Researching and Implementing an API for Automated Responses

1. API Research & Selection

Options for ChatGPT Integration:

- **OpenAI's Official API**
  - Directly sends prompts with sensor data to GPT-3.5/GPT-4.
  - Best for quick prototyping, well-documented, but has usage costs.
- **Self-Hosted Open-Source LLM**
  - Privacy-friendly, runs locally, but requires GPU resources.
- **Middleware**
  - Better prompt engineering and workflow management.
  - Useful for multi-step reasoning.
- **Custom Rule-Based API**
  - Fast, no LLM dependency, but inflexible for complex logic.

Recommended Approach:

Start with OpenAI's API for rapid testing, then explore self-hosting or hybrid models later.

2. How to Use the API

Steps to Implement OpenAI API:

1. **Format Sensor Data into a Structured Prompt**
   - Example:

     "Predict user availability based on:

     - Screen status: off for 10 mins

     - Last call: 30 mins ago

     - Motion: walking

Return: [Available: Yes/No, Confidence: High/Medium/Low, Reason: ...]"

2. **Make API Call (Python Example)**

```
import openai

response = openai.ChatCompletion.create(

  model="gpt-3.5-turbo",

  messages=[

    {"role": "system", "content": "Analyze sensor data for availability."},

    {"role": "user", "content": "Screen: off 10 mins. Last call: 30 mins ago. Motion: 
walking."}

  ]

)

print(response.choices[0].message.content)
```

3. **Parse & Display Response**
   - Extract key details (e.g., Available: No, Reason: User in motion but screen off for long).

3. App Navigation Flow

Key Screens & Structure:

- Enter the sensor data as input
- Button to input the data and trigger the API call
- The API in the backend sends the input data to ChatGPT and creates a prompt
- The result returned by ChatGPT about user availability gets returned to the app
- The returned result gets displayed in a textbox on the app

4. Next Steps

- **Testing:** Validate API with mock sensor data.
- **Optimization:** Improve prompts for better accuracy.
- **Privacy:** Mask sensitive data before API calls.
- **Fallback:** Add rule-based logic if API fails.