LOW-CODE APP & EXTRACT-TRANSFORM-LOAD (ETL) AUTOMATION ON MICROSOFT POWER PLATFORM

Transforming Solar PV Engineering with a Centralized Data Hub that Saved 50% in Data Acquisition Time

Ayokunle Samuel Adenigba
Analytics Engineer I Power Platform Expert
ayo.adenigba@leitway.com
+49 178 1854390

PROJECT INTRODUCTION

EXECUTIVE SUMMARY

Low-Code ETL Pipeline

Developed a low-code app and ETL pipeline using Power Platform to automate solar meteorological data collection from multiple data providers and process them effectively.

Technology Integration

Leveraged Power Apps, Power Automate, SQL Server, and Power BI to streamline data acquisition, transformation, and analytics.

Efficiency Improvements

Achieved 50% reduction in data acquisition time and eliminated redundant purchases through centralized data management.

Engineering Empowerment

Provided engineers with reliable, on-demand datasets and actionable insights to enhance simulation accuracy and project turnaround.

EXISTING SITUATION

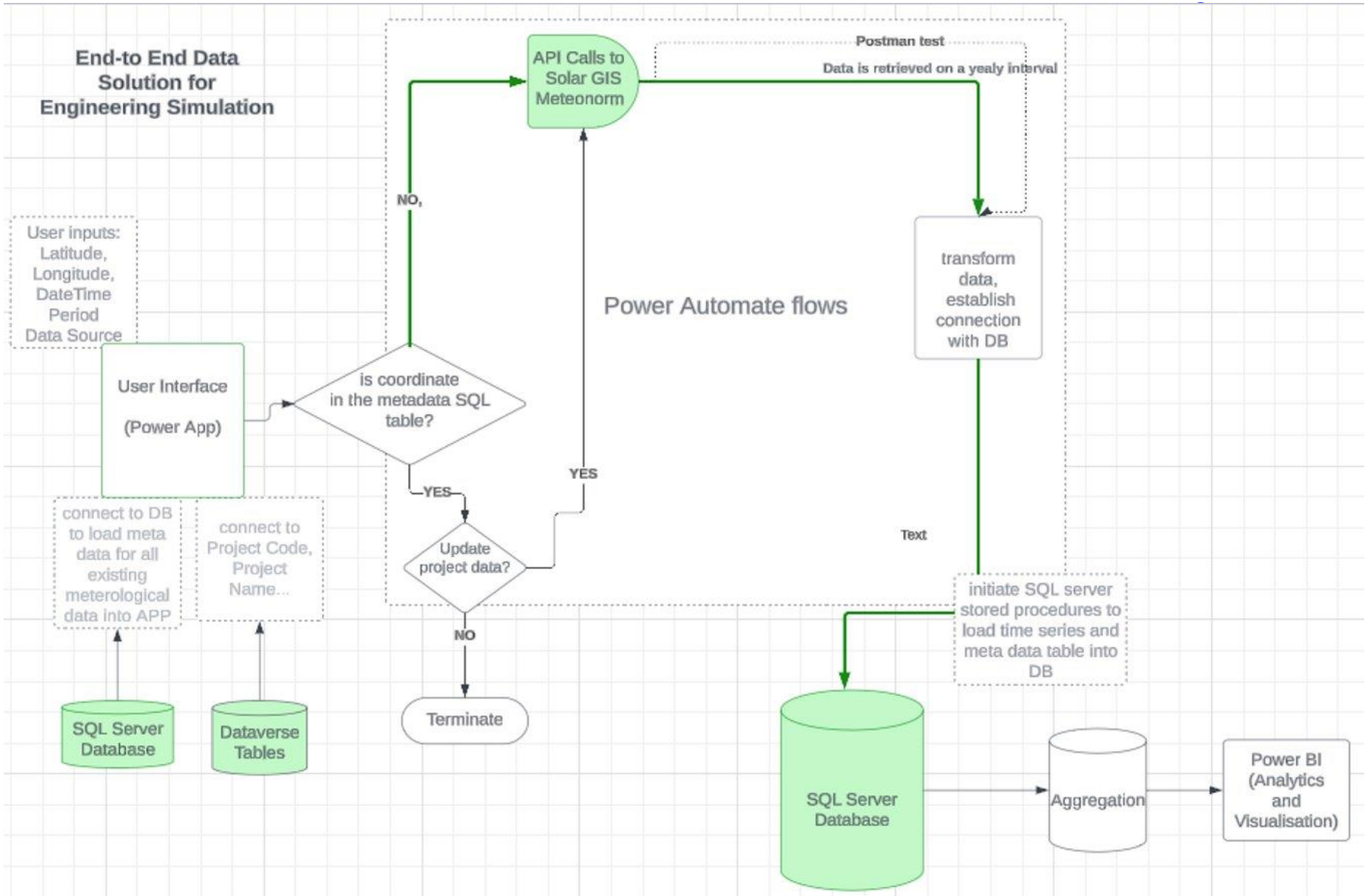
Solar PV teams and engineers situated at different locations must collaborate on projects development and system simulations. A fundamental requirement is the historical meteorological data for the proposed site. The existing process of collection, analysis and storage of this data followed a manual and fragmented approach, therefore providing little to no data governance, traceability and visibility, which are necessary for simulation accuracy, data integrity, and to avoid cost of of redundant data request.

Data description for a project: Up to 30 years of timeseries data on a 15-minute, hourly, daily or yearly interval from 3-4 different sources.
Development of PV system can take over a year depending on project size and complexity, so data update and traceability is paramount.

ARCHITECTURE AND WORKFLOW

ARCHITECTURE AT A GLANCE —

DATAFLOW DIAGRAM



SOLUTION ARCHITECTURE — LOW-CODE APP & ETL PIPELINE

User Data Intake

Power Apps provides an easy-to-use interface for submitting data with validation and deduplication.

Workflow Orchestration

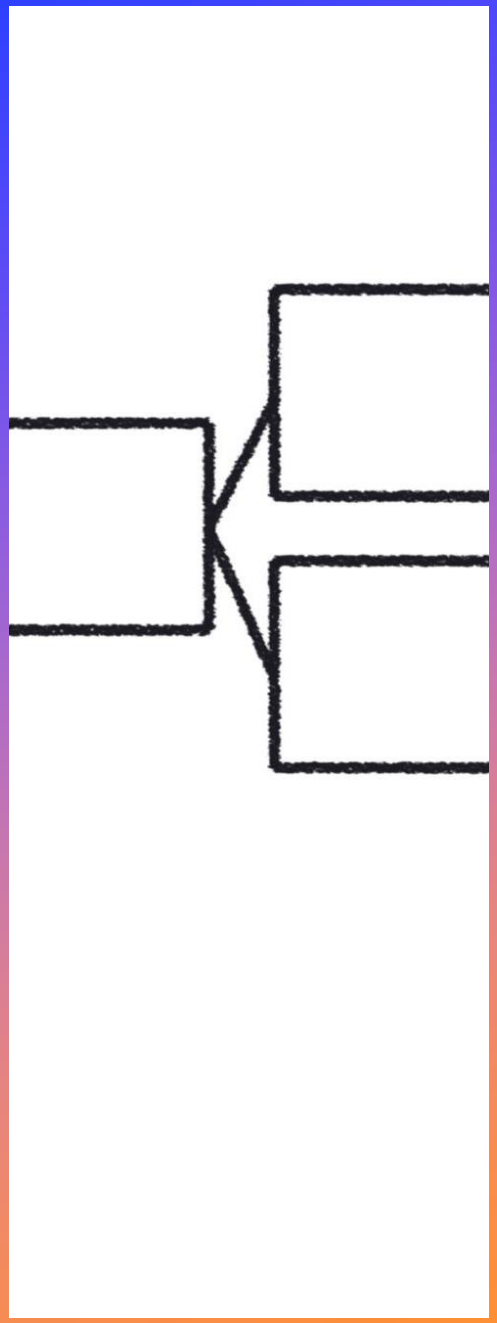
Power Automate manages data parsing, transformation, API calls, and asynchronous workflow polling.

Data Processing and Storage

SQL Server executes stored procedures to transform and store data into consistent, auditable formats.

User Notifications and Reporting

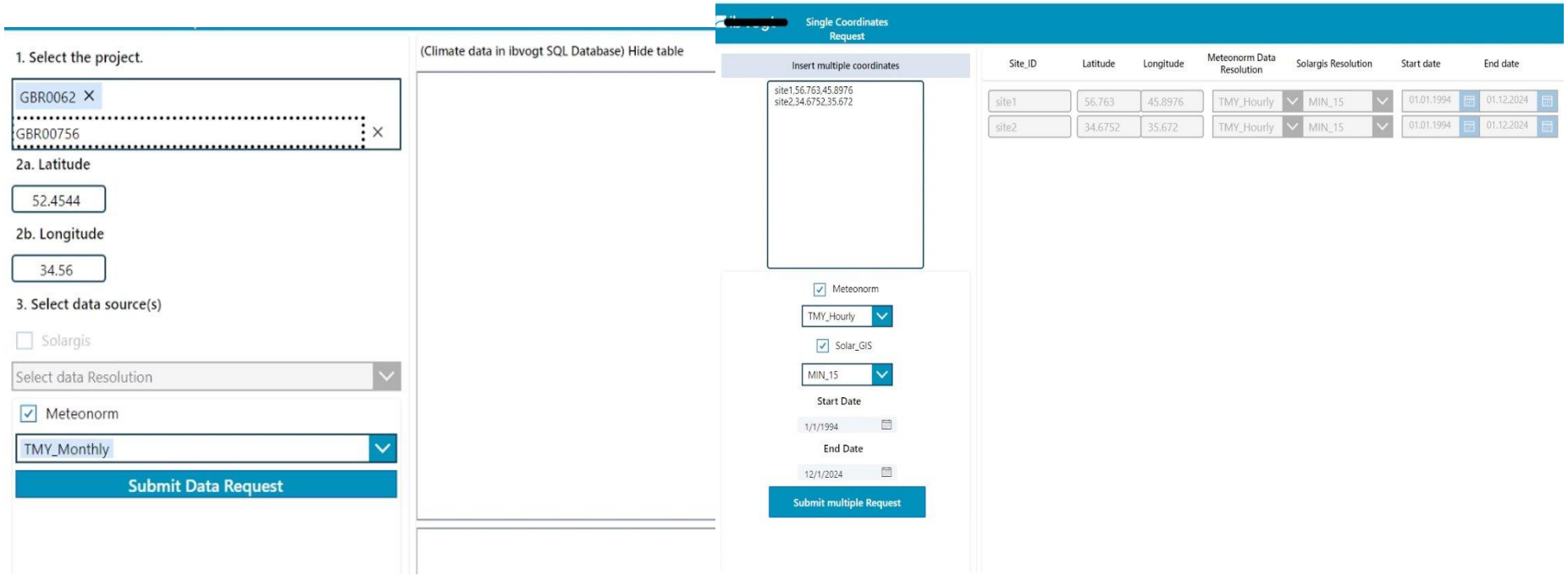
Notifications via Outlook and Teams keep users updated, while Power BI delivers interactive analytics dashboards.



POWER APP INTERFACE

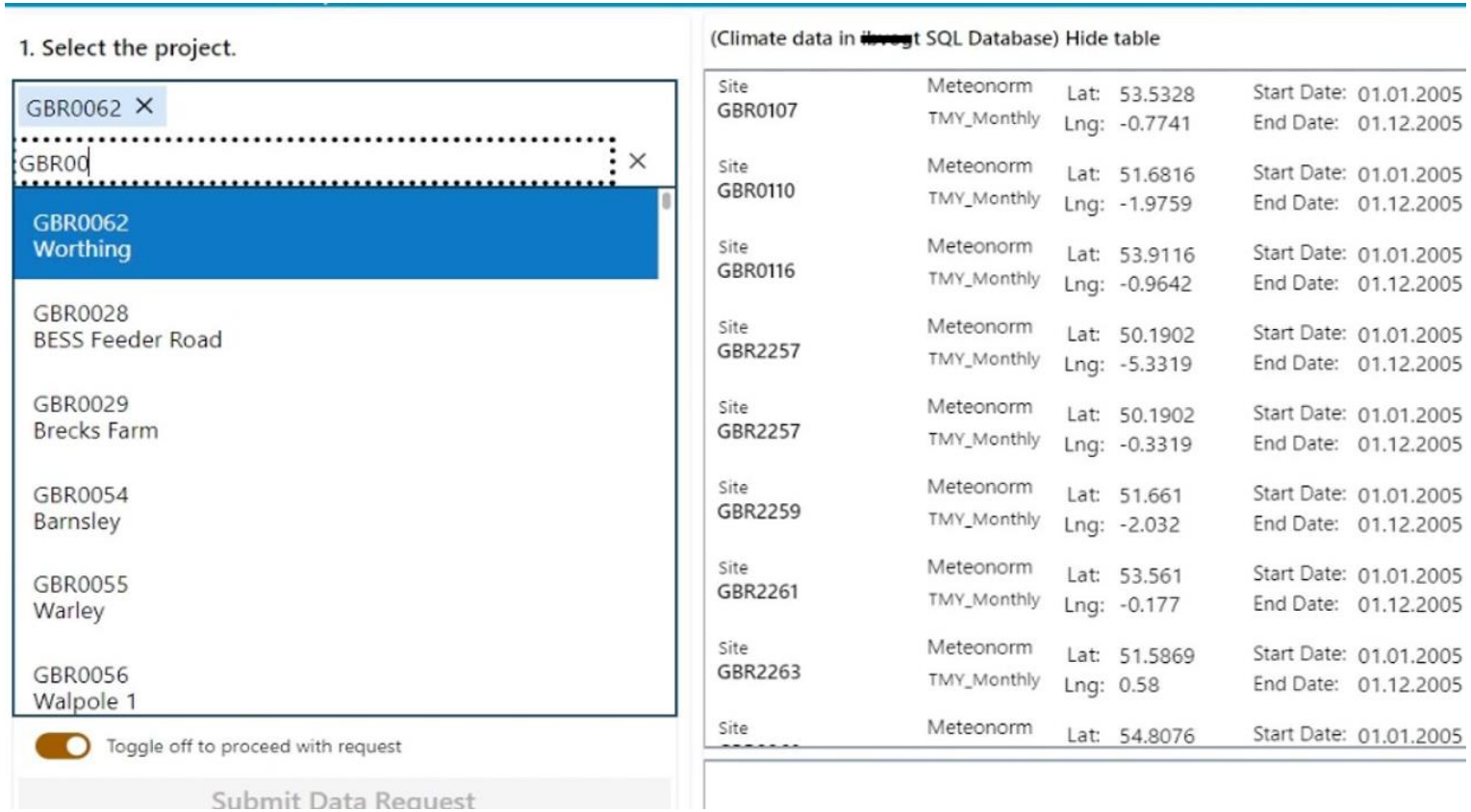
BUSINESS LOGIC AND INTAKE

FEATURE	DESCRIPTION
Project Selection	Users select existing projects from Dataverse to associate with data requests.
Location Input	Supports single and multiple coordinate entries for data acquisition.
Resolution Validation	Ensures users select appropriate data resolution before submission.
Duplicate Check	Alerts users if similar data already exists in the database.
Trigger Flow	Initiates Power Automate flow upon successful validation.


1. Select the project.
GBR0062
GBR00756
2a. Latitude
52.4544
2b. Longitude
34.56
3. Select data source(s)
Solargis
Select data Resolution
Meteonorm
TMY_Monthly
Submit Data Request
(Climate data in ibvogt SQL Database) Hide table
Single Coordinates Request
Insert multiple coordinates
site1,56.763,45.8976
site2,34.6752,35.672
Meteonorm
TMY_Hourly
Solar_GIS
MIN_15
Start Date
1/1/1994
End Date
12/1/2024
Submit multiple Request
Site_ID
Latitude
Longitude
Meteonorm Data Resolution
Solargis Resolution
Start date
End date
site1
56.763
45.8976
TMY_Hourly
MIN_15
01.01.1994
01.12.2024
site2
34.6752
35.672
TMY_Hourly
MIN_15
01.01.1994
01.12.2024

POWER APP INTERFACE

Activity: Selection of a project from dataverse. (Image - left side)
change images here to single snd multiple screen request



Power FX code within the app

```
If(
CountRows(SolargisTMYmultiRequest)> 0,
ForAll(SolargisTMYmultiRequest,
Patch(ResponseTable, Defaults(ResponseTable),
{ResponseDetails: SolargisTMYnewDataRequest.Run(JSON(ThisRecord)
));

/*
If(
CountRows(SolargisTMYmultiRequest)> 0,
ClearCollect(ResponseTable,
ForAll(SolargisTMYmultiRequest,
{ResponseDetails: SolargisTMYnewDataRequest.Run(JSON(ThisRecord)
));*/

If(CountRows(MeteonormMultiRequest)>0,
ForAll(MeteonormMultiRequest,
Patch(ResponseTable, Defaults(ResponseTable),
{ResponseDetails: MeteonormNewRequest.Run(JSON(ThisRecord), varU

Set(showLoadingImage, false);
Navigate(Request_StatusScreen);
```

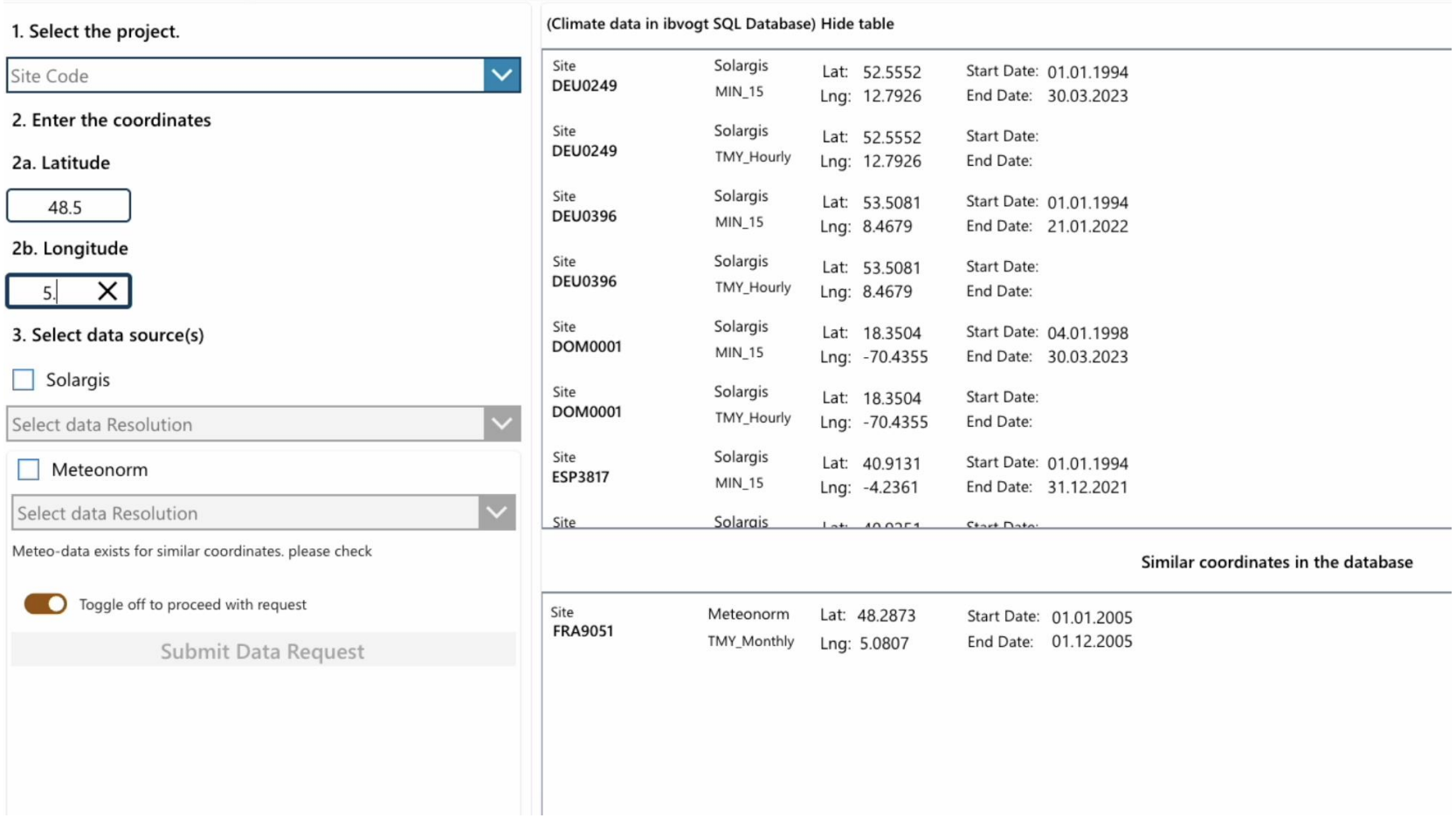
POWER APP INTERFACE

Intelligent Time & Cost Savings:

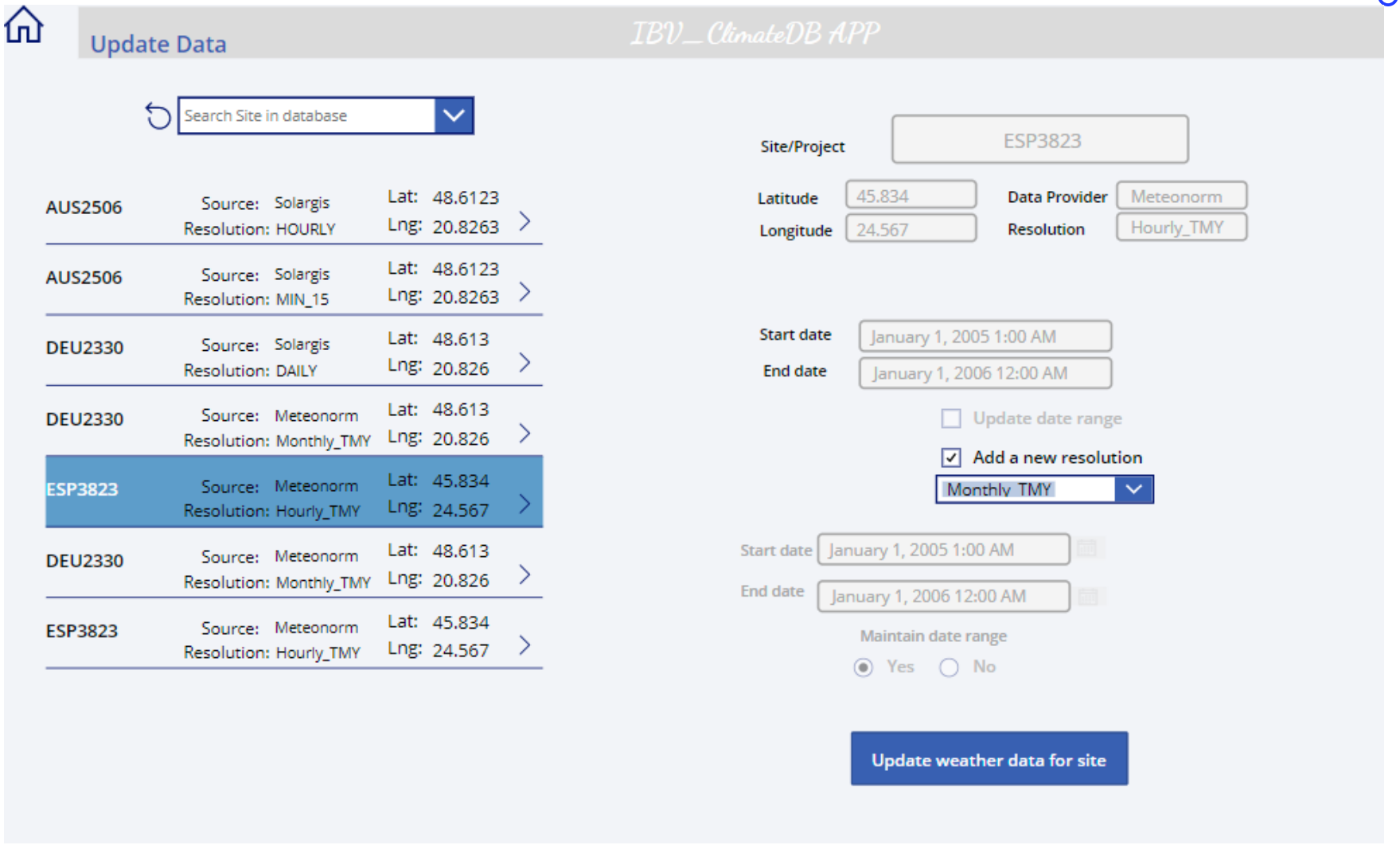
- Proactively notified users if data for a site already existed, preventing redundant API calls and saving cost.

(Image: arrow pointers)

- Trigger power Automat flow and inform user of a successfully initiated request.



Notify user: data exist for close location (coordinate)

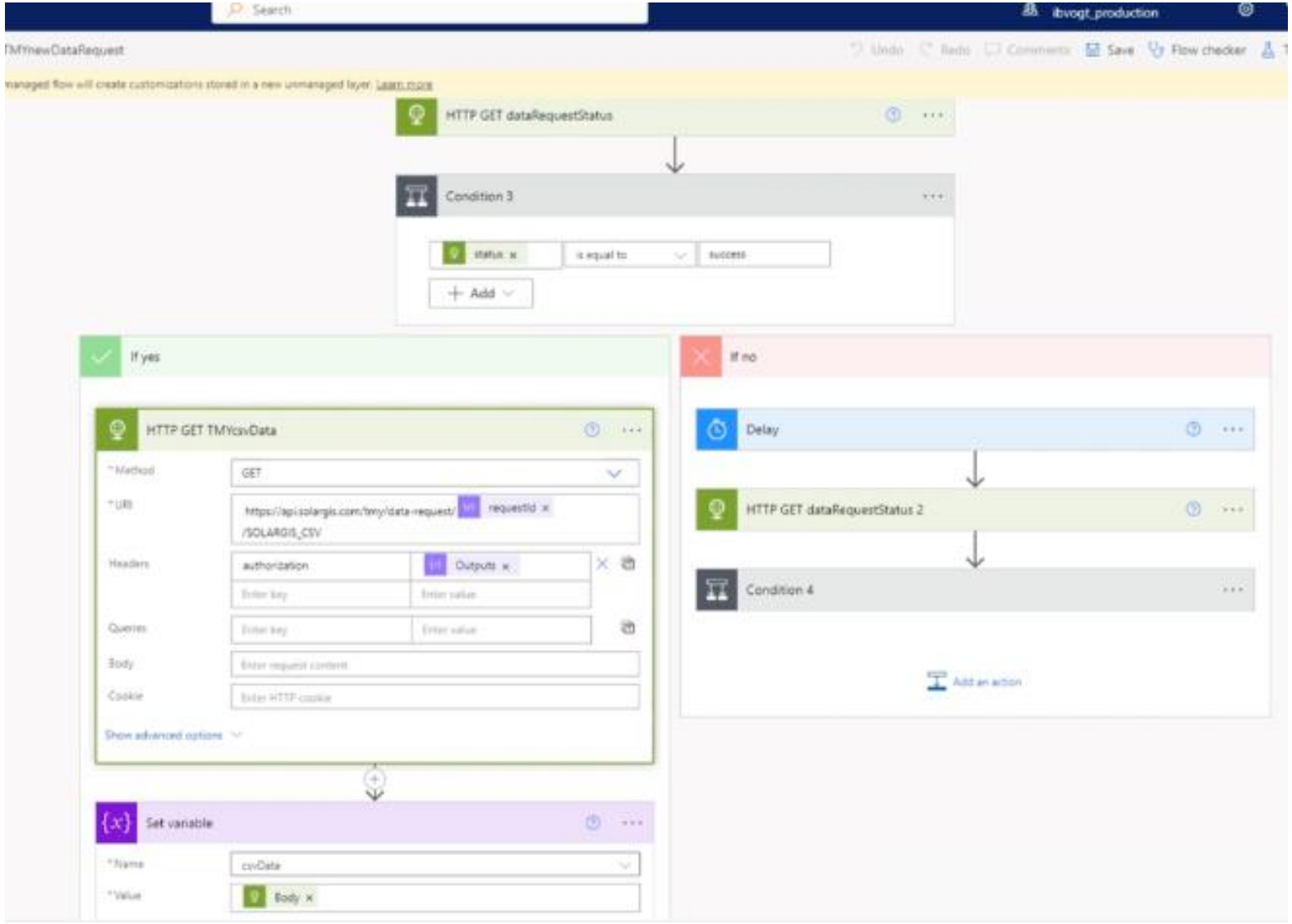

Update Data
IBV_ClimateDB APP
Search Site in database
AUS2506 Source: Solargis Resolution: HOURLY Lat: 48.6123 Lng: 20.8263
AUS2506 Source: Solargis Resolution: MIN_15 Lat: 48.6123 Lng: 20.8263
DEU2330 Source: Solargis Resolution: DAILY Lat: 48.613 Lng: 20.826
DEU2330 Source: Meteonorm Resolution: Monthly_TMY Lat: 48.613 Lng: 20.826
ESP3823 Source: Meteonorm Resolution: Hourly_TMY Lat: 45.834 Lng: 24.567
DEU2330 Source: Meteonorm Resolution: Monthly_TMY Lat: 48.613 Lng: 20.826
ESP3823 Source: Meteonorm Resolution: Hourly_TMY Lat: 45.834 Lng: 24.567
Site/Project ESP3823
Latitude 45.834
Longitude 24.567
Data Provider Meteonorm
Resolution Hourly_TMY
Start date January 1, 2005 1:00 AM
End date January 1, 2006 12:00 AM
Update date range
Add a new resolution
Monthly_TMY
Start date January 1, 2005 1:00 AM
End date January 1, 2006 12:00 AM
Maintain date range
Yes
No
Update weather data for site

POWER AUTOMATE ORCHESTRATION

DATA EXTRATION AND TRANSFORMATION

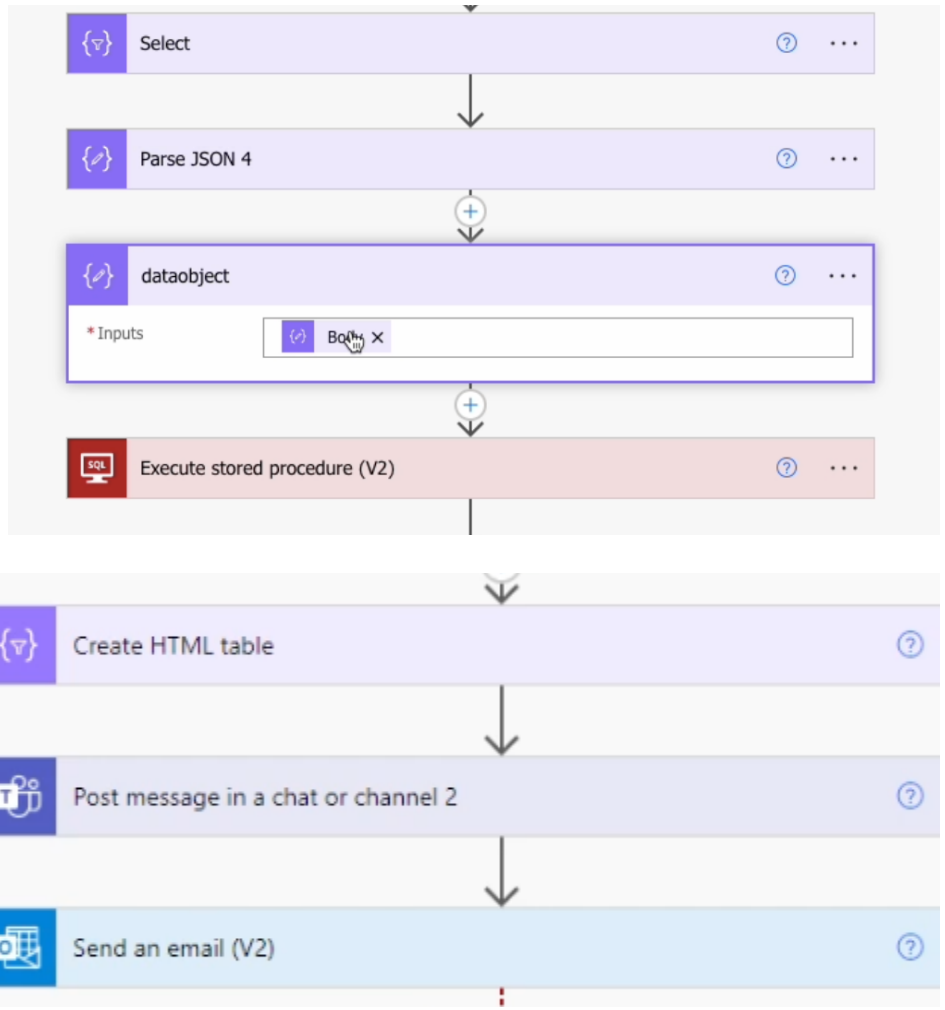
ACTION	PURPOSE
HTTP POST	Submit data request to external APIs.
Polling	Check status of asynchronous jobs until completion.
Condition Checks	Validate response success before proceeding.
Error Handling	Log and notify users of failed requests.
HTTP GET	Collect actual data (Extraction)
Data Transformation	Align API response with SQL schema. (JSON, XML processing)

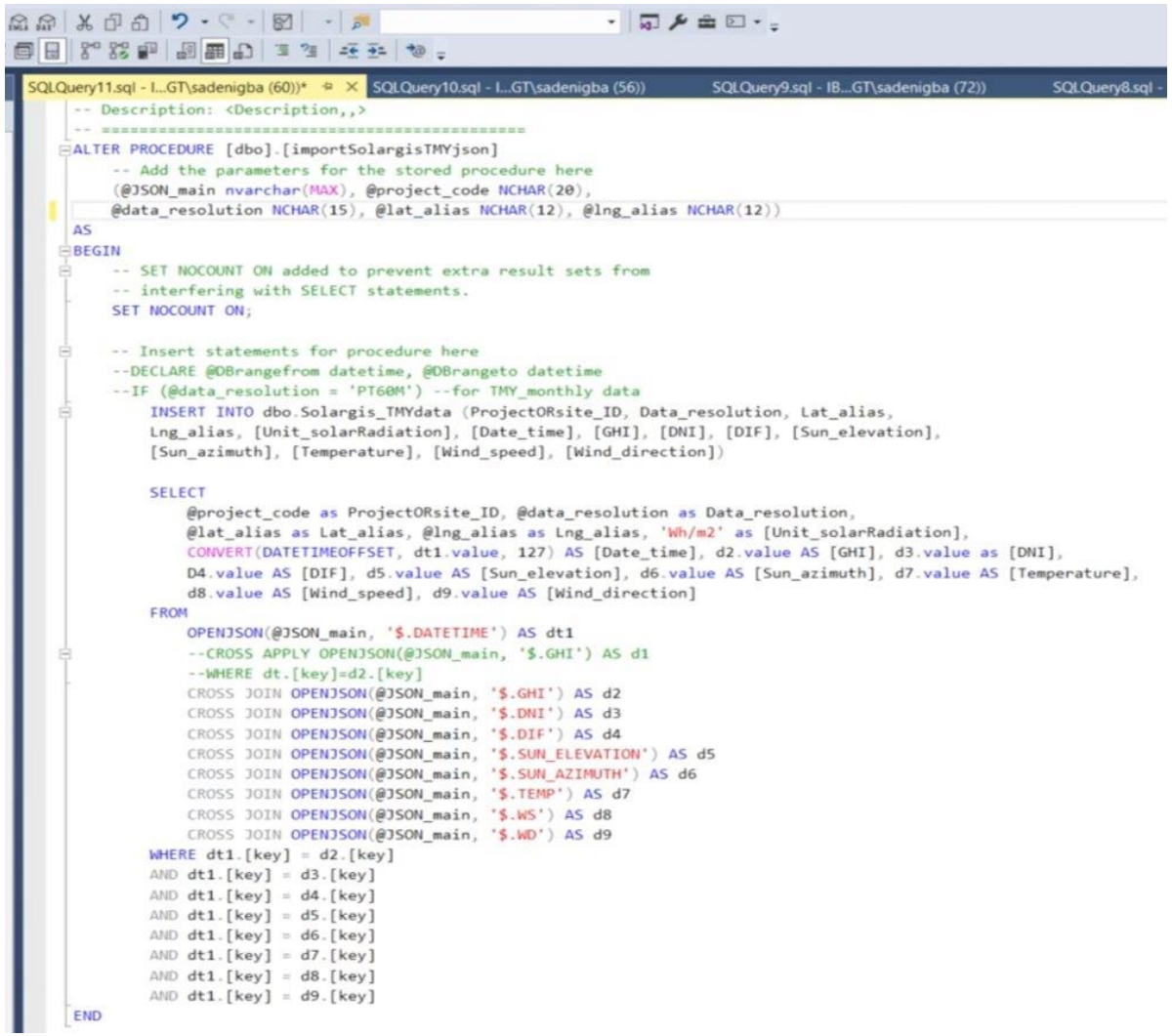
POWER AUTOMATE FLOW SNIPPET



DATA LOADING AND NOTIFICATIONS

ETL - DATA LOADING





```
-- Description: <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[importSolargisTMYjson]
    -- Add the parameters for the stored procedure here
    (@JSON_main nvarchar(MAX), @project_code NCHAR(20),
    @data_resolution NCHAR(15), @lat_alias NCHAR(12), @lng_alias NCHAR(12))
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    --DECLARE @DBrangefrom datetime, @DBrangeto datetime
    --IF (@data_resolution = 'PT60M') --for TMY_monthly data
        INSERT INTO dbo.Solargis_TMYdata (ProjectORsite_ID, Data_resolution, Lat_alias,
        Lng_alias, [Unit_solarRadiation], [Date_time], [GHI], [DNI], [DIF], [Sun_elevation],
        [Sun_azimuth], [Temperature], [Wind_speed], [Wind_direction])

        SELECT
            @project_code as ProjectORsite_ID, @data_resolution as Data_resolution,
            @lat_alias as Lat_alias, @lng_alias as Lng_alias, 'Wh/m2' as [Unit_solarRadiation],
            CONVERT(DATETIMEOFFSET, dt1.value, 127) AS [Date_time], d2.value AS [GHI], d3.value as [DNI],
            D4.value AS [DIF], d5.value AS [Sun_elevation], d6.value AS [Sun_azimuth], d7.value AS [Temperature],
            d8.value AS [Wind_speed], d9.value AS [Wind_direction]
        FROM
            OPENJSON(@JSON_main, '$.DATETIME') AS dt1
            --CROSS APPLY OPENJSON(@JSON_main, '$.GHI') AS d1
            --WHERE dt.[key]=d2.[key]
            CROSS JOIN OPENJSON(@JSON_main, '$.GHI') AS d2
            CROSS JOIN OPENJSON(@JSON_main, '$.DNI') AS d3
            CROSS JOIN OPENJSON(@JSON_main, '$.DIF') AS d4
            CROSS JOIN OPENJSON(@JSON_main, '$.SUN_ELEVATION') AS d5
            CROSS JOIN OPENJSON(@JSON_main, '$.SUN_AZIMUTH') AS d6
            CROSS JOIN OPENJSON(@JSON_main, '$.TEMP') AS d7
            CROSS JOIN OPENJSON(@JSON_main, '$.WS') AS d8
            CROSS JOIN OPENJSON(@JSON_main, '$.WD') AS d9
        WHERE dt1.[key] = d2.[key]
        AND dt1.[key] = d3.[key]
        AND dt1.[key] = d4.[key]
        AND dt1.[key] = d5.[key]
        AND dt1.[key] = d6.[key]
        AND dt1.[key] = d7.[key]
        AND dt1.[key] = d8.[key]
        AND dt1.[key] = d9.[key]
END
```

SQL STORED PROCEDURES AND USER ALERTS

PROCEDURE	FUNCTION
Execute Stored Procedure	Load data into SQL tables.
Update Row	Modify existing entries based on new data.
Remove XML Header	Clean API response for SQL compatibility.
Notification	Inform users of success or failure via email/Teams.
Audit Logging	Track request ID, rows ingested, and errors.

ANALYTICS AND INSIGHTS

POWER BI DOWNSTREAM ANALYSIS

Data Insights

Power BI provides engineers with insights by connecting directly to SQL Server tables and views using Import modes.

Data Quality and Analysis

DAX measures analyze data quality, availability, and engineering constraints to ensure reliable and accurate insights.

Visualization of Key Metrics

Dashboards visualize metrics such as data coverage, resolution distribution, and request status for comprehensive monitoring.

Integration with Simulation Tools

Power BI supports export interfaces and dataformat to simulation software(PVSyst), enabling seamless engineering workflow integration.

POWER BI DASHBOARD

TECHNICAL DESIGN AND OUTCOMES

ETL DESIGN CHOICES AND BUSINESS IMPACT

Robust ETL Architecture

The design uses staging tables, schema-aligned transformations, and MERGE upserts for reliability and scalability.

Idempotency and Polling

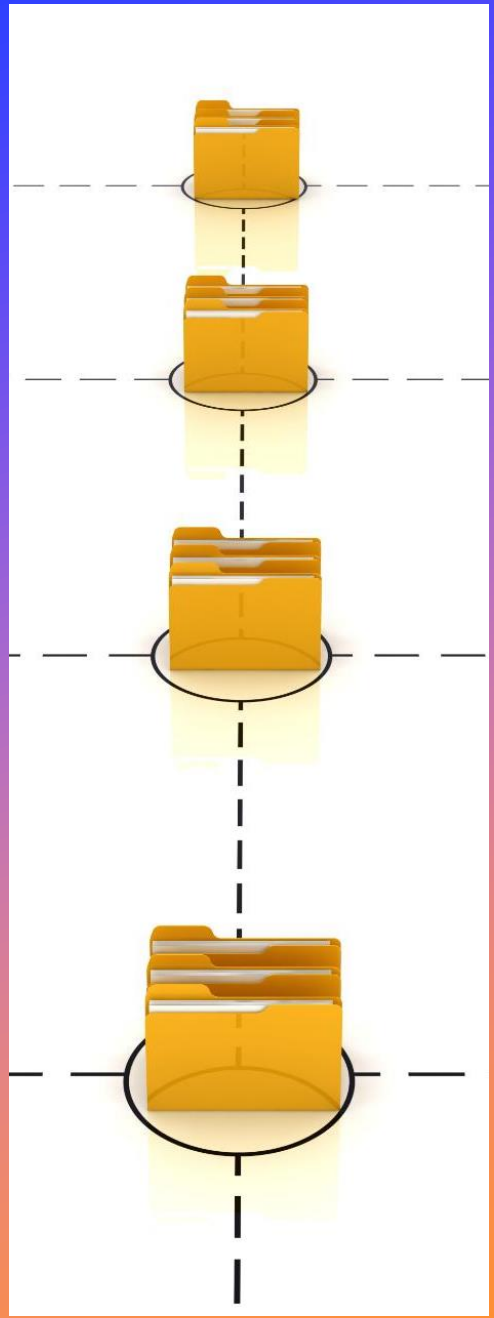
Composite keys ensure idempotency; polling with exponential backoff manages asynchronous API jobs efficiently.

Logging and Security

Comprehensive logging tracks IDs, latency, and errors; secure connectivity is maintained via gateways and connection references.

Business Impact and Future Plans

ETL improvements reduced acquisition time by 50%, eliminated redundant purchases, and enhanced simulation accuracy.



TEAM WORK & RESPONSIBILITIES

My responsibilities

- Development of the Power App user interface and Logic functions
- API integration with Solar GIS, Meteonorm.
- Development of Power Automate flows and logics.
- Data modelling , design and implementation in SQL Server, including stored procedures.
- Documentations

Collaborative effort and other aspect of the development

- Solution architecture and design
- Development of Power BI dashboards for data visualization and analysis.
- Testing and deployment and documentation of the solution.

RECOMMENDATION

Abhishek Gawde · 1st

Driving Business Growth through Digital Transformation | Microsoft Power Platform Expert | Automation & AI |

I had the privilege of working with Samuel Adenigba, a seasoned professional whose expertise in solar PV project analysis and energy performance optimization is truly exceptional. Samuel's contributions have been instrumental in driving the success of solar projects, through critical milestones such as project sales, commissioning, and operational performance.

Samuel is a master of leveraging data and technology to solve complex challenges. His proficiency in tools like Microsoft Fabric, Power Platform, SQL, and Python enables him to build scalable data solutions and automate processes that significantly enhance business intelligence and asset performance. His work in orchestrating big data pipelines, developing automation tools, and optimizing PV plant designs demonstrates his innovative approach to driving efficiency and value.

What sets Samuel apart is not just his technical acumen but also his ability to bridge the gap between technical and non-technical teams. His ability to communicate requirements and align stakeholders ensures due diligence across all phases of solar projects.

If you're developing a solar project or tackling the challenges of underperforming assets, Samuel is the expert you need. His hands-on experience in energy simulations, financial analysis, and engineering design, combined with his deep industry knowledge, makes him an invaluable asset to any team or organization.

It’s been an absolute pleasure collaborating with Samuel, and I highly recommend him for his unparalleled expertise and dedication to excellence in the solar energy sector.