

# 服飾購物網站

## Julia's Clothe Store

**【摘要】**以服飾購物網站為主，販賣商品均為女性美式復古辣妹風，並在右下角有設計 Ai 聊天機器人的視窗，可點擊圖標詢問機器人相關服飾網站的問題。此網站除了前端，還包括了後端，所以可以真的註冊並登入購選購，登入後也可以添加喜好項目，資料庫會記住用戶訊息，下次使用時並不會丟失原本選購的東西。

### 1. 前言

自從上大學以來，我一直對電子商務充滿了濃厚的興趣，尤其是服飾行業。作為一名喜歡穿搭的女性，我總是夢想著有一天開設自己的服飾店，專門售賣我喜歡的美式復古辣妹風服飾。這個期末專題不僅是我在 Python 課程中的重要項目，也是我實現電商副業夢想的第一步。

在設計這個服飾購物網站的過程中，我希望能夠結合前沿的技術和用戶友好的設計，為顧客提供一個便捷、愉快的購物體驗。為了達到這個目標，我特別設計了一個以 Coze 內嵌的 Ai 聊天機器人，放置在網站的右下角。Coze 機器人剛好是我在人工智慧導論另一堂課程所學到的東西，於是突發奇想也用來這分專題網站裡。

此外，剛好這學期在機械系學到了前、後端的息航關知識，我也致力於為這個網站開發一個全面的後端系統，使用戶不僅可以註冊和登錄，還能夠添加和管理他們的喜好項目。通過數據庫的支持，用戶的信息和購物車內容將被持久保存，下次使用時不會丟失原本選購的商品，提升用戶的購物體驗和滿意度。

總之，這個專題不僅是我在電子商務的第一步，雖然沒有真的上線，但為未來的發展打下堅實的基礎，更統整了我整個學期在各個課程所學的內容。

### 2. 運用技術

#### ● 前端技術

1. HTML: 用於創建網頁的基本結構和內容。

2. CSS: 用於設計和美化網頁的外觀和佈局，確保網站具有美觀的視覺效果。

3. JavaScript: 用於實現網頁的交互功能，使網站更加動態和用戶友好。

#### ● 後端技術

1. Python: 作為主要的編程語言，用於編寫後端邏輯和處理服務器端的任務。

2. Flask: 一個輕量級的 Python web 框架，用於構建後端服務，處理 HTTP 請求和響應，管理路由，連接數據庫等。

#### ● 數據庫

SQLite: 一個輕量級的嵌入式關係數據庫，用於存儲用戶數據、商品信息、購物車內容等。SQLite 易於集成，適合開發和測試階段的使用。

#### ● AI 聊天機器人

Coze 聊天機器人: 用於實現網站的 AI 聊天功能，放置在網站的右下角，幫助用戶解答關於網站及服飾的問題，提升用戶體驗。

### 3. 功能介紹

操作影片連結：<https://youtu.be/RxTTqhMEOwo>

#### 用戶註冊與登錄

The image displays two side-by-side screenshots of a website's user interface. Both screenshots feature a dark blue header bar with the text "the Store" on the left and navigation links "HOME", "ABOUT", and "SHOP" on the right. Below the header, there is a shopping cart icon.

The left screenshot shows the "Login" form. It has three input fields: "Name", "Password", and "Captcha". Below these fields is a CAPTCHA image showing the numbers "19-1111" and a "更换验证码" (Change CAPTCHA) button. At the bottom is a large red "登入" (Login) button. A small link "还未注册吗? 注册" (Not registered? Register) is located just below the button.

The right screenshot shows the "Sign up" form. It also has three input fields: "Name", "Password", and "Email". Below the "Email" field is a "验证码" (Captcha) input field. At the bottom is a large red "註冊" (Register) button. A small link "还未注册吗? 登录" (Not registered? Login) is located just below the button.

圖一

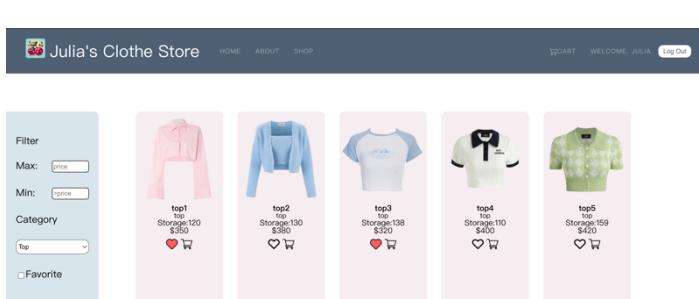
圖二

用戶首次使用時，可以通過簡單的註冊流程創建賬戶，輸入用戶名、電子郵件地址和密碼等基本信息，並會提供訊息引導使用者設置安全性較強的密碼（如圖二）。註冊完成後，用戶可以登入，在登入畫面我特別加上了驗證碼的功能（如圖一），登入完使用這些憑據登錄網站，訪問更多專屬功能，如添加商品到購物車、收藏喜好商品等。

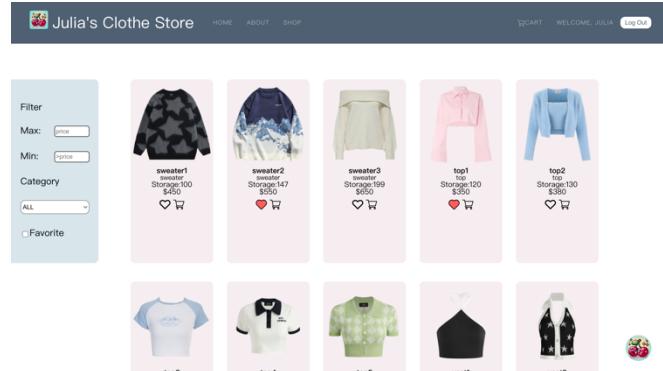
為了確保用戶信息的安全性，我們採用了 Flask 框架進行後端處理，並結合 SQLite 數據庫來存儲用戶的註冊信息。在用戶體驗方面，我們注重界面的簡潔和易用性。註冊和登錄頁面的設計遵循用戶友好原則，採用直觀的表單佈局和清晰的提示信息，幫助用戶順利完成操作。此外，我們還加入了密碼強度檢測功能，引導用戶設置安全性較高的密碼，增強賬戶的安全防護。

通過用戶註冊與登錄模塊，用戶可以獲得個性化的購物體驗，系統也能為每個用戶建立獨立的檔案，記錄其購物偏好和歷史數據。這不僅有助於提升用戶滿意度，還為後續的數據分析和個性化推薦奠定了基礎。

### 商品展示與分類



圖三



圖四

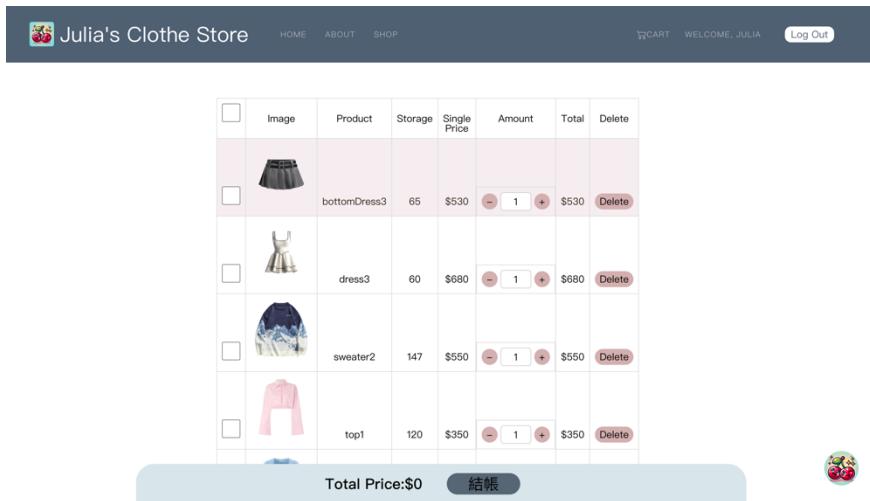
商品展示與分類是服飾購物網站的核心功能之一（如圖三、圖四），旨在幫助用戶方便地瀏覽和查找心儀的商品。在主頁上，我們設計了一個直觀且美觀的商品展示區域，每個商品都配有高清圖片、詳細描述、價格和庫存信息。用戶可以通過點擊商品卡片，進入商品詳情頁，查看更詳細的信息和更多圖片。

為了提升用戶的瀏覽體驗，我們引入了多種商品分類和篩選功能。用戶可以根據商品類別（如上衣、裙子、褲子等）快速找到所需的商品。此外，網站還提供價格區間篩選功能，用戶可以設置最小和最大價格，過濾出符合預算的商品。這些篩選和分類功能通過 JavaScript 實現，動態更新頁面內容，確保用戶操作的即時反饋。

在技術實現上，我們使用 HTML 和 CSS 進行頁面佈局和樣式設計，確保商品展示頁面的美觀性和響應性。JavaScript 負責處理用戶的篩選請求，並通過

Ajax 與後端的 Flask API 進行數據交互，從而實現商品數據的動態加載和更新。後端 API 從 SQLite 數據庫中獲取商品數據，並以 JSON 格式返回前端，確保數據傳輸的高效性和安全性。

## 購物車管理



The screenshot shows a shopping cart management interface for 'Julia's Cloth Store'. At the top, there is a navigation bar with links for HOME, ABOUT, SHOP, a shopping cart icon labeled 'CART', a welcome message 'WELCOME, JULIA', and a 'Log Out' button. Below the navigation bar is a table displaying four items in the cart:

	Image	Product	Storage	Single Price	Amount	Total	Delete
<input type="checkbox"/>		bottomDress3	65	\$530	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	\$530	<input type="button" value="Delete"/>
<input type="checkbox"/>		dress3	60	\$680	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	\$680	<input type="button" value="Delete"/>
<input type="checkbox"/>		sweater2	147	\$550	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	\$550	<input type="button" value="Delete"/>
<input type="checkbox"/>		top1	120	\$350	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	\$350	<input type="button" value="Delete"/>

At the bottom left, it says 'Total Price:\$0'. At the bottom right, there is a small user profile icon.

圖五

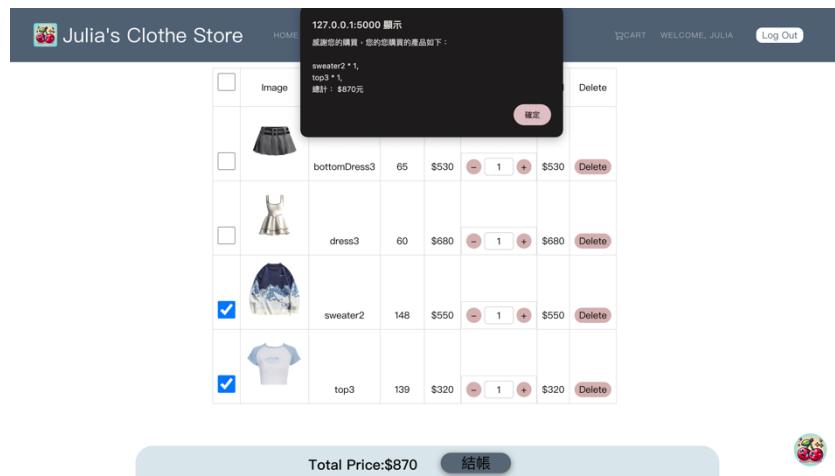
購物車管理功能是電子商務網站的關鍵模塊之一，它為用戶提供了方便的購物體驗，允許用戶在瀏覽商品時將心儀的商品添加到購物車，並在最終結算前進行查看和修改。我們的購物車管理系統設計簡潔、操作便捷，用戶可以隨時查看購物車中的商品，調整商品數量或刪除不需要的商品。（如圖五）

在技術實現方面，我們使用 JavaScript 實現購物車的前端交互，通過 DOM 操作實時更新購物車頁面。用戶點擊「加入購物車」按鈕時，JavaScript 會將商品信息存儲在瀏覽器的本地存儲中，同時更新購物車圖標中的商品數量。購物車頁面則從本地存儲中讀取商品信息，生成商品列表，並提供數量調整和刪除功能。

後端部分由 Flask 框架負責處理購物車的持久化存儲和數據同步。用戶登錄後，購物車數據會同步到 SQLite 數據庫中，確保用戶可以在不同設備上訪問同一個購物車內容。這種設計不僅提高了用戶體驗的連續性，還增強了數據的安全性和可靠性。

購物車頁面的設計考慮到了用戶的操作習慣，提供了清晰的商品信息展示，包括商品圖片、名稱、價格和庫存情況。用戶可以通過「+」和「-」按鈕或是直接輸入想購買的數量，系統會自動更新總價並實時顯示在頁面底部。刪除功能也十分便捷，用戶只需點擊「刪除」按鈕，即可將商品從購物車中移除。

## 結算系統

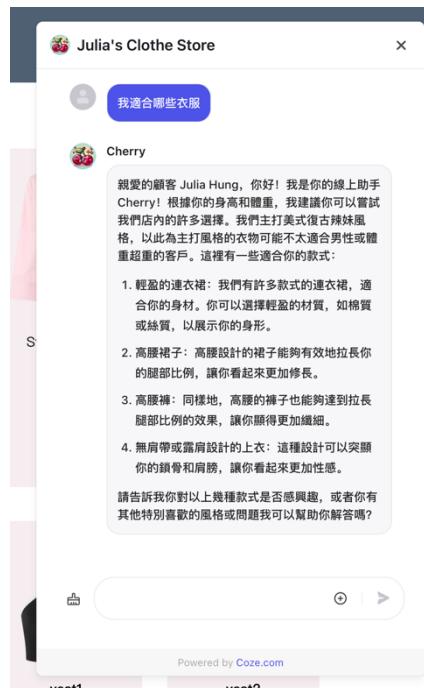


圖六

用戶在購物車頁面確認點擊需要購買的商品後，可以點擊「結帳」按鈕結帳，就會跳出 alert 顯示用戶購買的物品及總金額等，確認後，購買的商品就會從購物車消失而在商品頁該購買商品的庫存也會依照購買對應的數量去見少庫存量。（如圖六）

技術上，結算系統的前端部分採用 HTML 和 CSS 進行頁面設計，JavaScript 處理用戶輸入和表單驗證。後端由 Flask 框架負責處理使用者購物車的商品顯示及庫存的刪減。

## AI 聊天機器人



圖七

```
<script src="https://sf-cdn.coze.com/obj/unpkg-va/flo<br><script><br>    new CozeWebSDK.WebChatClient({<br>        config: {<br>            bot_id: '7340254462394679297',<br>        },<br>        componentProps: {<br>            title: `Julia's Cloth Store`,<br>            icon: 'https://i.meee.com.tw/xw3f1Xx.png',<br>        },<br>    });<br>    </script>
```

圖八

為了進一步提升用戶體驗，我們在網站中集成了 AI 聊天機器人，提供智能的在線智能客服服務（如圖七）。AI 聊天機器人位於網站的右下角，用戶可以點擊圖標與機器人進行對話，獲取關於商品、訂單、配送等方面的幫助。聊天機器人可以快速響應用戶的常見問題，提供實時的解決方案，減少用戶等待時間，提高服務效率。

技術實現上，我們採用了 Coze 聊天機器人技術，雖然這個技術在 Ai 人工智慧導論就有學過，但並沒有部署到網頁過，所以在這方面我又特別驗救了一下用法然後嵌入我的前端頁面（如圖八），實現與用戶的互動。聊天機器人的對話邏輯和知識庫存儲在後端，由 Flask 框架負責處理用戶的對話請求，並返回相應的回答。通過機器學習和自然語言處理技術，聊天機器人能夠理解用戶的提問，並提供準確和有用的回答。

## 4. 挑戰與解決方案

### 挑戰一：商品數量按鈕無法正常使用

在開發過程中，我遇到了一個關鍵問題，即商品數量按鈕無法正常工作。這個問題的根源在於，資料庫中的商品 ID 與加入購物車後的商品順序 ID 不一致，這導致在處理商品數量變更時無法正確匹配到相應的商品。

#### 解決方案：

為了解決這個問題，我需要對商品 ID 的處理進行更細緻的判斷。首先，我在前端使用 `getElementById` 和 `querySelector` 等 JavaScript 方法來獲取和操作 DOM 元素。這樣可以確保能夠精確定位到每個商品的數量按鈕，並進行相應的操作。

此外，我在後端設計了一個映射機制，將資料庫中的商品 ID 與購物車中的商品順序 ID 進行對應。這樣，每次用戶對商品數量進行操作時，系統都能夠通過映射正確找到對應的商品並更新其數量。在進行數量變更操作時，我還加入了對數據的校驗，確保數量的增減不會超出庫存範圍，並及時更新前端顯示的總價。

### 挑戰二：照片加載失敗與資料庫連線異常

在開發和測試過程中，我常常遇到照片加載失敗和資料庫連線異常的問題。經過仔細調查，發現這些問題大部分是由於 Flask 中的拼寫錯誤與資料庫中的不一致所導致的。

#### 解決方案：

為了解決這個問題，我進行了全面的代碼檢查和優化，確保 Flask 代碼中的每一個變量名稱、字段名稱與資料庫中的名稱完全一致。我還加入了詳細的錯誤日誌記錄功能，當系統發生錯誤時，能夠精確記錄錯誤信息，幫助我快速定位和修復問題。此外，我對資料庫連接進行了優化，確保連接的穩定性和可靠性。

#### 挑戰三：前端排版錯誤

在商品頁和分類頁載入過程中，我發現排版經常出現問題，兩個頁面的排版不一致。經過排查，發現問題是由於前端 HTML 代碼中缺少了一個結尾標籤，導致整個頁面的排版混亂。

#### 解決方案：

為了解決這個問題，我仔細檢查並修正了前端 HTML 代碼，確保所有的標籤都正確閉合，並且層級結構清晰。

通過這些挑戰的解決，我不僅提升了網站的功能和性能，也在實際操作中積累了寶貴的經驗，為未來的開發工作奠定了良好的基礎。

#### 挑戰四：navbar 的 CSS 樣式影響當前頁面內容

在開發過程中，我發現 navbar 的 CSS 樣式對當前頁面的內容物產生了不良影響，導致頁面元素顯示錯誤或佈局紊亂。

#### 解決方案：

為了解決這個問題，我對 navbar 的 CSS 進行了重構和優化，使用更具體的選擇器和嵌套結構來避免樣式衝突。此外，我引入了 CSS 模塊化的方法，將 navbar 的樣式與其他頁面元素分離，確保各個部分的樣式互不干擾。在開發過程中，我還通過瀏覽器的開發者工具進行實時調試，確保每個頁面的佈局和樣式都能正確顯示。

## 5. 結論與展望

在這次專題中，我的技術能力得到了顯著的提升。從前端的 HTML、CSS、JavaScript，到後端的 Python Flask，我掌握了完整的網站開發技術，並學會了如何將這些技術應用到實際項目中。這不僅提高了我的編程能力，也讓我對網站開發的整個流程有了更加全面的認識。同時，在開發過程中遇到的各種技術挑戰，如樣式衝突和功能實現等，讓我不斷思考和尋找解決方案，進一步鍛煉了我的問題解決能力。

在這個專題的過程中，我最大的收穫是對電子商務網站開發的全方位理解和實踐經驗。從需求分析、設計到最終的開發，每一個環節都讓我學到了很多寶貴的知識和技能。此外，我也深刻體會到持續學習和不斷探索的重要性。技術日新月異，只有不斷更新自己的知識儲備，才能在未來的職業生涯中立於不敗之地。

總之，這個專題是我進入電子商務領域的第一步，也是未來發展的起點。儘管還有很多需要改進和完善的地方，但我相信，這次經驗將成為我未來學習和工作的堅實基礎，激勵我在技術道路上不斷前行。這次專題的完成讓我對未來充滿了信心和期待，我會繼續努力，不斷提升自己的技術能力，為實現更大的目標而奮鬥。