# Aquarius Automation Project Documentation

## 1. Project Overview

This project automates the process of exporting employee timesheet data from PostgreSQL, transforming it into CSV format, storing it in OneDrive, and enabling Power BI dashboards with auto-refresh. The automation ensures up-to-date reporting without manual intervention.

## 2. End-to-End Workflow

| Step | Description |
|---|---|
| 1 | SQL script runs in PostgreSQL to fetch timesheet data. |
| 2 | Python script (developed in VS Code) exports data to CSV. |
| 3 | Python script uploads the CSV file to OneDrive. |
| 4 | Power BI connects to OneDrive CSV with auto-refresh enabled. |
| 5 | Power Query cleans and validates data (date standardization, missing values handling, renaming). |
| 6 | DAX measures and calculated columns build KPIs and analytics. |
| 7 | Dashboards visualize insights (KPI Cards, Pie, Funnel, Bar charts, and Slicers). |

## 3. Power BI – Data Cleaning, Validation, and Dashboard

**Transformations in Power Query:**

- Remove rows with missing/invalid hours.
- Standardize dates into consistent format.
- Rename columns for clarity.

**DAX and Calculated Columns:**

- Average weekly hours per employee.
- Weekly anomalies detection (employees with <5 hrs/week for >1 week).
- Pending hours tracker by department and work type.

**Dashboard Visuals:**

- KPI Cards (avg weekly hours, anomalies, pending hours).
- Pie chart: Timesheet approval summary (pending, approved, rejected).
- Bar chart: Active vs Total employees by department.
- Funnel chart: Employees exceeding 420 hrs with drill-through table.
- Slicers: Month, Quarter, Department.
- Drill Through, Tooltip Page, Bookmarks

## 4. Automation with Task Scheduler

Windows Task Scheduler ensures the Python script runs automatically at scheduled intervals.

**Key Points:**

- VS Code is only for development and testing.
- Python script (.py file) is saved and executed directly by Task Scheduler.

## 5. Final Data Flow