

# Technical Specification: IRPC Smart Survey Calculator (Google Sheets Edition)

## 1. Project Overview (ภาพรวมโครงการ)

**Application Name:** IRPC Smart Survey Calculator **Purpose:** Web Application สำหรับวิศวกรและช่างสำรวจของ IRPC เพื่อใช้คำนวณวงรอบ (Traverse), พื้นที่ (Area), และปริมาตรดินตัด-ดินถม (Cut/Fill Volume) พร้อมระบบฐานข้อมูลบน Google Sheets **Target Users:** Civil Engineers, Surveyors, Project Managers.

## 2. System Architecture (สถาปัตยกรรมระบบ - ปรับปรุงใหม่)

เปลี่ยนมาใช้โครงสร้างแบบ **Google Workspace Integration** เพื่อความง่ายในการเข้าถึงข้อมูลและการดูแลรักษา

- **Frontend:** HTML5, Tailwind CSS, JavaScript (Client-side)
  - ทำงานเป็น Web App ที่ Deploy ผ่าน Google Apps Script
- **Backend / Middleware: Google Apps Script (GAS)**
  - ทำหน้าที่เป็น API Gateway รับ-ส่งข้อมูลระหว่างหน้าเว็บกับ Google Sheets
  - ใช้ฟังก์ชัน `doGet()` สำหรับเสิร์ฟหน้าเว็บ และ `doPost()` สำหรับรับส่งข้อมูล JSON
- **Database: Google Sheets**
  - ใช้เป็นฐานข้อมูลหลักในการเก็บข้อมูล Job, รายละเอียดจุดพิกัด (Points), และผลลัพธ์การคำนวณ
- **Authentication:** Google Account (Access Control ผ่านการแชร์สิทธิ์ของ Google Sheets/Script)

## 3. Application Specifications (รายละเอียดฟังก์ชัน)

### 3.1 User Interface (UI Layout)

#### A. Sidebar (แถบเมนูด้านซ้าย - ใหม่)

- **Filter by Year:** Dropdown เลือกปี พ.ศ. ของงานสำรวจ
- **Job List:** แสดงรายการงาน (Job No. - Job Name) ตามปีที่เลือก
- **Behavior:** เมื่อคลิกที่รายการ ข้อมูล Job นั้นจะถูกโหลดมาแสดงที่หน้าจอหลัก

#### B. Main Dashboard (หน้าจอหลัก)

- **Toolbar Buttons (ปุ่มคำสั่ง):**
  1. **[+ Create New Job]:** ปุ่มสำหรับสร้างหน้าจอเพื่อเริ่มงานใหม่ (Generate Job No. อัตโนมัติ)

2. **[Save]:** ปุ่มสำหรับบันทึกงานใหม่ลง Google Sheets (Insert Row)
  3. **[Update]:** ปุ่มสำหรับบันทึกการแก้ไขงานเดิมกลับไปยัง Google Sheets (Update Row)
  4. **[Print Report]:** ปุ่มสร้างรายงาน PDF
- **Project Info Section:** ระบุ Job No., Job Name, วันที่
  - **Input Section:** (เหมือนเดิม) ตารางกรอกพิกัด และโหมดการนำเข้าข้อมูล
  - **Result Section:** (เหมือนเดิม) กราฟรูปแปลง และผลลัพธ์การคำนวณ

3.2 Feature Requirements

- **Data Synchronization:**
  - **Read:** อ่านข้อมูลจาก Google Sheets มาแสดงผลที่ Sidebar และโหลดลง Form
  - **Write:** เขียนข้อมูลใหม่ต่อท้ายแถวเดิมใน Sheets (Append)
  - **Update:** ค้นหา Job ID เดิมและแก้ไขข้อมูลในแถวนั้น
- **Validation:** ตรวจสอบข้อมูลก่อน Save (เช่น ต้องมีจุดอย่างน้อย 3 จุด, Job Name ห้ามว่าง)

4. Database Schema (โครงสร้างข้อมูลใน Google Sheets)

ใช้ Google Sheet ชื่อแผ่นงาน (Tab Name): "JobData"

Column	Header Name	Data Type	Description
A	Job ID	String	รหัสอ้างอิง (Unique Key) เช่น JOB-2024-001
B	Job Name	String	ชื่อโครงการ
C	Year	Number	ปี พ.ศ. (ใช้สำหรับ Filter ที่ Sidebar)
D	Created Date	Date	วันที่สร้างงาน
E	Updated Date	Date	วันที่แก้ไขล่าสุด
F	Input Mode	String	NE , GPS , หรือ DMD
G	Target Elev	Number	ระดับอ้างอิง (Design Level)
H	Points Data (JSON)	String	เก็บข้อมูลจุดพิกัดทั้งหมดในรูปแบบ JSON String *

I	Results (JSON)	String	เก็บผลลัพธ์ (Area, Volume) เพื่อเรียกดูได้ทันทีไม่ต้องคำนวณใหม่
J	Surveyor Name	String	ชื่อผู้สำรวจ (User Email)

\* หมายเหตุ: การเก็บ Points เป็น JSON String ใน Cell เดียว ช่วยลดความซับซ้อนในการจัดการ Database แบบ Relational บน Spreadsheet

## 5. Backend Logic (Google Apps Script Functions)

ฟังก์ชันหลักที่ต้องมีในไฟล์ .gs :

### 5.1 doGet(e)

- ให้บริการไฟล์ HTML ( index.html ) แก่ผู้ใช้งาน

### 5.2 getJobList(year)

- อ่านข้อมูลจาก Sheet "JobData"
- Filter แถวที่มี Column C ตรงกับปีที่ระบุ
- Return Array ของ Object: [{id: 'JOB-001', name: 'Survey Road A', date: '...'}]

### 5.3 loadJobDetail(jobId)

- ค้นหา Job ID ใน Column A
- Return ข้อมูลทั้งหมดของแถวนั้น รวมถึงการ Parse JSON จาก Column H และ I กลับมาเป็น Object

### 5.4 saveNewJob(dataObj)

- รับค่าจาก Frontend
- สร้าง Job ID ใหม่ (ตรวจสอบลำดับล่าสุดจาก Sheet หรือใช้ UUID)
- sheet.appendRow([jobId, name, year, date, ...])

### 5.5 updateJob(dataObj)

- ค้นหา Job ID ใน Column A
- อัปเดตข้อมูลในแถวที่พบด้วยข้อมูลใหม่
- อัปเดต Column E (Updated Date) เป็นเวลาปัจจุบัน

## 6. Calculation Logic (Client-side vs Server-side)

เนื่องจากใช้ Google Sheets เป็น Database แนะนำให้ย้าย Logic การคำนวณ (Area, Volume, DMD) มาทำที่ **Client-side (JavaScript ใน Browser)** เพื่อลดภาระของ Apps Script Quota และทำให้ UI ตอบสนองได้ทันที (Real-time interaction)

Apps Script จะทำหน้าที่เป็นเพียง **Data Layer** (CRUD Operations) เท่านั้น

### Security & Access Control

- ตั้งค่า Apps Script Deployment เป็น **"Execute as Me"** (รันในนามเจ้าของไฟล์)
- ตั้งค่า Access เป็น **"Anyone within IRPC"** (เฉพาะคนในองค์กรที่มีอีเมลบริษัท)
- ข้อมูลใน Google Sheets จะถูกเข้าถึงผ่าน Script เท่านั้น ผู้ใช้ทั่วไปไม่จำเป็นต้องมีสิทธิ์ Edit ไฟล์ Sheet