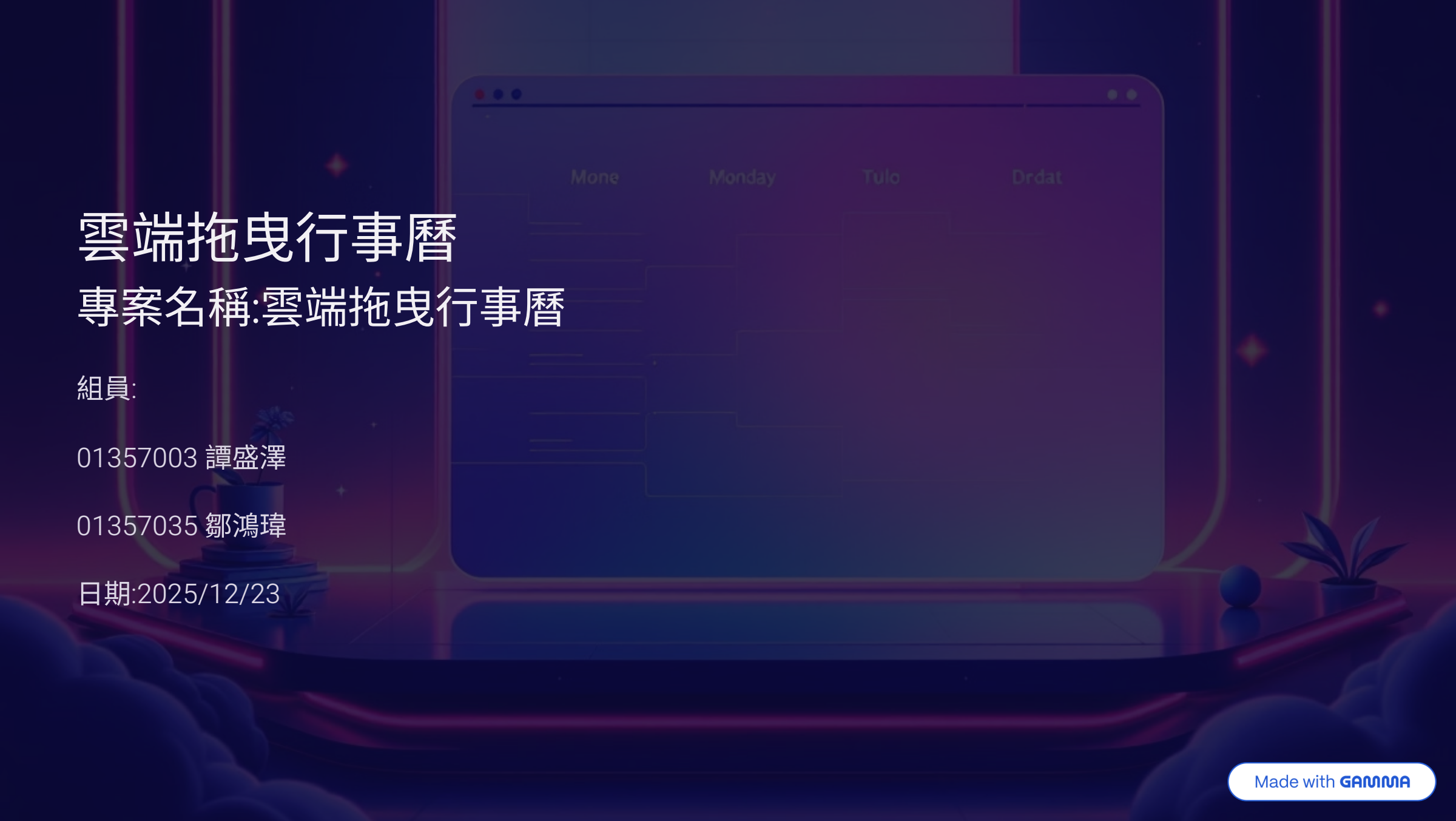


雲端拖曳行事曆

專案名稱:雲端拖曳行事曆

組員:

01357003 譚盛澤

01357035 鄒鴻瑋

日期:2025/12/23

創作動機

問題

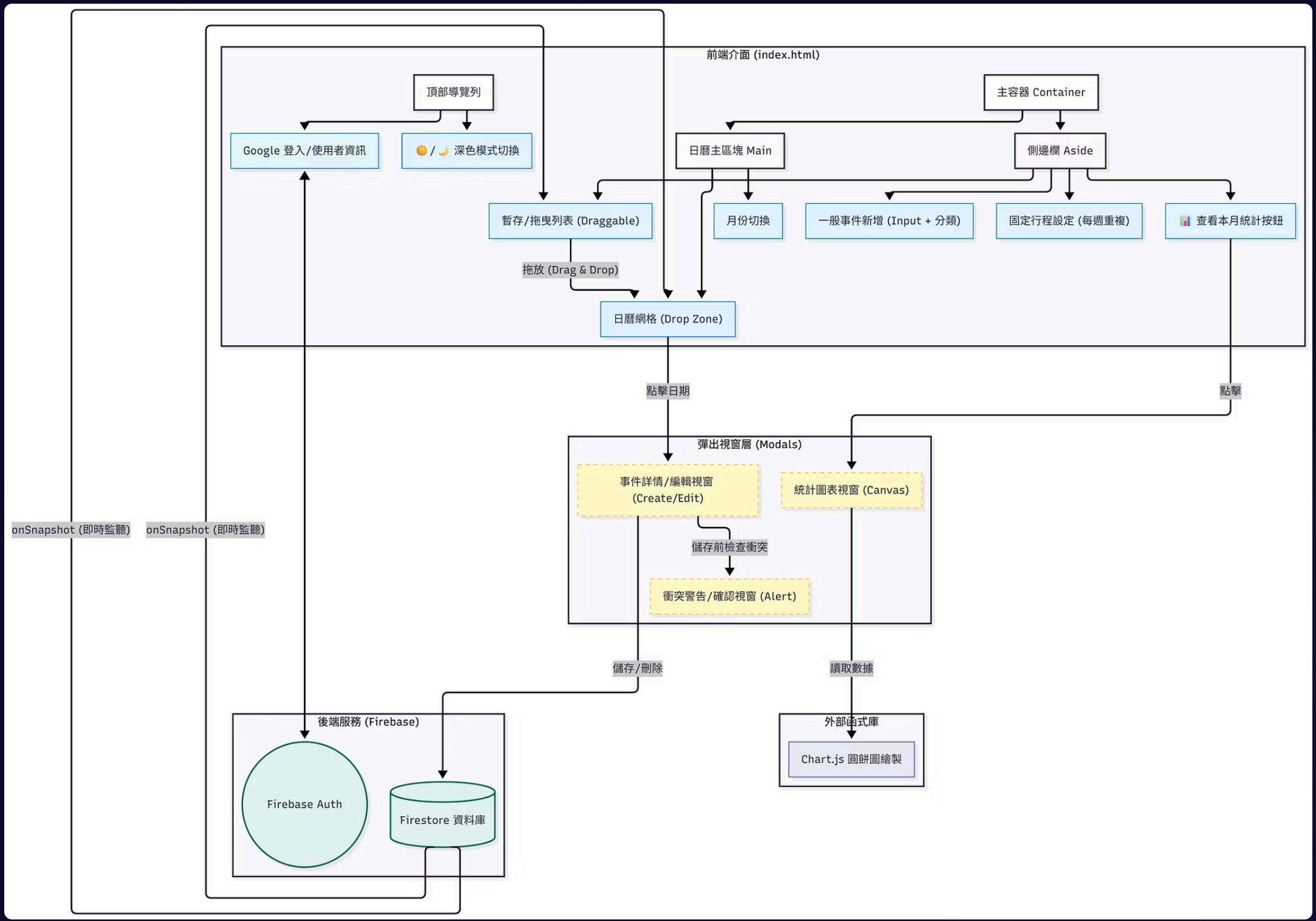
市面上的行事曆操作繁瑣,缺乏「便利貼」般的直覺拖曳感。

目標

- **直覺操作:**將「待辦事項」與「日曆」結合,用拖放 (Drag & Drop) 完成排程。
- **視覺享受:**採用 Glassmorphism (毛玻璃) 設計風格,提供深色/淺色模式切換。
- **數據驅動:**不只是紀錄,還能透過圖表分析時間分配 (工作、學習、個人)。

網站架構

Website Structure



單頁式應用 (SPA)

所有操作在同一頁面完成,無須跳轉。

核心區塊

- **Sidebar:**事件暫存區、每週固定行程管理、統計入。
- **Calendar Grid:**主要互動區,支援拖放與點擊編輯。
- **Firebase Backend:**負責使用者驗證與資料即時同步。

核心特色

01

雲端即時同步 (Real-time Sync)

- 使用 Firebase onSnapshot 監聽技術。
- A 電腦新增行程,B 手機無需重新整理立刻出現。

02

智慧衝突偵測 (Smart Conflict Detection)

- 新增或修改事件時,自動計算時間重疊。
- 演算法亮點:針對「每週固定行程」,若發生衝突,系統會詢問是否刪除整串行程。

03

資料視覺化 (Data Visualization)

- 繪製甜甜圈圖 (Doughnut Chart)。
- 自動計算本月「工作、學習、個人、重要」等分類的總時數佔比。

04

質感深色模式 (Dark Mode)

- 支援一鍵切換,自動記憶使用者偏好 (localStorage)。
- 針對深色背景優化了毛玻璃透明度與文字對比。

使用技術

前端 (Frontend)

- **HTML5 / CSS3:**Flexbox, CSS Grid, CSS Variables (主題切換)。
- **JavaScript (ES6+):**Async/Await, Modules (import/export), DOM 操作。
- **Chart.js:**動態圖表繪製。

後端與資料庫 (Backend as a Service)

- **Firebase Authentication:**Google 帳號一鍵登入。
- **Cloud Firestore:**NoSQL 資料庫,儲存 events (一般事件) 與 recurring_events (固定事件)。

script.js

```
import { initializeApp } from "https://www.gstatic.com/firebasejs/10.7.1/firebase-app.js";
import {
    getAuth,
    GoogleAuthProvider,
    signInWithPopup,
    signOut,
    onAuthStateChanged
} from "https://www.gstatic.com/firebasejs/10.7.1/firebase-auth.js";
import {
    getFirestore,
    collection,
    addDoc,
    updateDoc,
    deleteDoc,
    doc,
    query,
    where,
    onSnapshot
} from "https://www.gstatic.com/firebasejs/10.7.1/firebase-firestore.js";
```

技術難點與解決

難點 1:循環事件的單日修改

問題:設定了「每週一開會」,但某個週一想取消或改時間,怎麼辦?

解決:在資料庫設計 exceptions 陣列。當使用者刪除某天固定行程時,不刪除原始資料,而是將該日期推入 exceptions 陣列中隱藏顯示。

難點 2:拖曳後的資料狀態管理

問題:從側邊欄拖曳到日曆後,如何保留原始樣板但更新日曆狀態?

解決:區分 userEvents (側邊欄列表) 與 placedEvents (日曆上),拖曳時更新 Firestore 的 placedDates 欄位。

實際分工

01357003 譚盛澤

UI/UX 設計:HTML改動、CSS Glassmorphism 特效、深色模式配色設計、固定行程支援分類顏色顯示。

視覺化開發:實作統計圖表 Modal 與資料計算邏輯。

核心邏輯:改良時間衝突偵測演算法。

最重要的:做簡報。

01357035 鄒鴻瑋

前端:HTML 結構搭建

後端串接: Firebase 環境建置、Google 登入功能、Firestore 資料庫 CRUD。

核心邏輯:時間衝突偵測演算法、循環事件 (Recurring Events) 的例外處理邏輯。

資料流管理:實作 onSnapshot 即時監聽與資料同步。

未來展望

- 同步 Google Calendar 。
- 創建群組查看共同行事曆。
- 新增衝突暫存區。
- Line Bot 連動