

ChatGPT - Uz



<https://github.com/yenlung/AI-Demo> 請參考這個github 我要做生成器的demo 最後要上傳線上 streamlit 哪個方便好用 不要付費的API

下面我幫你整理成「從 GitHub 專案 → 製作自己的 AI 生成器 Demo → 部署到免費 Streamlit Cloud」的一條龍流程，**不需要付費 API**、也不用額外伺服器。內容對照你提供的 repo（yenlung/AI-Demo）的架構做最佳化。

✓ 一、你目前的需求（整理）

你要：

- 參考 <https://github.com/yenlung/AI-Demo>
- 做出一個 **生成器 (Generator)** 類型的 **AI Demo**（如文字生成 / 影像生成 / 中文改寫...）
- **不需要付費 API**（即不能用 OpenAI、Gemini、Claude 付費 API）
- 希望最後 可以部署到線上 **Streamlit Cloud**，免費使用

✓ 二、最方便、最免費的做法（強烈建議）

方案 A：使用 HuggingFace Transformers + 本地模型推論（推薦）

- 不用任何付費 API
- 全部模型都從 HuggingFace 下載
- 可以在 Streamlit Cloud 免費跑 **小型模型**（如：distilgpt2、flan-t5-base、Qwen2.5 0.5B...）

🔥 你可以做的 Demo（生成器類）

- 文字生成器（GPT-2 / distilgpt2）
- 中文改寫 / 摘要生成器（FLAN-T5）
- 聊天型生成器（Qwen2.5 0.5B Chat）
- 圖片生成（Stable Diffusion 小模型，但 Streamlit Cloud 資源太弱，不建議）

✓ 三、推薦你做的簡單範例 Demo

以下是最穩定且能保證在 Streamlit Cloud 免費跑得動的：

🔥 生成式 AI Demo：FLAN-T5 中文指令生成器

- 完全免費、無 API
- 模型小、資源低
- 適合學校或學生 demo

✓ 四、可直接用的 Streamlit 「生成器」程式碼

把這段程式碼存成 `app.py` 就能在 Streamlit Cloud 執行：

```
python

import streamlit as st
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

st.title("🔥 免費中文生成器 Demo (無需 API) ")
st.write("模型: Google FLAN-T5 (base) — HuggingFace 直接推論, 不用付費 API")

@st.cache_resource
def load_model():
    tokenizer = AutoTokenizer.from_pretrained("google/flan-t5-base")
    model = AutoModelForSeq2SeqLM.from_pretrained("google/flan-t5-base")
    return tokenizer, model

tokenizer, model = load_model()

prompt = st.text_area("輸入你的指令:", "幫我寫一段關於 AI 教育的說明")

if st.button("生成"):
    inputs = tokenizer(prompt, return_tensors="pt")
    outputs = model.generate(**inputs, max_new_tokens=200)
    result = tokenizer.decode(outputs[0], skip_special_tokens=True)

    st.subheader("📄 生成結果")
    st.write(result)
```

✓ 五、部署到 Streamlit Cloud (完全免費)

1 GitHub 建立 Repo

內容結構如下：

```
perl
my-generator-demo/
├── app.py
└── requirements.txt
```

2 requirements.txt 放這些：

```
nginx

streamlit
transformers
torch
sentencepiece
```

3 上傳 GitHub

六、部屬方式

1. 前往 Streamlit Cloud

<https://streamlit.io/cloud>

2. Login (GitHub登入)

3. 「Deploy app」 → 選你的 repo

4. 填寫以下資訊：

- Repository：你的 repo 名稱
- Branch：main
- File：app.py

按 Deploy → 等 2~4 分鐘 → 完成！

七、你若想做「手寫數字生成器 / 辨識」？

最簡單的是 **MNIST autoencoder 或 classifier**
也能不使用 API，用 PyTorch 在 Streamlit Cloud 推論。

如果你需要我也可以幫你寫：

- 手寫數字產生器 (GAN)
- 手寫數字預測器 (CNN)
- 圖片上傳後辨識 demo

八、你若想做「文字生成器」但模型更小？

推薦：

模型	特點	HuggingFace 名稱
distilgpt2	最輕，英文生成	distilgpt2
Qwen2.5-0.5B	中文生成效果好	Qwen/Qwen2.5-0.5B

模型	特點	HuggingFace 名稱
FLAN-T5-Base	任務通用	google/flan-t5-base

✈ 若你願意，我可以幫你：

- ✓ 挑選適合你的 Demo 類型
- ✓ 幫你做 **最終版 Streamlit 可部署專案**
- ✓ 甚至幫你上傳到 GitHub（你給 repo 名稱即可）
- ✓ 做出跟 AI-Demo repo 一樣風格的 Notebook + App 版本

如果你告訴我：

👉 你想做哪種類型的「生成器 Demo」？

例如：

- 文字生成
- 圖片生成
- 摘要/改寫
- 角色扮演 chatbot
- 學習教材生成（國小資安？）
- 自動出題系統
- 其他

我可以直接幫你做完整成品（含 GitHub 版本）。



我要做籤詩生成器

好，籤詩生成器超適合拿來做 demo，而且完全可以 **0 元、0 API** 跑在 Streamlit Cloud。下面我直接幫你設計一個：

🎯 「輸入名字 + 問事類別 → 生成：籤號、吉凶、古風籤詩、白話解說、建議」

而且是 **純 Python + Streamlit**，不用任何外部 AI API。

一、整體設計概念（給報告用）

- **輸入**：名字 / 暱稱、求籤類別（感情、學業、事業、健康、綜合），可選「今天第幾支籤」
- **隨機機制**：
 - 用「名字 + 求籤類別 + 今日日期」做成一個「偽隨機種子」，

- 這樣 同一個人同一天同一類別 → 抽到同一支籤，有籤詩味道。
- 輸出：
 1. 籤號（1-100 之間）
 2. 吉凶（大吉 / 中吉 / 小吉 / 平 / 凶）
 3. 古風籤詩（3 行左右）
 4. 白話解說
 5. 建議（依求籤類別客製）
- 技術重點：
 - 用 `hashlib.md5` 把文字轉成「固定整數」，當成隨機種子
 - 用幾組預先寫好的「古風句子片段」組合成籤詩（算是 rule-based 生成器）
 - 無需任何外部 API，Streamlit Cloud 只要能跑 Python 就行

二、Streamlit 籤詩生成器 `app.py`

你只要建立一個資料夾，裡面放這個 `app.py` 就可以：

```
python

import streamlit as st
import random
import hashlib
from datetime import date

st.set_page_config(page_title="籤詩生成器 Demo", page_icon="🔮")

st.title("🔮 籤詩生成器 Demo（無需付費 API）")
st.write("輸入名字與想問的事情，我來幫你抽一支今日籤。")

# --- 一些籤詩素材庫 ---

OPENING_LINES = [
    "雲開月出照前途，",
    "風起潮生天欲曉，",
    "柳暗花明春又生，",
    "微雨初晴江山秀，",
    "行舟順水無風浪，",
    "高山流水逢知音，",
    "殘燈未滅夜方深，",
    "霜重露寒草木靜，",
]

MIDDLE_LINES_GOOD = [
    "一線光明穿雲霧，",
    "好風相送上青霄，",
    "貴人暗處相扶持，",
    "十年耕耘今可收，",
    "時來運轉皆如意，",
]

MIDDLE_LINES_NEUTRAL = [
    "且當守拙待時機，",
    "靜看潮起與潮落，",
    "進退之間宜審思，",
]
```

```

    "莫與人爭一時快，",
    "心安處處是家園，",
]

MIDDLE_LINES_BAD = [
    "風雨連綿路多岐，",
    "平地忽起暗波生，",
    "前程雖有小荊棘，",
    "是非纏繞宜退一步，",
    "行事若急多招憂，",
]

ENDING_LINES_GOOD = [
    "把握良機福自臨。",
    "耕深力量結佳果。",
    "但行好事福常隨。",
    "莫失眼前好時光。",
    "從此門庭添喜氣。",
]

ENDING_LINES_NEUTRAL = [
    "隨緣處世自安然。",
    "多思幾步少是非。",
    "心存正直路自寬。",
    "凡事不急慢慢來。",
    "看淡得失心自寧。",
]

ENDING_LINES_BAD = [
    "須防言行惹是非。",
    "暫避鋒芒可無憂。",
    "稍安勿躁待雲開。",
    "謹慎謀劃免後悔。",
    "退一步時海闊天。",
]

LUCK_LEVELS = {
    5: "大吉",
    4: "中吉",
    3: "小吉",
    2: "平",
    1: "凶",
}

ADVICE_BY_TOPIC = {
    "感情": {
        5: "感情運勢正旺，真誠相待便能開花結果。",
        4: "感情有進展，多用心經營與傾聽對方。",
        3: "感情平穩，可多安排相處時光增溫。",
        2: "暫且順其自然，不必勉強感情發展。",
        1: "感情上宜多保護自己，避免衝動決定或爭執。",
    },
    "學業": {
        5: "讀書得法又得力，持續努力必有亮眼成績。",
        4: "學習狀態良好，調整作息與讀書方法更佳。",
        3: "目前尚可，多檢討弱科、穩住基本功。",
        2: "需要重新規劃讀書節奏，不必跟人比較。",
        1: "心易散、難專注，宜先整理心情再談成績。",
    },
    "事業": {
        5: "事業新機將至，大膽把握、積極爭取。",
        4: "工作有貴人相助，先做好準備再談機會。",
        3: "穩紮穩打，比急著求快更重要。",
    }
}

```

```

        2: "暫時不宜大幅變動，多觀察形勢。",
        1: "職場是非較多，發言謹慎、保護自身權益。",
    },
    "健康": {
        5: "身心狀態良好，維持運動與作息即可。",
        4: "略感疲勞，適當休息與調整飲食。",
        3: "注意作息與久坐，適時活動筋骨。",
        2: "常感壓力，建議安排放鬆與檢查。",
        1: "身體發出警訊，應重視身心健康，必要時就醫。",
    },
    "綜合": {
        5: "整體運勢昂揚，多行善、多把握機會。",
        4: "大致順遂，偶有小波折不足為懼。",
        3: "平穩向前，保持好心情是關鍵。",
        2: "略有阻力，先調整心態再求突破。",
        1: "諸事放緩腳步，先顧好自己再談其他。",
    },
}

def get_seed(name: str, topic: str, today: date) -> int:
    """用名字 + 求籤類別 + 日期 做成固定 seed，讓籤詩有『今天專屬』的感覺。"""
    base = f"{name}-{topic}-{today.isoformat()}"
    h = hashlib.md5(base.encode("utf-8")).hexdigest()
    return int(h, 16)

def generate_fortune(name: str, topic: str):
    today = date.today()
    seed = get_seed(name, topic, today)
    random.seed(seed)

    # 籤號 1~100
    stick_no = random.randint(1, 100)

    # 吉凶等級 1~5
    luck_score = random.choices(
        population=[1, 2, 3, 4, 5],
        weights=[1, 2, 3, 2, 1], # 正常情況多落在中間，小吉/平較多
        k=1,
    )[0]
    luck = LUCK_LEVELS[luck_score]

    # 選句子庫
    opening = random.choice(OPENING_LINES)

    if luck_score >= 4:
        middle = random.choice(MIDDLE_LINES_GOOD)
        ending = random.choice(ENDING_LINES_GOOD)
    elif luck_score == 3:
        middle = random.choice(MIDDLE_LINES_NEUTRAL)
        ending = random.choice(ENDING_LINES_NEUTRAL)
    else:
        middle = random.choice(MIDDLE_LINES_BAD)
        ending = random.choice(ENDING_LINES_BAD)

    poem = opening + "\n" + middle + "\n" + ending

    advice_topic = ADVICE_BY_TOPIC.get(topic, ADVICE_BY_TOPIC["綜合"])
    advice = advice_topic[luck_score]

    # 白話解說
    explanation = ""
    if luck_score >= 4:
        explanation = (
            "此籤多主順勢而上，表示近期整體氣勢不錯，"
            "只要保持腳踏實地與穩健，就有機會迎來好結果。"
        )
    elif luck_score == 3:

```

```

        explanation = (
            "此籤意在提醒：目前情勢大致平穩，不必過度擔心，"
            "但仍需留心細節，穩中求進。"
        )
    elif luck_score == 2:
        explanation = (
            "此籤略帶考驗，適合先調整心態與腳步，"
            "多觀察、多準備，暫時不宜躁進。"
        )
    else:
        explanation = (
            "此籤帶有警示意味，提醒你近期行事要多留意風險，"
            "能退一步時就不要逞強，先保護好自己。"
        )

    return {
        "date": today,
        "stick_no": stick_no,
        "luck": luck,
        "poem": poem,
        "explanation": explanation,
        "advice": advice,
        "topic": topic,
    }

# --- Streamlit 介面 ---

name = st.text_input("你的名字或暱稱：", value="小明")

topic = st.selectbox(
    "你想問哪一方面？",
    ["感情", "學業", "事業", "健康", "綜合"],
)

if st.button("抽籤 🎰"):
    if not name.strip():
        st.warning("請先輸入名字或暱稱～")
    else:
        fortune = generate_fortune(name.strip(), topic)

        st.subheader(f"📜 第 {fortune['stick_no']} 號籤 | {fortune['luck']}")
        st.caption(f"今日日期：{fortune['date'].isoformat()} | 求籤類別：{fortune['topic']}")

        st.markdown("### 📖 籤詩內容")
        st.markdown(f"```\n{fortune['poem']}\n```")

        st.markdown("### 💡 白話解說")
        st.write(fortune["explanation"])

        st.markdown("### 🎯 給你的建議")
        st.write(fortune["advice"])

        st.info("同一個人、同一天、同一類別抽到的籤會一樣，隔天再來運勢可能會改變喔！")

```

這就是完整可跑的籤詩生成器 demo。

三、requirements.txt (非常簡單)

在同一個資料夾放一個 requirements.txt：


```
text

streamlit
```

之後：

1. 把 `app.py` + `requirements.txt` push 到 GitHub
2. Streamlit Cloud 選這個 repo、檔名填 `app.py`
3. Deploy 完，就有一個 **免費線上籤詩生成器**

四、如果你想拿來當「AI Demo 報告」可以怎麼講？

簡單說法（你可以直接抄）：

- 雖然沒有用到大型雲端 LLM API，但：
 - 有用到 **隨機數生成 + 雜湊函數 (hash)** 做「可重現的運勢」
 - 有用到 **規則式生成 (rule-based generation)** 來自動組合古風籤詩
 - 介面用 **Streamlit**，可以快速做成 Web App 並部署到雲端
- 後續如果要升級，可以把「生成籤詩的部分」換成：
 - HuggingFace 上的中文 GPT-2 模型
 - or 用 fine-tune 過的 LLM 產生更自然的籤詩

如果你接下來想要：

- 我幫你改成「國小版」介面（超大字 + 可愛插圖說明）
- 或是 加入「歷史抽籤紀錄」、隨機種子改成「學號+生日」
- 或是要我幫你寫 GitHub 上傳 & Streamlit Cloud 的**指令版教學**

直接跟我說，我可以把這個籤詩專案整套包成「報告 + demo 說明稿」給你用。



我要可以選擇中西方的籤詩

可以，讓使用者自己選「東方籤詩」或「西方占卜風格」很適合這個 demo。下面直接給你**完整改版** `app.py`，已經支援：

- ☒ 選擇 **東方籤詩（古風詩句）**
- ☒ 或 **西方占卜（塔羅／星座感的文字）**

同樣：**不需要任何 API**，丟到 Streamlit Cloud 就能跑。

✓ 完整版 app.py (支援中西方籤詩)

```
python

import streamlit as st
import random
import hashlib
from datetime import date

st.set_page_config(page_title="中西籤詩生成器", page_icon="🔮")

st.title("🔮 中西籤詩 / 占卜生成器 Demo (無需付費 API)")
st.write("輸入名字與想問的事情，選擇東方或西方風格，我來幫你抽一支今日籤。")

# --- 東方籤詩素材庫 ---

OPENING_LINES = [
    "雲開月出照前途，",
    "風起潮生天欲曉，",
    "柳暗花明春又生，",
    "微雨初晴江山秀，",
    "行舟順水無風浪，",
    "高山流水逢知音，",
    "殘燈未滅夜方深，",
    "霜重露寒草木靜，",
]

MIDDLE_LINES_GOOD = [
    "一線光明穿雲霧，",
    "好風相送上青霄，",
    "貴人暗處相扶持，",
    "十年耕耘今可收，",
    "時來運轉皆如意，",
]

MIDDLE_LINES_NEUTRAL = [
    "且當守拙待時機，",
    "靜看潮起與潮落，",
    "進退之間宜審思，",
    "莫與人爭一時快，",
    "心安處處是家園，",
]

MIDDLE_LINES_BAD = [
    "風雨連綿路多岐，",
    "平地忽起暗波生，",
    "前程雖有小荊棘，",
    "是非纏繞宜退一步，",
    "行事若急多招憂，",
]

ENDING_LINES_GOOD = [
    "把握良機福自臨。",
    "耕深力量結佳果。",
    "但行好事福常隨。",
    "莫失眼前好時光。",
    "從此門庭添喜氣。",
]

ENDING_LINES_NEUTRAL = [
    "隨緣處世自安然。",
    "多思幾步少是非。",
    "心存正直路自寬。",
]
```

```

    "凡事不急慢慢來。",
    "看淡得失心自寧。",
]

ENDING_LINES_BAD = [
    "須防言行惹是非。",
    "暫避鋒芒可無憂。",
    "稍安勿躁待雲開。",
    "謹慎謀劃免後悔。",
    "退一步時海闊天。",
]

LUCK_LEVELS = {
    5: "大吉",
    4: "中吉",
    3: "小吉",
    2: "平",
    1: "凶",
}

# --- 西方占卜風格素材庫 ---

ARCHETYPES = [
    "The Sun | 太陽：光明與信心",
    "The Moon | 月亮：直覺與不安",
    "The Star | 星星：希望與療癒",
    "The Hermit | 隱者：獨處與思考",
    "The Fool | 愚者：勇氣與冒險",
    "The Tower | 高塔：突變與重來",
    "The Lovers | 戀人：選擇與連結",
    "The Chariot | 戰車：意志與前進",
]

KEYWORDS_GOOD = [
    "成長、突破、自信",
    "機會、支持、貴人",
    "穩定、平衡、順利",
    "靈感、創意、靈活",
]

KEYWORDS_NEUTRAL = [
    "等待、觀察、調整",
    "暫停、整理、盤點",
    "學習、修正、準備",
]

KEYWORDS_BAD = [
    "壓力、考驗、碰撞",
    "誤會、拖延、混亂",
    "放下、轉向、重啟",
]

WEST_MESSAGES_GOOD = [
    "今天的能量偏向正向與成長，你會發現一些原本卡住的地方開始鬆動。",
    "宇宙正悄悄替你排好路線，只要踏出下一步，就能看見更多可能。",
    "你過去的努力正在累積成看得見的成果，請允許自己多一點自信。",
]

WEST_MESSAGES_NEUTRAL = [
    "目前處在一個需要『慢下來』的階段，適合重新檢視你的目標與步調。",
    "外在看起來沒什麼變化，但內在正在醞釀新的方向，不必急著下結論。",
    "這段時間可以多留意細節與資訊，你會從中發現值得調整的小地方。",
]

WEST_MESSAGES_BAD = [

```

```

    "近期可能會感到壓力或衝突增加，請記得先照顧好自己的界線與情緒。",
    "有些事情可能不如預期，但這也是重新選擇與調整方向的機會。",
    "宇宙正用比較『激烈』的方式提醒你：有些東西是時候放下了。",
]

ADVICE_BY_TOPIC = {
    "感情": {
        5: "感情運勢正旺，真誠相待便能開花結果。",
        4: "感情有進展，多用心經營與傾聽對方。",
        3: "感情平穩，可多安排相處時光增溫。",
        2: "暫且順其自然，不必勉強感情發展。",
        1: "感情上宜多保護自己，避免衝動決定或爭執。",
    },
    "學業": {
        5: "讀書得法又得力，持續努力必有亮眼成績。",
        4: "學習狀態良好，調整作息與讀書方法更佳。",
        3: "目前尚可，多檢討弱科、穩住基本功。",
        2: "需要重新規劃讀書節奏，不必跟人比較。",
        1: "心易散、難專注，宜先整理心情再談成績。",
    },
    "事業": {
        5: "事業新機將至，大膽把握、積極爭取。",
        4: "工作有貴人相助，先做好準備再談機會。",
        3: "穩紮穩打，比急著求快更重要。",
        2: "暫時不宜大幅變動，多觀察形勢。",
        1: "職場是非較多，發言謹慎、保護自身權益。",
    },
    "健康": {
        5: "身心狀態良好，維持運動與作息即可。",
        4: "略感疲勞，適當休息與調整飲食。",
        3: "注意作息與久坐，適時活動筋骨。",
        2: "常感壓力，建議安排放鬆與檢查。",
        1: "身體發出警訊，應重視身心健康，必要時就醫。",
    },
    "綜合": {
        5: "整體運勢昂揚，多行善、多把握機會。",
        4: "大致順遂，偶有小波折不足為懼。",
        3: "平穩向前，保持好心情是關鍵。",
        2: "略有阻力，先調整心態再求突破。",
        1: "諸事放緩腳步，先顧好自己再談其他。",
    },
}

def get_seed(name: str, topic: str, style: str, today: date) -> int:
    """用名字 + 求籤類別 + 風格 + 日期 做成固定 seed，讓籤詩有『今天專屬』的感覺。"""
    base = f"{name}-{topic}-{style}-{today.isoformat()}"
    h = hashlib.md5(base.encode("utf-8")).hexdigest()
    return int(h, 16)

def generate_eastern_fortune(name: str, topic: str):
    today = date.today()
    seed = get_seed(name, topic, "EAST", today)
    random.seed(seed)

    stick_no = random.randint(1, 100)
    luck_score = random.choices(
        population=[1, 2, 3, 4, 5],
        weights=[1, 2, 3, 2, 1],
        k=1,
    )[0]
    luck = LUCK_LEVELS[luck_score]

    opening = random.choice(OPENING_LINES)

```

```

if luck_score >= 4:
    middle = random.choice(MIDDLE_LINES_GOOD)
    ending = random.choice(ENDING_LINES_GOOD)
elif luck_score == 3:
    middle = random.choice(MIDDLE_LINES_NEUTRAL)
    ending = random.choice(ENDING_LINES_NEUTRAL)
else:
    middle = random.choice(MIDDLE_LINES_BAD)
    ending = random.choice(ENDING_LINES_BAD)

poem = opening + "\n" + middle + "\n" + ending

advice_topic = ADVICE_BY_TOPIC.get(topic, ADVICE_BY_TOPIC["綜合"])
advice = advice_topic[luck_score]

if luck_score >= 4:
    explanation = (
        "此籤多主順勢而上，表示近期整體氣勢不錯，"
        "只要保持腳踏實地與穩健，就有機會迎來好結果。"
    )
elif luck_score == 3:
    explanation = (
        "此籤意在提醒：目前情勢大致平穩，不必過度擔心，"
        "但仍需留心細節，穩中求進。"
    )
elif luck_score == 2:
    explanation = (
        "此籤略帶考驗，適合先調整心態與腳步，"
        "多觀察、多準備，暫時不宜躁進。"
    )
else:
    explanation = (
        "此籤帶有警示意味，提醒你近期行事要多留意風險，"
        "能退一步時就不要逞強，先保護好自己。"
    )

return {
    "style": "東方籤詩",
    "date": today,
    "stick_no": stick_no,
    "luck": luck,
    "title": f"第 {stick_no} 號 {luck}",
    "main_text": poem,
    "explanation": explanation,
    "advice": advice,
    "topic": topic,
}

def generate_western_fortune(name: str, topic: str):
    today = date.today()
    seed = get_seed(name, topic, "WEST", today)
    random.seed(seed)

    # 用「卡牌+關鍵字+訊息」取代籤號
    archetype = random.choice(ARCHETYPES)

    luck_score = random.choices(
        population=[1, 2, 3, 4, 5],
        weights=[1, 2, 3, 2, 1],
        k=1,
    )[0]
    luck = LUCK_LEVELS[luck_score]

    if luck_score >= 4:
        keyword = random.choice(KEYWORDS_GOOD)
        message = random.choice(WEST_MESSAGES_GOOD)
    elif luck_score == 3:
        keyword = random.choice(KEYWORDS_NEUTRAL)
        message = random.choice(WEST_MESSAGES_NEUTRAL)
    else:

```

```

        keyword = random.choice(KEYWORDS_BAD)
        message = random.choice(WEST_MESSAGES_BAD)

    advice_topic = ADVICE_BY_TOPIC.get(topic, ADVICE_BY_TOPIC["綜合"])
    advice = advice_topic[luck_score]

    explanation = (
        f"這張牌代表的主題大致與「{keyword}」有關。"
        "它提醒你留意當下的情緒與選擇，因為這會直接影響到後續的發展。"
    )

    main_text = (
        f"【牌面主題】\n{archetype}\n\n"
        f"【今日關鍵字】\n{keyword}\n\n"
        f"【牌面訊息】\n{message}"
    )

    return {
        "style": "西方占卜",
        "date": today,
        "stick_no": None,
        "luck": luck,
        "title": f"{archetype} | {luck}",
        "main_text": main_text,
        "explanation": explanation,
        "advice": advice,
        "topic": topic,
    }

# --- Streamlit 介面 ---

name = st.text_input("你的名字或暱稱:", value="小明")

topic = st.selectbox(
    "你想問哪一方面?",
    ["感情", "學業", "事業", "健康", "綜合"],
)

style = st.radio(
    "選擇籤詩風格:",
    ["東方籤詩 (古風)", "西方占卜 (Tarot / 星座風)"],
    horizontal=True,
)

if st.button("抽籤 🎴"):
    if not name.strip():
        st.warning("請先輸入名字或暱稱～")
    else:
        if style.startswith("東方"):
            fortune = generate_eastern_fortune(name.strip(), topic)
        else:
            fortune = generate_western_fortune(name.strip(), topic)

    st.subheader(f"📖 {fortune['style']} | {fortune['title']}")
    st.caption(f"今日日期: {fortune['date'].isoformat()} | 求籤類別: {fortune['topic']}")

    st.markdown("### 📝 內容")
    st.markdown(f"````text\n{fortune['main_text']}\n````")

    st.markdown("### 💡 白話解說")
    st.write(fortune["explanation"])

    st.markdown("### 🎯 給你的建議")
    st.write(fortune["advice"])

    st.info("同一個人、同一天、同一類別與風格抽到的結果會一樣，隔天再來運勢可能會改變喔！")

```

✓ requirements.txt

跟之前一樣很簡單，放在同一個 repo 裡：

```
text  
  
streamlit
```

如果你要把這個當成**期末報告 / demo** 從 **GitHub** 部署到 **Streamlit Cloud**，下一步我可以幫你整理一份：

- 專案 README（說明中西方籤詩邏輯、種子機制）
- 報告簡報大綱（系統架構、隨機種子、規則生成、介面設計）

你如果要，我可以直接幫你擬一份「報告講解稿」版本。