



# An Internal Assistant That Never Guesses

---

A staff knowledge assistant built for a medical center, grounded only in real documents.

AI KNOWLEDGE BASE

## CASE STUDY

# An Internal Assistant That Never Guesses

How I built a staff knowledge assistant for a medical center that reads every document automatically, answers questions with the exact source cited, and plainly says it does not know rather than making something up.

## The situation

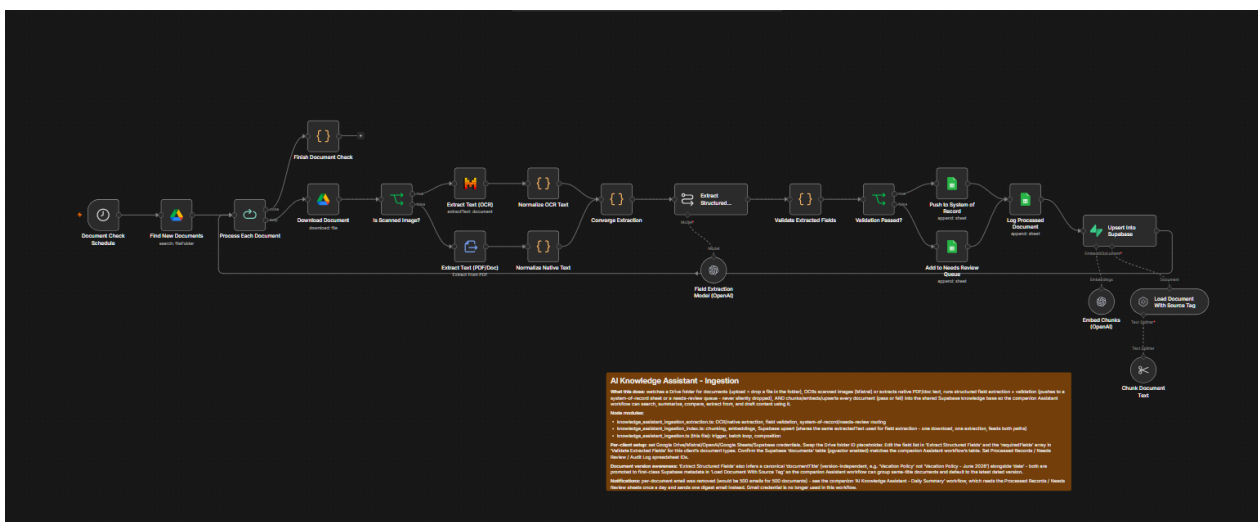
St. Celeste Medical Center had policies, procedures, and internal documents spread across folders, some scanned, some typed, some updated more recently than others. Staff needed fast, accurate answers, and in a medical setting there was no room for an assistant that guessed when it was not sure or pulled from an outdated version of a document without realising it.

## What I built

Two connected systems. One quietly watches for new documents, reads them, and files them into a searchable knowledge base the moment they land. The other is what staff actually talk to, one assistant that can answer questions, summarise, compare, extract details, or draft content, always grounded in the real documents on file, never invented.

## Every new document gets read and filed automatically

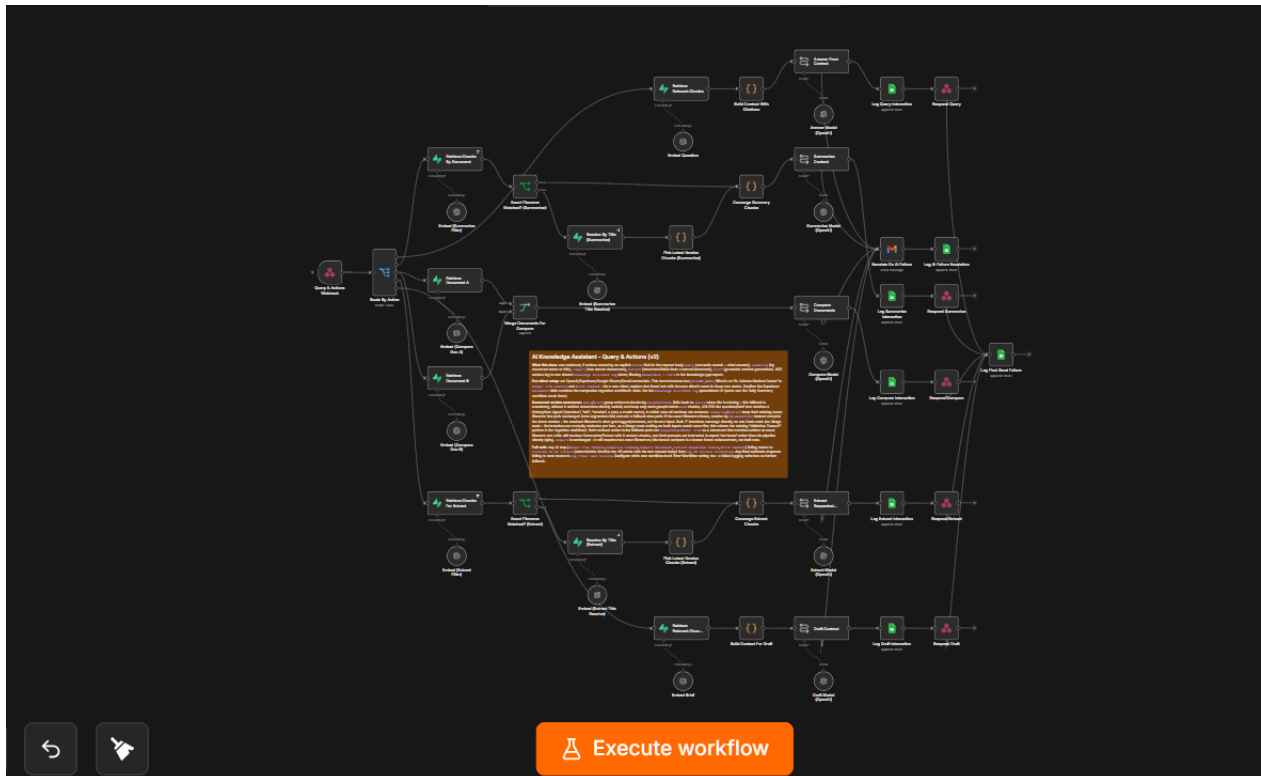
The system checks for new documents every fifteen minutes. A scanned image gets read with proper text recognition, a typed document has its text pulled out directly, and either way the key details get extracted automatically. If anything looks incomplete, it still goes into a review list rather than vanishing, and either way the document becomes searchable straight away.



The document intake system, live and running for St. Celeste Medical Center.

## One assistant, five different ways staff actually use it

The same assistant can answer a question and name exactly which document it came from, summarise an entire document in plain language, compare two documents side by side to show what genuinely differs between them, pull out specific details someone needs without reading the whole thing, or draft new content, all grounded only in what is actually on file.



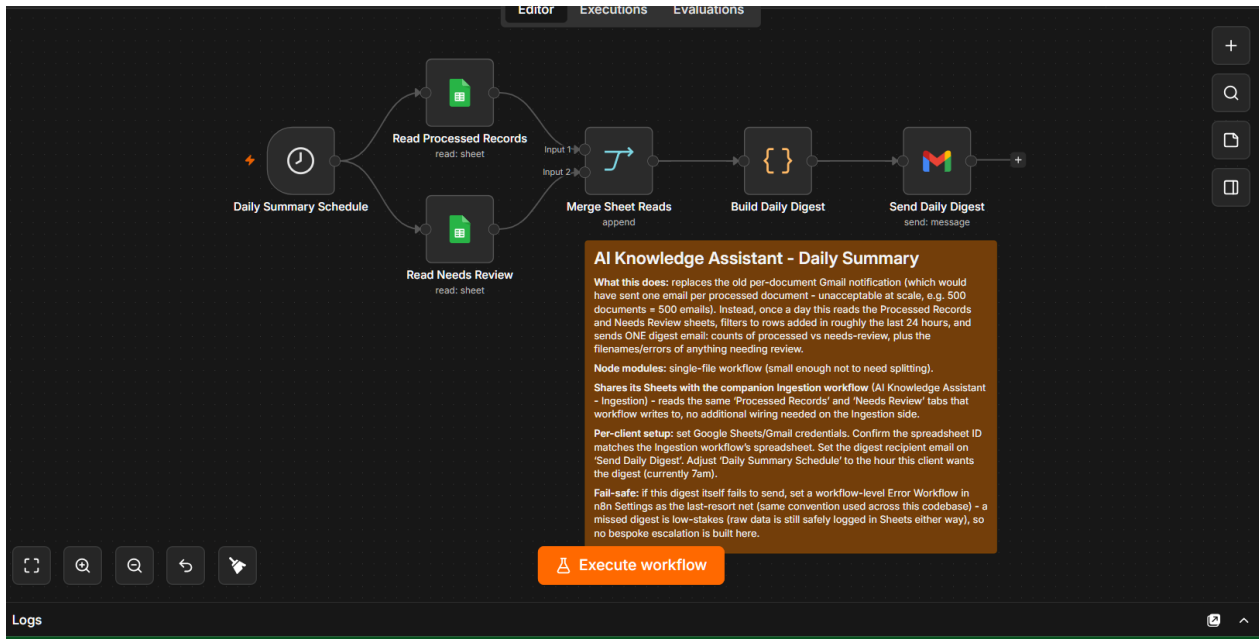
*The assistant staff actually talk to, live and running for St. Celeste Medical Center.*

### It always tells the truth about what it knows

If the knowledge base genuinely does not contain the answer, the assistant says so directly instead of guessing, which matters enormously in a medical setting. It also knows the difference between an old version of a document and the newest one, and defaults to the latest version unless someone specifically asks about an earlier one.

### One clear digest a day, not a flood of emails

Rather than sending a separate email every time a single document is processed, which would mean hundreds of emails on a busy day, one summary lands each morning with how many documents came in and which ones need a second look.



*The daily digest that replaced a flood of per document emails.*

## Nothing fails without someone knowing

If any step of the assistant genuinely fails, or a response cannot be delivered, it is automatically escalated and logged rather than quietly disappearing, so a failure is always seen, never hidden.

## What it actually changed

- Staff get accurate, cited answers instantly instead of digging through shared folders or asking around.
- Every document gets read, filed, and made searchable automatically, scanned images included.
- Answers are always grounded in real documents, the assistant says so plainly when it does not know rather than guessing.
- Comparing two policies, or pulling one detail out of a long document, takes seconds instead of a manual read through.
- One digest a day keeps everyone informed without burying anyone's inbox.

## THE RESULT

# What Changed Once Every Document Got Read Automatically

This is not about one flashy number. It is about what happens the moment a document lands, and whether staff can actually trust the answer they get back.

### BEFORE

- Staff had to dig through shared folders to find the right document, if they could find it at all.
- Scanned documents often sat unsearchable, as an image no one could search or read from.
- There was no reliable way to know if you had the latest version of a policy or an old one.
- Comparing two versions of a document meant opening both and reading them side by side by hand.
- An assistant that was not sure would either guess or stay silent, neither of which is safe here.

### AFTER

- ✓ Every document is read, filed, and searchable within minutes of landing, scans included.
- ✓ Staff get an answer with the exact source cited, not a best guess.
- ✓ The assistant defaults to the latest version of a document, unless someone asks for history.
- ✓ Comparing two documents, or pulling one detail out of a long one, takes seconds.
- ✓ If the knowledge base does not have the answer, it says so plainly instead of guessing.

# From A Document Landing To A Trusted Answer

---

1

## A document lands

Every fifteen minutes, the system checks the shared folder for anything new, a scanned image or a typed file, it makes no difference which.

2

## It gets read properly

Scanned pages are read with proper text recognition, typed documents have their text pulled directly. Either way, the words become something the system can actually work with.

3

## The key details get pulled out and checked

Names, dates, document type, and a plain title are extracted automatically, and checked for anything missing before the document moves on.

4

## It becomes part of the knowledge base

The document gets filed and indexed, so it is instantly searchable, whether or not everything about it came out perfectly.

5

## Staff get an answer they can trust

Whenever someone asks a question, the assistant answers using only what is actually on file, names its source, and says plainly when it does not know rather than guessing.

## Five Different Ways Staff Actually Use It

---

Every one of these runs through the same assistant, against the same knowledge base, always grounded in what is actually on file.

### Query

Ask a plain question and get back an answer with the exact document it came from named alongside it.

---

### Summarize

Turn an entire document into a short, plain language summary, without anyone having to read the whole thing.

---

### Compare

Put two documents side by side and see exactly what is genuinely similar and what actually differs between them.

---

### Extract

Pull out specific details from a document, a name, a date, a policy number, without reading through the rest of it.

---

### Draft

Generate new content grounded only in what is really on file, never inventing a detail that is not there.