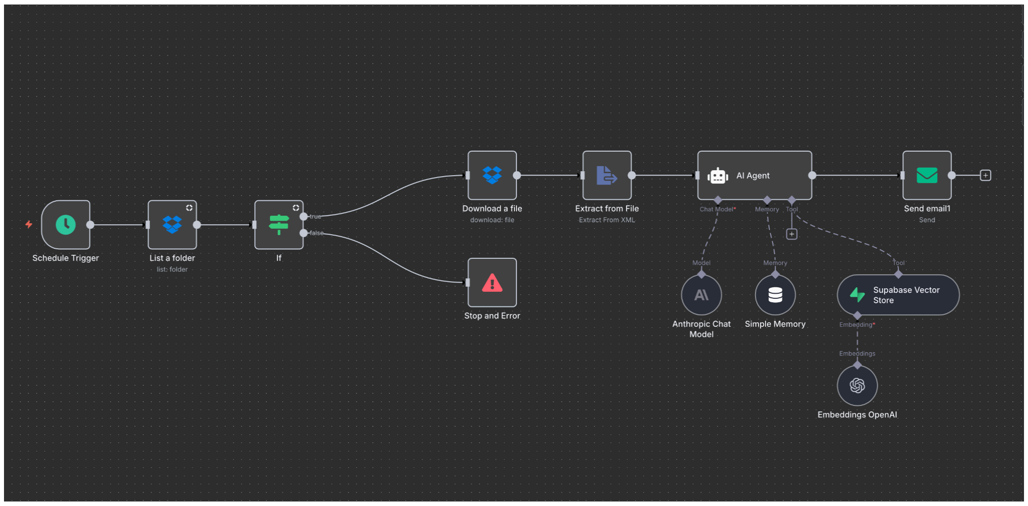
n8n Tutorial: When New RFPs Arrive, Have AI Create Tailored SOW Proposals for You



What You'll Get

An n8n workflow that learns from your historical bids and automatically generates customized SOWs when new RFPs arrive—saving 4-8 hours per proposal.

This system stores your past successful proposals in a searchable database, analyzes new project requirements, finds your most relevant past work, and generates tailored SOWs that match your company's proven approach and pricing strategies. The AI learns from your wins, not generic content.

What You'll Need

Time: 6–12 hours to set up (this is advanced)
Budget: \$3-\$30/mo
Experience: None required (but some workflow experience helpful, as this involves vector databases, embeddings, multi-step workflows)
Key tools: n8n account (Free self-hosted or \$20/mo cloud) – orchestrates workflow
Supabase account (Free* tier works) – stores docs with vector search
OpenAI API key (\$3-10/mo usage-based) – Powers SOW generation
Dropbox account (Free tier works) – stores your bids and PRDs

STEP 1 SET UP YOUR VECTOR DATABASE

Function: Creating a database that understands the meaning of your documents, not just keywords.

Time: 15-20 minutes

Actions:

1. Create a Supabase project

- Go to supabase.com and sign up
- Click "New Project"
- Name it something like "SOW-Generator"
- Wait for the project to initialize (~2 minutes)

2. Enable the vector extension

Go to the SQL Editor in Supabase and run this script:

→

SUPABASE SCRIPT
Copy/Paste This Entire Section

```
-- Enable vector extension for semantic search
CREATE EXTENSION IF NOT EXISTS vector;

-- Create documents table
CREATE TABLE documents (
  id BIGSERIAL PRIMARY KEY,
  content TEXT,
  metadata JSONB,
  embedding VECTOR(1536)
);

-- Create search function
CREATE FUNCTION match_documents(
  query_embedding VECTOR(1536),
  match_count INT DEFAULT 5
)
RETURNS TABLE(id BIGINT, content TEXT, metadata JSONB, similarity
FLOAT)
LANGUAGE SQL STABLE
AS $
  SELECT id, content, metadata,
  1 - (embedding <=> query_embedding) AS similarity
  FROM documents
  ORDER BY embedding <=> query_embedding
  LIMIT match_count;
$;
```

What this does: Creates a database that stores both your document text AND mathematical representations of their meaning (vectors), enabling semantic search—finding conceptually similar projects even if they use different words.

3. Save your credentials

- Go to Settings → API in Supabase
- Copy your **Project URL**
- Copy your **Service Role Key**
- Save these somewhere safe—you'll need them for n8n

Done when: Your SQL script runs without errors and you have your Supabase credentials saved.

STEP 2 BUILD "LOAD HISTORICAL BIDS" WORKFLOW

Function: Processing your past successful proposals and storing them in searchable format.

Time: 2-3 hours

Actions:

1. Create a new n8n workflow

- Log into n8n
- Click "Add Workflow"
- Name it "Load Historical Bids"

2. Add Manual Trigger

- Add node: "Manual Trigger"
- This lets you run the workflow whenever you're ready to process documents

Why manual? You'll only need to load historical documents once, then occasionally add new ones. No need for automatic scheduling here.

3. Add Dropbox: List Files

- Add node: "Dropbox"
- Action: "List"

- Operation: "List Files"
- Folder: `/Historical-Bids`
- Connect your Dropbox account
- Test to confirm it sees your bid documents

4. Add Loop Over Items

- Add node: "Loop Over Items"
- Batch Size: 5
- This processes documents in small groups to avoid overwhelming the system

5. Add Dropbox: Download File

- Add node: "Dropbox"
- Action: "Download"
- Path: `{{ $json.path_display }}` (pulls from previous step)
- This downloads each document as binary data

6. Add Extract From File

- Add node: "Extract From File"
- This automatically handles PDFs, Word docs, and Excel files
- Extracts clean text regardless of format

7. Add Code Node: Create Chunks

Add node: "Code"

Paste this JavaScript:

```
// Split document into chunks
const content = $input.first().json.data;
const fileName = $('Dropbox').first().json.name;
const chunkSize = 1000;
const chunks = [];

let start = 0;
while (start < content.length) {
  chunks.push({
    content: content.substring(start, start + chunkSize),
    metadata: {
      fileName: fileName,
      chunkIndex: chunks.length
    }
  });
  start += chunkSize;
}
```

```
return chunks.map(chunk => ({ json: chunk }));
```

Why chunk? Large documents need to be broken into smaller pieces so the AI can process them and create more precise matches when searching later.

8. Add Embeddings OpenAI

- Add node: "Embeddings OpenAI"
- Model: text-embedding-3-large
- Connect your OpenAI API key
- This converts text into vectors that capture semantic meaning

9. Add Supabase Vector Store

- Add node: "Supabase Vector Store"
- Mode: "Insert Documents"
- Table Name: `documents`
- Supabase Service Role Key: [paste your key from Step 1]
- Supabase Project URL: [paste your URL from Step 1]
- This stores both text and embeddings together

10. (Optional) Add Email/Slack notification

- Add node: "Send Email" or "Slack"
- Confirms when processing is complete
- Tells you how many documents were stored

11. Test the workflow

- Place 3-5 sample bid documents in your Dropbox `/Historical-Bids` folder
- Click "Execute Workflow"
- Wait for completion (~1-2 minutes depending on file sizes)
- Check Supabase to confirm documents were stored:
  - Go to your Supabase project
  - Click "Table Editor"
  - Select "documents" table
  - Confirm rows were added

Done when: Your historical bid documents are successfully stored in Supabase with embeddings.

STEP 3 BUILD THE "GENERATE SOW" WORKFLOW

Function: Automatically creating customized SOWs when new RFPs arrive.

Time: 3-4 hours

Actions:

1. Create a new n8n workflow

- Name it "Generate SOW"

2. Add Dropbox Trigger: File Created

- Add node: "Dropbox Trigger"
- Event: "File created"
- Folder: `/New-PRDs`
- This monitors continuously and triggers when a new PRD is uploaded

3. Add Dropbox: Download File

- Downloads the new PRD that triggered the workflow

4. Add Extract From File

- Extracts text from the PRD (handles PDF, Word, etc.)

5. Add OpenAI Chat Model: Analyze PRD

- Add node: "OpenAI Chat Model"
- Model: GPT-4o
- **System Message:** "Analyze this PRD and extract: project type, scope, key requirements, and budget indicators."
- **User Message:** `{{ $json.data }}` (the extracted PRD text)

Why analyze first? This creates a structured summary optimized for searching your historical bids, rather than searching with the raw document.

6. Add Supabase Vector Store: Search

- Add node: "Supabase Vector Store"
- Mode: "Get Many"
- Prompt: `Find similar projects to: {{ $json.choices[0].message.content }}`

- Limit: 5
- Supabase credentials: [same as before]

What this does: Searches your historical bids using semantic similarity and returns the 5 most relevant past projects.

7. Add OpenAI Chat Model: Generate SOW

Add node: "OpenAI Chat Model"

System Message:

```
You are an expert proposal writer. Based on the PRD and similar past
projects, create a comprehensive SOW including:
- Executive Summary
- Scope of Work
- Timeline & Milestones
- Pricing Breakdown
- Terms & Conditions
```

User Message:

```
PRD: {{ $('Extract From File').item.json.data }}

Similar past projects:
{{ $('Supabase Vector Store').all().map(doc =>
doc.json.content).join('\n---\n') }}

Generate a detailed SOW that matches our company's proven approach
and pricing patterns.
```

This is where the magic happens: The AI receives both the client's requirements (PRD) AND your company's proven approaches (similar past bids), allowing it to generate proposals that sound like your team wrote them.

8. Add Convert to File

- Add node: "Convert to File"
- Format: HTML or Markdown
- Filename: `SOW_{{ $now.format('YYYY-MM-DD') }}.html`
- Transforms the AI-generated text into a proper document

9. Add Gmail/SMTP: Send Email

- Add node: "Gmail" or "Send Email"

- To: Your email (or your team's)
- Subject: `New SOW Generated - {{ $now.format('YYYY-MM-DD') }}`
- Attachments: The generated SOW file
- Body:

```
Your SOW has been generated based on {{ $('Supabase Vector
Store').all().length }} similar historical projects.

Please review for accuracy before sending to the client.
```

10. Test the workflow

- Upload a sample PRD to your Dropbox `/New-PRDs` folder
- Wait 30-60 seconds for the workflow to run
- Check your email for the generated SOW
- Review the SOW for quality and accuracy

Done when: New PRDs automatically trigger SOW generation and you receive completed proposals via email.

STEP 4 REFINE YOUR HISTORICAL DATA

Function: Improving the quality of AI-generated SOWs by enriching your historical bid data.

Time: Ongoing (30-60 minutes per refinement cycle)

Actions: **1. Add rich metadata to your bids**

When loading documents in the future, enhance the metadata in your Code Node (Step 2, Action 7):

```
metadata: {
  fileName: fileName,
  chunkIndex: chunks.length,
  clientIndustry: "healthcare",  // Add manually
  projectSize: "large",          // Add manually
  techStack: "React, Node.js",   // Add manually
  outcome: "successful"          // Add manually
}
```

2. Review generated SOWs for patterns

- After generating 5-10 SOWs, review them
- Note which historical bids produce the best results
- Consider removing low-quality historical bids from your database

3. Adjust similarity threshold

In your search step (Step 3, Action 6), you can adjust how selective the matching is:

- Higher limit (e.g., 10): More examples, but potentially less relevant
- Lower limit (e.g., 3): Fewer examples, but more focused

4. Fine-tune the generation prompt

In the SOW generation step (Step 3, Action 7), add more specific guidance:

```
Additional guidelines:
- Our standard project phases are: Discovery, Design, Development, Testing, Launch
- Always include a 10% contingency buffer
- Payment terms are typically 50% upfront, 50% on completion
- Use our company's tone: professional but approachable
```

Done when: 70%+ of generated SOWs require only minor edits before sending to clients.

ROCKETJONES

Quick Start Checklist

- ☐ Created Supabase project and ran SQL setup script
- ☐ Saved Supabase credentials (Project URL, Service Role Key)
- ☐ Created Dropbox folders: `/Historical-Bids` and `/New-PRDs`
- ☐ Uploaded 5-10 historical bid documents to `/Historical-Bids`
- ☐ Built "Load Historical Bids" workflow in n8n (10 nodes)
- ☐ Tested loading workflow and confirmed documents in Supabase
- ☐ Built "Generate SOW" workflow in n8n (9 nodes)
- ☐ Tested with sample PRD and received generated SOW
- ☐ Reviewed SOW quality and refined prompts
- ☐ Set up email notification for new SOWs

Common Issues and Quick Fixes

"Supabase connection failing" → Double-check your Service Role Key (not the anon key). Verify Project URL includes `https://`

"Embeddings step is slow" → This is normal. Processing 10 documents with embeddings takes 2-3 minutes. Be patient.

"Generated SOWs are too generic" → Add more historical bids (aim for 15-20 minimum). Add rich metadata to improve matching quality.

"Search isn't finding relevant past projects" → The PRD analysis may be too vague. Enhance the analysis prompt in Step 3, Action 5 to extract more specific details.

"OpenAI costs are higher than expected" → Embeddings cost ~$0.02 per document. SOW generation costs ~$0.10-0.50 each. For 10 SOWs/month, expect ~$5-10/month.

Next-Level Enhancements

Once you have the basic system running reliably, consider:

Add Manual Approval	Version Control	Client-Specific Templates	Pricing Optimization
Pause workflow before sending, requiring team review	Store multiple versions of generated SOWs	Create separate vector stores for different client types	Track which historical pricing led to won vs. lost bids
Sends Slack message with "Approve" button	Track edits made to AI output	Match based on industry, company size, tech stack	Optimize pricing strategies based on data

---

Want help building this?

We Know This Workflow Inside and Out—Because We Built It for Ourselves

This tutorial gives you everything you need to build your own AI proposal generator. But we're not going to sugarcoat it: this is an **advanced workflow** involving vector databases, RAG architecture, and multi-step AI orchestration. Most teams need expert help to get it right.

Three ways Rocket Jones can help

- **Done-For-You Setup:** Complete system built, customized, and tested in 1-2 weeks
- **Guided Build:** We pair with your technical lead to build it together over 3-4 weeks
- **Advanced Features:** Add approval workflows, pricing optimization, CRM, etc.

Questions? Email us: info@rocketjones.com

**Supabase free tier (500MB database) sufficient for 100-200 bid documents. n8n free (self-hosted) or $20/mo cloud.*