下次問AI的，請問這樣我的requirement.txt 和 read.me 該怎麼寫?

uvicorn main:app --host 0.0.0.0 --port 8000 --reload

時區為台灣

{

"content": "string",

"posted": "2025-03-15T22:45:00",

"user\_id": 0

}

Main.py(新增及瀏覽能用)  
  
from pydantic import BaseModel

from fastapi import FastAPI, HTTPException

from datetime import datetime

import pymysql

import os

from dotenv import load\_dotenv

load\_dotenv() # 加載 .env 文件中的環境變數

# 連接到 MySQL 資料庫

db = pymysql.connect(

host=os.getenv("DB\_HOST"),

user=os.getenv("DB\_USER"),

password=os.getenv("DB\_PASSWORD"),

database=os.getenv("DB\_NAME")

)

cursor = db.cursor()

app = FastAPI()

# 定義預約噗文的 Pydantic 模型

class ScheduledPost(BaseModel):

content: str

posted: datetime

user\_id: int

@app.post("/api/scheduled\_posts")

async def create\_scheduled\_post(post: ScheduledPost):

# 使用 INSERT INTO 語句將預約噗文插入資料庫

try:

cursor.execute("""

INSERT INTO scheduled\_posts (user\_id, content, posted)

VALUES (%s, %s, %s);

""", (post.user\_id, post.content, post.posted))

db.commit() # 提交事務

return {"message": "Scheduled post created successfully", "data": post}

except Exception as e:

db.rollback() # 若有錯誤則回滾事務

raise HTTPException(status\_code=400, detail=f"Error saving to database: {e}")

@app.get("/api/scheduled\_posts")

async def get\_scheduled\_posts():

cursor.execute("SELECT id, user\_id, content, posted, is\_posted FROM scheduled\_posts")

posts = cursor.fetchall()

# 將資料轉換為 JSON 格式

post\_list = [

{

"id": post[0],

"user\_id": post[1],

"content": post[2],

"posted": post[3].strftime("%Y-%m-%d %H:%M:%S"), # 轉換 datetime 為字串

"is\_posted": bool(post[4])

}

for post in posts

]

return {"scheduled\_posts": post\_list}

scheduler.py

import pymysql

import os

from dotenv import load\_dotenv

import time

load\_dotenv()

print("Checking scheduled posts...")

# 確保從 .env 文件中加載正確的環境變數

db = pymysql.connect(

host=os.getenv("DB\_HOST"),

user=os.getenv("DB\_USER"),

password=os.getenv("DB\_PASSWORD"),

database=os.getenv("DB\_NAME")

)

def publish\_scheduled\_posts():

cursor = db.cursor()

try:

# 獲取所有待發佈的預約噗文

cursor.execute("SELECT id, user\_id, content FROM scheduled\_posts WHERE posted <= NOW() AND is\_posted = FALSE;")

posts = cursor.fetchall()

if posts:

print(f"Found {len(posts)} posts to publish.")

for post in posts:

post\_id, user\_id, content = post

# 發佈噗文，插入到 posts 表格

cursor.execute("INSERT INTO posts (user\_id, content) VALUES (%s, %s);", (user\_id, content))

# 更新 scheduled\_posts 表格中的 is\_posted 欄位

cursor.execute("UPDATE scheduled\_posts SET is\_posted = TRUE WHERE id = %s;", (post\_id,))

db.commit()

except Exception as e:

print(f"Error: {e}")

db.rollback()

while True:

publish\_scheduled\_posts()

time.sleep(60) # 每 60 秒檢查一次