

**NAME:Vishwa B**

**REGISTRATION NUMBER:22BCT0028**

**Career Counselor**

**Task:**

To Implement an ELIZA-like program, using substitutions. You might want to choose a different domain than a Rogerian psychologist, although keep in mind that you would need a domain in which your program can legitimately engage in a lot of simple repetition.

**Domain:** Career Counselor **Regular**

**Expressions used:**

**CODE:**

```
import re from transformers import AutoModelForCausalLM,
```

```
AutoTokenizer import torch
```

```
# Load DialoGPT-medium model_name = "microsoft/DialoGPT-
```

```
medium" tokenizer =
```

```
AutoTokenizer.from_pretrained(model_name) model =
```

```
AutoModelForCausalLM.from_pretrained(model_name)
```

```
# Conversation history chat_history_ids
```

```
= None
```

```
# Rule-based career guidance responses
```

```
career_responses = {    r"(hello|hi|hey|hii)": "Hello! How  
can I assist you today?",
```

r"(.\*looking.\*new job(.\*)": "What kind of job are you interested in? Let's work on your resume together.",

r"(.\*improve.\*resume(.\*)": "Tailor your resume to each job. Use action verbs and highlight your accomplishments.",

r"(.\*include.\*cover letter(.\*)": "Your cover letter should align your skills with the job. Be specific and concise.",

r"(.\*prepare.\*interview(.\*)": "Research the company, practice answers, and prepare questions to ask the interviewer.",

r"(.\*common.\*interview questions(.\*)": "Typical questions: 'Tell me about yourself', 'Why do you want this job?', etc.",

r"(.\*negotiate.\*salary(.\*)": "Know your worth. Use data to support your expected salary range confidently.",

r"(.\*don't.\*get.\*job offer(.\*)": "Don't be discouraged. Get feedback and improve for the next opportunity.",

r"(.\*linkedin profile(.\*)": "Use a clean photo, strong headline, detailed experiences. Update it regularly.",

r"(.\*build.\*professional network(.\*)": "Attend events, join groups, and engage on LinkedIn. Build real connections.",

r"(.\*include.\*hobbies.\*resume(.\*)": "Add hobbies only if they reflect skills like leadership, teamwork, or creativity.",

r"(.\*find.\*job openings(.\*)": "Use LinkedIn, company websites, and job portals. Stay active and consistent.",

r"(.\*skills.\*in-demand(.\*)": "Top skills include data analysis, digital marketing, Python, project management.",

r"(.\*change.\*careers(.\*)": "Identify transferable skills. Upskill as needed. Network and volunteer if possible.",

r"(.\*take.\*internship(.\*)": "Internships offer hands-on experience and networking—valuable for career growth.",

r"(.\*handle.\*job rejections(.\*)": "Rejections are redirections. Learn, adapt, and keep pushing forward.",

r"(.\*overqualified.\*job(.\*)": "Focus on how your skills can benefit the role. Show your genuine interest.",

r"(.\*explain.\*employment gaps(.\*)": "Explain positively—mention learning, volunteering, or skill-building.",

r"(.\*questions.\*ask.\*interview(.\*)": "Ask about team culture, success metrics, and future opportunities.",

r"(.\*follow up.\*interview(.\*)": "Send a thank-you email. Reaffirm your interest and mention a highlight.",

r"(.\*stand out.\*job market(.\*)": "Customize your resume, build a strong brand, and network strategically.",

r"(.\*wear.\*interview(.\*)": "Dress professionally—better to be slightly overdressed than underdressed.",

r"(.\*transition.\*freelancing.\*full-time job(.\*)": "Showcase your freelance projects and client impact.",

r"(.\*work-life balance(.\*)": "Prioritize, set boundaries, and make time for yourself. It's essential.",

r"(.\*develop.\*leadership skills(.\*)": "Lead small projects, seek feedback, and pursue leadership training.",

r"(.\*stay motivated.\*job search(.\*)": "Set goals, take breaks, and celebrate small wins. Keep at it!",

r"(.\*not sure.\*career(.\*)": "Explore interests via research, shadowing, or internships. Self-awareness helps.",

r"(.\*write.\*resignation letter(.\*)": "Keep it brief. Thank the company and offer help with the transition.",

r"(.\*offered.\*job.\*waiting.\*another offer(.\*)": "Ask for more time if needed. Be respectful and professional.",

r"(.\*company.\*right fit(.\*)": "Check their culture, values, and employee reviews. Ask the right questions.", r"(.\*first day.\*work(.\*)": "Be punctual, polite, and curious. Build rapport with your team.", r"(.\*thank you(.\*)": "You're welcome! Feel free to ask me anything.",

r"(.\*joke(.\*)": "Why don't scientists trust atoms? Because they make up everything 😊",

```
r"(.*)meaning.*life(.*)": "42. Or maybe pizza. Or purpose. Still figuring it out 🤔",  
r"(.*)successful(.*)": "Success = Effort + Patience + Growth mindset + Some coffee ☕"  
}  
  
def get_rule_response(user_input):  
    for pattern, response in career_responses.items():  
        if re.search(pattern, user_input):  
            return response  
  
    return None  
  
  
# Start chat print("🤖 Hello! I'm your Career Coach AI. Ask me anything about jobs or say  
'bye' to quit.\n")  
  
  
while True:  
    user_input = input("You: ").strip().lower()  
  
  
    if user_input in ["bye", "exit", "quit"]:  
        print("Bot: Wishing you the best in your career journey! 🖐")  
        break  
  
  
    # Check rule-based answers  
    rule_response = get_rule_response(user_input)  
  
    if rule_response:  
        print(f"Bot: {rule_response}")  
  
    continue
```

```
# AI fallback with HuggingFace model    input_ids = tokenizer.encode(user_input +  
tokenizer.eos_token, return_tensors="pt")    attention_mask =  
torch.ones_like(input_ids)  
  
if chat_history_ids is not None:  
    input_ids = torch.cat([chat_history_ids, input_ids], dim=-1)    attention_mask =  
    torch.cat([torch.ones_like(chat_history_ids), attention_mask], dim=-1)  
  
    chat_history_ids = model.generate(  
        input_ids,  
        attention_mask=attention_mask,  
        max_length=1000,  
        pad_token_id=tokenizer.eos_token_id,  
        do_sample=True,      top_p=0.95,  
        top_k=50,          temperature=0.7,  
    )  
  
    response = tokenizer.decode(chat_history_ids[:, input_ids.shape[-1]:][0],  
skip_special_tokens=True)    print(f"Bot: {response}")
```

## **Code:**

```

import re
from transformers import AutoModelForCausalLM, AutoTokenizer
import torch

# Load DialoGPT-medium
model_name = "microsoft/DialoGPT-medium"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

# Conversation history
chat_history_ids = None

# Rule-based career guidance responses
career_responses = {
    r"(hello|hi|hey|hii)": "Hello! How can I assist you today?", 
    r"(.)looking."new job(.*)": "What kind of job are you interested in? Let's work on your resume together.", 
    r"(.)improve."resume(.*)": "Tailor your resume to each job. Use action verbs and highlight your accomplishments.", 
    r"(.)include."cover letter(.*)": "Your cover letter should align your skills with the job. Be specific and concise.", 
    r"(.)prepare."interview(.*)": "Research the company, practice answers, and prepare questions to ask the interviewer.", 
    r"(.)common."interview questions(.*)": "Typical questions: 'Tell me about yourself', 'Why do you want this job?', etc.", 
    r"(.)negotiate."salary(.*)": "Know your worth. Use data to support your expected salary range confidently.", 
    r"(.)don't."get job offer(.*)": "Don't be discouraged. Get feedback and improve for the next opportunity.", 
    r"(.)linkedin profile(.*)": "Use a clean photo, strong headline, detailed experiences. Update it regularly.", 
    r"(.)build."professional network(.*)": "Attend events, join groups, and engage on LinkedIn. Build real connections.", 
    r"(.)include."hobbies."resume(.*)": "Add hobbies only if they reflect skills like leadership, teamwork, or creativity.", 
    r"(.)find."job openings(.*)": "Use LinkedIn, company websites, and job portals. Stay active and consistent.", 
    r"(.)skills."in-demand(.*)": "Top skills include data analysis, digital marketing, Python, project management.", 
    r"(.)change."careers(.*)": "Identify transferable skills. Upskill as needed. Network and volunteer if possible.", 
    r"(.)take."internship(.*)": "Internships offer hands-on experience and networking—valuable for career growth.", 
    r"(.)handle."job rejections(.*)": "Rejections are redirections. Learn, adapt, and keep pushing forward.", 
    r"(.)overqualified."job(.*)": "Focus on how your skills can benefit the role. Show your genuine interest.", 
    r"(.)explain."employment gaps(.*)": "Explain positively—mention learning, volunteering, or skill-building.", 
    r"(.)questions."ask interview(.*)": "Ask about team culture, success metrics, and future opportunities.", 
    r"(.)follow up."interview(.*)": "Send a thank-you email. Reaffirm your interest and mention a highlight.", 
    r"(.)stand out."job market(.*)": "Customize your resume, build a strong brand, and network strategically.", 
    r"(.)wear."interview(.*)": "Dress professionally—better to be slightly overdressed than underdressed."
}

r"(you're wrong|you erroneous)": "You're encouraged to review and improve for the next opportunity.", 
r"(.)linkedin profile(.*)": "Use a clean photo, strong headline, detailed experiences. Update it regularly.", 
r"(.)build."professional network(.*)": "Attend events, join groups, and engage on LinkedIn. Build real connections.", 
r"(.)include."hobbies."resume(.*)": "Add hobbies only if they reflect skills like leadership, teamwork, or creativity.", 
r"(.)find."job openings(.*)": "Use LinkedIn, company websites, and job portals. Stay active and consistent.", 
r"(.)skills."in-demand(.*)": "Top skills include data analysis, digital marketing, Python, project management.", 
r"(.)change."careers(.*)": "Identify transferable skills. Upskill as needed. Network and volunteer if possible.", 
r"(.)take."internship(.*)": "Internships offer hands-on experience and networking—valuable for career growth.", 
r"(.)handle."job rejections(.*)": "Rejections are redirections. Learn, adapt, and keep pushing forward.", 
r"(.)overqualified."job(.*)": "Focus on how your skills can benefit the role. Show your genuine interest.", 
r"(.)explain."employment gaps(.*)": "Explain positively—mention learning, volunteering, or skill-building.", 
r"(.)questions."ask interview(.*)": "Ask about team culture, success metrics, and future opportunities.", 
r"(.)follow up."interview(.*)": "Send a thank-you email. Reaffirm your interest and mention a highlight.", 
r"(.)stand out."job market(.*)": "Customize your resume, build a strong brand, and network strategically.", 
r"(.)wear."interview(.*)": "Dress professionally—better to be slightly overdressed than underdressed.", 
r"(.)transition."freelancing|full-time job(.*)": "Showcase your freelance projects and client impact.", 
r"(.)work-life balance(.*)": "Prioritize, set boundaries, and make time for yourself. It's essential.", 
r"(.)develop."leadership skills(.*)": "Lead small projects, seek feedback, and pursue leadership training.", 
r"(.)stay motivated."job search(.*)": "Set goals, take breaks, and celebrate small wins. Keep at it!", 
r"(.)not sure."career(.*)": "Explore interests via research, shadowing, or internships. Self-awareness helps.", 
r"(.)write."resignation letter(.*)": "Keep it brief. Thank the company for their help with the transition.", 
r"(.)offered."job waiting."another offer(.*)": "Ask for more time if loading. Be respectful and professional.", 
r"(.)company."right fit(.*)": "Check their culture, values, and employee reviews. Ask the right questions.", 
r"(.)first day."work(.*)": "Be punctual, polite, and curious. Build rapport with your team.", 
r"(.)thank you(.*)": "You're welcome! Feel free to ask me anything.", 
r"(.)joke(.*)": "Why don't scientists trust atoms? Because they make up everything 😊.", 
r"(.)meaning."life(.*)": "A2. Or maybe pizza. Or purpose. Still figuring it out 🍕.", 
r"(.)successful(.*)": "Success = Effort + Patience + Growth mindset + Some coffee ☕."
}

```

```

def get_rule_response(user_input):
    for pattern, response in career_responses.items():
        if re.search(pattern, user_input):
            return response
    return None

# Start chat
print("👋 Hello! I'm your Career Coach AI. Ask me anything about jobs or say 'bye' to quit.\n")

while True:
    user_input = input("You: ").strip().lower()

    if user_input in ["bye", "exit", "quit"]:
        print("Bot: Wishing you the best in your career journey! 🌟")
        break

    # Check rule-based answers
    rule_response = get_rule_response(user_input)
    if rule_response:
        print(f"Bot: {rule_response}")
        continue

    # AI fallback with HuggingFace model
    input_ids = tokenizer.encode(user_input + tokenizer.eos_token, return_tensors="pt")
    attention_mask = torch.ones_like(input_ids)

    if chat_history_ids is not None:
        input_ids = torch.cat([chat_history_ids, input_ids], dim=-1)
        attention_mask = torch.cat([torch.ones_like(chat_history_ids), attention_mask], dim=-1)

    chat_history_ids = model.generate(
        input_ids,
        attention_mask=attention_mask,
        max_length=1000,
        pad_token_id=tokenizer.eos_token_id,
        do_sample=True,
    )

    response = tokenizer.decode(chat_history_ids[:, input_ids.shape[-1]:][0], skip_special_tokens=True)
    print(f"Bot: {response}")

```

## Output:

You: Hi  
Bot: Hello! How can I assist you today?  
You: I'm looking for a new job  
Bot: What kind of job are you interested in? Let's work on your resume together.  
You: how to write a good resume?  
Bot: Be attractive and funny.  
You: what are some common interview questions?  
Bot: Typical questions: 'Tell me about yourself', 'Why do you want this job?', etc.  
You: tell me a joke  
Bot: Why don't scientists trust atoms? Because they make up everything 😊  
You: what is the meaning of life?  
Bot: 42. Or maybe pizza. Or purpose. Still figuring it out 🍕  
You: how do I become successful?  
Bot: Success = Effort + Patience + Growth mindset + Some coffee ☕  
You: thank you  
Bot: You're welcome! Feel free to ask me anything.  
You: Bye  
Bot: Wishing you the best in your career journey! 🙌

```
response = tokenizer.decode(chat_history_ids[:, input_ids.shape[-1]:][0], skip_special_tokens=True)
print(f"Bot: {response}")

...
generation_config.json: 100% [██████████] 124/124 [00:00<00:00, 3.29kB/s]
🤖 Hello! I'm your Career Coach AI. Ask me anything about jobs or say 'bye' to quit.

You: Hi
Bot: Hello! How can I assist you today?
You: I'm looking for a new job
Bot: What kind of job are you interested in? Let's work on your resume together.
You: How do I improve my resume?
Bot: Tailor your resume to each job. Use action verbs and highlight your accomplishments.
You: Should I include a cover letter?
Bot: Your cover letter should align your skills with the job. Be specific and concise.
You: How to prepare for interviews?
Bot: Research the company, practice answers, and prepare questions to ask the interviewer.
You: What are common interview questions?
Bot: Typical questions: 'Tell me about yourself', 'Why do you want this job?', etc.
You: How to negotiate salary?
Bot: Know your worth. Use data to support your expected salary range confidently.
You: I didn't get the job offer
Bot: I can still help, pm me.
You: How to build a good LinkedIn profile?
Bot: Use a clean photo, strong headline, detailed experiences. Update it regularly.
You: How to build my professional network?
Bot: Attend events, join groups, and engage on LinkedIn. Build real connections.
You: Should I include hobbies in my resume?
Bot: Add hobbies only if they reflect skills like leadership, teamwork, or creativity.
You: Where to find job openings?
Bot: Use LinkedIn, company websites, and job portals. Stay active and consistent.
You: What skills are in-demand?
Bot: Top skills include data analysis, digital marketing, Python, project management.
You: I want to change careers
Bot: Identify transferable skills. Upskill as needed. Network and volunteer if possible.
You: Should I take an internship?
Bot: Hello! How can I assist you today?
You: How to handle job rejections?
Bot: Rejections are redirections. Learn, adapt, and keep pushing forward.
You: I'm overqualified for a job
Bot: Focus on how your skills can benefit the role. Show your genuine interest.
You: How to explain employment gaps?
Bot: Explain positively-mention learning, volunteering, or skill-building.
You: What questions should I ask in an interview?
Bot: Ask about team culture, success metrics, and future opportunities.
You: How to follow up after an interview?
Bot: Send a thank-you email. Reaffirm your interest and mention a highlight.
You: How to stand out in the job market?
Bot: Customize your resume, build a strong brand, and network strategically.
You: What to wear for an interview?
Bot: Dress professionally-better to be slightly overdressed than underdressed.
You: How to transition from freelancing to full-time?
Bot: You can't. That's why we need to hire fulltime guys.
You: How to maintain work-life balance?
Bot: Prioritize, set boundaries, and make time for yourself. It's essential.
You: How to develop leadership skills?
Bot: Hello! How can I assist you today?
You: How to stay motivated during job search?
Bot: Set goals, take breaks, and celebrate small wins. Keep at it!
You: I'm not sure which career is right for me
Bot: Hello! How can I assist you today?
```



Untitled5.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

You: How to handle job rejections?  
Bot: Rejections are redirections. Learn, adapt, and keep pushing forward.  
... You: I'm overqualified for a job  
Bot: Focus on how your skills can benefit the role. Show your genuine interest.  
You: How to explain employment gaps?  
Bot: Explain positively—mention learning, volunteering, or skill-building.  
<> You: What questions should I ask in an interview?  
Bot: Ask about team culture, success metrics, and future opportunities.  
{x} You: How to follow up after an interview?  
Bot: Send a thank-you email. Reaffirm your interest and mention a highlight.  
You: How to stand out in the job market?  
Bot: Customize your resume, build a strong brand, and network strategically.  
You: What to wear for an interview?  
Bot: Dress professionally—better to be slightly overdressed than underdressed.  
□ You: How to transition from freelancing to full-time?  
Bot: You can't. That's why we need to hire fulltime guys.  
You: How to maintain work-life balance?  
Bot: Prioritize, set boundaries, and make time for yourself. It's essential.  
You: How to develop leadership skills?  
Bot: Hello! How can I assist you today?  
You: How to stay motivated during job search?  
Bot: Set goals, take breaks, and celebrate small wins. Keep at it!  
You: I'm not sure which career is right for me  
Bot: Hello! How can I assist you today?  
You: How to write a resignation letter?  
Bot: Keep it brief. Thank the company and offer help with the transition.  
You: I got one offer but waiting for another  
Bot: PM me  
You: How to know if a company is the right fit?  
Bot: Check their culture, values, and employee reviews. Ask the right questions.  
You: What to do on the first day of work?  
Bot: Be punctual, polite, and curious. Build rapport with your team.  
You: Tell me a joke  
Bot: Why don't scientists trust atoms? Because they make up everything 😊  
You: What's the meaning of life?  
Bot: 42. Or maybe pizza. Or purpose. Still figuring it out 🍕  
You: How to be successful?  
Bot: Success = Effort + Patience + Growth mindset + Some coffee ☕  
You: What's your name?  
Bot: I got a job offer for full time.  
You: What's the weather today?  
Bot: It's cold, a lot of people work late.  
You: Do you believe in destiny?  
Bot: Nah, I do not.  
You: Tell me about yourself  
Bot: It's cold.  
You: Can you help me find happiness?  
Bot: Thanks  
You: What's your favorite color?  
Bot: I will  
You: Who is the president of the US?  
Bot: that's my name  
You: