**1. 建立 / 登入 Facebook for Developers 帳戶**

到 Meta for Developers 建立應用（選 Business 類型）並完成基本設定。

**2. 將應用連結到你的 Meta Business（Business Manager）**

在 Business Settings 把 App 與你的 Business Account 關聯（必要時也要設定角色、權限）。

**3. 在 App Dashboard 產生快速測試 Token（若只做測試）**

App Dashboard 左側選單：WhatsApp → API setup → Generate access token（按鈕）。產生後要立即複製，部分介面只會顯示一次。

**4.（推薦）建立 System User 並生成長期/永久 Token**

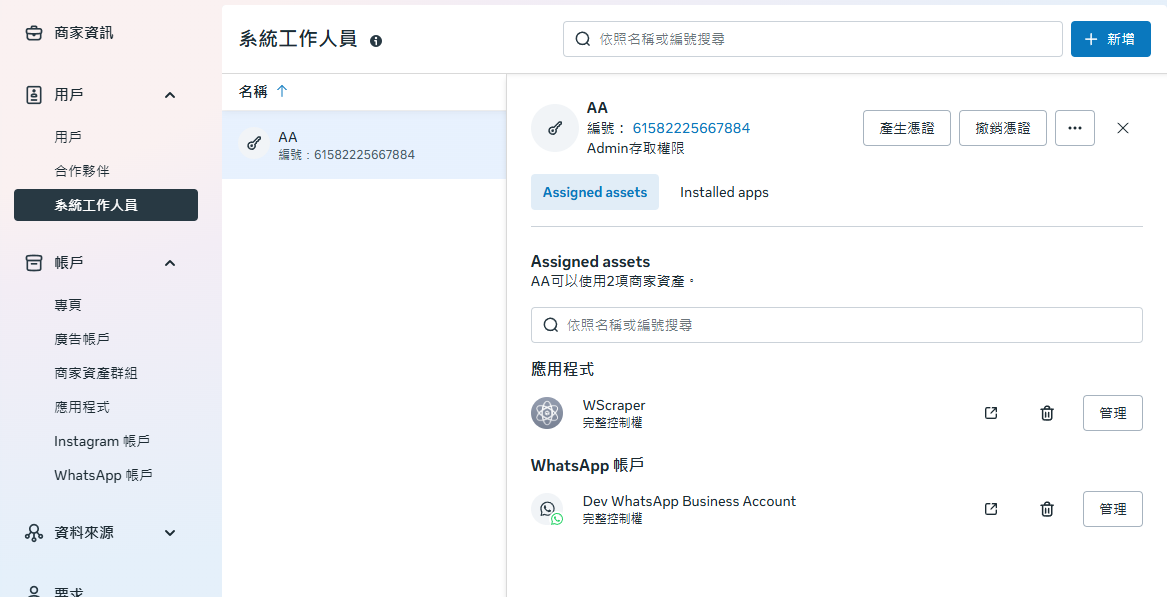
到 Meta Business Settings → System Users，建立一個 system user（代表你的後端服務）。  
<https://business.facebook.com/login/web/?next=https%3A%2F%2Fbusiness.facebook.com%2Fsettings>

給此 system user 所需的權限（WhatsApp assets / 管理權）並「生成 Token」。這樣生成的 token 可設定為長期或不易過期，比用戶短期 token 更適合伺服器端使用。注意：有時需要 Business 管理員在 Meta Business Suite 承認請求。  
  
如果你的目標是 讓後端伺服器透過 Cloud API 發送 / 接收 WhatsApp 訊息，

請至少勾選以下 權限（Permissions）：

類別 權限名稱 用途說明

WhatsApp whatsapp\_business\_messaging 發送訊息、使用 /messages API 必須

WhatsApp whatsapp\_business\_management 管理 WhatsApp Business 帳號、Webhook、模板訊息設定等需要  
  


**5. 取得 phone\_number\_id 與 WhatsApp Business Account ID**

在 App Dashboard 的 WhatsApp 設定或透過 Graph API 查詢可得（例如查詢你的 WABA 下的 phone numbers）。這兩個 ID 在發訊息時會用到（/v17.0/{phone\_number\_id}/messages 類似路徑）。 *注意: 如第一次發送信息給客戶，需其做出回復後才可以繼續發送信息*  


**6. 設定 Webhook（若需要接收訊息）\*\*\* 好像測試電話號碼係觸發不到此的 \*\*\***

在 App Dashboard 設定 webhook URL 與驗證 token，用以接收回呼事件。Webhook 驗證 token 是你自己選的一個安全字串。





**7. 安全與實務注意**

token 應妥善保管（不要放在前端程式碼、公開 Repo）。可以設定自動輪替、監控使用情況。若 token 泄漏，立即在 Business Settings 中撤銷或重產。某些介面只會顯示 token 一次，務必存好。

AI 設計的界面草圖



核心流程示意



## 主要三種 role

| **role** | **說明** |
| --- | --- |
| system | 系統指令或固定 Prompt，用來設定模型的角色、口吻、行為規則。例如：「你是客服助理，要用繁體中文禮貌回答。」 |
| user | 使用者發出的訊息。模型會視為問題或請求。 |
| assistant | 模型先前生成的回覆。可以用於多輪對話上下文保存。 |

## PHP 目錄結構建議

/public\_html/

├── index.html ← 前端聊天框

├── chat.php ← 後端 API

└── faq\_db.php ← FAQ 資料庫連線

**FAQ 資料庫結構 (MySQL)**

CREATE TABLE faq (

id INT AUTO\_INCREMENT PRIMARY KEY,

question TEXT NOT NULL,

answer TEXT NOT NULL,

embedding TEXT NOT NULL -- json\_encode 的 float array

);

## chat.php (cURL 版本)

<?php

session\_start();

header('Content-Type: application/json; charset=utf-8');

require\_once 'faq\_db.php'; // PDO $db

$OPENAI\_API\_KEY = "sk-xxxxx"; // ←換成你的 OpenAI Key

$MAX\_HISTORY = 5; // 保存最近對話輪數

$SIM\_THRESHOLD = 0.65; // 高相似度閾值

// 取得使用者訊息

$input = json\_decode(file\_get\_contents('php://input'), true);

$userMessage = trim($input['message'] ?? '');

if (!$userMessage) {

echo json\_encode(["reply"=>"請輸入訊息"]);

exit;

}

// 初始化 Session

if (!isset($\_SESSION['chat\_history'])) $\_SESSION['chat\_history'] = [];

$\_SESSION['chat\_history'][] = ["role"=>"user","content"=>$userMessage];

if (count($\_SESSION['chat\_history']) > $MAX\_HISTORY\*2) {

$\_SESSION['chat\_history'] = array\_slice($\_SESSION['chat\_history'],-$MAX\_HISTORY\*2);

}

// ---------------- Embedding ----------------

function getEmbedding($text,$apiKey){

$data=["model"=>"text-embedding-3-small","input"=>$text];

$ch=curl\_init('https://api.openai.com/v1/embeddings');

curl\_setopt($ch,CURLOPT\_RETURNTRANSFER,true);

curl\_setopt($ch,CURLOPT\_POST,true);

curl\_setopt($ch,CURLOPT\_HTTPHEADER,[

"Content-Type: application/json",

"Authorization: Bearer $apiKey"

]);

curl\_setopt($ch,CURLOPT\_POSTFIELDS,json\_encode($data));

$res = curl\_exec($ch);

curl\_close($ch);

$resData=json\_decode($res,true);

return $resData['data'][0]['embedding'] ?? [];

}

function cosineSimilarity($vec1,$vec2){

$dot=$norm1=$norm2=0;

foreach($vec1 as $i=>$v){ $dot+=$v\*$vec2[$i]; $norm1+=$v\*$v; $norm2+=$vec2[$i]\*$vec2[$i]; }

return $dot/(sqrt($norm1)\*sqrt($norm2)+1e-8);

}

// ---------------- 查 FAQ ----------------

$faqRows = $db->query("SELECT \* FROM faq")->fetchAll(PDO::FETCH\_ASSOC);

$userEmbedding = getEmbedding($userMessage,$OPENAI\_API\_KEY);

$topFAQs=[];

foreach($faqRows as $row){

$faqEmbedding = json\_decode($row['embedding'],true);

$sim = cosineSimilarity($userEmbedding,$faqEmbedding);

$topFAQs[]=['answer'=>$row['answer'],'sim'=>$sim,'question'=>$row['question']];

}

// 排序取 top 3

usort($topFAQs, fn($a,$b)=>$b['sim']<=>$a['sim']);

$relevantFAQ = array\_slice($topFAQs,0,3);

// ---------------- 判斷相似度 ----------------

$contextText = '';

$logSim = [];

foreach($relevantFAQ as $f){

$logSim[] = "{$f['question']} => sim={$f['sim']}";

if($f['sim'] >= $SIM\_THRESHOLD){

$contextText .= $f['answer'] . "\n";

}

}

// 如果找不到高相似度 FAQ → fallback 人工客服

if(trim($contextText) === ''){

$reply = "抱歉，我暫時無法提供相關資訊，您可以聯絡人工客服：service@yourcompany.com 或撥打 1234-5678。";

$\_SESSION['chat\_history'][]=["role"=>"assistant","content"=>$reply];

echo json\_encode(["reply"=>$reply,"log"=>$logSim]);

exit;

}

// ---------------- GPT System Prompt ----------------

$systemPrompt = <<<PROMPT

你是 XX 公司客服助理，回答要禮貌、繁體中文。

請根據以下 FAQ 回答使用者問題，如果無法找到答案，請禮貌告知使用者：

$contextText

PROMPT;

$messages=[["role"=>"system","content"=>$systemPrompt]];

$messages=array\_merge($messages,$\_SESSION['chat\_history']);

// ---------------- 呼叫 GPT ----------------

$data=["model"=>"gpt-4o-mini","messages"=>$messages];

$ch=curl\_init('https://api.openai.com/v1/chat/completions');

curl\_setopt($ch,CURLOPT\_RETURNTRANSFER,true);

curl\_setopt($ch,CURLOPT\_POST,true);

curl\_setopt($ch,CURLOPT\_HTTPHEADER,[

"Content-Type: application/json",

"Authorization: Bearer $OPENAI\_API\_KEY"

]);

curl\_setopt($ch,CURLOPT\_POSTFIELDS,json\_encode($data));

curl\_setopt($ch,CURLOPT\_TIMEOUT,20);

$res = curl\_exec($ch);

curl\_close($ch);

$resData=json\_decode($res,true);

$reply=$resData['choices'][0]['message']['content'] ?? "無法取得回覆。";

// 儲存 AI 回覆

$\_SESSION['chat\_history'][]=["role"=>"assistant","content"=>$reply];

echo json\_encode(["reply"=>$reply,"log"=>$logSim]);