**CREATE CHATBOT IN PYTHON**

**Phase 4 Submission Document**

**Project Title :** Creating chatbot

**Phase 4:Development Part 2**

**Topic:** Continue building the chatbot by integrating it into a web app using Flask.

**Creating Chatbot**

**Introduction:**

Integrating a chatbot into a web app using Flask combines the power of conversation AI with web-based accessibility. This process involves creating a Flask web application that communicates with the chatbot, allowing users to interact through a user-friendly interface. By merging these technologies, we can provide an engaging and interactive experience on our website or application.

**Chatterbot in python**

ChatterBot is a powerful Python library for building custom chatbots. When creating a chatbot, consider the following key factors:

* Define Your Target Audience
* Dataset for Training
* Natural Language of Communication
* Building the Chatbot
* Language Independence
* Custom Training Data

**Working of chatterbot**

The chatterbot works in the following manner:

* Input from the user
* Processing and Context Understanding
* Response Generation
* User Interaction

**Training the chatbot**

Training a chatbot using a dataset involves the process of teaching the chatbot to understand and generate responses based on the data it's provided. The dataset serves as a source of knowledge and conversation examples to improve the chatbot's conversational abilities.

## Developing a chatbot using flask

Developing a Chatbot with Flask and Kaggle Dataset Training

1. Create a Flask Web Application
2. Establish a Template Directory
3. Train the Chatbot with Kaggle Dataset
4. User Interaction with the Chatbot

**Project and Libraries setup**

I will be using **PyCharm**to develop this chatbot. Create a **Flask**project using PyCharm. Following libraries are required:

* chatterbot
* pytz
* sqlite3

**HTML template and CSS**

**index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<link rel="stylesheet" href="/static/style.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

</head>

<body>

<h1 class="jumbotron text-center">Chatterbot in Python using Flask Framework</h1>

<div class="container">

<div class="row">

<div class="col-sm-6 offset-sm-3">

<div id="chatbox" class="border border-success">

<p class="botText"><span>Hi! I'm Chatterbot</span></p>

</div>

<div id="userInput">

<input id="textInput" class="form-control" type="text" name="msg" placeholder="Type Your Message Here">

<input id="buttonInput" class="btn btn-success form-control" type="button" value="Send">

</div>

</div>

</div>

</div>

<script>

function getResponse() {

let userText = $("#textInput").val();

let userHtml = '<p class="userText"><span>' + userText + '</span></p>';

$("#textInput").val("");

$("#chatbox").append(userHtml);

document.getElementById('userInput').scrollIntoView({ block: 'start', behavior: 'smooth' });

$.get("/get", { msg: userText }).done(function(data) {

var botHtml = '<p class="botText"><span>' + data + '</span></p>';

$("#chatbox").append(botHtml);

document.getElementById('userInput').scrollIntoView({ block: 'start', behavior: 'smooth' });

});

}

$("#textInput").keypress(function(e) {

// If the Enter key is pressed

if (e.which == 13) {

getResponse();

}

});

$("#buttonInput").click(function() {

getResponse();

});

</script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

</body>

</html>

**style.css**

#textInput {

border: none;

border-bottom: 3px solid aqua;

}

.userText {

color: white;

font-family: monospace;

font-size: 17px;

text-align: right;

line-height: 30px;

}

.userText span {

background-color: #009688;

padding: 10px;

border-radius: 2px;

}

.botText {

color: white;

font-family: monospace;

font-size: 17px;

text-align: left;

line-height: 30px;

}

.botText span {

background-color: #EF5350;

padding: 10px;

border-radius: 2px;

}

**app.py**

from flask import Flask, render\_template, request

from chatterbot import ChatBot

from chatterbot.trainers import ChatterBotCorpusTrainer

import pandas as pd

app = Flask(\_\_name)

# Create chatbot

englishBot = ChatBot("Chatterbot", storage\_adapter="chatterbot.storage.SQLStorageAdapter")

trainer = ChatterBotCorpusTrainer(englishBot)

# Load the Kaggle dataset

data = pd.read\_csv('D:\\New\\simple-dialogs-for-chatbot.csv')# Update the file path as needed

# Train the chatbot with the Kaggle dataset

dialogs = data['User'] + data['Bot']

for dialog in dialogs:

trainer.train([dialog])

# Define app routes

@app.route("/")

def index():

return render\_template("index.html")

@app.route("/get")

# Function for the bot response

def get\_bot\_response():

userText = request.args.get('msg')

return str(englishBot.get\_response(userText))

if \_\_name\_\_ == "\_\_main\_\_":

app.run()

**Output:**

