

計概期末專題— Line Bot 串接 OpenAI

Group D

卓于心、陳樂宇、甘沛玉、柯芷儀

摘要

- 設計理念及目標
- 專案說明
- 安裝步驟
- 功能介紹

- 進階功能
- 專案困難點
- 附錄

設計理念及目標

現在最流行科技之一：人工智慧，已幫我們解決許多問題，目前全世界最廣泛使用的 AI-ChatGPT，能夠處理各式各樣的問題，例如提供建議、撰寫文章、翻譯文本等。此外，除了AI的發展，人們隨著時代也逐漸依賴社群軟體的存在，例如 Line、IG 以及 facebook 等。然而這次專題研究挑選 Line 的理由是因為 Line 在 2016 年推出了 Line Bot-聊天機器人，可以模擬真人與使用者進行對話，所以想要藉由這次專題討論的機會，研究並串接 Line Bot 和 OpenAI ChatGPT，實踐在 Line 進行智慧對話的目標使得 Line Bot 的功能及價值更加提升。

專案說明

■ 使用工具

- ngrok 服務（讓本地服務器可供 Line Bot 訪問）
- Line Channel（Line Bot）
- OpenAI、Google Gemini
- Google Cloud Firebase（對話紀錄儲存）

■ 大致功能

- 透過 Line Bot 與 AI 對話

一、安裝步驟

環境與前置作業設定

建立 Line Bot

- 進入 Line Developers 網站並按照網站說明成為開發者
- 建立 Provider
- 建立 Messaging API Channel

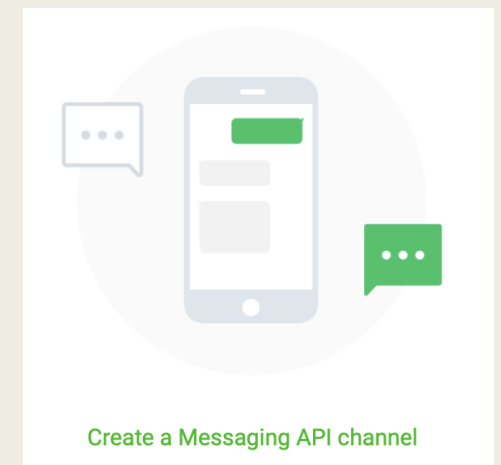
Create a new provider

Provider name [?]

✗ Don't leave this empty
✓ Don't use special characters (4-byte Unicode)
✓ Enter no more than 100 characters

A provider is an individual developer, company, or organization that provides services. For more details, see the [documentation](#).

Cancel Create



Line 開發者：<https://developers.line.biz/en/>

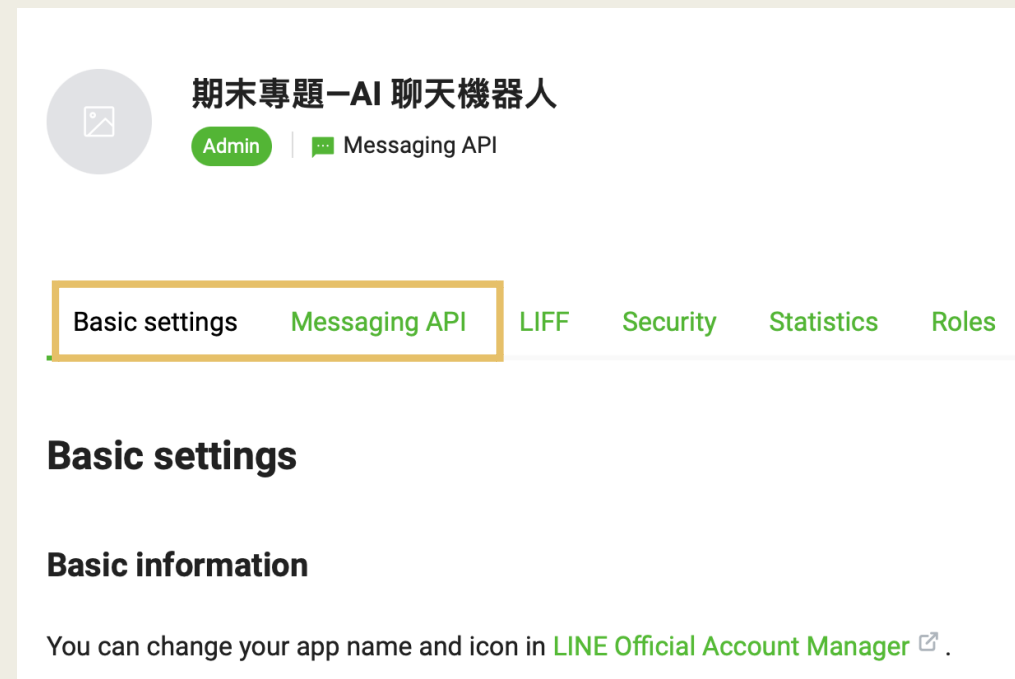
建立 Line Bot

- 命名機器人與設定機器人基本資訊
(不可留白，可自定義，未來可修改)

| | |
|---|--------------------|
| Channel name | 期末專題-AI 聊天機器人 |
| Note: The channel name can't be changed for seven days. | |
| ✓ Don't leave this empty | |
| ✓ Don't use special characters (4-byte Unicode) | |
| ✓ Enter no more than 20 characters | |
| Channel description | 可以儲存對話紀錄的 AI 聊天機器人 |
| ✓ Don't leave this empty | |
| ✓ Don't use special characters (4-byte Unicode) | |
| ✓ Enter no more than 500 characters | |
| Category | 網站 & 部落格 |
| ✓ Don't leave this empty | |
| Subcategory | 網站 & 部落格 (其他) |
| ✓ Don't leave this empty | |

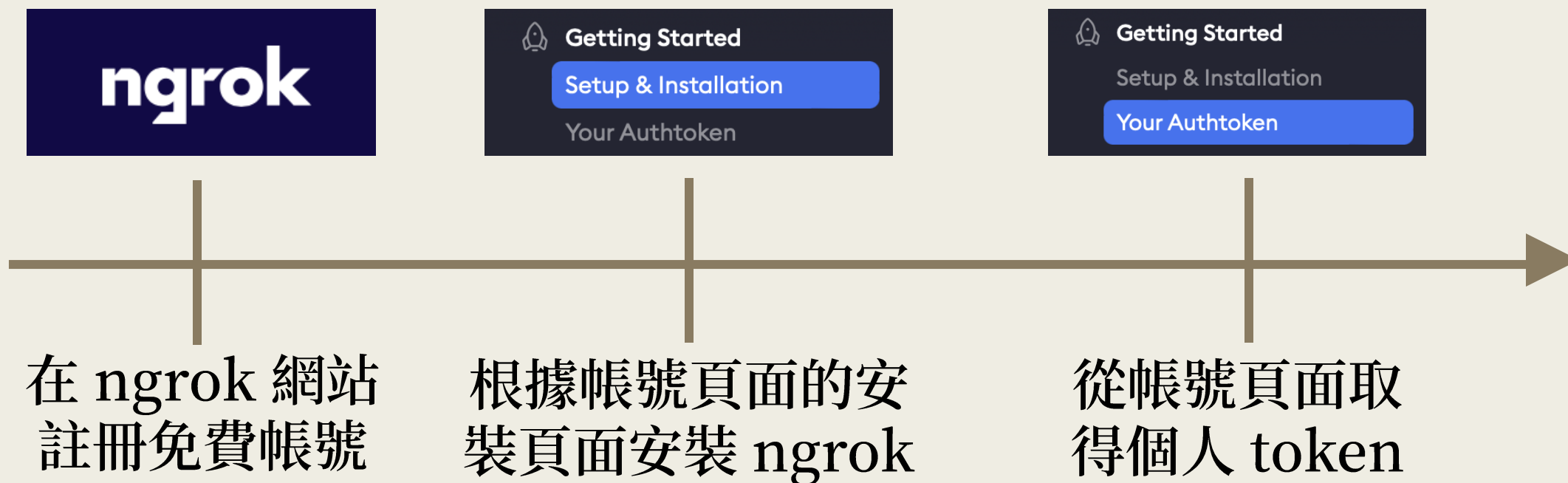
建立 Line Bot

- 透過 Basic Setting 取得 Channel secret
- 透過 Messaging API 中取得 Channel access token



https://developers.line.biz/console/channel/YOUR_CHANNEL_ID

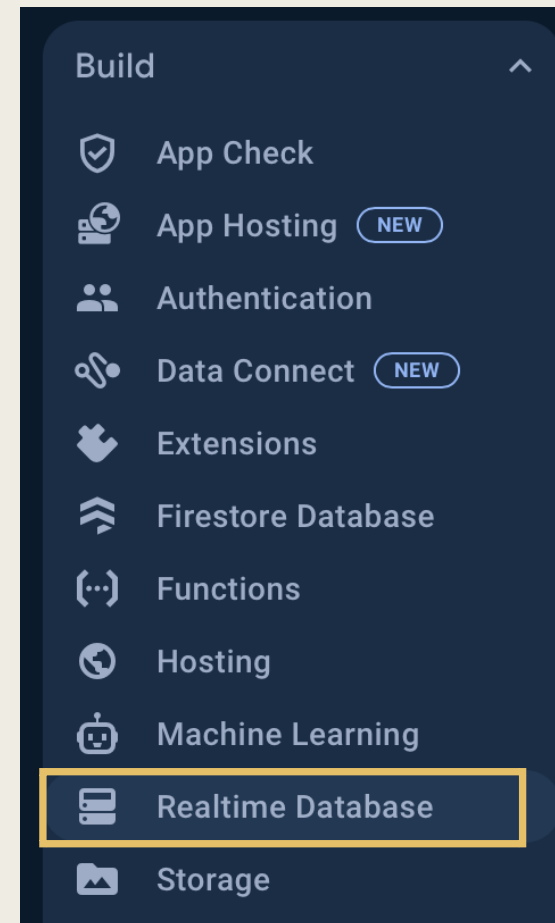
安裝 ngrok 並取得 token— 用於建立 Webhook



ngrok 網站： <https://ngrok.com> 帳號頁面： <https://dashboard.ngrok.com>

設定 Google Cloud Firebase

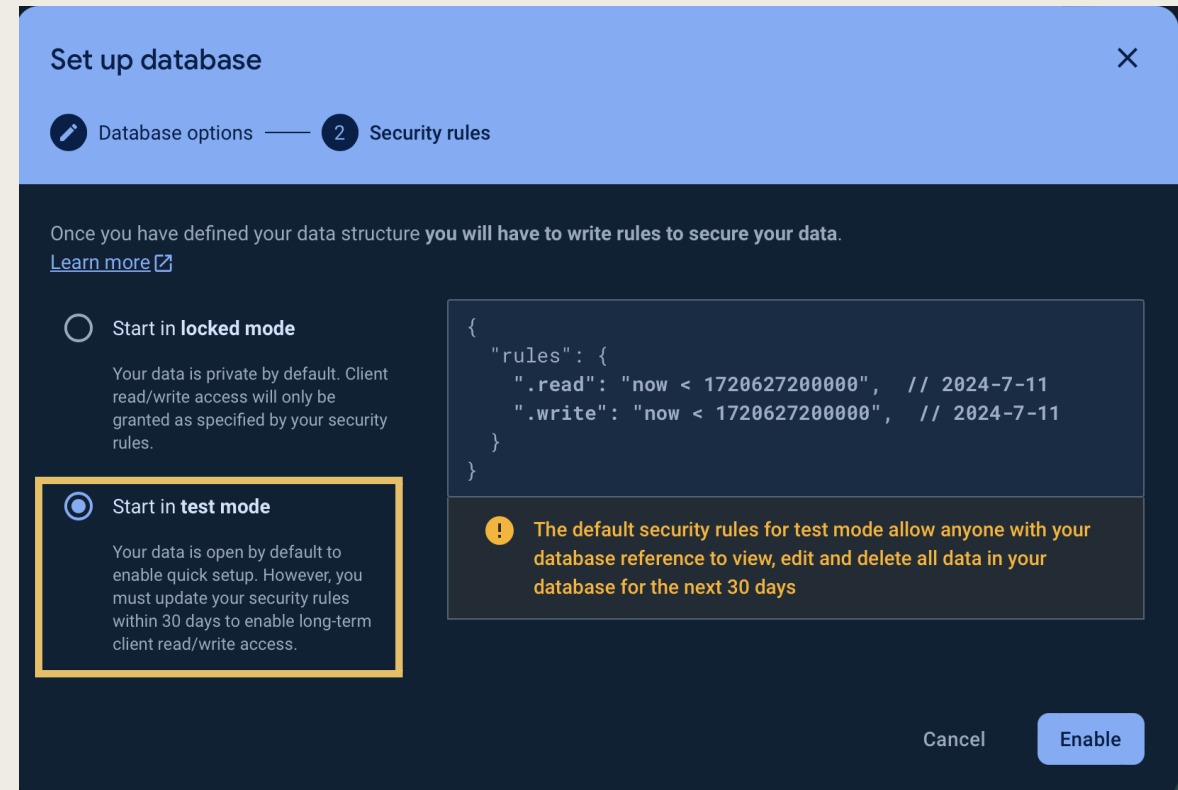
- 到 Firebase 創建 Google Cloud Project
- 在專案頁面側邊欄中找到 Realtime Database



Firebase 網站：<https://firebase.google.com>

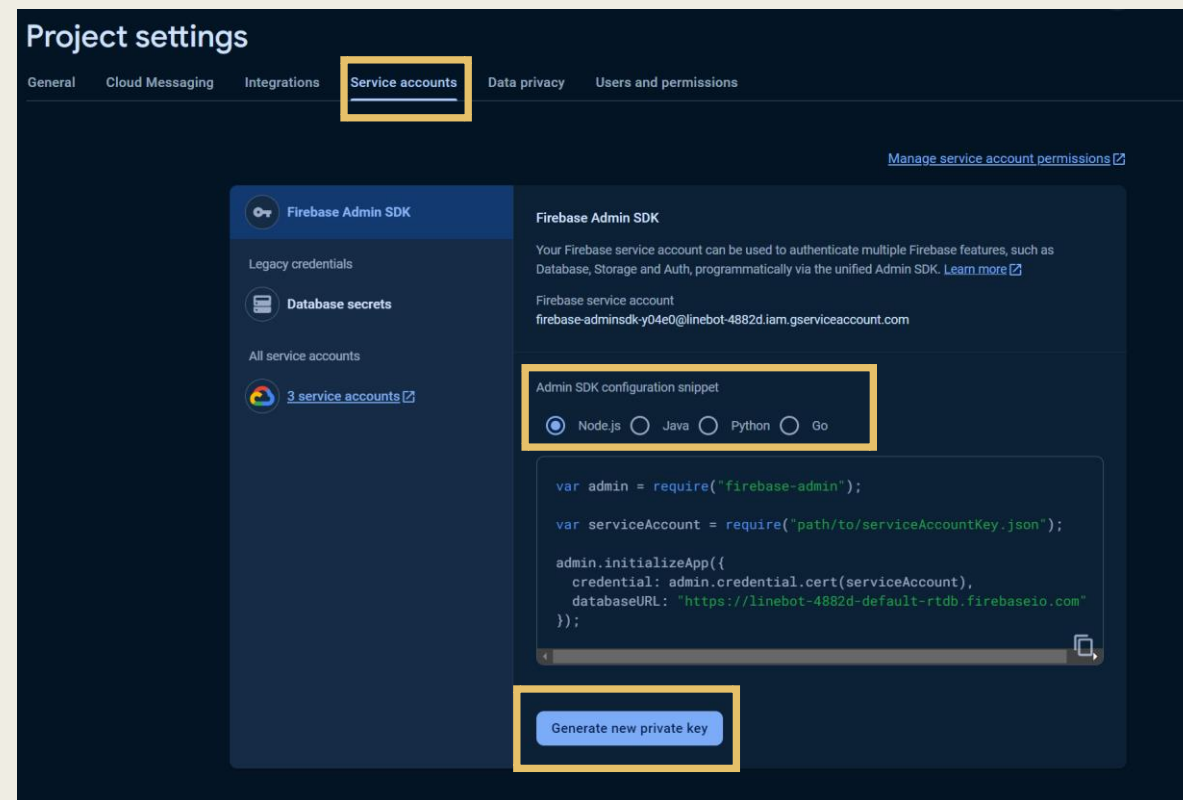
設定 Google Cloud Firebase

- 建立 database，設定為 test mode
- 取得 reference URL
 - https://YOUR_PROJECT_ID.firebaseio.com/



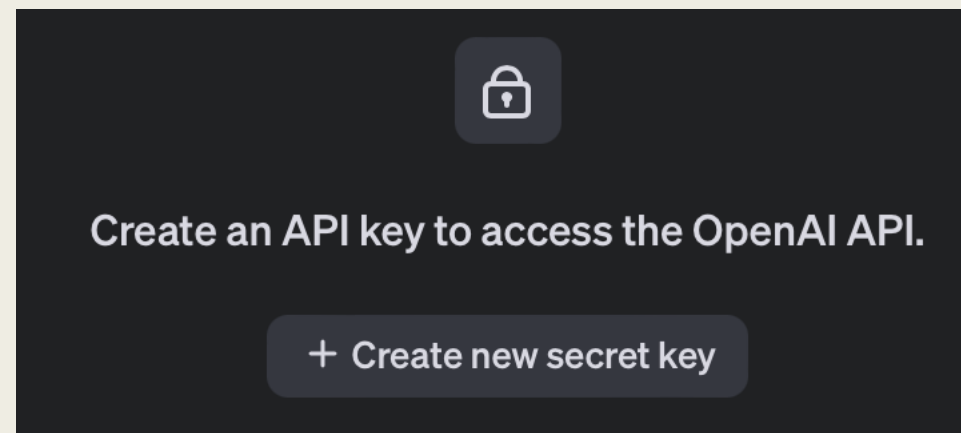
取得 Google Cloud Firebase Admin SDK

- 進入 Project setting
- 點選 Service accounts
- 選擇 Node.js
- 下載 private key (.json 檔)
- 放入與 LineBot.py 同資料夾



取得 OpenAI API Key

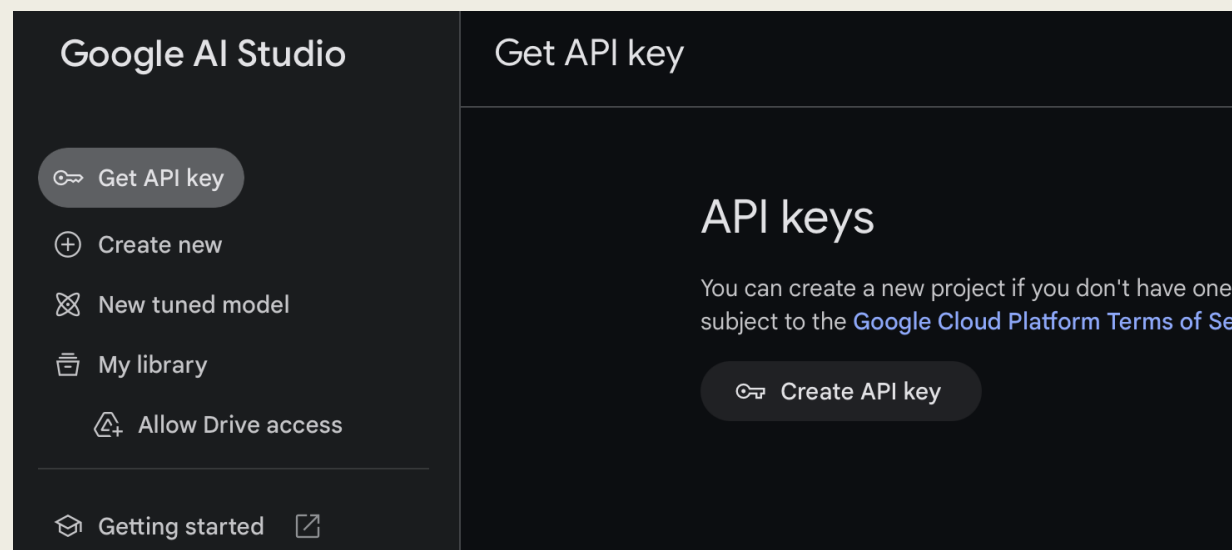
- 註冊 OpenAI 帳號
- 進入 OpenAI API 金鑰申請頁面
- 點選 Create new secret key



申請頁面：<https://platform.openai.com/account/api-keys>

取得 Google Gemini API Key

- 進入 Google AI for Developer 平台
- 點選 Get API Key in Google AI Studio
- 點選 Get API Key



平台：<https://ai.google.dev>

此時應該要具備……

- LINE_CHANNEL_SECRET
- LINE_CHANNEL_ACCESS_TOKEN
- FIREBASE_URL
- FIREBASE_ServiceAccount_Key.json
- GEMINI_KEY
- OPENAI_KEY

安裝函式庫— 於終端機輸入以下指令

- `pip install google.generativeai`
- `pip install Firebase_admin`
- `pip install Flask`
- `pip install line-bot-sdk`
- `pip install openai`
- `pip install time`
- `pip install requests`
- `pip install pyngrok`

設定 Line Bot access token、secret 和 Firebase



```
# 設定你的 Line Bot 的 access token 和 secret 以及 firebase
access_token = 'YOUR ACCESS TOKEN'
secret = 'YOUR SECRET'
firebase_url = "YOUR FIREBASE URL"
cred = firebase_admin.credentials.Certificate("path/to/serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': firebase_url
})
```

二、功能介紹

與 AI 對話、複誦

功能介紹—與 AI 對話

- 呼叫兩個 AI 的 API 並準備開始處理收到的訊息
 - 以訊息開頭前四個字元作為判別



```
def process_message(msg, conversation_id):  
  
    # 處理收到的訊息並呼叫 OpenAI Gemini API  
  
    openai.api_key = 'YOUR OPENAI API'  
    googleAiapi_key = "YOUR GEMINI API"  
    ai_msg = msg[:4].lower()
```

功能介紹—與 AI 對話

- 若前四字元為 opai — 運用 gpt-3.5-turbo 模型

```
● ● ●  
elif ai_msg == 'opai':  
  
    response = openai.ChatCompletion.create(  
        model='gpt-3.5-turbo',  
        messages=messages,  
        max_tokens=256,  
        temperature=0.5,  
    )  
  
    reply_msg = response['choices'][0]['message']['content'].strip()
```

功能介紹—與 AI 對話

- 若前四字元為 gemi — 運用 gemini-pro 模型



```
elif ai_msg == 'gemi' :
```

```
    genai.configure(api_key = googleAiapi_key)  
    model = genai.GenerativeModel('gemini-pro')
```

```
    response = model.generate_content(messages)  
    reply_msg = response.text.strip()
```

功能介紹—複誦

如果訊息開頭不等於 opai 或 gemi
訊息也不等於 !clear
則機器人複誦使用者的訊息

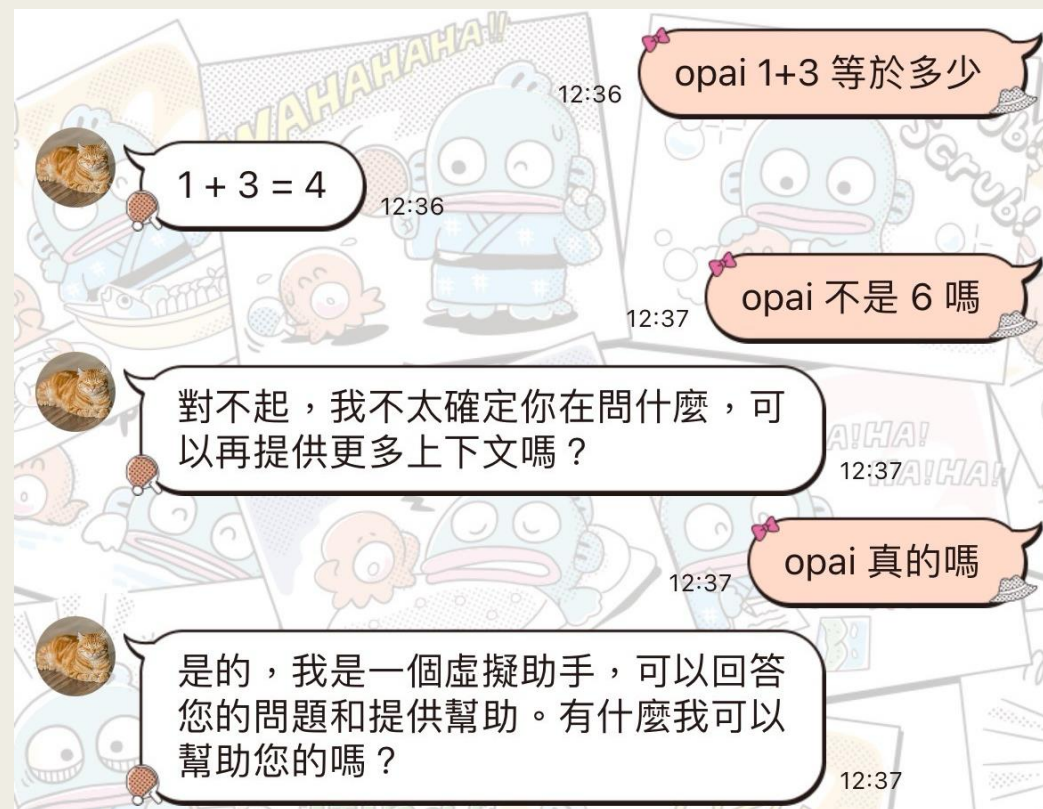


三、進階功能

根據上下文回應使用者

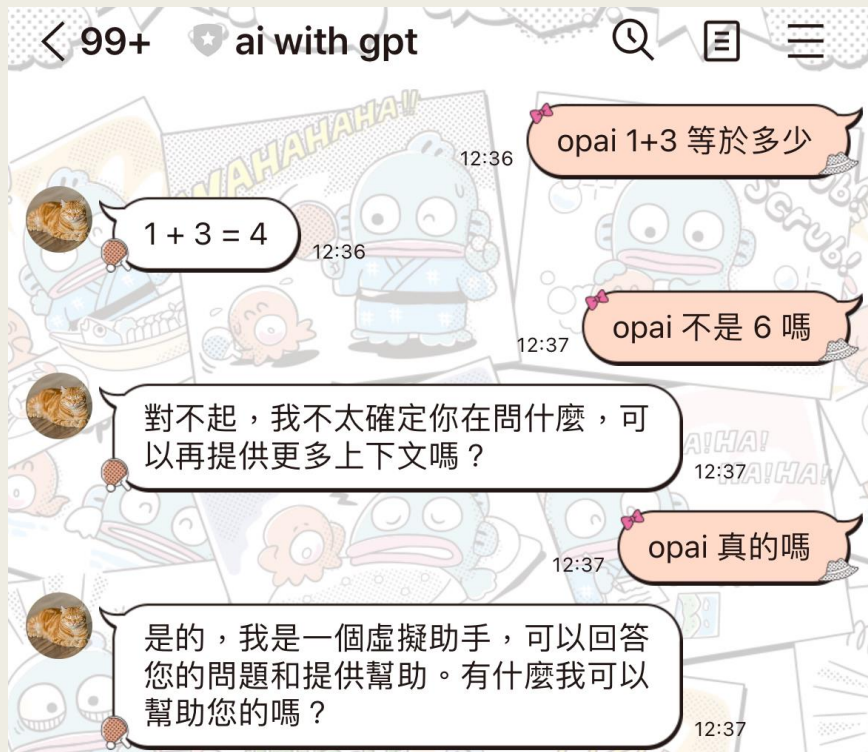
進階功能—延伸基本功能

添加原因：只有串接 AI 的功能，無法參考上文，故回答時可能會答非所問，缺乏訊息間連貫性



原參考：<https://steam.oxxostudio.tw/category/python/example/line-bot-openai-1.html>

進階功能—根據上下文回應使用者



原本功能

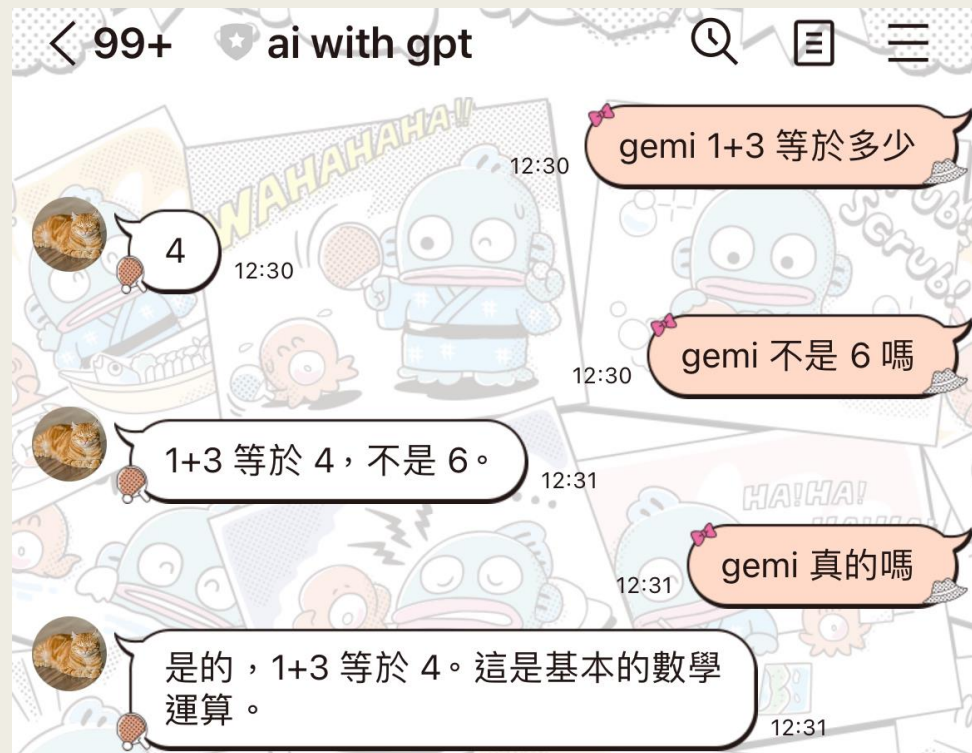


增加資料庫後

進階功能—根據上下文回應使用者



原本功能



增加資料庫後

進階功能—清除對話紀錄（開啟新對話）

- 判斷使用者的訊息（非判斷前四字元）

- !clear

- 清除所有對話紀錄

- !clear opai

- 清除與 ChatGPT 的對話紀錄

- !clear gemi

- 清除與 Gemini 的對話紀錄



額外加入功能 —

- 加入自動更新 Webhook 網址
 - 減少每次開啟機器人就需手動更新的麻煩
- 加入 Gemini (Google AI)
 - ChatGPT 需要收費，利用 Gemini 作為替代方案

四、專案困難點

專案困難點

1. 人工智慧的選擇：市面上有很多不同的生成式 ai 可以使用，但是調用的函數跟使用限制各有不同。例：如果要使用 openai 的 api，就要花錢購買額度。
2. 連接許多另外程式：無論是 line bot 的環境開發，還是在建立及串接的部分，需要引用且安裝許多另外的程式，複雜度相對較大。
3. LineBot SDK 改版：2023/07 升級到了 v.3 版本，教學網站資料較少，有些 v.2 版本的使用法已不支援。



分工

Group D

分工

- 程式碼：卓于心、陳樂宇
- 簡報：甘沛玉
- 報告：柯芷儀
- 最後統整：陳樂宇

參考資料

References

內文中的參考資料

1. 安裝 ngrok 並取得 token

<https://steam.oxxostudio.tw/category/python/example/ngrok.html>

2. 建立 Line Bot

<https://steam.oxxostudio.tw/category/python/example/line-developer.html>

3. 設定 Firebase

<https://ericlinyuting.com/LineBot-GPT-with-Gemini>

其他參考資料

- <https://github.com/line/line-bot-sdk-python/blob/master/README.rst>
- <https://axk51013.medium.com/技術分享-2023-03-01-chatgpt-api可以用啦-2435b6d23bbd>
- https://blog.jiatool.com/posts/gemini_api/
- <https://medium.com/anna-hsaio-前端開發記/firebase小學堂-開發chapter-2-資料庫建置-b93890518404>
- https://github.com/youjunjer/AutoUpdate_LineBot_WebhookURL/blob/main/ngrok_%E8%87%AA%E5%8B%95%E6%9B%B4%E6%96%B0webhook%E7%B6%B2%E5%9D%80.ipynb

感謝聆聽

Group D