**簡易社交網路平台**

**1. 系統簡介**

本系統是一個基於 **PHP + MySQL + Apache** 開發的簡易社交網路平台，透過 Docker 容器進行部署。用戶可以註冊帳號、登入系統，發佈動態貼文、按讚、留言、追蹤其他用戶，並管理個人貼文。系統提供直觀的介面和基礎社交互動功能。

**2. 系統功能**

**(1) 使用者管理**

* **註冊功能**：使用者可以註冊新帳號，輸入使用者名稱、Email 和密碼。
* **登入功能**：已註冊的用戶可以登入系統。
* **登出功能**：使用者可以安全地登出系統。

**(2) 動態牆功能**

* **發佈貼文**：使用者可以發佈文字內容及上傳圖片。
* **查看動態**：動態牆展示所有用戶的貼文，按照發佈時間倒序排列。
* **按讚功能**：用戶可以對貼文進行按讚，按第二次會取消按讚。
* **留言功能**：用戶可以對貼文進行留言。
* **編輯與刪除貼文**：發佈者可以編輯或刪除自己的貼文。

**(3) 追蹤功能**

* 用戶可以追蹤其他用戶，動態牆可顯示被追蹤用戶的貼文（未來擴展）。

**3. 技術架構**

**(1) 前端技術**

* **HTML**：頁面結構設計。
* **CSS**：提供響應式設計、按鈕樣式、貼文樣式等。

**(2) 後端技術**

* **PHP**：作為主要後端程式語言，負責處理用戶請求、資料操作及邏輯管理。
* **MySQL**：作為系統的資料庫，存儲用戶資料、貼文、按讚和留言等資料。
* **Apache**：提供 Web 服務，處理 HTTP 請求。

**(3) 容器化技術**

* **Docker**：將 PHP、MySQL、Apache 組合進行容器化管理，確保系統可以輕鬆部署與測試。
* **Docker Compose**：用於管理多個服務（如 web 和 db）。

**4. 系統操作流程**

**(1) 系統部署**

1. **啟動容器**：
   * 使用 docker-compose up -d 啟動 Apache 和 MySQL 容器。
2. **資料庫初始化**：
   * 匯入初始 SQL 文件，創建 users、posts、likes 和 comments 表。
3. **存取系統**：
   * 透過瀏覽器進入 http://localhost:8080。

**(2) 用戶操作**

1. **註冊與登入**：
   * 註冊新帳號並登入系統。
2. **發佈貼文**：
   * 使用表單輸入內容並上傳圖片。
3. **互動功能**：
   * 用戶可按讚、留言、追蹤其他用戶，並編輯或刪除個人貼文。
4. **登出**：
   * 安全登出系統。

**5. 資料庫設計**

**主要表格結構**

1. **users 表**：
   * 存儲使用者資訊。
2. CREATE TABLE users (
3. id INT AUTO\_INCREMENT PRIMARY KEY,
4. username VARCHAR(50) NOT NULL UNIQUE,
5. email VARCHAR(100) NOT NULL UNIQUE,
6. password VARCHAR(255) NOT NULL,
7. created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
8. );
9. **posts 表**：
   * 存儲貼文內容和圖片。
10. CREATE TABLE posts (
11. id INT AUTO\_INCREMENT PRIMARY KEY,
12. user\_id INT NOT NULL,
13. content TEXT NOT NULL,
14. image VARCHAR(255),
15. created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
16. FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE
17. );
18. **likes 表**：
    * 記錄用戶對貼文的按讚狀態。
19. CREATE TABLE likes (
20. id INT AUTO\_INCREMENT PRIMARY KEY,
21. post\_id INT NOT NULL,
22. user\_id INT NOT NULL,
23. created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
24. FOREIGN KEY (post\_id) REFERENCES posts(id) ON DELETE CASCADE,
25. FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE
26. );
27. **comments 表**：
    * 存儲用戶對貼文的留言。
28. CREATE TABLE comments (
29. id INT AUTO\_INCREMENT PRIMARY KEY,
30. post\_id INT NOT NULL,
31. user\_id INT NOT NULL,
32. comment TEXT NOT NULL,
33. created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
34. FOREIGN KEY (post\_id) REFERENCES posts(id) ON DELETE CASCADE,
35. FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE
36. );

**6. 系統特色**

* **簡單易用**：提供基本的社交互動功能，適合學習和展示。
* **容器化管理**：使用 Docker 技術，保證跨平台的一致性。
* **擴展性強**：未來可以加入更多功能，如通知系統、即時聊天等。

**7. 未來展望**

* **即時聊天功能**：加入 WebSocket 讓用戶能即時聊天。
* **個人檔案頁面**：提供用戶自我介紹和查看貼文的功能。
* **通知系統**：當用戶被按讚、留言或追蹤時發送通知。
* **圖片壓縮功能**：上傳圖片時自動進行壓縮，優化性能。

**8. 結論**

本系統作為一個簡易的社交網路平台，成功實現了用戶註冊、登入、貼文發佈及互動等基本功能，並通過容器化技術確保系統部署的靈活性。