

Web App 開發 - 期末報告

StoryVerse AI Hub - 一個利用生成式繪本來學習語言的網站

M組

R12521605 胡羽忻

B10303032 賀昭瑄

B12303035 蔡宏杰

B12303143 廖翊宏

B11208009 許睿中

B10801019 鄭可維

摘要

我們的目標是利用生成式AI技術，建立一個多語言且支持語音的繪本創作平台，將語言學習、創作、社群媒體等概念相結合，提供豐富的用戶體驗。網站專為學習語言的小朋友、家長及全世界所有語言愛好者設計。

我們致力於提供安全且免費的平台，讓用戶輕鬆管理和探索繪本創作，同時建立一個充滿支持與互助的學習社群。因此，在網站設計上包含兩種主要使用情境：第一，創作與學習，通過AI生成出整合了文字、語音和圖像的教育性繪本，支持多語言學習和語音播放功能。第二，管理與探索，用戶可以管理自己的創作，並探索其他用戶的作品，促進創意展示和交流。

在方法與技術上，前端使用了 theme wagon 上的開放模板 Savory 來進行修改，因其響應式功能和分頁設計與我們的專案相似。後端我們使用 Flask 框架結合 Firebase JavaScript SDK，實現前後端分離、分頁管理和 AI 腳本運行，並通過API實現前端與資料庫的高效交互。資料庫則利用非關聯式的 Google Firebase 以儲存多種形式資料，且其提供驗證、託管、資料庫、雲端代碼等API也方便開發者使用。在 AI 模型方面，我們利用多個開放模型，並設計了一整套流程，以生成連貫且準確的繪本內容，這允許使用者輸入標題和選擇語言，即可生成四頁繪本的插圖、內文和語音。

現階段，網站提供註冊、登入、個人、探索和創作頁面等多個使用者介面和完整功能，資料庫和後端的設計建構出有系統的資料存放和拿取，AI 模型也提供多種語言的繪本內容生成。我們的網站充分利用生成式AI的潛力，讓用戶創作多語言的有聲繪本，同時，平台的探索功能讓用戶可以瀏覽和欣賞其他人的作品，從而享受一個更全面、有趣的語言學習與創作體驗。未來計畫可針對前端的設計做改進，提供更有利於語言學習的功能；另外，AI 模型和網站的上架也是未來可以優化的部分。

1. 前言

1.1 動機

在當今快速發展的科技時代，將 AI 技術融入繪本創作，不僅可以簡化設計過程，還能賦予作品更多創意和個性化的元素。我們通過研究現有的 AI 繪本創作網站，發現這些平台在以下幾個方面有待改進，以增強使用者體驗：

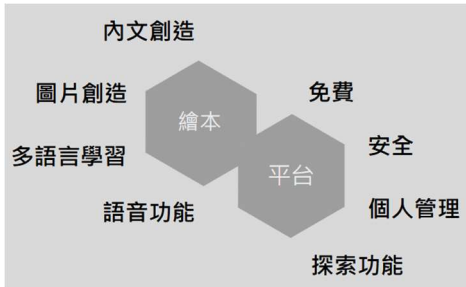
1. 插圖的一致性：用戶評價中顯示現有的服務在插圖風格和質感上往往不一致。

- II. 文本創作的自動化：部分平台僅提供插圖生成，用戶需自行撰寫配套文字，這降低了 AI 創作的潛力。
- III. 收費：我們傾向於提供免費平台，通過廣告收入來維持運營。
- IV. 語言的多樣性：多數平台僅提供英文版服務，忽略了非英語使用者的需求。
- V. 語音支持的缺乏：多數平台不支持語音朗讀功能。

我們的目標是消除這些障礙，創建一個更加直觀、包容且功能全面的 AI 繪本平台。同時，我們希望賦予這個平台更多意義，不單單僅是瀏覽和創作繪本，我們希望能將學習語言與繪本相結合。因此，網站將支持多語言學習，通過自動生成文本和語音互動功能，使學習語言變得生動有趣，豐富兒童和成年人的創作和學習體驗。

1.2 目標價值和客群

我們的網站專為學習語言的小朋友、家長以及任何對學習新語言感興趣的人而設計。為了提供利用繪本提供有趣的語言學習，我們從「繪本」和「平台」兩個方面進行發想：

<p>提供價值：</p> 	<ul style="list-style-type: none"> I. 「繪本」：我們將繪本與多種 AI 技術相結合，包括：內文創造功能，讓內容充分發揮 AI 的潛力，讓學習語言的過程變得有趣；自動圖片生成，快速創建有視覺吸引力的繪本；支持多語言學習，超過 15 種語言方便用戶進行學習；語音播放功能，幫助學習者聽到正確的發音，提升語言理解與學習效果。 II. 「平台」：我們提供一個免費、安全的平台，且用戶不僅可以輕鬆管理自己的繪本，也能自由探索各種語言的創作。
---	--

通過這些特色，我們的目標是讓語言學習變得更加生動且富有吸引力，通過文字、圖像、語音的連貫互動，滿足各年齡層學習者的需求。此外，平台的設計也旨在鼓勵用戶積極參與和分享，從而建立一個充滿支持與互助的學習社群。

1.3 預期使用情境

我們為網站的設計設想兩種主要的使用情境，方便之後的網站設計能圍繞著這兩個主要方向：

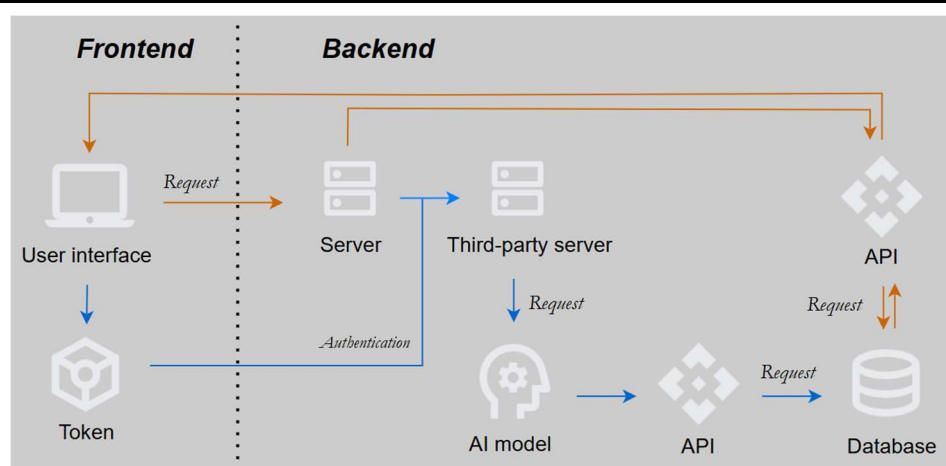
- I. 情境1：創作與學習

本平台讓使用者能夠用生成式 AI 生成繪本，並透過文字、語音和圖像的整合來學習新語言。這種方式旨在提供一個教育性與互動性兼具的學習工具，不僅讓語言學習變得更加有趣，也大大提高了記憶與理解的效率。
- II. 情境2：管理與探索

我們的平台不僅允許用戶管理自己的創作，同時也可以探索和互動其他用戶的作品。這一功能類似於社交媒體平台，如 Instagram 或小紅書，其中用戶可以發布和管理繪本。這樣的設計不僅使用者能夠展示自己的創意作品，還能與全球的語言學習者進行互動和交流，幫助提升學習的動機，也有助於建立一個鼓勵創意和知識分享的社群。

2. 方法

2.1 架構

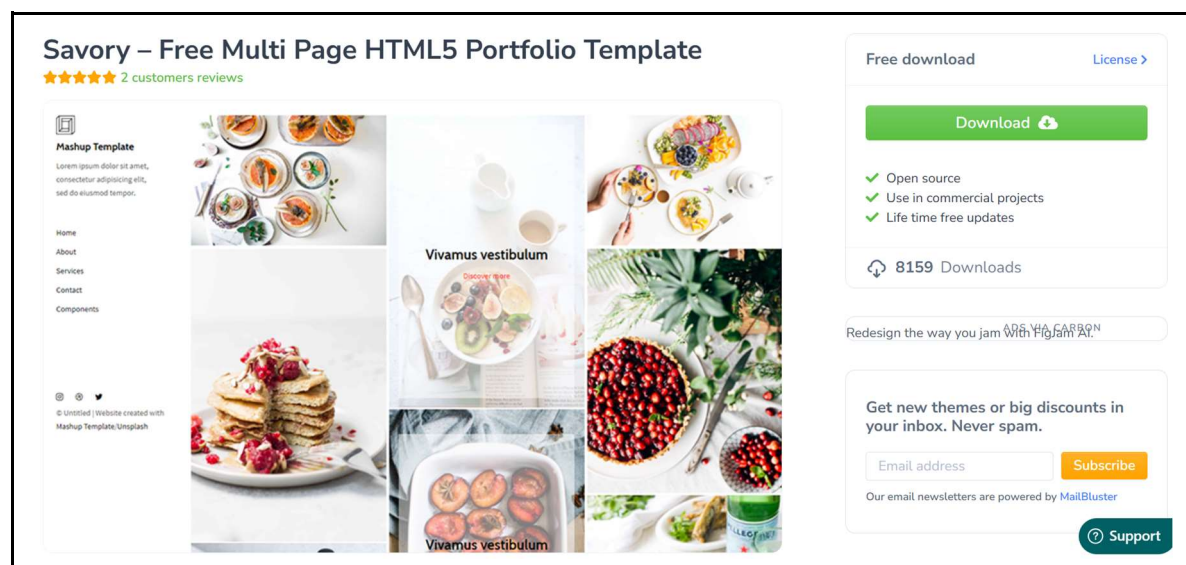


- I. 橘色箭頭表示從資料庫獲取數據並渲染到前端的過程。在不涉及任何利用AI的操作流程中，伺服器會向伺服器發出請求，這些請求會通過 API 與資料庫進行交互。數據從資料庫獲取後，經過處理並返回給前端，最終在使用者介面上渲染顯示。
- II. 橘色+藍色箭頭表示AI 生成、儲存及渲染的完整過程。若操作流程涉及 AI 功能，由於有串接公開模型的 serverless API，前端將發出請求，由後端伺服器接收後傳到第三方伺服器 (hugging face 平台) 進行身份驗證，再將請求轉發給 AI 模型進行處理。AI 模型處理完成後，結果會通過 API 傳送至資料庫進行儲存。最終，經過處理的數據會被前端調用並渲染顯示給用戶。

2.2 前端

2.2.1 前言

雖然課程中學到了許多技巧，但身為初學者的我們評估後，可能無法從零開始在一個月內達成我們想要的效果。因此，我們在網上搜尋到一個公開模板 (Savory) 符合我們的設計理念: <https://themewagon.com/themes/free-multi-page-html5-portfolio-template-free-download/>



這個模板本身是響應式設計 (RWD)，也使用了 Bootstrap 模板；首頁有圖卡瀏覽功能，符合我們希望在首頁展示所有故事的需求；每個分頁都有側邊欄提供跳轉其他分頁的功能，且

側邊欄本身也有響應式設計。我們新增了一些 JavaScript 腳本來編輯新的功能、並搭配原有的 HTML 及 CSS 來實現我們的設計需求。

2.2.2 概述各頁面功能

- I. 探索頁面 (首頁): 用戶可以在探索頁面瀏覽由其他用戶創作的故事。
- II. 創作頁面: 提供創建故事的工具和界面。
- III. 個人頁面: 用戶可以查看和編輯個人資料，並查看個人創作的故事。
- IV. 登入和註冊頁面: 用戶可以註冊新帳號或登入現有帳號。
- V. 故事頁面: 此頁面展示繪本的圖片、內文、和語音，因為一個繪本有四頁，因此我們也做了翻頁和頁碼的功能。

2.2.3 遇到的困難和解決方法

- I. 資料庫同步問題: 我們在登入時最初是採用將id存在 localStorage 的方式進行登入，因為當時不確定資料庫提供的接口。在資料庫提供接口後，我們對系統進行了一些修改，以解決這些問題。
- II. 創建故事的 ID 問題: 一開始沒有想到 ID 需要自己創建，這也導致了一些狀況發生。最後，我們修改了創建故事的流程，確保每個故事都有唯一的 ID。
- III. 團隊協作問題: 大家在協作時發現我們無法同時處理同一個問題，這樣會使問題變得更複雜。我們的解決方法是一起進行編碼和當面討論，以便及時解決問題。
- IV. 註冊和登入提示: 在註冊成功後，我們會調用資料庫的接口，並通過 JavaScript 提供提示信息，讓用戶收到驗證碼。註冊失敗、登入成功、登入失敗也都是通過資料庫的接口來實現的。然而，登出功能仍然使用 localStorage 寫法，因為發現如果全部都調用資料庫的接口，可能會有一些問題。

這些困難和解決方法讓我們更好地理解系統的設計和實現，並最終完成了一個功能完整的網站。

2.3 後端和伺服器

2.3.1 後端技術介紹

在本專案中，後端採用了 Flask 框架結合 Firebase JavaScript SDK 來實作，會以這種混合的方式是考量到 AI 使用 python 和其他部分的環境衝突，Flask 主要負責頁面的路由管理和 AI 函數的調用，Firebase JavaScript SDK 負責與 Firebase 資料庫的溝通，詳見2.4章節。

Flask	Flask 是輕量級的 Python Web 框架，用於處理 HTTP 請求並傳回對應的 HTML 範本。在本專案負責頁面的路由管理和 AI 函數的調用，寫在 app.py 中： <ol style="list-style-type: none">I. 提供頁面和路由管理：透過 Flask 的路由機制，我們能夠輕鬆地為每個頁面和功能定義對應的 URL，從而確保前端頁面的可維護性和邏輯結構的清晰。II. 呼叫 AI 函數產生繪本內容：在 /creator 端點中，Flask 伺服器接收來自前端的請求數據，並呼叫多個 AI 函數來產生繪本內容。生成的內容被儲存為 JSON 文件和多媒體文件，上傳到 Firebase 存儲，並產生可訪問的 URL，以便前端頁面能夠直接獲取和展示這些內容。
Firebase JavaScript SDK	Firebase JavaScript SDK 直接整合到前端程式碼中，負責與 Firebase 資料庫進行通訊。在本專案負責與資料庫的溝通，寫在每個前端分頁的 script 區塊： <ol style="list-style-type: none">I. 透過在前端頁面中使用 Firebase SDK，我們能夠直接從客戶端進行資料的讀取和寫入操作，從而簡化了後端的開發工作。

- II. Firebase 的即時資料庫功能也使得我們能夠輕鬆實現資料的即時更新和同步。

2.3.2 專案架構

專案的文件架構必須遵循 Flask 的慣例，結構如下：

- I. templates/：存放 html 文件，用於渲染前端頁面。
- II. static/：存放靜態文件，如 css、JavaScript 和圖片資源。
- III. model/：存放 AI 模型相關文件和環境安裝需求。
- IV. app.py：Flask 應用的主入口，定義路由和基本的伺服器配置。
- V. firebase_config.json：存放 Firebase 的設定檔。

透過這種架構，我們能夠清楚地分離前端與後端的職責，同時利用 Firebase 的強大功能，實現高效的資料管理和使用者體驗。

2.3.3 本地開發環境

整個系統的運作都在本地環境中進行，資訊如下：

- I. 顯示卡型號：NVIDIA GeForce RTX 4090
- II. 顯示卡驅動版本：535.171.04
- III. CUDA 版本：12.2
- IV. 顯示卡總顯存：24564 MiB

2.4 資料庫和API設計

2.4.1 資料庫 Google Firebase 介紹

在考慮要用何種資料庫中，目標是模仿Twitter(X.com)、小紅書、或是FB等社群媒體，做出基於生成繪本的社群平台，因其特性所以考慮使用非關聯式資料庫資料庫，Google Firebase, google firebase 有免費的驗證、託管、資料庫、雲端代碼等API可供開發者使用，對於中小企業與個人開發友善，免去要自己寫所有後端的問題。



免費方案	配額
儲存的資料量	1 GiB
文件讀取次數	每天 50,000 次
文件寫入次數	每天 20,000 次
文件刪除次數	每天 20,000 次
外送資料傳輸	每月 10 GiB

(google firebase 的使用者介面，目前開通了 Authentication, cloud firebase 與 fuunctions 功能)

(google firebase 有分兩種方案，Spark 與 Blaze, 小規模的用Spark 即可)

2.4.2 如何連線到 Google firebase

- I. 獲取金鑰 (專案設定 -> 一般設定 -> 我的專案)

- II. 在網頁中引入 firebase 的驗證，Python可用金鑰(json檔)，javascript只需引入下列到script中

```
<script src="https://www.gstatic.com/firebasejs/7.7.0/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.7.0/firebase-storage.js"></script>
<script src = "controller/upload.js">
```

(將後端資訊引入至html)

```
const firebaseConfig = {
  authDomain: "...",
  storageBucket: "...",
  projectId: "...",
};
firebase.initializeApp(firebaseConfig);
```

2.4.3 資料庫 API 的使用 - Storage 音檔與圖片的上傳

```
function upload(file = document.querySelector("#photo").files[0]){
  uploadProgress(file)
    .then(downloadURL => {
      // here can return URL to use
      // example: return downloadURL
      console.log("Download URL:", downloadURL);
    })
    .catch(error => {
      console.error("Error:", error);
    });
}

function loadTask(path, file) {
  const ref = firebase.storage().ref(path);
  const name = +new Date() + "-" + file.name;
  const metadata = {
    contentType: file.type
  };
  return ref.child(name).put(file, metadata);
}
```

2.4.4 資料庫 API 的使用 - 使用方式和架構設計

- I. 使用Google-Auth 的API關於用戶的註冊、登入、驗證、登出、得到登入用戶的資訊、重新驗證（對於更改密碼或使用戶資訊、刪除帳號），重設密碼、刪除帳號、寄出驗證信箱，要使用同樣引入google-auth資源庫。
- II. 思考如何設計資料庫形式- 用戶角度出發：
- 從用戶會做出的行為出發來設計資料庫，首先考慮用戶會做出甚麼行為：
- 用戶會生成故事、發布故事（分成有被發布與沒被發布的故事）
 - 用戶會更改故事（如果故事已經被發布，需要在公共區域更新）
 - 用戶會透過故事跳轉到創作者的網頁
 - 公共頁面的渲染
 - 個人頁面的渲染（用戶除了自己生成過的故事，還有自我介紹等）
- 由此，設計的Database有 public / private / users 三個資料夾，其中private 儲存用戶過去創作過的故事，users 存儲性別/手機號碼/信箱等等資訊，public存儲已經被發布過的故事。
- III. 思考如何設計故事形式：

參照了一般的四格漫畫，考慮到可擴展性與程式簡潔性，以陣列而非字典存儲，每個格子配有照片、音檔與文字和翻譯，其中照片與音檔以訪問連結存儲，一個故事中包含多個小格子、封面圖片、語言、創作者ID與標題。



(故事的架構, 由上到下的欄位為 故事的token、封面、語言、創作者用戶編號、故事、標題)

public	private	users
/ 故事token ...	/ 用戶ID / 故事token	/ 手機 / 信箱 / 性別 / 綁定信箱

IV. 困難：透過 fetch 回來的資料存入資料庫

見updateStories，照理說，應該在生成式AI 的地方將資料丟進資料庫，但是由於環境衝突及時間成本（AI生成僅支援Python，不過Python database API的使用需要更新版本種類，及要求使用權新資料庫），遂在前端接受到資料的第一步即將資料存入資料庫。

```

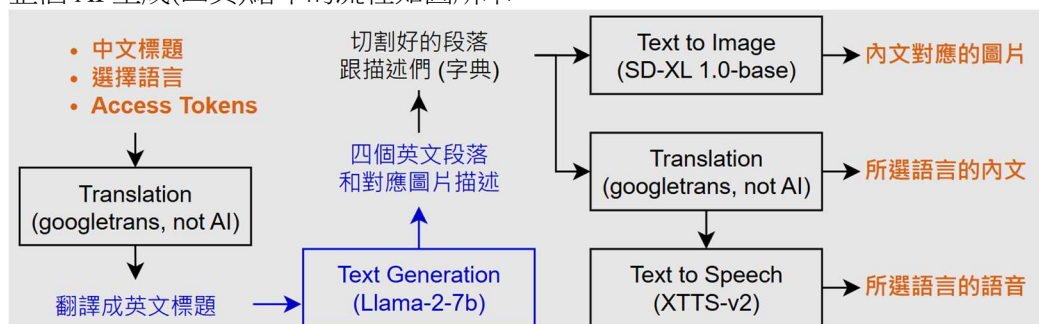
console.log('storyToken: ' + storyToken);
fetch('/creator', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({ title, language, token, uid, storyToken })
})
.then(response => response.json())
.then(data => {
  try {
    container = updateStories(data);
    // publishStories(data["storyToken"], container);
  } catch (error) {
    console.error(error);
  }
  myStoryImages(uid);
})

```

2.5 AI模型

2.5.1 流程和架構

整個 AI 生成(四頁)繪本的流程如圖所示：



- I. 輸入：用戶輸入繪本標題 (任何語言都行 不一定要中文)，選擇想要學習的語言，並提供他的 Model Access Tokens 以授權使用開放模型。
- II. 生成繪本內文和插圖描述：將用戶提供的標題翻譯成英文標題，放進 Text-Generation 模型，模型根據翻譯後的英文標題，生成四段英文短文和每段對應的圖片描述，翻譯成英文是因為測試後發現 llama 在英文的生成表現是最好。接著，再針對生成的短文進行整理，切割成字典以便於後續處理：{'paragraph 1': "...", 'illustration 1': "...", 'paragraph 2': "...", 'illustration 2': "...", 'paragraph 3': "...", 'illustration 3': "...", 'paragraph 4': "...", 'illustration 4': "...", 'title': "...'}。
- III. 生成圖片、翻譯後的內文和語音：將圖片描述放進 Text-to-Image 模型，生成四張插圖；將四段英文短文翻譯成用戶所選語言產生內文；再將翻譯後內文放進 Text-to-Speech 模型轉換為所選語言的語音。

2.5.2 各模型細節

Text Generation	<ol style="list-style-type: none"> I. 模型：Llama-2-7b, https://huggingface.co/meta-llama/Llama-2-7b II. Prompt：為了生成好的繪本內文和插圖描述，我們告訴模型要以安徒生的故事為例去發想；因為tokens限制，也希望每段不要超過30個字；格式也有特別要求，我們也會判斷生成結果，若不符合格式要求就會重新跑迴圈生成，直到符合為止，這樣可以避免模型生成未滿四個段落、或是沒有插圖描述。 <table border="1"> <tr> <td>Prompt</td><td> Taking Andersen's story as an example, write a 'very short' picture book story in English. It should include four paragraphs (no more than 30 words) and four illustration descriptions. Title: {title}. Format please follows: Title: Paragraph 1: Illustration 1: Paragraph 2: Illustration 2: Paragraph 3: Illustration 3: Paragraph 4: Illustration 4: </td></tr> </table>	Prompt	Taking Andersen's story as an example, write a 'very short' picture book story in English. It should include four paragraphs (no more than 30 words) and four illustration descriptions. Title: {title}. Format please follows: Title: Paragraph 1: Illustration 1: Paragraph 2: Illustration 2: Paragraph 3: Illustration 3: Paragraph 4: Illustration 4:
Prompt	Taking Andersen's story as an example, write a 'very short' picture book story in English. It should include four paragraphs (no more than 30 words) and four illustration descriptions. Title: {title}. Format please follows: Title: Paragraph 1: Illustration 1: Paragraph 2: Illustration 2: Paragraph 3: Illustration 3: Paragraph 4: Illustration 4:		
Text to Image	<ol style="list-style-type: none"> I. 模型：stable-diffusion-xl-base-1.0, https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0 II. Prompt：除了圖片的描述外，我們也指定插圖風格符合Joan Cornellà這位插畫家，之所以挑選他是因為我們試過其他人，最後發現他的風格最為強烈，所以生成的插圖會比較一致。但由於是跑四次迴圈來生成四張插圖，所以仍會有插圖不一致的情況出現，我們猜測或許須把前一次的圖也當input，才能真正保證圖片主角和風格完全一致，但就不 		

	<p>能單用 text-to-image 的模型。</p> <table> <tr> <td>Prompt</td><td> Style: Joan Cornellà. Content: {content} </td></tr> </table>	Prompt	Style: Joan Cornellà. Content: {content}
Prompt	Style: Joan Cornellà. Content: {content}		
Text to Speech	<ol style="list-style-type: none"> I. 模型：XTTS-v2，https://huggingface.co/coqui/XTTS-v2 II. 語言選項：英語、西班牙語、法語、德語、意大利語、葡萄牙語、波蘭語、土耳其語、俄語、荷蘭語、捷克語、阿拉伯語、中文、匈牙利語、韓語、日語和印地語。因為翻譯我們用 googletrans，語言選項有超過一百種，涵蓋非常多語言，所以我們網站上的語言選項就以 text-to-speech 能提供的這17種作為選項。 III. TTS 模型沒有提供 serverless API 可以串，所以是下載在本地來使用。 		

3. 結果

3.1 專案資料夾結構

<ol style="list-style-type: none"> I. app.py：Flask 應用的主入口，定義路由和基本的伺服器配置。 II. firebase_config.json：存放 Firebase 的設定檔。 III. templates/：主要放六個 html 分頁。 IV. static/：放 css 和 js 等。 <ol style="list-style-type: none"> A. assets/：放圖片資源。 B. css/：放模板的css和我們自訂的css。 C. hello-world/：放Firebase JavaScript SDK的API函數 (雖然這部分後來都直接放在各個 html 的 script 裡面)。 D. js/：放模板的js和我們自訂的js。 V. model/：放 AI 模型檔、環境安裝需求、文字轉語音的語音範本、文字轉語音的下載位置 (每次下載上傳資料庫後會刪除)。 	<pre> website ├── API │ ├── functions.pdf │ ├── app.py │ ├── firebaseconfig.json │ └── main.85741bff.map ├── model │ ├── init_.py │ ├── model.py │ ├── requirements.txt │ ├── speaker.mp3 │ ├── speech │ └── notes.md ├── static │ ├── assets │ │ ├── images │ │ │ ├── logo.jpg │ │ │ ├── profile.png │ │ └── mashup-logo.svg │ ├── css │ │ ├── main.82cfd66e.css │ │ └── style.css │ ├── hello-world │ │ ├── database.rules.json │ │ ├── firebase.json │ │ ├── note.txt │ │ └── public │ │ ├── 404.html │ │ ├── clickme.html │ │ ├── controller │ │ ├── auth.js │ │ ├── config.js │ │ ├── insert.js │ │ ├── upload.js │ │ └── userStatus.js │ ├── upload.js │ ├── views │ │ ├── insert.html │ │ └── public.html │ └── storage.rules ├── js │ ├── login.js │ └── main.85741bff.js └── templates ├── contact.html ├── creator.html ├── index.html ├── profile.html ├── project.html ├── signup.html └── static ├── hello-world │ └── public │ ├── controller │ └── userStatus.js </pre>
---	--

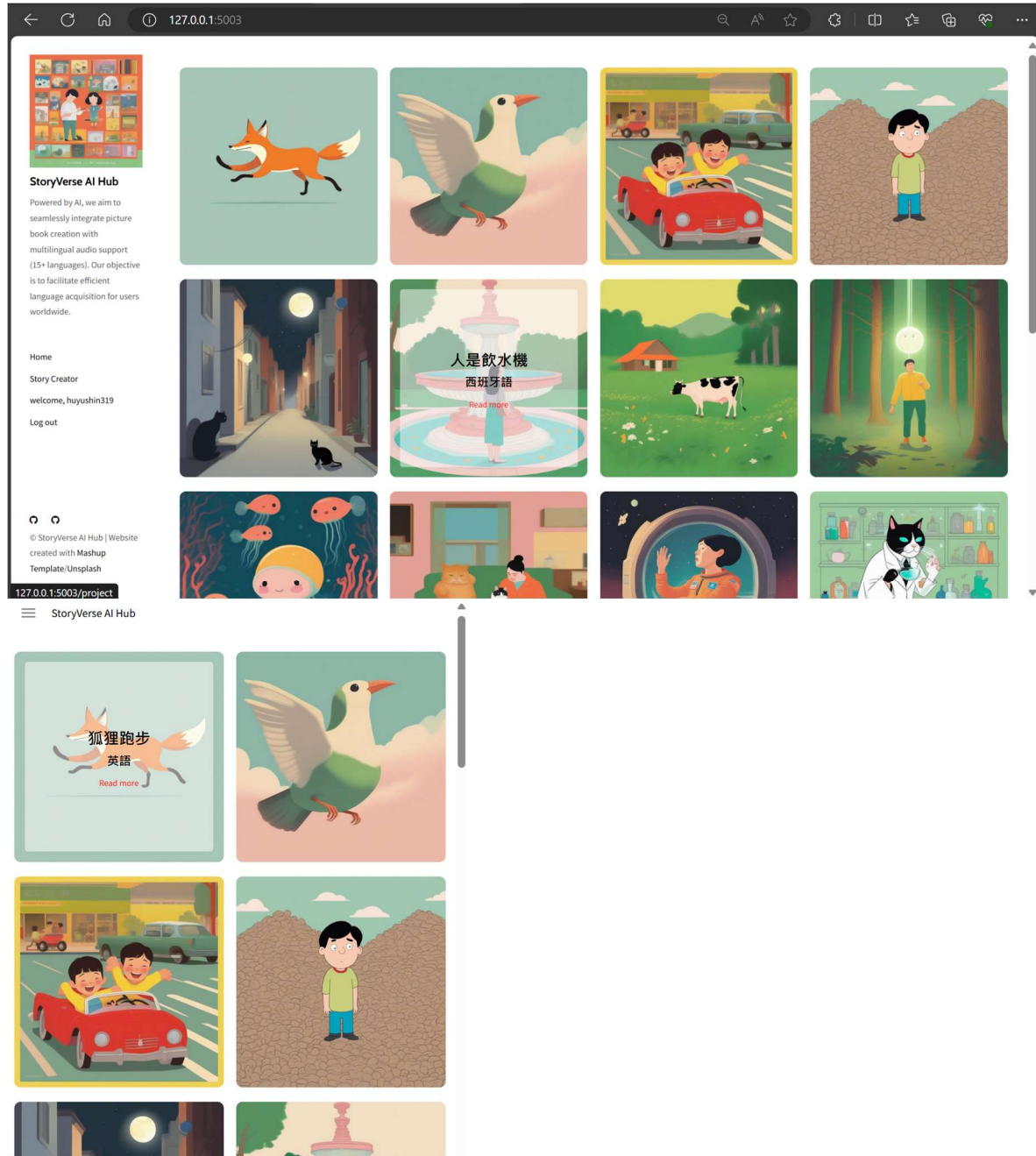
3.2 成果展示

Demo : <https://www.youtube.com/watch?v=iFsHQ6Nd-IY>

3.2.1 探索頁面 (index.html)

用戶可以在探索頁面瀏覽由其他用戶創作的故事。

在此頁面上展示了多個故事的封面，每個封面上都標明了故事名稱和所使用的語言，方便用戶快速識別和選擇他們感興趣的內容，點擊進去即可看到故事內容。在頁面的左側邊欄，可以看到我們網站的 logo 和一段簡短的網站介紹，還提供了多個分頁連結，用戶可以輕鬆導航到網站的其他部分，包括 Home (就是探索頁面)、Story Creator、login 或 profile + logout 等功能頁面。網站也提供響應式設計，當頁面縮放，側邊欄會自動縮進左上角的按鈕中。

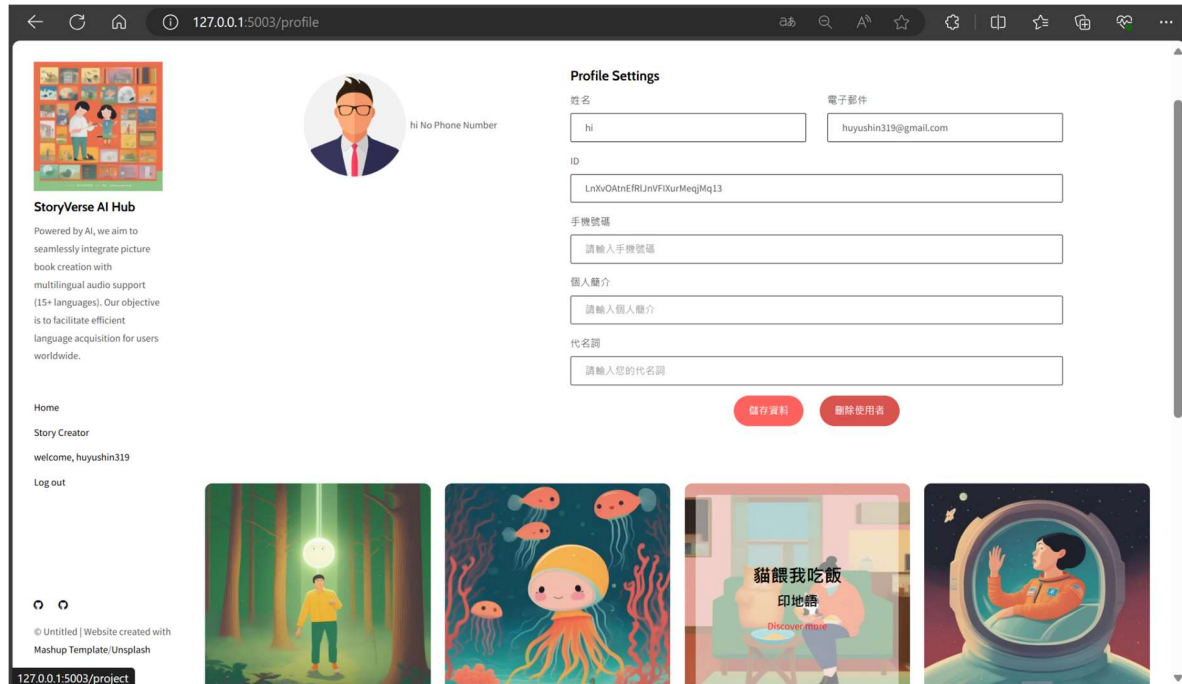


3.2.2 個人頁面 (profile.html)

在個人頁面，用戶可以進行個人檔案的編輯和管理，並查看過去的創作記錄。

在頁面的右側，用戶可以編輯個人信息，包括修改姓名、電子郵件、手機號碼、個人簡介和

代名詞等信息。修改完成后，用户可以点击「储存资料」按钮来保存更改。如果用户需要删除帐户，可以点击「删除使用者」按钮。页面的下方展示了用户过去创作的故事，这些故事的封面同样标明了故事名称和语言，点击进去即可看到故事内容。



127.0.0.1:5003/profile

StoryVerse AI Hub

Powered by AI, we aim to seamlessly integrate picture book creation with multilingual audio support (15+ languages). Our objective is to facilitate efficient language acquisition for users worldwide.

Home

Story Creator

welcome, huyushin319

Log out

© Untitled | Website created with Mashup Template/Unsplash

127.0.0.1:5003/project

Profile Settings

姓名

hi

電子郵件

huyushin319@gmail.com

ID

LnXvOAtnERlJnVFXurMeqMq13

手機號碼

請輸入手機號碼

個人簡介

請輸入個人簡介

代名詞

請輸入您的代名詞

儲存資料

刪除使用者

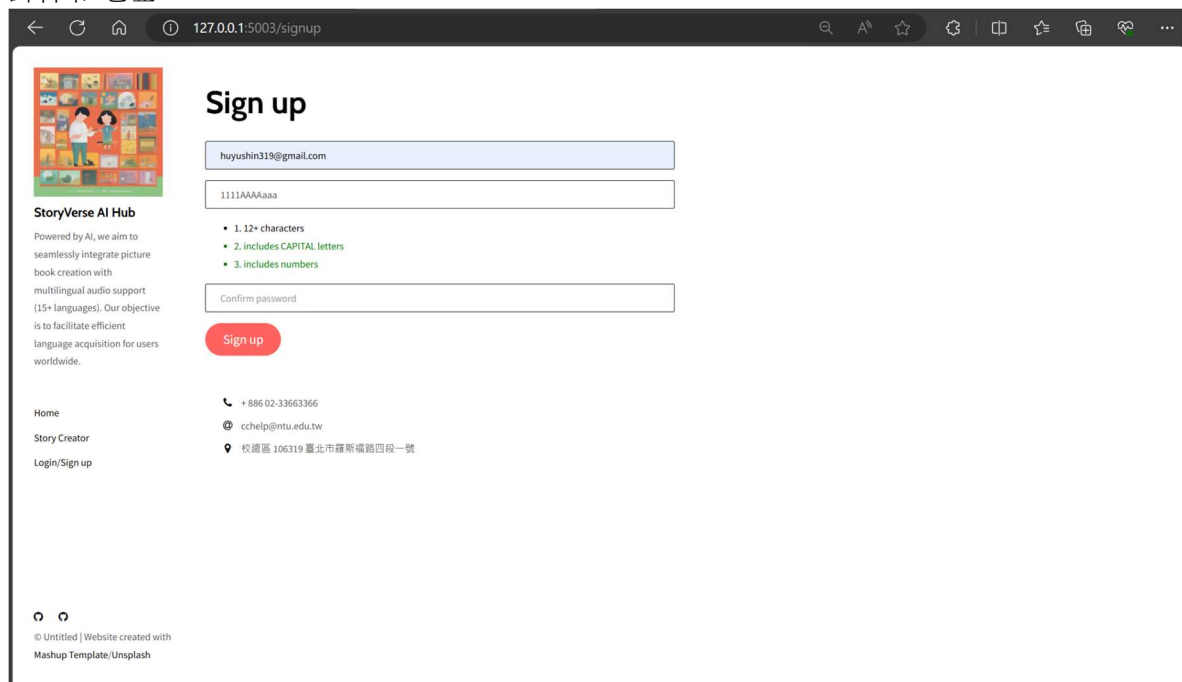
貓餵我吃飯

印地語

Discover more

3.2.3 註冊頁面 (signup.html)

在註冊頁面，用戶可以通過輸入電子郵件和符合規定的密碼來創建新帳戶，以創作繪本。註冊時，密碼必須至少包含12個字符，包括大寫字母和數字，且用戶需要再次輸入密碼以確認其一致性。填寫完信息並確認密碼後，點擊按鈕完成註冊。接著用戶在 email 會收到驗證信，點擊完成驗證即可使用帳戶。在頁面的底部，提供了台大的聯繫信息，包括電話、電子郵件和地址。



127.0.0.1:5003/signup

StoryVerse AI Hub

Powered by AI, we aim to seamlessly integrate picture book creation with multilingual audio support (15+ languages). Our objective is to facilitate efficient language acquisition for users worldwide.

Home

Story Creator

Login/Sign up

© Untitled | Website created with Mashup Template/Unsplash

Sign up

huyushin319@gmail.com

1111AAAAaa

- 1. 12+ characters
- 2. includes CAPITAL letters
- 3. includes numbers

Confirm password

Sign up

+ 886 02-33663366

cchelp@ntu.edu.tw

校總區 106319 臺北市羅斯福路四段一號

3.2.4 登入頁面 (contact.html)

在登入頁面，用戶需要輸入電子郵件和密碼來訪問他們的帳戶。
頁面設計簡潔，讓用戶能夠快速輸入信息並點擊登入按鈕完成登入。

The screenshot shows a web browser at the address 127.0.0.1:5003/contact. The page title is "Login / Sign up". On the left, there is a logo for "StoryVerse AI Hub" and a sidebar menu with links: Home, Story Creator, and Login/Sign up. The main content area has a text input field for the email (huyushin319@gmail.com) and a password input field (masked with dots). Below these are "Login" and "Sign Up" buttons. At the bottom, contact information is listed: a phone number (+886 02-33663366), an email (cchelp@ntu.edu.tw), and an address (校總區 106319 臺北市羅斯福路四段一號). The footer mentions "© Untitled | Website created with Mashup Template/Unsplash".

3.2.5 建立頁面 (creator.html)

在建立故事頁面，用戶可以創建新的繪本。

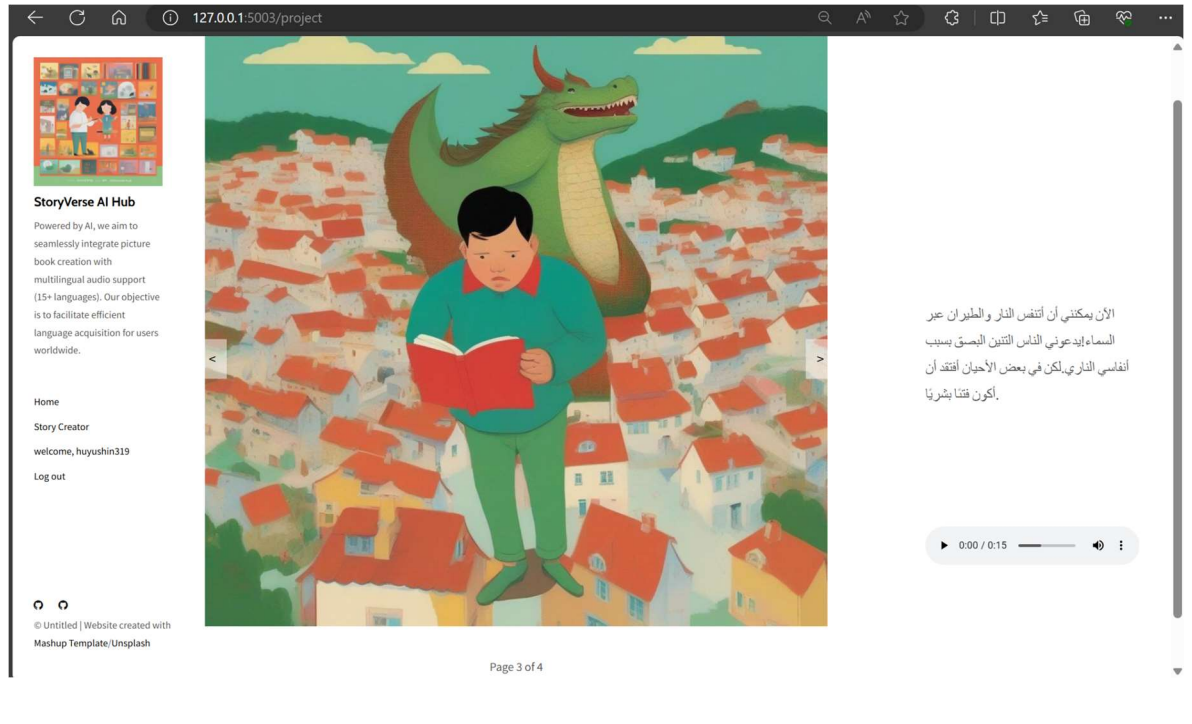
用戶需輸入標題，並從選單中選擇想要學習的語言，還需從指定連結獲取訪問令牌，用於API串接模型，最後點擊提交按鈕提交創建請求。過幾分鐘後，頁面下會出現故事封面 (截圖尚未創建故事)，點擊進去即可看到故事內容。同時該故事會被新增到個人頁面和探索頁面中。

The screenshot shows a web browser at the address 127.0.0.1:5003/creator. The page title is "Create a picture book". On the left, there is a logo for "StoryVerse AI Hub" and a sidebar menu with links: Home, Story Creator, and a welcome message for huyushin319 with a "Log out" link. The main content area has a "Title:" text input field, a "Language:" dropdown menu (currently showing "英語"), a "User Access Token:" text input field with a "Get from here" link, and a "Submit" button. The footer mentions "© Untitled | Website created with Mashup Template/Unsplash".

3.2.6 故事頁面 (project.html)

在繪本故事頁面，用戶可以閱讀和聆聽繪本故事的內容。

每一頁展示繪本故事的插圖和內文，並顯示當前頁碼，頁面還提供語音播放功能，用戶可以聆聽故事內容的語音版本，內文和語音以創建時選擇的語言作為呈現。此外，用戶可以通過翻頁按鈕來瀏覽故事的不同頁面，體驗連貫的閱讀過程。



4. 結論

4.1 總結與未來計畫

4.1.1 專案總結

- I. 使用者介面功能：包含註冊 (驗證信)、登入登出系統、個人頁面、探索頁面、創作頁面，為使用者提供了全面的功能和體驗。
- II. AI 繪本內容生成：當使用者輸入繪本標題，即可實現繪本內文、圖像、語音的生成，並能將內容和語音翻譯成 15 種以上的語言，達到了將生成式 AI 與繪本、語言學習相結合的目標。
- III. 資料庫管理：實現了多種形式資料的存儲，並為每個使用者和故事生成專屬的id和token，按照使用者和其故事整齊存放，為資料管理提供了良好的組織。

4.1.2 未來計畫

- I. 語言分類與篩選：目前各種語言的故事是展示在同一頁面，未來應新增依照語言類型和程度分類與篩選的功能，更優化語言學習的使用者體驗和便利性。
- II. 多語言內容展示：目前當使用者選擇一種語言後，生成的故事將只能以該種語言來翻看，若能改成雙語、或直接將翻譯選項放置在故事旁邊，使同一故事具有多種語言的內容和語音，更能提供語言學習的目的。
- III. 下載功能：新增將內容輸出為有聲書檔案的下載功能，提供使用者離線學習。
- IV. AI 模型最佳化：優化 AI 模型的速度、連貫性和準確率，或產生多張影像供使用者選擇，提升使用者的創作體驗。

- V. 圖像生成影片的 AI：未來也許能新增圖像生成影片的 AI，將網站發展成為繪本動畫網站，為使用者提供更生動豐富的體驗。
- VI. 伺服器上架：現在是架設在本地伺服器，若考慮上架，可以購買雲端伺服器，並使用 firebase 有配合的 Google 提供的託管服務，可考慮將網站上架，安插廣告賺取收益。

4.2 協作方式

- I. 共享帳號：原本要在 colab 運行和協作，但 colab 的用量限制和後端架設限制會變太麻煩，後來我們選擇使用相同的電腦帳號來協作。這不是常見的協作方式，但主要考量是 AI 模型需要在 GPU 環境運行，大部分組員都沒有能使用的本地環境，所以連到其中一個組員的實驗室電腦帳號來操作。後來發現這樣反而能即時看到程式碼的更動，也省去合併分支的時間，算是提升蠻大的效率，不過未來協作還是會盡量避免這種方式。
- II. 即時溝通：透過 LINE 群組保持聯繫，及時交流專案進度、遇到的問題以及解決方案，確保專案順利進行。
- III. 每週進度回報和討論：每週這門課的下課時間，我們進行進度回報，串接的部分也會互相詢問並討論解決辦法。
- IV. Git 版本控制：我們使用 Git 管理專案程式碼，紀錄每個小階段的更動內容，加上實驗室伺服器期末使用量大，進行備份避免當機或遺失。

4.3 分工表

人員	工作
胡羽忻	AI 模型、Flask 後端、PM
賀昭瑄	資料庫、Firebase JavaScript SDK 後端和 API 函數提供
蔡宏杰	前端、前端的 API 串接
廖翊宏	前端、前端的 API 串接
許睿中	前端、前端的 API 串接
鄭可維	AI 模型的 API 串接

參考資料

- I. https://createbookai.com/?ref=deepgram&utm_source=deepgram&utm_medium=referral
- II. <https://www.childbook.ai/>
- III. <https://themewagon.com/themes/free-multi-page-html5-portfolio-template-free-download/>