

# Billed Reports Automation System

**Category:** Reporting and dashboards

---

**Problem:** Weekly billed report imports required manual validation, categorization, and duplicate checks.

---

**What I Built:** Created a controlled CSV ingestion engine with automated categorization, duplicate prevention, staff summaries, and structured reporting views.

---

**Tools Used:** Google Sheets, Google Apps Script, Google Drive, MailApp, Custom Sidebar UI

---

**Output:** Clean billed data sheet, staff efficiency report, structured email summary

---

# Billed Reports Automation System

Google Sheets + Drive + Apps Script + Custom Sidebar

---

## Quick backstory

This started because billed reports were technically “working”... but quietly consuming hours every week.

CSV files were exported from one system.

They were uploaded manually.

Headers had to be checked.

Practitioner names needed cleaning.

Categories were applied manually.

Travel lines were double-checked.

Duplicates had to be spotted by eye.

Then someone still had to filter, summarize, and build a staff breakdown for management.

Nothing was broken.

But it was repetitive, fragile, and dependent on someone remembering all the rules.

So I turned it into a structured workflow inside Google Sheets, using the tools the team already relies on.

---

## The Problem

Each billed report cycle looked something like this:

- Download CSV export
- Check if headers match the master sheet
- Clean practitioner names
- Split job roles manually
- Categorize service types (NDIS, HCP, Private, Travel, etc.)
- Adjust quantity based on item price

- Cap travel entries when required
- Remove duplicate entries
- Skip excluded practitioners
- Manually move files into archive folders
- Build a filtered staff summary
- Double-check nothing was missed

If this is done weekly, it easily becomes:

- 1.5 to 3 hours per cycle
- 6 to 12 hours per month
- Continuous risk of small but expensive mistakes

The real issue was not just time.  
It was inconsistency and cognitive load.

---

## The System I Built

Instead of treating billed reports as a spreadsheet task, I turned it into a guided operational flow.

---

### Step 1: Controlled CSV Upload

Inside the spreadsheet, users open a custom branded sidebar.

They can:

- Drag and drop a CSV file
- Or click to upload
- The file must contain “billedItemReport” in its name
- Duplicate filenames are blocked
- Invalid files are rejected

No guessing. No messy imports.

---

## Step 2: Smart Import Engine

When “Run Billed Report” is clicked, the system:

- Validates headers strictly
- Auto-repairs minor header mismatches if allowed
- Skips specific practitioners
- Cleans and normalizes data
- Splits Practitioner into:
  - Name (Column G)
  - Job Role (Column P)
- Derives job role from item text if missing
- Categorizes services automatically into:
  - NDIS Face to Face
  - NDIS Non Face to Face
  - NDIS Written Reports
  - Home Care Package
  - Private Patient
  - Provider Travel
- Computes Quantity based on item type and price
- Applies a special rule:
  - If an invoice contains NDIS and a dated Provider Travel line, travel quantity is capped at 0.25 when needed
- Builds a composite key to prevent duplicates
- Skips already imported entries
- Appends only clean, validated rows

- Renames the sheet with a Manila timestamp if new rows were added
- Moves processed files to a Drive “Processed” folder
- Sends a structured success or failure email summary

What used to be manual validation across multiple columns now runs in seconds.

---

### **Step 3: Quick Views Inside the Sheet**

I added a custom THC Tools sidebar with:

- Staff Efficiency view
- Appointment Tracking view
- Clear Filters option

Each view is powered by hidden helper columns and controlled filter criteria.

Users don’t touch formulas.

They don’t manually apply filters.

They just click.

---

### **Step 4: Staff Efficiency Extraction**

When Staff Efficiency view is active, users can:

- Choose presets:
  - This week
  - Last week
  - This month
  - Last month
- Or define a custom date range

The system then:

- Filters only valid rows

- Aggregates quantity per practitioner
- Builds a pivot-style sheet automatically
- Adds grand totals
- Formats the report
- Activates the new sheet

No manual pivot tables.  
No copy-paste summaries.

---

## The Tools Used

- Google Apps Script
- Google Sheets automation
- Google Drive file handling
- MailApp email summaries
- Custom HTML and CSS sidebar
- Dynamic filter criteria
- Timezone-aware timestamp handling

All built inside Google Workspace.  
No third-party subscription tools.

---

## The Measurable Result

Before automation:

- 1.5 to 3 hours per billed report cycle
- Manual header checks
- Manual category validation

- Manual deduplication
- Manual filtering for staff summaries
- High cognitive switching between Drive and Sheets

After automation:

- Under 5 minutes of human interaction
- Automated validation
- Automated categorization
- Automatic duplicate prevention
- One-click filtered views
- Instant pivot-style staff breakdown
- Automatic file movement
- Structured success/failure email logs

Estimated time savings:

- 6 to 10 hours per month
- 70 to 85 percent reduction in manual effort
- Near elimination of duplicate-entry risk

That is 70 to 120 hours saved per year for a small allied health team running regular billed reports.

---

## Why This Matters Beyond Billed Reports

This structure can be reused for:

- Payroll imports
- Aged debtor updates
- Invoice reconciliation

- KPI dashboards
- Weekly operations reports
- Recurring executive summaries
- Follow-up tracking systems
- Controlled bulk email workflows

Anything that starts with:

“We export a CSV and then manually clean it...”

can be converted into a guided internal system like this.

It is not about adding new software.

It is about making the existing system work properly.