video.save(video_path) # 파일 저장

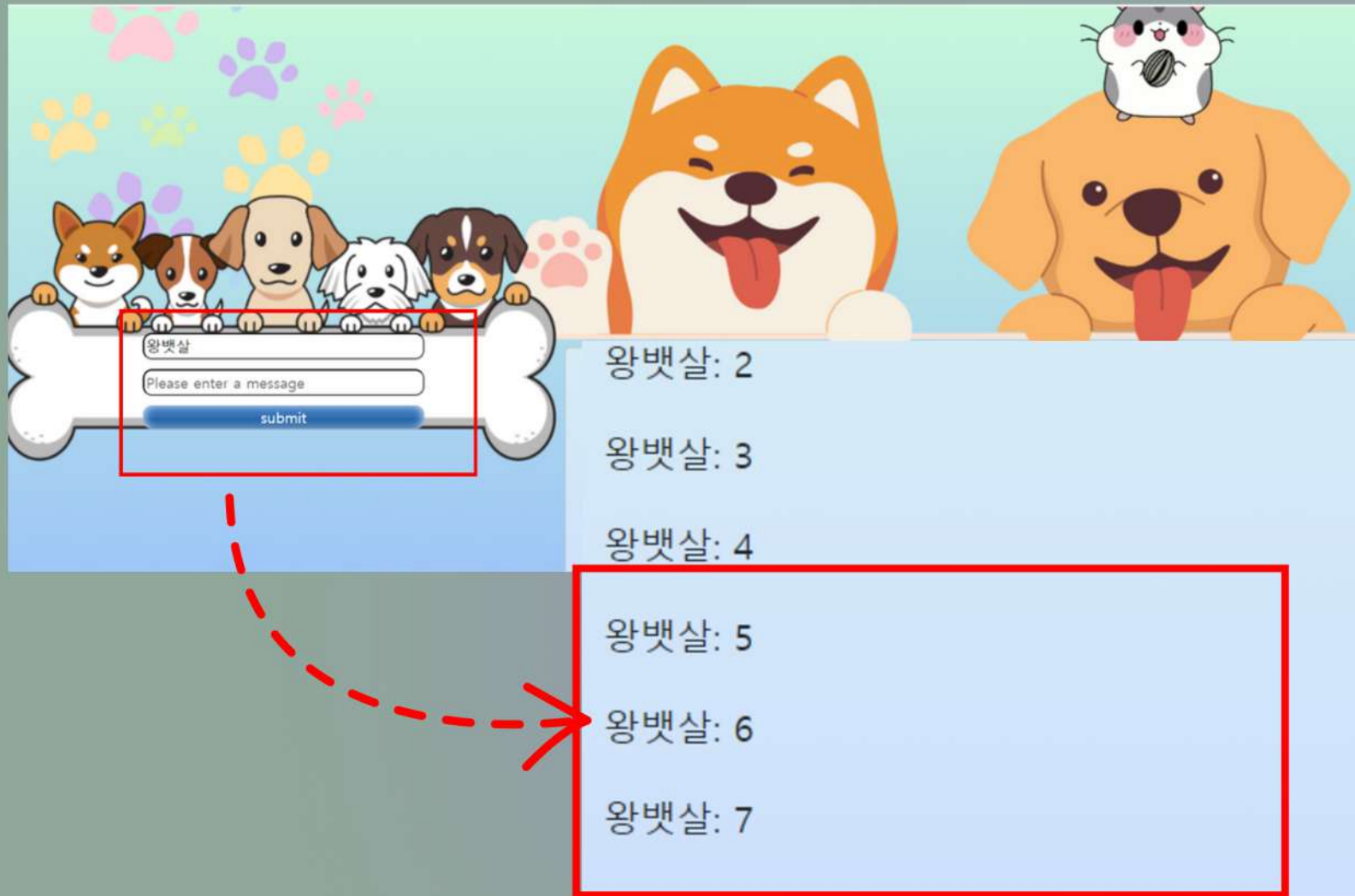
    video_record = Video( # 새로운 동영상 레코드 생성
        user_id=g.user.id, # 사용자 ID 설정
        title=title, # 제목 설정
        content=content, # 설명 설정
        video_url=os.path.join('/static/video_uploads', video_filename), # 동영상 URL 설정
        username=username
    )
    db.session.add(video_record) # 데이터베이스에 추가
    db.session.commit() # 변경사항 커밋

    return jsonify({'success': True, 'video_url': video_record.video_url}) # 성공 응답 반환
```

- HTML, view, CSS
- JavaScript, jQuery , AJAX
- DBbrowser(sqlite3)

각 기능별 코드

- 채팅



- HTML, view, CSS
- JavaScript, socket io
- DBbrowser(sqlite3)

```
@bp.route('/upchat')
def sock():
    return render_template('chating/chat.html')
    """_summary_
    클라이언트(html,js) - 메시지를 입력했음 → 메세지전송 →
    서버(py)쪽에서는 받는 애가 있음 → 서버가 클라이언트쪽으로 메세지를 전송
    클라이언트에서 서버가 전달하는 애를 받아야함
    """

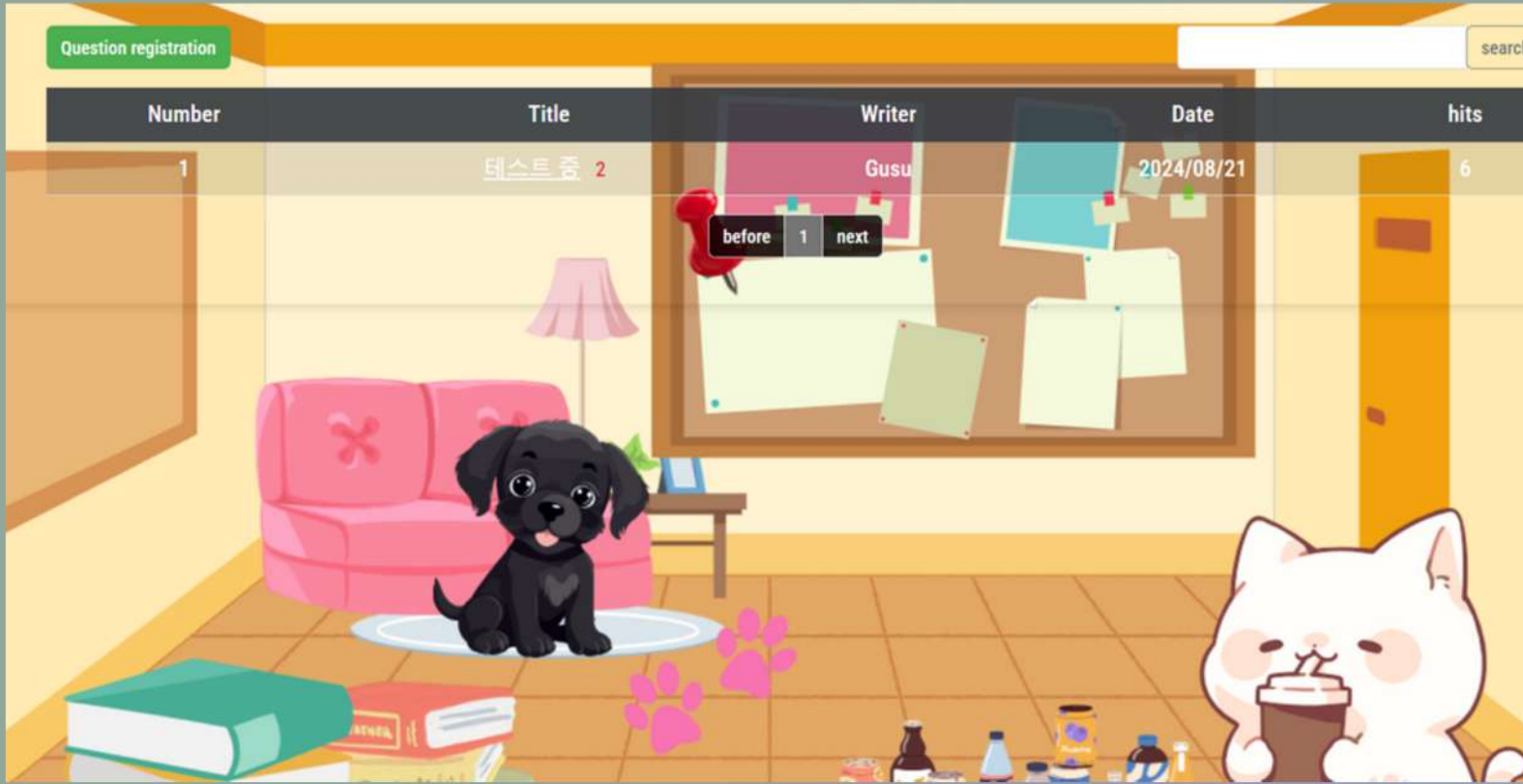
logging.basicConfig(level=logging.INFO)

@socketio.on('message', namespace='/upchat')
def handle_message(msg):
    logging.info(f"Received message: {msg}")
    try:
        # 메시지를 데이터베이스에 저장
        conn = sqlite3.connect('chat.db')
        c = conn.cursor()
        c.execute("INSERT INTO messages (username, message) VALUES (?, ?)", (msg['username'], msg['message']))
        conn.commit()
        conn.close()
        logging.info("Message saved to database")
    except Exception as e:
        logging.error(f"Error saving message to database: {e}")

    try:
        # 모든 클라이언트에 메시지 전송
        send(msg, broadcast=True, namespace='/upchat')
        logging.info(f"Sent message to all clients: {msg}")
    except Exception as e:
        logging.error(f"Error sending message to clients: {e}")
```

각 기능별 코드

- 게시판메인



- HTML, view, CSS
- DBbrowser(sqlite3)

```
<tbody>
{% if question_list %}
{% for question in question_list.items %}
<tr class="text-center">
<td>{{ question_list.total - ((question_list.page-1) * question_list.per_page) - loop.index0 }}</td>
<td class="text-start">
<a href="{{ url_for('question.detail', question_id=question.id) }}">{{ question.subject }}</a>

    {% if question.answer_set|length > 0 %}      You, 5 days ago • commit
    <span class="text-danger small mx-2">{{ question.answer_set|length }}</span>
    {% endif %}
</td>/.text-start
<td>{{ question.user.username}}</td>
<td>{{ question.create_date.strftime('%Y/%m/%d') }}</td>
<td>{{ question.view_count }}</td> <!-- 조회수 표시 -->
</tr>/.text-center
{% endfor %}
{% else %}
<tr>
<td colspan="5">Please write down your questions</td>
</tr>
{% endif %}
```


- 게시판 등록하기



- HTML, view, CSS
- DBbrowser(sqlite3)

```
<div class = 'question'>

<div class="list-form">
  <h5 class="my-3 border-bottom pb-2">Question registration</h5>
  <form method="post" class="post-form my-3">
    {{ form.csrf_token }}
    <!-- 오류표시 Start -->
    {% for field, errors in form.errors.items() %}
    <div class="alert alert-danger" role="alert">
      <strong>{{ form[field].label }}</strong>: {{ ', '.join(errors) }}
    </div>/.alert.alert-danger
    {% endfor %}
    <!-- 오류표시 End -->
    <div class="form-group">
      <label for="subject">TITLE</label>
      <input type="text" class="form-control" name="subject" id="subject"
        value="{{ form.subject.data or '' }}">/#subject.form-control
    </div>/.form-group
    <div class="form-group">
      <label for="content">CONTENT</label>
      <textarea class="form-control" name="content"
        id="content" rows="10">{{ form.content.data or '' }}</textarea>/#content.form-control
    </div>/.form-group
    <button type="submit" class="btn btn-primary">SAVE</button>
  </form>/.post-form.my-3
</div>/.list-form
```

- 게시판답변달기



- HTML, view, CSS
- DBbrowser(sqlite3)

각 기능별 코드

- 게시판 답변달기 코드

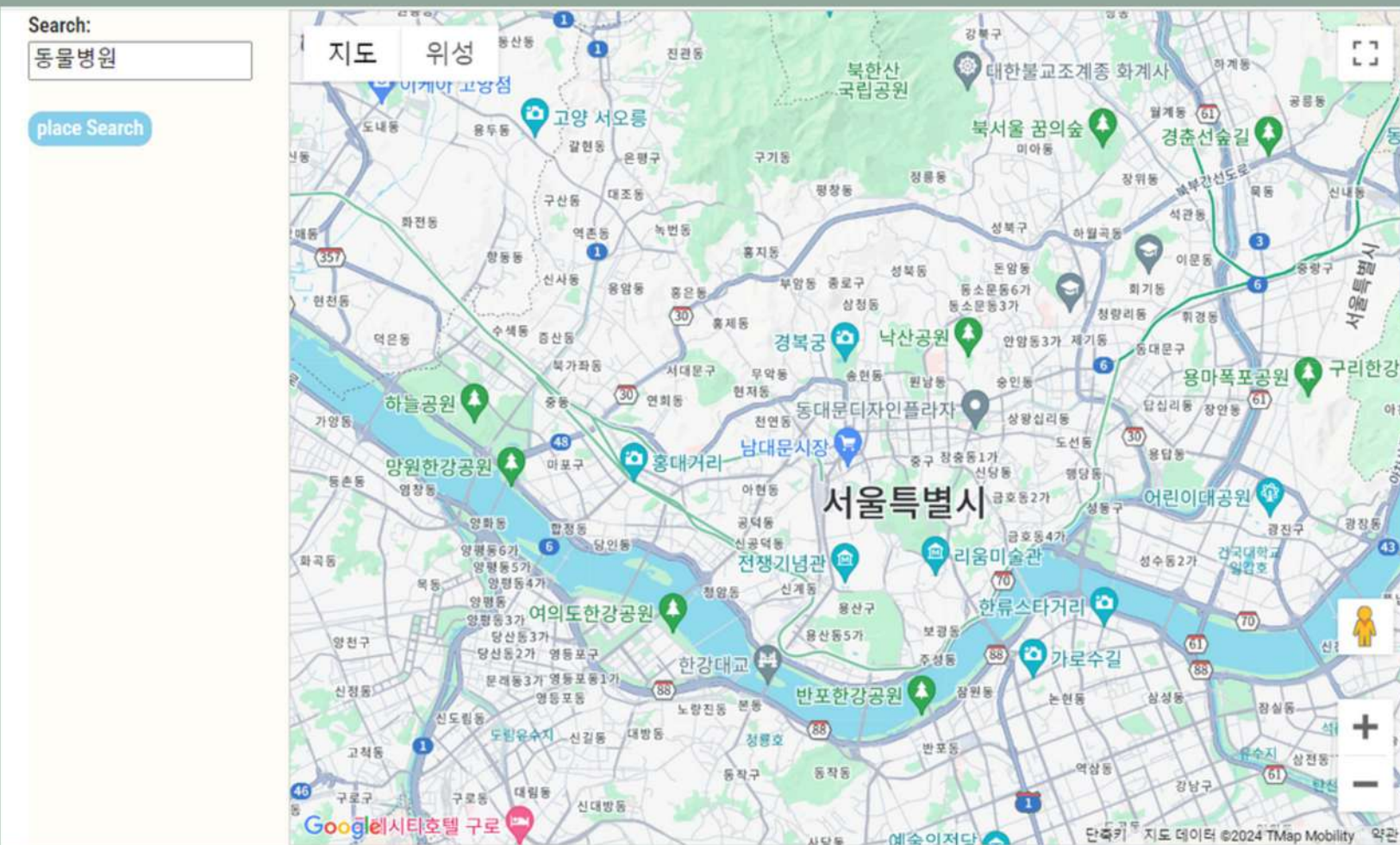
```
<!-- 답변 목록 -->
<h5 class="border-bottom my-3 py-2">{{ question.answer_set|length }}answers here.</h5>
{% for answer in question.answer_set %}
<a id="answer_{{ answer.id }}"></a>
<div class="card my-3">
  <div class="card-body">
    <div class="card-text" style="white-space: pre-line;">{{ answer.content }}</div>
    <div class="d-flex justify-content-end">
      {% if answer.modify_date %}
      <div class="badge bg-light text-dark p-2 text-start mx-3">
        <div class="mb-2">modified at</div>
        <div>{{ answer.modify_date }}</div>
      </div>/.badge.bg-light.text-dark.p-2.text-start.mx-3
      {% endif %}
      You, 5 days ago • commit
      <div class="badge bg-light text-dark p-2 text-start mx-3">
        <div class="mb-2">{{ answer.user.username }}</div>
        <div>{{ answer.create_date }}</div>
      </div>/.badge.bg-light.text-dark.p-2.text-start.mx-3
    </div>/.d-flex.justify-content-end
  </div>/.card-body
</div>/.card.my-3
{% if g.user == answer.user %}
  <a href="{{ url_for('answer.modify', answer_id=answer.id) }}"
  class="btn btn-sm btn-outline-secondary">Modify</a>/.btn.btn-sm.btn-outline-secondary
  <a href="#" class="delete btn btn-sm btn-outline-secondary "
  data-uri="{{ url_for('answer.delete', answer_id=answer.id) }}">Delete</a>/.delete.btn.btn-sm.btn-outli
  {% endif %}
</div>/.my-3
</div>/.card-body
</div>/.card.my-3
```

```
<!-- 답변등록-->
<form action="{{ url_for('answer.create', question_id=question.id) }}" method="post" class="my-3">
  {{ form.csrf_token }}
  {% include "form_errors.html" %}
  <div class="mb-3">
    <label for="content">Contents of the answer</label>
    <textarea {% if not g.user %}disabled{% endif %}
      name="content" id="content" class="form-control" rows="10"></textarea>/#content.for
    </div>/.mb-3
    <input type="submit" value="Reply registration" class="btn btn-primary">
  </form>/.my-3
</div>/.container.my-3
</div>/.detail
{% endblock %}
{% block script %}

<script type='text/javascript'>
const delete_elements = document.getElementsByClassName("delete");
Array.from(delete_elements).forEach(function(element) {
  element.addEventListener('click', function() {
    if(confirm("Are you sure you want to delete it?")) {
      location.href = this.dataset.uri;
    }
  });
});
});
```


각 기능별 코드

- 지도



name	address	number
건국본동물병원	1464-22 Juan-dong, Michuhol-gu, Incheon, South Korea	032-864-0075
루이스24시동물병원	232 Baekbeom-ro, Namdong-gu, Incheon, South Korea	032-710-1237
펫홈종합동물병원	South Korea, 1471-17 남구 인천광역시 KR	032-427-7588
정서동물병원	390-4 간석4동 Namdong-gu, Incheon, South Korea	032-422-5399
인천스카이동물메디컬센터	South Korea, KR 인천광역시 남동구 남동대로799번길 34 지웰시티푸르지오 C동 301호	032-715-7959
24시퍼스트동물의료센터	South Korea, Incheon, Michuhol-gu, Gyeongwon-daero, 890 KR 인천광역시 남구 1층 101호	032-435-6333
애니힐동물병원	South Korea, dong, Inha-r	
부평스카이동물의료센터	351 Buheung South Korea	
24시스카이동물의료센터	586 Inju-dae Korea	
Chansaem Animal Hospital	South Korea, 64 타원벨리	
동암동물병원	South Korea, 570-5	
24시포인트동물	78 Nakseom	

- HTML, view, CSS
- Google API
- JavaScript, AJAX

```
@bp.route('/search')
def search():
    return ...

<h1>Google Maps API 예제</h1>
<form id="searchForm">
    <label for="place">장소 검색:</label>
    <input type="text" id="place" name="place" required><br><br>
    <input type="submit" value="장소 찾기">
</form> #searchForm
<div id="map" style="height: 500px; width: 100%;"></div>
<div id="place-results">
    <table id="results-table" border="1">
        <tr>
            <th>이름</th>
            <th>주소</th>
            <th>전화번호</th>
        </tr>
    </table> #results-table
</div> #place-results
<script>
    function initMap() {
        var map = new google.maps.Map(document.getElementById('map'), {
            zoom: 7,
            center: {lat: 37.5665, lng: 126.9780} // 서울을 초기 중심으로 설정
        });
        window.map = map; // 전역 변수로 설정하여 이후에 접근 가능하도록 함
    }
</script>
```

Thank You