```
!pip install openai gradio requests
```

Show hidden output

```
import openai
import requests
import json

# 👉 Your JSON Config
config = {
    "api_key": "sk-proj-Xcy2elggFsylRwTwCjm2aLZY1ZgPfHptbVn5aT0svELbGchCM6YZX5Dh_q5YTDlSukaJG-FBaUT3BlbkFJsEc9gxhvaL-x9x6xmIXXRkscm6SMPI
    "agent_role": "You are an Expert in CIVIL SERVICE EXAM PREPARATION. Your task is to GUIDE and SUPPORT candidates in preparing effec
    "agent_instructions": "Your task is to PROVIDE candidates with IN-DEPTH knowledge and strategies to EXCEL in the civil service exam
    "model": "gpt-4o-mini",
    "temperature": 0.7,
    "top_p": 0.9
}

# 👉 Set OpenAI API key
openai.api_key = config['api_key']


def get_rag_context(query):
    rag_url = "https://rag-prod.studio.lyzr.ai/retrieve"
    payload = {
        "rag_id": "67dc24f1bd8ae1a9aaa5cffd",
        "query": query,
        "top_k": 5
    }
    response = requests.post(rag_url, json=payload)
    data = response.json()
    docs = data.get("documents", [])
    context = "\n\n".join([doc.get("content", "") for doc in docs])
    return context
```

```
import openai
import requests
import json

def get_rag_context(query):
    rag_url = "https://rag-prod.studio.lyzr.ai/retrieve"
    payload = {
        "rag_id": "67dc24f1bd8ae1a9aaa5cffd",
        "query": query,
        "top_k": 5
    }
    response = requests.post(rag_url, json=payload)
    data = response.json()
    docs = data.get("documents", [])
    context = "\n\n".join([doc.get("content", "") for doc in docs])
    return context

def chat_with_educator(user_prompt):
    # Get context from RAG
    rag_context = get_rag_context(user_prompt)
    system_message = {
        "role": "system",
        "content": f"{config['agent_role']}\n\n{config['agent_instructions']}\n\nRelevant context:\n{rag_context}"
    }
    user_message = {
        "role": "user",
        "content": user_prompt
    }

    client = openai.OpenAI(api_key=config['api_key'])

    response = client.chat.completions.create(
        model=config['model'],
        messages=[system_message, user_message],
        temperature=config['temperature'],
        top_p=config['top_p']
    )

    return response.choices[0].message.content


question = "How should I start preparing for UPSC Prelims?"
answer = chat_with_educator(question)
```

```
print("🤖 Educator IAS:", answer)
```

```
---------------------------------------------------------------------------
RateLimitError                            Traceback (most recent call last)
/tmp/ipython-input-12-3363392857.py in <cell line: 0>()
      1 question = "How should I start preparing for UPSC Prelims?"
----> 2 answer = chat_with_educator(question)
      3 print("🤖 Educator IAS:", answer)

                                    4 frames
/usr/local/lib/python3.11/dist-packages/openai/_base_client.py in request(self, cast_to, options, stream, stream_cls)
   1035 
   1036                 log.debug("Re-raising status error")
-> 1037                 raise self._make_status_error_from_response(err.response) from None
   1038 
   1039             break

RateLimitError: Error code: 429 - {'error': {'message': 'You exceeded your current quota, please check your plan and billing
details. For more information on this error, read the docs: https://platform.openai.com/docs/guides/error-codes/api-errors.',
'type': 'insufficient_quota', 'param': None, 'code': 'insufficient_quota'}}
```

```
import gradio as gr

def gradio_chat(user_input):
    return chat_with_educator(user_input)

gr.Interface(
    fn=gradio_chat,
    inputs="text",
    outputs="text",
    title="Educator IAS",
    description="Ask anything about UPSC & Civil Services Preparation"
).launch(share=True)
```

```
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://26ceaa6b64afefc3a6.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working
```

Educator IAS

Ask anything about UPSC & Civil Services Preparation

user_input

output

Clear Submit Flag

Use via API · Built with Gradio · Settings