

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

就讀學校(請勾選): ☒國立陽明交通大學 ☐國立臺灣大學

團隊名稱: EMONION

團隊聯絡人: 覃柏鈞

團隊成員姓名: 張雅安、陳紫瑜、覃柏鈞、夏心謙、池旻柔

團隊指導教授: 甘致群

健康守護者 365：居家全年智慧健康監測應用程式
VitalCare 365： smart monitoring healthcare application
for users at home all the time

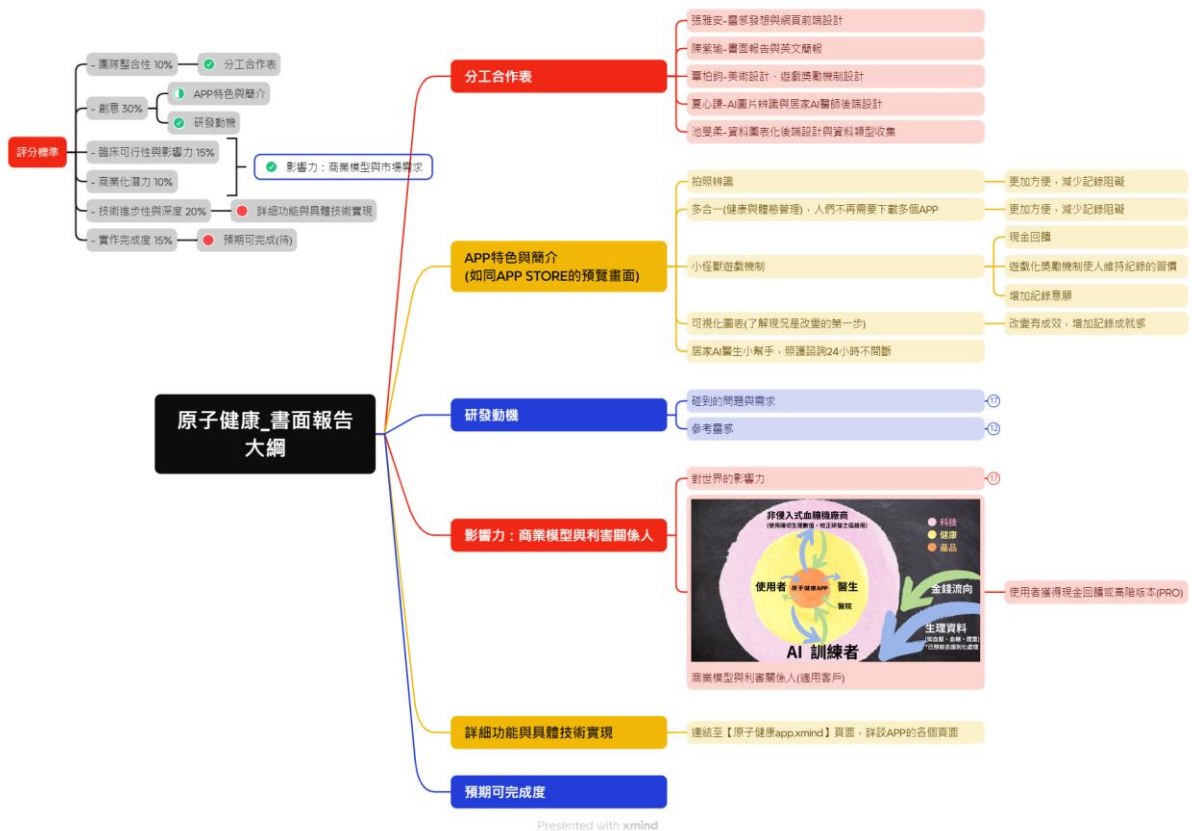
「陽明交大與台大生醫電資創新設計競賽」
複賽書面報告書
目錄

頁碼

一、	大綱	
	>.....	3
二、	動機與目標	
	>.....	3
三、	市場需求	
	>.....	3
四、	APP 簡介	
	>	5
五、	數據儲存與圖表生成	
	>.....	8
六、	AI 對話	
	>.....	11
七、	OpenAI 嵌入	
	>.....	12
八、	圖片數值判讀	
	>.....	13
九、	參考文獻	
	>.....	15

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

一、大綱



二、動機與目標

- 想法：人不是不重視自己的健康，而是無法立即看到成效，讓你變健康、變不健康的都是「原子般小的日常習慣」，若透過這個可視化健康的 APP，建立一個數位孿生小寵物，便可能能有效增加人們去進行有益健康的行為。
- 期許：期望讓病人能獲得連續性的照護。醫生可根據最實際的情況提供專業知識與建議，並適時關心使用者。

因此，我們以「原子健康」為名，設計了一款 APP。此系統有助於使用者建立良好的健康習慣，透過 AI 對話、視情況協助掛號，以及小遊戲與動畫，提升維持健康的樂趣。

三、市場需求

- 數據：身處於 AI 世代的我們，科技與科學的發展，皆得有數據的佐證及輔助，因此從醫療的角度來看，有病患或健康者提供連續的生理數值檢測結果對臨床研究相當重要，然而多數人沒有動機提供他人自己的健康數據，除此之外，24 小時不間段偵測生理數值相當困難。因此，我們需要一款 APP 不僅可以協助使用者記錄自己的健康狀況，更能於使用者同意的情況下，提供醫師、研究人員作為參考。

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

2. 全人照護：醫生沒有那麼多時間來好好花了一個病患身上，受健保制度與醫病比大的影響，醫生得在短時間內(如三分鐘)就了解病患的狀況並進行診斷與醫治，而病人卻得在短短三分鐘內把過去整整三天的狀況告知醫生，不僅會與當時的情況有所誤差，醫生也無從得知發病當下病患的體溫、心律、血壓等生理數值。若有 APP 協助記錄數據，醫師便能了解病患一整天、一整周的醫療資料，不僅能供醫生參考、輔助診斷，更能提供資料訓練 AI 模型，使病人在醫生不在時，也能記錄資料，更全面地受照護。
3. 商業價值

使用者、醫院、醫師、廠商、APP 開發者等多方合作，形成互益結構。開發者提供產品與技術，協助搭建病人與院方間的橋樑，使用者透過付費功能使開發者產生收益；開發者在使用者同意的情況下提供數據、資料給研究者與 AI 訓練者，提升功能，將健康及科技回饋於大眾。



「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

四、APP 簡介



你的健康助手

Hi! 歡迎來到 app 系，我是你的健康小寵物~ 請幫我變成你最喜欢的樣子再取個名字吧!



我的名字

之後就由我來陪伴你啦!

「了解自己是改善現狀的第一步」

請幫我閱讀個資和使用說明 若同意請在下方打勾。



☒ 個資使用說明
☒ 操作細節說明

太好啦! 我們的健康之旅即將開始。開始前, 讓我更了解你吧~

我的資料

1. gmail:

設置密碼:

確認密碼:

2. 姓名:

身高:

體重:

生日:

性別:

3. 為什麼想用此 APP?

4. 如何得知此 APP?

5. 你希望建立紀錄哪些數值的目標?

<input type="checkbox"/> 血糖	<input type="checkbox"/> ...	<input type="checkbox"/> ...
<input type="checkbox"/> 血氧	<input type="checkbox"/> ...	<input type="checkbox"/> ...
<input type="checkbox"/> 血壓	<input type="checkbox"/> ...	<input type="checkbox"/> ...

6. 期待自己多久紀錄一次生理數值呢?

7. 是否有重大疾病?

8. 是否有長期用藥?

9. 是否有過敏史?

 快來試試! 看個小量綱吧~

6. 擴充跳出選購畫面:

基本介紹

Free

Pro 介紹

NT...\$

是否願意將資料提供做為科學研究? (先先識別化, 不聯心, 僅用)

☐ 醫院、科研使用聲明

☒ 同意 → 可 free 使用 pro

☐ skip → 進入畫面



1. APP 歡迎頁面

以軟體吉祥物作為歡迎頁面，不同疾病追蹤、功能下，有不同的對應吉祥物，增加軟體互動性與趣味性。

2. 註冊與登入

若為新用戶，則在此頁面以電子郵件註冊帳號；舊用戶則重新登入，亦可選擇記住用戶，省去登入時間。

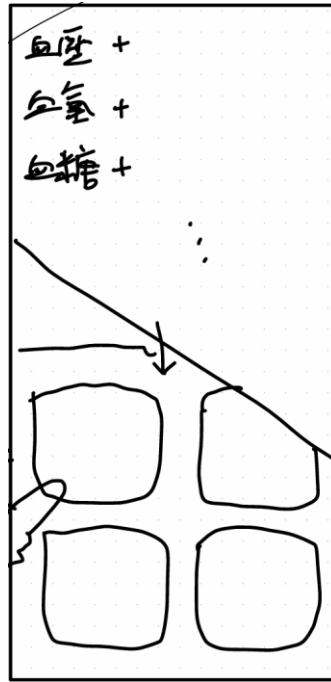
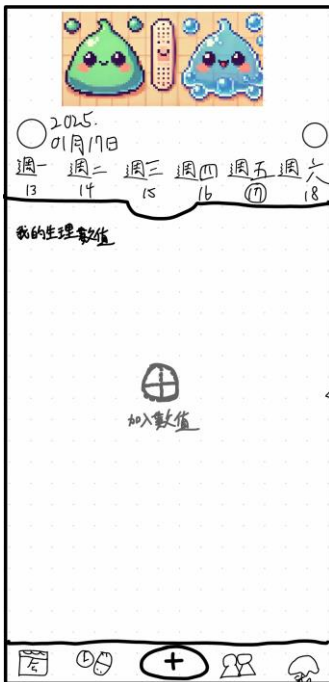
註冊時需填寫相關資料，提供個人化 APP 功能，填寫內容包括：

- (1) 基本資料：姓名、出生年月日、性別、身高、體重
- (2) 使用原因：日常健康紀錄、疾病狀態追蹤等...
- (3) 期望目標：體重、血壓、血糖控制等...
- (4) 記錄項目：用藥、生理數值（儀器測量）、生活習慣等...
- (5) 記錄頻率：一日數次、一周一次等.....
- (6) 疾病史、正在使用的藥物調查
- (7) 是否願意提供數據做為科學研究？
- (8) APP Pro 版本功能展示

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書



3. 使用界面



(1) 首頁 Dashboard

- 顯示小寵物的狀態（例如健康指數、心情表情）。
- 用戶每日健康數據的簡要概覽（例如體重、步數、血壓）。
- 行為建議卡片（例如 "今天多喝水"、"記得拍攝數據"）。

(2) 數據輸入/拍照上傳

- 手動輸入數據：簡單的表單，讓用戶輸入如體重、心率等。
- 拍照上傳數據：按鈕觸發上傳模組，展示拍照辨識後的數據結果。
- 結果驗證：用戶可以編輯或確認系統識別的數據。

(3) 小寵物互動

- 寵物根據數據的變化做出表情或動作。
- 用於提醒功能（如 "今天還沒量體重哦！"）。
- 顯示小寵物的「健康進度條」。

(4) 數據視覺化

- 使用簡單的折線圖或圓形圖，展示用戶健康數據的趨勢。
- 提供時間範圍篩選（如本週、本月）。

(5) 通知與成就

- 任務完成後的通知（如 "成功記錄 3 天！"）。
- 用戶可以查看已完成的健康任務或累積的獎勵。

(6) 用戶資料頁面

- 顯示個人基本資料（如年齡、性別）。
- 設定個人目標（如減重目標）。

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書



4. 時鐘提醒功能

顯示使用者的量測時間、用藥時間，傳送通知提醒使用者定期監測生理數值及用藥。

5. 與醫師、AI 對話

將使用者於 APP 上記錄的數值推送給醫師，提供醫師更長時間、精確的病患數據記錄。除此之外，訓練 ChatGPT 模組，針對醫療相關問題提出就醫建議。

6. 個人主頁

資料經系統化統整，清楚呈現，有助於輔助個人化醫療；實用性、互動性兼具。

7. 輔助套件

- (1) 螢幕小工具：使用者能在手機的畫面中加入小工具以提醒自己（限 ios）
- (2) 檢視統計：生成以時間為 x 軸的統計折線圖，讓使用者能選擇幾個數據來看看彼此間有無相關性（如：高血壓藥食用後隔多久血壓控制成功）
- (3) 就醫 QRcode：將生理數據打包成醫院或診所可用的格式，讓醫生判讀。
- (4) 問卷：使用滿意度調查

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

五、數據儲存與圖表生成

透過 Python 的網頁框架 Flask 撰寫程式碼，功能包括：

1. 建立資料庫模型

- (1) User：使用者註冊資料，包括姓名、電子郵件、加密後的密碼
- (2) HealthDataType：儲存健康數據，包含不同數據類型以及對應的單位
- (3) HealthData：將數據連結至特定使用者和模組

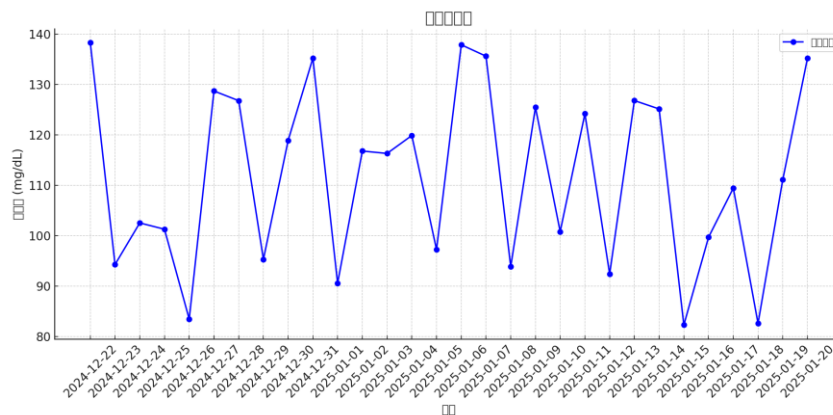
2. API

- (1) 註冊：加密安全儲存使用者密碼
- (2) 上傳健康數據：讀取圖片辨識後的數值，
- (3) 獲取健康數據：允許使用者依照時間查詢數值
- (4) 圖表生成：生成圖表後以 Base64 返回

3. 圖表生成

- (1) matplotlib 繪製生理數值趨勢圖表
- (2) Base64 將圖表以文字方式顯示於網頁或應用程式

4. 圖表範例



程式碼：

```
from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy
from flask_bcrypt import Bcrypt
import matplotlib.pyplot as plt
import io
import base64
from datetime import datetime
```

```
# 初始化 Flask 應用
app = Flask(__name__)
bcrypt = Bcrypt(app)
```

```
# 配置數據庫
```


「陽明交大與台大生醫電資創新設計競賽」

複賽書面報告書

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+mysqlconnector://root:password@localhost/health_app'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
```

數據庫模型

```
class User(db.Model):
    __tablename__ = 'Users'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    weight = db.Column(db.Float, nullable=False)
    email = db.Column(db.String(100), unique=True, nullable=False)
    username = db.Column(db.String(100), unique=True, nullable=False)
    password_hash = db.Column(db.String(255), nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
```

```
class HealthDataType(db.Model):
    __tablename__ = 'HealthDataTypes'
    id = db.Column(db.Integer, primary_key=True)
    type_name = db.Column(db.String(50), nullable=False)
    unit = db.Column(db.String(20), nullable=False)
```

```
class HealthData(db.Model):
    __tablename__ = 'HealthData'
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('Users.id'), nullable=False)
    data_type_id = db.Column(db.Integer, db.ForeignKey('HealthDataTypes.id'), nullable=False)
    value = db.Column(db.Float, nullable=False)
    recorded_at = db.Column(db.DateTime, nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
```

初始化數據庫

```
with app.app_context():
    db.create_all()
```

用戶註冊 API

```
@app.route('/register', methods=['POST'])
def register():
    data = request.json
    hashed_password = bcrypt.generate_password_hash(data['password']).decode('utf-8')
    new_user = User(
        name=data['name'],
        weight=data['weight'],
        email=data['email'],
        username=data['username'],
        password_hash=hashed_password
    )
    db.session.add(new_user)
    db.session.commit()
    return jsonify({"message": "用戶註冊成功"}), 201
```

上傳健康數據 API

```
@app.route('/upload_data', methods=['POST'])
def upload_data():
    data = request.json
    new_data = HealthData(
        user_id=data['user_id'],
```

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

```
data_type_id=data['data_type_id'],
value=data['value'],
recorded_at=datetime.strptime(data['recorded_at'], '%Y-%m-%d %H:%M:%S')
)
db.session.add(new_data)
db.session.commit()
return jsonify({"message": "數據上傳成功"}), 201

# 獲取健康數據 API
@app.route('/get_data', methods=['GET'])
def get_data():
    user_id = request.args.get('user_id')
    data_type_id = request.args.get('data_type_id')

    records = HealthData.query.filter_by(user_id=user_id,
data_type_id=data_type_id).order_by(HealthData.recorded_at).all()
    result = [
        {
            "recorded_at": record.recorded_at.strftime('%Y-%m-%d %H:%M:%S'),
            "value": record.value
        } for record in records
    ]
    return jsonify(result), 200

# 生成健康數據圖表 API
@app.route('/generate_chart', methods=['GET'])
def generate_chart():
    user_id = request.args.get('user_id')
    data_type_id = request.args.get('data_type_id')

    # 查詢數據
    records = HealthData.query.filter_by(user_id=user_id,
data_type_id=data_type_id).order_by(HealthData.recorded_at).all()
    dates = [record.recorded_at.strftime('%Y-%m-%d') for record in records]
    values = [record.value for record in records]

    # 檢查是否有數據
    if not records:
        return jsonify({"error": "No data available"}), 404

    # 繪製圖表
    plt.figure(figsize=(10, 6))
    plt.plot(dates, values, marker='o', linestyle='-', label='趨勢', color='b')
    plt.title("健康數據趨勢圖", fontsize=16)
    plt.xlabel("日期", fontsize=12)
    plt.ylabel("數值", fontsize=12)
    plt.grid(True)
    plt.xticks(rotation=45)
    plt.tight_layout()

    # 將圖表轉為 Base64
    buffer = io.BytesIO()
    plt.savefig(buffer, format='png')
    buffer.seek(0)
    image_base64 = base64.b64encode(buffer.getvalue()).decode('utf-8')
    buffer.close()
```

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

```
return jsonify({"chart": image_base64}), 200
```

```
if __name__ == '__main__':  
    app.run(debug=True)
```

六、AI 對話

1. HTML：#chat-container 聊天頁面，包含顯示區#chat-box、輸入區#user-input、傳送按鈕
2. 樣式設計：CSS 設定樣式，包括背景色、邊框、圓角，將使用者訊息和機器人訊息分為兩種顏色
3. JavaScript 函數：
 - (1) appendMessage()：新增聊天訊息，根據發送者為使用者或機器人設定不同的樣式，確保新訊息顯示後滾動至底部
 - (2) sendMessage()：獲取使用者輸入的訊息並顯示於聊天框中，並由/ask 傳送 POST，傳送輸入的內容
4. 伺服器交互：fetch 發送 HTTP 的請求，將使用者輸入的內容傳送至伺服器；伺服器以 JSON 格式回復

程式碼：

```
<!-- 聊天區域 -->  
<div id="chat-container" style="max-width: 600px; margin: auto; border: 1px solid #ddd; padding: 20px;">  
    <div id="chat-box" style="max-height: 400px; overflow-y: scroll; margin-bottom: 20px;">  
        <div class="chat-message bot" style="margin: 10px 0; padding: 10px; background-color: #f0f0f0; border-radius: 5px;">  
            你好！有什麼我可以幫忙的嗎？  
        </div>  
    </div>  
    <input type="text" id="user-input" placeholder="請輸入問題..." style="width: 80%; padding: 10px; margin-right: 10px; border-radius: 5px; border: 1px solid #ccc;">  
    <button onclick="sendMessage()" style="padding: 10px 15px; background-color: #007bff; color: white; border: none; border-radius: 5px; cursor: pointer;">  
        發送  
    </button>  
</div>  
  
<script>  
// 更新訊息顯示  
function appendMessage(message, sender) {  
    const chatBox = document.getElementById("chat-box");  
    const newMessage = document.createElement("div");  
    newMessage.classList.add("chat-message", sender);  
    newMessage.style.margin = "10px 0";  
    newMessage.style.padding = "10px";  
    newMessage.style.borderRadius = "5px";
```

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

```
// 根據發送者設定背景色
if (sender === "user") {
    newMessage.style.backgroundColor = "#d1f7d1";
    newMessage.style.textAlign = "right";
} else {
    newMessage.style.backgroundColor = "#f0f0f0";
}

newMessage.textContent = message;
chatBox.appendChild(newMessage);
chatBox.scrollTop = chatBox.scrollHeight; // 滾動到底部
}

// 發送問題給後端
async function sendMessage() {
    const userInput = document.getElementById("user-input").value;
    if (!userInput.trim()) return; // 若輸入框為空則不發送

    appendMessage(userInput, "user"); // 顯示使用者輸入的訊息
    document.getElementById("user-input").value = ""; // 清空輸入框

    try {
        // 向後端發送請求，並獲取回應
        const response = await fetch('/ask', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/x-www-form-urlencoded'
            },
            body: `user_input=${encodeURIComponent(userInput)}`
        });

        const data = await response.json();
        appendMessage(data.answer, "bot"); // 顯示 ChatGPT 回應
    } catch (error) {
        appendMessage("發生錯誤，請稍後再試！", "bot");
    }
}
</script>
```

七、OpenAI 嵌入

1. Flask 設定：app=FLASK(__name__)創建 Flask，使用 Flask 路由處理使用者的要求。
2. OpenAI API
 - (1)openai.api_key：設定 OpenAI API 密鑰用於驗證
 - (2)Openai.Completion.create()：呼叫 OpenAI 的 GPT 模型生成機器人的回覆
3. 路由
 - (1)/路由：返回一個 HTML 頁面，作為使用者聊天的前端界面
 - (2)/ask 路由：透過 POST 取得使用者的問題，並將 OpenAI API 的回應反回至前端

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

程式碼：

```
import os
import openai
from flask import Flask, render_template, request, jsonify

app = Flask(__name__)

# OpenAI API 密鑰設定
openai.api_key = 'sk-proj-
V8VACUjFfclitZoX9yblK28kBRi_Na_na4AWUCG7eA6emJcyEA5quARK7HV37VtKsAdQTBctO0T3BlbkFJf2V
LH0za3WgCNyDoSE8g4oqJF1GzltW7VTAPLWOBI28tOzpESVwfViTznOTwL2hZWH9oMBEQA' # 替換成
你的 OpenAI API 密鑰

@app.route('/')
def index():
    return render_template('index.html') # 這個是你的 HTML 頁面

@app.route('/ask', methods=['POST'])
def ask():
    user_input = request.form['user_input'] # 從前端獲取用戶輸入

    # 呼叫 OpenAI API 生成 ChatGPT 回應
    try:
        response = openai.Completion.create(
            model="gpt-4", # 或者 "gpt-3.5-turbo"
            prompt=user_input,
            max_tokens=150,
            temperature=0.7
        )
        answer = response.choices[0].text.strip() # 取得 ChatGPT 回應
        return jsonify({'answer': answer})
    except Exception as e:
        print(f"Error: {e}")
        return jsonify({'answer': '抱歉，發生了錯誤，請稍後再試！'})

if __name__ == '__main__':
    app.run(debug=True)
```

八、圖片數值判讀

1. 圖片上傳：讀取 png、jpg、jpeg、gif 檔，並將圖片儲存至 UPLOAD_FOLDER。
2. OCR 光學字元辨識：使用 Tesseract OCR 從圖片擷取文字，利用
pytesseract.image_to_string 擷取內容，使用英文(lang='eng')辨識
3. 儀器辨識：透過關鍵字匹配儀器與數值
4. 圖片處理：OpenCV 處理以提高 OCR 準確率
 - (1)cv2.cvtColor：轉成灰度圖
 - (2)cv2.threshold：二值化處理圖片
5. API：/analyze-image，以 POST 接收使用者圖片，並返回數值、儀器、單位

「陽明交大與台大生醫電資創新設計競賽」

複賽書面報告書

程式碼：

```
import os
import pytesseract
from flask import Flask, request, jsonify
from PIL import Image
import cv2

# 設定 Tesseract 路徑 (僅 Windows 用戶需要此步驟)
pytesseract.pytesseract_tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe' # 替換成你安裝的
Tesseract 路徑

app = Flask(__name__)

# 設定圖片上傳資料夾
UPLOAD_FOLDER = 'uploads'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# 檢查文件格式
def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# 儀器識別的簡單規則 (已更新為身高體重測量儀)
def analyze_instrument(text):
    text = text.lower()

    # 血壓計
    if "blood pressure" in text or "bp" in text or "血壓計" in text:
        return "血壓計", "血壓", "測量結果", "mmHg"

    # 血氧機
    elif "blood oxygen" in text or "spo2" in text or "血氧" in text or "血氧計" in text:
        return "血氧計", "血氧飽和度", "測量結果", "%"

    # 身高體重測量儀
    elif "height and weight" in text or "身高體重" in text or "體重" in text:
        return "身高體重測量儀", "身高和體重", "測量結果", "cm / kg"

    # 未知儀器
    return "未知儀器", "未知項目", "未知結果", "未知單位"

@app.route('/analyze-image', methods=['POST'])
def analyze_image():
    if 'file' not in request.files:
        return jsonify({'error': '沒有選擇檔案'}), 400

    file = request.files['file']
    if file and allowed_file(file.filename):
        # 保存檔案
        filename = file.filename
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(filepath)

        # 使用 OpenCV 讀取圖片
        img = cv2.imread(filepath)
```

「陽明交大與台大生醫電資創新設計競賽」 複賽書面報告書

```
# 使用 Tesseract OCR 進行識別
# 可以對圖片進行一些預處理來提高識別準確率
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
_, binary = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)

# 使用 Tesseract 進行 OCR 識別
text = pytesseract.image_to_string(binary, lang='eng')

# 打印提取的文本
print("提取的文本：", text)

# 分析儀器和數值
instrument, item, result, unit = analyze_instrument(text)

return jsonify({
    'instrument': instrument,
    'item': item,
    'result': result,
    'unit': unit,
    'extracted_text': text
})
else:
    return jsonify({'error': '不支持的文件格式'}), 400

if __name__ == '__main__':
    app.run(debug=True)
```

九、參考資料

1. Wang, B.J., Syu, F.J. Acceptability of Mobile Health Apps for Medical Care Measured using UTAUT2. Journal of Journal of Gerontechnology and Service Management. 2016; 4(4): 483-494.
2. Lin, P.H., Lan, Y.L., Chen, C.C., Lin, Y.C., Liu, W.C., Yan, Y.H. The Correlation of Health Literacy, Health Awareness and Healthy Promoting Lifestyles in Rehabilitation Patients. Taipei City Medical Journal. 2021; 18(1): 32-43.
3. 外溢保單：郝充仁.外溢保單會越來越多.康健雜誌.2024.07;頁 48-49
4. 許惠恒.AI 快思你慢想. 9789572985366th ed. 聯合報系
5. 邱宜君.AI 顧健康，哪個族群最愛用？. 康健雜誌. 2024; 306: 28-32.
6. 邱宜君. 愈健康賺愈多. 康健雜誌. 2024; 306: 118-126.