# CREATEACHATBOTINPYTHONTEAMMEMBER810621104012: KAVIYARASAN.G

**PHASE-3 SUBMISSION DOCUMENTPROJECTTITLE:CREATEA CHATBOTINPYTHON**

# PHASE3:Developmentpart1

**Topic: Start building the create a chatbot in python model byusingloadingandpre-processingthedataset.**



**INTRODUCTION:**

In recent years, the adoption and use cases of chatbots have been onthe rise. With advancements in Natural Language Processing (NLP)and the introduction of models like ChatGPT, chatbots have becomeincreasinglypopularandpowerfultoolsforautomatingconversations.

In this article, we will explore the process of creating a simple chatbotusing Python and NLP techniques. Whether you’re interested inbuilding a virtual assistant, customer support bot, or simply want toexplore the fascinating world of chatbots, this article will guide youthroughthestepsofcreatingyourown.So,let’sdiveinandharnessthepotentialofchatbotsinthiseraoftechnologicaladvancement.

PreprocessingUserInput:

Beforewecanprocessuserinput,weneedtopreprocessitby:

* **Tokenization**:theprocessofbreakingdownasentenceortextintoindividualwordsortokens.Inthisstep,weuse

thenltk.word\_tokenize()functionfromtheNLTKlibrarytosplittheuserinputintoalistoftokens.Forexample,thesentence*"Howare*

*youdoingtoday?"*wouldbetokenizedinto['How','are','you','doing','today','?'].

1. **Lowercasing:**toensureconsistencyandremovecasesensitivity,we convert all tokens to lowercase using the lower()method. Thishelps in matching words regardless of whether they are inuppercaseorlowercase.
2. **Lemmatization**:itreduceswordstotheirbaseordictionaryform,known as the lemma. It helps in reducing inflected forms to theirbaseformandnormalizesthewords.Weuse

the WordNetLemmatizerclass from the NLTK library to performlemmatization on each token. For example, the word "*running*"wouldbelemmatizedto"*run*,"and"*better*"wouldremainasitis.

1. **JoiningTokens**:thelaststepistojointhelemmatizedtokensback into a single string. This step is necessary as many NLPtechniques and algorithms expect input in the form of a stringrather than a list of tokens. We use the ' '.join()method toconcatenate the tokens with a space in between, resulting in apreprocesseduserinputstring.

def preprocess\_input(user\_input):lemmatizer=WordNetLemmatizer()

tokens=nltk.word\_tokenize(user\_input.lower())

lemmatized\_tokens=[lemmatizer.lemmatize(token)fortokenintokens]return''.join(lemmatized\_tokens)

## BuildingtheChatbotCore

Thechatbotcoreincludescreatingintentrecognition,entityextraction,andresponsegenerationcomponents.Inthisexample,wewillfocusona simple response generation mechanism using the TF-IDFvectorization technique and cosine similarity for matching user inputwithpredefinedresponses.

defgenerate\_response(user\_input,corpus):tfidf\_vectorizer=TfidfVectorizer()

tfidf\_matrix = tfidf\_vectorizer.fit\_transform(corpus)user\_input=preprocess\_input(user\_input)

user\_input\_vector = tfidf\_vectorizer.transform([user\_input])similarities=cosine\_similarity(user\_input\_vector,tfidf\_matrix)max\_similarity\_index=similarities.argmax()

response=corpus[max\_similarity\_index]returnresponse

## PuttingitAllTogether

Let’sputeverythingtogetherandcreateasimplechatbotthatrespondstopredefinedqueries.

corpus=[

'Hello',

'Howareyou?','Whatisyourname?','Tell me a joke','Goodbye',

'Whatistheweatherliketoday?',

'Canyourecommendagoodrestaurantnearby?','HowcanIcontactcustomersupport?',

'Tell me the latest news','Whatisthemeaningoflife?'

]

print("Chatbot:Hello!HowcanIassistyou?")#Chatbotinteractionloop

whileTrue:

user\_input=input("User:")

response=generate\_response(user\_input,corpus)print("Chatbot:",response)

ifuser\_input.lower()=='goodbye':break

During each iteration of the loop, the chatbot takes the user’s input,preprocesses it, compares it with the corpus using TF-IDF and cosinesimilarity,andselectsthemostrelevantresponsefromthecorpus.Theselectedresponseisthenprintedasthechatbot’sreply.

