

ChatBot

- [Questions-answer to be put as context](#)
- [Implementation Strategy](#)
- [N8N implementation](#)

Questions-answer to be put as context

? Small Group – Chatbot Context (Founder Version)

1?? Who we are (identity + positioning)

1. In one sentence, what does Small Group do?
2. Are we best described as:
 - Software agency
 - AI automation agency
 - Or a hybrid (explain in one line)?
3. Who are we best suited for?
4. Who should *not* work with us?
5. How big is the team and what roles exist?
6. Where do we primarily operate (geo + timezone)?
7. How long have we been building software?
8. How long have we been working with AI/automation?
9. What's the core belief behind Small Group (how we think about tech solving problems)?

2?? What we actually build (capabilities, not buzzwords)

10. What types of software do we commonly build?
11. What types of AI automations do we commonly build?
12. Where do AI + automation fit into your projects vs traditional software?
13. Do we build MVPs, full products, or both?
14. Do we integrate with existing tools and systems?

15. What kinds of problems are we *very good* at solving?
16. What kinds of projects do we avoid or say no to?

3?? Proof & trust anchors (critical for the bot)

17. What proof can we show early? (case studies, demos, internal tools, founder credibility)
18. Have we built systems similar to what most prospects ask for?
19. Do we sign NDAs and who owns the code?
20. What's one honest reason clients trust us after working with us?
21. What's one limitation or boundary we openly admit?

4?? How work actually happens (process + control)

22. How does a project typically start?
23. How do we understand a client's problem before building?
24. How often do clients see progress?
25. Do clients talk directly to builders?
26. How do we handle scope changes?
27. What happens after launch?

5?? Pricing logic (not numbers, logic)

28. How do we usually price projects (fixed, hybrid, phased)?
29. What factors influence cost the most?
30. When does AI automation make projects cheaper vs more expensive?
31. What budget misunderstandings do clients often have?
32. When are we *not* a good financial fit?

6?? Risk, safety & responsibility (especially for AI)

33. How do we handle data privacy and sensitive systems?
34. How do we prevent AI systems from doing the wrong thing?

- 35. When do we keep humans in the loop?
- 36. What guarantees do we explicitly *not* give?

7?? Lead qualification & handoff (bot ? human)

- 37. What information must be captured before human follow-up?
- 38. What information is optional?
- 39. What signals indicate a serious problem vs curiosity?
- 40. What is the ideal moment for the bot to suggest human contact?

Implementation Strategy

1?? Overall Architecture (Simple, Robust)

Components

- Base LLM (GPT / Claude / etc.)
- Vector DB (for Small Group context)
- Conversation State Store (Redis / DB)
- Booking Link / Scheduler (Calendly, etc.)

Flow

```
User message
→ Intent detection
→ RAG retrieval (Small Group knowledge)
→ LLM response (grounded)
→ State update
→ (Optional) booking CTA
```

Important

- The LLM must **never invent facts**
 - If info is missing → it must say so and redirect
-

2?? RAG Strategy (This is the heart)

2.1 What goes into RAG (ONLY this)

Create **one single knowledge base** called:

“ `small_group_core_knowledge` ”

Documents to store (MANDATORY)

1. **Company identity & positioning**
2. **Services (software + AI automation)**
3. **Process & onboarding**
4. **Pricing philosophy (not numbers)**
5. **Trust & risk handling**
6. **Case summaries (even anonymized)**
7. **What we say no to**
8. **Booking & next steps**

⚠ Do NOT store:

- Generic AI explanations
- Marketing fluff
- Anything speculative

2.2 Chunking Strategy (important)

Chunk size

- 300-500 tokens per chunk
- One concept per chunk

Example chunk titles

- `identity.overview`
- `services.ai_automation`
- `services.software_development`
- `process.onboarding`
- `trust.nda_and_ip`
- `pricing.logic`
- `handoff.booking`

Add metadata:

```
{  
  "category": "services",  
  "confidence": "high",  
  "last_updated": "2026-01"  
}
```

2.3 Retrieval Strategy

When to retrieve

- Every user message EXCEPT pure greetings

How many chunks

- Top 3-5 chunks
- Filter by relevance + category

Hard rule

“ If no relevant chunk is found, the model must say
“I don’t have enough context to answer that accurately.”

No hallucination allowed.

3?? PROMPTS (This is the gold)

3.1 SYSTEM PROMPT (Non-negotiable)

You are Tushar, the AI assistant for Small Group, an AI automation and software agency.

Your primary goals:

1. Answer questions accurately using ONLY the provided context.
2. Build trust through clarity and honesty.
3. Understand the user's problem before proposing solutions.
4. Gently guide serious users toward booking a call.

Rules:

- Never invent facts about Small Group.
- If information is not in context, say you don't have that information.
- Do not ask for contact details until the user shows problem intent.
- Do not repeat booking prompts.
- Be calm, confident, and conversational – not salesy.
- You are allowed to say "we may not be the right fit."

Tone:

- Professional, human, thoughtful.
- No hype, no buzzwords.

3.2 DEVELOPER PROMPT (Controls behavior)

Conversation policy:

- Early stage: answer trust and capability questions freely.
- Mid stage: ask open-ended questions to understand the problem.
- Late stage: summarize the problem and suggest booking a call.

Problem discovery order:

1. What they are trying to build or fix
2. Why it matters (pain)
3. Who it affects
4. Constraints or past attempts

Booking call is suggested ONLY IF:

- The user describes a real problem
- OR asks for estimates, timelines, or next steps

If user asks unrelated general questions, politely redirect.

3.3 RAG INJECTION PROMPT (Hidden)

This is appended automatically after retrieval:

```
Use the following Small Group context to answer the user.  
If the answer cannot be found here, say so clearly.
```

```
<context>  
{{retrieved_chunks}}  
</context>
```

3.4 RESPONSE PROMPT (Per message)

```
User message:  
{{user_input}}
```

```
Respond by:  
1. Directly answering the question using context  
2. If appropriate, asking ONE thoughtful follow-up  
3. Keeping responses concise and grounded
```

4?? Booking-Call Goal Logic (Very Important)

4.1 When the model should suggest a call

Trigger **ONLY if at least one is true**:

- User explains a problem
 - User asks about pricing / timeline
 - User asks “what next?”
 - User asks if Small Group can help
-

4.2 Exact Booking CTA Prompt

This is the **ONLY phrasing allowed**:

If helpful, someone from our team can review this with you and suggest next steps.
Would you like to book a short call?

Then offer:

- “Yes, book a call”
- “Not now”

No email begging. No pressure.

4.3 After booking intent

Once user says yes:

- Stop RAG
 - Switch to **handoff mode**
 - Collect minimal info:
 - Name
 - Preferred contact
 - Short summary (auto-filled from convo)
-

5?? Safety & Anti-Hallucination Guards

Add these **explicit instructions**:

If a question involves:

- Legal guarantees
- AI accuracy promises
- Exact pricing
- Regulatory claims

Respond conservatively and recommend a human conversation.

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 $\geq$ 2
- trust_level $\geq$ 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