# TITLE:- BUILDING A BASIC AI APPLICATION USING PYTHON

#### **OVERVIEW: -**

The chatbot is designed to act as a communicative agent, replying to user input according to predefined rules and patterns. To improve its understanding of user queries, it uses basic text preprocessing techniques including tokenization and stopword removal.

### **CODE STRUCTURE: -**

The code is divided into multiple functions:

preprocess\_text(text): Takes relevant tokens by tokenizing the input text, converting it to lowercase, and removing stop words.

greet(): Says "hi" to the user when they start talking.

get user name(): Asks the user to provide their name.

respond\_to\_input(user\_input): Based on preset patterns, this function analyzes the preprocessed input and produces a response.

Chat(): Until the user selects to end the chat, it greets them, asks for their name, processes their inputs, and responds.

### **FUNCTIONALITY: -**

When the user initiates the chat, they are welcomed and asked to provide their name. The chatbot reacts to responses from users by using particular keywords and patterns. For example, when asked which US state is the best, it responds with a predetermined message, says "hi" and "bye" when prompted. Until the user decides to end the conversation by typing "exit," it will continue.

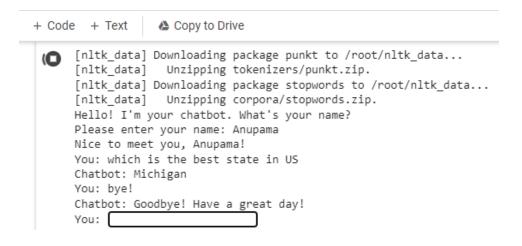
## CODE(part-1): -

```
# Variables
name = "John"
       age = 25
       is_student = True
       # Print variables
       print("Name:", name)
print("Age:", age)
       print("Is student?", is_student)
       # Conditional statement
       if age < 18:
           print("You are a minor.")
       elif age >= 18 and age < 60:
          print("You are an adult.")
           print("You are a senior citizen.")
       # Lists
       fruits = ["apple", "orange", "banana", "grape"]
       # Loop through the list and print each fruit
       print("Fruits:
       for fruit in fruits:
          print(fruit)
       # Add a new fruit to the list
       fruits.append("kiwi")
       print("Updated Fruits:", fruits)
 + Code + Text
                  Copy to Drive
√ Dictionaries
        person = {"name": "Alice", "age": 30, "is_student": False}
        # Access and print values from the dictionary
        print("\nPerson Information:")
        print("Name:", person["name"])
        print("Age:", person["age"])
        print("Is student?", person["is_student"])
        # Update a value in the dictionary
        person["age"] = 32
        print("Updated Age:", person["age"])
        # Loop through the dictionary and print key-value pairs
        print("\nPerson Dictionary:")
        for key, value in person.items():
            print(f"{key}: {value}")
```

### **OUTPUT: -**

# CODE(part-2): -

# **OUTPUT:** -



### **CONCLUSION: -**

In conclusion, the chatbot uses NLTK for text processing in order to converse with users on the most basic level. However it is not dependent on context or has a wide range of responses. The accuracy of it can be increased with improvements like error management and a variety of responses, which may create the basis for interactive bots with greater complexity.