

# 書客來線上書店系統 - 完整操作手冊

## 目錄

書客來線上書店系統 - 完整操作手冊 .....	1
目錄 .....	1
系統概述 .....	3
1.1 系統簡介 .....	3
1.2 主要功能 .....	3
1.3 技術特色 .....	3
系統架構 .....	4
2.1 文件結構 .....	4
2.2 資料庫架構 .....	4
2.3 系統流程圖 .....	4
安裝與設置 .....	6
3.1 系統需求 .....	6
3.2 安裝步驟 .....	6
3.3 初始化設置 .....	6
用戶功能操作指南 .....	8
4.1 會員註冊 .....	8
4.2 會員登入 .....	10
4.3 忘記密碼 .....	11
4.4 購物車管理 .....	13
4.5 訂單查詢與管理 .....	14
管理員功能操作指南 .....	16
5.1 資料庫管理工具 .....	16
5.2 資料庫狀態監控 .....	16
5.3 用戶管理 .....	16
5.4 訂單管理 .....	17
5.5 資料備份與恢復 .....	17
5.6 系統維護 .....	18
開發者指南 .....	20

6.1 程式碼結構.....	20
6.2 API 使用說明.....	21
6.3 自訂開發.....	22
6.4 部署指南.....	24
故障排除.....	24
7.1 常見問題.....	24
7.2 除錯工具.....	26
7.3 效能問題.....	27
7.4 瀏覽器相容性.....	27
技術規格.....	28
8.1 系統需求.....	28
8.2 技術棧.....	28
8.3 資料庫規格.....	29
8.4 API 文件.....	30
8.5 安全性考量.....	31
更新日誌.....	32
版本 1.3.0 (2025-06-03) .....	32
版本 1.2.0 (2025-05-28) .....	32
版本 1.1.0 (2025-05-27) .....	32
版本 1.0.0 (2025-05-26) .....	33
技術支援.....	34
聯絡資訊.....	34
常用資源.....	34

---

# 系統概述

## 1.1 系統簡介

書客來線上書店系統是一個完整的電商平台，專為書籍銷售而設計。系統採用前端 HTML/CSS/JavaScript 技術棧，使用 IndexedDB 作為客戶端資料庫，提供完整的用戶註冊、登入、購物車、訂單管理等功能。

## 1.2 主要功能

### 用戶功能

- 會員註冊與登入
- 忘記密碼重置
- 商品瀏覽與搜尋
- 購物車管理
- 訂單查詢與管理
- 個人資料管理

### 管理功能

- 用戶管理
- 訂單管理
- 資料庫管理
- 系統統計
- 資料匯入匯出

## 1.3 技術特色

- 無後端設計: 使用客戶端資料庫，簡化部署
  - 響應式設計: 支援桌面、平板、手機等多種裝置
  - 現代化介面: 使用現代 CSS 設計和動畫效果
  - 完整的錯誤處理: 友善的錯誤提示和處理機制
  - 安全性考量: 密碼強度檢測、用戶身份驗證
-

# 系統架構

## 2.1 文件結構

書客來線上書店/

- |— index.html                   # 首頁
- |— login.html                 # 登入頁面
- |— register.html            # 註冊頁面
- |— ordersearch.html        # 訂單查詢頁面
- |— shopping.html            # 購物車頁面
- |— admin-tools.html        # 管理工具頁面
- |— js/
  - |— database.js            # 資料庫管理系統
- |— css/
  - |— (內嵌在 HTML 中)      # 樣式表
- |— README.md                # 說明文件

## 2.2 資料庫架構

### IndexedDB 資料表

1. **users** (用戶表)
  - 主鍵: email
  - 索引: userId, email
  - 欄位: userId, email, password, name, phone, address, registrationDate, lastLoginDate, isActive
2. **orders** (訂單表)
  - 主鍵: orderNumber
  - 索引: userEmail, orderDate, status
  - 欄位: orderNumber, userEmail, items, totalAmount, paymentMethod, shippingAddress, orderDate, status, statusText
3. **carts** (購物車表)
  - 主鍵: userEmail
  - 欄位: userEmail, items, lastUpdated

## 2.3 系統流程圖

用戶訪問 → 首頁 → 登入/註冊 → 瀏覽商品 → 加入購物車 → 結帳 → 訂單管理



資料庫初始化 → 用戶驗證 → 資料存取 → 狀態管理 → 錯誤處理

---

# 安裝與設置

## 3.1 系統需求

### 最低需求

- 支援 HTML5 的現代瀏覽器
- JavaScript 啟用
- IndexedDB 支援
- 1GB 可用磁碟空間

### 建議需求

- Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
- 2GB RAM
- 網路連線（用於載入外部資源）

## 3.2 安裝步驟

1. 下載系統文件
2. # 解壓縮系統文件到本地目錄
3. unzip 書客來線上書店.zip
4. cd 書客來線上書店
5. 啟動本地伺服器
6. # 使用 Python 啟動簡單 HTTP 伺服器
7. python -m http.server 8000
- 8.
9. # 或使用 Node.js
10. npx http-server -p 8000
11. 瀏覽器訪問
12. 打開瀏覽器，訪問: <http://localhost:8000>

## 3.3 初始化設置

1. 首次訪問系統
  - 系統會自動初始化 IndexedDB 資料庫
  - 創建必要的資料表結構
  - 載入測試資料（開發環境）
2. 測試帳號

- 一般用戶: test@bookstore.com / 123456
- 管理員: admin@bookstore.com / admin123

### 3. 資料庫管理工具

- 訪問: <http://localhost:8000/admin-tools.html>
  - 可進行資料庫操作和系統管理
-

# 用戶功能操作指南

## 4.1 會員註冊

步驟說明：

### 1. 進入註冊頁面

- 點擊首頁右上角「加入會員」
- 或直接訪問 register.html

### 2. 填寫基本資料

- 電子郵件 (必填): 作為登入帳號使用
- 密碼 (必填): 至少 6 個字元，系統會顯示強度指示
- 確認密碼 (必填): 必須與密碼一致
- 姓名 (必填): 真實姓名
- 手機號碼 (選填): 用於訂單通知
- 地址 (選填): 用於配送地址

### 3. 密碼設置要求

- 最少 6 個字元
- 建議包含大小寫字母、數字和特殊符號
- 系統會即時顯示密碼強度 (弱/中/強)

### 4. 同意服務條款

- 必須勾選「同意服務條款」才能註冊
- 可選擇是否接收電子報

### 5. 完成註冊

- 點擊「立即註冊」按鈕
- 系統驗證資料無誤後完成註冊
- 自動跳轉到登入頁面

注意事項：

- 每個電子郵件只能註冊一次
- 密碼不可與電子郵件相同
- 手機號碼建議填寫真實號碼以便接收通知

系統操作範例：



## 加入書客來會員

享受更多購書優惠與專屬服務

### 基本資料

電子郵件 \*

M1329007@cgu.edu.tw

此電子郵件將作為您的登入帳號

密碼 \*

.....



確認密碼 \*

.....



密碼長度至少6個字元

姓名 \*

賴靈瑞

手機號碼

0901120057

用於訂單通知（選填）

地址

Queensland Brain Institute, Building 79, The University of Queensland Brisbane QLD 4072 Australia

用於配送地址（選填）

- ☒ 我已閱讀並同意 [服務條款](#) 和 [隱私權政策](#)
- ☒ 我同意接收書客來的電子報與行銷資訊（可隨時取消訂閱）

 立即註冊

已經是會員了？ [立即登入](#)

## 4.2 會員登入

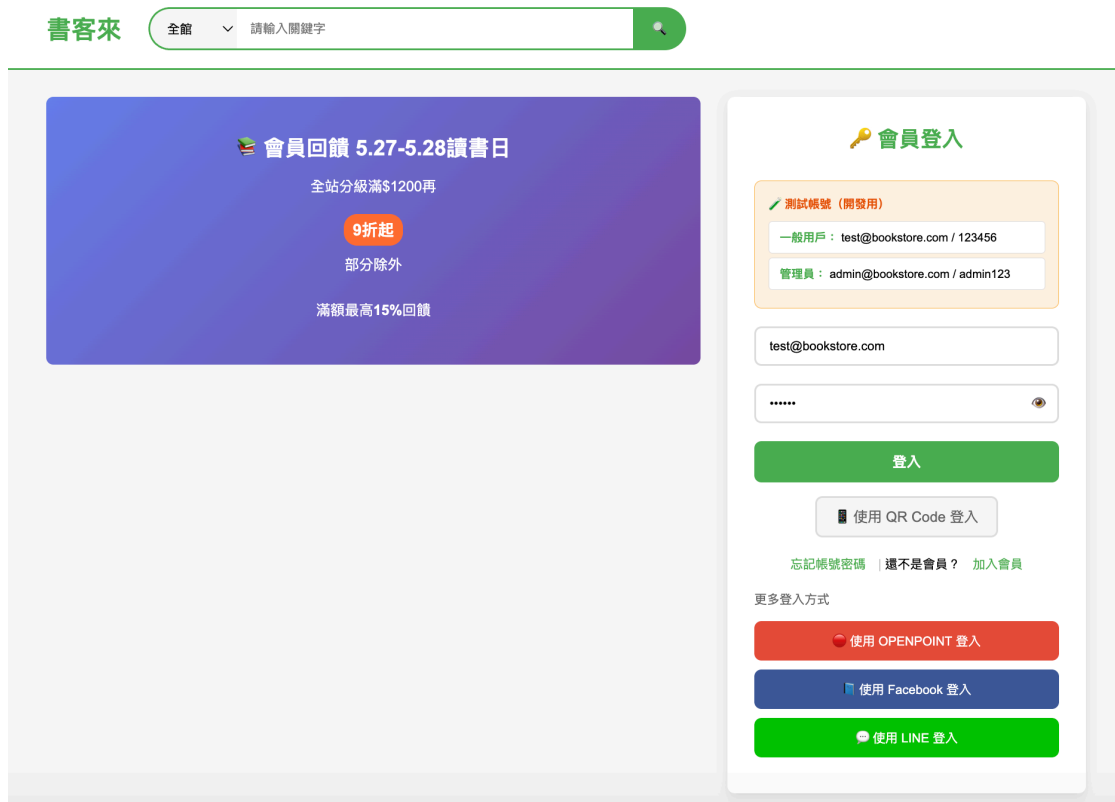
步驟說明：

1. 進入登入頁面
  - 點擊首頁右上角「會員登入」
  - 或直接訪問 `login.html`
2. 輸入登入資訊
  - 電子郵件：註冊時使用的信箱
  - 密碼：註冊時設定的密碼
3. 登入方式
  - 一般登入：輸入帳號密碼
  - 測試帳號：點擊預設測試帳號快速填入
  - QR Code 登入：(開發中)
  - 第三方登入：Facebook、LINE、OPENPOINT (開發中)
4. 登入後功能
  - 自動同步購物車資料
  - 記錄登入狀態
  - 跳轉到原本要訪問的頁面

密碼顯示功能：

- 點擊密碼框右側的眼睛圖示可切換密碼顯示/隱藏

系統操作範例：



## 4.3 忘記密碼

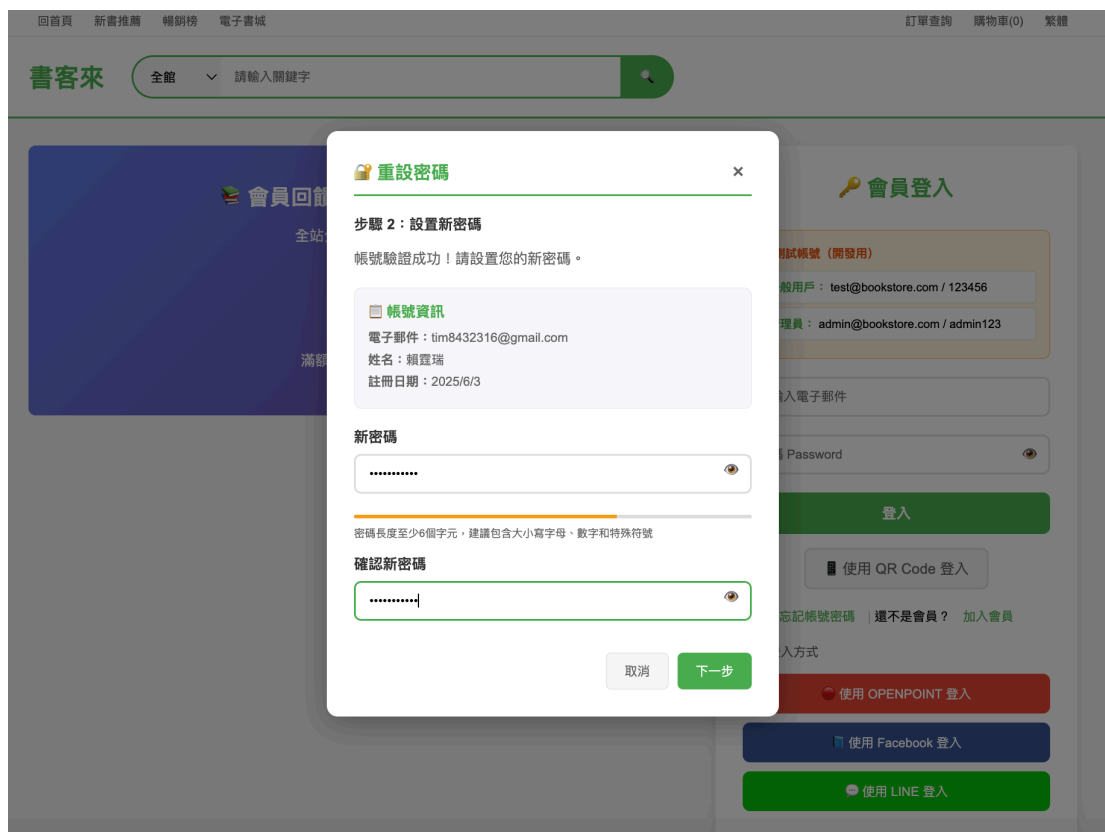
完整重置流程：

1. 啟動密碼重置
  - 在登入頁面點擊「忘記帳號密碼」
  - 彈出密碼重置視窗
2. 步驟一：驗證電子郵件
  - 輸入註冊時使用的電子郵件
  - 系統驗證帳號是否存在
  - 確認帳號狀態是否正常
3. 步驟二：設置新密碼
  - 顯示帳號基本資訊確認身份
  - 輸入新密碼 (至少 6 個字元)
  - 確認新密碼 (必須相同)
  - 即時顯示密碼強度指示
4. 步驟三：完成重置
  - 密碼更新成功
  - 可立即使用新密碼登入

安全性機制：

- 只有註冊用戶才能重置密碼
- 檢查帳號是否被停用
- 密碼強度要求與註冊相同

系統操作範例：



## 4.4 購物車管理

基本操作：

### 1. 查看購物車

- 點擊右上角「購物車」圖示
- 顯示當前購物車商品數量

### 2. 新增商品到購物車

- 在商品頁面點擊「加入購物車」
- 選擇商品數量
- 系統自動更新購物車

### 3. 修改購物車

- 調整商品數量
- 移除不需要的商品
- 即時計算總金額

### 4. 購物車同步

- 登入用戶：購物車資料保存在資料庫
- 未登入用戶：購物車資料保存在本地
- 登入時自動合併本地和雲端購物車



系統操作範例：

[回首頁](#) [新書推薦](#) [暢銷榜](#) [電子書城](#) 歡迎，測試用戶 [登入](#) [訂單查詢](#) [繁體](#)

書客來 全館 每日口袋99

因應近期台灣出口法規修訂，部分商品涉及原產地或稅關限制，暫不提供寄送至美國的服務。

優惠訊息  
5/27-5/28讀書日全站分級滿\$1200再9折起(部分除外)，滿額最高15%回饋

購物明細		免費贈品	點數回饋	下次再買清單	注意事項	
<input type="checkbox"/>	商品明細	優惠價/點數兌換價	數量	小計	庫存	變更明細
<input checked="" type="checkbox"/>	 財富的普通常識 (紀念典藏版) 66折特惠中	\$ 330 \$ 500	- 3 +	\$ 990	有	<a href="#">下次再買</a> <a href="#">刪除</a>
<input checked="" type="checkbox"/>	 明智殺介的奔走【博客來特別版】 79折優惠	\$ 363 \$ 460	- 1 +	\$ 363	有	<a href="#">下次再買</a> <a href="#">刪除</a>

[整批放入下次再買](#) [整批刪除](#) [添加測試商品](#) [同步購物車](#) [清空購物車](#)

共 2 項商品，數量 4 個  
金額/小計：NT\$ 1,353 元 [回首頁](#) [海外結帳](#) [國內結帳](#)

重要提醒

- 購物車中的商品會自動保存到您的帳戶中，可在不同裝置間同步。
- 商品總件數大於或多於15項，可能不適用『7-11取貨』。(說明)

## 4.5 訂單查詢與管理

訂單查詢功能：

1. 進入訂單查詢
  - 登入後點擊右上角「訂單查詢」
  - 或訪問 [ordersearch.html](#)
2. 查看訂單列表
  - 顯示所有訂單記錄
  - 包含訂單編號、時間、金額、狀態等資訊
3. 訂單篩選
  - 按狀態篩選：全部/處理中/已出貨/已完成
  - 按時間篩選：最近一週/一月/三月/一年
4. 訂單統計
  - 總訂單數量
  - 累計消費金額
  - 已完成/處理中訂單數量

訂單詳細操作：

1. 查看訂單詳情
  - 點擊訂單編號查看完整資訊
  - 包含商品清單、付款方式、配送地址等
2. 下載發票
  - 點擊「發票」按鈕（開發中）
3. 取消訂單
  - 處理中的訂單可以取消
  - 確認後狀態變更為「已取消」

訂單狀態說明：

- 處理中：訂單已建立，正在處理
- 已出貨：商品已發貨，運送中
- 已完成：訂單完成，商品已送達
- 已取消：訂單已被取消

系統操作範例：

書客來

全部

每日口袋99



[書客來](#) > [會員專區](#) > [訂單查詢](#)

交易紀錄

訂單查詢

[退換貨與開立收據](#)

[發票/折價券查詢](#)

[下次預購清單](#)

[關懷閱讀訂閱](#)

產品品項管理

[電子書庫](#)

[有聲書庫](#)

[數位商品庫查詢管理](#)

[下載使用紀錄](#)

會員資料/帳戶管理

[個人資料管理/地址設定](#)

[會員分級資訊](#)

訂單查詢

1    \$989    0    1  
 總訂單數    總金額    已完成    處理中

狀態

全部

處理中

已出貨

已完成

時間

最近一週

最近一月

最近三月

最近一年

顯示您的所有訂單記錄。如需查詢更久前的訂單資料，請聯繫客服。電子書訂單請到[電子書庫](#)查詢。

訂單編號	訂購時間	付款方式	訂單金額	商品資訊	處理狀態	操作
BK20250603366501	2025/6/3 上午 9:16:06	超商取貨付款	NT\$ 989	Lloyd's Register of Yachts 1897 等 3 項商品    總計 3 件	處理中	<div>詳情</div> <div>發票</div> <div>取消</div>

# 管理員功能操作指南

## 5.1 資料庫管理工具

訪問管理工具：

1. 開啟瀏覽器訪問 `admin-tools.html`
2. 系統會顯示警告提示此為開發工具
3. 等待資料庫連線完成

主要功能區域：

## 5.2 資料庫狀態監控

狀態指示卡：

- 資料庫狀態：顯示連線狀態（已連線/載入中/錯誤）
- 用戶總數：系統中註冊用戶數量
- 訂單總數：系統中訂單數量
- 總銷售額：累計銷售金額

操作按鈕：

- 重新整理狀態：更新所有統計資料
- 初始化測試資料：創建預設測試用戶和訂單
- 匯出資料：將所有資料匯出為 JSON 文件
- 匯入資料：從 JSON 文件匯入資料（開發中）
- 清空資料庫：刪除所有資料並重新初始化

## 5.3 用戶管理

載入用戶列表：

1. 點擊「載入用戶列表」按鈕
2. 系統顯示所有註冊用戶資訊

用戶列表包含：

- 用戶 ID
- 電子郵件



- 姓名
- 註冊時間
- 帳號狀態 (活躍/停用)
- 最後登入時間

**用戶管理操作：**

- **創建測試用戶：**手動新增測試帳號
- **查看用戶詳情：**檢視完整用戶資訊
- **停用/啟用帳號：**管理用戶狀態

## 5.4 訂單管理

**載入訂單列表：**

1. 點擊「載入訂單列表」按鈕
2. 系統顯示所有訂單記錄

**訂單列表包含：**

- 訂單編號
- 用戶郵件
- 訂單金額
- 付款方式
- 訂單狀態
- 訂購時間
- 商品數量

**訂單管理操作：**

- **創建測試訂單：**手動新增測試訂單
- **查看訂單詳情：**檢視完整訂單資訊
- **更新訂單狀態：**修改訂單處理狀態
- **訂單統計分析：**檢視銷售統計

## 5.5 資料備份與恢復

**資料匯出：**

1. 點擊「匯出資料」按鈕
2. 系統生成包含所有用戶和訂單的 JSON 文件

3. 自動下載到本地電腦
4. 文件命名格式: bookstore\_data\_export\_YYYY-MM-DD.json

**匯出資料格式：**

```
{  
  "exportDate": "2025-06-03T10:30:00.000Z",  
  "version": "1.0",  
  "users": [...],  
  "orders": [...]  
}
```

**資料匯入：**（開發中）

- 選擇 JSON 格式的備份文件
- 系統驗證文件格式
- 確認匯入操作
- 恢復資料並重建索引

## 5.6 系統維護

**清空資料庫：**

1. 點擊「清空資料庫」按鈕
2. 輸入確認文字：「清空資料庫」
3. 系統刪除所有資料
4. 自動重新初始化基本結構
5. 載入預設測試資料

**操作日誌：**

- 所有管理操作都會記錄在日誌中
- 顯示操作時間、類型和結果
- 協助除錯和問題追蹤

**注意事項：** ⚠️ 管理工具具有強大的權限，請謹慎操作 ⚠️ 清空資料庫操作無法復原，請先備份資料 ⚠️ 生產環境請勿暴露管理工具給一般用戶

系統操作範例：

書客來資料庫管理工具

回到書店

登入頁面

註冊頁面

購物車

開發工具警告：此頁面僅供開發和測試使用。在生產環境中，請勿暴露此管理工具給一般用戶。所有資料庫操作都會影響系統狀態，請謹慎使用。

資料庫狀態

已連線

資料庫運作正常

用戶總數

3

已註冊的會員數量

訂單總數

2

系統中的訂單數量

總銷售額

\$2,454

累計銷售金額

資料庫操作

重新整理狀態

初始化測試資料

匯出資料

匯入資料

清空資料庫

用戶管理

載入用戶列表

創建測試用戶

訂單管理

載入訂單列表

創建測試訂單

操作日誌

[啟動] 資料庫管理工具已載入

[2025/6/3 上午9:00:38] [INFO] 管理工具初始化完成

[2025/6/3 上午9:00:38] [INFO] 正在檢查資料庫狀態...

[2025/6/3 上午9:00:38] [SUCCESS] 狀態更新完成 - 用戶：3，訂單：2，銷售額：\$2,454

# 開發者指南

## 6.1 程式碼結構

資料庫管理類 (database.js)：

```
class BookstoreDatabase {  
    constructor()          // 初始化資料庫  
    init()                 // 建立資料庫連線  
    registerUser()         // 用戶註冊  
    loginUser()            // 用戶登入  
    getUser()              // 獲取用戶資料  
    updateUser()           // 更新用戶資料  
    resetUserPassword()    // 重置密碼  
    createOrder()          // 創建訂單  
    getUserOrders()        // 獲取用戶訂單  
    saveCart()             // 保存購物車  
    getCart()              // 獲取購物車  
    // ... 其他方法  
}
```

主要管理類：

```
// 登入管理器  
class LoginManager {  
    handleSubmit()         // 處理登入表單  
    syncCartData()         // 同步購物車資料  
    showError()            // 顯示錯誤訊息  
    showSuccess()          // 顯示成功訊息  
}  
  
// 註冊管理器  
class RegisterManager {  
    handleSubmit()         // 處理註冊表單  
    validateForm()         // 表單驗證  
    checkPasswordStrength() // 密碼強度檢查  
}
```

```
// 忘記密碼管理器
class ForgotPasswordManager {
  nextStep()          // 進入下一步
  validateNewPassword() // 驗證新密碼
  displayUserInfo()    // 顯示用戶資訊
}
```

```
// 訂單管理器
class OrderManager {
  loadUserOrders()    // 載入用戶訂單
  filterOrders()      // 篩選訂單
  updateOrderDisplay() // 更新顯示
}
```

## 6.2 API 使用說明

### 用戶相關 API：

```
// 註冊新用戶
const user = await bookstoreDB.registerUser({
  email: 'user@example.com',
  password: 'password123',
  name: '用戶姓名',
  phone: '0912345678',
  address: '用戶地址'
});

// 用戶登入
const user = await bookstoreDB.loginUser('email', 'password');

// 獲取用戶資料
const user = await bookstoreDB.getUser('email');

// 更新用戶資料
const updatedUser = await bookstoreDB.updateUser('email', {
  name: '新姓名',
  phone: '新電話'
});
```

// 重置密碼

```
const user = await bookstoreDB.resetUserPassword('email', 'newPassword');
```

### 訂單相關 API：

// 創建訂單

```
const order = await bookstoreDB.createOrder({
  userEmail: 'user@example.com',
  items: [...],
  totalAmount: 1000,
  paymentMethod: 'credit_card',
  shippingAddress: '配送地址'
});
```

// 獲取用戶訂單

```
const orders = await bookstoreDB.getUserOrders('user@example.com');
```

// 更新訂單狀態

```
const order = await bookstoreDB.updateOrderStatus(
  'orderNumber',
  'shipped',
  '已出貨'
);
```

### 購物車相關 API：

// 保存購物車

```
await bookstoreDB.saveCart('user@example.com', cartItems);
```

// 獲取購物車

```
const cartItems = await bookstoreDB.getCart('user@example.com');
```

// 清空購物車

```
await bookstoreDB.clearCart('user@example.com');
```

## 6.3 自訂開發

### 新增功能步驟：

#### 1. 設計資料結構

- 決定是否需要新的資料表
- 定義資料欄位和索引
- 2. 更新資料庫類
  - 在 database.js 中新增相關方法
  - 處理資料的增刪改查
- 3. 建立前端介面
  - 設計 HTML 結構
  - 添加 CSS 樣式
  - 實作 JavaScript 邏輯
- 4. 整合功能
  - 連接前端和資料庫
  - 添加錯誤處理
  - 測試功能運作

樣式自訂：

/\* 主要顏色變數 \*/

```
:root {  
  --primary-color: #4CAF50;  
  --secondary-color: #45a049;  
  --error-color: #e74c3c;  
  --success-color: #27ae60;  
  --warning-color: #f39c12;  
}
```

/\* 通用按鈕樣式 \*/

```
.btn {  
  padding: 12px 20px;  
  border: none;  
  border-radius: 8px;  
  cursor: pointer;  
  transition: all 0.3s;  
}
```

```
.btn-primary {  
  background-color: var(--primary-color);  
  color: white;  
}
```

## 6.4 部署指南

本地開發環境：

```
# 啟動本地伺服器  
python -m http.server 8000  
# 或  
npx http-server -p 8000
```

生產環境部署：

1. 靜態網站託管
  - GitHub Pages
  - Netlify
  - Vercel
  - Firebase Hosting
2. **Apache/Nginx 設定**
3. # Apache .htaccess
4. RewriteEngine On
5. RewriteCond %{REQUEST\_FILENAME} !-f
6. RewriteCond %{REQUEST\_FILENAME} !-d
7. RewriteRule ^(.\*)\$ index.html [QSA,L]
8. **HTTPS 設定**
  - 現代瀏覽器要求 HTTPS 才能使用 IndexedDB
  - 建議使用 SSL 憑證

性能優化：

- 壓縮 CSS 和 JavaScript 文件
  - 啟用瀏覽器快取
  - 使用 CDN 加速資源載入
  - 優化圖片大小和格式
- 

## 故障排除

### 7.1 常見問題

問題 1：資料庫初始化失敗



**症狀：**

- 頁面顯示「資料庫尚未準備就緒」
- 無法註冊或登入用戶
- 管理工具顯示連線失敗

**解決方案：**

1. 確認瀏覽器支援 IndexedDB
2. 檢查瀏覽器控制台是否有錯誤訊息
3. 清除瀏覽器快取和資料
4. 嘗試使用無痕模式開啟
5. 重新整理頁面等待資料庫載入

**問題 2：用戶無法登入**

**症狀：**

- 提示「找不到此用戶」或「密碼錯誤」
- 確認帳號密碼正確但仍無法登入

**解決方案：**

1. 檢查電子郵件是否正確（不分大小寫）
2. 確認密碼沒有多餘空格
3. 嘗試使用測試帳號登入
4. 使用忘記密碼功能重置密碼
5. 檢查資料庫中是否存在該用戶

**問題 3：購物車資料遺失**

**症狀：**

- 登入後購物車變空
- 重新整理頁面後購物車資料消失

**解決方案：**

1. 確認是否已登入會員
2. 檢查本地儲存空間是否已滿
3. 清除瀏覽器快取後重新添加商品

4. 檢查資料庫中購物車資料是否正常

#### 問題 4：訂單資料顯示異常

症狀：

- 訂單頁面無法載入
- 顯示「沒有符合條件的訂單」
- 訂單統計資料錯誤

解決方案：

1. 確認已登入正確的用戶帳號
2. 檢查篩選條件是否過於嚴格
3. 嘗試重新整理頁面
4. 使用管理工具檢查訂單資料完整性

## 7.2 除錯工具

瀏覽器開發者工具：

1. **Console (控制台)**
2. // 檢查資料庫狀態
3. `console.log(window.bookstoreDB);`
- 4.
5. // 查看所有用戶
6. `window.bookstoreDB.getAllUsers().then(console.log);`
- 7.
8. // 查看所有訂單
9. `window.bookstoreDB.getAllOrders().then(console.log);`
10. **Application (應用程式)**
  - IndexedDB: 查看資料庫結構和資料
  - Local Storage: 檢查本地儲存
  - Session Storage: 檢查會話儲存
11. **Network (網路)**
  - 檢查資源載入狀態
  - 確認外部檔案是否正常載入

資料庫管理工具：

- 使用 `admin-tools.html` 檢查系統狀態

- 查看操作日誌了解系統運作
- 匯出資料進行離線分析

### 7.3 效能問題

頁面載入緩慢：

1. 檢查網路連線速度
2. 清除瀏覽器快取
3. 關閉不必要的瀏覽器擴充功能
4. 確認硬體資源足夠

資料庫操作緩慢：

1. 檢查資料量是否過大
2. 清理過期資料
3. 重建資料庫索引
4. 分批處理大量資料

### 7.4 瀏覽器相容性

支援的瀏覽器：

- Chrome 60+
- Firefox 55+
- Safari 11+
- Edge 79+

不支援的功能：

- Internet Explorer (任何版本)
- 較舊版本的行動瀏覽器
- 停用 JavaScript 的瀏覽器

相容性檢查：

```
// 檢查 IndexedDB 支援
if (!window.indexedDB) {
    alert('您的瀏覽器不支援 IndexedDB，請升級瀏覽器版本');
}
```

```
// 檢查 LocalStorage 支援
if (!window.localStorage) {
    alert('您的瀏覽器不支援 LocalStorage 功能');
}
```

---

## 技術規格

### 8.1 系統需求

最低系統需求：

- CPU: 1GHz 單核心處理器
- RAM: 2GB
- 硬碟: 1GB 可用空間
- 瀏覽器: 支援 HTML5 和 ES6 的現代瀏覽器

建議系統需求：

- CPU: 2GHz 雙核心處理器
- RAM: 4GB
- 硬碟: 5GB 可用空間
- 瀏覽器: Chrome 90+, Firefox 88+, Safari 14+

### 8.2 技術棧

前端技術：

- HTML5: 語意化標記
- CSS3: 響應式設計、動畫效果
- JavaScript (ES6+): 現代 JavaScript 語法
- IndexedDB: 客戶端資料庫

開發工具：

- Chrome DevTools: 除錯和效能分析
- Visual Studio Code: 程式碼編輯
- Git: 版本控制
- HTTP Server: 本地開發伺服器

## 8.3 資料庫規格

### IndexedDB 設定：

- 資料庫名稱: BookstoreDB
- 版本: 1
- 大小限制: 依瀏覽器而定 (通常 > 250MB)

### 資料表結構：

-- users 表 (概念性 SQL)

```
CREATE TABLE users (  
    email VARCHAR(255) PRIMARY KEY,  
    userId VARCHAR(50) UNIQUE,  
    password VARCHAR(255),  
    name VARCHAR(100),  
    phone VARCHAR(20),  
    address TEXT,  
    registrationDate DATETIME,  
    lastLoginDate DATETIME,  
    isActive BOOLEAN  
);
```

-- orders 表 (概念性 SQL)

```
CREATE TABLE orders (  
    orderNumber VARCHAR(50) PRIMARY KEY,  
    userEmail VARCHAR(255),  
    items TEXT, -- JSON 格式  
    totalAmount DECIMAL(10,2),  
    paymentMethod VARCHAR(50),  
    shippingAddress TEXT,  
    orderDate DATETIME,  
    status VARCHAR(20),  
    statusText VARCHAR(50)  
);
```

-- carts 表 (概念性 SQL)

```
CREATE TABLE carts (  
    userEmail VARCHAR(255) PRIMARY KEY,
```

```
items TEXT, -- JSON 格式
lastUpdated DATETIME
);
```

## 8.4 API 文件

**BookstoreDatabase** 類別方法：

方法名稱	參數	回傳值	描述
init()	無	Promise<IDBDatabase>	初始化資料庫
registerUser(userData)	Object	Promise<User>	註冊新用戶
loginUser(email, password)	String, String	Promise<User>	用戶登入驗證
getUser(email)	String	Promise<User>	獲取用戶資料
updateUser(email, data)	String, Object	Promise<User>	更新用戶資料
resetUserPassword(email, password)	String, String	Promise<User>	重置用戶密碼
createOrder(orderData)	Object	Promise<Order>	創建新訂單
getUserOrders(email)	String	Promise<Order[]>	獲取用戶訂單
updateOrderStatus(orderNumber, status, statusText)	String, String, String	Promise<Order>	更新訂單狀態
saveCart(email, items)	String, Array	Promise<Cart>	保存購物車
getCart(email)	String	Promise<Array>	獲取購物車
getAllUsers()	無	Promise<User[]>	獲取所有用戶(管理)

方法名稱	參數	回傳值	描述
getAllOrders()	無	Promise<Order[]>	獲取所有訂單(管理)
checkDatabaseStatus()	無	Promise<Object>	檢查資料庫狀態

## 8.5 安全性考量

### 資料保護：

- 密碼以明文存儲（生產環境需要加密）
- 用戶資料僅存在客戶端
- 無伺服器端資料傳輸

### 輸入驗證：

- 電子郵件格式驗證
- 密碼長度檢查
- 表單欄位驗證
- XSS 防護 (HTML 編碼)

### 權限控制：

- 用戶只能存取自己的資料
  - 管理工具需要特殊權限
  - 會話管理和登入狀態檢查
-

# 更新日誌

## 版本 1.3.0 (2025-06-03)

### 新增功能：

- 完整的忘記密碼功能
- 三步驟密碼重置流程
- 即時密碼強度指示器
- 美觀的模態窗設計

### 功能改進：

- 優化資料庫事務處理
- 改善用戶註冊流程
- 增強錯誤處理機制
- 提升響應式設計

### 修復問題：

- 修復註冊時的事務衝突
- 解決密碼顯示切換問題
- 修正購物車同步機制

## 版本 1.2.0 (2025-05-28)

### 新增功能：

- 訂單查詢與管理系統
- 訂單篩選和統計功能
- 用戶專屬訂單頁面
- 訂單狀態管理

### 功能改進：

- 資料庫連動優化
- 真實訂單資料顯示
- 用戶體驗改善

## 版本 1.1.0 (2025-05-27)



### 新增功能：

- 資料庫管理工具
- 用戶和訂單管理
- 資料匯出功能
- 系統統計儀表板

### 功能改進：

- 資料庫初始化優化
- 事件監聽機制改善
- 錯誤處理強化

### 版本 1.0.0 (2025-05-26)

### 初始版本：

- 基本系統架構
  - 用戶註冊與登入
  - IndexedDB 資料庫
  - 響應式網頁設計
-

# 技術支援

## 聯絡資訊

### 開發團隊：

- 系統架構師: Claude AI Assistant
- 技術支援: 透過開發平台聯繫

### 問題回報：

- 系統錯誤: 提供詳細錯誤訊息和重現步驟
- 功能建議: 描述期望功能和使用場景
- 效能問題: 提供瀏覽器版本和系統環境

## 常用資源

### 官方文件：

- MDN Web Docs (IndexedDB): [https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB\\_API](https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API)
- Web APIs Reference: <https://developer.mozilla.org/en-US/docs/Web/API>

### 開發工具：

- Chrome DevTools: <https://developer.chrome.com/docs/devtools/>
- Firefox Developer Tools: <https://developer.mozilla.org/en-US/docs/Tools>

### 學習資源：

- JavaScript 教學: <https://javascript.info/>
- CSS 參考文件: <https://developer.mozilla.org/en-US/docs/Web/CSS>

---

**版權聲明：** © 2025 書客來線上書店系統。本文件包含系統操作和技術實作的詳細說明，僅供授權用戶使用。

**文件版本：** v1.3.0

**最後更新：** 2025 年 6 月 3 日

**適用系統版本：** 書客來線上書店 v1.3.0