

建立一個獨立 Python Flask 網站專案，運用任何你學過的前後端知識，完成以下驗證系統。

要求一：建立首頁

執行寫好的 Python 程式，啟動 Flask 伺服器，並建立網站的首頁如下。

首頁網址：<http://127.0.0.1:3000/>

首頁功能：使用 form 表單建立帳號，密碼輸入框和登入按鈕。

首頁設計如下：

歡迎光臨，請輸入帳號密碼

登入系統

帳號

密碼

要求二：建立帳號密碼驗證功能

使用者可以透過首頁輸入帳號密碼，並且將帳號密碼透過 POST 方法傳送到後端程式進行驗證，根據驗證結果，將使用者導向到成功、或失敗的頁面。

驗證功能網址：<http://127.0.0.1:3000/signin>

連線方法：POST

成功頁面網址：<http://127.0.0.1:3000/member/>

參考設計如下：顯示成功登入系統訊息



失敗頁面網址：<http://127.0.0.1:3000/error/?message=自訂的錯誤訊息>

參考設計如下：錯誤訊息，必須動態的從網址列的 Query String 取得，顯示在主畫面中。



必須滿足的運作流程：

1. 使用者在首頁，輸入帳號、密碼。
2. 使用者點擊登入，使用 **POST** 方法連線至【驗證功能網址】，驗證網址的後端程式取得使用者輸入的帳號、密碼，並進行以下驗證程序。
 - a. 如果帳號、密碼任一為空，代表輸入有誤，將使用者導向到【失敗頁面網址】，並顯示「請輸入帳號、密碼」
 - b. 如果帳號、密碼都是 **test**，代表驗證成功，將使用者導向到【成功頁面網址】
 - c. 如果帳號、密碼任一錯誤，代表驗證失敗，將使用者導向到【失敗頁面網址】，並顯示「帳號、或密碼輸入錯誤」

要求三:記錄使用者狀態

延續要求二，進一步的使用 Session 在後端追蹤、記錄使用者狀態。關於 Session 的使用可以參考 [Flask Session](#) 或任何其他文件。

登出功能網址：<http://127.0.0.1:3000/signout>

連線方法：GET

成功頁面網址：<http://127.0.0.1:3000/member/>

參考設計如下：



必須滿足的運作流程：

1. 驗證成功後，在後端 Session 中記錄使用者狀態為 **已登入**，並導向【成功頁面網址】
2. 在成功頁面網址中加入登出連結，使用者點擊，連線到【登出功能網址】，在後端 Session 中紀錄使用者狀態為 **未登入**，並導向【首頁】
3. 在【成功頁面網址】的後端程式，檢查使用者狀態，使用者狀態若為 **未登入**，必須直接導向到【首頁】，不顯示登入後的網頁內容。