

관광지 추천 옵션 조정



```
# streamlit_app.py

import streamlit as st
from dotenv import load_dotenv
import os
from langchain_openai import OpenAI

# 환경 변수 로드
load_dotenv()
OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")

# LangChain LLM 초기화
llm = OpenAI()

# 페이지 설정
st.set_page_config(
    page_title="한국 관광지 추천",
    page_icon="🌐",
    layout="centered",
    initial_sidebar_state="expanded"
)

# --- 사이드바 ---
with st.sidebar:
    st.markdown("## 설정")
    st.markdown("관광지 추천 옵션을 선택하세요.")

    num_places = st.selectbox("추천 받을 관광지 수", [3, 5], index=0)
    user_prompt = st.text_area(
        "질문 내용",
        f"한국의 대표적인 관광지 {num_places}군데를 추천해주세요.",
        height=100
    )
    recommend_btn = st.button("관광지 추천받기")

# --- 메인 화면 ---
st.title("🇰🇷 한국 관광지 추천 시스템")
st.markdown(
    """
    여행을 준비 중이신가요?
    이 앱은 OpenAI의 LLM을 기반으로, 한국에서 꼭 가봐야 할 관광지를 추천해드립니다.
    """
)
```

```
st.markdown("---")

# 결과 처리
if recommend_btn:
    with st.spinner("추천 중입니다. 잠시만 기다려주세요..."):
        try:
            response = llm.invoke(user_prompt)
            st.success("추천이 완료되었습니다!")
            st.markdown("### 📌 추천 결과")
            st.text_area("관광지 리스트", response, height=250)
        except Exception as e:
            st.error(f"오류가 발생했습니다: {e}")

# 푸터
st.markdown("---")
st.caption("© 2025 Korea Travel AI | Powered by LangChain & OpenAI")

위 코드에서 옵션을 조절할 수 있도록 해줘.
```

당신의 `streamlit_app.py`에서 **관광지 추천 옵션을 조절할 수 있도록** 확장하려면, 사이드바에 **옵션 필드**를 추가하고 이를 프롬프트에 반영하면 됩니다. 아래는 다음과 같은 옵션을 조절할 수 있도록 개선한 버전입니다:

✓ 추가된 기능

- **관광지 수 선택**
- **여행 테마 선택** (예: 자연, 역사, 음식 등)
- **여행 지역 입력** (예: 제주도, 서울 등)

✓ 개선된 코드

```
python

# streamlit_app.py

import streamlit as st
from dotenv import load_dotenv
import os
from langchain_openai import OpenAI

# 환경 변수 로드
load_dotenv()
OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")

# LangChain LLM 초기화
llm = OpenAI()

# 페이지 설정
st.set_page_config(
    page_title="한국 관광지 추천",
    page_icon="🇰🇷",
    layout="centered",
    initial_sidebar_state="expanded"
)
```

```
# --- 사이드바 ---
with st.sidebar:
    st.markdown("## 설정")
    st.markdown("관광지 추천 옵션을 선택하세요.")

    num_places = st.selectbox("추천 받을 관광지 수", [3, 5, 7, 10], index=0)

    theme = st.selectbox(
        "여행 테마 선택",
        ["전체", "자연", "역사/문화", "음식", "액티비티", "가족 여행"],
        index=0
    )

    region = st.text_input("여행 지역 (선택사항)", placeholder="예: 서울, 제주도")

    user_prompt = st.text_area(
        "질문 내용",
        f"한국의 대표적인 관광지 {num_places}군데를 추천해주세요.",
        height=100
    )

    recommend_btn = st.button("관광지 추천받기")

# --- 메인 화면 ---
st.title("🇰🇷 한국 관광지 추천 시스템")
st.markdown(
    """
    여행을 준비 중이신가요?
    이 앱은 OpenAI의 LLM을 기반으로, 한국에서 꼭 가봐야 할 관광지를 추천해드립니다.
    """
)
st.markdown("---")

# 결과 처리
if recommend_btn:
    with st.spinner("추천 중입니다. 잠시만 기다려주세요..."):
        try:
            # 사용자 옵션에 따라 프롬프트 생성
            final_prompt = f"{region + ' 지역에서' if region else ''} "
            final_prompt += f"{theme if theme != '전체' else ''} 테마로 "
            final_prompt += f"{num_places}군데의 한국 관광지를 추천해줘. "
            final_prompt += f"요청 내용: {user_prompt}"

            response = llm.invoke(final_prompt)
            st.success("추천이 완료되었습니다!")
            st.markdown("### 📌 추천 결과")
            st.text_area("관광지 리스트", response, height=250)
        except Exception as e:
            st.error(f"오류가 발생했습니다: {e}")

# 푸터
st.markdown("---")
st.caption("© 2025 Korea Travel AI | Powered by LangChain & OpenAI")
```

💡 추가 아이디어 (필요 시 확장)

- 날짜 선택 기능 (여행 시기 기반 추천)
- 가족/커플/혼자 등의 여행 유형 선택
- 필터링된 결과를 지도에 표시 (향후 folium 또는 streamlit-folium 활용)

필요 시 그중 일부도 바로 적용해드릴 수 있어요.



```
(venv)
사전교육@DESKTOP-FIOHMRJ MINGW64 ~/Desktop/pre_training/langchian_test (main)
$ streamlit run streamlit_app.py
bash: streamlit: command not found
```

bash: streamlit: command not found 오류는 현재 가상환경(venv)에 **streamlit** 패키지가 설치되어 있지 않거나, 실행 경로(PATH)에 제대로 연결되지 않았을 때 발생합니다.

✓ 해결 방법

1. 가상환경에 streamlit 설치

먼저, 아래 명령어로 **streamlit**을 설치하세요:

```
bash

pip install streamlit
```

설치 후 다시 실행:

```
bash

streamlit run streamlit_app.py
```

2. 설치 확인

정상 설치 여부를 확인하려면 아래 명령어를 입력하세요:

```
bash

which streamlit
```

Windows에서 가상환경이 제대로 활성화돼 있다면 보통 다음과 같은 경로가 나옵니다:

```
swift

/c/Users/사전교육/Desktop/pre_training/langchian_test/venv/Scripts/streamlit
```

3. (권장) 실행 명령에 경로 명시

간혹 bash(MINGW64)에서 직접 **streamlit** 명령이 안 될 수 있습니다. 이 경우 전체 경로로 실행할 수도 있습니다:

```
bash

./venv/Scripts/streamlit run streamlit_app.py
```

📌 참고: 가상환경 활성화가 안 되어 있다면?

가상환경을 다시 활성화하세요 (Windows 기준):

```
bash

source venv/Scripts/activate
```

그리고 다시 실행하세요:

```
bash

streamlit run streamlit_app.py
```

필요하시면 `requirements.txt`도 함께 만들어 드릴게요.



temperature같은 인자를 조절할 수 있게 해줘.

좋습니다! **temperature**는 LLM의 **창의성/무작위성**을 조절하는 중요한 인자입니다. 이를 사용자에게 조절할 수 있게 하려면 다음을 수행하면 됩니다:

✅ 변경 요약

- 사이드바에 **temperature** 슬라이더 추가 (0.0 ~ 1.0)
- OpenAI 인스턴스를 동적으로 생성 (temperature 적용 위해)
- 코드 구조 일부 개선

🧠 설명

temperature를 `OpenAI()` 클래스 생성 시에 지정해야 하므로, 버튼 클릭 시마다 새로 `llm = OpenAI(temperature=...)`로 초기화합니다.

✅ 최종 코드

```
python

# streamlit_app.py

import streamlit as st
from dotenv import load_dotenv
import os
from langchain_openai import OpenAI

# 환경 변수 로드
load_dotenv()
OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")

# 페이지 설정
st.set_page_config(
    page_title="한국 관광지 추천",
    page_icon="🌐",
    layout="centered",
    initial_sidebar_state="expanded"
)

# --- 사이드바 ---
with st.sidebar:
    st.markdown("## 설정")
    st.markdown("관광지 추천 옵션을 선택하세요.")

    num_places = st.selectbox("추천 받을 관광지 수", [3, 5, 7, 10], index=0)

    theme = st.selectbox(
        "여행 테마 선택",
        ["전체", "자연", "역사/문화", "음식", "액티비티", "가족 여행"],
        index=0
    )

    region = st.text_input("여행 지역 (선택사항)", placeholder="예: 서울, 제주도")

    temperature = st.slider(
        "LLM 창의성 (Temperature)",
        min_value=0.0,
        max_value=1.0,
        value=0.7,
        step=0.1,
        help="값이 클수록 다양하고 창의적인 응답을 생성합니다."
    )

    user_prompt = st.text_area(
        "질문 내용",
        f"한국의 대표적인 관광지 {num_places}군데를 추천해주세요.",
        height=100
    )

    recommend_btn = st.button("관광지 추천받기")

# --- 메인 화면 ---
st.title("🇰🇷 한국 관광지 추천 시스템")
st.markdown(
    """
    여행을 준비 중이신가요?
    이 앱은 OpenAI의 LLM을 기반으로, 한국에서 꼭 가봐야 할 관광지를 추천해드립니다.
    """
)
st.markdown("---")

# 결과 처리
if recommend_btn:
    with st.spinner("추천 중입니다. 잠시만 기다려주세요..."):
        try:
            # temperature 반영된 LLM 초기화
```

```

llm = OpenAI(temperature=temperature)

# 사용자 옵션에 따라 프롬프트 생성
final_prompt = f"{region + ' 지역에서' if region else ''} "
final_prompt += f"{theme if theme != '전체' else ''} 테마로 "
final_prompt += f"{num_places}군데의 한국 관광지를 추천해줘. "
final_prompt += f"요청 내용: {user_prompt}"

response = llm.invoke(final_prompt)
st.success("추천이 완료되었습니다!")
st.markdown("### 📌 추천 결과")
st.text_area("관광지 리스트", response, height=250)
except Exception as e:
    st.error(f"오류가 발생했습니다: {e}")

# 푸터
st.markdown("---")
st.caption("© 2025 Korea Travel AI | Powered by LangChain & OpenAI")

```

🎯 결과

- 사용자는 슬라이더를 통해 LLM의 창의성 정도를 직접 조절할 수 있습니다.
- `temperature = 0.0` → 매우 보수적, 정적인 추천
- `temperature = 1.0` → 다양하고 창의적인 추천

필요하시면 `top_p`, `max_tokens` 같은 인자도 조절할 수 있게 확장해드릴 수 있어요.