Name :- Arpit Gharadiya

Branch :- Computer science and engineering

State :- Gujarat

ChatPy – Python Chatbot Code Explanation

➔ **Explanation**

👉 This document explains the **ChatPy Chatbot** developed using **Python's Tkinter library** for GUI and datetime for time/date-based responses.

➔ **ChatPy – Python Chatbot Code Explanation**

➔ **Explanation**

👉 This document explains the **ChatPy Chatbot** developed using **Python's Tkinter library** for GUI and datetime for time/date-based responses.

➢ **import tkinter as tk**

➤ Imports the **Tkinter library** which is used for creating GUI applications in Python.

➢ **from tkinter import scrolledtext**

➤ Imports the **ScrolledText widget**, which is a text box with scrollbars.

➢ **import datetime**

➤ Imports the **datetime module** to fetch the current system date and time.

➢ **responses = {...}**

➤ A **dictionary** of static responses where keys are user inputs (like "hello", "time", etc.) and values are chatbot replies.

➤ "time" and "date" keys dynamically use datetime.datetime.now().strftime(...) to show live time and date.

➢ **def get_response(msg):**

➤ Defines a function that processes user input and returns an appropriate response.

Inside the function:

✓ msg.lower().strip() — Converts user input to lowercase and removes extra spaces for easier matching.

✓ Checks if input contains "my name is" to extract and remember the user's name dynamically.

✓ Returns predefined response if the message exists in the responses dictionary.

✓ Else, returns "I don't understand that. Type 'help' for options."

➢ **def send():**

➤ Function that is called when the user clicks the **Send** button.

Inside the function:

✓ Gets input from the entry box.

✓ If input is empty, does nothing.

✓ Inserts the user input and the chatbot's response into the chat area.

✓ Clears the entry box after sending.

➢ **root = tk.Tk()**

➤ Creates the main application window.

➢ **root.title(...)**

➤ Sets the **title of the application window** as “ChatPy - Python Chatbot”.

➢ **root.geometry("500x500")**

➤ Defines the **window size** to 500 pixels by 500 pixels.

➢ **chat_area = scrolledtext.ScrolledText(...)**

➤ Adds a scrollable multi-line text area for displaying the conversation.

✓ Uses wrap=tk.WORD, sets font, width, and height.

✓ Starts with a greeting from the bot.

➢ **chat_area.config(state='normal')**

➤ Enables the user to insert new messages into the chat area.

➢ **frame = tk.Frame(root)**

➤ Creates a frame to hold the entry box and send button.

➢ **entry_box = tk.Entry(...)**

➤ Adds an **entry box** for the user to type messages.

➢ **send_button = tk.Button(...)**

➤ Adds a **Send** button that triggers the send() function.

➢ **root.mainloop()**

➤ Starts the **Tkinter event loop** which keeps the window running and responsive.

✓ **Features of ChatPy**

→Replies to greetings like “hello”, “hi”.

→Understands basic queries like “what is python”, “who are you”.

→Provides live **date and time**.

→Can respond to **custom input** like “my name is John”.

→Shows conversation history in a scrollable chat window.

→ Clean GUI with entry box and send button.