

延續上週作業建立的會員系統，運用前後端分離的思維，在會員頁中增添更多的功能。

#### 要求一：後端建立查詢會員資料的 API

在資料庫中已經包含一些會員的資料，透過 Python Flask 後端建立 API 供前端使用。

查詢會員資料 API 規格如下：

**Endpoint** 基礎網址：<http://127.0.0.1:3000/api/members>

**Request Method** 連線方法：GET

**Query String** 要求字串：username=要查詢的會員帳號

**URL Example** 呼叫範例：<http://127.0.0.1:3000/api/members?username=ply>

---

**Response Body** 回應格式：按照以下規定的 JSON 格式，回應查詢結果

成功查詢到會員資料，回應以下資訊：

```
{  
    "data": {  
        "id": 3,  
        "name": "強強",  
        "username": "strong"  
    }  
}
```

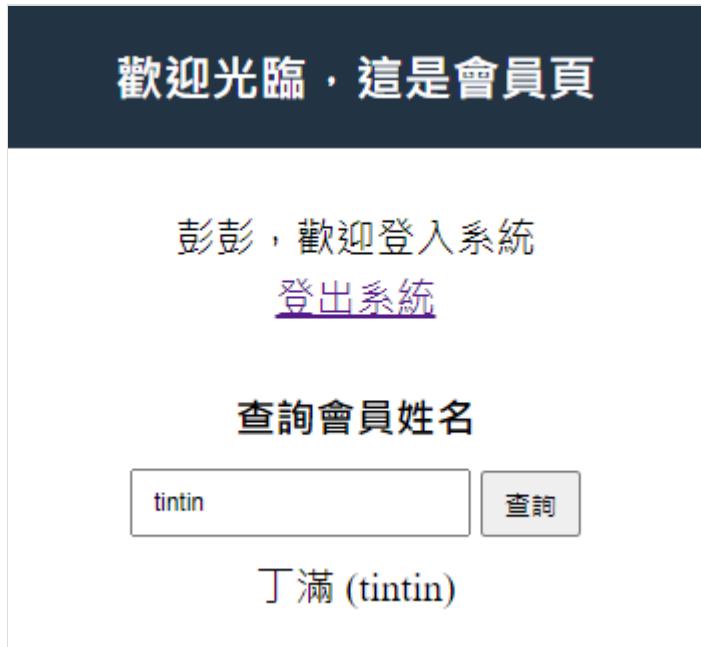
沒有查詢到任何資料，回應以下資訊：

```
{  
    "data": null  
}
```

要求二：前端透過 **Fetch** 連接 **API** 查詢會員資料，並顯示在畫面上

會員頁面網址：<http://127.0.0.1:3000/member/>

參考設計如下：



必須滿足的運作流程：

1. 完成要求一，建立【查詢會員資料 API】
2. 會員登入成功，在會員頁面中輸入要查詢的其他會員帳號，透過 JavaScript 的網路連線技術 (Fetch) 串接後端【查詢會員資料 API】，後端程式必須確實從資料庫中根據輸入的帳號查詢會員資料，並把查詢結果回應給前端。
3. JavaScript 程式取得查詢結果後，將資料顯示在會員頁面中。

### 要求三:完成修改姓名的功能 (Optional)

使用前後端分離的架構完成一個功能，必須建立後端 API 以及前端的互動介面，請按照以下要求完成修改會員姓名的功能。

修改會員姓名 API 規格如下：

**Endpoint** 基礎網址：<http://127.0.0.1:3000/api/member>

**Request Method** 連線方法：POST

**Request Header** 請求標頭：

標頭名稱	標頭內容
Content-Type	application/json

**Request Body** 請求資料：按照以下規定的 JSON 格式，發送請求

```
{  
    "name": "新的使用者姓名"  
}
```

---

**Response Body** 回應格式：按照以下規定的 JSON 格式，回應更新結果

成功更新資料：

```
{  
    "ok": true  
}
```

會員未登入，更新失敗：

```
{  
    "error": true  
}
```

會員頁面網址：<http://127.0.0.1:3000/member/>

參考設計如下：

<p>彭彭，歡迎登入系統 <a href="#">登出系統</a></p> <p><b>查詢會員姓名</b></p> <p><input type="text" value="tintin"/> <input type="button" value="查詢"/></p> <p>丁滿 (tintin)</p> <p><b>更新我的姓名</b></p> <p><input type="text" value="彭彭彭"/> <input type="button" value="更新"/></p>	<p>彭彭彭，歡迎登入系統 <a href="#">登出系統</a></p> <p><b>查詢會員姓名</b></p> <p><input type="text" value="tintin"/> <input type="button" value="查詢"/></p> <p>丁滿 (tintin)</p> <p><b>更新我的姓名</b></p> <p><input type="text" value="輸入新的姓名"/> <input type="button" value="更新"/></p> <p>更新成功</p>
---	---

必須滿足的運作流程：

1. 建立【修改會員姓名 API】
2. 會員登入成功，在會員頁面中輸入新的姓名，透過 JavaScript 的網路連線技術 (Fetch) 串接後端【修改會員姓名 API】，後端程式必須根據前端的輸入，以及在 Session 中存放的帳號，更新當前登入會員在資料庫中的姓名，並把更新狀況回應給前端。
3. JavaScript 程式取得更新結果後，將結果顯示在會員頁面中。