**詳細功能與解說 (9/12 MVP版本)**

這份文件定義了妳在9月12日前**需要完成**的所有功能範疇，不多也不少。

**1. 核心體驗：**

* **無需登入**：網站不設任何會員註冊或登入系統，所有使用者皆為匿名訪客。
* **單向傳遞**：僅支援「丟信」與「撿信」的單向情感流動，不設回覆功能。

**2. 丟信者 (The Sender) 流程：**

* **觸發**：使用者在首頁點擊「寫一封信」按鈕。
* **書寫介面**：
  + 提供一個簡潔、沉浸式的書寫頁面。
  + 使用者可輸入信件內容（建議字數上限，如300字）。
  + （可選功能）提供3-5個預設的**分類標籤**讓使用者選擇（如：給過去的信、無名的感謝、小小的秘密），增加儀式感。
* **投遞儀式**：點擊「投入時光之海」按鈕後，觸發一個**關鍵動畫**（如信紙捲起，裝入瓶中，漂向遠方），並顯示成功訊息。
* **冷卻機制 (Cooldown)**：投遞成功後，觸發2小時的冷卻計時。計時結束前，使用者無法再次丟信。此計時器使用瀏覽器的 localStorage 儲存，僅針對單一裝置的單一瀏覽器有效。

**3. 撿信者 (The Receiver) 流程：**

* **觸發**：使用者在首頁點擊「撿一個瓶中信」按鈕。
* **拾取儀式**：觸發一個**關鍵動畫**（如一個瓶子漂到岸邊，點擊後瓶塞開啟），增加期待感。
* **閱讀介面**：動畫結束後，信件內容以優雅的排版呈現。
* **短暫留存**：信件內容僅在當前頁面顯示。一旦使用者關閉分頁、重新整理或離開頁面，此封信即**永久消失**，無法再次讀取，以此體現「一期一會」的核心概念。
* **冷卻機制 (Cooldown)**：成功撿信後，觸發12小時的冷卻計時。計時結束前，無法再次撿信。

**4. 技術底層：**

* **前端**：專注於HTML, CSS, JavaScript，實現所有介面、動畫與互動。
* **後端**：建立簡單的API，包含兩個主要端點：
  + POST /api/letters：接收新信件並存入資料庫。
  + GET /api/letters/random：從資料庫中隨機撈取一封未被讀取過的信件。
* **資料庫**：僅需一個letters資料表，欄位包含id, content, category\_tag, created\_at, is\_delivered (布林值)。

**全新AI協作式甘特圖 (Gantt Chart) 2.0**

這個新時程的核心是「**人機協作，並行開發**」。我們會大量利用AI完成重複性的基礎工作，讓妳能把寶貴的時間專注在**整合、優化和創意**上。

**第一週 (8/14 - 8/21)：AI詠唱！風格確立與素材生成**

**目標：不做任何一行手動程式碼，而是產出所有「構成元素」和「設計藍圖」。**

* **AI 任務**：
  + **視覺風格 (OpenArt/Dreamina)**：生成網站的關鍵視覺。
    - *Prompt範例*: "A lonely glass bottle washed ashore on a misty beach at dawn, digital painting, soft pastel colors, tranquil and melancholic mood." (一個在黎明時分被沖上迷霧海灘的孤獨玻璃瓶，數位繪畫，柔和的粉彩色調，寧靜憂鬱的氛圍。)
    - *Prompt範例*: "A beautiful piece of old, weathered parchment paper, with burnt edges, blank, top-down view, realistic texture." (一張美麗、風化的舊羊皮紙，邊緣有燒焦痕跡，空白，俯視角，逼真的紋理。)
  + **氛圍音樂 (Suno)**：生成網站的背景音樂和音效。
    - *Prompt範例 (背景音樂)*: "A quiet, looping background sound of gentle ocean waves crashing on a sandy shore, with a hint of a lonely piano melody, peaceful, slightly melancholic, 2 minutes loop."
    - *Prompt範例 (音效)*: "Sound effect of a cork popping out of a glass bottle." (軟木塞從玻璃瓶中彈出的音效) / "Sound effect of writing with a quill pen on paper." (鵝毛筆在紙上書寫的音效)
* **妳的任務 (設計與整合)**：
  + [ ] 在Figma中，將AI生成的圖片整合成一個**視覺情緒板 (Mood Board)**。
  + [ ] 基於這些素材，**完成所有頁面的UI設計**。這一步妳是總監，決定最終的樣子。
* **🏆 里程碑 #1 (8/21前)**：**Figma高保真設計稿完成，且所有需要的圖片、音樂、音效素材都已生成完畢。**

**第二週 (8/22 - 8/28)：AI造骨！前後端架構生成與串接**

**目標：讓AI產出網站的骨架，妳負責把它們接起來。**

* **AI 任務**：
  + **前端骨架 (Trae/ChatGPT/Copilot)**：把妳的Figma設計截圖給AI，或是用文字描述。
    - *Prompt範例*: "Based on this UI design, generate the HTML and CSS code for a responsive webpage. Use a mobile-first approach. Create three pages: home.html, write.html, read.html."
  + **後端骨架 (Trae/ChatGPT/Copilot)**：明確指示AI妳需要的功能。
    - *Prompt範例*: "使用 Python 的 Flask 框架，建立一個簡單的 API。包含兩個路徑：1. POST /api/letters，用來接收 JSON 格式的信件內容並存入 SQLite 資料庫。2. GET /api/letters/random，用來從資料庫隨機讀取一封信並回傳。請提供完整的 Python 程式碼和所需的 database schema。"
* **妳的任務 (整合與除錯)**：
  + [ ] 在VS Code中，整理AI生成的HTML/CSS/JS前端檔案，以及Python後端檔案。
  + [ ] **親手執行後端程式，建立資料庫。**
  + [ ] **最關鍵的一步**：修改前端的JavaScript，讓它能夠成功呼叫妳本地運行的後端API。**這是妳本週的核心學習與挑戰**，專心把這件事做好。
* **🏆 里程碑 #2 (8/28前)**：**網站的本地版本可以跑起來了！** 妳可以在自己的電腦上，成功寫一封信，看到它存進資料庫，並能成功撿到一封信。

**第三週 (8/29 - 9/4)：手動注魂！動畫、音效與細節打磨**

**目標：AI的工作已基本完成，現在輪到妳這位「總監」為作品注入靈魂和魔法。**

* **AI 任務**：可以向AI請教特定效果的作法。
  + *Prompt範例*: "Give me a CSS animation code for an element to fade in and float up gently." (給我一個讓元素輕柔淡入並向上浮動的CSS動畫程式碼。)
* **妳的任務 (優化與創造)**：
  + [ ] **親手**為「丟信」和「開信」的過程加上精緻的**CSS/JavaScript動畫**。這是AI無法取代的「體感」設計。
  + [ ] **親手**將Suno生成的背景音樂和音效，在適當的時機（如頁面載入、按鈕點擊）觸發播放。
  + [ ] **親手**調整CSS，讓網站在手機和電腦上看起來都一樣完美（響應式設計RWD的最後一哩路）。
  + [ ] **親手**測試所有流程，確保沒有Bug。
* **🏆 里程碑 #3 (9/4前)**：**網站所有功能和體驗細節完成**。它現在看起來和感覺起來，都像一個有靈魂的完整作品。

**第四週 (9/5 - 9/12)：部署上線與完美發表**

**目標：將妳的作品呈現給全世界，並準備好妳的精彩報告。**

* **AI 任務**：
  + **部署指導 (Trae/ChatGPT/Copilot)**：
    - *Prompt範例*: "我有一個 Python Flask 後端和一個純 HTML/CSS/JS 前端，請給我一份詳細的步驟教學，說明如何將它免費部署到 Vercel 平台上。"
  + **簡報草稿 (Gamma)**：
    - *Prompt範"*: "為一個名為「時與光的漂流郵局」的網站專案製作一份發表簡報。專案理念是受日本漂流郵局啟發的匿名信件交換網站。大綱應包含：1. 專案靈感與理念 2. 網站功能與特色 3. 設計風格與技術實現 4. 未來展望。"
* **妳的任務 (總結與演練)**：
  + [ ] 根據AI的指導，**完成網站部署**，取得一個公開的網址和QR Code。
  + [ ] 使用Gamma或Canva，**美化妳的簡報**，放入妳的Figma設計圖、網站截圖和QR Code。
  + [ ] **完整地演練妳的報告至少3次**。確保流程順暢，時間控制在要求內。
* **🏆 里程碑 #4 (9/12前)**：**妳準備好了！** 網站已上線，簡報已完成，妳充滿信心，只待上台發光。

這個AI協作流程，將妳從繁重的程式碼勞動中解放出來，讓妳能真正專注於**創意、設計和體驗**。這是一條絕對可行且高效的路，讓我們開始吧！

**甘特圖 (Gantt Chart) 與里程碑 (Milestones) 1.0**

這是一份為期四週的衝刺計畫，目標是在9/12完成所有工作。

**第一週 (8/14 - 8/21)：企劃定案與視覺設計**

* **任務**：
  + [ ] **(1-2天)** 最後確認所有功能細節，繪製使用者流程圖。
  + [ ] **(5-6天)** **在Figma中完成所有頁面的UI/UX設計**。包含：首頁、書寫頁面、信件呈現頁面。決定好字體、配色、按鈕樣式與整體氛圍。
* **🏆 里程碑 #1 (8/21前)**：**Figma高保真設計稿完成**。妳應該要有一個可以點擊、可以展示給別人看，看起來像個真網站的完整設計原型。

**第二週 (8/22 - 8/28)：技術基建與後端開發**

* **任務**：
  + [ ] **(2-3天)** 建立前後端專案的開發環境，選擇妳熟悉的技術框架。
  + [ ] **(4-5天)** **完成後端API與資料庫的建置**。妳需要能透過API工具（如Postman）成功新增一封信到資料庫，並能成功隨機讀取一封信。
* **🏆 里程碑 #2 (8/28前)**：**後端API功能通過測試**。網站的「大腦和倉庫」已經準備就緒。

**第三週 (8/29 - 9/4)：前端核心功能開發**

* **任務**：
  + [ ] **(7天)** **將Figma設計稿轉換成真實網頁，並串接後端API**。這是工作量最大的一週。目標是完成最核心的功能：可以實際在網站上寫信、送出，並能撿到一封來自資料庫的真實信件。完成冷卻計時器的基本功能。
* **🏆 里程碑 #3 (9/4前)**：**網站核心功能(MVP)可操作**。整個「丟信->存入資料庫->從資料庫撿信->顯示」的流程可以順暢地在網站上運作。

**第四週 (9/5 - 9/12)：優化、部署與發表準備**

* **任務**：
  + [ ] **(3-4天) 魔法時刻 (Magic Moment)**：將省下來的時間全力投入到**動畫與音效**的優化上。這是讓網站從「能用」到「驚艷」的關鍵！
  + [ ] **(1-2天) 網站部署**：將妳的網站上線到一個公開的網址（可使用Vercel, Netlify, Heroku等免費平台），確保全世界都能訪問。
  + [ ] **(2-3天) 簡報製作與演練**：製作妳的發表簡報，內容需包含：專案簡介、**現場Demo (附上QR Code)**、功能解說、設計理念，以及最重要的\*\*「未來展望」\*\*（把妳們討論的登入、回覆系統放進來）。務必演練幾次！
* **🏆 里程碑 #4 (9/12前)**：**專案凍結！** 網站已上線，Figma設計定稿，發表簡報完成。妳已經準備好上台了！

**WEEK 1 (2025.08.14 – 2025.08.21**

根據我們之前訂下的時程，**第一週 (也就是從今天 8/14 開始到 8/21)** 的核心任務是： **【風格確立與素材生成】**

目標是在不做任何手動程式開發的情況下，先把網站的靈魂——也就是**視覺和聽覺**的素材全部準備好，並在Figma中完成設計藍圖。

**本週的具體行動清單 (Action List)**

妳今天的任務，就是可以開始化身為「藝術總監」，玩玩這些AI工具，為妳的「回憶港灣」定義出它的樣貌和聲音。

**1. 產出核心視覺 (使用 OpenArt / Dreamina)**

妳需要一些關鍵圖片來定義網站的風格，並直接用在妳的Figma設計稿和最終的網站上。

**試試看這些提示詞 (Prompt)：**

* **主視覺 - 溫暖港灣風：**

A quiet and safe harbor at twilight, gentle water, a single warm lantern glowing on an old wooden pier, peaceful and serene, digital painting, style of Studio Ghibli. (黃昏時分寧靜安全的港灣，平靜的水面，一盞溫暖的燈籠在舊木碼頭上發光，平和而寧靜，數位繪畫，吉卜力工作室風格。)

* **主視覺 - 孤寂療癒風：**

A glass bottle with a message inside, washed up on a misty, secluded beach cove, soft morning light, photorealistic, cinematic, tranquil mood. (一個裝有信的玻璃瓶，被沖上一個有霧的、僻靜的海灣，柔和的晨光，照片般逼真，電影感，寧靜的氛圍。)

* **素材 - 信紙：**

A beautiful piece of thick, textured watercolor paper, with soft, deckled edges, blank, top-down view, warm tone. (一張美麗、有紋理的厚水彩紙，邊緣有柔和的毛邊，空白，俯視角，暖色調。)

**2. 產出氛圍音樂與音效 (使用 Suno)**

聲音是沉浸感體驗的關鍵！

* **背景音樂 (希望能循環播放)：**

A calm, peaceful, and looping instrumental piece. Features gentle lapping water against a pier, a very distant and soft piano melody, and the occasional sound of a lone seagull. No beats, minimalist, suitable for background ambiance.

* + **中文可嘗試**: 一段平靜、可循環的背景音樂，以溫柔的海浪拍打碼頭聲為主，搭配非常遙遠且輕柔的鋼琴旋律，偶爾有海鷗的叫聲，無節拍，簡約，適合氛圍音樂。
* **關鍵音效：**
  + **點擊按鈕時**: Soft, magical, low-volume chime sound effect. (輕柔、神奇、低音量的鐘聲/風鈴音效。)
  + **信件送出時**: Sound effect of a piece of paper being folded, then a gentle "swoosh" sound. (紙張被折疊的音效，然後是一聲輕柔的「嗖」聲。)
  + **打開瓶中信時**: Sound effect of a cork gently popping out of a glass bottle, followed by the sound of paper unrolling. (軟木塞輕輕從玻璃瓶中彈出的音效，接著是紙張展開的聲音。)

**妳的下一步：**

開始玩吧！不用追求一次就成功，多嘗試幾次不同的關鍵字組合，看看AI能給妳帶來什麼驚喜。當妳收集到一些滿意的圖片和音樂後，我們就可以進入下一步——把它們整合到妳的Figma設計稿裡，讓妳的「回憶港灣」第一次被具體地看見。

**子齊’s tips**

妳的老師給了妳一個**非常棒、非常主流、而且對妳目前狀況來說大大簡化了工作**的建議！

**「小白話」翻譯機：老師到底說了什麼？**

讓我們來解碼老師的話：

**1. 「前端用原生」**

* **這句話的意思是**：網站的「門面」（使用者看得到、可以互動的部分），我們就用最基礎、最根本的三種語言來打造：
  + **HTML**：蓋房子的**鋼筋水泥**，決定網頁的「結構」（這裡有一個標題、那裡有一個按鈕）。
  + **CSS**：房子的**油漆和裝潢**，決定網頁的「外觀」（標題是什麼顏色、按鈕長什麼樣子）。
  + **JavaScript (JS)**：房子的**水電和機關**，決定網頁的「互動」（按下按鈕後會發生什麼事）。
* **為什麼老師這麼建議？**：因為妳的專案核心是**氛圍和簡單互動**，用這三種「原生」語言就綽綽有餘，而且能讓妳真正學到網頁的根本。這就像老師建議妳先用最基本的紅、黃、藍三原色來學畫畫，而不是一開始就給妳一套複雜的繪圖軟體。**這對妳是好事，能大大降低學習難度！**

**2. 「後端架構用 JS Express」**

* **這句話的意思是**：網站的「廚房和倉庫」（使用者看不到的、處理數據和邏輯的地方），我們也用 JavaScript 來打造！
  + **Node.js**：這是一個「**翻譯官**」。原本 JavaScript 只能在瀏覽器（前端）裡工作，有了 Node.js 這個工具，JavaScript 就能在伺服器（後端）上運作了。
  + **Express.js**：這是 Node.js 的一個「**萬用工具箱**」(框架)。如果說 Node.js 讓妳能在後端說 JavaScript，那 Express 就是給了妳一套現成的句型和模板，讓妳可以非常快速、有條理地搭建出後端的功能（例如接收信件、讀取信件的API）。
* **為什麼老師這麼建議？**：這是一個**絕頂聰明**的建議！原本妳的計畫是前端用JS，後端用Python，這代表妳要同時應付兩種不同的語言和環境。老師的建議讓妳**只需要專心學好一種語言——JavaScript！** 妳用JS寫前端的互動，也用JS寫後端的邏輯。這會讓妳的學習曲線平滑非常多！

**總結老師的建議就是**：讓我們用同一種語言 (JavaScript)，來搞定所有事吧！這會讓整個專案單純很多。

**好的，那我們現在該做什麼？**

萬事起頭難，我們從準備好開發環境開始。這就像畫畫前，先把畫布和顏料準備好一樣。

請妳跟著下面的步驟，我們來完成今天的第一個任務：**安裝妳的開發工具**。

**第一步：安裝程式碼編輯器 (妳應該有了)**

* **工具**：**Visual Studio Code (VS Code)**
* **說明**：這是妳寫程式碼的地方，就像妳的數位筆記本。
* **連結**：如果需要重新安裝，可以到這裡下載：<https://code.visualstudio.com/>

**第二步：安裝 Node.js (最重要的第一步！)**

* **工具**：**Node.js**
* **說明**：這就是我們剛剛說的「翻譯官」，讓妳的電腦能聽懂用在後端的 JavaScript。安裝它，妳就同時擁有了在後端執行JS的能力。
* **如何安裝**：
  1. 前往 Node.js 官方網站：<https://nodejs.org/>
  2. 網站首頁會有兩個大大的綠色按鈕。請點擊左邊那個標有「**LTS**」字樣的按鈕來下載。LTS 代表「長期支援版」，是**最穩定、最推薦**初學者使用的版本。
  3. 下載下來後，像安裝普通軟體一樣，不斷點擊「下一步」完成安裝即可。
  4. **驗證是否成功**：安裝完畢後，打開妳的**終端機 (Terminal)** (在Windows上可以叫命令提示字元或PowerShell，在Mac上就是終端機)，輸入 node -v 然後按下 Enter。如果出現了版本號碼 (例如 v20.11.1)，就代表妳成功了！