Conveos
software company

💼 Resume Chatbot

🔧 Objective:

Create a simple AI-like chatbot web application that responds to user prompts about your Resume/Profile. The chatbot should simulate an interview experience by interpreting input questions and returning relevant responses based on pre-defined data.
🛑 The project has to be uploaded on GitHub in a public repository.

🧩 Project Components:

1. Frontend (Web Interface)

Build a simple UI (HTML/React/Vue/Angular/etc.) where the user can:

- Type in a prompt
- Send it to the backend
- View the chatbot's reply

Recommended: React+TailwindCSS, HTML/JS + TailwindCSS

2. Backend (API Server)

Develop an API (Nodejs,JAVA,Python) with:

- A single `POST /chat` endpoint that:
  - Accepts a user prompt
  - Processes the prompt to identify intent (via keyword detection)
  - Returns an appropriate predefined answer

Recommended: Nodejs+Express

NOTES:
Feel free to use any language or framework you are more comfortable with.

🛑 The project has to be uploaded on GitHub in a public repository.

📘 Example Prompts & Expected Responses

These are examples. You can have your own format and type of responses.

Prompt	Detected Keywords	Response
"Tell me something about yourself"	about yourself, who are you	"Hi, I'm Alex, a computer science student passionate about building web applications and AI."
"What projects have you worked on?"	projects, worked on	"I've developed a weather app, a personal budgeting tool, and a RESTful API for a task manager."
"What are your skills?"	skills, expertise	"I'm skilled in JavaScript, Python, React, and SQL."
"Where did you study?"	study, education	"I graduated from XYZ University with a degree in Computer Science."
"What are your hobbies?"	hobbies, interests	"I enjoy hiking, reading science fiction, and working on open-source projects."
etc…		

✅ Evaluation Criteria

- **Functionality**: Does the chatbot respond correctly based on keyword detection?
- **Code Structure**: Is the code modular, readable, and well-documented?
- **Creativity**: Are there thoughtful additions, which will make the experience of the user or the selection of the answers more easily?
- **Git:** Be able to keep a clean commit tree with correct commit messages.

🧠 Hints for Applicants (Optional)

- Use a simple keyword-to-response map (a dictionary/object).
- Keep the UI clean and minimal – focus on functionality.
- Optional: add a list of sample questions the user can click on.

GOOD LUCK!