AWaDH
Agriculture and Water Technology Development Hub

AI CHATBOT FOR CPS LAB USING n8n

What will you learn from this module:

In this experiment you will learn to build an AI-powered chatbot in n8n that fetches live CPS Lab sensor data through an API and answers the user's questions using an OpenAI language model.

Requirements:

- Internet connection
- n8n Cloud account **or** local n8n installation
- Web browser
- OpenAI API key (or n8n Gateway credits)
- CPS Sensor API access

Prerequisites:

- Basic knowledge of workflow automation
- Basic understanding of APIs
- Basic understanding of JSON
- Basic understanding of AI/chatbots

Workflow Overview:

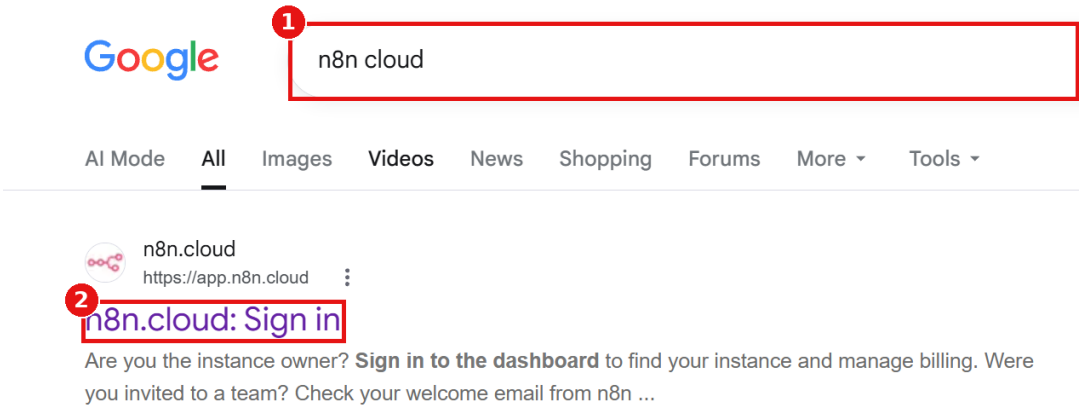
The chatbot is built from four nodes. The user's message starts the workflow, the sensor API is called, and the AI model answers using the returned data.

Node name	n8n node type	Purpose
Chat Trigger	Chat Trigger (On chat message)	Opens the chat window and starts the workflow when the user sends a message.
Fetch CPS Sensor Data (API)	HTTP Request	Calls the CPS Lab sensor API and gets the live sensor values.
OpenAI Chat Model	OpenAI Chat Model	Provides the language model (gpt-3.5-turbo) to the chain. Needs an OpenAI credential (Gateway credits or your own API key).
Analyze Sensor Data	Basic LLM Chain	Sends the question + sensor data to the model and returns the answer to the chat.

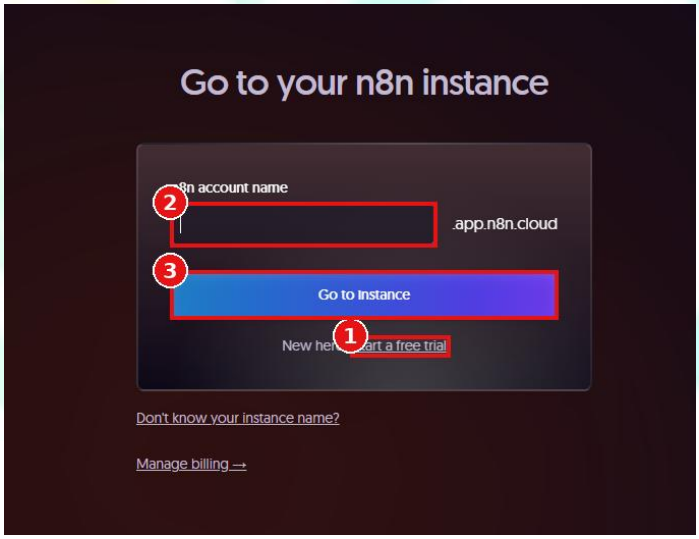
Setup and Configuration:

Part A – Setup using n8n Cloud (online)

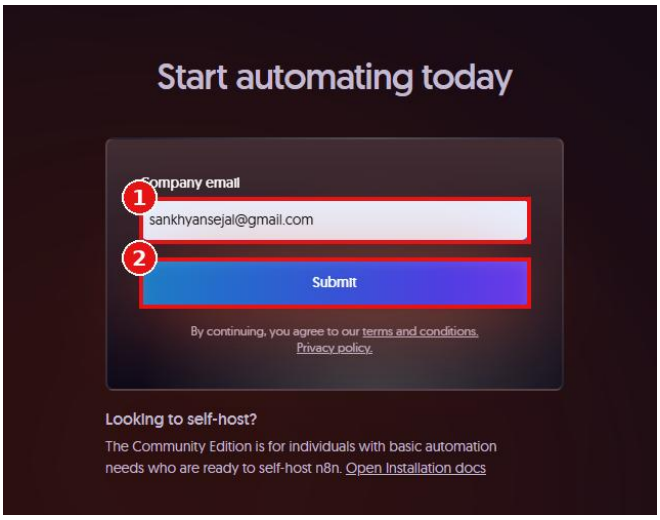
➢ Open a web browser and search for **n8n cloud [1]**. Click on the first link, **n8n.cloud: Sign in [2]**, as shown in the picture below.



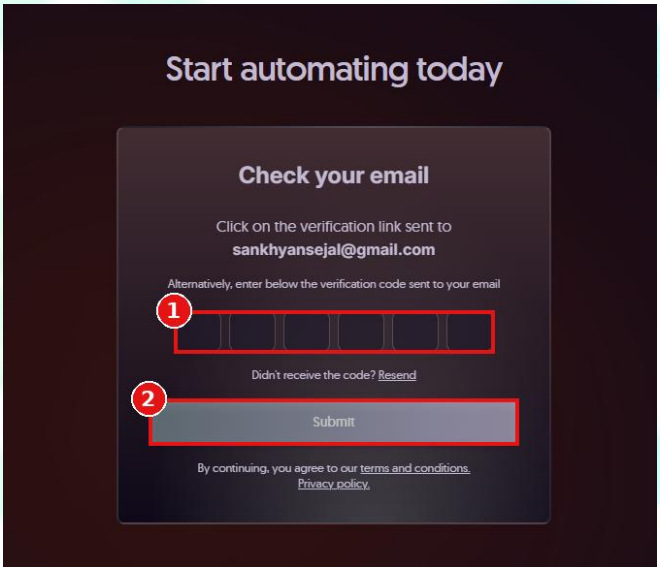
➢ On the sign-in page click on **Start a free trial [1]**. (If you already have an account, enter your **n8n account name [2]** and click **Go to instance [3]**.)



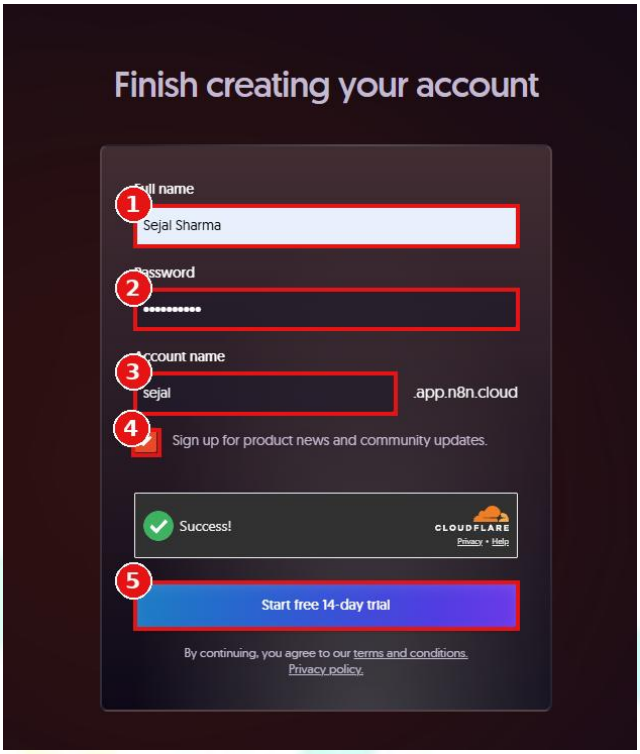
➢ Enter your **company/e-mail address [1]** and click on **Submit [2]**.



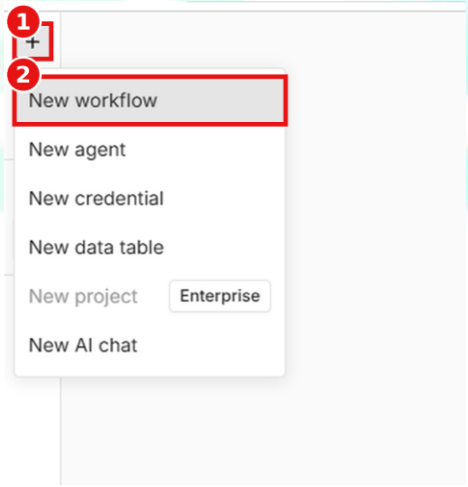
➢ Open your mailbox and click the verification link, **or** type the verification code in the boxes **[1]** and click **Submit [2]**.



➢ On the **Finish creating your account** page enter your **Full name [1]**, **Password [2]** and **Account name [3]**.
➢ Tick the checkbox **[4]**, then click on the start-trial button **[5]** to create your n8n Cloud account.



➢ After logging in, you will land on the n8n dashboard. Click on the **+ [1]** button and select **New workflow [2]** to open an empty workflow canvas.
➢ Rename the workflow (top-left) to **CPS Lab Assistant**.

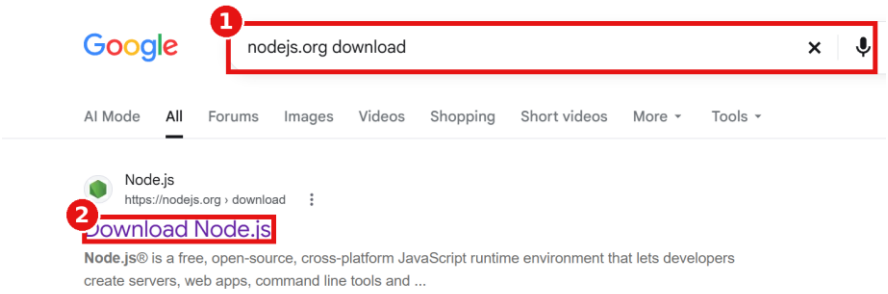


Part B – Setup using local n8n installation (on your computer)

Use this method if you want to run n8n on your own computer instead of n8n Cloud. After finishing Part B, continue with Part C.

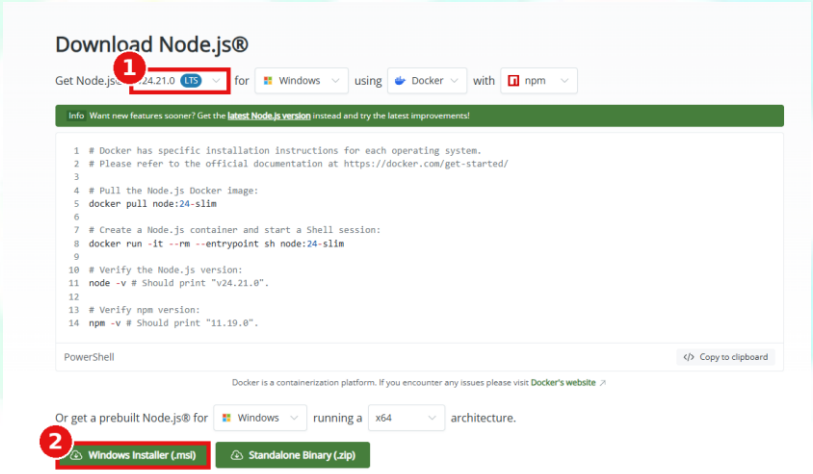
Step 1: Install Node.js

➢ Search for **nodejs.org download [1]** on Google and click on the first link **Download Node.js [2]**.



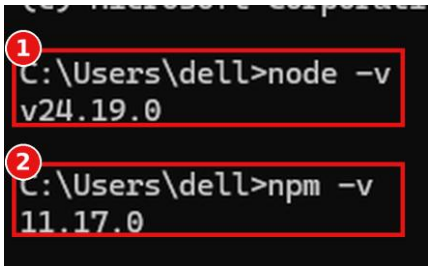
➢ On the download page select the **LTS [1]** version (n8n needs Node.js between **20.19 and 24.x**) and click on **Windows Installer (.msi) [2]**.

➢ Open the downloaded file, click **Next** on every page, accept the license, click **Install** and then **Finish**.



➢ Open **Command Prompt** (press **Win + R**, type **cmd** and press Enter) and check the installation by typing the commands below. The version numbers must be shown for **node [1]** and **npm [2]**.

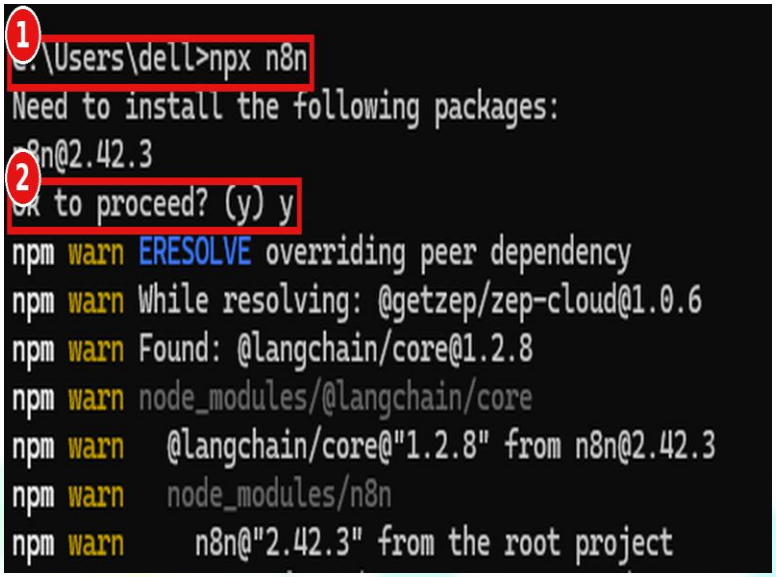
```
node -v
npm -v
```



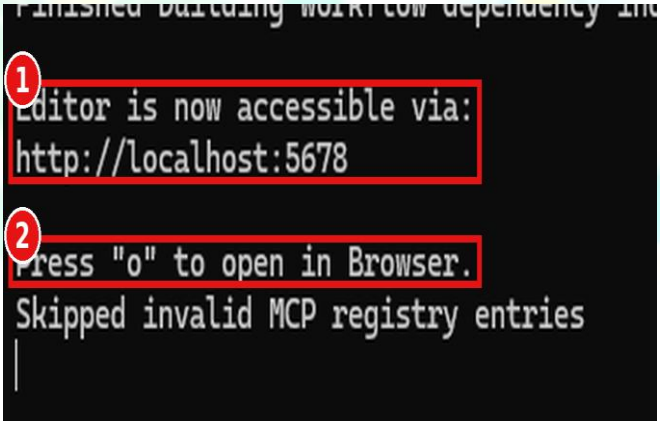
Step 2: Start n8n

➢ In the Command Prompt type **npx n8n [1]** and press Enter. If it asks **Ok to proceed? [2]** type **y** and press Enter. The first start downloads n8n, so wait for a few minutes.

```
npx n8n
```



➢ When the message **Editor is now accessible via http://localhost:5678 [1]** appears, n8n is running. Press **o [2]** to open it in the browser (or open it manually in Step 3). **Do not close this window** while using n8n.

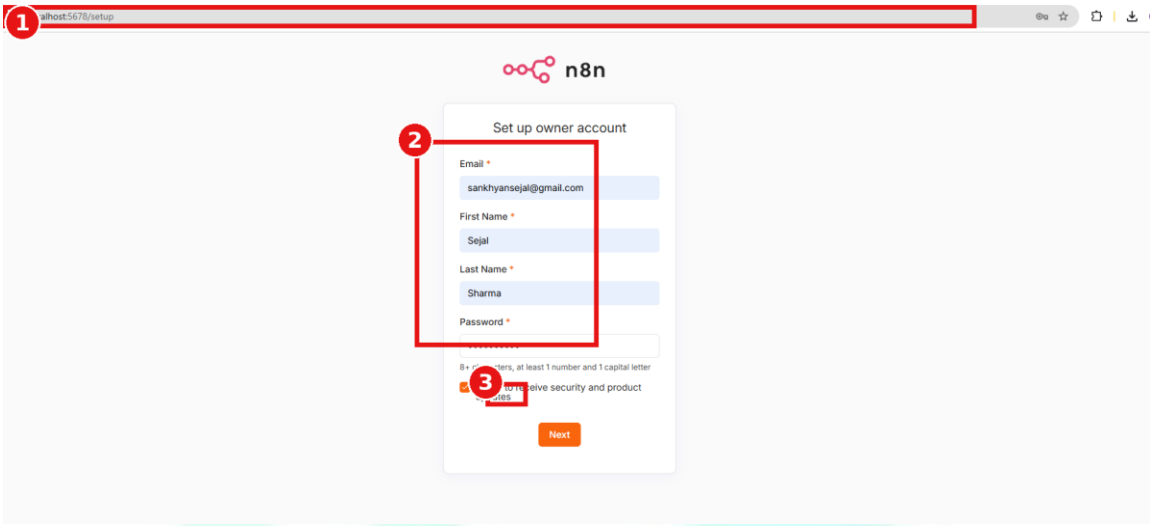


➢ **Optional (permanent install):** instead of npx you can install n8n once with the first command below and start it any time with the second command.

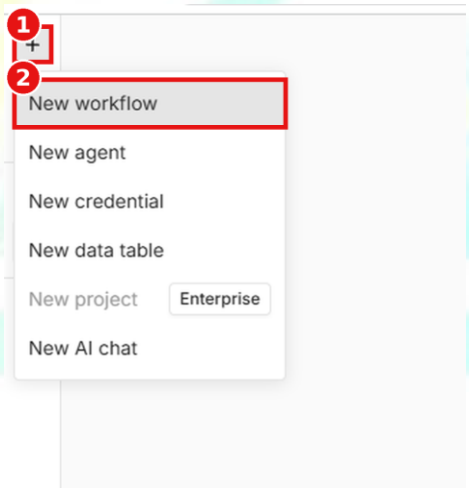
```
npm install n8n -g
n8n start
```

Step 3: Open the n8n editor

- Open a web browser and go to **http://localhost:5678 [1]**.
- On the first start, the **Set up owner account** page appears. Enter your **Email, First Name, Last Name and Password [2]** and click on **Next [3]**. Skip the optional questions that follow.



- The n8n dashboard opens. Click on the **+ [1]** button, select **New workflow [2]** and rename the workflow to **CPS Lab Assistant**.



- **Optional – using Docker:** if Docker Desktop is installed, run the commands below instead of npx and then open **http://localhost:5678**.

```
docker volume create n8n_data
docker run -it --rm --name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n docker.n8n.io/n8nio/n8n
```

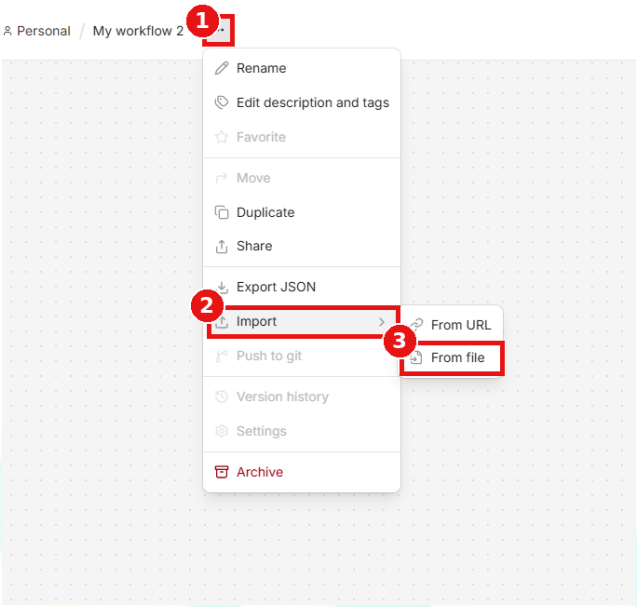
Part C – Import the workflow using JSON code

➢ Copy the complete JSON code given below, paste it in **Notepad** and save the file as **cps_lab_assistant.json [1]** (set **Save as type** to **All files**).

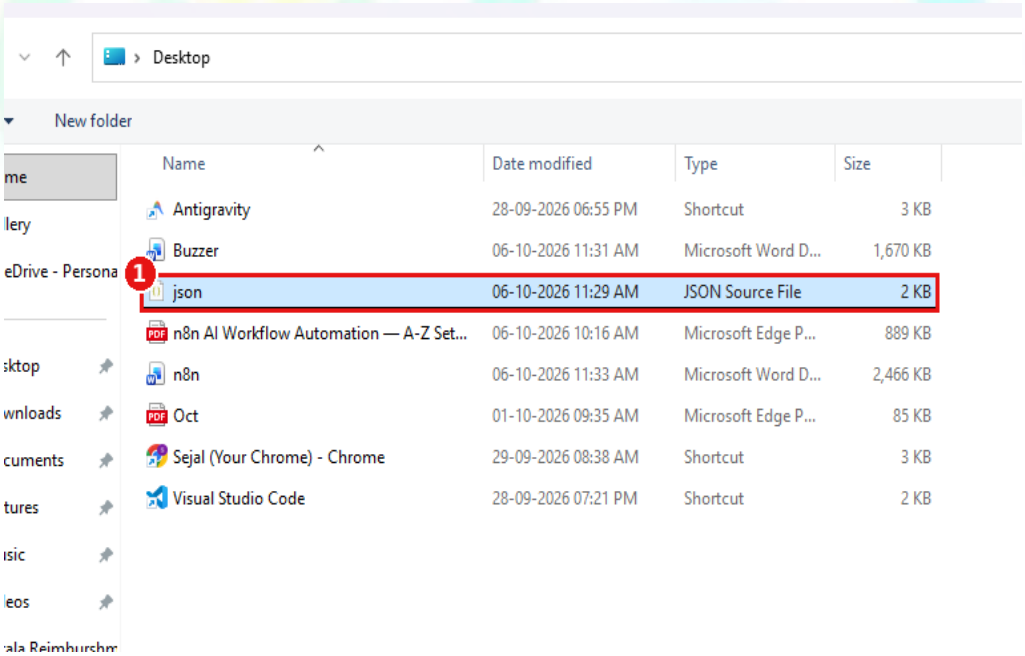
```
{
  "nodes": [
    {
      "parameters": {},
      "id": "chat-trigger",
      "name": "Chat Trigger",
      "type": "@n8n/n8n-nodes-langchain.chatTrigger",
      "typeVersion": 1.1,
      "position": [ 100, 300 ]
    },
    {
      "parameters": {
        "url": "https://gtk47vexob.execute-api.us-east-1.amazonaws.com/polytechnicdata",
        "sendQuery": true,
        "queryParameters": {
          "parameters": [
            { "name": "deviceid", "value": "7" },
            { "name": "startdate", "value": "01-10-2026" },
            { "name": "enddate", "value": "01-10-2026" },
            { "name": "key", "value": "Annam@2025" }
          ]
        },
        "options": {}
      },
      "id": "fetch-sensors",
      "name": "Fetch CPS Sensor Data (API)",
      "type": "n8n-nodes-base.httpRequest",
      "typeVersion": 4.2,
      "position": [ 300, 300 ]
    },
    {
      "parameters": {
        "model": {
          "__rl": true,
          "mode": "id",
          "value": "gpt-3.5-turbo"
        },
        "options": {}
      },
      "id": "openai-model",
      "name": "OpenAI Chat Model",
      "type": "@n8n/n8n-nodes-langchain.lmChatOpenAi",
      "typeVersion": 1.2,
      "position": [ 500, 450 ]
    },
    {
      "parameters": {
        "promptType": "define",
        "text": "=You are the CPS Lab Assistant. The user asked: {{ $('Chat Trigger').item.json.chatInput }}\n\nHere
is the live sensor data from the lab: {{ JSON.stringify($json) }}\n\nPlease answer their question based on the sensor
data."
      },
      "id": "analyze-sensor",
      "name": "Analyze Sensor Data",
      "type": "@n8n/n8n-nodes-langchain.chainLlm",
      "typeVersion": 1.4,
      "position": [ 500, 300 ]
    }
  ],
  "connections": {
    "Chat Trigger": {
      "main": [ [ { "node": "Fetch CPS Sensor Data (API)", "type": "main", "index": 0 } ] ]
    },
    "Fetch CPS Sensor Data (API)": {
      "main": [ [ { "node": "Analyze Sensor Data", "type": "main", "index": 0 } ] ]
    },
    "OpenAI Chat Model": {
      "ai_languageModel": [ [ { "node": "Analyze Sensor Data", "type": "ai_languageModel", "index": 0 } ] ]
    }
  }
```

```
}
```

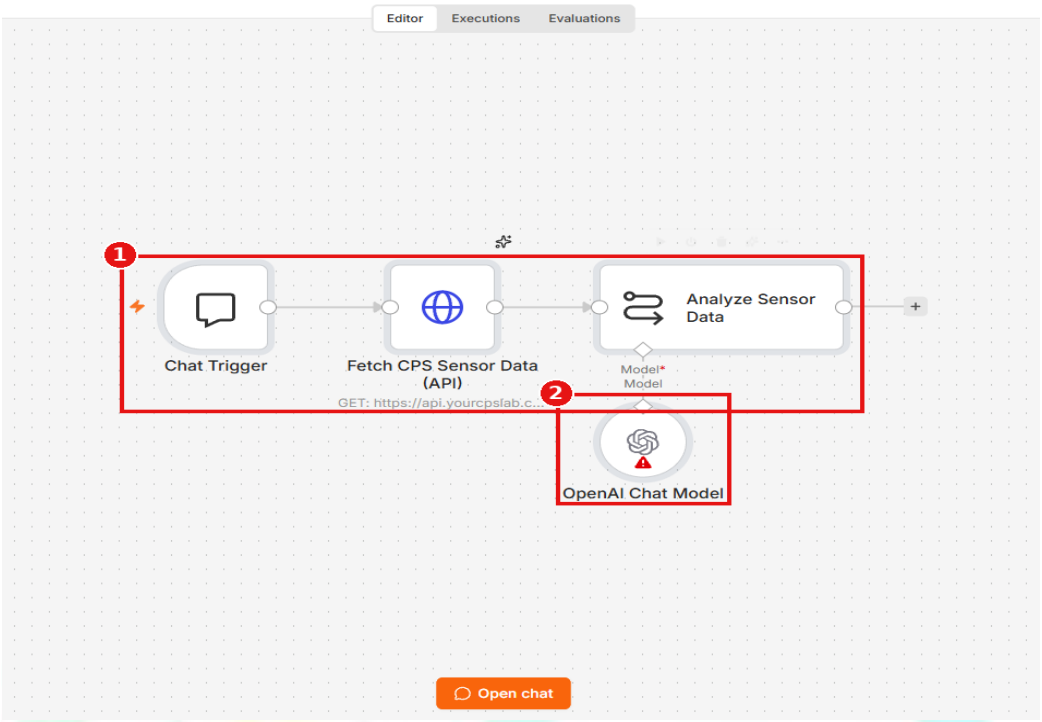
➢ Open an empty workflow (created in Part A or Part B). Click on the **⋯ menu [1]** next to the workflow name, then click on **Import [2]** → **From file [3]**.



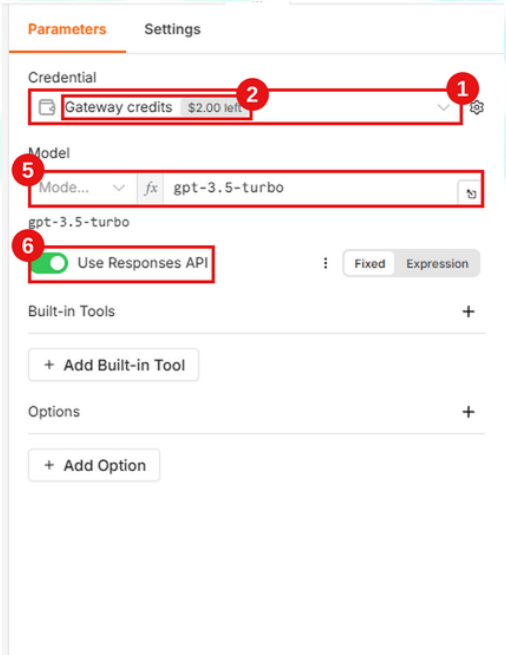
➢ In the file window select the saved **JSON file [1]** and click on **Open**.



➢ All the nodes appear on the canvas, already connected to each other **[1]**. The **OpenAI Chat Model [2]** shows a red warning sign because its credential is not added yet. (Alternatively, copy the JSON code, click on an empty canvas and press **Ctrl + V**.)

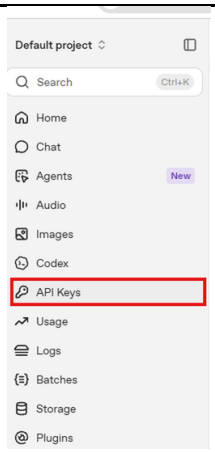


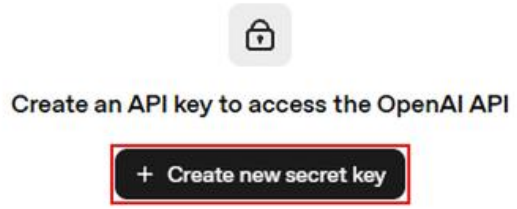
➢ Open the **OpenAI Chat Model** node (double-click it). In the **Parameters** tab, click the **Credential [1]** drop-down and select **Gateway credits [2]** (the free credits provided by n8n; the remaining balance, e.g. $2.00 left, is shown next to it). With Gateway credits you do **not** need to enter an OpenAI API key.

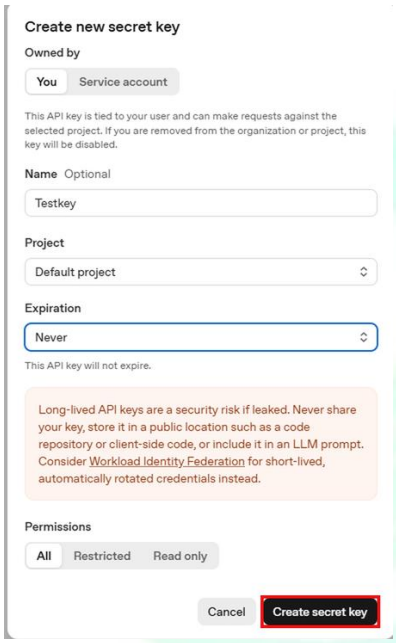


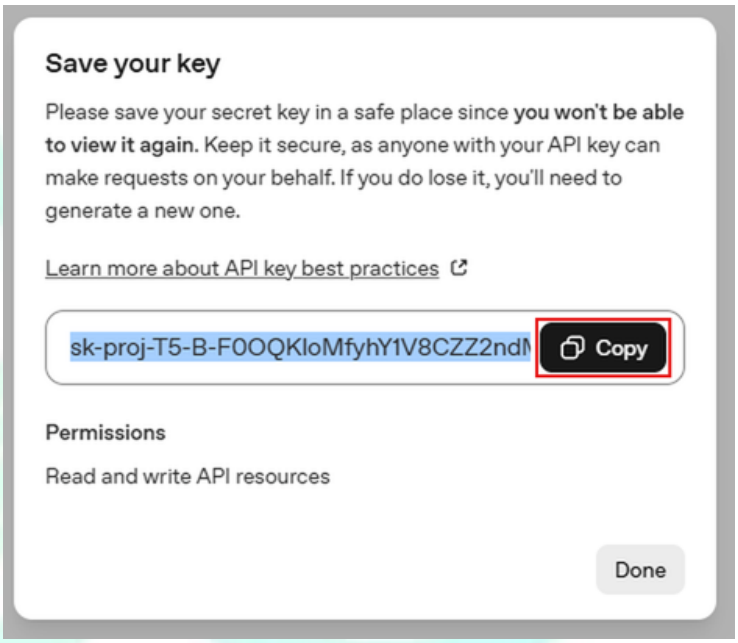
➢ **Where to write your own OpenAI API key:** if you want to use your own key, click the **Credential [1]** drop-down and choose **Create new credential**. Open **platform.openai.com → API keys → Create new secret**

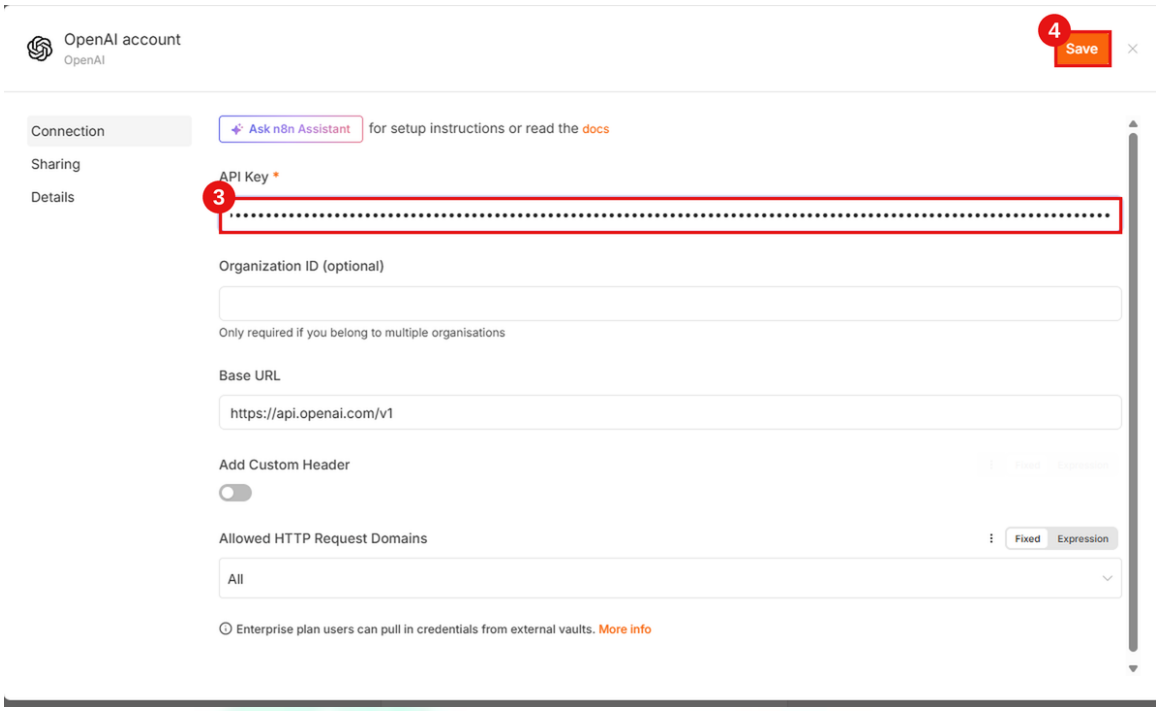
key, copy the key, paste it in the **API Key [3]** field of the credential window and click on **Save [4]**. The key is stored safely inside n8n; never write it in the prompt or in the HTTP Request node.



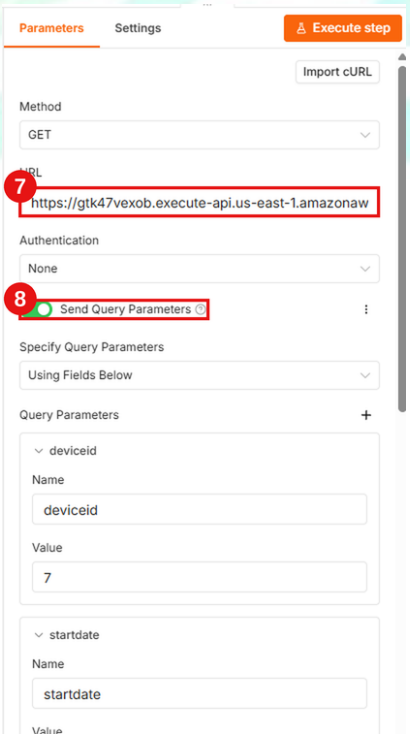








- Under **Model [5]** the value **gpt-3.5-turbo** is shown in the expression field. Keep the **Use Responses API [6]** switch **ON** (green) and leave **Built-in Tools** and **Options** empty.
- Open the **Fetch CPS Sensor Data (API)** node. Set **Method** to **GET** and in the **URL [7]** field enter the API address without the "?" part: **https://gtk47vexob.execute-api.us-east-1.amazonaws.com/polytechnicdata**
- Switch ON **Send Query Parameters [8]**, click **Add Parameter** and add these four name / value pairs: **deviceid** = 7, **startdate** = 01-10-2026, **enddate** = 01-10-2026, **key** = Annam@2025 (your API key). Dates are in **dd-mm-yyyy** format. (Alternative: paste the complete link with "?deviceid=7&startdate=…" directly in the URL field.)



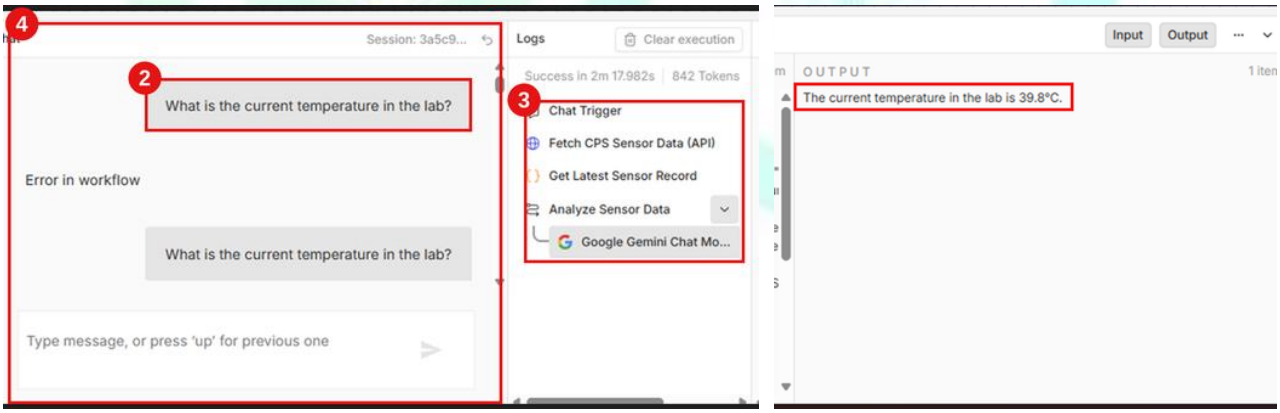
- Click on **Save** at the top-right corner to save the workflow.
- **Note:** In the prompt, JSON.stringify($json) passes the whole API response to the AI. If your API returns the readings inside a field named **data**, you may use **{{ $json.data }}** instead.
- **Tip (today's data):** to always fetch the current date, set **startdate** and **enddate** to the expression **{{ $now.toFormat('dd-MM-yyyy') }}** instead of a fixed date.

Settings used in each node:

Node name	Setting	Value
Chat Trigger	Mode	Default (chat opens inside n8n)
Fetch CPS Sensor Data (API)	Method / URL	GET / https://gtk47vexob.execute-api.us-east-1.amazonaws.com/polytechnicdata + query parameters deviceid, startdate, enddate, key
Analyze Sensor Data	Source for Prompt	Define below – prompt: "You are the CPS Lab Assistant…" (already filled by the JSON)
OpenAI Chat Model	Credential / Model	Gateway credits (or your own OpenAI API key) / gpt-3.5-turbo, Use Responses API ON

Part D – Test the chatbot

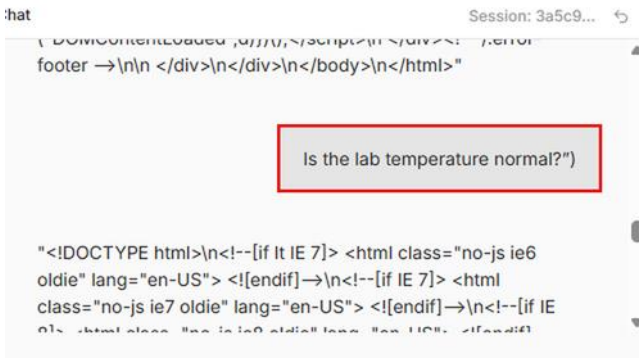
- Click on the **Open chat [1]** button at the bottom of the canvas. The chat window opens.
- Type a question such as **"What is the current temperature in the lab?" [2]** and press Enter.
- The nodes turn **green [3]** one after another and the AI answer based on live sensor data is shown in the chat window **[4]**.

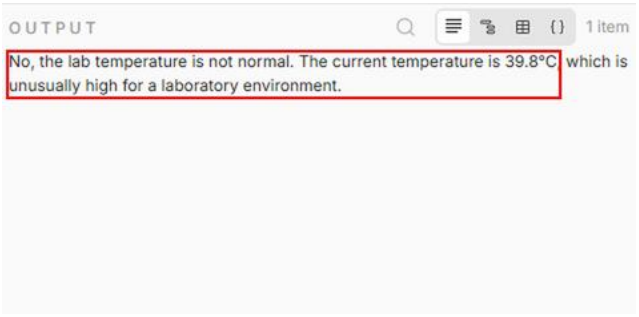


- To make the chatbot always available, switch the **Active** toggle **[1]** (top-right) ON. Open the **Chat Trigger** node and enable **Make Chat Publicly Available [2]** to get a shareable chat link.
- On a **local** installation the chat link works only on your own computer (localhost).

❖ OUTPUT :-

➢ The chatbot reads the live CPS Lab sensor data and replies to the user's question in natural language, as shown below.





Troubleshooting:

Problem	Possible reason	Solution
HTTP Request node shows an error	Wrong URL or the API is not reachable	Check the URL, test it in a browser and make sure the CPS API is online.
OpenAI node shows "Invalid API key" or "Quota exceeded"	Wrong key or no credit in the OpenAI account	Create a new key and check billing/usage on platform.openai.com. If you use Gateway credits, check the balance shown in the Credential drop-down.
AI answer does not match the question	Chat input is missing in the prompt	Use the prompt of Step 3 that contains the Chat Trigger input.
Pasted JSON does not appear on the canvas	JSON copied incompletely	Copy the complete code from the first "{" to the last "}" and paste again.
'npx' or 'node' is not recognized	Node.js not installed or the terminal was opened before installing	Install Node.js LTS, then close and reopen Command Prompt.
n8n shows an error about the Node.js version	Node.js version is outside 20.19 – 24.x	Install a supported Node.js LTS version and run the command again.
localhost:5678 does not open	n8n is not running or the terminal was closed	Start n8n again and keep the Command Prompt window open.
Port 5678 is already in use	Another n8n or program uses the port	Close the other n8n window or restart the computer and start n8n again.