

A mini Project Report on
AI PSYCHOLOGICAL CHATBOT

Submitted for partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF BUSINESS ADMINISTRATION IN
ARTIFICIAL INTELLEGENCE**

BY

GAURAV KUMAR	25030422050
AANYA ARNEY	25030422003
KARPI RIBA	25030422057

UNDER THE GUIDANCE OF

Dr. Dawa Chyophel Lepcha



Symbiosis Artificial Intelligence Institute



Certificate

This is to certify that the project work entitled "**AI
PSYCHOLOGICAL CHATBOT**" that is submitted by

GAURAV KUMAR	25030422050
AANYA ARNEY	25030422003
KARPI RIBA	25030422057

In partial fulfillment for the award of the degree of BACHELOR OF BUSINESS ADMINISTRATION IN ARTIFICIAL INTELLIGENCE is a record of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

Declaration

We **GAURAV , AANYA** and **KARPI**, hereby declare that the report of the Mini Project work entitled "**AI PSYCHOLOGICAL CHATBOT**" which is being submitted to the **Symbiosis Artificial Intelligence Institute**, in partial fulfillment of the requirement for the award of the Degree of **BACHELOR OF BUSINESS ADMINISTRATION IN ARTIFICIAL INTELLIGENCE**, is a bonafide record of the work carried out by us. The material contained in this report has not been submitted to any other University or Institution for the award of any degree.

Place: **Lavale,Pune**

Date: 4th November 2025

(GAURAV KUMAR - 25030422044)

(AANYA - 25030422003)

(KARPI - 25030422057)

ABSTRACT

The growing demand for accessible, empathetic, and affordable mental health support has encouraged the creation of conversational agents that provide basic emotional assistance. This mini-project presents a *Psychological Chatbot* built entirely in Python, designed to offer users non-judgmental, supportive dialogue for managing mild stress and anxiety. Using simple Natural Language Processing (NLP) techniques such as keyword matching and rule-based response selection, the chatbot identifies emotional cues in user input and responds with contextually appropriate, empathetic messages. The system runs as a command-line interface (CLI) application, utilizing Python's built-in libraries without requiring external dependencies. Focused on simplicity, reliability, and accessibility, this project demonstrates practical skills in NLP fundamentals, Python programming, and the development of text-based interactive systems for mental wellness support.

Keywords: Python, Chatbot, NLP, Emotional Support, Mental Health, Rule-Based System, CLI Application.

Introduction

In today's fast-paced world, stress, anxiety, and emotional problems are common, but stigma, lack of access to therapists, or financial constraints may make people reluctant to seek professional help. As a result, many people do not get timely emotional support, which can lead to a decline in their mental health.

By creating a psychological chatbot, a conversational Python application intended to offer basic emotional support, stress relief, and mental health awareness, this

project seeks to address this problem. Simple natural language processing (NLP) techniques are used by the chatbot to comprehend user input and provide sympathetic, helpful messages or recommendations.

Existing Solutions

These are a few of the most well-known and widely used AI chatbots for psychologists currently in use:

1. Woebot: employs cognitive behavioral strategies. helps users manage their anxiety and depression symptoms by providing them with skill-building activities, mood monitoring, and structured, therapeutic dialogues.
2. Wysa: An amiable penguin is frequently used to personify the AI chatbot. It offers journaling, meditation, and guided self-help

tools. For more thorough care, users can also choose to connect with human therapists or coaches.

3. Youper: It promotes mood monitoring and offers individualized analysis and cognitive behavioral therapy exercises based on the user's emotional tendencies.

Methodology

1. Programming Tools (Built-in)

Tool/Concept	Purpose in the Code
Python	The core programming language used to define the logic, data structure, and execution environment.
import random	A standard Python module used to select a response randomly from the list of available phrases for any given mood or prompt category.
Dictionary (responses)	The data structure used to organize the pre-written responses, mapping keyword categories (like

Tool/Concept	Purpose in the Code
	"sad", "happy", "hi") to lists of possible replies.
Conditional Logic (if/elif)	Used inside the get_response function to determine which category the user's input falls into by checking if certain keywords are present.
Command Line Interface (CLI)	The use of input() and print() functions means the application runs directly in a console or terminal, without a graphical user interface (GUI).

2. Psychological/NLP Technique

Technique	Description
Keyword Matching /	This is the method the bot uses to "understand" the

Technique	Description
Pattern Recognition	<p>user. It simply checks if specific words ("sad", "angry", "hello") exist in the input string. This is the simplest, most fundamental technique for conversation in a chatbot.</p>
Rule-Based System	<p>The entire application follows a fixed set of rules (if 'sad' in input, then choose a 'sad' response). The responses are pre-defined and the bot cannot generate new, creative text outside of the dictionary.</p>

Implementation

The screenshot shows a code editor window with a dark theme. The file being edited is `ai_pysch.py`. The code defines a function `get_response` that takes user input and returns a random response based on keywords like "hi", "sad", "happy", "angry", or "mad". It also handles exit commands ("bye", "exit", "quit"). The main loop reads user input, calls `get_response`, and prints the AI's response. The terminal below shows the script running and interacting with the user.

```
ai_pysch.py X
D: > SAI STUDY > Python > mini bot project > ai_pysch.py > ...
30
31 def get_response(user_input):
32     user_input = user_input.lower()
33     if "hi" in user_input or "hello" in user_input or "hey" in user_input:
34         return random.choice(responses["hi"])
35     elif "sad" in user_input:
36         return random.choice(responses["sad"])
37     elif "happy" in user_input:
38         return random.choice(responses["happy"])
39     elif "angry" in user_input or "mad" in user_input:
40         return random.choice(responses["angry"])
41     else:
42         return random.choice(responses["default"])
43
44 print("AI Psychologist: Hi! I'm your AI psychologist. How are you feeling today?")
45 while True:
46     user_input = input("You: ")
47     if user_input.lower() in ["bye", "exit", "quit"]:
48         print("AI Psychologist: Take care of yourself. Goodbye!")
49         break
50     response = get_response(user_input)
51     print("AI Psychologist:", response)
52

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\anya> & C:/Users/anya/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/SAI STUDY/Python/mini bot project/ai_pysch.py"
AI Psychologist: Hi! I'm your AI psychologist. How are you feeling today?
You: HI
AI Psychologist: Hello! How are you feeling today?
You: bad
AI Psychologist: I see. Tell me more about that.
You: im sad
AI Psychologist: I'm sorry you're feeling sad. Want to talk about what's bothering you?
You: no
AI Psychologist: Hmm, can you explain that a bit more?
You: 
```

This Python code creates a basic, rule-based chatbot that simulates a dialogue with a psychologist.

Here's a brief summary:

Dictionary of Responses: It has a default response as well as a collection of pre-written responses arranged by mood (hi, sad, happy, and angry).

acquire_response This function looks for words like "sad," "happy," or "angry" in the user's input.

Random Selection: The bot utilizes the random module to choose one of the potential responses for a given mood after locating a keyword.

Conversation Loop: Occasionally The true loop repeatedly takes user input and prints the bot's response in the command line until the user types a command like "bye" or "quit."

Because the bot operates solely on keyword matching and pre-established rules, it is simple.

Result

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\anya> & C:/Users/anya/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/SAII STUDY/Python/mini bot project/ai_pysch.py"
AI Psychologist: Hi! I'm your AI psychologist. How are you feeling today?
You: hi
AI Psychologist: Hello! How are you feeling today?
You: sad
AI Psychologist: That sounds tough. What usually helps you feel better?
You: movies
AI Psychologist: Hmm, can you explain that a bit more?
You: |
```

The chatbot recognizes emotions from user input and extends a warm greeting.

- Asks follow-up questions and responds with empathy.

- Preserves an organic dialogue loop

Your AI psychologist chatbot is operating correctly, as evidenced by this output, which reads user input, interprets it, and provides contextually relevant answers.

Conclusion

The chatbot recognizes emotions from user input and extends a warm greeting.

- Asks follow-up questions and responds with empathy.
- Preserves an organic dialogue loop

Your AI psychologist chatbot is operating correctly, as evidenced by this output, which reads user input, interprets it, and provides contextually relevant answers.

Refferences

The programming language utilized, Python 3.11, was one of the tools utilized.

- VS Code – the integrated development environment (IDE) for writing and running your Python code.
- Command Line / Terminal – where you interact with the chatbot.

Thank
You