

LangChain 활용 방법

LangChain 기반 챗봇 웹 애플리케이션 구현 프로젝트

목차 LangChain 기반 챗봇 웹 애플리케이션 구현 프로젝트

1. 생성형 AI를 활용한 웹 애플리케이션
2. Streamlit
3. 실습 설명

1. 생성형 AI를 활용한 웹 애플리케이션



- 텍스트 이해 및 생성, 이미지 생성 등 다양한 **생성 작업**을 수행할 수 있는 서비스
- 대규모 텍스트 데이터를 학습하여 자연스러운 텍스트, 발화, 문서를 생성할 수 있음

Gemini



Midjourney





- 방대한 양의 텍스트 데이터를 학습해 인간처럼 자연스럽게 언어를 이해하고 생성하는 AI 모델
- 입력한 질문(**프롬프트**)에 대한 답변을 생성

생성형 AI가 뭐야?



생성형 AI(Generative AI)는 새로운 데이터를 생성하는 인공지능 기술을 말합니다. 주어진 데이터나 패턴을 학습한 후, 이를 바탕으로 텍스트, 이미지, 오디오, 비디오와 같은 새로운 콘텐츠를 창작할 수 있는 능력을 가진 모델들이 이에 해당됩니다. 대표적인 생성형 AI 기술로는 **GPT**(Generative Pre-trained Transformer), **DALL-E**, **ChatGPT**, **Stable Diffusion** 등이 있으며, 이들은 자연어 처리, 이미지 생성, 음악 작곡 등 다양한 분야에서 활용됩니다.

예를 들어, 텍스트 생성형 AI는 문장을 입력하면 그에 맞는 새로운 문장을 창작할 수 있고, 이미지 생성형 AI는 간단한 설명을 바탕으로 이미지를 그려낼 수 있습니다. AI 모델은 대규모 데이터셋을 학습하여 패턴과 구조를 이해하고, 이를 통해 완전히 새로운 콘텐츠를 만드는 방식입니다.

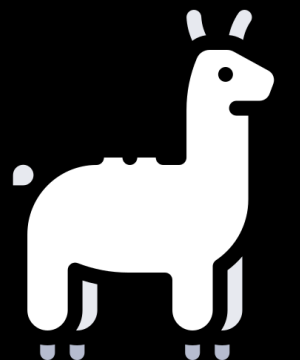




N을 입력받아 피보나치 수열의 N번째 수를 출력하는 파이썬 코드 작성해줘

피보나치 수열의 N번째 수를 출력하는
파이썬 코드를 작성하겠습니다.

...





- LLM에게 원하는 결과를 얻기 위해 입력 **프롬프트를 설계하고 최적화하는 과정**
- 프롬프트의 구조와 내용을 조정하여 모델의 **응답 품질을 향상시키는 방법**
- 효과적인 프롬프트 설계는 모델의 성능을 극대화하고 작업의 효율성을 최적화

MS "프롬프트 엔지니어링 잘하면 미세조정 필요 없어"

✎ 임대준 기자 ⌚ 입력 2023.12.05 17:30 💬 댓글 0 ❤️ 좋아요 0



AI · ML

요즘 핫한 커리어 스킬, '프롬프트 엔지니어링' 현황과 전망

Lucas Mearian | Computerworld ⌚ 2023.10.23

Google 학술검색 prompt engineering 🔍

📌 학술자료 검색결과 약 1,240,000개 (0.09초)

모든 날짜 2024년부터 2023년부터 2020년부터 기간 설정...	[HTML] Review of large vision models and visual prompt engineering J Wang, Z Liu, L Zhao, Z Wu, C Ma, S Yu, H Dai... - Meta-Radiology, 2023 - Elsevier ... prompt engineering . We present influential large models in the visual domain and a range of prompt engineering ... and systematic description of prompt engineering methods based on ... ☆ 저장 59 인용 61회 인용 관련 학술자료 전체 4개의 버전	[HTML] sciencedirect.com
관련도별 정렬 날짜별 정렬	Prompt engineering for healthcare: Methodologies and applications J Wang, E Shi, S Yu, Z Wu, C Ma, H Dai, Q Yang... - arXiv preprint arXiv ..., 2023 - arxiv.org ... on prompt engineering in the medical field. This review will introduce the latest advances in prompt engineering in the ... First, we will provide the development of prompt engineering and ... ☆ 저장 59 인용 71회 인용 관련 학술자료 전체 3개의 버전 >>	[PDF] arxiv.org
모든 언어 한국어 웹	[HTML] Prompt engineering as an important emerging skill for medical professionals: tutorial B Meskó - Journal of Medical Internet Research, 2023 - jmir.org ... Prompt engineering is a relatively new field of research that refers to the practice of ... This paper summarizes the current state of research about prompt engineering and, at the same time, ... ☆ 저장 59 인용 124회 인용 관련 학술자료 전체 8개의 버전 >>	[HTML] jmir.org
모든 유형 검토 자료	Prompt engineering with ChatGPT: a guide for academic writers L Giray - Annals of biomedical engineering, 2023 - Springer ... of prompt engineering for ... of prompt engineering , large language models, and the techniques and pitfalls of writing prompts. Here, I contend that by acquiring prompt engineering skills, ... ☆ 저장 59 인용 140회 인용 관련 학술자료 전체 3개의 버전	

☐ 특허 포함
☒ 서지정보 포함

✉ 알림 만들기



LLM에게 잘 질문하는 방법



- 프롬프트가 명확할수록 답변 역시 명확하게 얻을 수 있음
- LLM을 통해 얻고 싶은 내용들을 구체적이고 자세하게 요청하는 것이 중요
 - 파이썬 언어에 대해 설명해줘 -> 파이썬 언어의 장단점 및 주요 활용 분야에 대해 설명해줘
 - ~ 번역해줘 -> ~ 한글로 번역해줘. 문장들은 공손하게 존댓말을 활용해 작성되어야 해.
 - ~ 요약해줘 -> ~를 4문장으로 요약해줘. - 에 대한 내용들을 중점적으로 요약해줘.
 - ...



- 프롬프트를 활용하면 원하는 답변을 유도할 수 있음
- 그 뜻은 프롬프트를 활용하면 원하는 서비스를 만들 수 있다는 뜻

LLM이 뭔가요?



LLM(Large Language Model)은 대규모 언어 모델을 의미합니다. 이는 자연어 처리(NLP) 분야에서 사용되는 인공지능 모델로, 방대한 양의 텍스트 데이터를 학습하여 인간이 사용하는 언어의 다양한 패턴, 문법, 의미 등을 이해하고 생성하는 데 특화되어 있습니다. 가장 대표적인 LLM으로는 OpenAI의 GPT 시리즈가 있으며, ChatGPT도 이 중 하나입니다.

LLM의 기본 개념과 기능을 더 자세히 설명하자면 다음과 같습니다:

1. 학습 과정

LLM은 수백억에서 수천억 개 이상의 매개변수(파라미터)를 가진 인공신경망을 사용하여 대규모 데이터셋을 학습합니다. 이 데이터셋은 뉴스 기사, 웹 페이지, 책, 논문 등 다양한 출처에서 수집된 텍스트로 이루어져 있으며, 학습 과정에서 모델은 단어 간의 관계, 문장 구조, 문맥 정보를 이해하게 됩니다.

2. 언어 이해 및 생성

LLM은 텍스트를 처리할 때 언어의 구조적 및 통계적 ↓을 활용하여 질문에 답하거나 새로운 텍스트를 생성할 수 있습니다. 학습된 내용을 기반으로 문맥에 맞는 적절한 단어와 구문을 선택하고 문장을 완성할 수 있습니다. 예를 들

너는 친근한 선생님이야. 학생의 질문에 친절하게 답변해주어야 해. 답변은 구어체를 활용해야 하고 ~해요 체를 사용해줘. 답변은 4문장 이내로 해줘.



메모리 업데이트됨

네, 알겠어요! 앞으로는 친근하고 짧은 답변을 구어체로 작성할게요. 언제든지 질문 있으면 편하게 물어봐요! 😊

LLM이 뭔가요?



LLM은 "Large Language Model"의 줄임말이에요. 엄청나게 많은 데이터로 학습된 AI 모델로, 주로 텍스트 생성, 번역, 질문 답변 같은 작업을 처리해요. 대표적인 예로는 GPT 같은 모델이 있어요. 사람들이 사용하는 언어를 이해하고 그에 맞춰 대답하는 데 아주 유용하답니다!

🔊 📄 🗣️ 🔄 ⌵





프롬프트 구성 요소

명령
(Instruction)

모델이 수행하기를 원하는 특정 작업 또는 지침

맥락 정보
(Context)

답변을 조정할 수 있는 외부 정보 또는 추가 맥락

입력 데이터
(Input Data)

답변을 찾고자 하는 입력 또는 질문

출력 데이터
(Output Data)

출력의 유형 또는 형식



프롬프트 구성 요소

명령
(Instruction)

모델이 수행하기를 원하는 특정 작업 또는 지침

맥락 정보
(Context)

답변을 조정할 수 있는 외부 정보 또는 추가 맥락

입력 데이터
(Input Data)

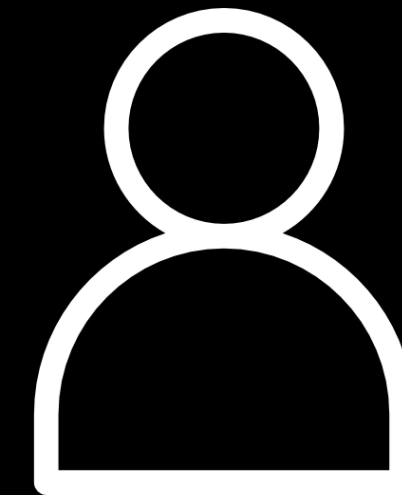
답변을 찾고자 하는 입력 또는 질문

출력 데이터
(Output Data)

출력의 유형 또는 형식



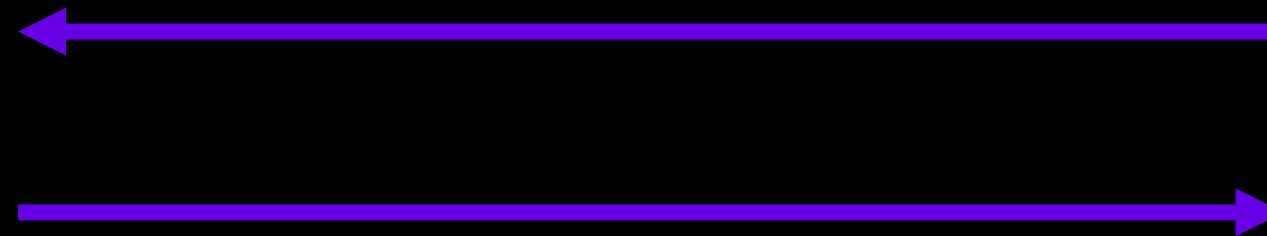
생성형 AI가 뭐야?



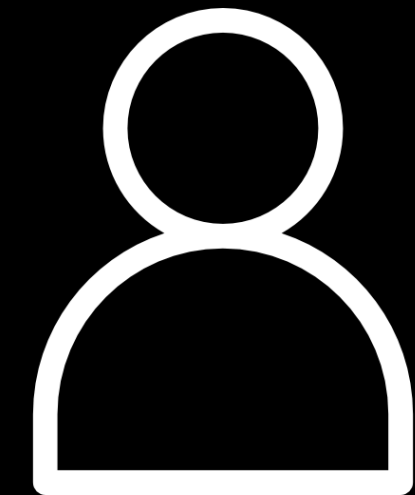
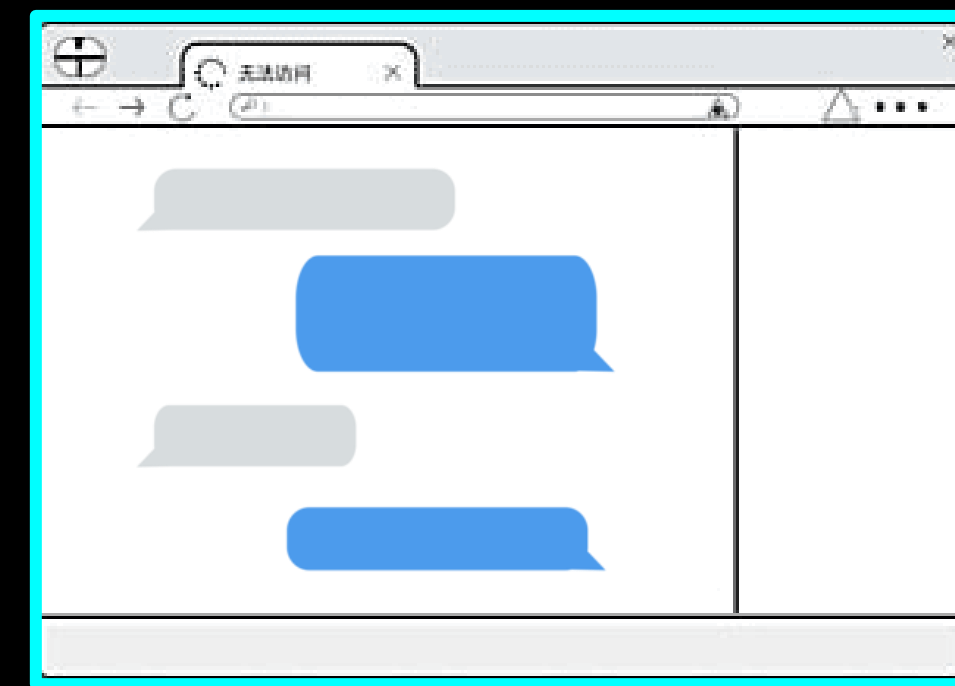
생성형 AI는 입력된 데이터를 바탕으로 새로운 텍스트, 이미지, 음악 등 창의적인 콘텐츠를 만들어내는 인공지능 기술을 말합니다. 대표적으로...



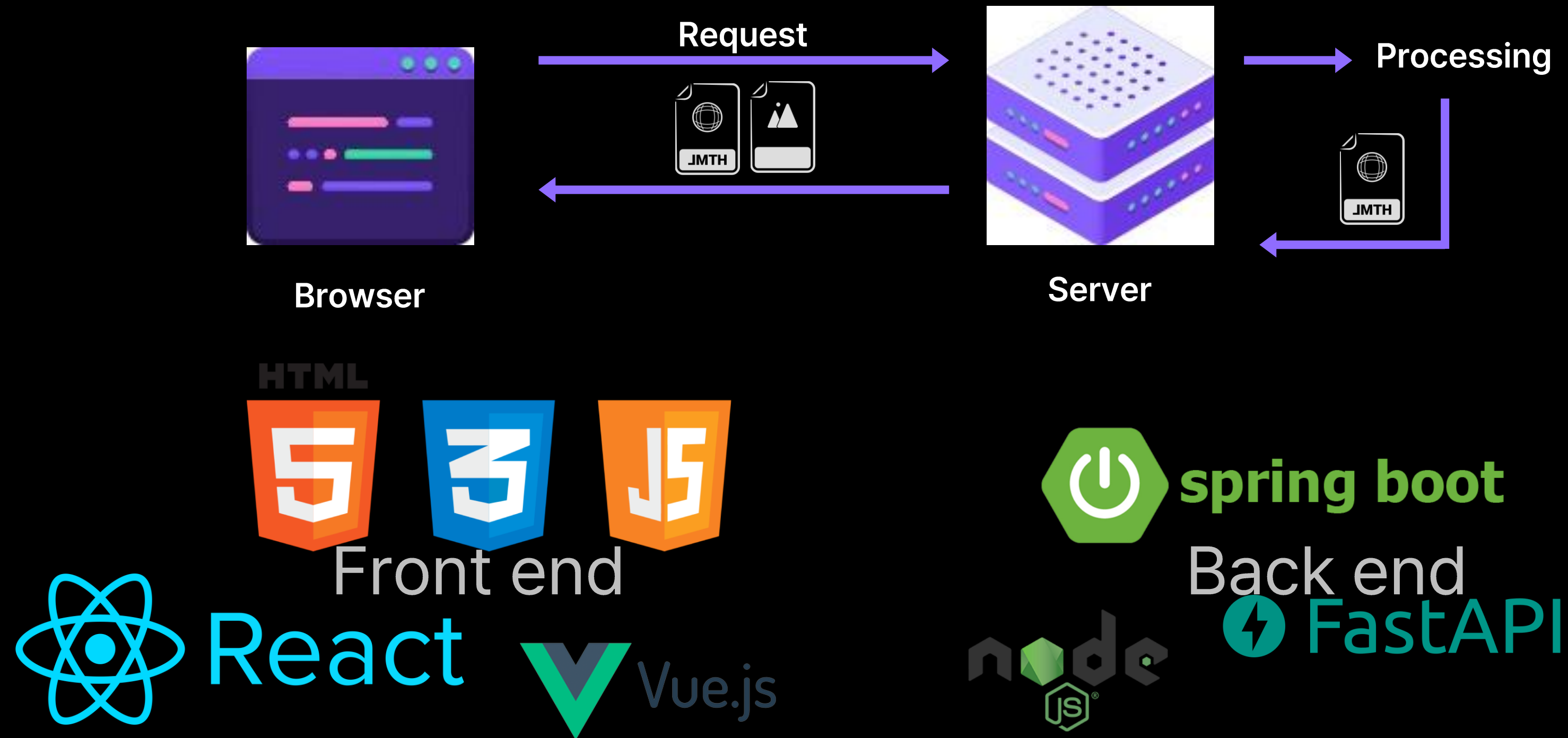
생성형 AI가 뭐야?



생성형 AI는 입력된 데이터를 바탕으로 새로운 텍스트, 이미지, 음악 등 창의적인 콘텐츠를 만들어내는 인공지능 기술을 말합니다. 대표적으로...

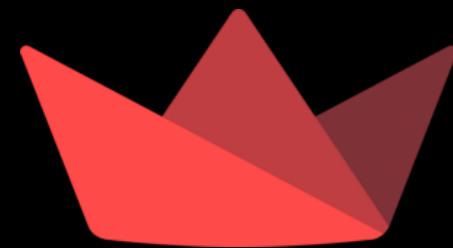


동적페이지



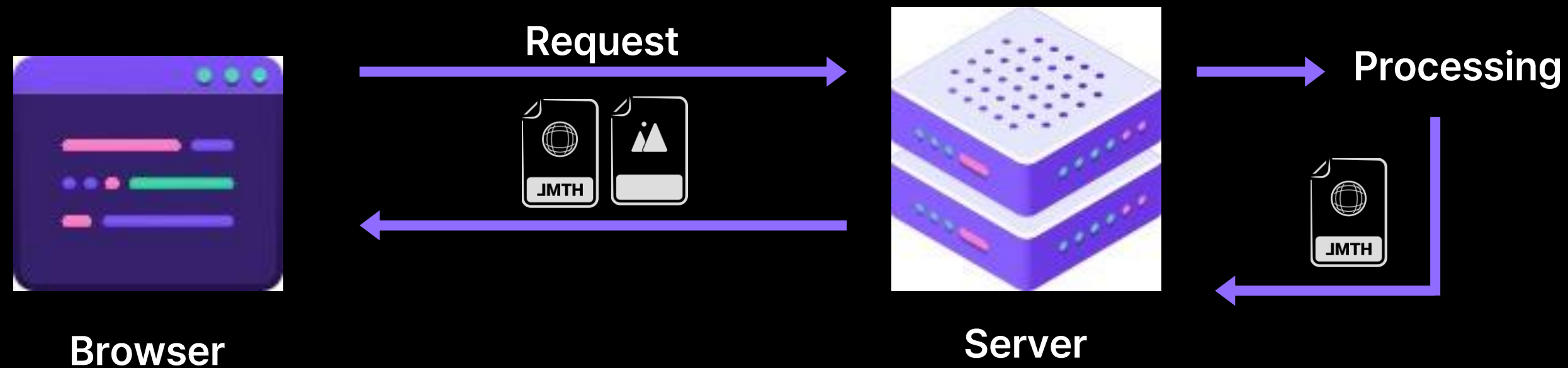
2. Streamlit

- Python을 사용하여 간단하게 웹 애플리케이션을 만들 수 있는 오픈 소스 프레임워크
- 복잡한 웹 개발 기술 없이도 웹 앱 형태로 공유 가능
- 실시간으로 데이터를 업데이트하고 사용자 상호작용을 지원해 다양한 활용이 가능

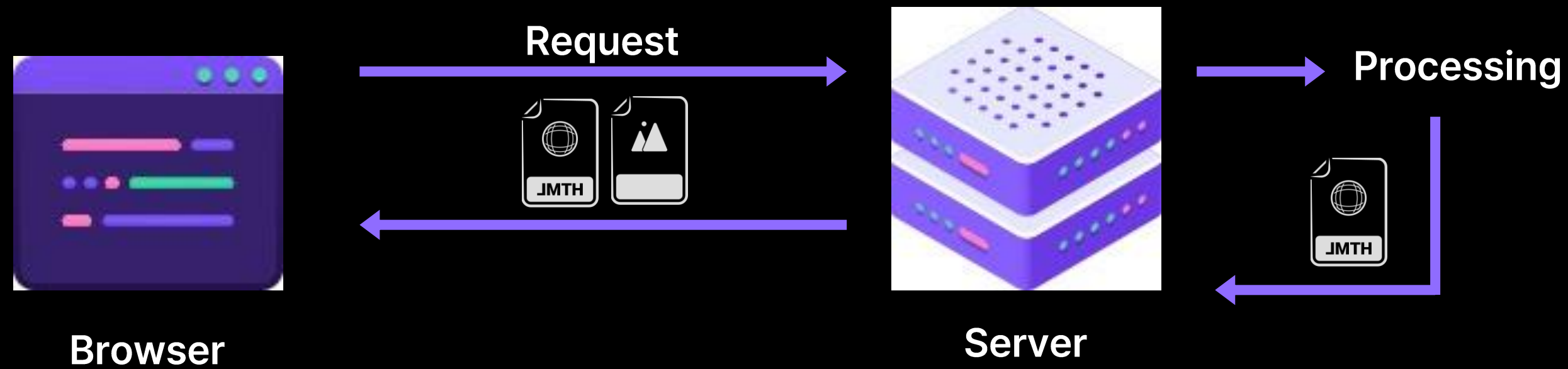


Streamlit

동적페이지



동적페이지



Streamlit



장점	단점
사용 방법을 배우기 쉽다.	자유도가 낮다.
웹 애플리케이션을 개발하는데 드는 시간이 단축된다.	특정 부분을 바꾸기 어려울 수 있다.
Front end와 Back end에 대한 지식을 크게 요하지 않는다.	일부 Streamlit Cloud에 의존적일 수 있다. = 배포 서비스가 끝날 수 있음
배포하는 것 역시 단순하다.	
비용이 들지 않는다.	



Streamlit과 다른 웹 프레임워크 비교

LangChain 활용 방법

LangChain 기반 챗봇
웹 애플리케이션 구현 프
로젝트

Streamlit

```
import streamlit as st
import pandas as pd
import numpy as np
import altair as alt

df = pd.DataFrame(
    np.random.randn(200, 3),
    columns=['a', 'b', 'c'])

c = alt.Chart(df).mark_circle().encode(
    x='a', y='b', size='c', color='c', tooltip=['a', 'b', 'c'])

st.write(c)
```

같은 내용 대비 양이 적은
Streamlit 코드

```
from flask import Flask, render_template
import pandas as pd
import numpy as np
import altair as alt

app = Flask(__name__)

@app.route('/')
def index():
    df = pd.DataFrame(
        np.random.randn(200, 3),
        columns=['a', 'b', 'c'])

    c = alt.Chart(df).mark_circle().encode(
        x='a', y='b', size='c', color='c', tooltip=['a', 'b', 'c'])

    chart = c.to_json()

    return render_template('index.html', chart=chart)

if __name__ == '__main__':
    app.run(debug=True)
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Flask Altair Example</title>
    <script src="https://cdn.jsdelivr.net/npm/vega@5"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-lite@5"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-embed@6"></script>
</head>
<body>
    <div id="chart"></div>
    <script>
        const chart = JSON.parse({{ chart|safe }});
        vegaEmbed('#chart', chart);
    </script>
</body>
</html>
```

같은 내용 대비 양이 많은
Flask 코드

- Streamlit Docs에서 Streamlit에서 제공하는 다양한 UI와 사용법을 확인할 수 있습니다.
- <https://docs.streamlit.io/develop/api-reference>

Example

```
import streamlit as st

st.button("Reset", type="primary")
if st.button("Say hello"):
    st.write("Why hello there")
else:
    st.write("Goodbye")
```

Reset

Say hello

Goodbye

3. 실습 설명



실습 환경 세팅

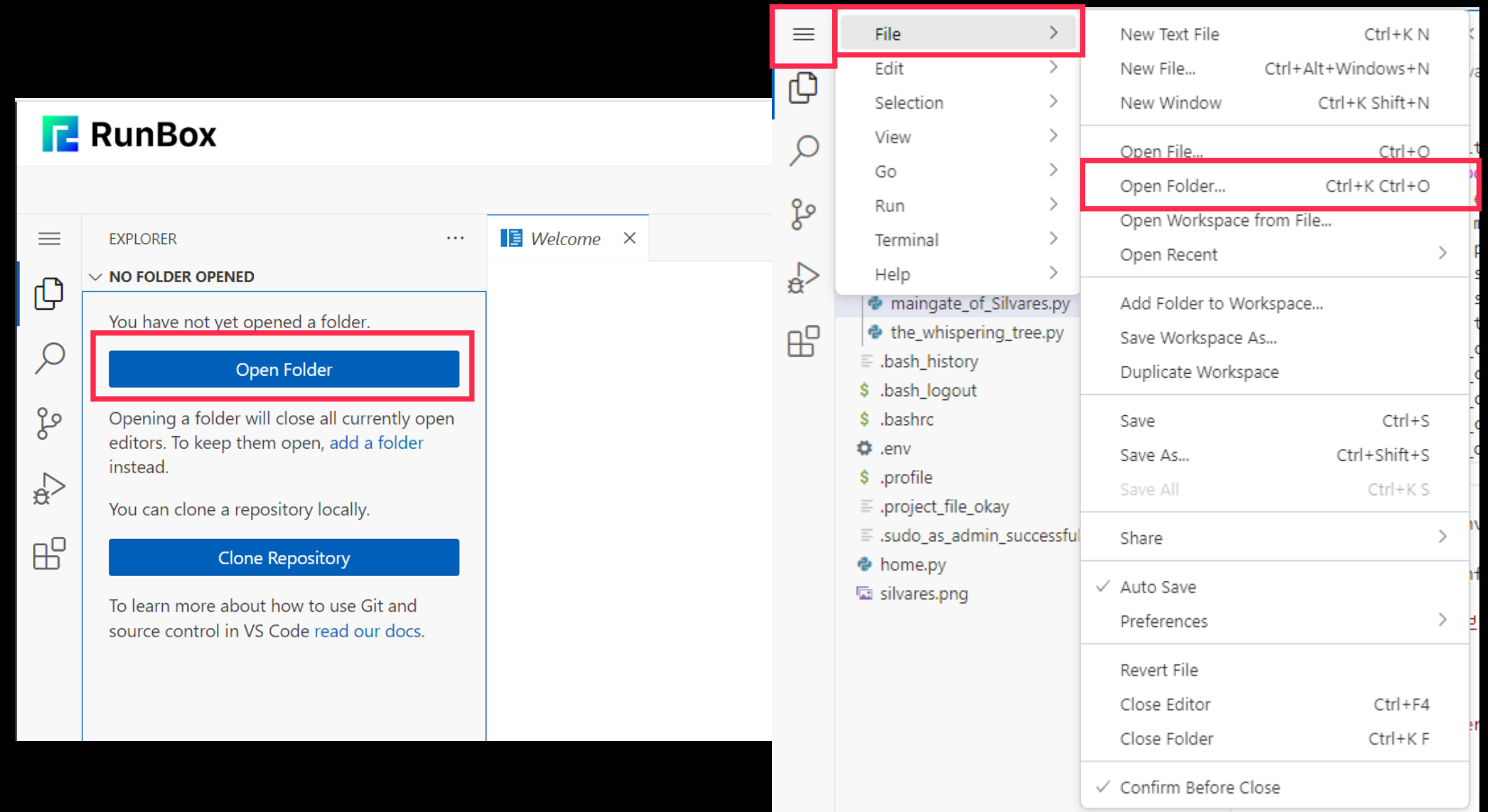
LangChain 활용 방법

LangChain 기반 챗봇
웹 애플리케이션 구현 프
로젝트

실습 설명

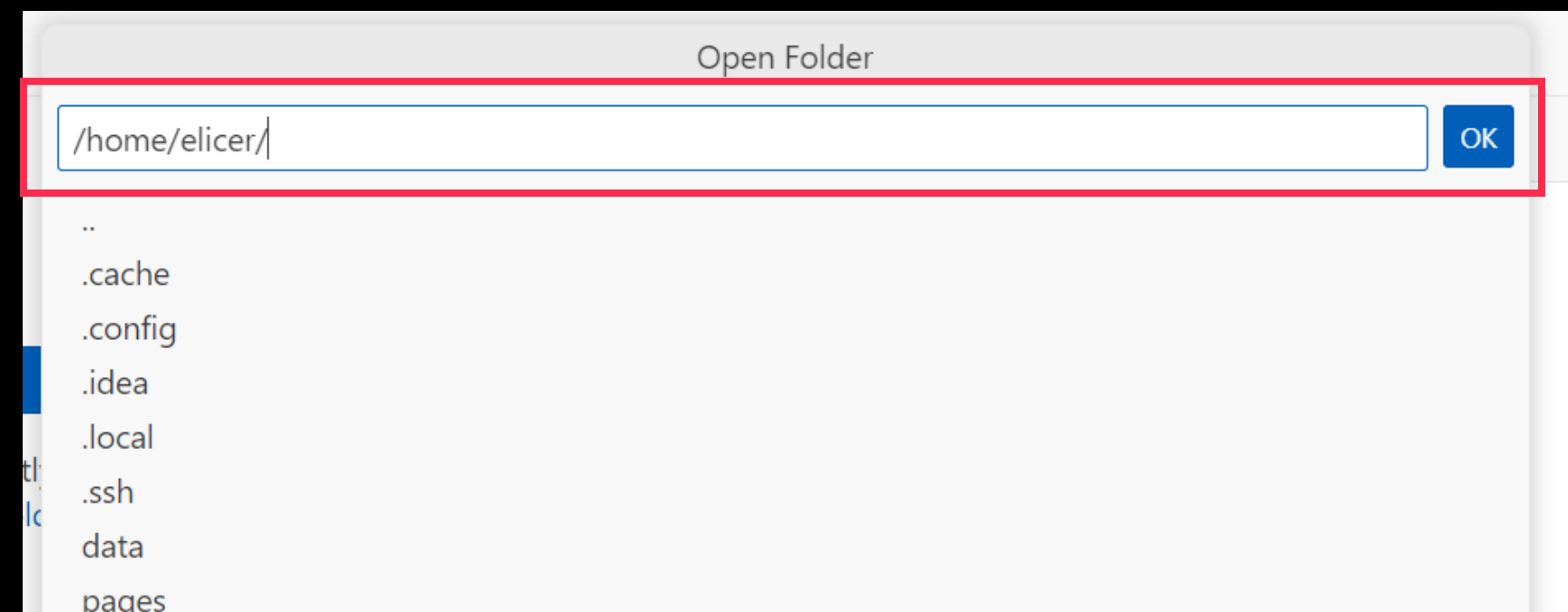
1

Streamlit 실습에 접속하여
왼쪽 상단에 'Open Folder' 버튼을 클릭합니다.



2

'Open Folder' 버튼 클릭 후
기본으로 설정된 /home/elicer/ 경로에
'OK' 버튼을 눌러 접속합니다.





OpenAI API key 설정

LangChain 활용 방법

LangChain 기반 챗봇
웹 애플리케이션 구현 프
로젝트

실습 설명

- 1 ".env" 파일에 들어가
본인의 API key를 입력해주세요.

EXPLORER

ELICER

- .cache
- .config
- .idea
- .local
- .ssh
- pages
 - 01_RAG_with_Youtube.py
- .bash_history
- .bash_logout
- .bashrc
- .env**
- .profile
- .project_file_okay
- .sudo_as_admin_successful
- home.py

.env

```
1 OPENAI_API_KEY = "이곳에 api key를 입력하세요"
```

↓

.env

```
1 OPENAI_API_KEY = "sk-dlrjtdmsdPtldydAPIzldlqslek123"
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS 1

elicer@f463ad16efd7:~\$



웹페이지 실행하기

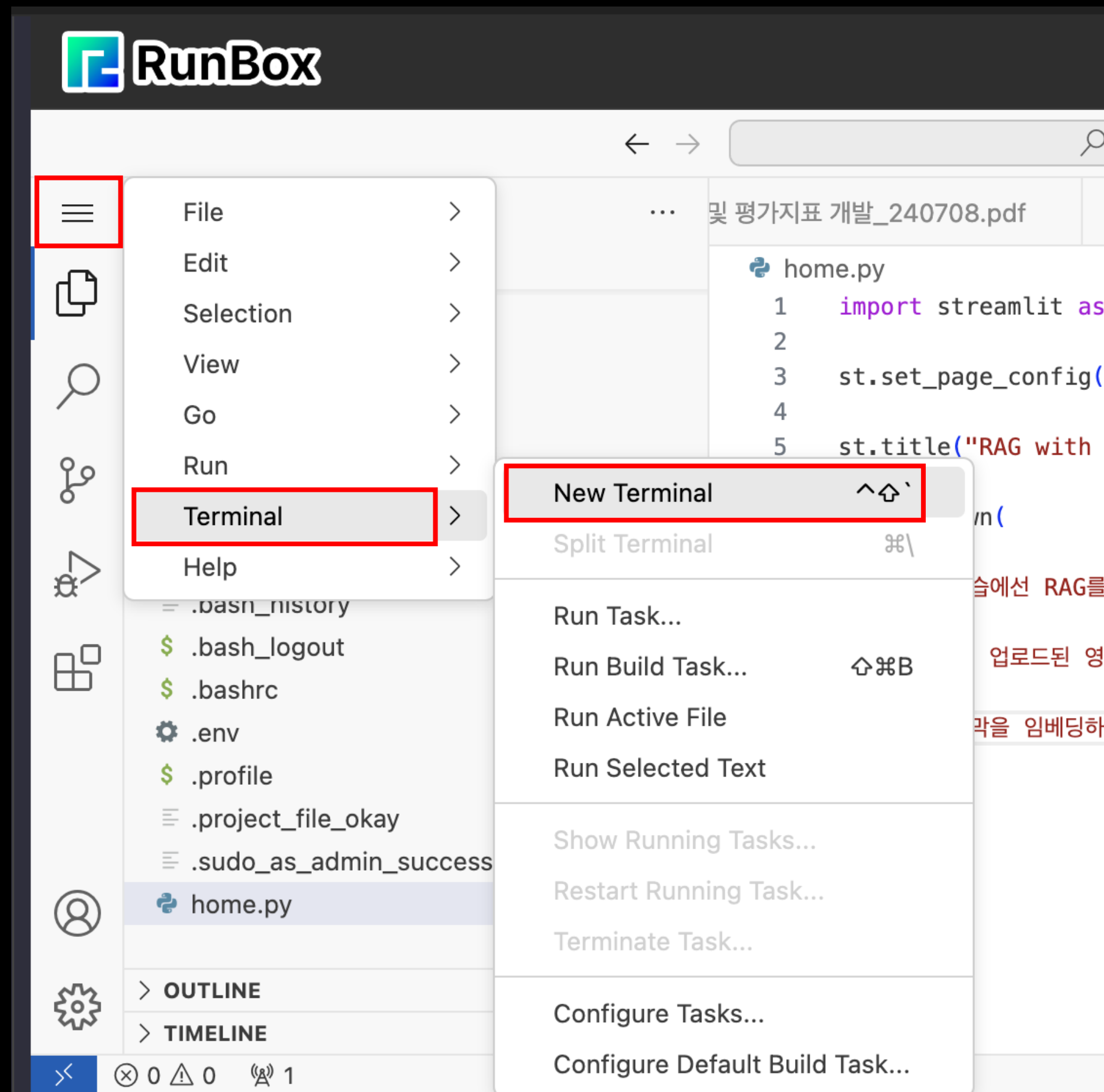
LangChain 활용 방법

LangChain 기반 챗봇
웹 애플리케이션 구현 프
로젝트

실습 설명

1

우측 그림과 같이 순서대로 클릭하여
터미널을 실행시킵니다





- 2 하단에 TERMINAL 창이 켜지면
`streamlit run home.py` 명령어로
프로세스를 실행시킵니다.

The screenshot shows a VS Code terminal window with the 'TERMINAL' tab selected. The terminal output shows the command 'streamlit run home.py' being executed. The output includes a message about collecting usage statistics, a confirmation that the Streamlit app is viewable in the browser, and the network and external URLs. A notification bubble at the bottom right states that the application is available on port 8501 and provides a link to 'See all forwarded ports' and a button to 'Open in Browser'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2 python + v [ ] [ ] ... ^ x
○ elicor@ced55a561785:~$ streamlit run home.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: http://10.0.2.100:8501
External URL: http://14.35.173.251:8501

(i) Your application running on port 8501 is available. See all forwarded ports
Open in Browser
```



웹페이지 실행하기

LangChain 활용 방법

LangChain 기반 챗봇
웹 애플리케이션 구현 프
로젝트

실습 설명

- 3 이후 'PORTS' 탭으로 이동하여
생성된 포트를 우클릭한 후,
'Open in Browser'를 클릭하여
새롭게 뜨는 창을 확인합니다.

home.py

```
1 import streamlit as st
2
3 st.set_page_config(page_title="Home", page_icon=":house:")
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

Port	Forwarded Ad...	Running Process	Origin
8501	https://...	/usr/local/bin/python /home/elicer/...	Auto Forwarded

Add Port (우클릭) Open in Browser 3

python + v ... a Port

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2

elicer@ced55a561785:~\$ streamlit run home.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

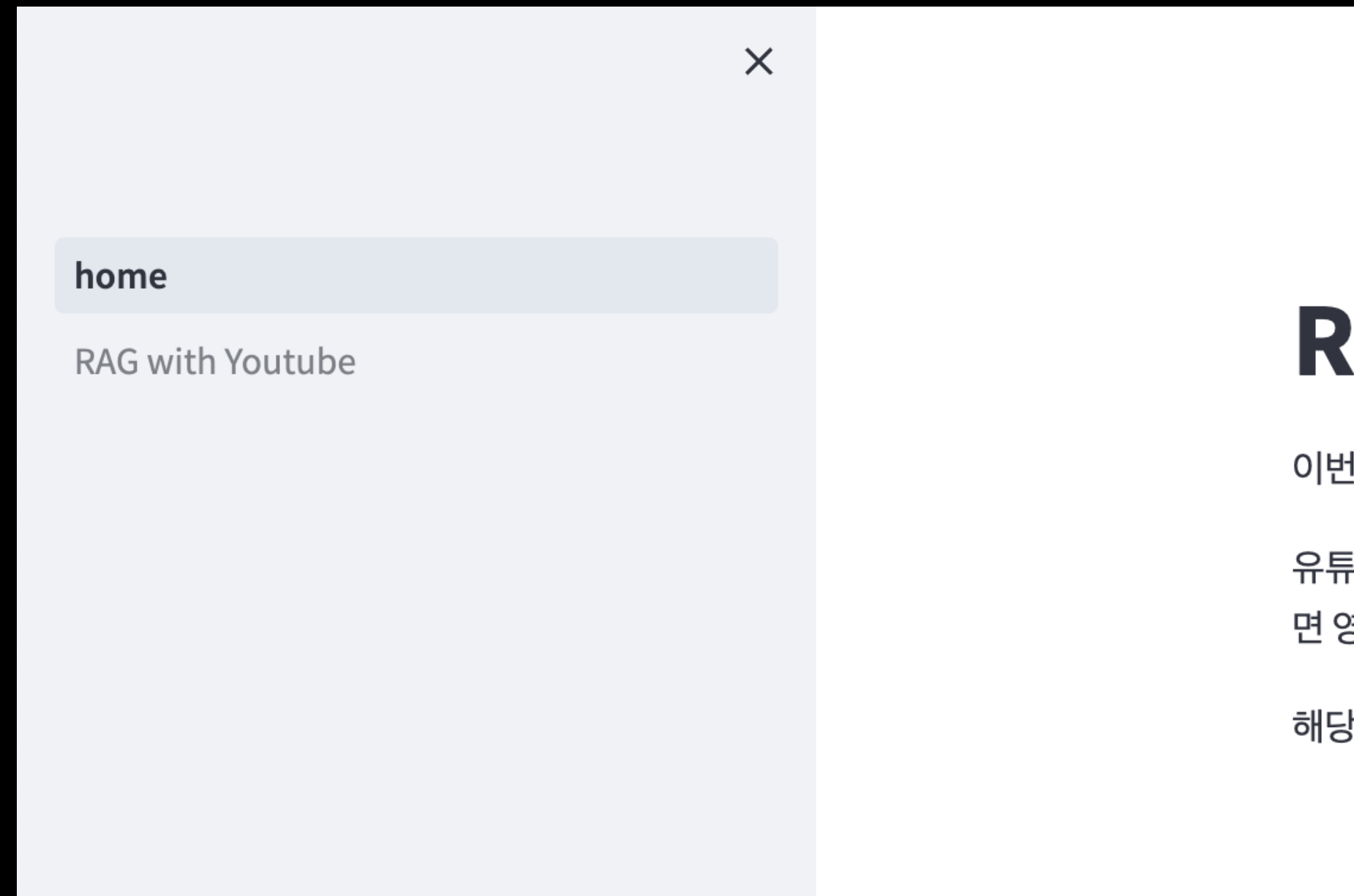
Network URL: http://10.0.2.100:8501
External URL: http://14.35.173.251:8501

Your application running on port 8501 is available. See all forwarded ports

Open in Browser



- 4 Home 페이지의 좌측에는 페이지의 목록이 있습니다.
home 이 있고 그 밑으로 실습에 활용할 페이지의
목록이 있습니다.
실습 순서에 맞추어 이동하면서 실습을 수행합니다.





- 5 Streamlit은 .py 소스코드에 변경사항이 발생하면 자동으로 이를 감지하여, 우측 상단에 'Rerun'과 'Always rerun' 옵션을 표시합니다.
- 'Rerun': 클릭 시 현재 페이지에 변경사항 반영
 - 'Always rerun': 클릭 시 세션이 종료될 때까지 소스코드 변경사항 자동으로 반영



Source file changed.

Rerun

Always rerun



RAG with Youtube Video