□ Day 2 AI Automations with No-Code

No-code AI automation allows you to connect AI models (like GPT or image generators) with everyday apps and tools **without writing any code**. You can automate tasks like email replies, content generation, customer support, data extraction, and more — all visually and easily.

□ What Is No-Code AI Automation?

No-code automation means using platforms that allow users to **design workflows** using a visual drag-and-drop interface instead of programming.

When combined with **AI tools** (like GPT, image models, or speech-to-text), you can:

- **Automate business workflows**
- **Create smart chatbots**
- **Generate content**
- **Integrate AI into your everyday tools** (like Google Sheets, Slack, Notion)

□ Top No-Code AI Automation Platforms

Platform	What It Does	Notes
Zapier	Automate tasks between apps using AI steps	GPT, DALL·E, Hugging Face integrations
Make (Integromat)	Powerful visual workflows with HTTP & AI modules	Great for complex automations
Pipedream	No-code + low-code workflows with GPT APIs	Dev-friendly
Bubble	Build AI-powered web apps visually	Can integrate with OpenAI, Replicate
Airtable + Automations	Smart spreadsheet + built-in automation	Use GPT for auto responses, tagging
Voiceflow	No-code AI conversation design	Great for chatbots & voice apps
Retool	Build internal AI tools fast	Low-code, great UI
OpenAI GPTs (ChatGPT Plus)	Custom GPTs with actions	Use OpenAPI + files + instructions

---

□ Example Use Cases

□ Content Creation

- Auto-generate blog posts from outlines
- Social media captioning
- Email marketing personalization

□ Customer Support

- Smart reply to common support emails (Zapier + GPT)
- Summarize tickets (Make + OpenAI)
- Generate support docs from transcripts

□ Data Handling

- Extract structured data from emails or PDFs
- Summarize meeting transcripts
- Auto-tag incoming messages based on sentiment

□ Smart Chatbots

- No-code GPT chatbot on your website (Voiceflow, Landbot)
- Lead qualification bot
- Internal knowledge assistant

□ Automation Examples

Trigger	AI Action	Output
New email in Gmail	Use GPT to summarize	Send to Slack
New row in Google Sheets	Fill out missing column with GPT	Save to Notion
User fills a form	Generate email response	Send via Outlook
New audio file	Transcribe with Whisper	Store in Google Drive

□ Tools You Can Plug In

□ AI Models

- **OpenAI (GPT, DALL·E, Whisper)**
- **Hugging Face Spaces**
- **Replicate** (hosted models like SDXL, Whisper, etc.)
- **Stability AI (Stable Diffusion)**

□ Automation & Integration

- **Zapier** (easy + wide app support)
- **Make.com** (advanced)
- **IFTTT** (simpler workflows)
- **Webhooks** (to trigger AI via HTTP)

□□ Example: "Smart Meeting Summary Bot"

Tools Used: Zoom → Zapier → OpenAI → Google Docs

Workflow:

1. Trigger: New Zoom recording uploaded
2. Transcribe: Use Whisper API (via Make or Zapier)
3. Summarize: Use GPT to summarize key takeaways
4. Output: Auto-generate and store a formatted Google Doc

□ Benefits of No-Code AI Automation

Advantage	Why It Matters
□ Easy to use	No programming skills required
□ Smart	Use AI to handle unstructured content
□ Integrates widely	Works with 1,000s of apps
□ Fast	Build and deploy automations in hours
□ Cost-effective	Reduce manual workload and errors

□ Bonus: Security + Best Practices

- Use **environment variables** to store API keys securely
- Always **test with dummy data** first
- Add **fallback steps** if AI fails
- Monitor cost usage for APIs (e.g., OpenAI)

□ Ready to Start?

□ Beginner Projects to Try:

- ✉□ Auto-email replier with GPT + Gmail + Zapier
- □ Weekly newsletter generator using Google Sheets + GPT
- □ Product description writer from Shopify → OpenAI
- □ FAQ chatbot for your website using Voiceflow or ChatGPT Custom GPTs

What Is n8n?

n8n (pronounced "n-eight-n") is a **powerful open-source workflow automation tool** — similar to Zapier or Make — but with:

□ **More flexibility**
□ **More control**
□ **Self-hosting support**
□ **Built-in code nodes** for custom logic

It's a **hybrid no-code + low-code** platform that's ideal for **AI-powered automations**.

□ Why Use n8n for AI Automation ?

Feature	Benefit
□ API integrations	Connect to OpenAI, Hugging Face, Replicate, etc.
□ Code node	Add JavaScript when you need extra logic
□ Webhooks	Use GPT or other models in real-time via HTTP
□ Self-hosted or Cloud	Full data control and privacy
□ File handling	Great for automation with PDFs, docs, and files
□ Modular workflows	Build complex, multi-step AI workflows easily

---

□ AI Use Cases with n8n

Here are **common AI-powered automations** built with n8n:

Use Case	Description
Email summarizer	Monitor inbox → send content to GPT → output summary
Document analyzer	Upload PDF → parse text → extract data using OpenAI
Blog post writer	Input topic via webhook → auto-generate article with GPT
AI chatbot backend	Collect message → process with GPT → return reply via API
Meeting recap generator	Audio file → Whisper transcribe → GPT summarize → Send to Notion/Slack

---

□ How To Use OpenAI in n8n

Here's a simplified **workflow example**:

□ GPT-Powered Auto-Reply

Trigger: New email (via IMAP or Gmail module)
Steps:

1. Extract subject + body
2. Send to OpenAI (GPT API) via **HTTP Request node**
3. Parse GPT response
4. Send reply using **Send Email node**

Bonus: Add **delay, filters, conditionals** or sentiment analysis!

□ Sample Node Config (OpenAI API Call)

Use the **HTTP Request node** with:

- **Method**: `POST`
- **URL**: `https://api.openai.com/v1/chat/completions`
- **Headers**:
  - `Authorization: Bearer YOUR_API_KEY`
  - `Content-Type: application/json`
- **Body (JSON)**:

```
{
  "model": "gpt-4",
  "messages": [
    { "role": "system", "content": "You are a helpful assistant." },
    { "role": "user", "content": "Summarize this email: {{ $json.body }}" }
  ]
}
```

□ Combine With Other Tools

You can combine GPT with:

Tool	Purpose
Google Sheets	Store or log results
Notion	Auto-generate notes or tasks
Telegram / Slack	AI assistants in chat
Twilio / WhatsApp	Create SMS/WhatsApp bots
Whisper API	Transcribe voice to text
Replicate API	Use Stable Diffusion for image generation

□ Bonus: Why Choose n8n Over Zapier or Make?

Feature	n8n	Zapier / Make
Self-hosting	□ Yes	□ No
Open-source	□ Yes	□ No
Free tier	□ Generous	□ Limited
Code flexibility	□ JS in workflows	□ Limited
AI model control	□ Full API access	□ Limited endpoints
Best for devs or power users	□	□

□ Learn More

- □ n8n.io
- □ n8n Docs
- □ YouTube tutorials: Search *"n8n OpenAI workflow"*

n8n whatsapp chatbot

Let's walk through a **step-by-step guide** to create an **AI chat interface in n8n using Ollama + LLaMA 3**.

This setup lets you create a **local/private AI chatbot** by:

1. Running **LLaMA 3 via Ollama**
2. Connecting it to **n8n** using HTTP requests
3. Creating a **chat-style automation workflow**

□ Requirements

Tool	Description
□ **Ollama**	Runs LLaMA 3 (or other LLMs) locally via API
□ **n8n**	Automation platform (self-hosted or cloud)
□ **LLaMA 3**	A high-quality open-source language model

□ Optional: Frontend or webhook trigger To make a chat interface (e.g. Telegram, website, etc.)

□ Step-by-Step Guide

□ Step 1: Install & Run Ollama with LLaMA 3

If you haven't already:

```
# Install Ollama (macOS, Windows, Linux)
curl -fsSL https://ollama.com/install.sh | sh

# Run LLaMA 3 model
ollama run llama3
```

Once running, the Ollama API is available at:
`http://localhost:11434/api/generate`

□ Step 2: Open n8n and Create a New Workflow

1. Launch your local n8n (or open your cloud instance).
2. Create a **new workflow**.
3. Add a **trigger node**:
    - Use **Webhook** (for frontend or app)
    - OR **Telegram Trigger**, **Schedule**, etc.

□ Step 3: Add an HTTP Request Node (to call Ollama)

□ *Configuration:*

- **Node Name**: "Call LLaMA 3 via Ollama"
- **Method**: `POST`
- **URL**: `http://localhost:11434/api/generate`
- **Authentication**: None (Ollama runs locally)
- **Headers**:
    - `Content-Type`: `application/json`

□ *JSON Body:*

```
{
  "model": "llama3",
  "prompt": "{{$json["message"]}}",
  "stream": false
}
```

□ Replace `message` with the variable from the trigger node (like a Telegram message or form input).

□ Step 4: Parse the AI Response

- Use a **Set node** or a **Function node** to extract the response text.

Example for Function Node:

```
return [{
  json: {
    reply: $json.response
  }
}];
```

□ Step 5: Send the AI Response Back

Depending on your trigger:

- **Webhook**: Use `Respond to Webhook` node
- **Telegram**: Use `Telegram > Send Message` node
- **Slack**: Use `Slack > Send Message` node
- **Frontend**: Send back as JSON or HTML

□ Step 6: Save and Activate Workflow

- Click **"Activate"**
- Test by sending a message to the webhook/Telegram bot

□ Sample Use Case: Web-Based Chatbot (Webhook)

Here's how you can simulate a full chatbot:

Node Type	Purpose
Webhook Trigger	Accept POST from frontend (message input)
HTTP Request	Send prompt to Ollama + LLaMA 3
Set / Function	Extract response text
Respond to Webhook	Send back response JSON

Frontend (example JS fetch):

```
fetch('https://your-n8n-server/webhook/ai-chat', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({ message: "What's the weather like today?" })
})
.then(res => res.json())
.then(data => console.log(data.reply));
```

□ Final Output Flow

```
User Input → [Webhook] → [Ollama via HTTP] → [Parse] → [Reply]
```

□ Pro Tips

Tip	Benefit
Use **stream: true** for streaming replies	More dynamic UX (advanced)
Add **context or memory** using variables	Simulate a chat memory
Chain with **OpenAI**, **Whisper**, or **image models**	Multimodal workflows
Use **n8n credentials manager**	To securely store model configs

---

□ Summary

Step	What You Did
1	Installed Ollama + LLaMA 3
2	Created n8n workflow with trigger
3	Called Ollama API with prompt
4	Extracted AI response
5	Returned it to chat or UI

you're looking to use the **n8n + Ollama + LLaMA 3 chatbot** via a **chat interface**, likely embedded in a website or web app using a **Webhook** in n8n as the backend. Let's build that step by step.

□ Goal

Create a **real-time chatbot UI** (like ChatGPT) on your website that:

1. Sends messages to an **n8n webhook**
2. n8n sends the prompt to **LLaMA 3 (via Ollama)**
3. The response is returned and displayed in the **chat UI**

□ Requirements Recap

Tool	Purpose
n8n	Automation logic + Webhook
Ollama	Runs LLaMA 3 locally via API
HTML + JS	Frontend chat interface

□ Step-by-Step Implementation

□ 1. Run LLaMA 3 via Ollama

Ensure it's running:

```
ollama run llama3
```

Test it via terminal or:

```
curl http://localhost:11434/api/generate \
  -d '{ "model": "llama3", "prompt": "Hello!" }'
```

□ 2. Build the n8n Workflow

□ *A. Create Webhook Trigger*

- Add a **Webhook** node
- Set:
  - **HTTP Method**: `POST`
  - **Path**: `llama-chat` (or anything you like)
  - **Response Mode**: `On Received`

□ This will generate a webhook like:

```
http://localhost:5678/webhook/llama-chat
```

(Or your public n8n domain)

□ *B. Add HTTP Request to Ollama*

- Add an **HTTP Request** node:
  - **Method**: `POST`
  - **URL**: `http://localhost:11434/api/generate`
  - **Body**:
  - `{`
  - `  "model": "llama3",`
  - `  "prompt": "{{$json["message"]}}",`
  - `  "stream": false`
  - `}`
- **Headers**:
  - `Content-Type: application/json`

□ *C. Send Response Back*

Add a **Respond to Webhook** node:

- Use:
- `{`
- `  "reply": {{$json["response"]}}`
- `}`

□ 3. Test in n8n

Send a POST request manually:

```
curl -X POST http://localhost:5678/webhook/llama-chat \
  -H "Content-Type: application/json" \
  -d '{"message":"What is AI?"}'
```

 You should get a JSON response from LLaMA 3.

□ 4. Build a Simple HTML Chat Interface

```
<!DOCTYPE html>
<html>
<head>
  <title>AI Chat (n8n + LLaMA 3)</title>
  <style>
    body { font-family: sans-serif; max-width: 600px; margin: auto; }
    #chat { border: 1px solid #ccc; padding: 10px; height: 400px; overflow-y:
auto; margin-bottom: 10px; }
    .msg { margin: 5px 0; }
    .user { color: blue; }
    .bot { color: green; }
  </style>
</head>
<body>

  <h2>Chat with LLaMA 3</h2>
  <div id="chat"></div>

  <input id="input" type="text" placeholder="Type your message..."
style="width: 80%;" />
  <button onclick="sendMessage()">Send</button>

  <script>
    async function sendMessage() {
      const input = document.getElementById('input');
      const chat = document.getElementById('chat');
      const message = input.value.trim();
      if (!message) return;

      // Show user message
      const userMsg = document.createElement('div');
      userMsg.className = 'msg user';
      userMsg.textContent = "You: " + message;
      chat.appendChild(userMsg);

      input.value = '';
      chat.scrollTop = chat.scrollHeight;

      // Send to n8n webhook
      const response = await fetch('http://localhost:5678/webhook/llama-
chat', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ message })
      });
```

```
      const data = await response.json();

      // Show bot reply
      const botMsg = document.createElement('div');
      botMsg.className = 'msg bot';
      botMsg.textContent = "LLaMA 3: " + data.reply;
      chat.appendChild(botMsg);
      chat.scrollTop = chat.scrollHeight;
    }
  </script>

</body>
</html>
```

 Change the webhook URL to your **public n8n endpoint** if you're hosting remotely.

□ Bonus: Chat History / Context

To simulate chat memory:

- Add a **memory variable** (store chat history in a variable or database)
- Append it to the prompt:
- `"prompt": "Conversation so far: ...\nUser: {{$json["message"]}}"`

□ Summary

Step	What You Did
1	Run LLaMA 3 with Ollama
2	Create n8n webhook workflow
3	Connect to Ollama via HTTP node
4	Respond back to Webhook
5	Build frontend chat interface