# Can You Build a Survey in Node-RED?

Yes — but with limitations. Here are different ways and considerations:

## 1. Use Node-RED Dashboard (or UI) to Build Forms

- Node-RED Dashboard has *form input* type widgets where you can collect user responses (text, number, checkbox, etc.). snap4city.org
- You can wire those form inputs to a flow: when a user submits, the flow receives a message (`msg.payload`) with form data.
- You can store the data in a database (e.g., SQLite, MySQL, Mongo, etc.), or send it somewhere else (HTTP, MQTT, etc.).

**Pros**:

- Everything is inside Node-RED.
- You control the storage, validation, how you process data.

**Cons**:

- UI is limited: Node-RED Dashboard is fairly basic compared to dedicated survey platforms.
- Not trivial to do *complex* survey logic (conditional questions, branching, "if answer = X then next question Y").
- Maintaining sessions / user identity: if you want to track individual respondents across multiple pages / days, you need to handle storing "who is this" yourself. As one forum post notes:

  > "you will have to … maintain the state of that id over the life of the survey … You will also need a suitable database." Node-RED Forum

- UI and UX are not as polished as survey tools (unless you build a custom web UI).

---

## 2. Use UI Builder (or custom web UI) + HTTP Nodes

Instead of using the built-in Dashboard nodes, you can:

- Use **node-red-contrib-uibuilder** (or similar) to build a custom web page (HTML/JS) for your survey.
  - In the UI builder, you can design survey pages exactly how you like (nice form layout, wizard-style, multi-page).
  - When user submits, send data back to Node-RED via websocket or HTTP.
- In Node-RED, receive the data via HTTP-in or websocket, then process/store it.

**Pros**:

- Very flexible UI (you’re not limited by dashboard widgets).
- Easy to build multi-page survey, conditional logic, etc.
- Better user experience is possible.

**Cons**:

- Requires more web development (HTML, JS, CSS).
- More work to wire up and maintain.
- You need to design session management if you want to track who is filling the survey (unless anonymous).

---

## 3. Use External Survey Tools + Integrate with Node-RED

Another approach is: use a survey service (Google Forms, LimeSurvey, Typeform, etc.) and connect Node-RED to that service.

- Example: Host survey on LimeSurvey (open source), then use Node-RED to pull results (via APIs) or get webhooks when someone submits.
- Or, use Google Forms + Google Sheets, then Node-RED to read form responses from the sheet.

**Pros**:

- Leverage survey platforms built for this (UI, analytics, branching, response export).
- Less custom work in Node-RED for the survey UI.

**Cons**:

- Less control over UI or flow inside Node-RED.
- You depend on external service.