當然沒問題！由於我是文字回應，無法直接提供 Excel 檔。但你可以依下列說明順利製作這兩個 Excel 表格：

## **1. 技術整合細分任務彙總表（可貼到 Excel）**

| **主軸** | **主要階段** | **子任務** | **內容說明** | **主要工具/技術** | **產出/里程碑** |
| --- | --- | --- | --- | --- | --- |
| A. Boost.space | API 研究 | 1.1 API 能力盤點 | 盤點 Boost.space 可串接的 API、 探索授權模式、API 資料結構 | API Docs、Postman | API 盤點報告 |
| A. Boost.space | 對接與驗證 | 1.2 連線與測試 | 以 TypeScript 撰寫示範程式連接 Boost.space，進行回應驗證 | TypeScript、Node.js | Demo Code 及測試記錄 |
| A. Boost.space | 概念驗證 | 1.3 撰寫最小用例 Demo | 用實際數據拉取、資料同步，驗證整合作業流程 | TypeScript、Boost.space API | POC 報告 |
| B. Scripting.App 腳本應用 | 執行環境技術選型 | 2.1 調查 GCP 可用腳本環境 | 分析 Cloud Functions/Run, 安全隔離/沙箱實踐 | GCP 功能比較、TypeScript | 技術選型報告 |
| B. Scripting.App 腳本應用 | 執行環境技術選型 | 2.2 多語言支援評估 | 評估 TypeScript/JS/Python 混用適宜性與限制 | npm/yarn、Python venv | 多語言支援報告 |
| B. Scripting.App 腳本應用 | 權限與隔離 | 2.3 IAM權限設計及沙箱操作 | 設計 Service Account/IAM policy 隔離每段腳本執行權限 | GCP IAM、Policy 文檔 | IAM 案例設計稿 |
| B. Scripting.App 腳本應用 | 套件依賴管理 | 2.4 提供自動 npm/yarn 依賴處理方案 | 腳本自動安裝、依賴版本管理Auto/Cache | Node.js、npm/yarn | 依賴管理設計說明 |
| B. Scripting.App 腳本應用 | API 介面設計 | 2.5 定義腳本 API 並編寫 type 定義 | 設計腳本可呼叫的內部函式，撰寫.d.ts 文件 | TypeScript、型別檔 | 初版 API 文檔 |
| B. Scripting.App 腳本應用 | 蘋果捷徑增強 | 2.6 腳本產生/支援捷徑串接 | 提供 TypeScript 腳本與蘋果捷徑的互動模板 | TypeScript、Shortcuts | 捷徑腳本範例 |
| C. GCP 整合 | 服務選型 | 3.1 各服務用途需求比對 | 梳理功能對應服務－如Cloud Functions、Run、BigQuery等 | GCP Comparison Docs | 服務選型清單 |
| C. GCP 整合 | 身份授權 | 3.2 Service Account 建置 | 建立服務專用帳戶及細粒度權限配置試驗 | GCP IAM、Service Account | 權限設計文檔 |
| C. GCP 整合 | 身份授權 | 3.3 驗證用戶/應用程序身份 | 整合 Google Identity/Firebase Auth 驗證途徑 | GCP Identity, Firebase | 登入流程圖與示例 |
| C. GCP 整合 | SDK 整合 | 3.4 導入 cloud client for TypeScript | 實作與包裝常用 GCP 服務 TS SDK | GCP Node.js SDK、TypeScript | 封裝模組 |
| C. GCP 整合 | 服務移轉/PoC | 3.5 打造 TypeScript 術腳本-雲端自動部署 | 預計至少完成腳本雲上部署、Boost.space數據流串接 | GCP Console、TS、API | 雲端測試案例 |
| C. GCP 整合 | 數據流策略 | 3.6 規劃數據儲存流向 | 依 Boost.space→Cloud Function/Run→BigQuery→前端流程設計 | GCP Data Flow Tools | 數據架構圖 |
| D. GitHub 整合 | 倉庫建立 | 4.1 倉庫分類/權限設置 | Monorepo/多Repo方案討論與設計 | Git/GitHub | 倉庫結構說明文件 |
| D. GitHub 整合 | 分支管理 | 4.2 定義 Gitflow / PR 審查流程 | 按主線/開發/功能/熱修繕設分支規則 | GitHub Branch/PR | 分支協作規範 |
| D. GitHub 整合 | CI/CD流程 | 4.3 GitHub Actions 配置 | 規劃 Lint/Build/Test/Deploy至 GCP Artifact/CD | GitHub Actions、GCP | CI/CD yaml 檔 |
| D. GitHub 整合 | 議題追蹤 | 4.4 Issue 群組/Project 進度追蹤 | 利用 GitHub Issues/Projects 建任務牆 | GitHub Issues/Projects | 專案管理指南 |

## **2. 技術棧重點/效益/後續（可貼到 Excel）**

| **項目** | **子項** | **說明** |
| --- | --- | --- |
| 主要語言 | TypeScript | 全端主力、高適應度 |
| 腳本語言 | JS/TS/Python | 擴充性佳、支援廣泛 |
| 雲平台 | Google Cloud | 高彈性&AI/ML前瞻 |
| 數據存儲 | BQ, Firestore, Cloud SQL | 依情境選用 |
| 版本控管 | GitHub | 協作成熟、CI/CD完善 |
| 效益 | - | 整合廣/自動化深/可靠彈性/AI友善 |
| 展望 | - | 腳本管理平台, Vertex AI 深化, DevOps 極致 |

### **如何操作？**

1. 選取上表内容（包含欄位標題）。
2. 於 Excel 的任一工作表空白處「貼上」。
3. 可使用 [資料] > [分欄] 工具，讓資料自動分到各個欄位內。
4. 根據需要調整欄寬/加上篩選等格式。

### **補充**

如需 *CSV* 來源，請參考以下純文字格式（可用記事本存成 .csv）：

#### **表一 CSV 內容**

主軸,主要階段,子任務,內容說明,主要工具/技術,產出/里程碑

A. Boost.space,API 研究,1.1 API 能力盤點,盤點 Boost.space 可串接的 API、 探索授權模式、API 資料結構,API Docs、Postman,API 盤點報告

A. Boost.space,對接與驗證,1.2 連線與測試,以 TypeScript 撰寫示範程式連接 Boost.space，進行回應驗證,TypeScript、Node.js,Demo Code 及測試記錄

A. Boost.space,概念驗證,1.3 撰寫最小用例 Demo,用實際數據拉取、資料同步，驗證整合作業流程,TypeScript、Boost.space API,POC 報告

B. Scripting.App 腳本應用,執行環境技術選型,2.1 調查 GCP 可用腳本環境,分析 Cloud Functions/Run, 安全隔離/沙箱實踐,GCP 功能比較、TypeScript,技術選型報告

B. Scripting.App 腳本應用,執行環境技術選型,2.2 多語言支援評估,評估 TypeScript/JS/Python 混用適宜性與限制,npm/yarn、Python venv,多語言支援報告

B. Scripting.App 腳本應用,權限與隔離,2.3 IAM權限設計及沙箱操作,設計 Service Account/IAM policy 隔離每段腳本執行權限,GCP IAM、Policy 文檔,IAM 案例設計稿

B. Scripting.App 腳本應用,套件依賴管理,2.4 提供自動 npm/yarn 依賴處理方案,腳本自動安裝、依賴版本管理Auto/Cache,Node.js、npm/yarn,依賴管理設計說明

B. Scripting.App 腳本應用,API 介面設計,2.5 定義腳本 API 並編寫 type 定義,設計腳本可呼叫的內部函式，撰寫.d.ts 文件,TypeScript、型別檔,初版 API 文檔

B. Scripting.App 腳本應用,蘋果捷徑增強,2.6 腳本產生/支援捷徑串接,提供 TypeScript 腳本與蘋果捷徑的互動模板,TypeScript、Shortcuts,捷徑腳本範例

C. GCP 整合,服務選型,3.1 各服務用途需求比對,梳理功能對應服務－如Cloud Functions、Run、BigQuery等,GCP Comparison Docs,服務選型清單

C. GCP 整合,身份授權,3.2 Service Account 建置,建立服務專用帳戶及細粒度權限配置試驗,GCP IAM、Service Account,權限設計文檔

C. GCP 整合,身份授權,3.3 驗證用戶/應用程序身份,整合 Google Identity/Firebase Auth 驗證途徑,GCP Identity, Firebase,登入流程圖與示例

C. GCP 整合,SDK 整合,3.4 導入 cloud client for TypeScript,實作與包裝常用 GCP 服務 TS SDK,GCP Node.js SDK、TypeScript,封裝模組

C. GCP 整合,服務移轉/PoC,3.5 打造 TypeScript 術腳本-雲端自動部署,預計至少完成腳本雲上部署、Boost.space數據流串接,GCP Console、TS、API,雲端測試案例

C. GCP 整合,數據流策略,3.6 規劃數據儲存流向,依 Boost.space→Cloud Function/Run→BigQuery→前端流程設計,GCP Data Flow Tools,數據架構圖

D. GitHub 整合,倉庫建立,4.1 倉庫分類/權限設置,Monorepo/多Repo方案討論與設計,Git/GitHub,倉庫結構說明文件

D. GitHub 整合,分支管理,4.2 定義 Gitflow / PR 審查流程,按主線/開發/功能/熱修繕設分支規則,GitHub Branch/PR,分支協作規範

D. GitHub 整合,CI/CD流程,4.3 GitHub Actions 配置,規劃 Lint/Build/Test/Deploy至 GCP Artifact/CD,GitHub Actions、GCP,CI/CD yaml 檔

D. GitHub 整合,議題追蹤,4.4 Issue 群組/Project 進度追蹤,利用 GitHub Issues/Projects 建任務牆,GitHub Issues/Projects,專案管理指南

#### **表二 CSV 內容**

項目,子項,說明

主要語言,TypeScript,全端主力、高適應度

腳本語言,JS/TS/Python,擴充性佳、支援廣泛

雲平台,Google Cloud,高彈性&AI/ML前瞻

數據存儲,BQ, Firestore, Cloud SQL,依情境選用

版本控管,GitHub,協作成熟、CI/CD完善

效益,-,整合廣/自動化深/可靠彈性/AI友善

展望,-,腳本管理平台, Vertex AI 深化, DevOps 極致

有任何欄位/內容想再調整都可以補充說明喔！