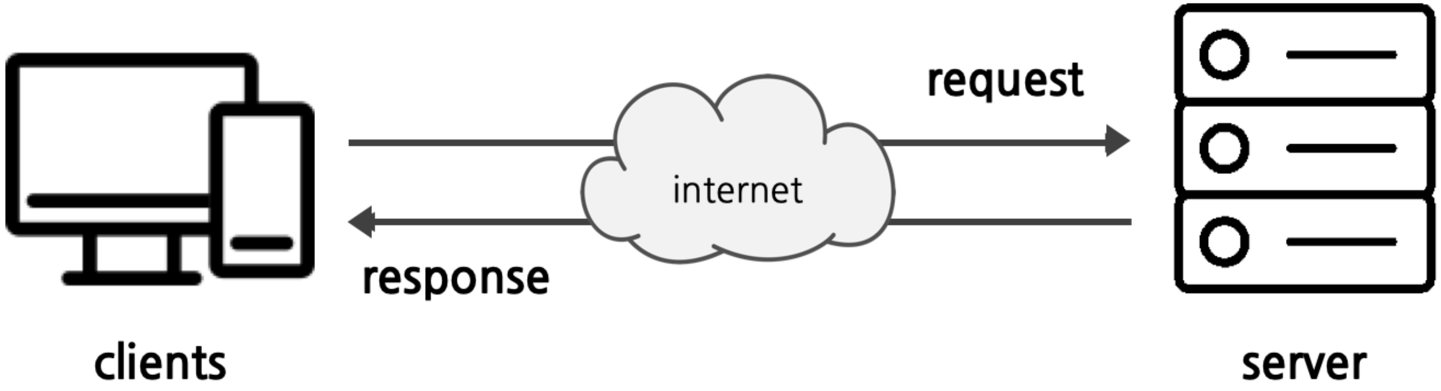
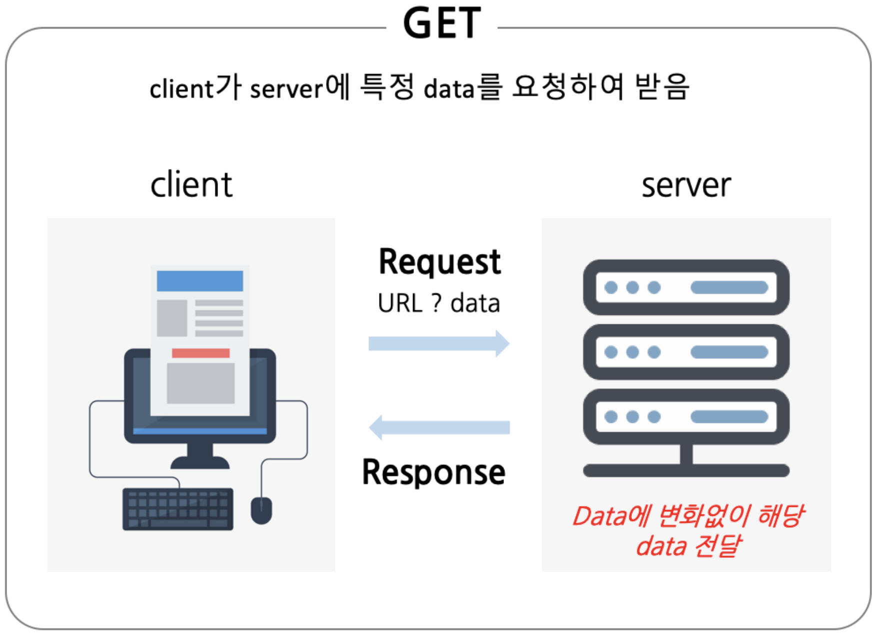
# 7. AI 애플리케이션 개발

Unlock your creativity with LLMs

박경규

# 웹서비스 구조



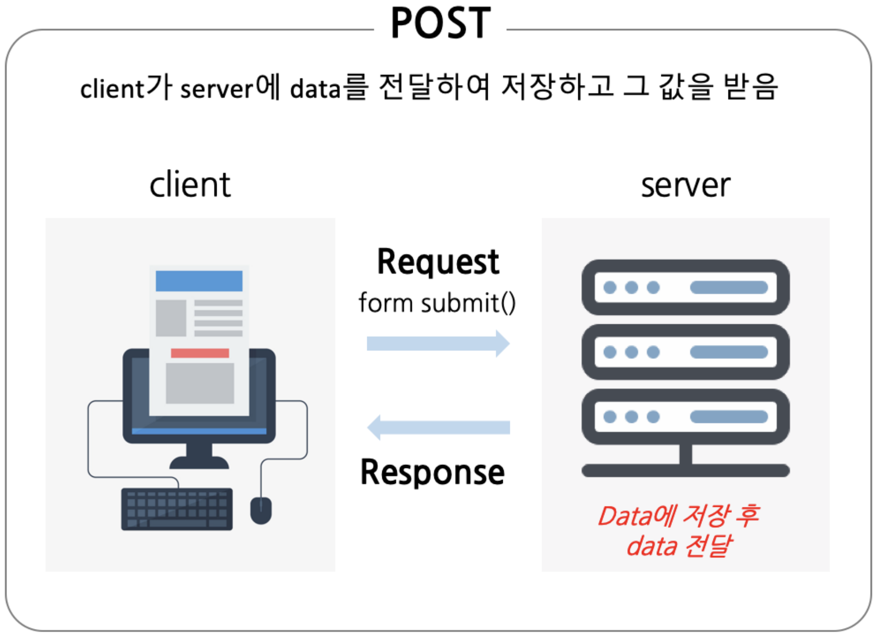


## GET

client가 server에 특정 data를 요청하여 받음

*Data에 변화없이 해당 data 전달*

**VS**



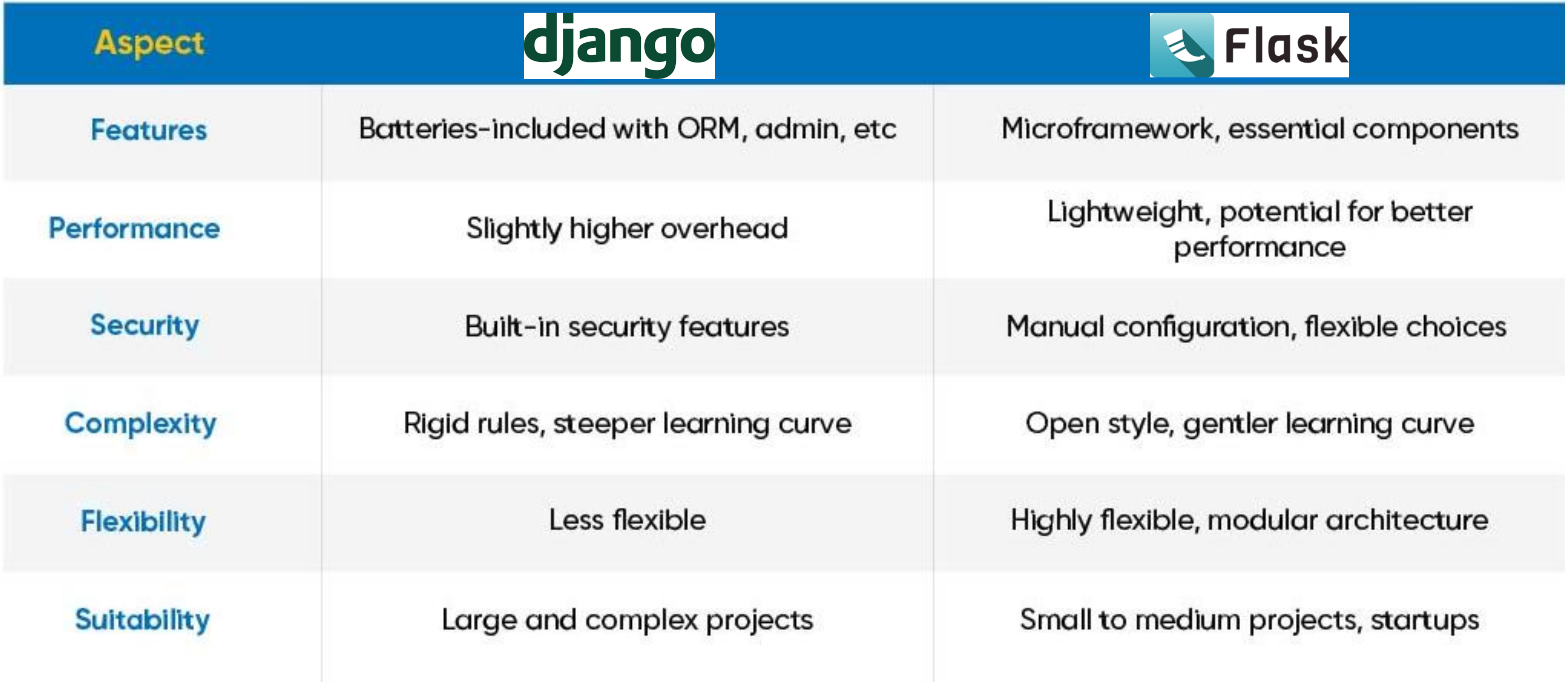
## POST

client가 server에 data를 전달하여 저장하고 그 값을 받음

*Data에 저장 후 data 전달*

https://bit.ly/2PTfLwE

# 웹 개발 프레임워크



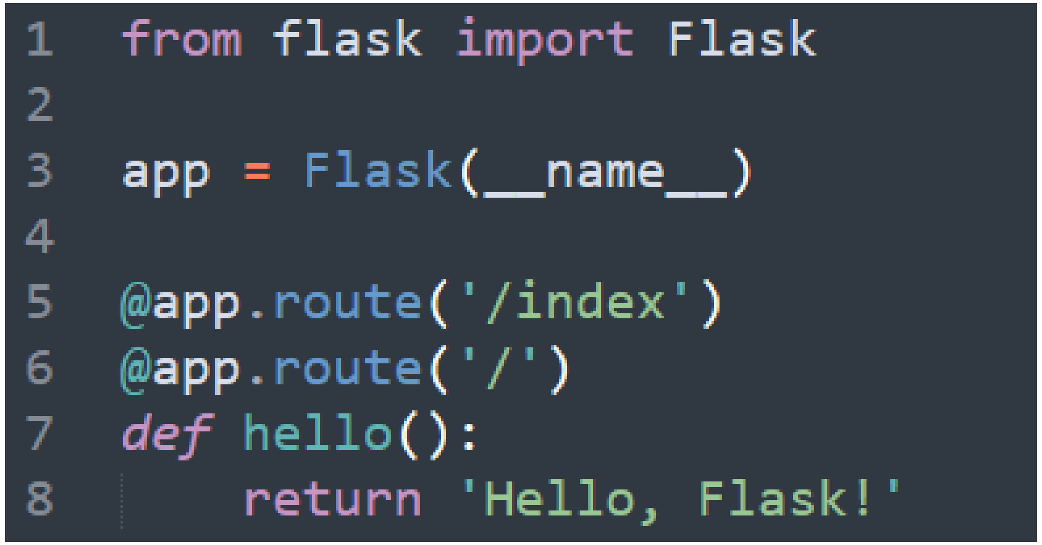
| Aspect | django | Flask |
|---|---|---|
| Features | Batteries-included with ORM, admin, etc | Microframework, essential components |
| Performance | Slightly higher overhead | Lightweight, potential for better performance |
| Security | Built-in security features | Manual configuration, flexible choices |
| Complexity | Rigid rules, steeper learning curve | Open style, gentler learning curve |
| Flexibility | Less flexible | Highly flexible, modular architecture |
| Suitability | Large and complex projects | Small to medium projects, startups |

# Flask

## ■ Flask 설치

- pip install flask

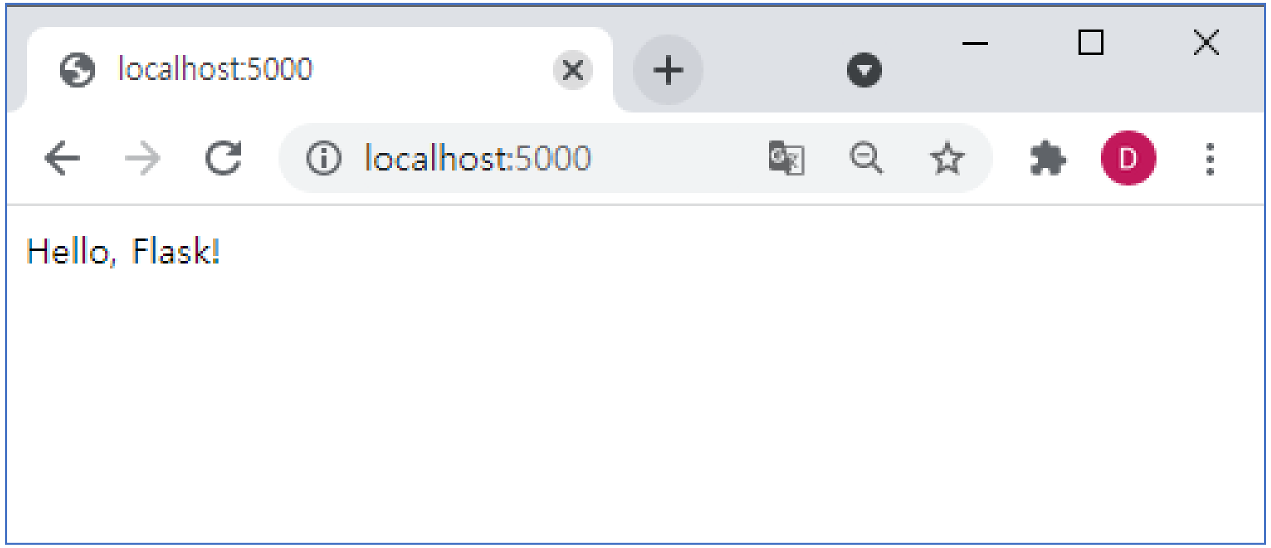
## ■ 프로그램 작성 : app.py



```
from flask import Flask

app = Flask(__name__)

@app.route('/index')
@app.route('/')
def hello():
    return 'Hello, Flask!'
```

## ■ 프로그램 실행

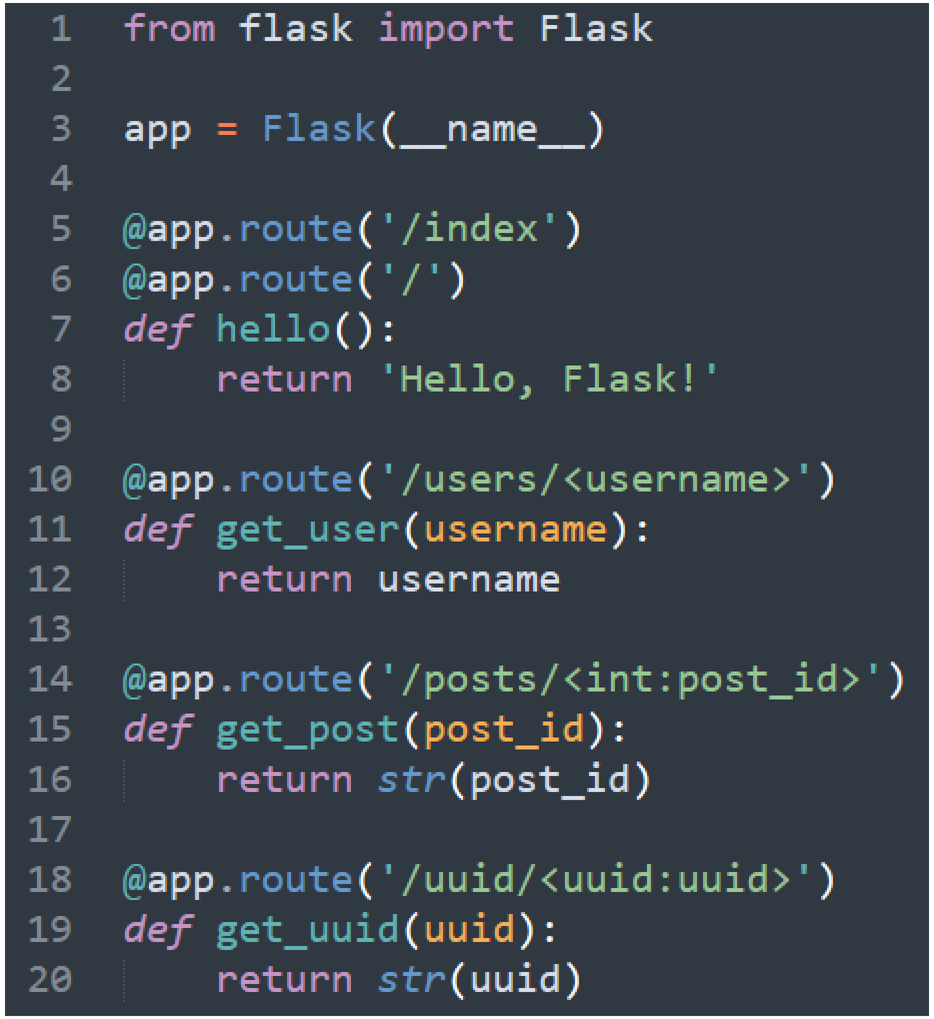
- flask run
- 웹브라우저에서 http://127.0.0.1:5000/ 접속



```
Hello, Flask!
```

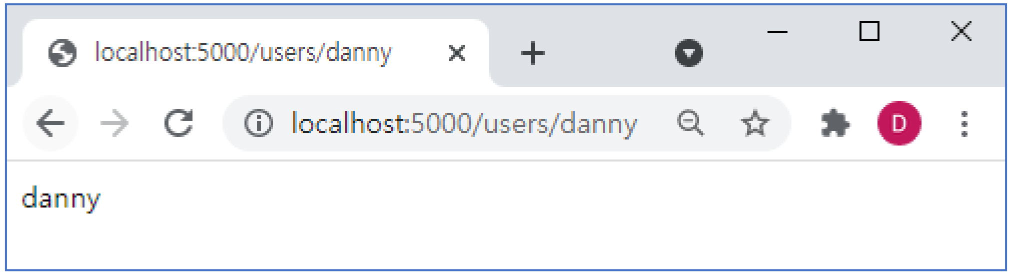
# Flask – 파라미터(parameter)

## app.py



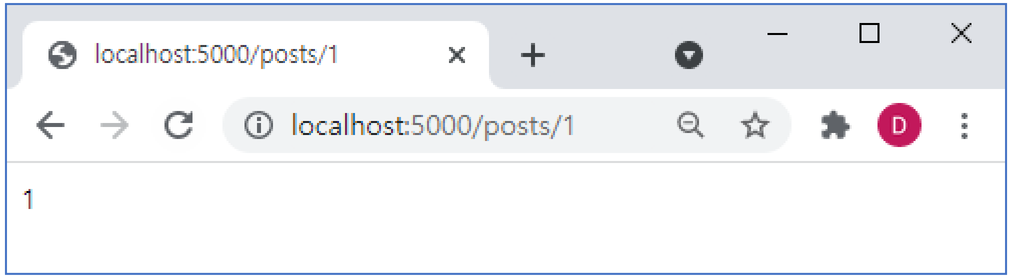
```python
from flask import Flask

app = Flask(__name__)

@app.route('/index')
@app.route('/')
def hello():
    return 'Hello, Flask!'

@app.route('/users/<username>')
def get_user(username):
    return username

@app.route('/posts/<int:post_id>')
def get_post(post_id):
    return str(post_id)

@app.route('/uuid/<uuid:uuid>')
def get_uuid(uuid):
    return str(uuid)
```

http://localhost:5000/users/danny



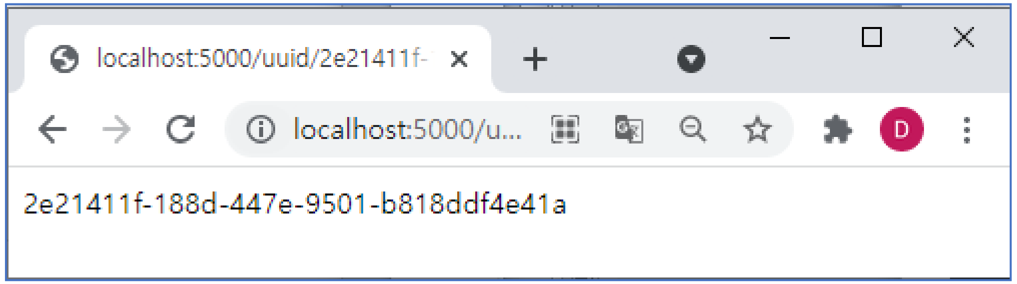
```
danny
```

http://localhost:5000/posts/1



```
1
```

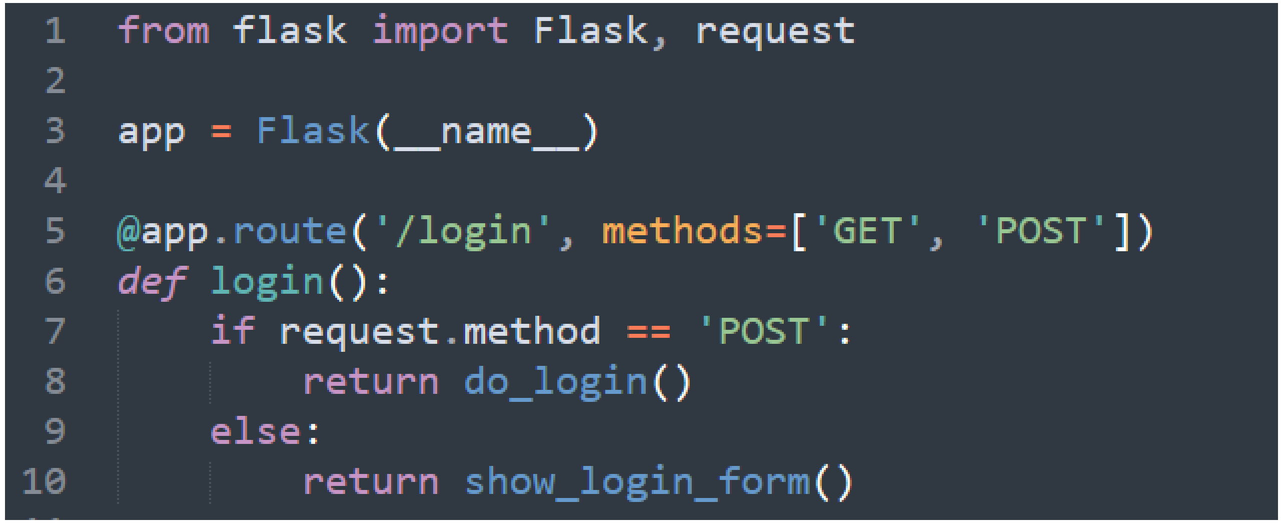
http://localhost:5000/uuid/2e21411f-188d-447e-9501-b818ddf4e41a



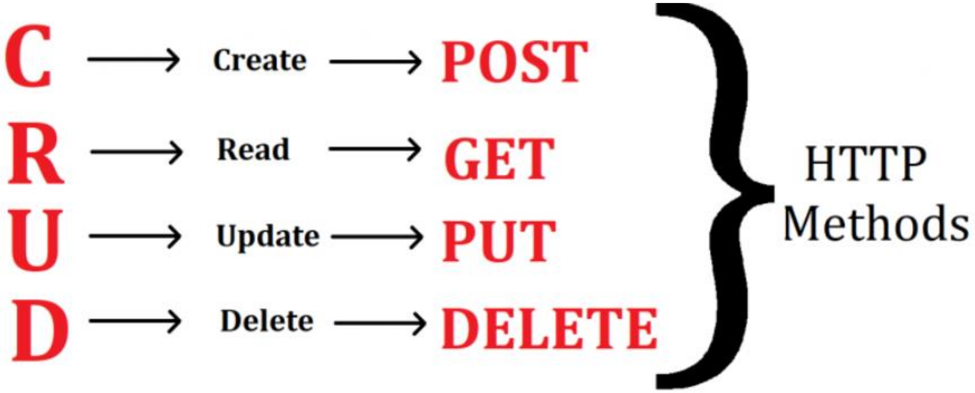
```
2e21411f-188d-447e-9501-b818ddf4e41a
```

# Flask – 메서드(method)

## login.py



```
from flask import Flask, request

app = Flask(__name__)

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        return do_login()
    else:
        return show_login_form()
```

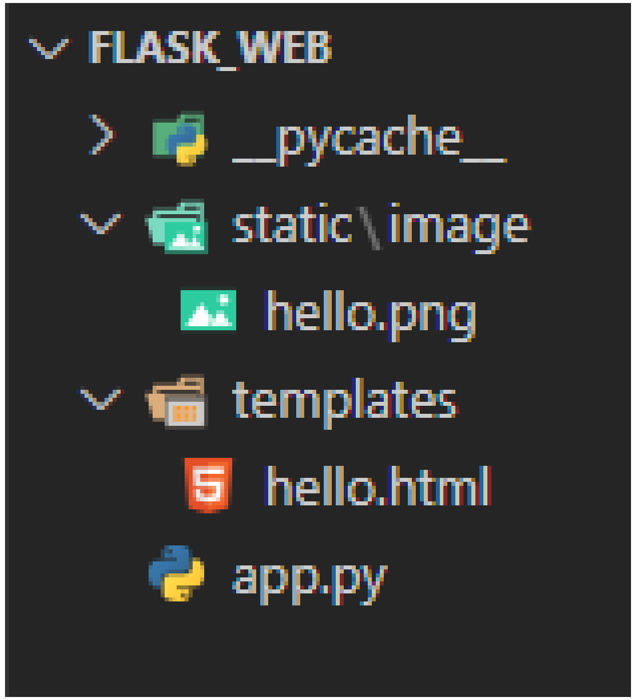


*HTTP Method*

# Flask – 템플릿

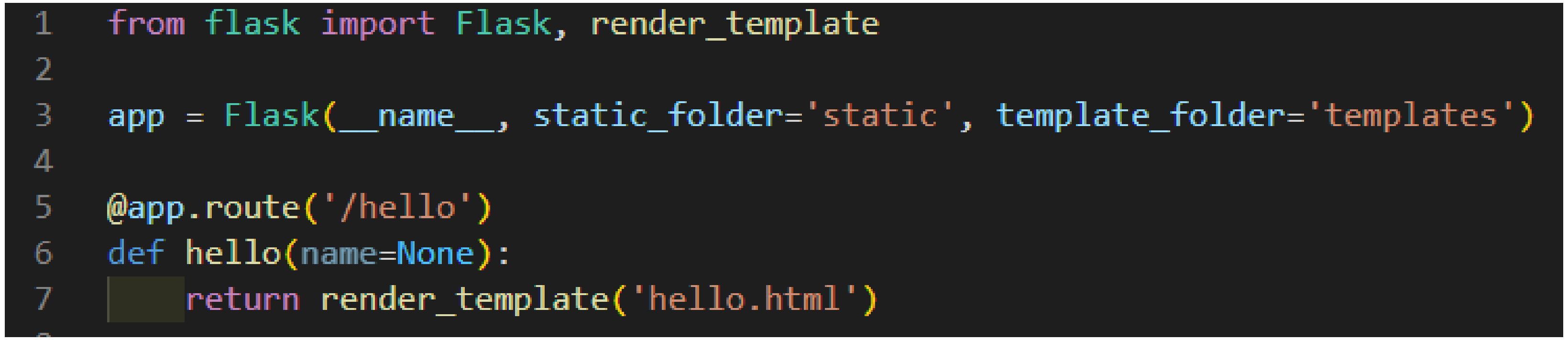
hello-app

## ■ 디렉토리 구조



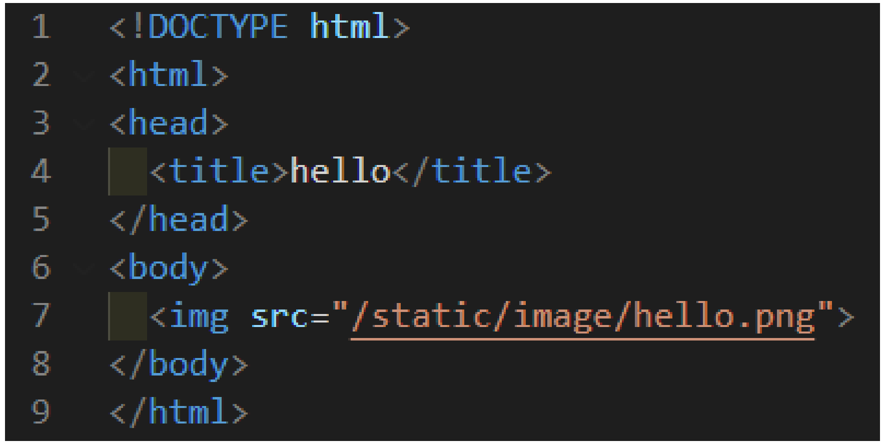
```
FLASK_WEB
    _pycache_
    static\image
        hello.png
    templates
        hello.html
    app.py
```

## ■ app.py

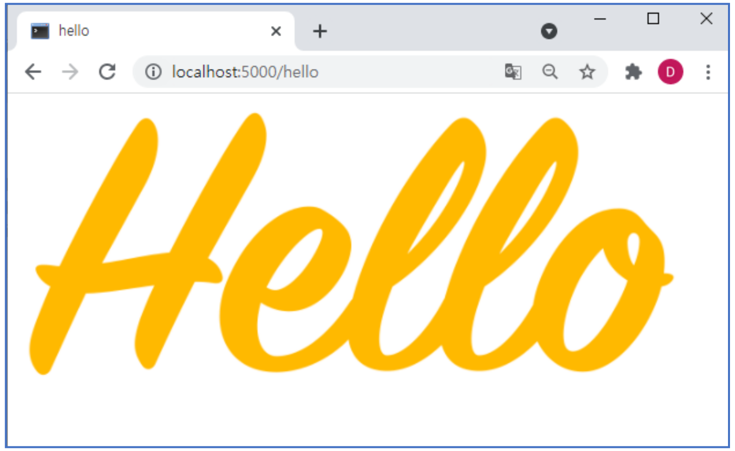


```python
from flask import Flask, render_template

app = Flask(__name__, static_folder='static', template_folder='templates')

@app.route('/hello')
def hello(name=None):
    return render_template('hello.html')
```

## ■ hello.html



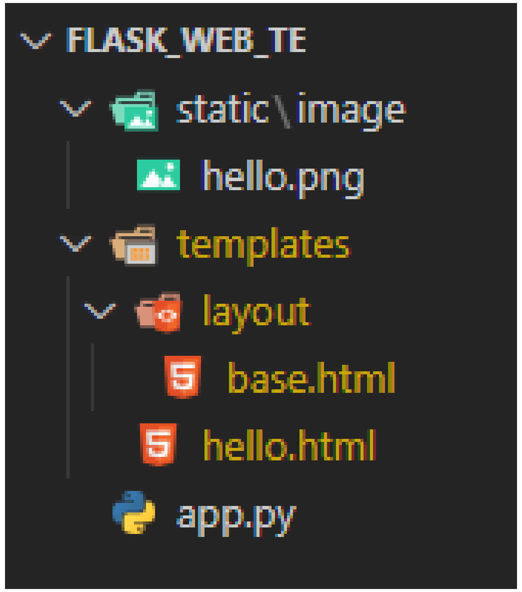
```html
<!DOCTYPE html>
<html>
<head>
    <title>hello</title>
</head>
<body>
    <img src="/static/image/hello.png">
</body>
</html>
```



# Flask – 템플릿엔진

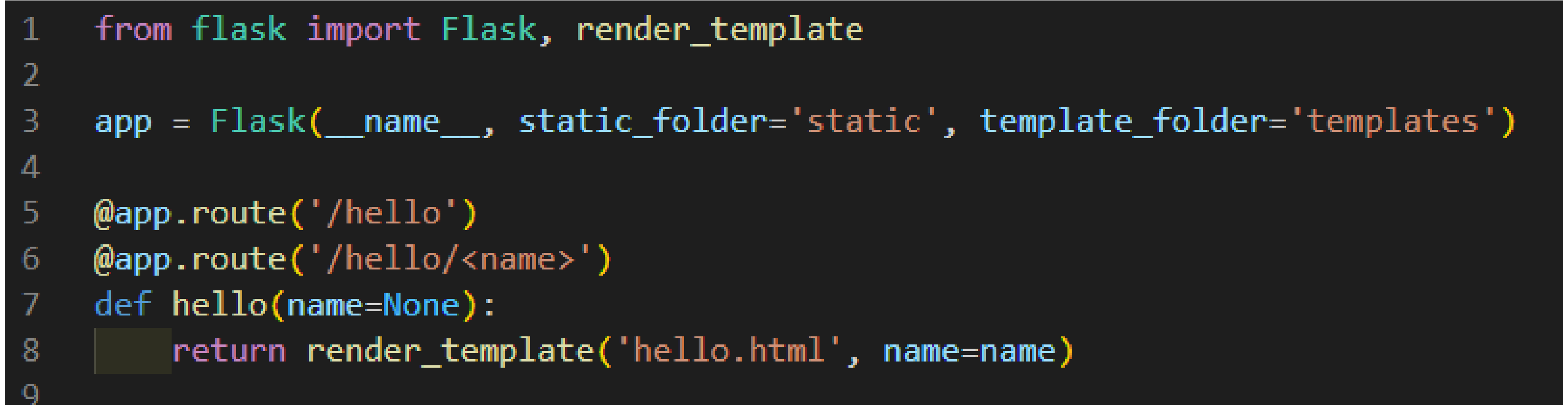
hello-app-te

## 디렉토리 구조



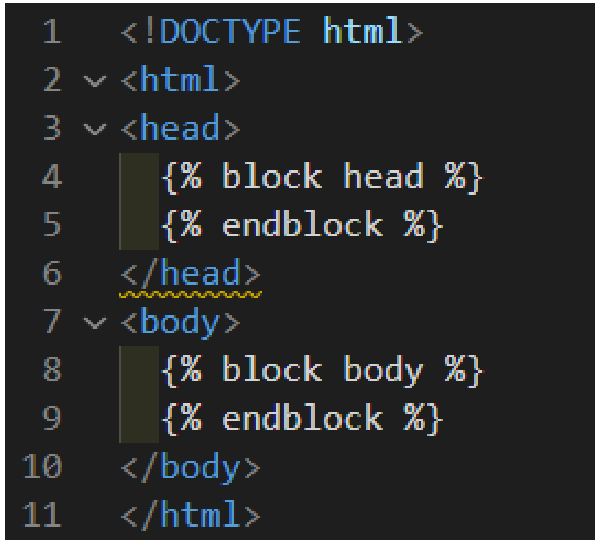
```
FLASK_WEB_TE
  static\image
    hello.png
  templates
    layout
      base.html
    hello.html
  app.py
```

## app.py



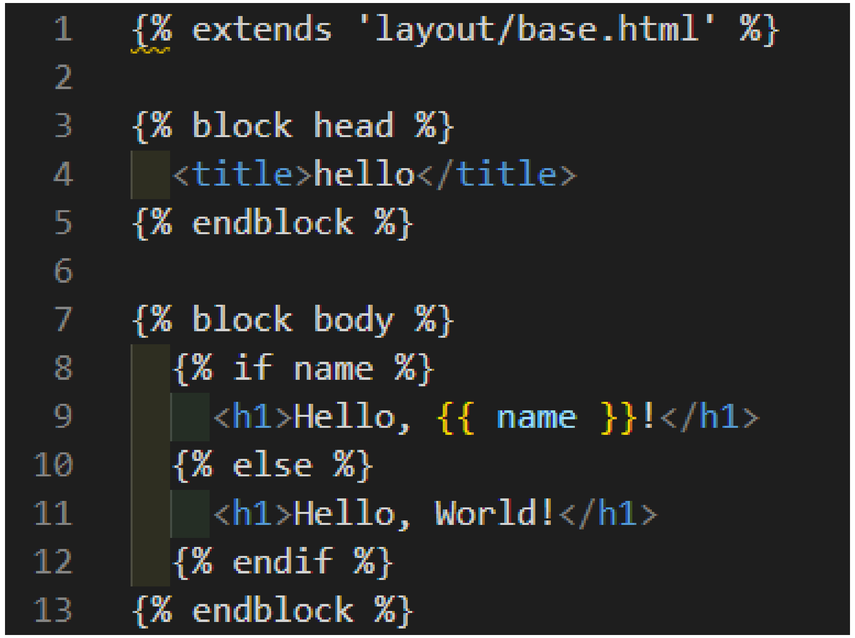
```python
from flask import Flask, render_template

app = Flask(__name__, static_folder='static', template_folder='templates')

@app.route('/hello')
@app.route('/hello/<name>')
def hello(name=None):
    return render_template('hello.html', name=name)

```

## base.html



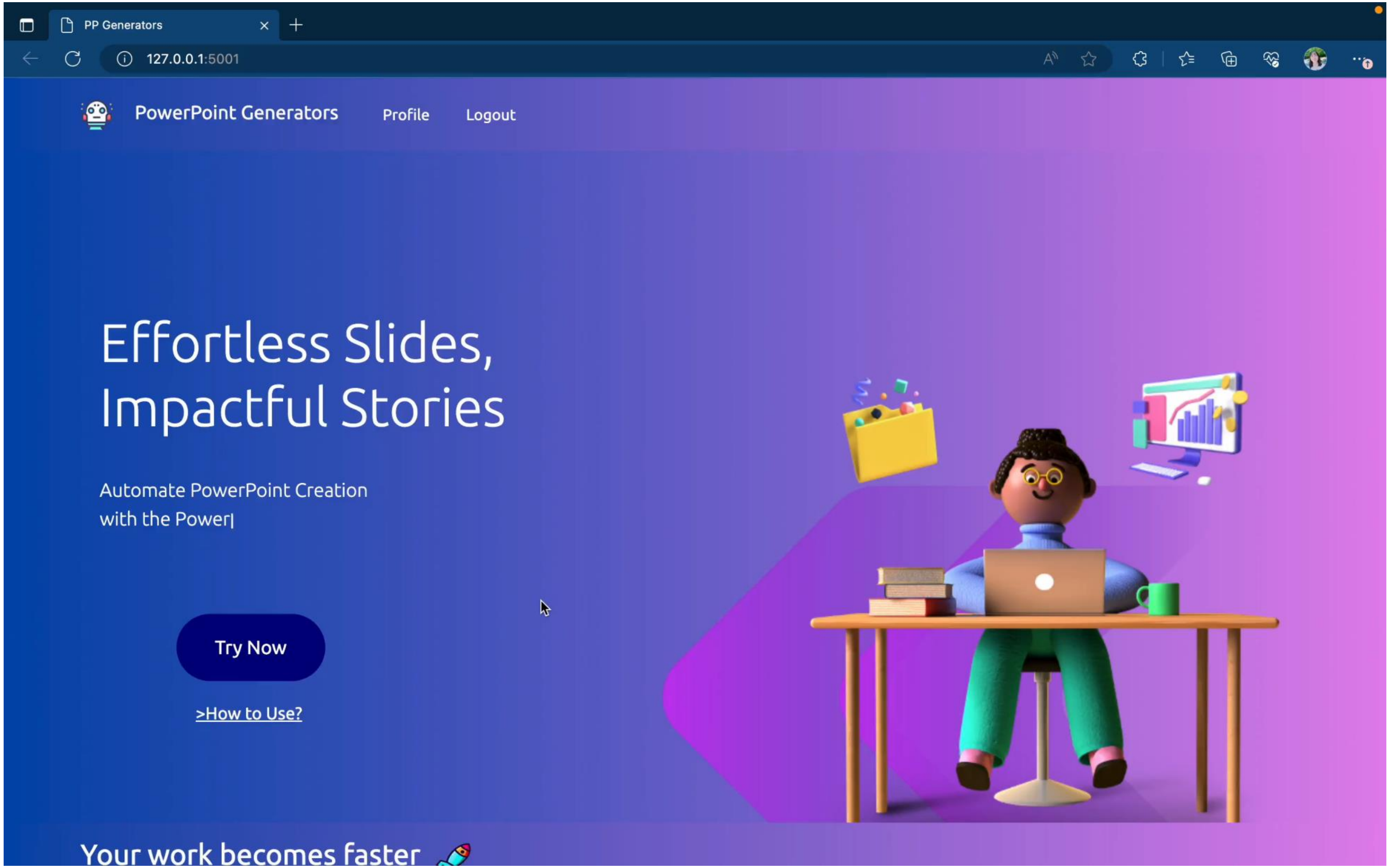
```html
<!DOCTYPE html>
<html>
<head>
    {% block head %}
    {% endblock %}
</head>
<body>
    {% block body %}
    {% endblock %}
</body>
</html>
```

## hello.html



```html
{% extends 'layout/base.html' %}

{% block head %}
    <title>hello</title>
{% endblock %}

{% block body %}
    {% if name %}
        <h1>Hello, {{ name }}!</h1>
    {% else %}
        <h1>Hello, World!</h1>
    {% endif %}
{% endblock %}
```

# PPT 생성앱 개발



PP Generators

127.0.0.1:5001

PowerPoint Generators

Profile

Logout

Effortless Slides,
Impactful Stories

Automate PowerPoint Creation
with the Power|

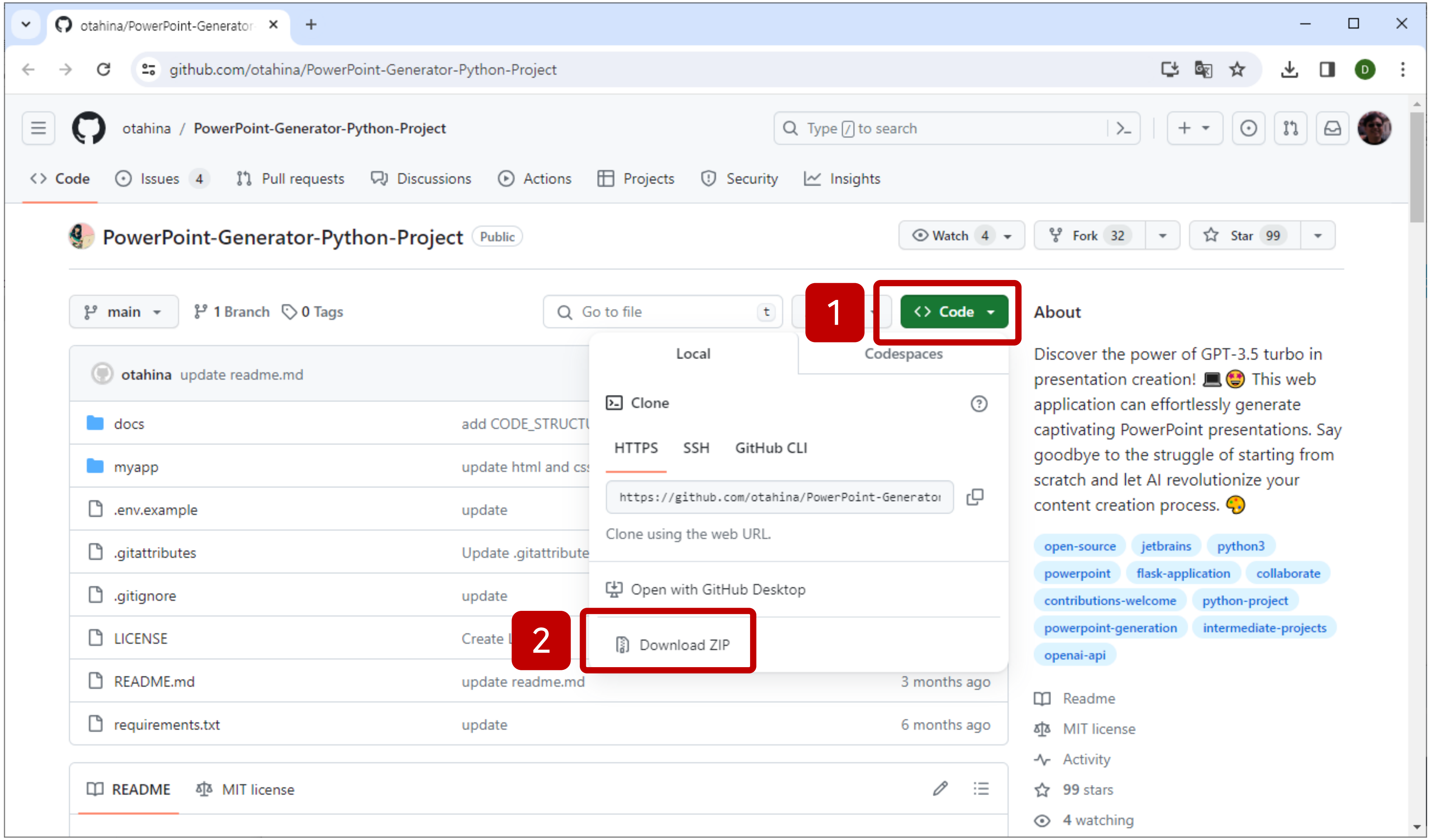
Try Now

>How to Use?

Your work becomes faster

# PPT 생성앱 개발

■ 소스 다운로드 https://github.com/otahina/PowerPoint-Generator-Python-Project



otahina update readme.md

| | | |
|---|---|---|
| docs | add CODE_STRUCTU | |
| myapp | update html and css | |
| .env.example | update | |
| .gitattributes | Update .gitattribute | |
| .gitignore | update | |
| LICENSE | Create l | |
| README.md | update readme.md | 3 months ago |
| requirements.txt | update | 6 months ago |

About

Discover the power of GPT-3.5 turbo in presentation creation! 💻😍 This web application can effortlessly generate captivating PowerPoint presentations. Say goodbye to the struggle of starting from scratch and let AI revolutionize your content creation process. 🎨

open-source jetbrains python3 powerpoint flask-application collaborate contributions-welcome python-project powerpoint-generation intermediate-projects openai-api

Readme

MIT license

Activity

99 stars

4 watching

# PPT 생성앱 개발

### 1. 소스 디렉토리에 있는 .env.example 파일명을 .env 로 변경하고 API 키를 입력

```
# Secret Key for Flask(You can decide on your own)
SECRET_KEY=your_secret_key

# OpenAI API Key
OPENAI_API_KEY=your_openai_key

# Pexels API Key
PEXELS_API_KEY=your_pexels_key
```

https://www.pexels.com/api/

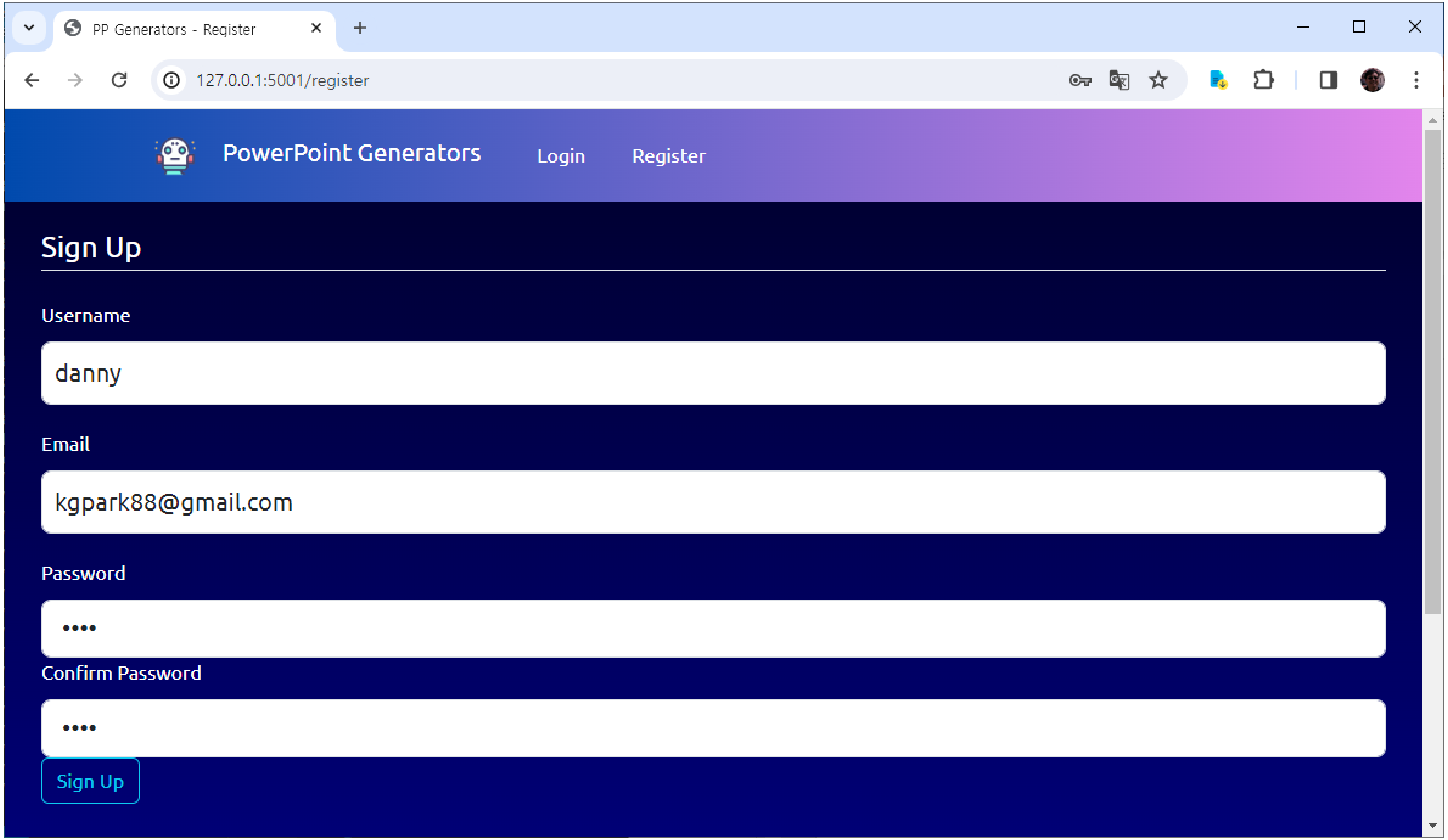
### 2. 파이썬 패키지 설치

```
pip install -r requirements.txt
```

### 3. 실행

```
cd myapp

python flaskapp.py
```

# PPT 생성앱 – 가입 화면



Sign Up

Username

danny

Email

kgpark88@gmail.com

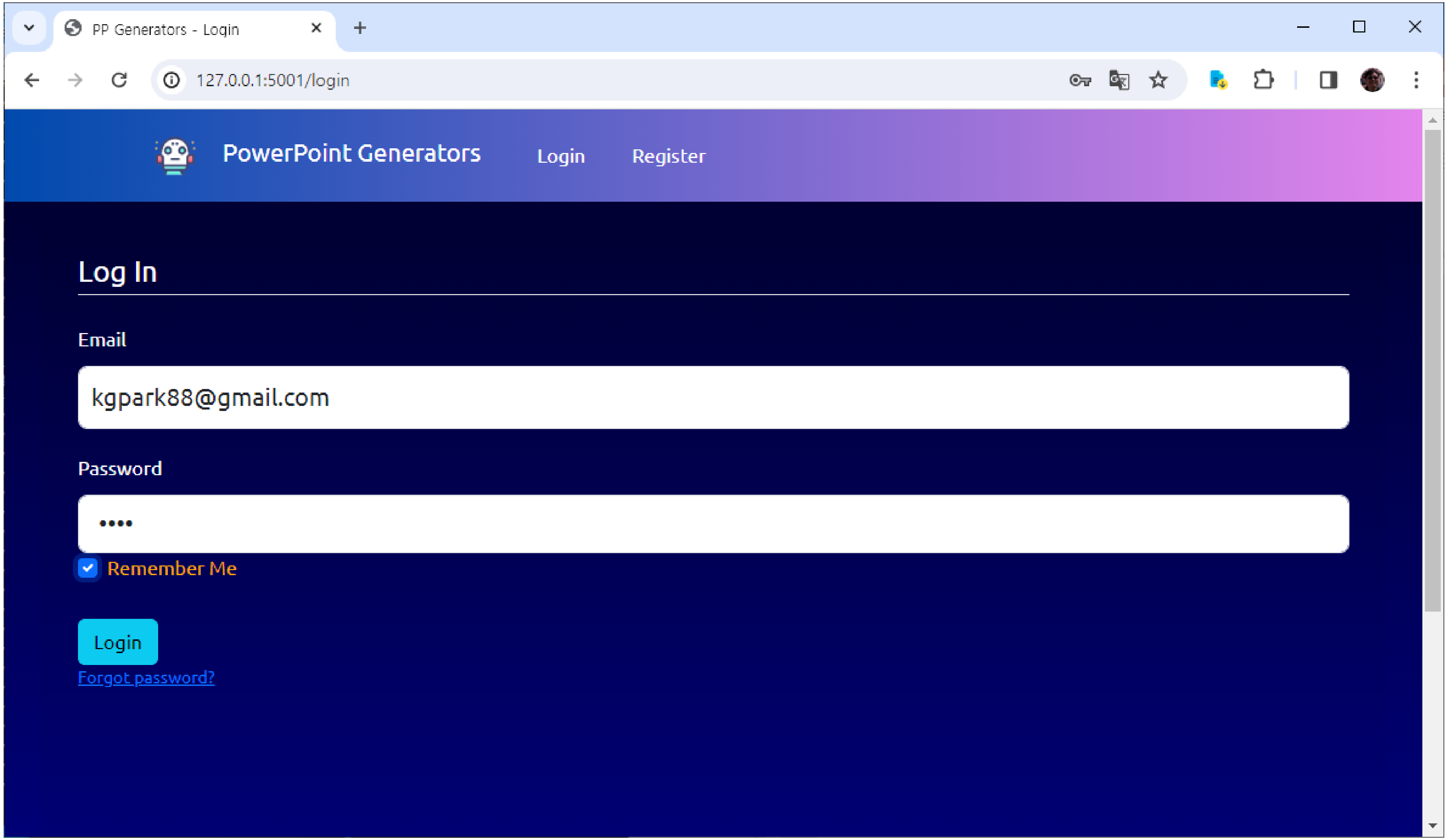
Password

••••

Confirm Password

••••

# PPT 생성앱 – 로그인 화면



## Log In

Email

kgpark88@gmail.com

Password

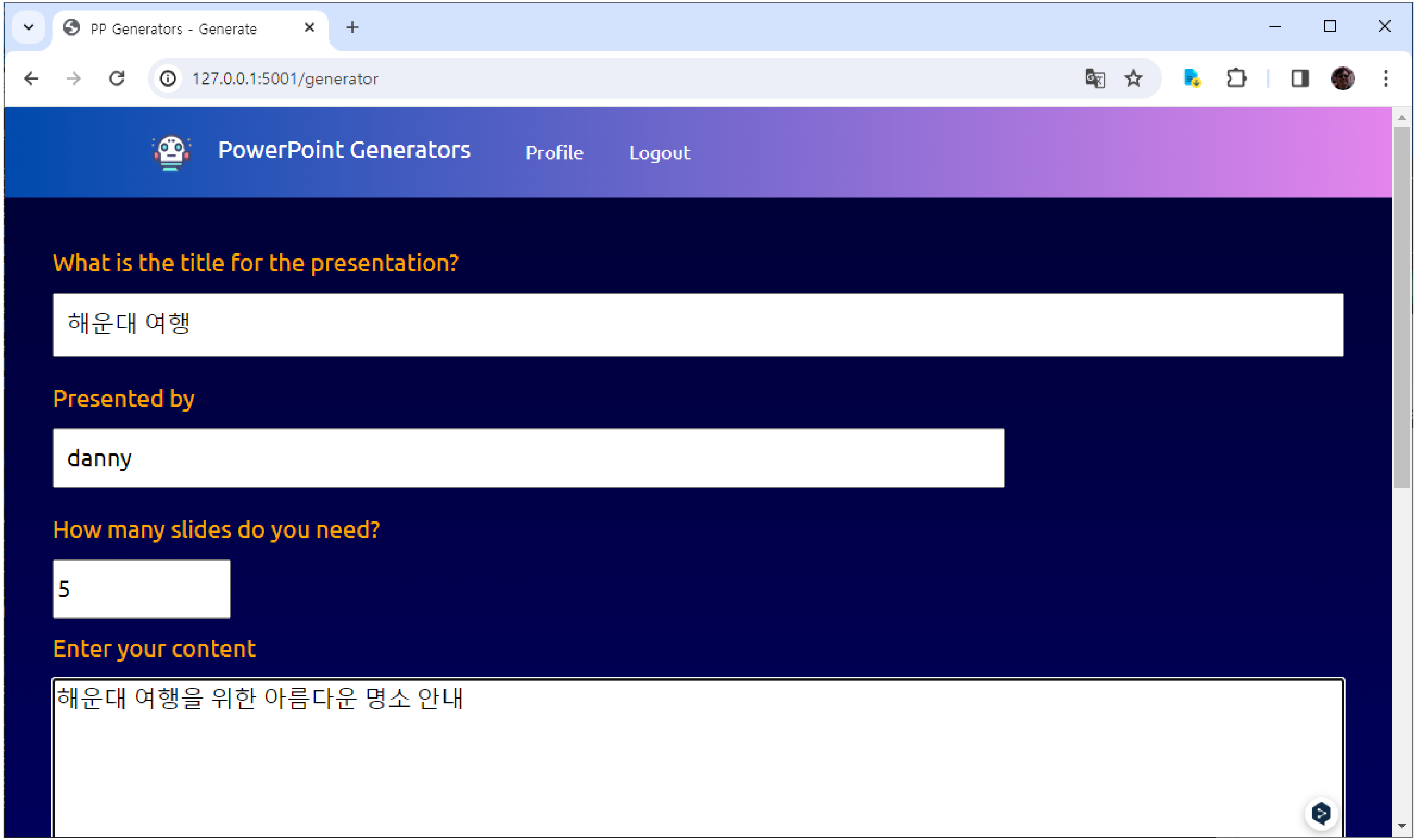
••••

☑ Remember Me

Login

Forgot password?

# PPT 생성앱 – 파워포인트 생성



What is the title for the presentation?

해운대 여행

Presented by

danny

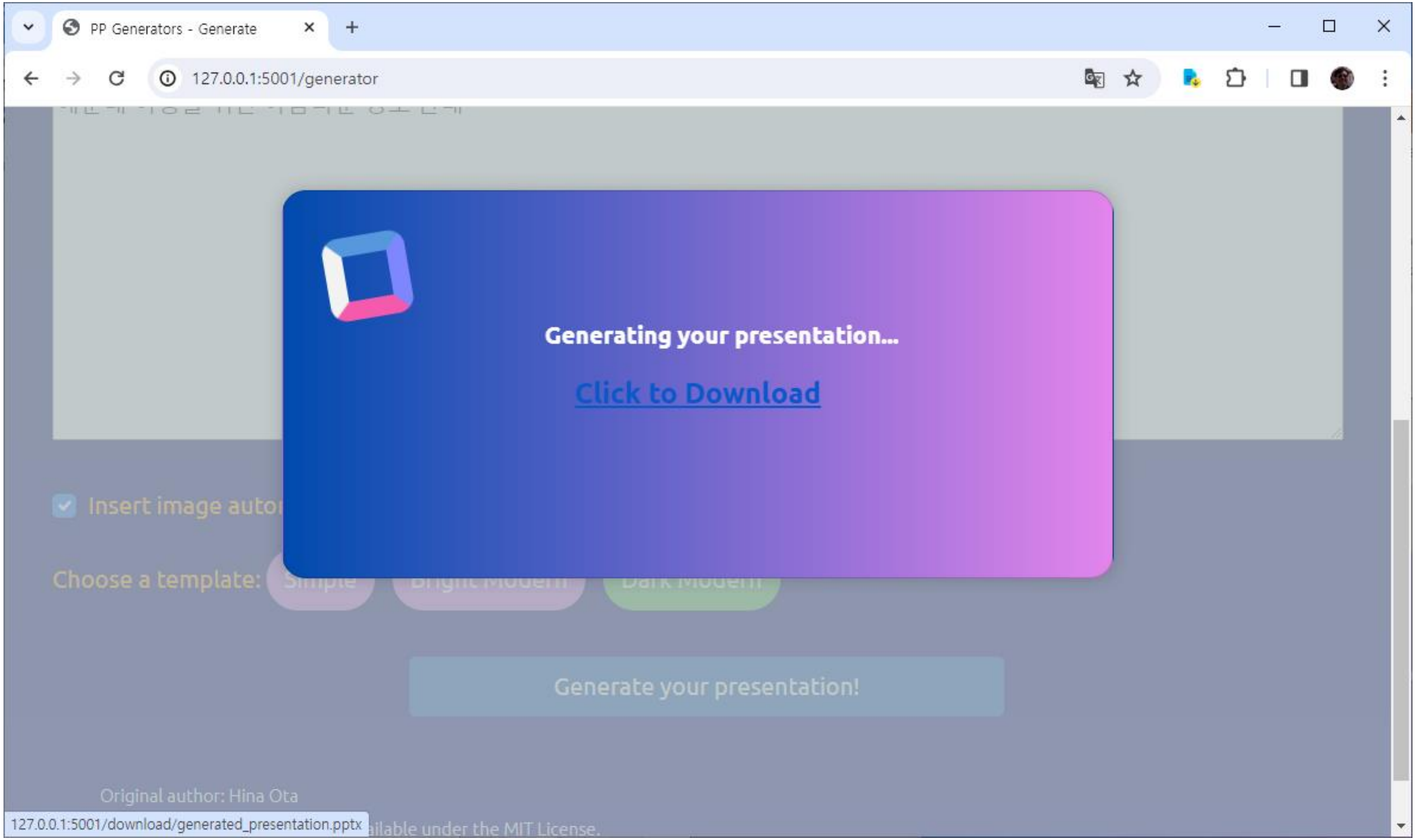
How many slides do you need?

5

Enter your content

해운대 여행을 위한 아름다운 명소 안내

# PPT 생성앱 - 다운로드



PP Generators - Generate
127.0.0.1:5001/generator

Generating your presentation...

Click to Download

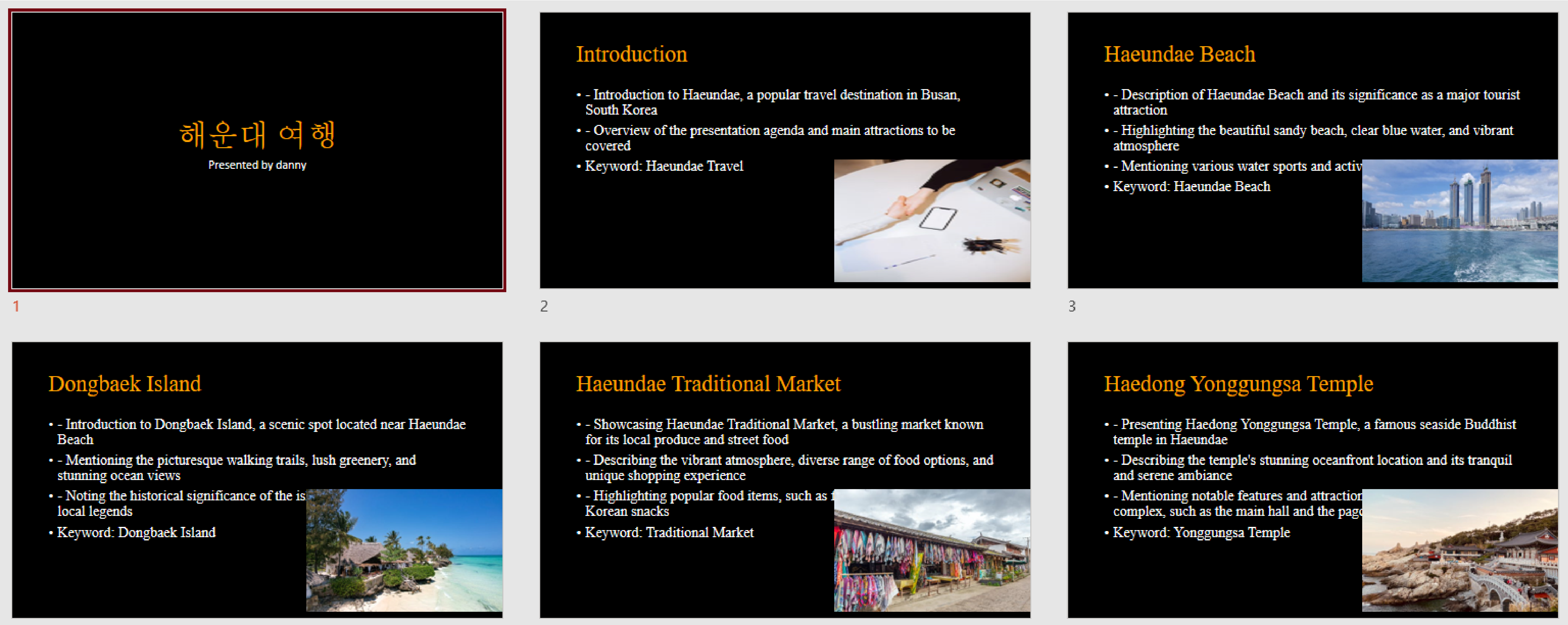
☑ Insert image auto
Choose a template: Simple Bright Modern Dark Modern
Generate your presentation!

Original author: Hina Ota

127.0.0.1:5001/download/generated_presentation.pptx

ailable under the MIT License.

# PPT 생성앱 – 결과



1

해운대 여행

Presented by danny

2

## Introduction

- - Introduction to Haeundae, a popular travel destination in Busan, South Korea
- - Overview of the presentation agenda and main attractions to be covered
- Keyword: Haeundae Travel

3

## Haeundae Beach

- - Description of Haeundae Beach and its significance as a major tourist attraction
- - Highlighting the beautiful sandy beach, clear blue water, and vibrant atmosphere
- - Mentioning various water sports and activ
- Keyword: Haeundae Beach

## Dongbaek Island

- - Introduction to Dongbaek Island, a scenic spot located near Haeundae Beach
- - Mentioning the picturesque walking trails, lush greenery, and stunning ocean views
- - Noting the historical significance of the is local legends
- Keyword: Dongbaek Island

## Haeundae Traditional Market

- - Showcasing Haeundae Traditional Market, a bustling market known for its local produce and street food
- - Describing the vibrant atmosphere, diverse range of food options, and unique shopping experience
- - Highlighting popular food items, such as f Korean snacks
- Keyword: Traditional Market

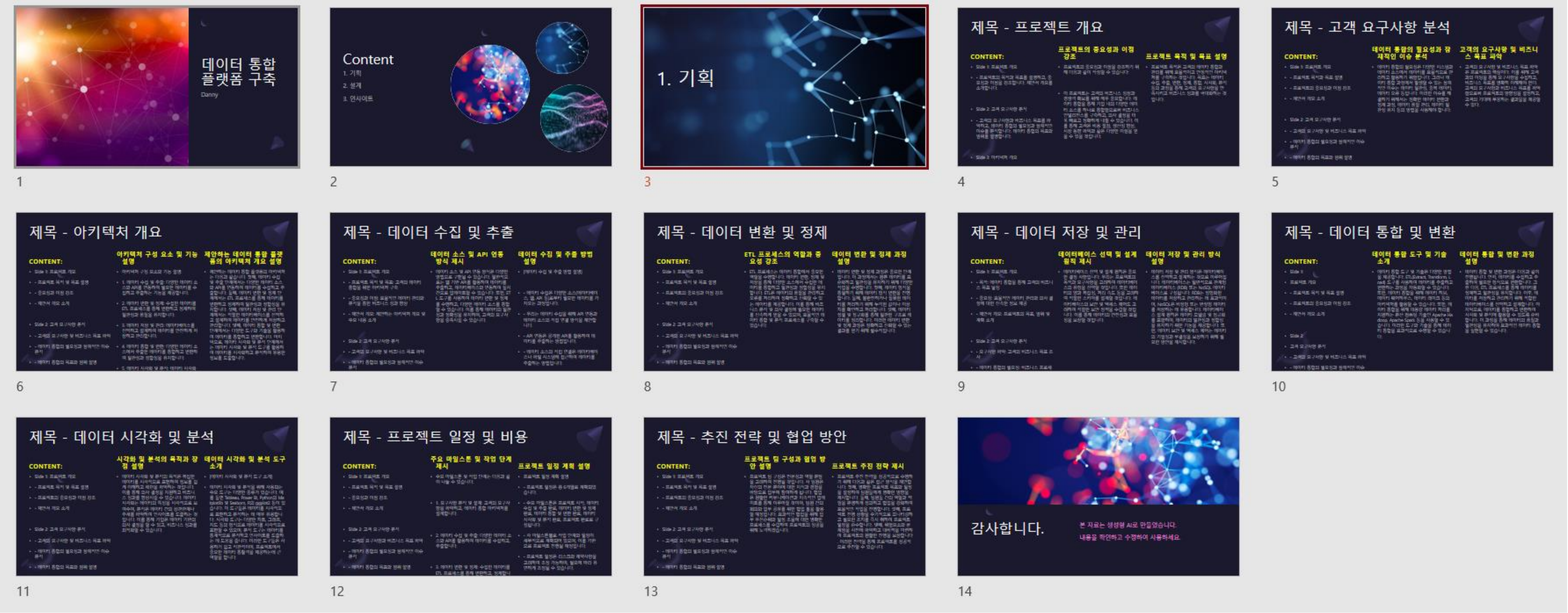
## Haedong Yonggungsa Temple

- - Presenting Haedong Yonggungsa Temple, a famous seaside Buddhist temple in Haeundae
- - Describing the temple's stunning oceanfront location and its tranquil and serene ambiance
- - Mentioning notable features and attractior complex, such as the main hall and the pag
- Keyword: Yonggungsa Temple

# 실습 – 기획/제안서 자동 생성

auto-generator.ipynb

# 실습 – 기획/제안서 자동 생성



1

데이터 통합 플랫폼 구축

Danny

2

Content

1. 기획
2. 설계
3. 인사이트

3

1. 기획

4

제목 - 프로젝트 개요

CONTENT:

프로젝트의 중요성과 이점 강조

프로젝트 목적 및 목표 설명

5

제목 - 고객 요구사항 분석

CONTENT:

데이터 통합의 필요성과 잠재적인 이슈 분석

고객의 요구사항 및 비즈니스 목표 파악

6

제목 - 아키텍처 개요

CONTENT:

아키텍처 구성 요소 및 기능 설명

제안하는 데이터 통합 플랫폼의 아키텍처 개요 설명

7

제목 - 데이터 수집 및 추출

CONTENT:

데이터 소스 및 API 연동 방식 제시

데이터 수집 및 추출 방법 설명

8

제목 - 데이터 변환 및 정제

CONTENT:

ETL 프로세스의 역할과 중요성 강조

데이터 변환 및 정제 과정 설명

9

제목 - 데이터 저장 및 관리

CONTENT:

데이터베이스 선택 및 설계 원칙 제시

데이터 저장 및 관리 방식 설명

10

제목 - 데이터 통합 및 변환

CONTENT:

데이터 통합 도구 및 기술 소개

데이터 통합 및 변환 과정 설명

11

제목 - 데이터 시각화 및 분석

CONTENT:

시각화 및 분석의 목적과 장점 설명

데이터 시각화 및 분석 도구 소개

12

제목 - 프로젝트 일정 및 비용

CONTENT:

주요 마일스톤 및 작업 단계 제시

프로젝트 일정 계획 설명

13

제목 - 추진 전략 및 협업 방안

CONTENT:

프로젝트 팀 구성과 협업 방안 설명

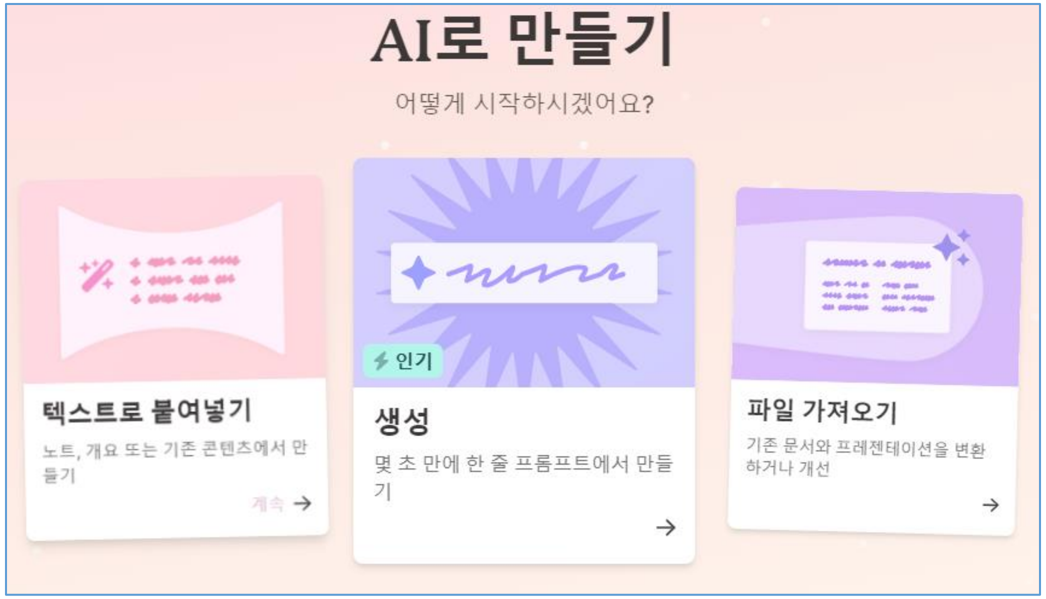
프로젝트 추진 전략 제시

14

감사합니다.

본 자료는 생성형 AI로 만들었습니다.
내용을 확인하고 수정하여 사용하세요.

# Gamma https://gamma.app/



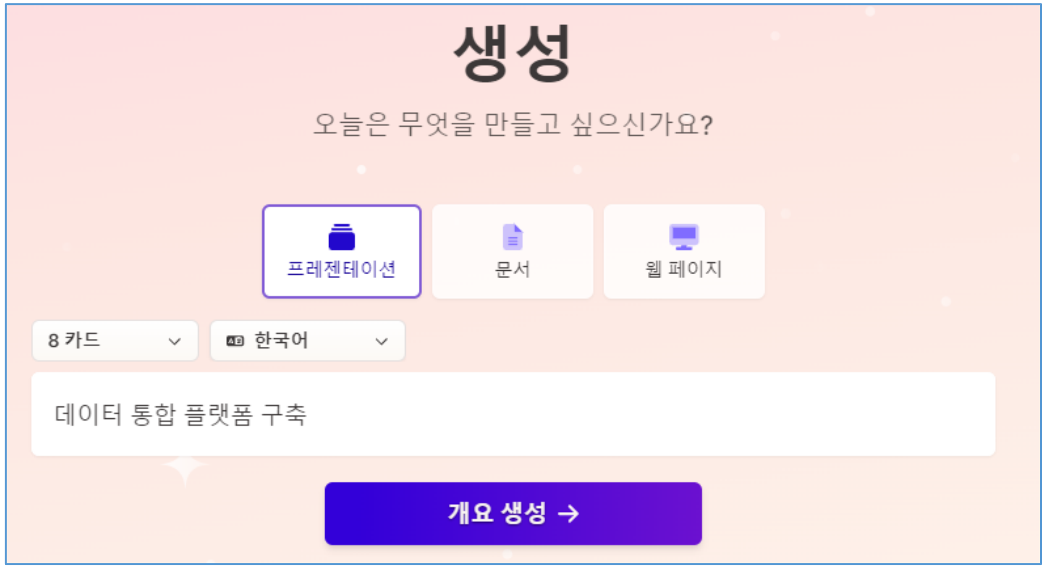
## AI로 만들기

어떻게 시작하시겠어요?

**텍스트로 붙여넣기**
노트, 개요 또는 기존 콘텐츠에서 만들기
계속 →

인기
**생성**
몇 초 만에 한 줄 프롬프트에서 만들기 →

**파일 가져오기**
기존 문서와 프레젠테이션을 변환하거나 개선 →



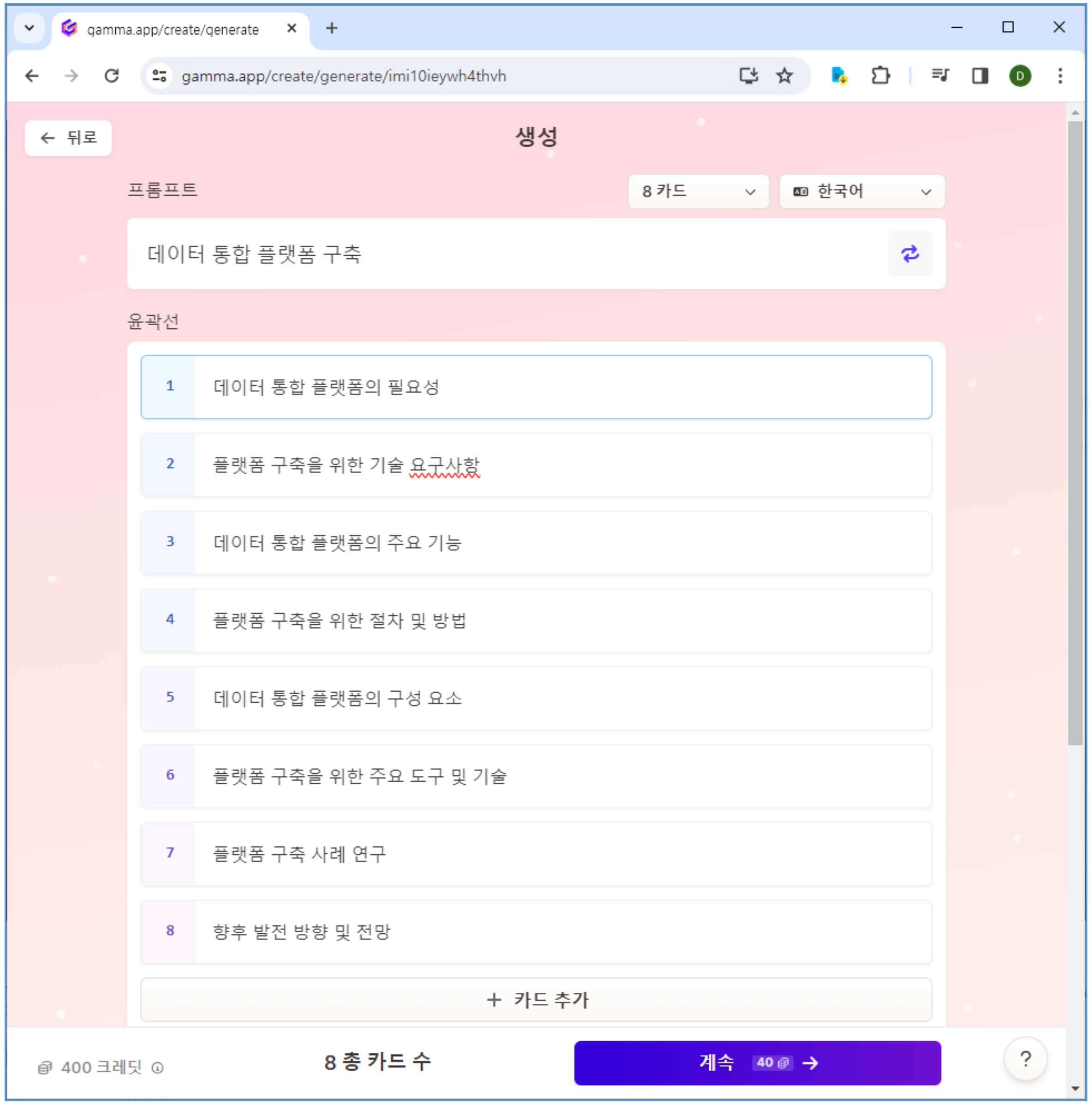
## 생성

오늘은 무엇을 만들고 싶으신가요?

8 카드
한국어

데이터 통합 플랫폼 구축

개요 생성 →



gamma.app/create/generate/imi10ieywh4thvh

생성

프롬프트

8 카드
한국어

데이터 통합 플랫폼 구축

윤곽선

1 데이터 통합 플랫폼의 필요성
2 플랫폼 구축을 위한 기술 요구사항
3 데이터 통합 플랫폼의 주요 기능
4 플랫폼 구축을 위한 절차 및 방법
5 데이터 통합 플랫폼의 구성 요소
6 플랫폼 구축을 위한 주요 도구 및 기술
7 플랫폼 구축 사례 연구
8 향후 발전 방향 및 전망

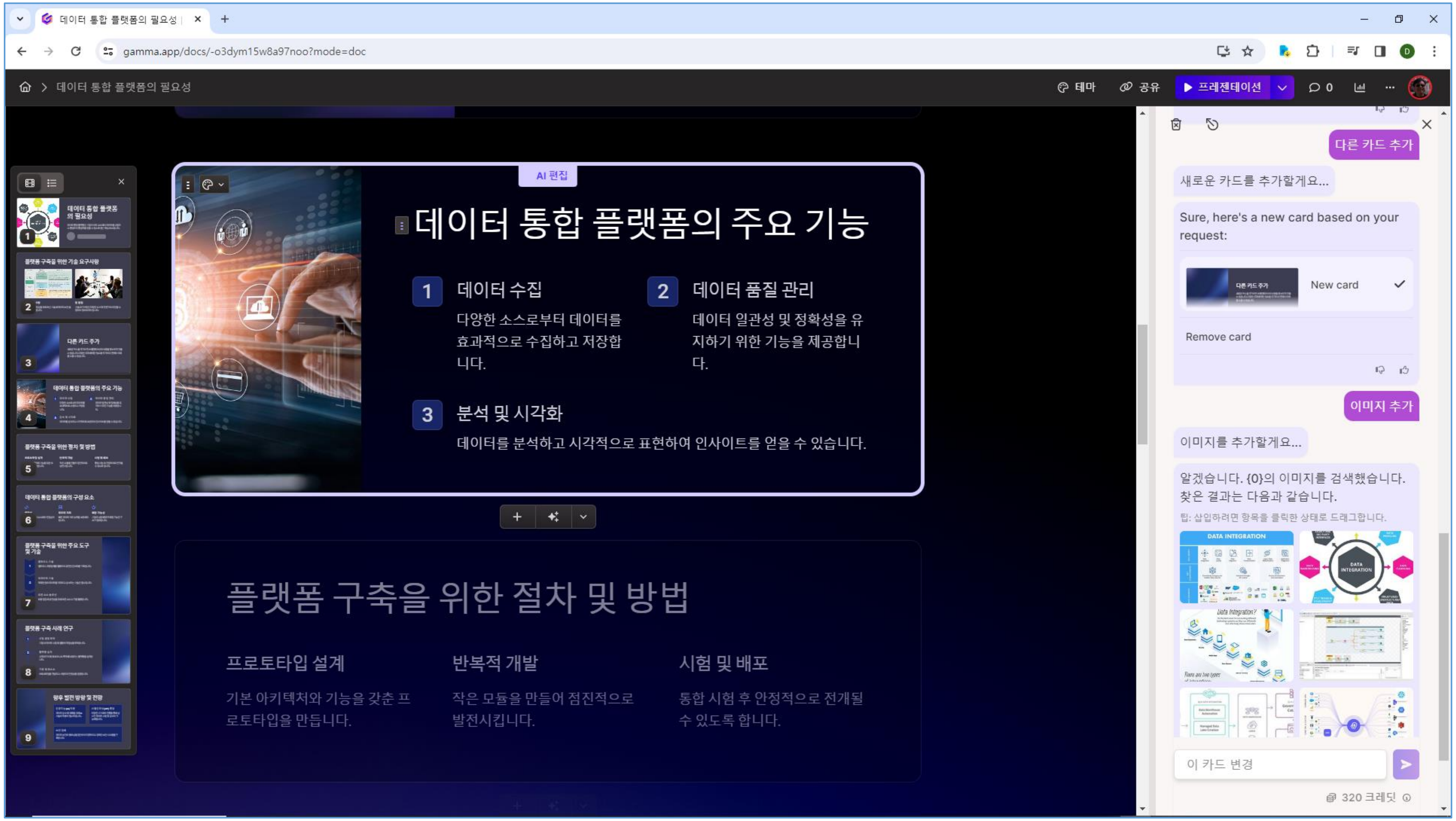
+ 카드 추가

400 크레딧

8 총 카드 수

계속 40 →

Gamma https://gamma.app/



데이터 통합 플랫폼의 필요성
gamma.app/docs/-o3dym15w8a97noo?mode=doc
데이터 통합 플랫폼의 필요성
테마
공유
프레젠테이션
AI 편집

# 데이터 통합 플랫폼의 주요 기능

1 데이터 수집

다양한 소스로부터 데이터를 효과적으로 수집하고 저장합니다.

2 데이터 품질 관리

데이터 일관성 및 정확성을 유지하기 위한 기능을 제공합니다.

3 분석 및 시각화

데이터를 분석하고 시각적으로 표현하여 인사이트를 얻을 수 있습니다.

# 플랫폼 구축을 위한 절차 및 방법

프로토타입 설계

기본 아키텍처와 기능을 갖춘 프로토타입을 만듭니다.

반복적 개발

작은 모듈을 만들어 점진적으로 발전시킵니다.

시험 및 배포

통합 시험 후 안정적으로 전개될 수 있도록 합니다.

다른 카드 추가

새로운 카드를 추가할게요...

Sure, here's a new card based on your request:

New card ✓

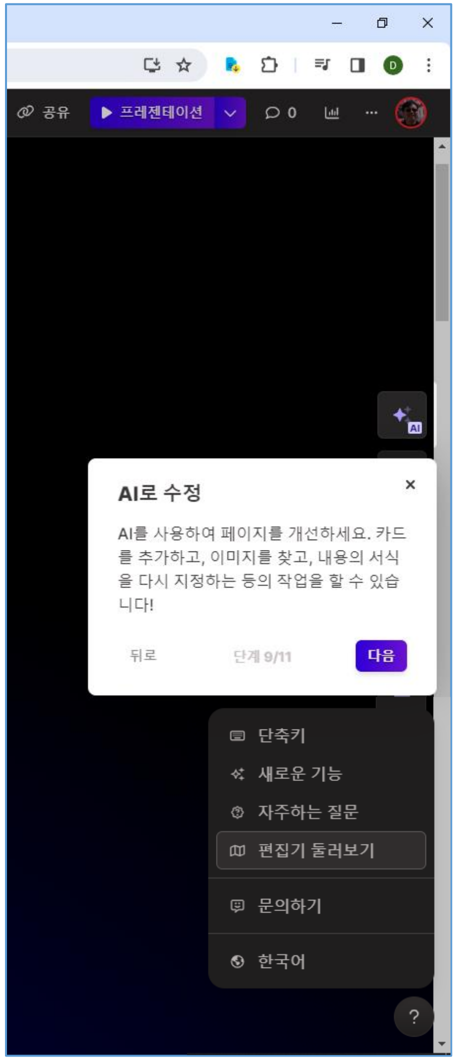
Remove card

이미지 추가

이미지를 추가할게요...

알겠습니다. {0}의 이미지를 검색했습니다. 찾은 결과는 다음과 같습니다.

팁: 삽입하려면 항목을 클릭한 상태로 드래그합니다.



AI로 수정

AI를 사용하여 페이지를 개선하세요. 카드를 추가하고, 이미지를 찾고, 내용의 서식을 다시 지정하는 등의 작업을 할 수 있습니다!

뒤로 단계 9/11 다음

# 실습 - 유튜브 요약

I'M SO EXCITED TO BE HERE.
THANK YOU.
I SPENT THE LAST YEAR SHOOTING A

somehow five years later
covet hits

YouTube Loader.ipynb

# 실습 - LangServe

LangServe는 FastAPI와 통합되어 있으며 LangChain Runnable과 Chain을 REST API로 쉽게 배포할 수 있습니다.

## 🦜🏓 LangServe

- 입력 및 출력 자동 추론
- 오류 메시지 표시
- 엔드포인트 제공

  /invoke

  /batch

  /stream

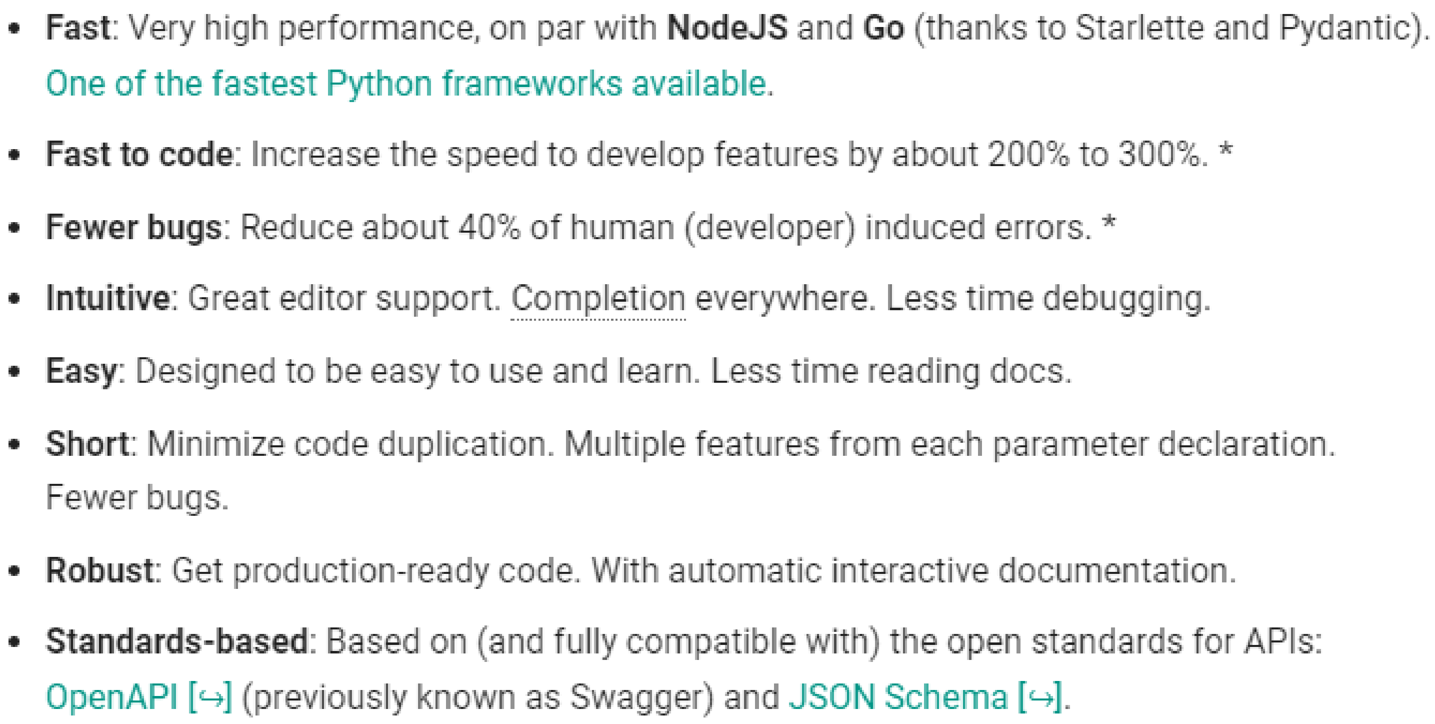
  /stream_log

  /playground
- LangSmith 추적 기능 내장

https://python.langchain.com/docs/langserve

## FastAPI

*FastAPI framework, high performance, easy to learn, fast to code, ready for production*



- **Fast**: Very high performance, on par with **NodeJS** and **Go** (thanks to Starlette and Pydantic). One of the fastest Python frameworks available.
- **Fast to code**: Increase the speed to develop features by about 200% to 300%. *
- **Fewer bugs**: Reduce about 40% of human (developer) induced errors. *
- **Intuitive**: Great editor support. Completion everywhere. Less time debugging.
- **Easy**: Designed to be easy to use and learn. Less time reading docs.
- **Short**: Minimize code duplication. Multiple features from each parameter declaration. Fewer bugs.
- **Robust**: Get production-ready code. With automatic interactive documentation.
- **Standards-based**: Based on (and fully compatible with) the open standards for APIs: OpenAPI [↪] (previously known as Swagger) and JSON Schema [↪].

https://fastapi.tiangolo.com/

# 실습 - LangServe

my-app

### 1. Python 3.11.8 버전 설치

설치 파일 다운로드 : https://www.python.org/downloads/

### 2. Python 가상환경 설치

```
python -m venv venv
venv₩Scripts₩activate
```

### 3. LangChain 패키지 설치

```
pip install langchain langchain-openai langchain-cli
```

```
pip list
```

```
Package              Version
-------------------- ----------
fastapi              0.109.2
langchain            0.1.7
langchain-cli        0.0.21
langchain-community  0.0.20
langchain-core       0.1.23
langchain-openai     0.0.6
langserve            0.0.41
langsmith            0.0.87
openai               1.12.0
pydantic             2.6.1
pydantic_core        2.16.2
```

### 4. 환경변수에 API KEY 추가

```
# Windows
setx OPENAI_API_KEY=sk-...
setx LANGCHAIN_TRACING_V2=true
setx LANGCHAIN_API_KEY=<your-api-key>
setx LANGCHAIN_PROJECT=<your-project>  # if not specified, defaults to "default"

# macOS/Linux
export OPENAI_API_KEY=sk-...
export LANGCHAIN_TRACING_V2=true
export LANGCHAIN_API_KEY=<your-api-key>
export LANGCHAIN_PROJECT=<your-project>  # if not specified, defaults to "default
```

### 5. LangChain 프로젝트 생성

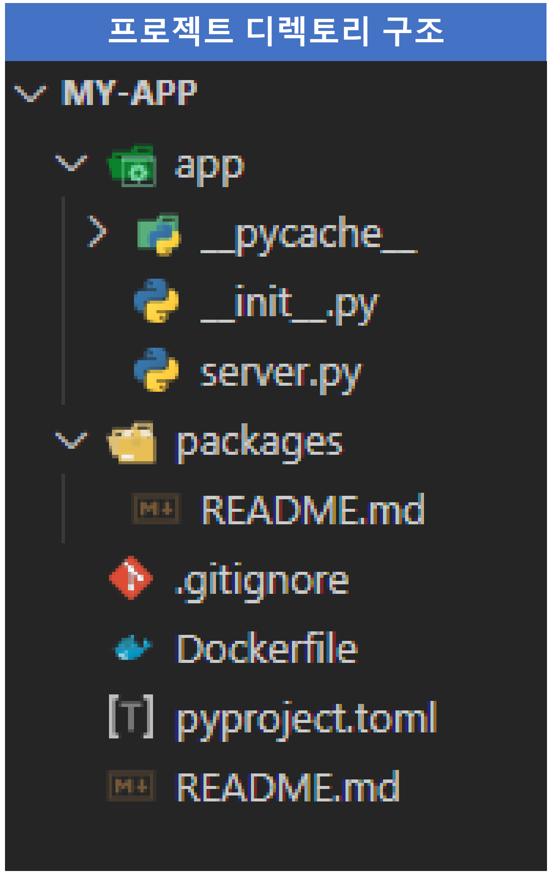
```
langchain app new my-app
```

# 실습 - LangServe

🙌 my-app

### 6. 프로그램 개발

my-app/app/server.py 파일 수정



프로젝트 디렉토리 구조

```
∨ MY-APP
  ∨ app
    > __pycache__
      __init__.py
      server.py
  ∨ packages
      README.md
    .gitignore
    Dockerfile
    pyproject.toml
    README.md
```

```python
from fastapi import FastAPI
from fastapi.responses import RedirectResponse
from langserve import add_routes
from langchain_openai import ChatOpenAI
from langchain_core.prompts import ChatPromptTemplate

app = FastAPI()

@app.get("/")
async def redirect_root_to_docs():
    return RedirectResponse("/docs")

prompt = ChatPromptTemplate.from_messages(
    [
        (
            "system",
            "사용자 입력을 해적 말투로 번역",
        ),
        ("human", "{text}"),
    ]
)
model = ChatOpenAI()
chain = prompt | model

add_routes(app, chain, path="/pirate-speak")


if __name__ == "__main__":
    import uvicorn

    uvicorn.run(app, host="0.0.0.0", port=8000)
```

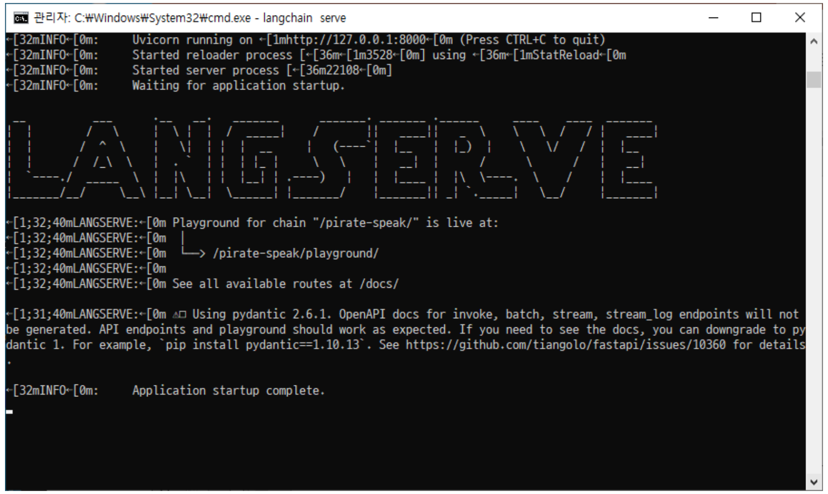
# 실습 - LangServe

my-app

### 7. 프로그램 실행

cd my-app

langchain serve

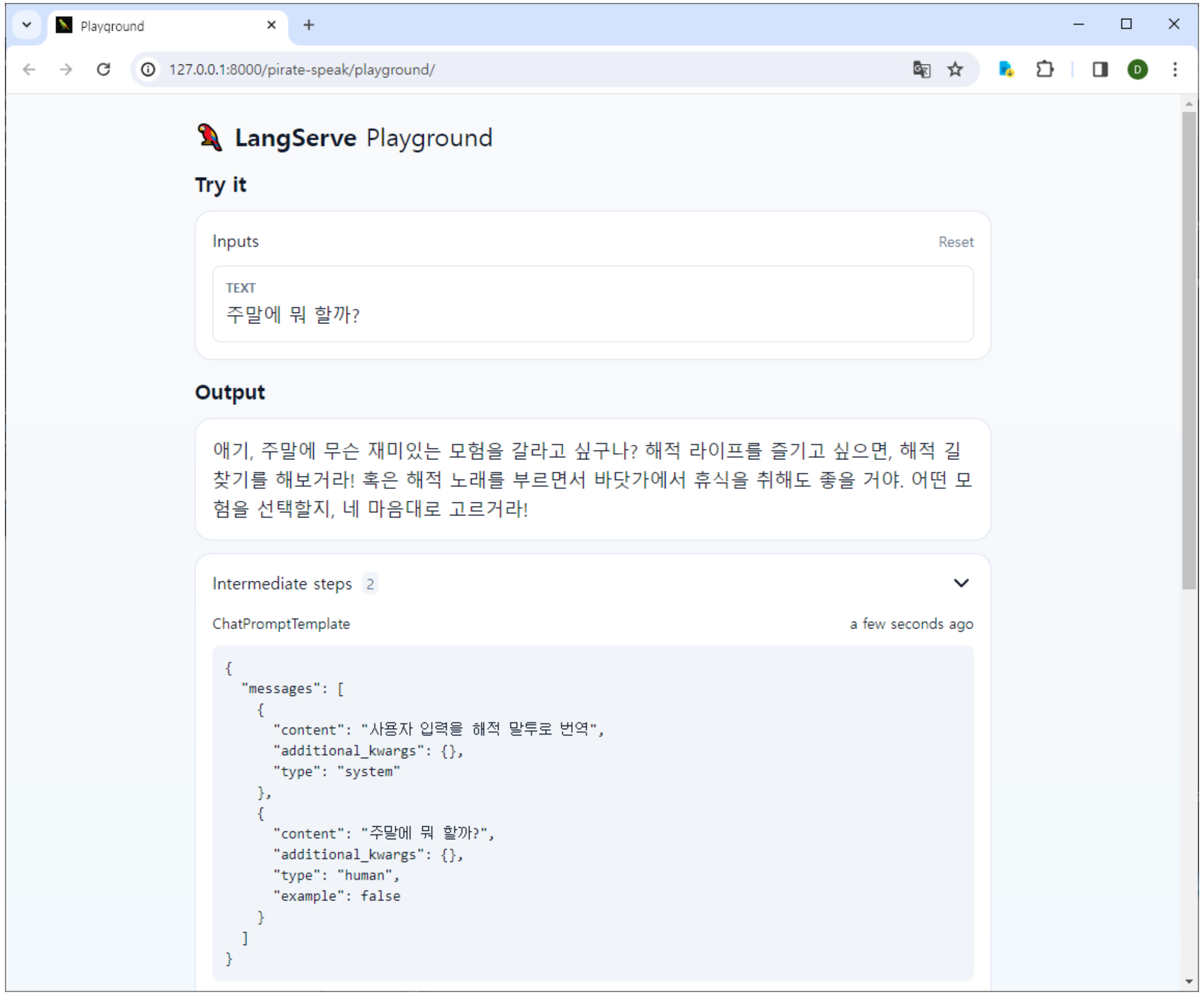


관리자: C:₩Windows₩System32₩cmd.exe - langchain serve

```
←[32mINFO←[0m:     Uvicorn running on ←[1mhttp://127.0.0.1:8000←[0m (Press CTRL+C to quit)
←[32mINFO←[0m:     Started reloader process [←[36m←[1m3528←[0m] using ←[36m←[1mStatReload←[0m
←[32mINFO←[0m:     Started server process [←[36m22108←[0m]
←[32mINFO←[0m:     Waiting for application startup.

←[1;32;40mLANGSERVE:←[0m Playground for chain "/pirate-speak/" is live at:
←[1;32;40mLANGSERVE:←[0m  │
←[1;32;40mLANGSERVE:←[0m  └──> /pirate-speak/playground/
←[1;32;40mLANGSERVE:←[0m
←[1;32;40mLANGSERVE:←[0m See all available routes at /docs/

←[1;31;40mLANGSERVE:←[0m ⚠︎ Using pydantic 2.6.1. OpenAPI docs for invoke, batch, stream, stream_log endpoints will not
be generated. API endpoints and playground should work as expected. If you need to see the docs, you can downgrade to py
dantic 1. For example, `pip install pydantic==1.10.13`. See https://github.com/tiangolo/fastapi/issues/10360 for details
.

←[32mINFO←[0m:     Application startup complete.
```

### 8. Playground

http://127.0.0.1:8000/pirate-speak/playground/



LangServe Playground

**Try it**

Inputs  Reset

TEXT
주말에 뭐 할까?

**Output**

애기, 주말에 무슨 재미있는 모험을 갈라고 싶구나? 해적 라이프를 즐기고 싶으면, 해적 길찾기를 해보거라! 혹은 해적 노래를 부르면서 바닷가에서 휴식을 취해도 좋을 거야. 어떤 모험을 선택할지, 네 마음대로 고르거라!

Intermediate steps 2

ChatPromptTemplate  a few seconds ago

```
{
  "messages": [
    {
      "content": "사용자 입력을 해적 말투로 번역",
      "additional_kwargs": {},
      "type": "system"
    },
    {
      "content": "주말에 뭐 할까?",
      "additional_kwargs": {},
      "type": "human",
      "example": false
    }
  ]
}
```

# 실습 - LangServe Examples

| 설명 | 링크 |
|---|---|
| **LLMs** Minimal example that reserves OpenAI and Anthropic chat models. Uses async, supports batching and streaming. | server, client |
| **Retriever** Simple server that exposes a retriever as a runnable. | server, client |
| **Conversational Retriever** A Conversational Retriever exposed via LangServe | server, client |
| **Agent** without **conversation history** based on OpenAI tools | server, client |
| **Agent** with **conversation history** based on OpenAI tools | server, client |
| RunnableWithMessageHistory to implement chat persisted on backend, keyed off a session_id supplied by client. | server, client |
| RunnableWithMessageHistory to implement chat persisted on backend, keyed off a conversation_id supplied by client, and user_id (see Auth for implementing user_id properly). | server, client |
| Configurable Runnable to create a retriever that supports run time configuration of the index name. | server, client |
| Configurable Runnable that shows configurable fields and configurable alternatives. | server, client |
| **APIHandler** Shows how to use APIHandler instead of add_routes. This provides more flexibility for developers to define endpoints. Works well with all FastAPI patterns, but takes a bit more effort. | server |
| **LCEL Example** Example that uses LCEL to manipulate a dictionary input. | server, client |
| **Auth** with add_routes: Simple authentication that can be applied across all endpoints associated with app. (Not useful on its own for implementing per user logic.) | server |
| **Auth** with add_routes: Simple authentication mechanism based on path dependencies. (No useful on its own for implementing per user logic.) | server |
| **Auth** with add_routes: Implement per user logic and auth for endpoints that use per request config modifier. (**Note**: At the moment, does not integrate with OpenAPI docs.) | server, client |
| **Auth** with APIHandler: Implement per user logic and auth that shows how to search only within user owned documents. | server, client |
| **Widgets** Different widgets that can be used with playground (file upload and chat) | server |
| **Widgets** File upload widget used for LangServe playground. | server, client |

https://python.langchain.com/docs/langserve#examples

# 실습 – 로컬 환경에 sLLM 사용

- **sLLM(smaller Large Language Model , 소형 언어 모델)**

- LLM과 비교했을 때 매개변수의 수가 수십 억~수백 억개로 비교적 크기가 작은 언어모델
- 비용절감, 보안, 특정 도메인에 활용 목적으로 사용
- 특정 도메인 사용용도로 sLLM 을 사용하는 경우가 많아지고 있음

- **Ollama 설치 및 sLLM 사용 방법**

1. Ollama 설치
- 로컬 환경에서 다양한 언어 모델을 실행할 수 있게 지원하는 오픈소스
- 모델 종류 : https://ollama.com/library
- 설치 파일 다운로드 : https://ollama.ai/

2. sLLM 모델 실행

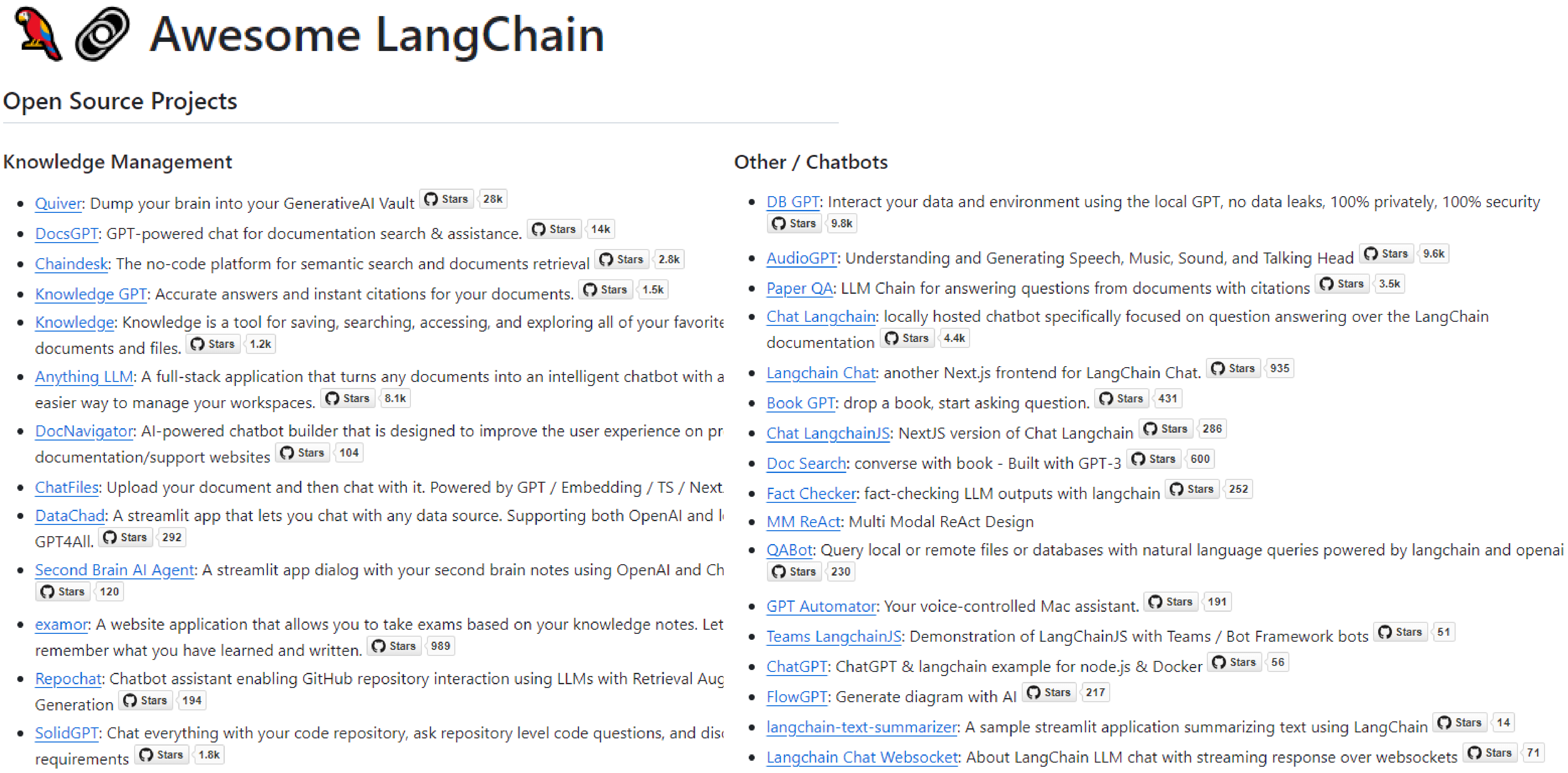
```
ollama run llama2
```

3. 애플리케이션 개발

```
from langchain_community.llms
import Ollama

llm = Ollama(model="llama2")
llm.invoke("Hello")
```

- **Llama 2 (Large Language Model Meta AI)**

- 메타에서 공개한 상업적으로도 이용 가능한 오픈 소스 sLLM



# 🦜🔗 Awesome LangChain

## Open Source Projects

### Knowledge Management

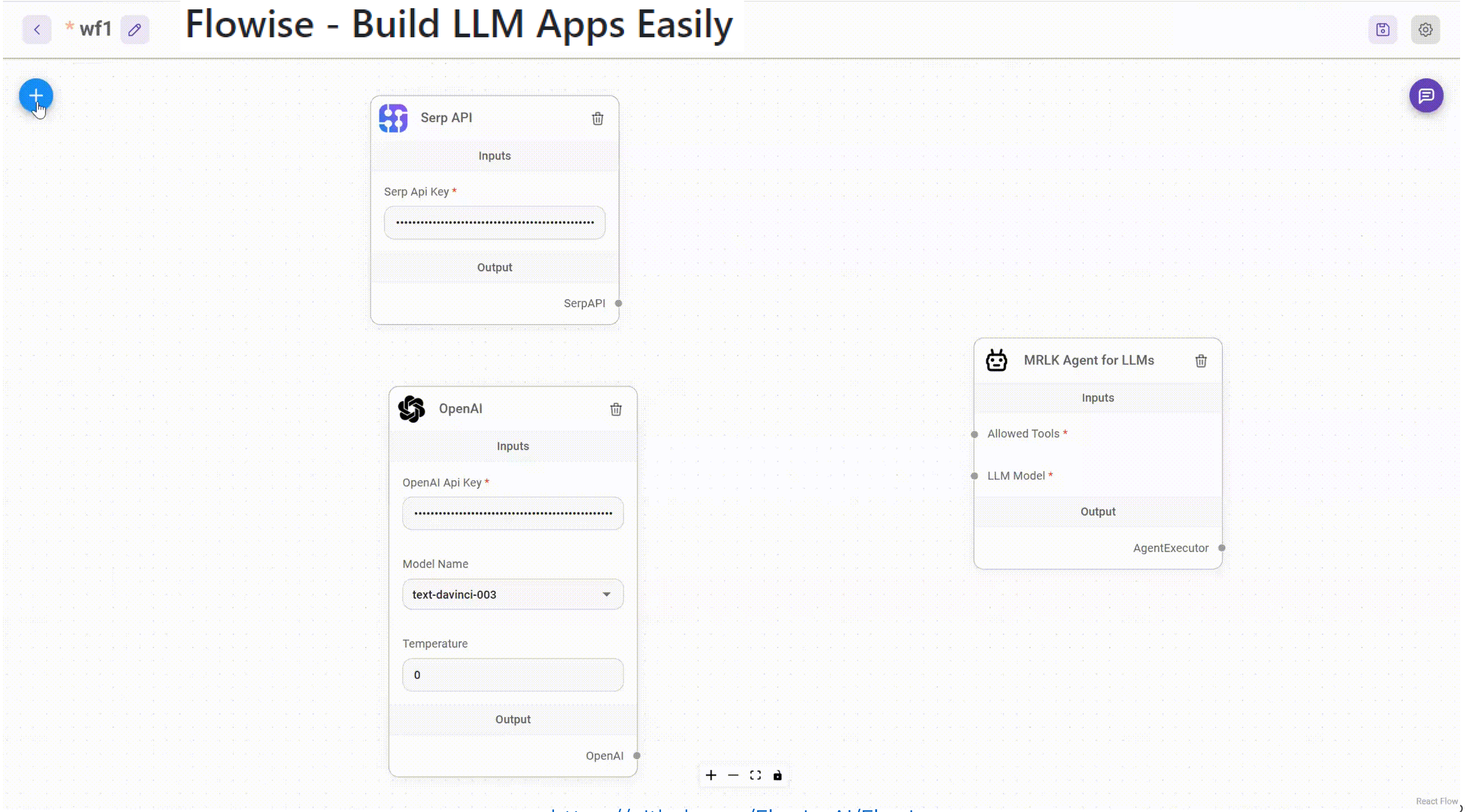
- Quiver: Dump your brain into your GenerativeAI Vault Stars 28k
- DocsGPT: GPT-powered chat for documentation search & assistance. Stars 14k
- Chaindesk: The no-code platform for semantic search and documents retrieval Stars 2.8k
- Knowledge GPT: Accurate answers and instant citations for your documents. Stars 1.5k
- Knowledge: Knowledge is a tool for saving, searching, accessing, and exploring all of your favorite documents and files. Stars 1.2k
- Anything LLM: A full-stack application that turns any documents into an intelligent chatbot with a easier way to manage your workspaces. Stars 8.1k
- DocNavigator: AI-powered chatbot builder that is designed to improve the user experience on pr documentation/support websites Stars 104
- ChatFiles: Upload your document and then chat with it. Powered by GPT / Embedding / TS / Next.
- DataChad: A streamlit app that lets you chat with any data source. Supporting both OpenAI and l GPT4All. Stars 292
- Second Brain AI Agent: A streamlit app dialog with your second brain notes using OpenAI and Ch Stars 120
- examor: A website application that allows you to take exams based on your knowledge notes. Let remember what you have learned and written. Stars 989
- Repochat: Chatbot assistant enabling GitHub repository interaction using LLMs with Retrieval Aug Generation Stars 194
- SolidGPT: Chat everything with your code repository, ask repository level code questions, and dis requirements Stars 1.8k

### Other / Chatbots

- DB GPT: Interact your data and environment using the local GPT, no data leaks, 100% privately, 100% security Stars 9.8k
- AudioGPT: Understanding and Generating Speech, Music, Sound, and Talking Head Stars 9.6k
- Paper QA: LLM Chain for answering questions from documents with citations Stars 3.5k
- Chat Langchain: locally hosted chatbot specifically focused on question answering over the LangChain documentation Stars 4.4k
- Langchain Chat: another Next.js frontend for LangChain Chat. Stars 935
- Book GPT: drop a book, start asking question. Stars 431
- Chat LangchainJS: NextJS version of Chat Langchain Stars 286
- Doc Search: converse with book - Built with GPT-3 Stars 600
- Fact Checker: fact-checking LLM outputs with langchain Stars 252
- MM ReAct: Multi Modal ReAct Design
- QABot: Query local or remote files or databases with natural language queries powered by langchain and openai Stars 230
- GPT Automator: Your voice-controlled Mac assistant. Stars 191
- Teams LangchainJS: Demonstration of LangChainJS with Teams / Bot Framework bots Stars 51
- ChatGPT: ChatGPT & langchain example for node.js & Docker Stars 56
- FlowGPT: Generate diagram with AI Stars 217
- langchain-text-summarizer: A sample streamlit application summarizing text using LangChain Stars 14
- Langchain Chat Websocket: About LangChain LLM chat with streaming response over websockets Stars 71

https://github.com/kyrolabs/awesome-langchain

# Flowise - Build LLM Apps Easily



Serp API

Inputs

Serp Api Key *

••••••••••••••••••••••••••••••••••••••••••••••

Output

SerpAPI

OpenAI

Inputs

OpenAI Api Key *

••••••••••••••••••••••••••••••••••••••••••••••

Model Name

text-davinci-003

Temperature

0

Output

OpenAI

MRLK Agent for LLMs

Inputs

Allowed Tools *

LLM Model *

Output

AgentExecutor

https://github.com/FlowiseAI/Flowise

AgentGPT

Assemble, configure, and deploy autonomous AI Agents in your browser.

https://github.com/reworkd/AgentGPT

# Gradio Tools

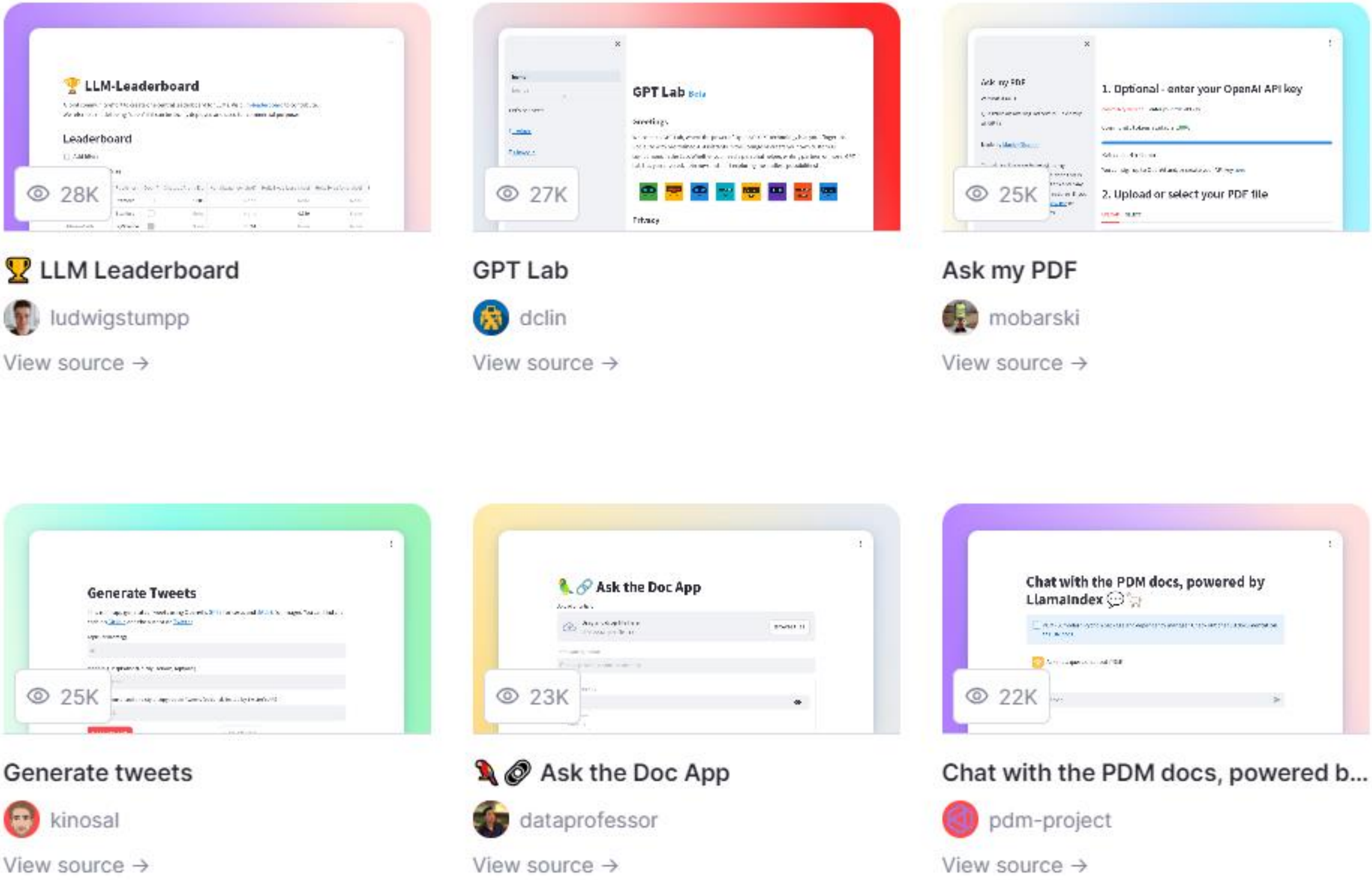
## Gradio 🤝 LLM Agents

https://github.com/FlowiseAI/Flowise

# Streamlit

**A faster way to build and share data apps**

Streamlit turns data scripts into shareable web apps in minutes.
All in pure Python. No front-end experience required.



LLM-Leaderboard
Leaderboard
GPT Lab Beta
1. Optional - enter your OpenAI API key
2. Upload or select your PDF file
Generate Tweets
Ask the Doc App
Chat with the PDM docs, powered by LlamaIndex

👁 28K
🏆 LLM Leaderboard
ludwigstumpp
View source →

👁 27K
GPT Lab
dclin
View source →

👁 25K
Ask my PDF
mobarski
View source →

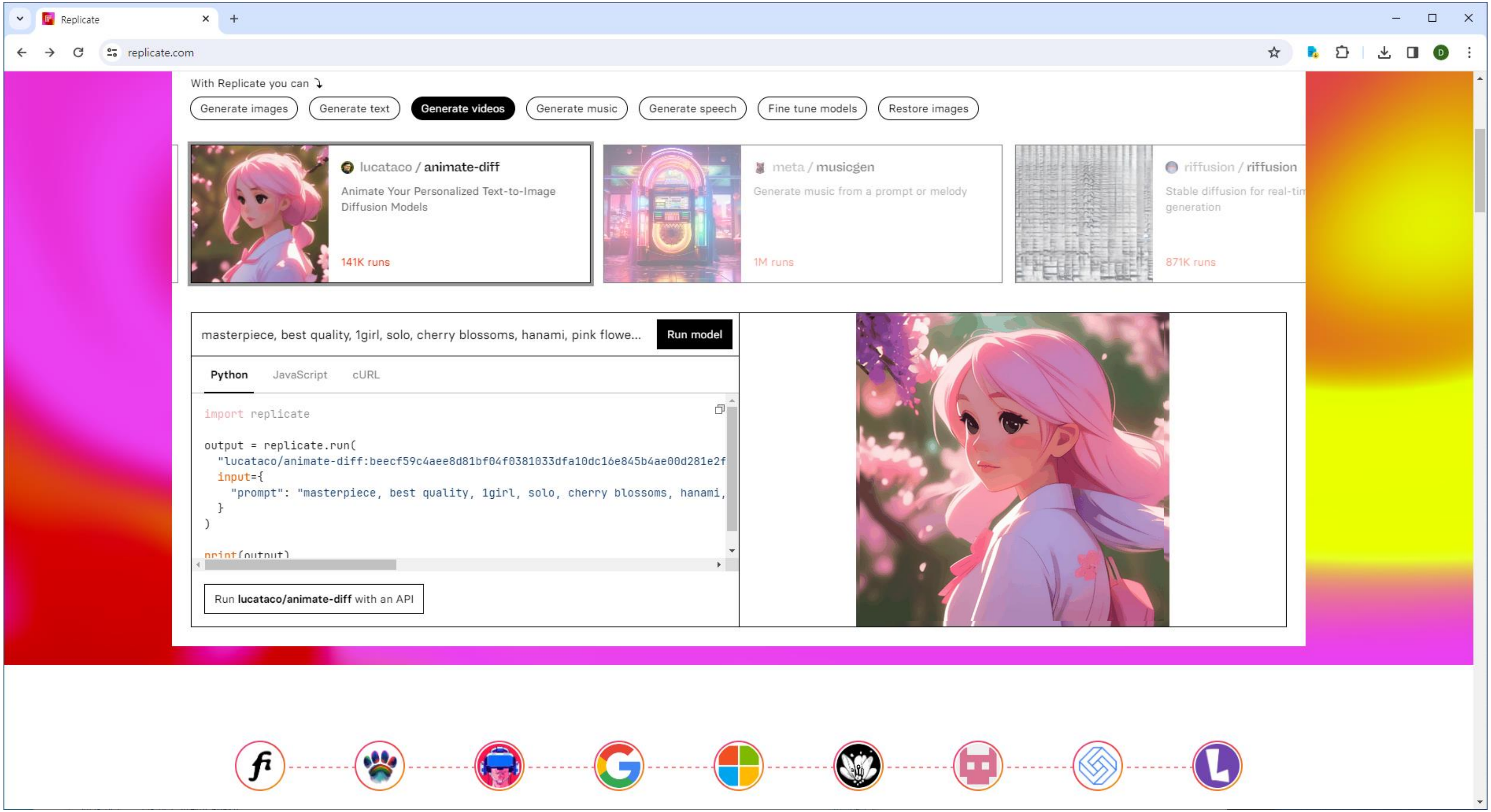
👁 25K
Generate tweets
kinosal
View source →

👁 23K
🦜🔗 Ask the Doc App
dataprofessor
View source →

👁 22K
Chat with the PDM docs, powered b...
pdm-project
View source →

https://streamlit.io/gallery?category=llms replicate **Run AI with an API.** Run and fine-tune open-source models. Deploy custom models at scale. All with one line of code.

Explore



With Replicate you can

lucataco / animate-diff
Animate Your Personalized Text-to-Image Diffusion Models
141K runs

meta / musicgen
Generate music from a prompt or melody
1M runs

riffusion / riffusion
Stable diffusion for real-tim generation
871K runs

masterpiece, best quality, 1girl, solo, cherry blossoms, hanami, pink flowe...

```
import replicate

output = replicate.run(
  "lucataco/animate-diff:beecf59c4aee8d81bf04f0381033dfa10dc16e845b4ae00d281e2f
  input={
    "prompt": "masterpiece, best quality, 1girl, solo, cherry blossoms, hanami,
  }
)

print(output)
```

Run **lucataco/animate-diff** with an API

https://replicate.com/

# Thank you

kgpark88@gmail.com