How To Write JSON To Create the Question Tree

1. The Basic Concept

The questionnaire is a **Decision Tree**.

1. We start with a few "Root" questions.
2. Depending on the user's answer (Yes or No), we can trigger **new questions** (Sub-questions).
3. Every answer adds or subtracts **Points** from specific diseases.
4. At the end, the app combines these points with the AI Scan to give a final probability.

2. File Structure

You will be editing a file named `questions.json`. Every question is a "Block" wrapped in curly braces `{ }`.

A Standard Question Block

```
{
  "id": "root_bleeding",
  "text": "Has the spot bled spontaneously?",
  "points_if_yes": {
    "Melanoma": "high",
    "Benign Keratosis": "n_high"
  },
  "points_if_no": {
    "Melanoma": "n_low"
  },
  "next_questions_if_yes": ["sub_how_long"],
  "next_questions_if_no": []
}
```

3. The Fields Explained

id (Required)

A unique nickname for the question.

- **Must be unique.** No two questions can have the same ID.
- **Root Questions:** Must start with root_ (e.g., root_color, root_size). These appear first.
- **Sub Questions:** Can be named anything (e.g., sub_itch, q_pain).
- **Format:** Use lowercase letters and underscores _. No spaces.

text (Required)

The actual text shown to the user on the screen.

- Keep it simple and non-medical (e.g., use "itchy" instead of "pruritus").

points_if_yes & points_if_no

This determines how the diagnosis changes based on the answer.

- **Format:** "Disease Name": "Score Code"
- You can list as many diseases as you want.

The Score Codes:

I decided to use low, med etc because if we may decide to change the numerics, I can easily change them witout changing too much of the code or the json.

“low”	Low Indication	+5
“med” or “medium”	Medium Indication	+10
“high”	High Indication	+20
“very_high”	Critical Indication	+40
“n_low”	Negative Low	-5
“n_med”	Negative Medium	-10
“n_high”	Negative High	-20
“n_very_high”	Negative Critical	-40

next_questions_if_yes & next_questions_if_no

This is how you create the "Tree."

- This is a **List** of IDs to show next.
- If the user answers "Yes", the app looks for the IDs inside next_questions_if_yes and adds them to the screen immediately.
- **Important:** You must verify that the ID you type here actually exists somewhere else in the file.

4. Valid Disease Names

You must use these exact spellings (Case Sensitive): (we can change these easily but they are case sensitive)

- Melanoma
- Basal Cell
- Benign Keratosis
- Dots Group
- Red Regions
- Skin Dryness
- Unknown Normal

5. Examples

Example 1: A Root Question that rules out Melanoma

If the spot is perfectly symmetrical, it's likely benign. If not, it's risky.

```
{
  "id": "root_symmetry",
  "text": "Is the spot perfectly symmetrical (one half matches the other)?",
  "points_if_yes": {
    "Unknown Normal": "high",
    "Melanoma": "n_high"
  },
  "points_if_no": {
    "Melanoma": "med"
  },
  "next_questions_if_no": ["sub_border_irregularity"],
  "next_questions_if_yes": []
}
```

Example 2: A Sub-Question about Pain

This question only appears if triggered by another question. It gives points but triggers no further questions.

```
{
  "id": "sub_pain",
  "text": "Is the spot painful to the touch?",
  "points_if_yes": {
    "Basal Cell": "med",
    "Melanoma": "low"
  },
  "points_if_no": {},
  "next_questions_if_yes": [],
  "next_questions_if_no": []
}
```

6. Critical Rules (The "Don't Break the App" List)

1. **Watch Your Commas ,**

    - In JSON, every line *except the last one* in a block needs a comma.

      Wrong:

      ```
      "Melanoma": "high"

      "Basal Cell": "low"
      ```

      Right:

      ```
      "Melanoma": "high",

      "Basal Cell": "low"
      ```

1. **Quotes are Mandatory**

    - **Everything (Keys and Values) must be inside double quotes " ".**
    - **Wrong: Melanoma: high**
    - **Right: "Melanoma": "high"**

2. **Unique IDs**

    - **Never use the same ID twice. If you copy-paste a block, change the id immediately.**

3. **No Dead Links**

    - **If you put "next_questions_if_yes": ["sub_itch"], you must have a question block defined with "id": "sub_itch". If not, the app will crash (and show a red error screen).**

7. Workflow Recommendation

1. **Draft on Paper: Draw the decision tree first. Arrows represent "Yes" or "No".**
2. **Write the Sub-Questions First: Write the sub_ blocks at the bottom of the file.**
3. **Write the Root Questions: Write the root_ blocks at the top and link them to the sub-questions you just wrote.**
4. **Validate: Check that every opening brace { has a closing brace } and every list [has a closing].**
5. **I recommend copy-pasting the previously written code tbh...**