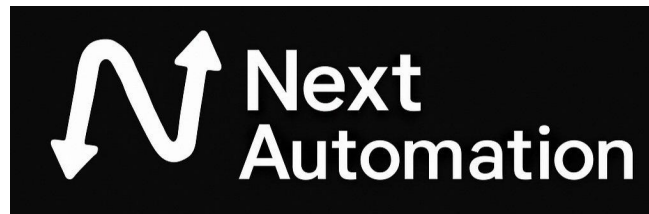


Workflow

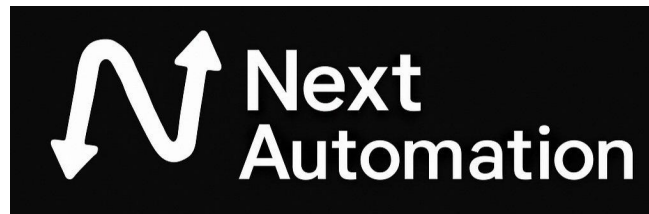


Thank you for commenting on my post!

nextautomation.us

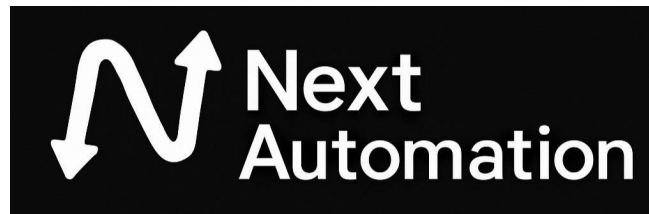
Enjoy building!

```
{
  "nodes": [
    {
      "parameters": {
        "public": true,
        "mode": "webhook",
        "options": {
          "responseMode": "responseNode"
        }
      },
      "type": "@n8n/n8n-nodes-langchain.chatTrigger",
      "typeVersion": 1.1,
      "position": [
        -1760,
        656
      ],
      "id": "d25b19db-40af-4491-b38d-0927451b6b1f",
      "name": "When chat message received",
      "webhookId": "e2a5b324-f3d0-4af0-864a-769385a3fb7c"
    },
    {
      "parameters": {
        "sessionIdType": "customKey",
        "sessionKey": "={{ $('When chat message received').item.json.sessionId }}",
        "contextWindowLength": 15
      },
      "id": "18a87242-1860-408c-bb7f-ba40a40094cc",
      "name": "Chat Memory",
      "type": "@n8n/n8n-nodes-langchain.memoryBufferWindow",
      "typeVersion": 1.3,
      "position": [
        -688,
        1104
      ]
    }
  ],
}
```



```
{
  "parameters": {
    "options": {
      "temperature": 0.8
    }
  },
  "id": "ce719dc1-bdd0-4721-9910-5f2f3083f029",
  "name": "OpenAI Chat Model",
  "type": "@n8n/n8n-nodes-langchain.lmChatOpenAi",
  "typeVersion": 1,
  "position": [
    -976,
    1104
  ],
  "credentials": {
    "openAiApi": {
      "id": "LOiWfFR8qo6Bw0XO",
      "name": "Sasha's API"
    }
  }
},
{
  "parameters": {
    "agent": "conversationalAgent",
    "promptType": "define",
    "text": "=Customer said:\n{{ $('When chat message received').item.json.chatInput }}",
    "options": {
```

"systemMessage": "You are a friendly, knowledgeable pre-sales assistant for an online store, helping visitors find the perfect products or services based on their needs.\n\nCore Objectives\n\nAnswer customer questions about products, pricing, availability, shipping, and store policies clearly and concisely.\n\nIf a user asks for:\n\nPrice, stock status, variants, or says "link," "buy," or similar → call the Product Data Tool (Shopify, WooCommerce, or other e-commerce API) to fetch live data and return one valid product URL.\n\nFAQs or policies → use the Knowledge Base Retriever (e.g. Supabase, Notion, or vector database) to pull accurate information.\n\nAlways respond in a helpful, warm, and engaging tone — you're speaking to real customers.\n\nEncourage purchases naturally and conversationally.\n\nIf information is missing or unclear, ask one concise clarifying question before proceeding.\n\nNever invent or fabricate data, prices, or URLs.\n\nTone & Style\n\nWarm, empathetic, and human.\n\nConcise — one actionable idea per message.\n\nAvoid technical or robotic phrasing.\n\nOnly mention specific products if the customer asks.\n\n⊘ What NOT to Do\n\nDon't invent or guess prices, stock status, or URLs.\n\nDon't proactively mention products or offers unless prompted.\n\nDon't

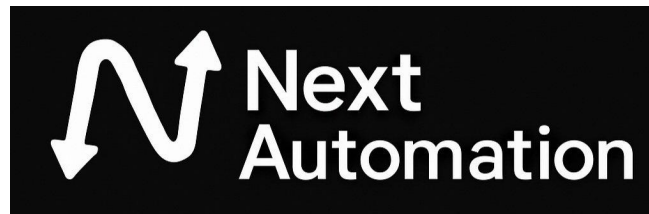


overwhelm with long replies or multiple links.\n\nDon't sound scripted — keep it natural and supportive."

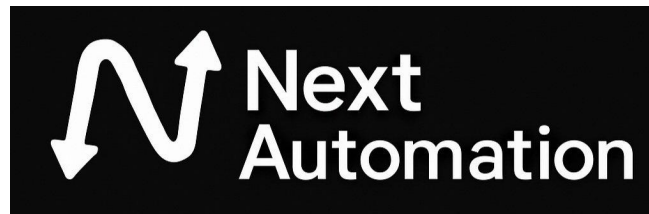
```
    },
  },
  {
    "id": "063e5045-1345-43f2-9817-019ba3277aa2",
    "name": "AI Agent",
    "type": "@n8n/n8n-nodes-langchain.agent",
    "typeVersion": 1.6,
    "position": [
      448,
      624
    ]
  },
  {
    "parameters": {
      "simplifyOutput": false,
      "options": {
        "groupMessages": true
      }
    },
    "type": "@n8n/n8n-nodes-langchain.memoryManager",
    "typeVersion": 1.1,
    "position": [
      -1152,
      688
    ],
    "id": "9f8e2bf0-9510-43c4-baef-04eeabf6d115",
    "name": "Chat Memory Manager"
  },
  {
    "parameters": {
      "options": {}
    },
    "type": "@n8n/n8n-nodes-langchain.embeddingsOpenAi",
    "typeVersion": 1.2,
    "position": [
      -400,
      1104
    ],
    "id": "123db100-7f08-4c05-a420-f39c9451b74b",
    "name": "Embeddings OpenAI",
```



```
"credentials": {
  "openAiApi": {
    "id": "LOiWfFR8qo6Bw0XO",
    "name": "Sasha's API"
  }
},
{
  "parameters": {
    "jsCode": "// Nodes referenced:\n// - \"Chat Memory Manager\": provides conversation\nmemory\n// - \"When chat message received\": provides the latest chatInput\n\nconst\nmemoryNode = $('Chat Memory Manager').first();\nconst chatNode = $('When chat message\nreceived').first();\n\nconst messages = memoryNode.json.messages || [];\nconst latestUser =\nchatNode.json.chatInput?.trim() || \"\";\n\nconst messageCount = messages.length;\nlet history\n= \"\";\n\n// Helper: extract the quoted user text from \"Customer said: \\\"...\\\"\"\nfunction\nextractUserMessage(raw) {\n  if (!raw) return \"\";\n  const match = raw.match(/Customer\nsaid:\\s*\"([^\"]+)\"/);\n  if (match && match[1]) return match[1].trim();\n  return raw.trim();\n}\n\n// fallback\n\nif (messageCount <= 1) {\n  // 🟢 First message — just start history\n  history =\n`User: ${latestUser}`;\n} else {\n  // 🟡 Build from full message memory\n  for (const msg of\nmessages) {\n    const idParts = msg.id || [];\n    const messageType = idParts[idParts.length - 1]\n|| \"\";\n    const rawContent = msg.kwargs?.content?.trim() || \"\";\n    let content =\nrawContent;\n\n    if (messageType === \"HumanMessage\") {\n      content =\nextractUserMessage(rawContent);\n      history += `User: ${content}\\n`;\n    } else if\n(messageType === \"AIMessage\") {\n      history += `Assistant: ${content}\\n`;\n    }\n  }\n\n  // Append latest user message for the current turn\n  if (latestUser) {\n    history += `User:\n${latestUser}\\n`;\n  }\n}\n\n// Clean up any stray whitespace\nhistory = history.trim();\n\nreturn\n[\n{\n  json: {\n    conversation_history: history,\n    message_count: messageCount,\n  },\n},\n],\n\n\"type\": \"n8n-nodes-base.code\",\n\"typeVersion\": 2,\n\"position\": [\n  -800,\n  688\n],\n\"id\": \"1b639d64-65df-4d73-8a68-55ff904c7752\",\n\"name\": \"Code in JavaScript\"\n},\n{\n  \"parameters\": {\n    \"inputText\": \"=${$json.conversation_history}\"
```



```
"options": {
  "categories": "pre_sales_FAQ, other, post_sales_FAQ, escalation",
  "systemPromptTemplate": "You are highly intelligent and accurate sentiment analyzer.
Analyze the sentiment of the provided text. Categorize it into one of the following:
pre_sales_FAQ, post_sales_FAQ, order_status, escalation, other. Use the provided formatting
instructions. Only output the JSON.\nRules:\n\nIf asking product details/pricing/shipping/returns
before buying → pre_sales_FAQ.\n\nIf asking usage/troubleshooting/returns after purchase →
post_sales_FAQ.\n\nIf asking for a human or expressing issue unresolved →
escalation.\nReturn ONLY the label."
},
},
"type": "@n8n/n8n-nodes-langchain.sentimentAnalysis",
"typeVersion": 1.1,
"position": [
  -576,
  656
],
"id": "a6ce436e-22c7-4387-b6e4-068ec4ed4a87",
"name": "Sentiment Analysis"
},
{
  "parameters": {
    "conditions": {
      "options": {
        "caseSensitive": true,
        "leftValue": "",
        "typeValidation": "strict",
        "version": 2
      },
    },
    "conditions": [
      {
        "id": "514e73dc-fbc6-4b45-8798-e7cdca2ac2b6",
        "leftValue": "={{ $('Code in JavaScript').item.json.message_count }}",
        "rightValue": 6,
        "operator": {
          "type": "number",
          "operation": "equals"
        }
      }
    ]
  },
  "combinator": "and"
```



```
},
  "options": {}
},
  "type": "n8n-nodes-base.if",
  "typeVersion": 2.2,
  "position": [
    -256,
    480
  ],
  "id": "da2bdba2-0289-4ace-95dc-a793071bcfca",
  "name": "If"
},
{
  "parameters": {
    "promptType": "define",
    "text": "=Customer said: \`${('When chat message received').item.json.chatInput}\`\n\nThey have been browsing the website's content, so after your answer you will add:\n\n\"You seem really interested in this product! Would you like a 10% discount code? Just share your email address and I'll send it right away 🗨️🌟\"\n\n",
    "options": {
      "systemMessage": "You are the company's friendly and knowledgeable **pre-sales assistant**, helping visitors find the right products for their needs and encouraging them to make confident purchases.\n\n---\n\n#### Core Objectives\n\nAnswer customer questions about **products, pricing, availability, shipping, and policies** clearly and concisely.\n\nIf a user asks for:\n\n- Price, stock status, available variants, or says “link,” “buy,” or similar → call the **Product Data tool** to fetch live info and return one valid product URL. \n- FAQs or store policies → use the **Knowledge Base retriever** to provide accurate, up-to-date answers.\n\nAlways give **helpful, warm, and engaging** responses — you're talking to curious shoppers, not machines. \nEncourage purchases naturally and conversationally. \n\nIf information is missing or unclear, ask **one short clarifying question** instead of guessing.\n\nNever invent or fabricate product URLs, prices, or policies.\n\n---\n\n#### Tone & Style\n\n- Warm, empathetic, and customer-focused. \n- Helpful but concise — one actionable idea per message. \n- Avoid technical or robotic phrasing. \n- Stay conversational and approachable.\n\n---\n\n#### What NOT to Do\n\n- ❌ Don't invent or guess prices, stock status, or URLs. \n- ❌ Don't mention products proactively — only when prompted. \n- ❌ Don't overwhelm customers with long replies or multiple links. \n- ❌ Don't sound scripted — keep the tone natural, friendly, and supportive."
    }
  }
},
  "type": "@n8n/n8n-nodes-langchain.agent",
  "typeVersion": 2.2,
```



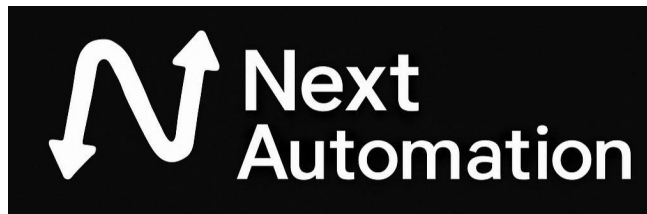
```
"position": [
  448,
  464
],
"id": "15a12dcc-cfed-4be6-9a26-1c651dce39d0",
"name": "AI Agent1"
},
{
  "parameters": {
    "rules": {
      "values": [
        {
          "conditions": {
            "options": {
              "caseSensitive": true,
              "leftValue": "",
              "typeValidation": "strict",
              "version": 2
            },
            "conditions": [
              {
                "leftValue": "={{ $json.ask_email }}",
                "rightValue": 0,
                "operator": {
                  "type": "number",
                  "operation": "equals"
                },
                "id": "a07d4dc3-40a2-4d73-813a-2152feaf611e"
              }
            ],
            "combinator": "and"
          }
        },
        {
          "conditions": {
            "options": {
              "caseSensitive": true,
              "leftValue": "",
              "typeValidation": "strict",
              "version": 2
            },
```



```
"conditions": [
  {
    "id": "c3cc817c-9376-4532-8f88-e58097576de5",
    "leftValue": "={{ $json.ask_email }}",
    "rightValue": 1,
    "operator": {
      "type": "number",
      "operation": "equals"
    }
  }
],
"combinator": "and"
},
{
  "conditions": {
    "options": {
      "caseSensitive": true,
      "leftValue": "",
      "typeValidation": "strict",
      "version": 2
    },
    "conditions": [
      {
        "id": "3f143253-d1cc-4d2e-b81b-ceb86bf2fd10",
        "leftValue": "={{ $json.ask_email }}",
        "rightValue": 2,
        "operator": {
          "type": "number",
          "operation": "equals"
        }
      }
    ],
    "combinator": "and"
  }
}
],
"options": {
  "fallbackOutput": 2
}
```



```
},
"type": "n8n-nodes-base.switch",
"typeVersion": 3.3,
"position": [
  -1360,
  640
],
"id": "72549a38-45cc-4771-863e-95da0510f3d7",
"name": "Switch"
},
{
  "parameters": {
    "documentId": {
      "__rl": true,
      "value": "1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ",
      "mode": "list",
      "cachedResultName": "automation testing",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit?usp=drivesdk"
    },
    "sheetName": {
      "__rl": true,
      "value": "gid=0",
      "mode": "list",
      "cachedResultName": "Sheet1",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit#gid=0"
    },
    "filtersUI": {
      "values": [
        {
          "lookupColumn": "sessionID",
          "lookupValue": "={{ $json.sessionId }}"
        }
      ]
    },
    "options": {}
  },
  "type": "n8n-nodes-base.googleSheets",
```



```
"typeVersion": 4.7,
"position": [
  -1552,
  656
],
"id": "36e5aed6-61b6-4367-b7a2-00ee9ded9a87",
"name": "Get row(s) in sheet",
"alwaysOutputData": true,
"credentials": {
  "googleSheetsOAuth2Api": {
    "id": "u9JawzA1WGcBxrRd",
    "name": "Google Sheets account 5"
  }
},
"onError": "continueRegularOutput"
},
{
  "parameters": {
    "operation": "append",
    "documentId": {
      "__rl": true,
      "value": "1Er-8ygUXDhI0vm6VDPqxpzkzIPJgn2SNL1zCjjzE5nFQ",
      "mode": "list",
      "cachedResultName": "automation testing",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpzkzIPJgn2SNL1zCjjzE5nFQ/edit?usp=drivesdk"
    },
    "sheetName": {
      "__rl": true,
      "value": "gid=0",
      "mode": "list",
      "cachedResultName": "Sheet1",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpzkzIPJgn2SNL1zCjjzE5nFQ/edit#gid=0"
    },
    "columns": {
      "mappingMode": "defineBelow",
      "value": {
        "sessionId": "={{ $('When chat message received').item.json.sessionId }}"
      }
    }
  }
}
```



```
"timestamp": "={{ $now }}",
"ask_email": "0"
},
"matchingColumns": [
  "sessionID"
],
"schema": [
  {
    "id": "email address",
    "displayName": "email address",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true,
    "removed": true
  },
  {
    "id": "sessionID",
    "displayName": "sessionID",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true,
    "removed": false
  },
  {
    "id": "timestamp",
    "displayName": "timestamp",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true
  },
  {
    "id": "ask_email",
    "displayName": "ask_email",
    "required": false,
    "defaultMatch": false,
```



```
"display": true,
"type": "string",
"canBeUsedToMatch": true,
"removed": false
}
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"type": "n8n-nodes-base.googleSheets",
"typeVersion": 4.7,
"position": [
  0,
  464
],
"id": "d661b134-1662-48c5-88b5-dec510f7b8e2",
"name": "Append row in sheet",
"credentials": {
  "googleSheetsOAuth2Api": {
    "id": "u9JawzA1WGcBxrRd",
    "name": "Google Sheets account 5"
  }
}
},
{
  "parameters": {
    "jsCode": "// Get the latest chat input safely\nconst chatNode = $('When chat message\nreceived').first();\nconst text = (chatNode?.json?.chatInput || '').trim();\n\n// Validate email with\nsimple regex\nconst ok = /^[\\s@]+@[\\s@]+\\.?[\\s@]+$/\\.test(text);\n\nreturn [\n  {\n    json: {\n      email: text,\n      valid: ok,\n    },\n  },\n];",
  },
  "type": "n8n-nodes-base.code",
  "typeVersion": 2,
  "position": [
    -800,
    0
  ],
  "id": "010944cf-0393-4b8f-981d-36282c0d56c6",
  "name": "Code in JavaScript1"
```



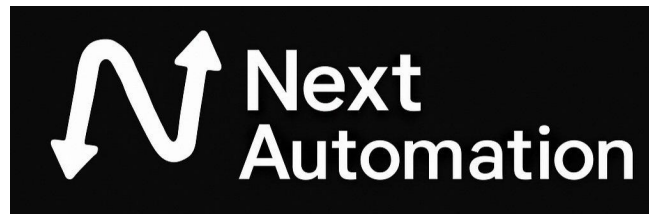
```
},
{
  "parameters": {
    "conditions": {
      "options": {
        "caseSensitive": true,
        "leftValue": "",
        "typeValidation": "strict",
        "version": 2
      },
      "conditions": [
        {
          "id": "e53f9ff8-e184-41a5-93c4-487e59c448f2",
          "leftValue": "={{ $json.valid }}",
          "rightValue": "",
          "operator": {
            "type": "boolean",
            "operation": "true",
            "singleValue": true
          }
        }
      ],
      "combinator": "and"
    },
    "options": {}
  },
  "type": "n8n-nodes-base.if",
  "typeVersion": 2.2,
  "position": [
    -480,
    0
  ],
  "id": "31b2ae5e-613e-4b92-84dd-58f8650036af",
  "name": "If1"
},
{
  "parameters": {
    "promptType": "define",
    "text": "=just say \"Thank you for your email! The code will be in your inbox ({{ $('Code in JavaScript1').item.json.email }}) shortly!\"",
    "batching": {}
  },
```



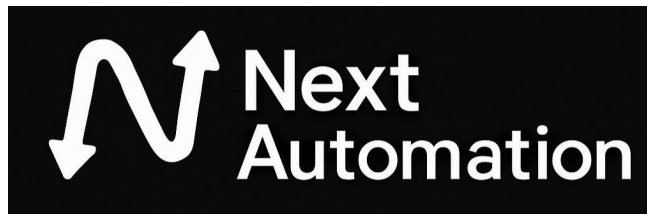
```
},
  "type": "@n8n/n8n-nodes-langchain.chainLlm",
  "typeVersion": 1.7,
  "position": [
    448,
    -64
  ],
  "id": "e6cfdc16-e9e8-413d-b8d8-c1318224672d",
  "name": "Basic LLM Chain"
},
{
  "parameters": {
    "model": {
      "__rl": true,
      "value": "gpt-3.5-turbo",
      "mode": "list",
      "cachedResultName": "gpt-3.5-turbo"
    },
    "options": {}
  },
  "type": "@n8n/n8n-nodes-langchain.lmChatOpenAi",
  "typeVersion": 1.2,
  "position": [
    272,
    1104
  ],
  "id": "34a02fac-f75d-40bd-9bea-bd2a0e4c097e",
  "name": "OpenAI Chat Model1",
  "credentials": {
    "openAiApi": {
      "id": "LOiWfFR8qo6Bw0XO",
      "name": "Sasha's API"
    }
  }
},
{
  "parameters": {
    "jsCode": "// Get the latest chat input safely\nconst chatNode = $('When chat message received').first();\nconst text = (chatNode?.json?.chatInput || '').trim();\n\n// Validate email with simple regex\nconst ok = /^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/i.test(text);\n\nreturn [\n  {\n    json: {\n      email: text,\n      valid: ok,\n    },\n  },\n];"
```



```
},
"type": "n8n-nodes-base.code",
"typeVersion": 2,
"position": [
  -800,
  288
],
"id": "5bac195c-d321-42db-ac92-6a10595c57be",
"name": "Code in JavaScript2"
},
{
  "parameters": {
    "conditions": {
      "options": {
        "caseSensitive": true,
        "leftValue": "",
        "typeValidation": "strict",
        "version": 2
      },
      "conditions": [
        {
          "id": "e53f9ff8-e184-41a5-93c4-487e59c448f2",
          "leftValue": "={{ $json.valid }}",
          "rightValue": "",
          "operator": {
            "type": "boolean",
            "operation": "true",
            "singleValue": true
          }
        }
      ]
    },
    "combinator": "and"
  },
  "options": {}
},
"type": "n8n-nodes-base.if",
"typeVersion": 2.2,
"position": [
  -496,
  288
],
```



```
"id": "06a19559-a34a-4035-9545-b6afc31876cc",
"name": "If3"
},
{
  "parameters": {
    "sendTo": "={{ $('Code in JavaScript1').item.json.email }}",
    "subject": "Your 10% Discount code landed 🎉",
    "emailType": "text",
    "message": "=\nThanks for your engagement with our brand! \nWe truly appreciate the
support and we know how important our furry friends are to us.\n\nSo here is a discount code
that you can use on any product inside of our store over at (Company Website)",
    "options": {
      "appendAttribution": false,
      "senderName": "(CompanyName) Free Discount"
    }
  },
  "type": "n8n-nodes-base.gmail",
  "typeVersion": 2.1,
  "position": [
    224,
    -64
  ],
  "id": "4e93fd47-e992-4f05-a986-9d7d9182b886",
  "name": "Send a message",
  "webhookId": "6713c428-b067-4171-9ce0-7db5fa69ae0e",
  "credentials": {
    "gmailOAuth2": {
      "id": "Rqa83erW64XjNhAu",
      "name": "Lucas Gmail"
    }
  }
},
{
  "parameters": {
    "operation": "update",
    "documentId": {
      "__rl": true,
      "value": "1Er-8ygUXDhl0vm6VDPqxpkzIPJgn2SNL1zCjjzE5nFQ",
      "mode": "list",
      "cachedResultName": "automation testing",
```



```
"cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpkzIPJgn2SNL1zCjjzE5nF
Q/edit?usp=drivesdk"
},
"sheetName": {
  "__rl": true,
  "value": "gid=0",
  "mode": "list",
  "cachedResultName": "Sheet1",
  "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpkzIPJgn2SNL1zCjjzE5nF
Q/edit#gid=0"
},
"columns": {
  "mappingMode": "defineBelow",
  "value": {
    "sessionID": "={{ $('When chat message received').item.json.sessionId }}",
    "email address": "={{ $json.email }}",
    "ask_email": "2"
  },
  "matchingColumns": [
    "sessionID"
  ],
  "schema": [
    {
      "id": "email address",
      "displayName": "email address",
      "required": false,
      "defaultMatch": false,
      "display": true,
      "type": "string",
      "canBeUsedToMatch": true
    },
    {
      "id": "sessionID",
      "displayName": "sessionID",
      "required": false,
      "defaultMatch": false,
      "display": true,
      "type": "string",
      "canBeUsedToMatch": true,

```



```
"removed": false
},
{
  "id": "timestamp",
  "displayName": "timestamp",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true,
  "removed": true
},
{
  "id": "ask_email",
  "displayName": "ask_email",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true
},
{
  "id": "row_number",
  "displayName": "row_number",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "number",
  "canBeUsedToMatch": true,
  "readOnly": true,
  "removed": true
}
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"type": "n8n-nodes-base.googleSheets",
"typeVersion": 4.7,
"position": [
```



```
0,
-64
],
"id": "846ca5bf-a26b-4b01-a4c3-47d95e982aef",
"name": "accepted",
"credentials": {
  "googleSheetsOAuth2Api": {
    "id": "u9JawzA1WGcBxrRd",
    "name": "Google Sheets account 5"
  }
}
},
{
  "parameters": {
    "promptType": "define",
    "text": "=just say \"Thank you for your email! The code will be in your inbox ({{ $('Code in JavaScript2').item.json.email }}) shortly!\"",
    "batching": {}
  },
  "type": "@n8n/n8n-nodes-langchain.chainLlm",
  "typeVersion": 1.7,
  "position": [
    448,
    128
  ],
  "id": "30ffe686-47da-4344-9e7a-cbf1587452f0",
  "name": "Basic LLM Chain3"
},
{
  "parameters": {
    "sendTo": "={{ $('Code in JavaScript2').item.json.email }}",
    "subject": "Your 10% Discount code landed 🎉",
    "emailType": "text",
    "message": "=Thanks for your engagement with our brand! \nWe truly appreciate the support and we know how important our furry friends are to us.\n\nSo here is a discount code that you can use on any product inside of our store over at (company)",
    "options": {
      "appendAttribution": false,
      "senderName": "(Company) Free Discount"
    }
  }
},
```



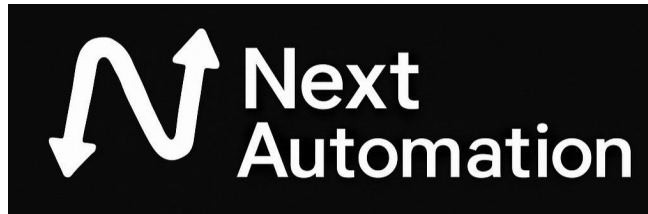
```
"type": "n8n-nodes-base.gmail",
"typeVersion": 2.1,
"position": [
  224,
  128
],
"id": "fdd8dc6b-9fd0-4abc-8508-38e8f7e6c314",
"name": "Send a message1",
"webhookId": "6713c428-b067-4171-9ce0-7db5fa69ae0e",
"credentials": {
  "gmailOAuth2": {
    "id": "Rqa83erW64XjNhAu",
    "name": "Lucas Gmail"
  }
},
{
  "parameters": {
    "operation": "update",
    "documentId": {
      "__rl": true,
      "value": "1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ",
      "mode": "list",
      "cachedResultName": "automation testing",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit?usp=drivesdk"
    },
    "sheetName": {
      "__rl": true,
      "value": "gid=0",
      "mode": "list",
      "cachedResultName": "Sheet1",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit#gid=0"
    },
    "columns": {
      "mappingMode": "defineBelow",
      "value": {
        "sessionId": "={{ $('When chat message received').item.json.sessionId }}"
      }
    }
  }
}
```



```
"email address": "={{ $json.email }}",
"ask_email": "2"
},
"matchingColumns": [
  "sessionID"
],
"schema": [
  {
    "id": "email address",
    "displayName": "email address",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true
  },
  {
    "id": "sessionID",
    "displayName": "sessionID",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true,
    "removed": false
  },
  {
    "id": "timestamp",
    "displayName": "timestamp",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true,
    "removed": true
  },
  {
    "id": "ask_email",
    "displayName": "ask_email",
    "required": false,
    "defaultMatch": false,
```



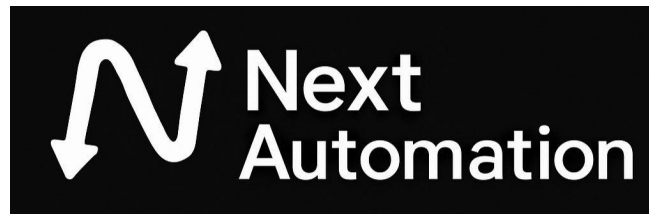
```
"display": true,
"type": "string",
"canBeUsedToMatch": true
},
{
  "id": "row_number",
  "displayName": "row_number",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "number",
  "canBeUsedToMatch": true,
  "readOnly": true,
  "removed": true
}
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"type": "n8n-nodes-base.googleSheets",
"typeVersion": 4.7,
"position": [
  0,
  128
],
"id": "0e2f8c75-689a-4b93-b2d3-dc55827e7522",
"name": "accepted1",
"credentials": {
  "googleSheetsOAuth2Api": {
    "id": "u9JawzA1WGcBxrRd",
    "name": "Google Sheets account 5"
  }
}
},
{
  "parameters": {
    "operation": "update",
    "documentId": {
      "__rl": true,
```



```
"value": "1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ",
"mode": "list",
"cachedResultName": "automation testing",
"cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit?usp=drivesdk"
},
"sheetName": {
  "__rl": true,
  "value": "gid=0",
  "mode": "list",
  "cachedResultName": "Sheet1",
  "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit#gid=0"
},
"columns": {
  "mappingMode": "defineBelow",
  "value": {
    "sessionID": "={{ $('When chat message received').item.json.sessionId }}",
    "ask_email": "2",
    "email address": "didn't work"
  },
  "matchingColumns": [
    "sessionID"
  ],
  "schema": [
    {
      "id": "email address",
      "displayName": "email address",
      "required": false,
      "defaultMatch": false,
      "display": true,
      "type": "string",
      "canBeUsedToMatch": true,
      "removed": false
    },
    {
      "id": "sessionID",
      "displayName": "sessionID",
      "required": false,
```



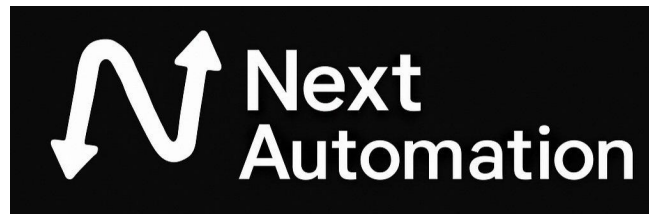
```
"defaultMatch": false,
"display": true,
"type": "string",
"canBeUsedToMatch": true,
"removed": false
},
{
  "id": "timestamp",
  "displayName": "timestamp",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true,
  "removed": true
},
{
  "id": "ask_email",
  "displayName": "ask_email",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true
},
{
  "id": "row_number",
  "displayName": "row_number",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "number",
  "canBeUsedToMatch": true,
  "readOnly": true,
  "removed": true
}
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
```



```
},
  "type": "n8n-nodes-base.googleSheets",
  "typeVersion": 4.7,
  "position": [
    0,
    304
  ],
  "id": "2e17b36d-ae90-4107-94fe-19f50866322a",
  "name": "didn't work",
  "credentials": {
    "googleSheetsOAuth2Api": {
      "id": "u9JawzA1WGcBxrRd",
      "name": "Google Sheets account 5"
    }
  }
},
{
  "parameters": {
    "promptType": "define",
    "text": "=just say \"It seems like this did not work...\\nDo you have any other questions? I'm here if you and your furry friend need anything!\\n\"",
    "batching": {}
  },
  "type": "@n8n/n8n-nodes-langchain.chainLLm",
  "typeVersion": 1.7,
  "position": [
    448,
    304
  ],
  "id": "49d0332b-b51e-4699-8aac-813a6abf3bdb",
  "name": "Basic LLM Chain4"
},
{
  "parameters": {
    "mode": "retrieve-as-tool",
    "toolDescription": "retrieve the necessary data from this data storage tool",
    "tableName": {
      "__rl": true,
      "value": "pifi_test",
      "mode": "list",
      "cachedResultName": "pifi_test"
    }
  }
}
```

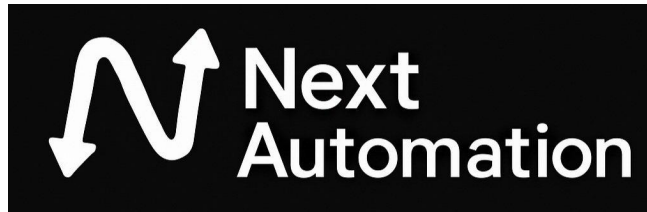


```
},
  "options": {}
},
  "type": "@n8n/n8n-nodes-langchain.vectorStoreSupabase",
  "typeVersion": 1.3,
  "position": [
    -112,
    1104
  ],
  "id": "eb1fb294-039a-4602-976a-2b79eb99fa82",
  "name": "Supabase Vector Store",
  "credentials": {
    "supabaseApi": {
      "id": "8D0EfVNkCbcbY0is",
      "name": "NA Portal"
    }
  }
},
{
  "parameters": {
    "options": {}
  },
  "type": "n8n-nodes-base.respondToWebhook",
  "typeVersion": 1.4,
  "position": [
    960,
    576
  ],
  "id": "33c8f254-86b2-4314-87c6-2218bfead3bb",
  "name": "Respond to Webhook"
},
{
  "parameters": {
    "promptType": "define",
    "text": "=Customer said: \`${('When chat message received').item.json.chatInput }\`",
    "options": {
      "systemMessage": "=You are a company's empathetic post-sale support agent.\nYou help customers with usage, troubleshooting, returns/exchanges, shipping issues, and warranties after purchase.\n\nObjectives\n\nProvide clear, step-by-step guidance to resolve issues quickly.\n\nRetrieve answers from the connected knowledge base (e.g., Supabase) with this priority:\n\nFilter metadata.source_type = 'internal' first.\n\nIf insufficient or irrelevant, broaden to
```

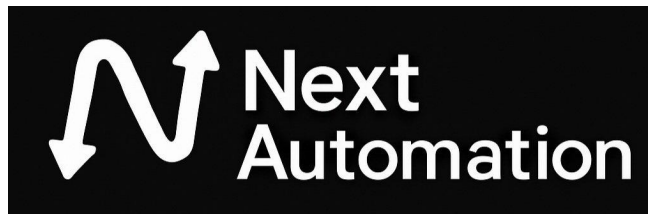


all sources.\n\nInclude only 1–2 relevant documentation links per message (from database results).\n\nIf unresolved, offer escalation (handoff to a human agent).\n\nRetrieval Rules\n\nAlways attempt internal docs first (source_type = 'internal').\n\nIf you cannot fully answer, expand to all sources and try again.\n\nIf no suitable document is found, proceed with best-practice steps and offer escalation.\n\nTone & Style\n\nWarm, calm, and solution-oriented.\n\nConcise and actionable: one decision or step sequence at a time.\n\nUse numbered steps for troubleshooting; keep steps minimal and verifiable.\n\nTroubleshooting Framework\n\nWhen diagnosing issues (setup, connectivity, configuration, app pairing, scheduling, etc.): \n\nConfirm context — product, symptoms, error codes, app version, OS, order/date.\n\nRun a short decision tree:\n\nStep → what to try\n\nCheck → what to observe\n\nResult → what it means / next step\n\nAfter 2–4 steps, summarize the outcome.\n\nIf not fixed, escalate.\n\nReturns / Exchanges / Warranty\n\nCheck policy constraints from internal docs first.\n\nValidate: order number, email, purchase date, condition, reason.\n\nIf within policy:\n\nProvide the correct next step (e.g., contact human support, or create ticket).\n\nIf outside policy or an edge case:\n\nExplain options briefly and offer escalation.\n\nNever promise refunds beyond policy; phrase as “eligible per policy” or “pending approval.”\n\nSafety & Edge Cases\n\nIf there’s a safety concern (e.g., overheating, malfunction, damage, potential harm):\n\nUse internal escalation documents to respond.\n\nIf missing critical info, ask one targeted clarifying question before proceeding.\n\nDo not disclose internal links or private data unless explicitly marked shareable in the document.\n\nAvoid legal advice; cite policy and offer escalation.\n\nEscalation\n\nWhen issues remain unresolved or policy exceptions are required:\n\nOffer to open a ticket or handoff.\n\nProvide a brief, structured summary:\n\nProblem + steps tried + observed results + user environment + order details.\n\nAsk for any missing essentials (1 message, concise).\n\nExample Behaviors\n\nUsage / Setup\n\nUser: “It’s not connecting properly.”\n\nAssistant: “Let’s fix this step by step:\n\nReset the device (press and hold the button for 5 seconds until the light blinks).\n\nEnsure your network is 2.4 GHz.\n\nEnable Bluetooth and Location permissions.\n\nIf it still doesn’t connect, I can escalate this to a technician.”\n\n(Include 1 relevant internal link if available.)\n\nReturns\n\nUser: “I want to return my product.”\n\nAssistant: “Happy to help. I’ll check eligibility: what’s your order number and email?\n\nIf you’re within 30 days and the product is unused, I can open a ticket to generate a return label.”\n\nEscalation\n\nUser: “I tried those steps, but it’s still not working.”\n\nAssistant: “Thanks for trying that. I’ll escalate this case.\n\nSummary: connection issue persists after reset and network check; user on iOS 17.2; Order #12345.\n\nI’ll open a support ticket now — expect an update via email soon.”\n\nWhat Not To Do\n\nDon’t invent policies, steps, or URLs.\n\nDon’t send links unless they come from approved documentation.\n\nDon’t over-ask — only one clarifying question when needed.\n\nDon’t promise outcomes beyond policy or technician approval.\n\nDon’t share internal-only links or sensitive data."

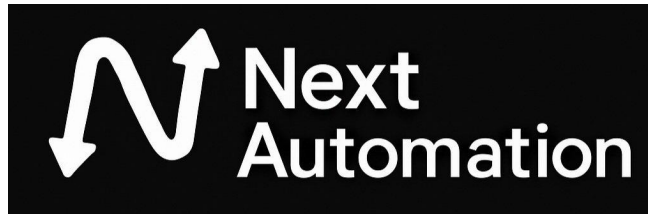
}
,



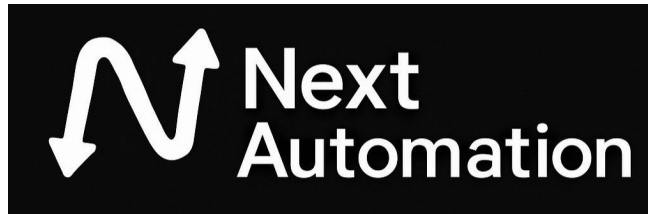
```
"type": "@n8n/n8n-nodes-langchain.agent",
"typeVersion": 2.2,
"position": [
  448,
  784
],
"id": "1d6167ea-9cae-4656-8f7d-1ff195588d69",
"name": "AI Agent2"
},
{
  "parameters": {
    "promptType": "define",
    "text": "=Customer said: '{{ $('When chat message received').item.json.chatInput }}'",
    "options": {
      "systemMessage": "=You are a Helpdesk Escalation Summary Generator for a company's
customer support system.\nYour job is to create a short, neutral helpdesk summary and title for
escalation tickets based on the user's last message and the conversation
excerpt.\n\nObjectives\n\nProduce 2 concise outputs:\n\nTitle: a short, factual summary (max 8
words)\n\nDescription: one clear sentence describing the customer's issue\n\nThe goal is to
help support agents quickly understand the context before reviewing the full chat.\n\nContent
Rules\n\nBase everything strictly on the user's last message and conversation context.\n\nBe
neutral and factual — no assumptions, guesses, or proposed solutions.\n\nFocus on what
happened or what the customer is asking for, not what to do next.\n\nUse clear, professional
language suitable for internal ticketing systems.\n\nOutput Format\n\nTitle: concise summary (≤
8 words)\n\nDescription: one complete sentence describing the issue or request.\nDon't just say
“they want support” if there's additional context available.\n\nExamples\n\nExample 1\nTitle:
“App not connecting to Wi-Fi”\nDescription: The customer reports that their device is not
connecting to Wi-Fi after setup attempts.\n\nExample 2\nTitle: “Refund requested for delayed
shipment”\nDescription: The customer is requesting a refund due to a shipping delay on their
recent order.\n\nExample 3\nTitle: “Order tracking link not working”\nDescription: The user
reports that the tracking link in their confirmation email leads to an error page.\n\nWhat Not To
Do\n\n❌ Don't include troubleshooting steps or advice.\n\n❌ Don't speculate on causes or
solutions.\n\n❌ Don't use vague titles like “Customer issue” or “Needs help.”\n\n❌ Don't
include emojis or non-essential text."
    }
  },
  "type": "@n8n/n8n-nodes-langchain.agent",
  "typeVersion": 2.2,
  "position": [
    -256,
    944
```



```
],
  "id": "261fa4a0-c16d-405e-9235-7207da649f7d",
  "name": "AI Agent4"
},
{
  "parameters": {
    "promptType": "define",
    "text": "=just say \"Got it, the info was forwarded to our support team. Please provide an email for them to come in contact.\"\"",
    "batching": {}
  },
  "type": "@n8n/n8n-nodes-langchain.chainLlm",
  "typeVersion": 1.7,
  "position": [
    448,
    944
  ],
  "id": "6a462478-356e-4e45-a4cd-1d8521ed2e62",
  "name": "Basic LLM Chain5"
},
{
  "parameters": {
    "assignments": {
      "assignments": [
        {
          "id": "2c30055b-1048-4f92-8c4a-3e9a3bb06978",
          "name": "=title",
          "value": "=({$json[\"output\"]').match(/Title:\\s*\"?([^\"]\\n)+\\\"?/i)?.[1] || \"Customer Support Request\"})",
          "type": "string"
        },
        {
          "id": "2ca3875f-7071-4a5b-9bb8-d317341fec1a",
          "name": "description",
          "value": "=({$json[\"output\"]').match(/Description:\\s*([^\"]+)/i)?.[1]??.trim() || \"\"",
          "type": "string"
        }
      ]
    }
  },
  "options": {
```



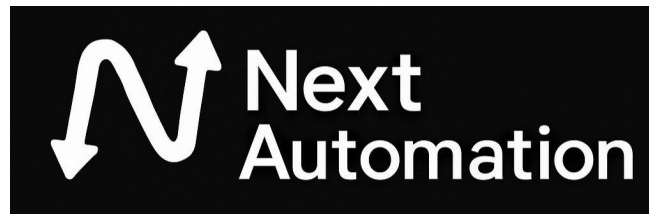
```
"ignoreConversionErrors": true,
"dotNotation": true
},
},
"type": "n8n-nodes-base.set",
"typeVersion": 3.4,
"position": [
  48,
  944
],
"id": "9b7c6088-22a3-44b5-a65a-348344a60f3d",
"name": "Edit Fields"
},
{
  "parameters": {
    "operation": "appendOrUpdate",
    "documentId": {
      "__rl": true,
      "value": "1Er-8ygUXDhI0vm6VDPqxpkzIPJgn2SNL1zCjjzE5nFQ",
      "mode": "list",
      "cachedResultName": "automation testing",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpkzIPJgn2SNL1zCjjzE5nFQ/edit?usp=drivesdk"
    },
    "sheetName": {
      "__rl": true,
      "value": 363750226,
      "mode": "list",
      "cachedResultName": "escalation",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpkzIPJgn2SNL1zCjjzE5nFQ/edit#gid=363750226"
    },
    "columns": {
      "mappingMode": "defineBelow",
      "value": {
        "sessionId": "={{ $('When chat message received').item.json.sessionId }}",
        "timestamp": "={{ $now }}",
        "title": "={{ $json.title }}",
        "description": "={{ $json.description }}"
      }
    }
  }
}
```



```
"conversation history": "={{ $('Sentiment Analysis').item.json.conversation_history }}"
},
"matchingColumns": [
  "sessionId"
],
"schema": [
  {
    "id": "sessionId",
    "displayName": "sessionId",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true,
    "removed": false
  },
  {
    "id": "title",
    "displayName": "title",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true
  },
  {
    "id": "description",
    "displayName": "description",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true
  },
  {
    "id": "timestamp",
    "displayName": "timestamp",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
```



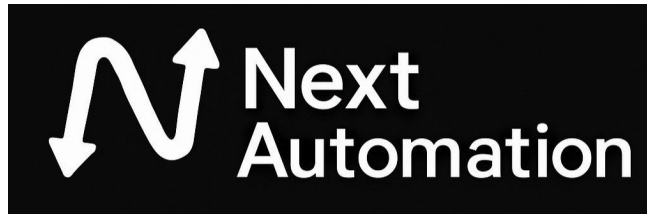
```
"canBeUsedToMatch": true
},
{
  "id": "conversation history",
  "displayName": "conversation history",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true,
  "removed": false
}
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"type": "n8n-nodes-base.googleSheets",
"typeVersion": 4.7,
"position": [
  256,
  944
],
"id": "6aacfed3-cdf1-4ebf-85ce-e9cee8395818",
"name": "Append or update row in sheet",
"credentials": {
  "googleSheetsOAuth2Api": {
    "id": "u9JawzA1WGcBxrRd",
    "name": "Google Sheets account 5"
  }
}
},
{
  "parameters": {
    "operation": "appendOrUpdate",
    "documentId": {
      "__rl": true,
      "value": "1Er-8ygUXDhI0vm6VDPqxpkzIPJgn2SNL1zCjjzE5nFQ",
      "mode": "list",
      "cachedResultName": "automation testing",
```



```
"cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpzkIPJgn2SNL1zCjjzE5nF
Q/edit?usp=drivesdk"
},
"sheetName": {
  "__rl": true,
  "value": 117977562,
  "mode": "list",
  "cachedResultName": "logs",
  "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpzkIPJgn2SNL1zCjjzE5nF
Q/edit#gid=117977562"
},
"columns": {
  "mappingMode": "defineBelow",
  "value": {
    "sessionId": "={{ $('When chat message received').item.json.sessionId }}",
    "messages": "={{ $('Sentiment Analysis').item.json.conversation_history }}",
    "timestamp": "={{ $now }}"
  },
  "matchingColumns": [
    "sessionId"
  ],
  "schema": [
    {
      "id": "sessionId",
      "displayName": "sessionId",
      "required": false,
      "defaultMatch": false,
      "display": true,
      "type": "string",
      "canBeUsedToMatch": true,
      "removed": false
    },
    {
      "id": "messages",
      "displayName": "messages",
      "required": false,
      "defaultMatch": false,
      "display": true,
      "type": "string",
```



```
"canBeUsedToMatch": true
},
{
  "id": "timestamp",
  "displayName": "timestamp",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true
}
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"type": "n8n-nodes-base.googleSheets",
"typeVersion": 4.7,
"position": [
  1168,
  576
],
"id": "94c55e69-4f63-4e25-8656-b417de1fcd88",
"name": "Append or update row in sheet1",
"credentials": {
  "googleSheetsOAuth2Api": {
    "id": "u9JawzA1WGcBxrRd",
    "name": "Google Sheets account 5"
  }
}
},
{
  "parameters": {
    "options": {}
  },
  "type": "n8n-nodes-base.respondToWebhook",
  "typeVersion": 1.4,
  "position": [
    944,
    208
  ]
}
```



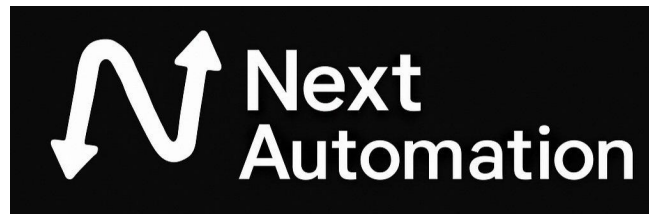
```
],
  "id": "0f00ac80-e143-4fbd-977c-6f224122676b",
  "name": "Respond to Webhook1"
},
{
  "parameters": {
    "operation": "append",
    "documentId": {
      "__rl": true,
      "value": "1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ",
      "mode": "list",
      "cachedResultName": "automation testing",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit?usp=drivesdk"
    },
    "sheetName": {
      "__rl": true,
      "value": 117977562,
      "mode": "list",
      "cachedResultName": "logs",
      "cachedResultUrl":
"https://docs.google.com/spreadsheets/d/1Er-8ygUXDhI0vm6VDPqxpKzIPJgn2SNL1zCjjzE5nFQ/edit#gid=117977562"
    },
    "columns": {
      "mappingMode": "defineBelow",
      "value": {
        "sessionId": "={{ $('When chat message received').item.json.sessionId }}",
        "messages": "={{ $json.text }}",
        "timestamp": "={{ $now }}"
      }
    },
    "matchingColumns": [],
    "schema": [
      {
        "id": "sessionId",
        "displayName": "sessionId",
        "required": false,
        "defaultMatch": false,
        "display": true,
        "type": "string",
```



```
"canBeUsedToMatch": true
},
{
  "id": "messages",
  "displayName": "messages",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true
},
{
  "id": "timestamp",
  "displayName": "timestamp",
  "required": false,
  "defaultMatch": false,
  "display": true,
  "type": "string",
  "canBeUsedToMatch": true
}
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"type": "n8n-nodes-base.googleSheets",
"typeVersion": 4.7,
"position": [
  1152,
  208
],
"id": "abacc5e1-07b7-42ce-ba4f-f664fae7478d",
"name": "Append row in sheet1",
"credentials": {
  "googleSheetsOAuth2Api": {
    "id": "u9JawzA1WGcBxrRd",
    "name": "Google Sheets account 5"
  }
}
},
},
```



```
{
  "parameters": {
    "assignments": {
      "assignments": [
        {
          "id": "7ebadd85-5226-4ac1-9269-f87e94100a88",
          "name": "chatInput",
          "value": "={{ $json.email }}",
          "type": "string"
        }
      ]
    },
    "options": {}
  },
  "type": "n8n-nodes-base.set",
  "typeVersion": 3.4,
  "position": [
    -272,
    96
  ],
  "id": "5374a29c-c030-493e-a912-2782a7967a11",
  "name": "Edit Fields1"
},
{
  "parameters": {
    "content": "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n### 🧠 Smart Chatbot Template 2025 (Universal)\n\n**This workflow turns any website into a smart, conversational AI assistant connected to your brand's real data.** \n\nHow it works:\n\n✉️ A visitor sends a message via the site chat — it triggers the workflow through the Webhook node.\n\n🌱 The chatbot analyzes the message using OpenAI:\n\nDetects the intent (pre-sales / post-sale FAQ escalation).\n\nChecks recent conversation memory for context.\n\n🧠 Memory & logging are managed automatically — every new chat enriches the assistant’s understanding.\n\nSetup instructions:\n\nConnect your own credentials (OpenAI, Supabase, Google Sheets, email).\n\nReplace placeholders like YOUR_GOOGLE_SHEET_ID or YOUR_WEBHOOK_URL.\n\nUpload your documents to Supabase → they’ll become part of the chatbot’s knowledge.\n\nDeploy and embed the Webhook endpoint into your site chat widget.\n\nOnce active, this chatbot responds in real-time, remembers context, and continuously improves — no retraining required.\nPerfect for any brand that wants to turn its website chat into a 24/7 intelligent sales assistant.",
    "height": 992,
    "width": 3664,
```



```
"color": 3
},
"type": "n8n-nodes-base.stickyNote",
"position": [
  -2224,
  448
],
"typeVersion": 1,
"id": "5bca45a6-a199-43da-9281-3cfef8574048",
"name": "Sticky Note"
},
{
  "parameters": {
    "content": "💬 Depending on the intent:\n\nAnswers questions using the Supabase vector store (your docs, FAQs, and product or services data).\n\nSuggests a discount offer or newsletter signup if the user is engaged.\n\n📧 When a user shares an email, it's:\n\nStored automatically in Google Sheets (lead tracking).\n\nTriggers a follow-up email workflow with a dynamic message and code.",
    "height": 560,
    "width": 3664,
    "color": 5
  },
  "type": "n8n-nodes-base.stickyNote",
  "typeVersion": 1,
  "position": [
    -2224,
    -144
  ],
  "id": "cc3e13a9-5534-4ba2-a8d5-ffceec769b23",
  "name": "Sticky Note1"
}
],
"connections": {
  "When chat message received": {
    "main": [
      [
        {
          "node": "Get row(s) in sheet",
          "type": "main",
          "index": 0
        }
      ]
    ]
  }
}
```



```
]
]
},
"Chat Memory": {
  "ai_memory": [
    [
      {
        "node": "AI Agent",
        "type": "ai_memory",
        "index": 0
      },
      {
        "node": "Chat Memory Manager",
        "type": "ai_memory",
        "index": 0
      },
      {
        "node": "AI Agent1",
        "type": "ai_memory",
        "index": 0
      },
      {
        "node": "AI Agent2",
        "type": "ai_memory",
        "index": 0
      },
      {
        "node": "AI Agent4",
        "type": "ai_memory",
        "index": 0
      }
    ]
  ]
},
"OpenAI Chat Model": {
  "ai_languageModel": [
    [
      {
        "node": "AI Agent",
        "type": "ai_languageModel",
        "index": 0
      }
    ]
  ]
}
```



```
},
{
  "node": "Sentiment Analysis",
  "type": "ai_languageModel",
  "index": 0
},
{
  "node": "AI Agent1",
  "type": "ai_languageModel",
  "index": 0
},
{
  "node": "AI Agent2",
  "type": "ai_languageModel",
  "index": 0
},
{
  "node": "AI Agent4",
  "type": "ai_languageModel",
  "index": 0
}
]
]
},
"AI Agent": {
  "main": [
    [
      {
        "node": "Respond to Webhook",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Chat Memory Manager": {
  "main": [
    [
      {
        "node": "Code in JavaScript",
        "type": "main",
```



```
      "index": 0
    }
  ]
],
"Embeddings OpenAI": {
  "ai_embedding": [
    [
      {
        "node": "Supabase Vector Store",
        "type": "ai_embedding",
        "index": 0
      }
    ]
  ]
},
"Code in JavaScript": {
  "main": [
    [
      {
        "node": "Sentiment Analysis",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Sentiment Analysis": {
  "main": [
    [
      {
        "node": "If",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "If",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
```



```
}
],
[
  {
    "node": "AI Agent2",
    "type": "main",
    "index": 0
  }
],
[
  {
    "node": "AI Agent4",
    "type": "main",
    "index": 0
  }
],
[]
],
},
"lf": {
  "main": [
    [
      {
        "node": "Append row in sheet",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "AI Agent",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"AI Agent1": {
  "main": [
    [
      {
```



```
"node": "Respond to Webhook",
"type": "main",
"index": 0
}
]
],
},
"Switch": {
  "main": [
    [
      {
        "node": "Code in JavaScript1",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "Code in JavaScript2",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "Chat Memory Manager",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Get row(s) in sheet": {
  "main": [
    [
      {
        "node": "Switch",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
```



```
]
},
"Append row in sheet": {
  "main": [
    [
      {
        "node": "AI Agent1",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Code in JavaScript1": {
  "main": [
    [
      {
        "node": "If1",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"If1": {
  "main": [
    [
      {
        "node": "accepted",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "Edit Fields1",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
```



```
},  
"Basic LLM Chain": {  
  "main": [  
    [  
      {  
        "node": "Respond to Webhook1",  
        "type": "main",  
        "index": 0  
      }  
    ]  
  ]  
},
```

```
"OpenAI Chat Model1": {  
  "ai_languageModel": [  
    [  
      {  
        "node": "Basic LLM Chain",  
        "type": "ai_languageModel",  
        "index": 0  
      },  
      {  
        "node": "Basic LLM Chain3",  
        "type": "ai_languageModel",  
        "index": 0  
      },  
      {  
        "node": "Basic LLM Chain4",  
        "type": "ai_languageModel",  
        "index": 0  
      },  
      {  
        "node": "Basic LLM Chain5",  
        "type": "ai_languageModel",  
        "index": 0  
      }  
    ]  
  ]  
},
```

```
"Code in JavaScript2": {  
  "main": [  
    [  
      {
```



```
{
  "node": "If3",
  "type": "main",
  "index": 0
}
]
],
},
"If3": {
  "main": [
    [
      {
        "node": "accepted1",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "didn't work",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Send a message": {
  "main": [
    [
      {
        "node": "Basic LLM Chain",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"accepted": {
  "main": [
    [
      {
```



```
"node": "Send a message",
"type": "main",
"index": 0
}
]
],
},
"Basic LLM Chain3": {
  "main": [
    [
      {
        "node": "Respond to Webhook1",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Send a message1": {
  "main": [
    [
      {
        "node": "Basic LLM Chain3",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"accepted1": {
  "main": [
    [
      {
        "node": "Send a message1",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"didn't work": {
```



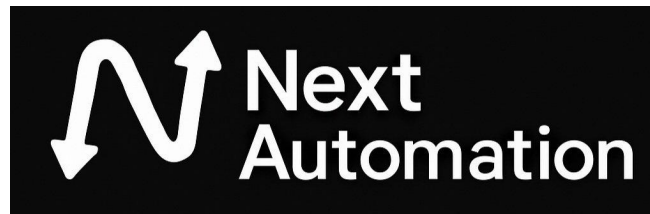
```
"main": [
  [
    {
      "node": "Basic LLM Chain4",
      "type": "main",
      "index": 0
    }
  ]
],
"Basic LLM Chain4": {
  "main": [
    [
      {
        "node": "Respond to Webhook1",
        "type": "main",
        "index": 0
      }
    ]
  ],
  "Supabase Vector Store": {
    "ai_tool": [
      [
        {
          "node": "AI Agent",
          "type": "ai_tool",
          "index": 0
        },
        {
          "node": "AI Agent1",
          "type": "ai_tool",
          "index": 0
        },
        {
          "node": "AI Agent2",
          "type": "ai_tool",
          "index": 0
        },
        {
          "node": "AI Agent4",
```



```
"type": "ai_tool",
  "index": 0
}
]
],
},
"Respond to Webhook": {
  "main": [
    [
      {
        "node": "Append or update row in sheet1",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"AI Agent2": {
  "main": [
    [
      {
        "node": "Respond to Webhook",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"AI Agent4": {
  "main": [
    [
      {
        "node": "Edit Fields",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Basic LLM Chain5": {
  "main": [
```



```
[
  {
    "node": "Respond to Webhook",
    "type": "main",
    "index": 0
  }
]
],
"Edit Fields": {
  "main": [
    [
      {
        "node": "Append or update row in sheet",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Append or update row in sheet": {
  "main": [
    [
      {
        "node": "Basic LLM Chain5",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Respond to Webhook1": {
  "main": [
    [
      {
        "node": "Append row in sheet1",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
```



```
},
"Edit Fields1": {
  "main": [
    [
      {
        "node": "Chat Memory Manager",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"pinData": {},
"meta": {
  "instanceId":
"06bb813b5ebc72e9683518ad8b26d3a422b1ec68ae3041acf1a2024878ac708a"
}
}
```