N8N implementation

---

1. High-level architecture (mental model)

Chatbot = 4 brains working together

1. **LLM (reasoning + conversation)**
2. **RAG (facts about Small Group)**
3. **Conversation State (what we know about user)**
4. **Conversion Logic (when to push booking)**

n8n orchestrates all 4.

---

2. Knowledge you will store (RAG collections)

Create **4 small, sharp knowledge bases** (don’t overdo it).

KB-1: Company & Trust

- What is Small Group
- What problems you solve
- Founder background
- Why you’re credible
- How you work
- NDA, ownership, engagement model

KB-2: AI Automation

- What AI automation means (in plain English)

- Examples:
  - CRM automation
  - Lead follow-ups
  - WhatsApp / email / voice bots
  - Internal ops automation
  - AI agents
- What you **don’t** do (important for trust)

KB-3: Software Development

- MVPs
- SaaS
- Internal tools
- Web/mobile apps
- When custom software is needed vs automation

KB-4: Process & Next Steps

- How discovery works
- How calls work
- What happens after call
- Typical timelines
- Pricing philosophy (not numbers)

Each doc chunk should answer **one question**, not paragraphs.

---

3. Core system prompt (passed to LLM)

This is **non-negotiable**. This is what makes it feel human and focused.

```
You are the AI assistant for Small Group, a team that does BOTH:
1) AI automation (agents, workflows, ops automation)
2) Custom software development (MVPs, internal tools, SaaS)

Your goals, in priority order:
1) Clearly understand the user’s problem
2) Answer trust and capability questions honestly
3) Decide whether AI automation, custom software, or a mix is appropriate
4) Move qualified users toward booking a call
```

```
Rules:
- Never oversell
- If something is unclear, ask ONE sharp clarifying question
- If the problem is real and non-trivial, suggest a call
- Do not push booking too early
- Sound like a smart founder, not sales copy
- Use retrieved knowledge as ground truth
- If info is missing, say so transparently
```

---

4. Conversation control prompt (dynamic, injected)

This is updated on every message:

```
Known user info:
- Problem summary: {{problem_summary}}
- Industry: {{industry}}
- Company size: {{company_size}}
- Urgency level: {{urgency}}
- Budget signal: {{budget_signal}}
- Trust level: {{trust_level}}

Your task:
- Either deepen problem understanding
- Or answer a trust/capability question
- Or move toward booking a call if criteria are met
```

---

5. Qualification logic (VERY important)

Only push for a call if **2 of 3 are true**:

- Clear business problem
- Problem cannot be solved with a simple tool
- User shows intent (cost, timeline, “how”, “can you”)

This logic lives in n8n, not the LLM alone.

6. n8n workflow (node-by-node)

1. **Webhook / Chat Trigger**

- Input: user message, session_id
- Output: raw user text

---

2. **Session Memory (Redis / DB)**

- Fetch conversation state:
  - problem_summary
  - trust_level (0–3)
  - qualification_score (0–3)
  - last_intent

---

3. **Intent Classifier (LLM – cheap model)**

Classify message into:

- TRUST_QUESTION
- AI_AUTOMATION_QUERY
- SOFTWARE_QUERY
- PROBLEM_STATEMENT
- PRICING_TIMELINE
- GENERAL_CHAT

Store result.

---

4. **RAG Retriever**

Based on intent:

- TRUST → KB-1
- AI automation → KB-2
- Software → KB-3
- Process / next steps → KB-4

Use:

- Top 3–5 chunks
- Semantic similarity
- Small chunk size (300–500 tokens)

---

5. **Main LLM Response Node**

Inputs:

- System prompt
- Conversation control prompt
- Retrieved RAG context
- User message

Outputs:

- Natural response
- Implicit signal: `should_ask_question`, `should_suggest_call`

---

6. **Problem Extractor (LLM or rules)**

Extract and update:

- Problem summary (1–2 lines)
- Industry
- Company size
- Urgency (low / medium / high)

Update session memory.

---

7. **Qualification Scorer (IF + Function node)**

Increment score if:

- User describes workflow pain
- Mentions scale, team, cost, or time
- Asks “how would you”, “can you build”, “what’s the approach”

---

8. **Call Suggestion Gate**

IF:

- qualification_score ≥ 2
- trust_level ≥ 1

→ allow booking CTA

Else:

- continue conversation

---

9. **Booking CTA Generator**

Soft, founder-style CTA:

Examples:

- “This sounds like something we should look at properly — want to hop on a quick call?”
- “Hard to give a clean answer without context. A 20-min call might save you weeks.”

Attach:

- Calendly link
- Optional form (problem summary auto-filled)

---

10. **Human Handoff (optional)**

If user explicitly asks:

- “Can I talk to someone”
- “Who will handle this”

→ notify Slack / email with:

- Conversation summary
- Problem summary
- Qualification score

---

7. What NOT to do (hard advice)

- □ Don’t dump pricing in chatbot
- □ Don’t act like “AI-only” or “dev-only”
- □ Don’t ask 10 discovery questions

- Don’t force Calendly on first 3 messages

---

8. Why this works (founder logic)

- Feels like **talking to a thinking founder**
- AI automation is positioned as **leverage**, not buzzword
- Software is positioned as **when automation isn’t enough**
- Booking feels like the **natural next step**, not a funnel trick

---

Revision #1
Created 2026-02-02 17:22:06 UTC by Chandan Kumar
Updated 2026-02-02 17:23:25 UTC by Chandan Kumar