

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): -

```
+ Code + Text | Copy to Drive

✓ 0s ▶ # 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.")
else:
    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

✓ 0s ▶ # 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 + Text Copy to Drive
✓
01 Name: John
Age: 25
Is student? True
You are an adult.
Fruits:
apple
orange
banana
grape
Updated Fruits: ['apple', 'orange', 'banana', 'grape', 'kiwi']

Person Information:
Name: Alice
Age: 30
Is student? False
Updated Age: 32

Person Dictionary:
name: Alice
age: 32
is_student: False
```

CODE(part-2): -

```
+ Code + Text Copy to Drive
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

nltk.download('punkt')
nltk.download('stopwords')

def preprocess_text(text):
    # Tokenize the input text
    tokens = word_tokenize(text.lower())

    # Remove stopwords (common words that usually don't carry much meaning)
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [word for word in tokens if word.isalnum() and word not in stop_words]

    return filtered_tokens

def greet():
    print("Hello! I'm your chatbot. What's your name?")

def get_user_name():
    return input("Please enter your name: ")

def respond_to_input(user_input):
    preprocessed_input = preprocess_text(user_input)

    if any(word in preprocessed_input for word in ['hello', 'hi']):
        return "Hello there! How can I help you?"
    elif any(word in preprocessed_input for word in ['which', 'is', 'the', 'best', 'state', 'us']):
```

```
+ Code + Text Copy to Drive
    elif any(word in preprocessed_input for word in ['hello', 'hi']):
        return "Hello there! How can I help you?"
    elif any(word in preprocessed_input for word in ['which', 'is', 'the', 'best', 'state', 'us']):
        return "Which?"
    elif any(word in preprocessed_input for word in ['bye', 'goodbye']):
        return "Goodbye! Have a great day!"
    else:
        return "I'm not sure how to respond to that. Can you ask me something else?"

def chat():
    greet()
    user_name = get_user_name()
    print(f"Nice to meet you, {user_name}!")

    while True:
        user_input = input("You: ")


        if user_input.lower() == "exit":
            print("Chatbot: Goodbye! Have a great day!")
            break

        response = respond_to_input(user_input)
        print("Chatbot:", response)

if __name__ == "__main__":
    chat()
```

OUTPUT: -

+ Code + Text Copy to Drive

 [nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
Hello! I'm your chatbot. What's your name?
Please enter your name: Anupama
Nice to meet you, Anupama!
You: which is the best state in US
Chatbot: Michigan
You: bye!
Chatbot: Goodbye! Have a great day!
You:

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.