



**LearnTube**  
**by CareerNinja**

**Assignment:** Advanced conversational chatbot for data gathering and deployment

**Objective:**

Inspired by Typeform's chat feature, especially Formless AI's user data collection approach, we aim to create an interactive conversational chatbot that actively engages users while discreetly handling data collection.

This chatbot should operate as a persuasive conversationalist, encouraging users to share their information willingly. For example:

**User:** "I'm not sure if I want to share my email."

**Chatbot:** "I completely understand your concern. Your email is safe with us and will only be used for sending updates on our services and exclusive offers. Is there anything specific you're concerned about?"

**User:** "I'm not comfortable sharing my phone number."

**Chatbot:** "Your privacy is important to us. Rest assured, we take all necessary precautions to protect your data. If you're not comfortable sharing your phone, is there another way we can stay in touch with you?"

The chatbot should take a proactive approach, initiating interactions and adeptly transitioning into small talk or different topics when users hesitate to share details. For example:

Chatbot asks for the user's name, and the user declines:

**User:** "I'd rather not share my name."

**Chatbot:** "No worries, you don't have to share your name if you're not comfortable with it. So, what brings you here today? Is there anything in particular you'd like to know or discuss?"

Once user confidence is established, the bot seamlessly returns to data collection. For example:

**User:** "I'm having a great day!"

**Chatbot:** "That's fantastic to hear! By the way, I have my birthday party coming up in a week, and I'd like to invite you. Could you please share your email or phone number so I can send you the invite?"

Moreover, this chatbot should have the capability that it can edit user data in the save database if the user mentions during the conversation that they provided dummy data earlier and now

wish to correct or update their details, demonstrating flexibility and trustworthiness. For example:

**User:** Hey when you asked me for my name I actually gave you a fake name as I wasn't sure earlier, but now since we have talked and I have gotten to know you, you can actually call me Rahul.

**Chatbot:** Ok Rahul, sure thanks for being so transparent with me, well Rahul would you like to know more about the query you had ?..

These examples illustrate how the chatbot can engage users effectively, handle hesitations, and transition smoothly between topics to encourage data collection indirectly.

Also create a backend for the same to be used as a service with REST APIs for appropriate functions/tasks the bot will need to do.

### **Pointers for the Solution :**

- For the main conversation flow you can use the current LLMs but only the ones which are on a similar level of GPT turbo, using GPT4 and LLMs on its level is not allowed.
- Solution should have a function to combat prompt injection.
- Solution must include Named Entity Recognition (NER):
  - Implement NER to identify and extract specific pieces of information (e.g., names, dates, locations) from user responses. The extracted data should be stored in a structured format. It should also have code to check for basic validity of data for age, name, phone number, email etc.
- Solution must include one of the Following: [Either 1 OR 2 ]
  1. Sentiment Analysis: Implement sentiment analysis to understand the emotional tone of the conversation. This can help the chatbot adapt its responses based on the user's mood and provide more personalized interactions.
  2. Multi-Language Support: Enable your chatbot to handle conversations in multiple language inputs/outputs using Transformer models (No api call based models)

### **Sharing the Solution via GitHub Repository**

Create a GitHub repository to host the code. Include clear instructions on how to run the code and replicate the chat experience. Upload the code files, including the Python script(s), any necessary dependencies, and a README explaining the project.