*Chatbot (project-chat) code and output..*

|  |
| --- |
| *Chatbot code:*  import random  import json  import datetime  RESPONSES\_FILE = "responses.json"  # Load responses from a JSON file  def load\_responses():      """ Load responses.json without this file program only reply in default responses which doesn't provide info"""      try:          with open(RESPONSES\_FILE, "r") as file:              return json.load(file)      except FileNotFoundError:          print(f"Error: {RESPONSES\_FILE} not found.")          return {"default": ["I'm not sure how to respond to that."]}  def save\_chat\_log(log, user\_name, agent\_name):      """ Save conversation between user and agent in separate .txt file with time"""      log\_filename = f"chat\_log\_{datetime.datetime.now().strftime('%Y\_%m\_%d\_%H\_%M\_%S')}.txt"      with open(log\_filename, "w") as file:          for entry in log:              file.write(f"{user\_name} (User): {entry['user']}\n{agent\_name} (Agent): {entry['agent']}\n\n")  def get\_user\_name():      """ Ask name to the user if name doen't given then show error """      while True:          user\_name = input("Welcome to Poppleton University Chat! What's your name? ").strip()          if user\_name:              return user\_name.capitalize() # Capitalize the first letter of the user\_name if user type apeal it convert to Apeal          print("Please enter a name to continue.")  def toggle\_listening\_mode(current\_mode):      """ For listening mode user don't need reply from the program """      new\_mode = not current\_mode      mode\_status = "ON" if new\_mode else "OFF"      print(f"Listening mode is now {mode\_status}.")      return new\_mode  def handle\_blank\_input(blank\_count):      """ auto disconnect if user give more than 3 empty reply """      blank\_count += 1      if blank\_count >= 3:          print("Oops! The agent has disconnected. Please try again later.")          return blank\_count, True      return blank\_count, False  def get\_agent\_response(user\_input, responses, user\_name, agent\_name):      """ Reply from agent and in responses add user\_name and agent\_name in reply """      lower\_input = user\_input.lower()      for keyword, replies in responses.items():          if keyword in lower\_input:              return random.choice(replies).replace("{name}", user\_name).replace("{agent\_name}", agent\_name) # name in responses      return random.choice(responses["default"]).replace("{name}", user\_name) # if keywords doesn't found than reply from default  def chatbot():      """ This part of code for main code """      responses = load\_responses()      agent\_names = ["Steve Smith", "Rohit Sharma", "Travis Head", "Virat Kohli", "MS Dhoni"]      random\_disconnect\_chance = 0.1      user\_name = get\_user\_name()      agent\_name = random.choice(agent\_names)      print(f"Hello, {user\_name}! My name is {agent\_name}. How can I assist you today?")      print("You can type 'listening mode' to enter a mode where I will not respond to your inputs.")      chat\_log = []      blank\_input\_count = 0      listening\_mode = False      while True:          user\_input = input(f"{user\_name}: ").strip()          if user\_input.lower() in ["bye", "quit", "exit"]:              print("Thank you for chatting with us! Have a great day!")              break          if user\_input.lower() == "listening mode":              listening\_mode = toggle\_listening\_mode(listening\_mode)              continue          if listening\_mode:              print("Agent is listening...")              chat\_log.append({"user": user\_input, "agent": "Agent is listening..."})              continue          if not user\_input:              blank\_input\_count, should\_disconnect = handle\_blank\_input(blank\_input\_count)              if should\_disconnect:                  break          else:              blank\_input\_count = 0          if random.random() < random\_disconnect\_chance:              print("Oops! The agent has disconnected. Please try again later.")              break          response = get\_agent\_response(user\_input, responses, user\_name, agent\_name)          print(f"Agent ({agent\_name}): {response}")          chat\_log.append({"user": user\_input, "agent": response})      save\_chat\_log(chat\_log, user\_name, agent\_name)  if \_\_name\_\_ == "\_\_main\_\_":      chatbot() |

|  |
| --- |
| *responses.json:*  *{*  *"hello": [*  *"Hi there, {name}! How can I assist you today?",*  *"Hello, {name}! How's it going?",*  *"Greetings, {name}! What can I do for you today?"*  *],*  *"course": [*  *"We offer many courses, {name}. Can you specify your area of interest?",*  *"{name}, are you looking for undergraduate or postgraduate courses?",*  *"Our university has courses ranging from arts to sciences, {name}. Need help deciding?"*  *],*  *"undergraduate": [*  *"Poppleton University offers a variety of undergraduate programs, including BSc Computing, Business Management, and Engineering.",*  *"Explore all undergraduate courses at Poppleton University on our official website."*  *],*  *"postgraduate": [*  *"At Poppleton University, we offer postgraduate programs like MSc Computing, MBA, and Data Science.",*  *"Looking to advance your career, {name}? Our postgraduate programs are designed for future leaders.",*  *"Visit Poppleton University's website to explore all our postgraduate courses."*  *],*  *"admissions": [*  *"Admissions office hours are 10 AM to 4 PM.",*  *"You can contact admissions at admissions@poppleton.ac.uk, {name}.",*  *"Admissions are located in the main administration building."*  *],*  *"fees": [*  *"The tuition fee details are available on the university website.",*  *"Fee payments can be made online or at the finance office, {name}.",*  *"Late fee submissions will incur a penalty of 5%."*  *],*  *"fee structure": [*  *"For graduate courses, the fee is around 1 lakh 70 thousands per semester, {name}.",*  *"You can find a detailed breakdown of the fee structure on the university website, {name}."*  *],*  *"scholarships": [*  *"Scholarship applications are open from January to March.",*  *"You can apply for need-based and merit-based scholarships, {name}.",*  *"Visit the scholarships office for assistance in applying."*  *],*  *"library": [*  *"The library opens at 9 AM every day.",*  *"Feel free to visit the library for quiet study spaces, {name}.",*  *"You can borrow books from the library."*  *],*  *"sports": [*  *"The sports complex is open from 6 AM to 10 PM.",*  *"You can join the football team by registering at the ECA, {name}.",*  *"Basketball courts are available."*  *],*  *"events": [*  *"Check the university website for upcoming events.",*  *"There's a music concert this Friday at the auditorium, {name}.",*  *"The science fair is happening next month."*  *],*  *"housing": [*  *"On-campus housing applications open in March.",*  *"Visit the housing office for more details, {name}.",*  *"Housing queries can be emailed to  poppleton2025.ac.uk."*  *],*  *"exams": [*  *"Exam schedules are posted on the student portal.",*  *"Contact your department for exam-related queries, {name}.",*  *"Make sure to bring your student ID to exams."*  *],*  *"transport": [*  *"The campus shuttle runs every 20 minutes.",*  *"You can find the bus schedule on the university website, {name}.",*  *"Parking permits are available from the transport office."*  *],*  *"clubs": [*  *"The debate club meets every Wednesday at 5 PM.",*  *"You can join the chess club by signing up online, {name}.",*  *"The photography club is hosting a workshop this weekend."*  *],*  *"financial\_aid": [*  *"Financial aid applications can be submitted online.",*  *"Visit the financial aid office for consultation, {name}.",*  *"Financial aid decisions are communicated within 4 weeks."*  *],*  *"campus": [*  *"The campus map is available on the university website.",*  *"Feel free to explore the botanical garden near the library, {name}.",*  *"Campus tours are conducted every Friday at 2 PM."*  *],*  *"support": [*  *"The IT support desk is available 24/7.",*  *"Counseling services are located in the health center, {name}.",*  *"Academic advisors are available by appointment."*  *],*  *"joke": [*  *"Why don't scientists trust atoms? Because they make up everything!",*  *"Why was the math book sad? It had too many problems.",*  *"What do you call fake spaghetti? An impasta!"*  *],*  *"cricket": [*  *"Fun fact: {agent\_name} once scored a century in T20 format!",*  *"Did you know? Cricket is played in over 100 countries worldwide.",*  *"{agent\_name} thinks cricket is the most exciting sport ever!"*  *],*  *"coffee": [*  *"The coffee bar is open from 8 AM to 8 PM.",*  *"You can grab coffee at the main building cafeteria, {name}.",*  *"Coffee is available near the library entrance."*  *],*  *"default": [*  *"I'm sorry, {name}. I didn't quite understand that. Can you rephrase?",*  *"{name}, could you provide more details? I'd be happy to help.",*  *"Apologies, {name}. I didn't catch that. Can you clarify?",*  *"Let me check on that for you, {name}."*  *]*  *}* |

|  |
| --- |
| Output: |