**To-Do List**

**第四組**

**組員: 3B017154 羅治圳 (後端設計)**

**3B017024 蘇致安 (前端開發)**

**3B017043 呂祐岷 (報告製作)**

**目 錄**

1. **學習目標 2**
2. **To-Do List簡介 3**
3. **專案架構 3**
4. **環境介紹 4**
5. **前端開發工具 4**
6. **前端實作方法 5**
7. **後端開發工具 6**
8. **後端實作方法 6**
9. **系統流程圖 7**
10. **To-Do List使用方法 8**
11. **參考資料 12**
12. **學習目標**
13. **理解Web應用程式架構**
    * + - 清楚了解「路由 → 控制 → 模板渲染 → 資料庫操作」整個請求流程。
        - 透過 routes (/, /add, /edit, /task/<id>/delete 等) 體驗 URL routing 機制。
14. **熟悉 HTTP 請求與表單資料處理**

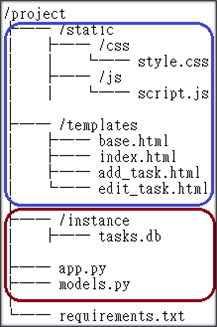
* 學習 GET 與 POST 請求的差異及使用時機。
* 使用 request.form 接收前端表單資料。

1. **學會使用 ORM 操作資料庫**

* 練習建立資料模型（Task class），理解ORM(Object Relational Mapping **物件關聯對映)與資料表的對應關係。**
* 操作 CRUD (Create, Read, Update, Delete) 全流程。

1. **To-Do List簡介**

什麼是 To-Do List ? To-Do List全名叫做待辦事項清單，是一種用於記錄和管理事件的表單，列出需要完成的任務或事件的清單後，可以幫忙建立和確認日常工作或個人目標，其中包括該事件的描述、截止日期、優先級與是否完成等資訊，用於幫助個人或團隊組織和管理工作或日常活動，以便更好地安排時間和完成任務。

1. **專案架構**

**圖1.專案架構圖**

整體架構分為前端與後端，其中HTML、CSS

與JavaScript等開發使用者介面的檔案為前端，

而使用Python、Flask進行資料與路由處理的檔

案、SQLite的資料庫為後端。

1. **環境介紹**

使用Windows作為開發時主要的作業系統，並使用Visual Studio Code作為主要開發環境。

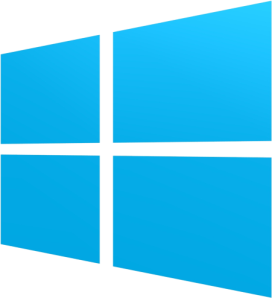
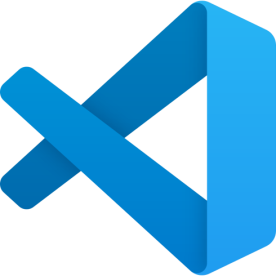


圖2 Windows作業系統

圖3 Visual Sttudio Code

1. **前端開發工具**
   1. 前端網頁結構使用HTML撰寫。
   2. CSS進行網頁美觀處理。
   3. JavaScript進行事件的狀態切換與刪除。
   4. Bootstrap作為主要UI框架。
   5. 透過Jinja實現頁面繼承。

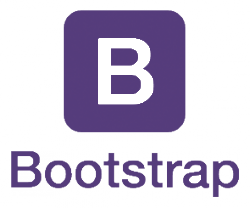


圖8 Jinja

圖7 Bootstrap

圖4 JavaScript

圖5 CSS

圖4 HTML

1. **前端實作方法**
   1. 資料庫初始化
   2. 任務管理功能

|  |  |  |
| --- | --- | --- |
| **頁面** | **說明** | **互動功能** |
| base.html | 繼承共通模板設計 | 響應式導行列 |
| index.html | 主頁面 | 任務狀態切換 |
| edit\_task.html | 任務編輯頁面 | 表單驗證  資料更新 |
| add\_task.html | 新增任務頁面 | 表單驗證、提交 |

1. **後端開發工具**
   1. 後端使用Python作為主要撰寫語言。
   2. 使用Flask進行路由建置、處理請求與回應。
   3. SQLite為資料庫，進行資料儲存與查詢。
   4. SQLAlchemy則為Flask提供了更直接的ORM操作，且方便整合配置和上下文管理。



圖10 Flask

圖 5

圖9 Python



圖12 SQLAlchemy

圖11 SQLite

1. **後端實作方法**
   1. 資料庫初始化
   2. 任務管理功能

|  |  |  |
| --- | --- | --- |
| **功能** | **說明** | **Route** |
| 顯示任務列表 | 依分類篩選或顯示全部 | / |
| 新增任務 | 填寫任務表單新增資料 | /add |
| 編輯任務 | 編輯已存在任務內容 | /edit/<int:task\_id> |
| 切換完成狀態 | 切換任務完成與否 | /task/<int:task\_id>/toggle |
| 刪除任務 | 刪除指定任務 | /task/<int:task\_id>/delete |

表 2

1. **系統流程圖**

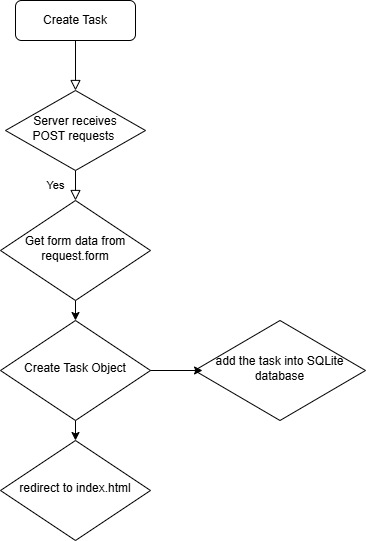
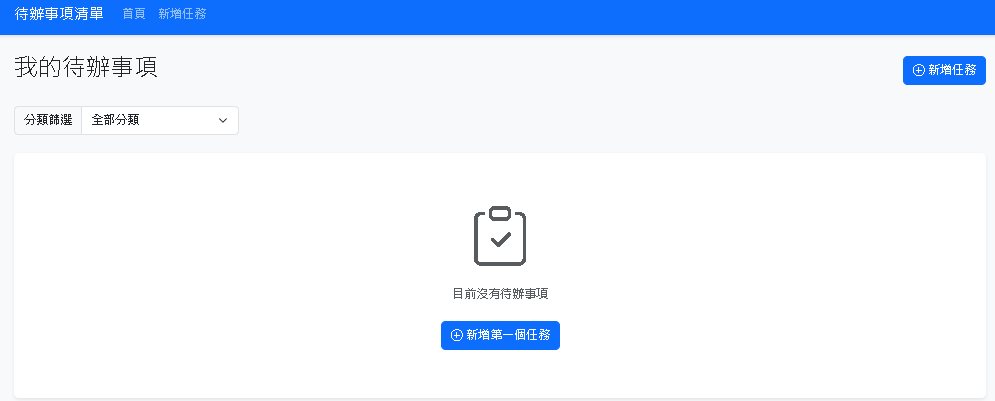
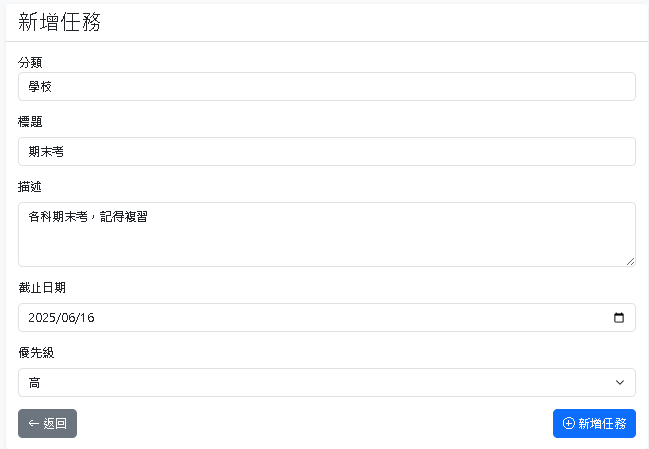


圖13

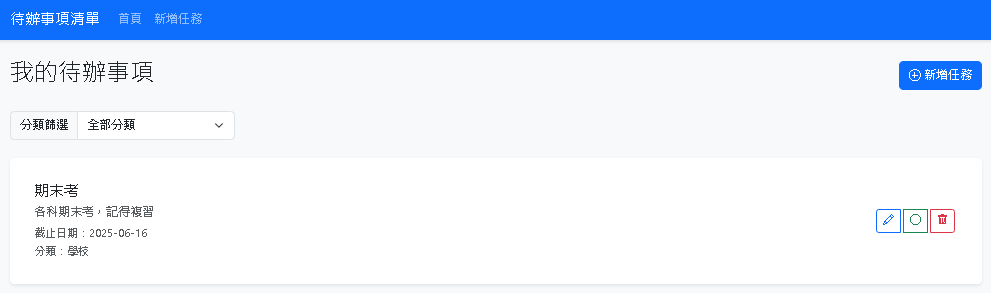
1. **To-Do List使用方法**
   1. 第一次使用，點擊按鈕即可生成第一件任務



* 1. 點擊按鈕後，進入新增任務介面，輸入任務的相關訊息，並按下新增任務，即可新增任務。

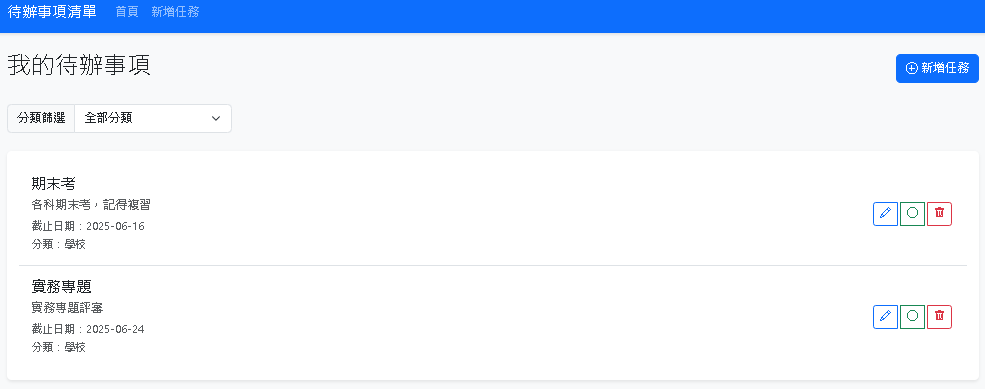
****

* 1. 填寫完成並按下新增後，可以看見待辦事項新增了一件任務。



* 1. 新增第一件任務之後，也可透過新增任務按鈕繼續新增任務。

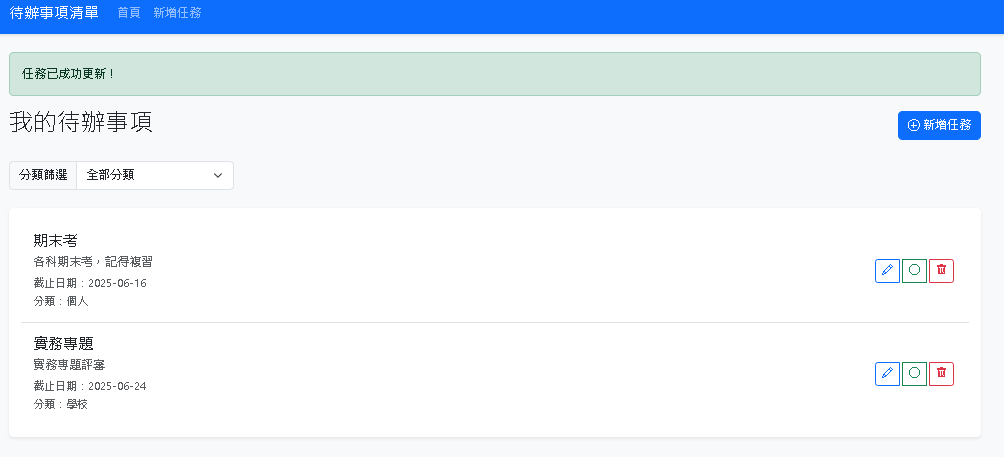


* 1. 再按下新增任務按鈕後，可以看到又有新的待辦事項生成了。
  2. 點擊編輯按鈕後，可以對任務的所有屬性進行修改，也可將該任務標註為是否完成，之後點擊儲存變更，就會將任務進行更新。

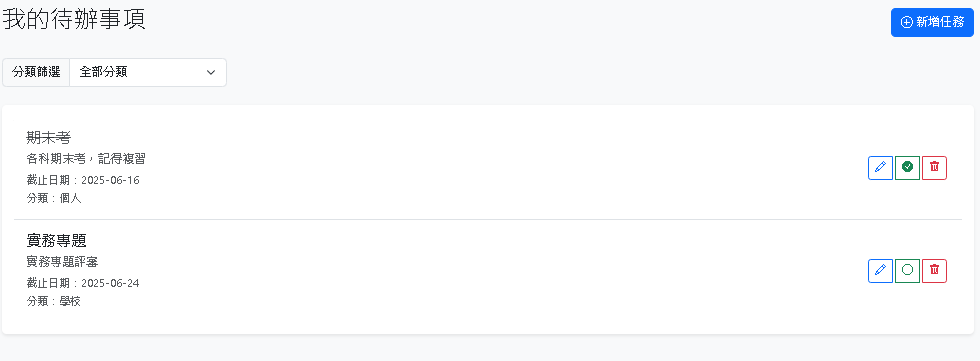


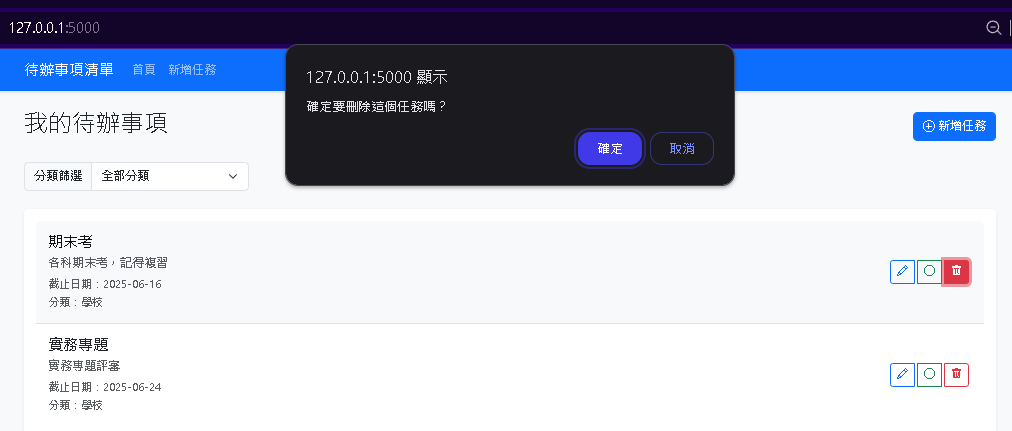
* 1. 儲存更新後，會跳出成功更新的通知，並且可以看到期末考

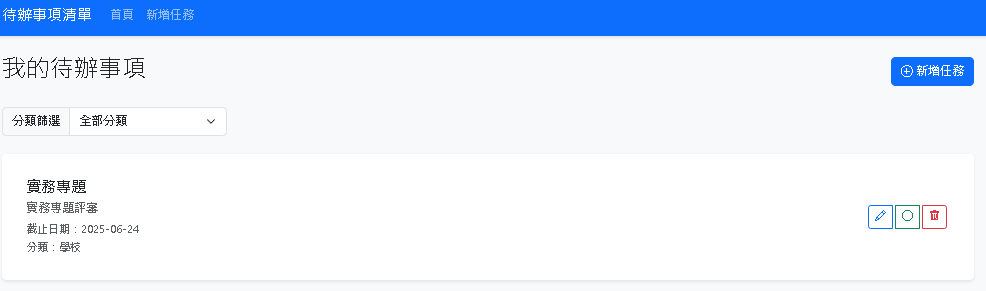
的分類更新為個人，編輯後的內容有確實更新。



* 1. 點擊編輯旁的按鈕，即可標註該任務已經完成，並且該任務的標題也會畫上刪除線。



* 1. 點擊刪除按鈕，便會出現提示視窗，確認是否刪除該任務。
  2. 按下確認後，回到首頁即可看到，剛剛刪除的任務已經消失在待辦事項中了。



1. **參考資料**

<https://hackmd.io/@peterju/S1FoLvgbkg\>

<https://hackmd.io/@peterju/SkpFlj40s>

<https://hackmd.io/@peterju/HJftWmE5kg>

<https://hackmd.io/@peterju/BJmEbXV9Jg>

<https://hackmd.io/@peterju/Hkpa1mVcye>

<https://hackmd.io/@peterju/r1-Pk7V51l>

<https://zh.wikipedia.org/zh-tw/Flask>

<https://zh.wikipedia.org/zh-tw/SQLite>

<https://zh.wikipedia.org/zh-tw/HTML>

<https://zh.wikipedia.org/zh-tw/CSS>

[https://zh.wikipedia.org/zh-tw/JavaScript](https://zh.wikipedia.org/zh-tw/HTML)