



LLM

web

application

project

SKN5 Term Project 4

LLM web application team project__

SKN5 Term Project 4 - 2팀

최영민 martinus.choi@gmail.com

서장호 sjh008272@gmail.com

김지연 kjiyoun02@gmail.com

조주영 nammin8269@gmail.com



LLM

web

application

project

프로젝트 개요 ; Project outline ____

프로젝트 기능 ; project function ____

기술 스택 ; technology stack ____

시스템 아키텍처 ; system architecture ____

트러블 슈팅 ; trouble shooting ____

팀원 분담 ; Division of team members ____

개발 일정 ; Development Schedule ____



LLM

web

application

project

Project outline ____

프로젝트 명

LLM 활용 웹 어플리케이션 제작

개발 일정

24.12.18 ~ 24.12.19

특징

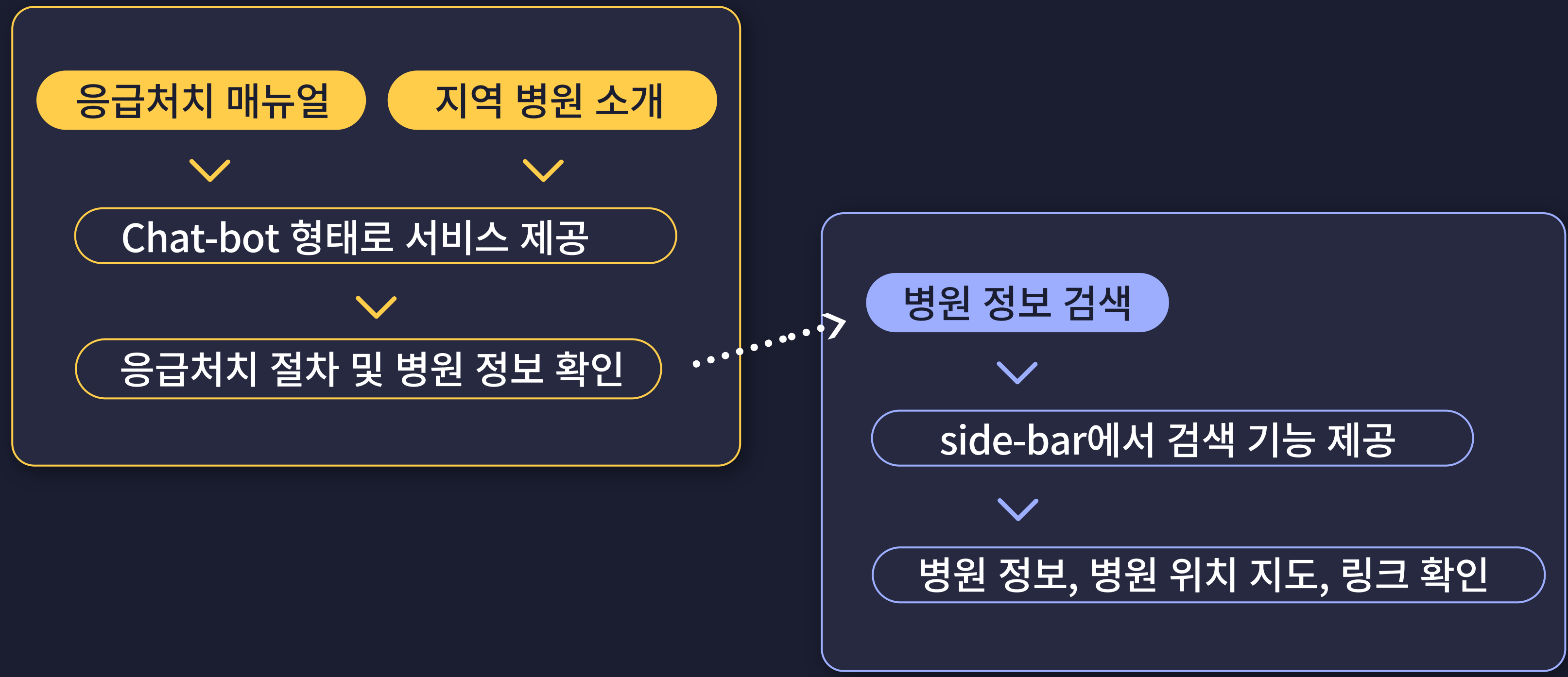
OpenAI API 기반 LLM 모델 파인튜닝

Django 기반 웹 어플리케이션 구현

RAG Chain 구성, Django 웹 서빙

AWS 활용 웹 어플리케이션 배포

project function ____





LLM

web

application

project

technology stack ____



서버

Django

HTML, CSS, JS, BootStrap

SQL-Lite



LLM / RAG Chain

OpenAI API

Langchain

Chroma



배포

AWS

NGINX, GUNICORN

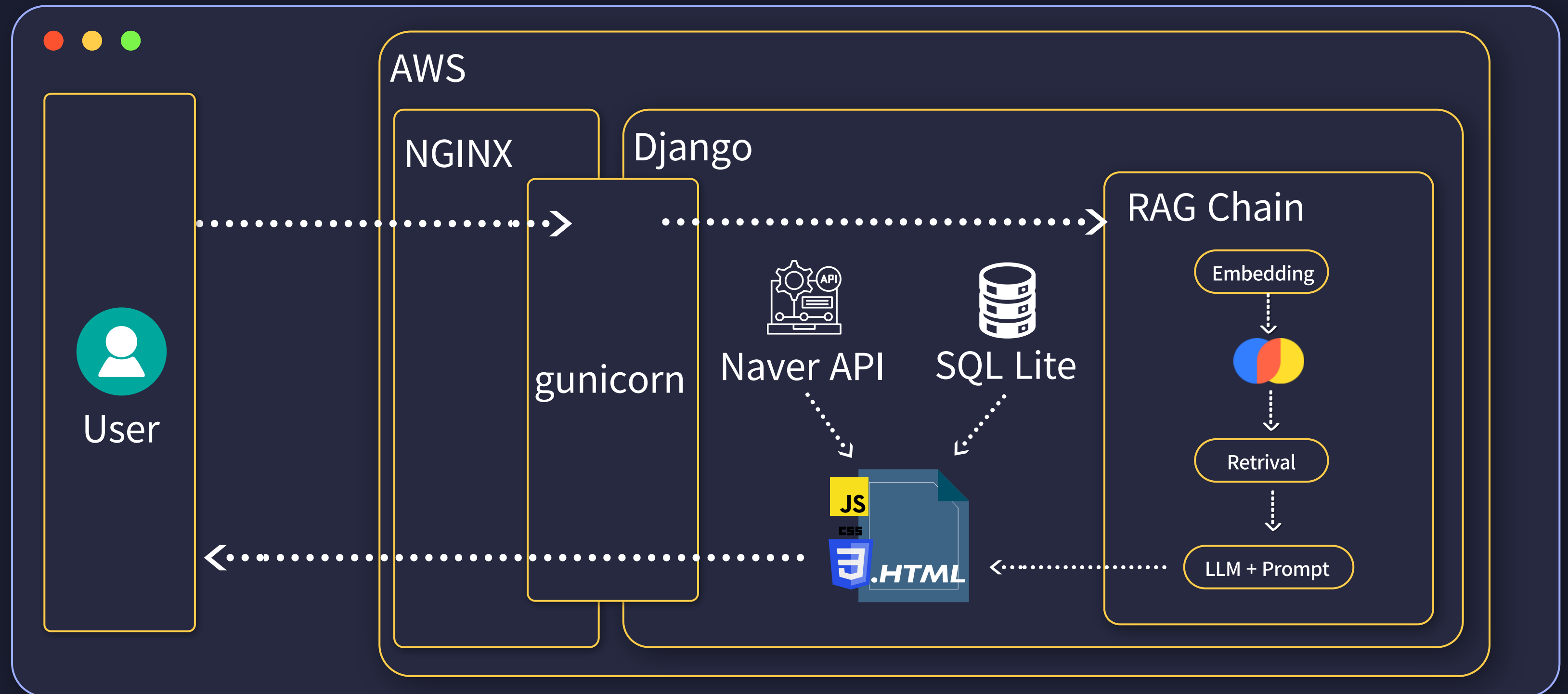


검색 API

Naver OpenAPI

Folium

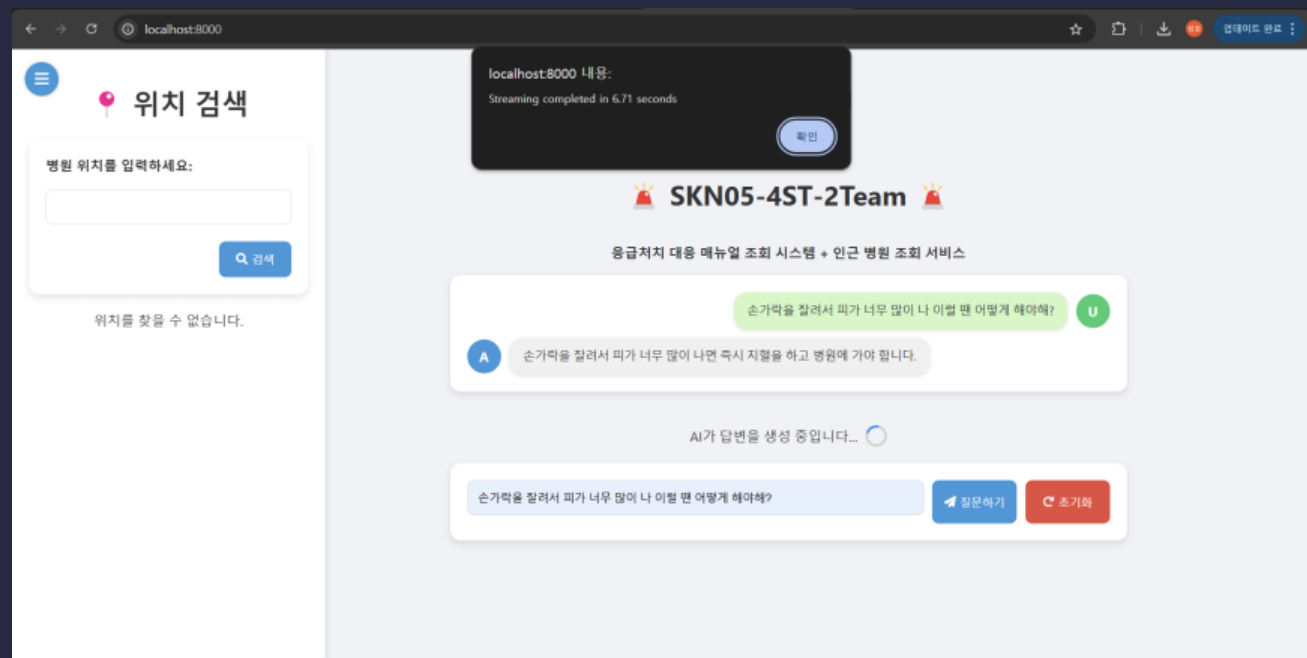
system architecture



trouble shooting ____

Trouble

Chat-bot의 응답 속도가 늦음



응답 완료 까지 6.71초 소요

원인 분석

사용자의 매 Query 마다
RAG Chain 객체 생성

```
if query:
    try:
        print(f"Query type: {type(query)}, Query content: {query}")

        chainer = EmergencyRAGChainer(db_path='./db/chromadb_1')
        chain = chainer.create_rag_chain()

        response = chain.invoke(query)
    except Exception as e:
        return JsonResponse({'error': str(e)})
```

매 응답마다 Chain 구성 시간 소요



LLM

web

application

project

trouble shooting ____

Trouble Shooting

서버 최초 실행 시에만 Chain 생성

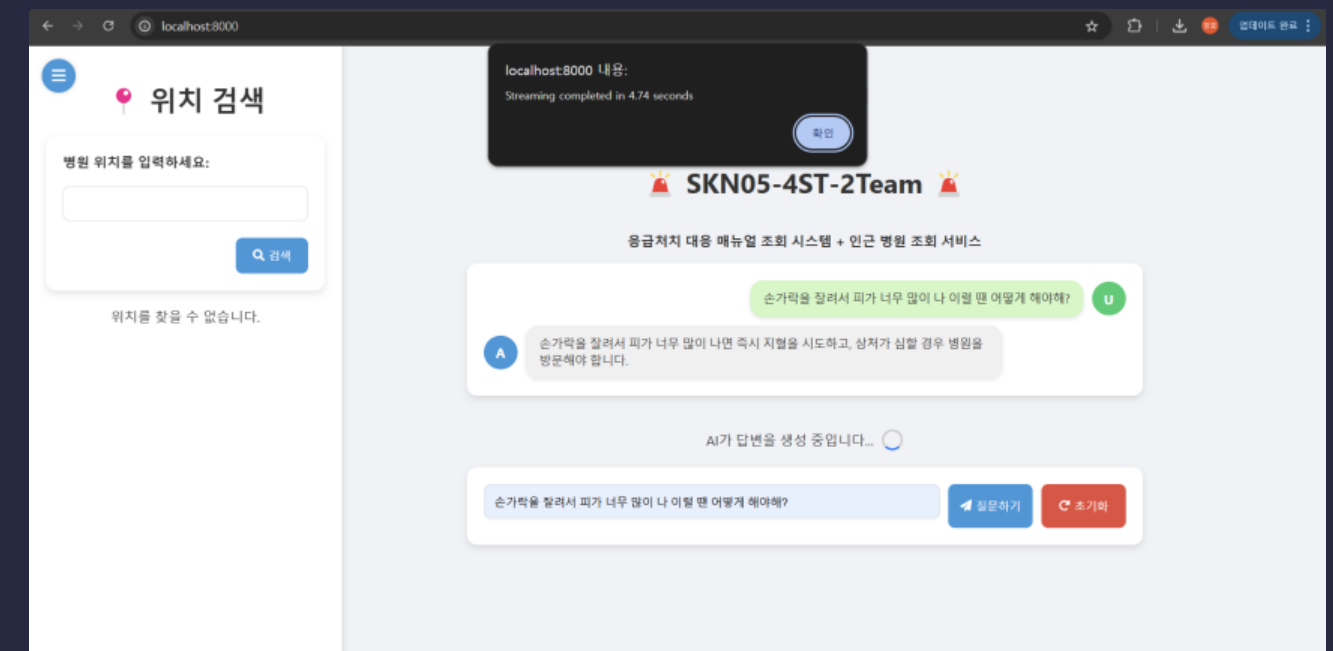
```
chain = get_chain()
print("chain created")

# Main view for the index
def index(request):
    if query:
        try:
            response = chain.invoke(query)
        except Exception as e:
            return JsonResponse({'error': str(e)})
```

index 함수 외부에서 chain 생성

개선 결과

응답 속도 개선 확인



응답 완료 까지 4.74초 소요 (2초 감소)



LLM

web

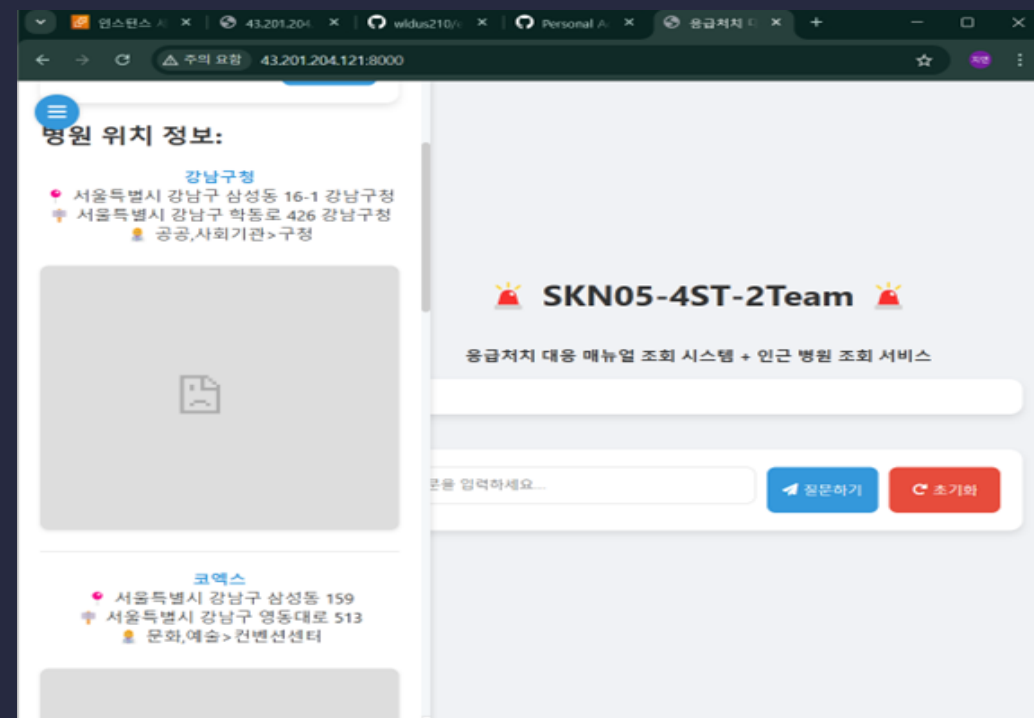
application

project

trouble shooting ____

Trouble

AWS 웹 배포 시 지도 출력 불안정





LLM

web

application

project

trouble shooting ____

원인 분석 1

NGINX :

1. proxy_buffering, proxy_cache 수행 중
2. proxy_read_timeout 기본값 60초 연결 후 끊김

```
server {  
    listen 80;  
    server_name 3.39.36.49;  
  
    location / {  
        proxy_pass http://unix:/home/ubuntu/emergency/emergency_app.sock;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
  
    location /static/ {  
        alias /home/ubuntu/emergency/staticfiles/;  
    }  
  
    location /media/ {  
        alias /home/ubuntu/emergency/media/;  
    }  
}
```

원인 분석 2

gunicorn : Group -> ubuntu로 설정 되어있음

```
[Unit]  
Description=gunicorn daemon for Django  
After=network.target  
  
[Service]  
User=ubuntu  
Group=ubuntu  
WorkingDirectory=/home/ubuntu/emergency  
ExecStart=/home/ubuntu/venv/bin/gunicorn \\\n    --workers 3 \\\n    --bind unix:/home/ubuntu/emergency/emergency_app.sock \\\n    emergency_app.wsgi:application  
  
[Install]  
WantedBy=multi-user.target
```



LLM

web

application

project

trouble shooting ____

Trouble Shooting 1

NGINX :

1. proxy_buffering off, proxy_cache off
: 서버와 클라이언트 간 데이터를 실시간으로 전달
2. proxy_read_timeout 300
: 서버가 느리게 응답해도 연결이 끊기지 않도록 함

```
server {
    listen 80;
    server_name 3.39.36.49;

    location / {
        proxy_pass http://unix:/home/ubuntu/emergency/emergency_app.sock;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_buffering off;
        proxy_cache off;
        proxy_read_timeout 300;
    }

    location /static/ {
        alias /home/ubuntu/emergency/staticfiles/;
    }

    location /media/ {
        alias /home/ubuntu/emergency/media/;
    }
}
```

Trouble Shooting 2

gunicorn :

1. www-data 서비스 그룹 추가 후 설정
: 파일 권한 및 소켓 접근 권한 문제 해결

```
[Unit]
Description=gunicorn daemon for Django
After=network.target

[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/home/ubuntu/emergency
ExecStart=/home/ubuntu/venv/bin/gunicorn \
    --workers 3 \
    --worker-class gevent \
    --bind unix:/home/ubuntu/emergency/emergency_app.sock \
    emergency_app.wsgi:application

[Install]
WantedBy=multi-user.target
```



LLM

web

application

project

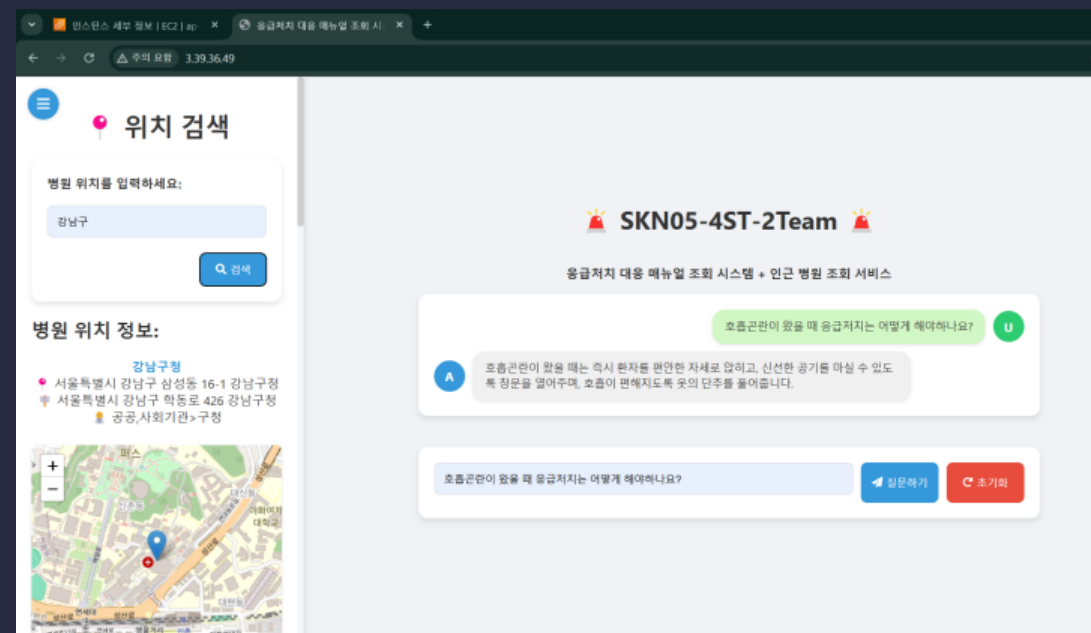
trouble shooting ____

Trouble Shooting 3

권한 설정 :

1. `sudo chmod -R 755 /home/ubuntu/emergency/staticfiles/`
: staticfiles 경로에 있는 파일들에 접근 할 수 있도록 권한을 부여
2. `sudo chown -R ubuntu:www-data /home/ubuntu/emergency/staticfiles/`
: 그룹을 www-data로 설정 ➡ 정적 파일 접근에 필요한 그룹 권한을 부여

개선 결과





LLM

web

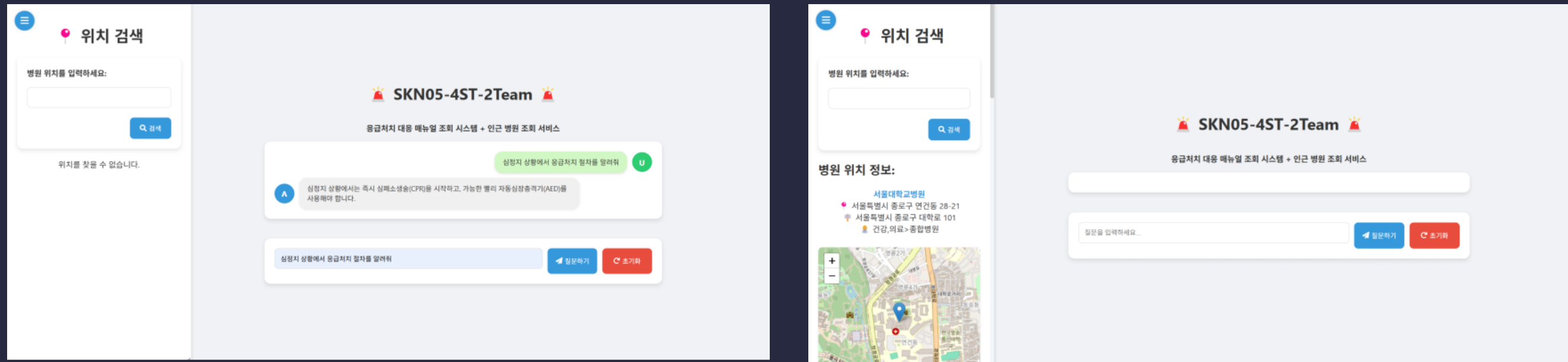
application

project

trouble shooting ____

Trouble

채팅 중 side-bar에서 검색 실시할 때 전체 페이지가 새롭게 렌더링
→ 채팅 내역 사라짐





LLM

web

application

project

trouble shooting ____

원인 분석

위치 검색 폼이 일반적인 HTML 폼 제출 방식으로 처리,
: 폼이 제출될 때 서버로 요청이 전송되고, 서버는 전체 페이지를 다시 렌더링하여 응답
: 이 과정에서 페이지의 모든 상태가 초기화, 진행 중이던 채팅 내역 삭제



Side-bar 만 렌더링 실시하도록 수정 필요



LLM

web

application

project

trouble shooting ____

Trouble Shooting 1

Scripts.js 업데이트 :

1. AJAX를 사용하여 검색 요청을 처리
: 페이지 전체가 새로고침되지 않고 사이드바만 업데이트
: 진행 중인 채팅 내역이 사라지지 않고 유지

```
document.getElementById('location-form').addEventListener('submit', function(event) {
  event.preventDefault();
  const formData = new FormData(this);
  fetch('', {
    method: 'POST',
    body: formData,
    headers: {
      'X-Requested-With': 'XMLHttpRequest'
    }
  })
  .then(response => response.text())
  .then(html => {
    document.getElementById('location-results').innerHTML = html;
  })
  .catch(error => console.error('Error:', error));
});
```

Trouble Shooting 2

views.py 업데이트 :

1. 요청이 AJAX 요청인지 확인
: 사이드바에 필요한 HTML만 반환

```
def index(request):
    if 'chat_history' not in request.session:
        request.session['chat_history'] = []

    location_data = {'items': []}
    map_html_paths = []

    if request.method == 'POST':
        if 'location_query' in request.POST:
            location_query = request.POST.get('location_query', '')
            location_data = search_location(location_query)
            map_html_paths = generate_folium_map(location_data)

            if request.headers.get('x-requested-with') == 'XMLHttpRequest': # AJAX 요청인 경우
                html = render_to_string('partials/location_results.html', { # 부분 템플릿 렌더링
                    'location_data': location_data, # 위치 데이터
                    'map_html_paths': map_html_paths, # 지도 데이터
                })
            return HttpResponse(html) # 응답 반환
```




LLM

web

application

project

trouble shooting ____

Trouble Shooting 3

부분 렌더링 html 생성 :

1. 부분 템플릿을 새로 생성

: 위치 검색 결과가 이 부분 템플릿으로 렌더링되며, AJAX 응답으로 반환

```
{% if location_data.items %} <!-- 위치 데이터가 있는 경우 -->
{% load custom_tags %} <!-- 커스텀 태그 로드 -->
<h2>병원 위치 정보:</h2>
{% for item, map_html_path in location_data.items|zip_lists:map_html_paths %} <!-- 위치 데이터와 지도 데이터를 함께 반복 -->
  <div class="response">
    <p>
      <a href="{{ item.link }}" target="_blank"><b>{{ item.title|safe }}</b></a><br>
      📍 {{ item.address }}<br>
      📍 {{ item.roadAddress }}<br>
      🏠 {{ item.category }}
    </p>
    {% if map_html_path %}
      <div class="map-container">
        <iframe src="{{ map_html_path }}"></iframe>
      </div>
    {% else %}
      <p>지도를 생성할 수 없습니다.</p>
    {% endif %}
  </div>
{% endfor %}
{% else %}
  <p>위치를 찾을 수 없습니다.</p>
{% endif %}
```




LLM

web

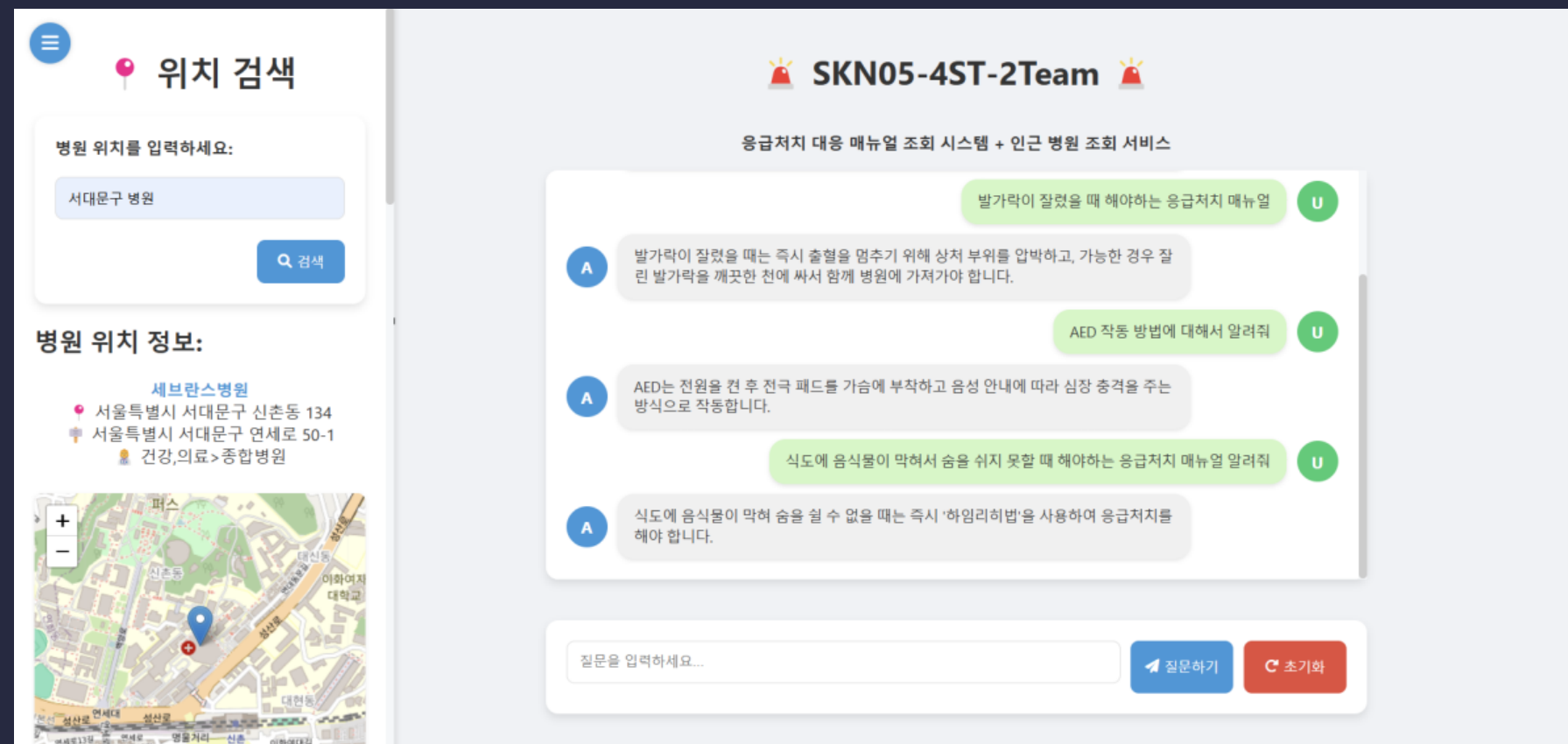
application

project

trouble shooting ____

개선 결과

채팅 내역 유지 확인



Division of team members ____

RAG-Chain, Back-end

최영민

- └ RAG-Chain 웹 서빙
- └ Django 웹 서빙 최적화
- └ 시스템 점검 / 개선

Front-end, Back-end

서장호

- └ Chat-bot 화면
- └ Side-bar 검색 화면
- └ Django 웹 어플리케이션 구현

LLM, Front-end, Web Publish

김지연

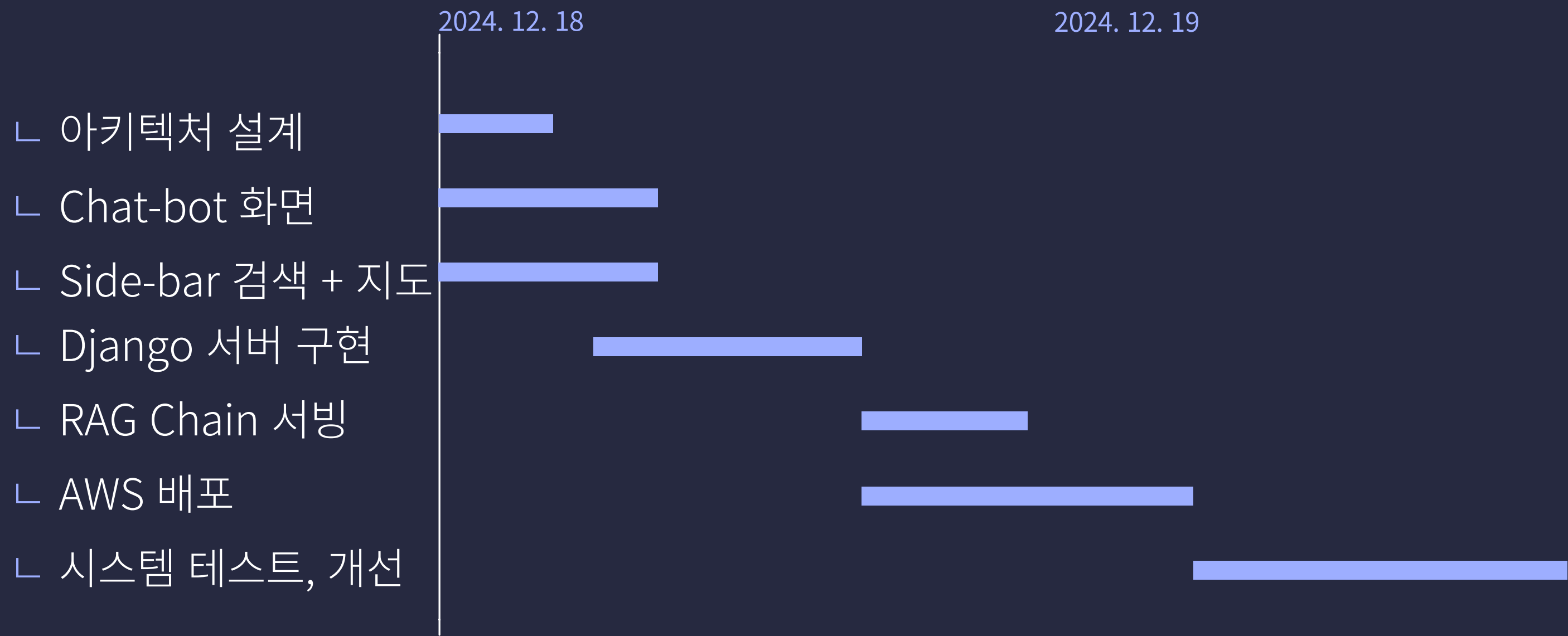
- └ Django 웹 어플리케이션 배포
- └ AWS 배포, 지도 API 안정화
- └ 파인튜닝 LLM 모델 웹 서빙

Front-end

조주영

- └ Chat-bot 화면

Development Schedule



9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



LLM

web

application

project

자유롭게 질문해주세요!