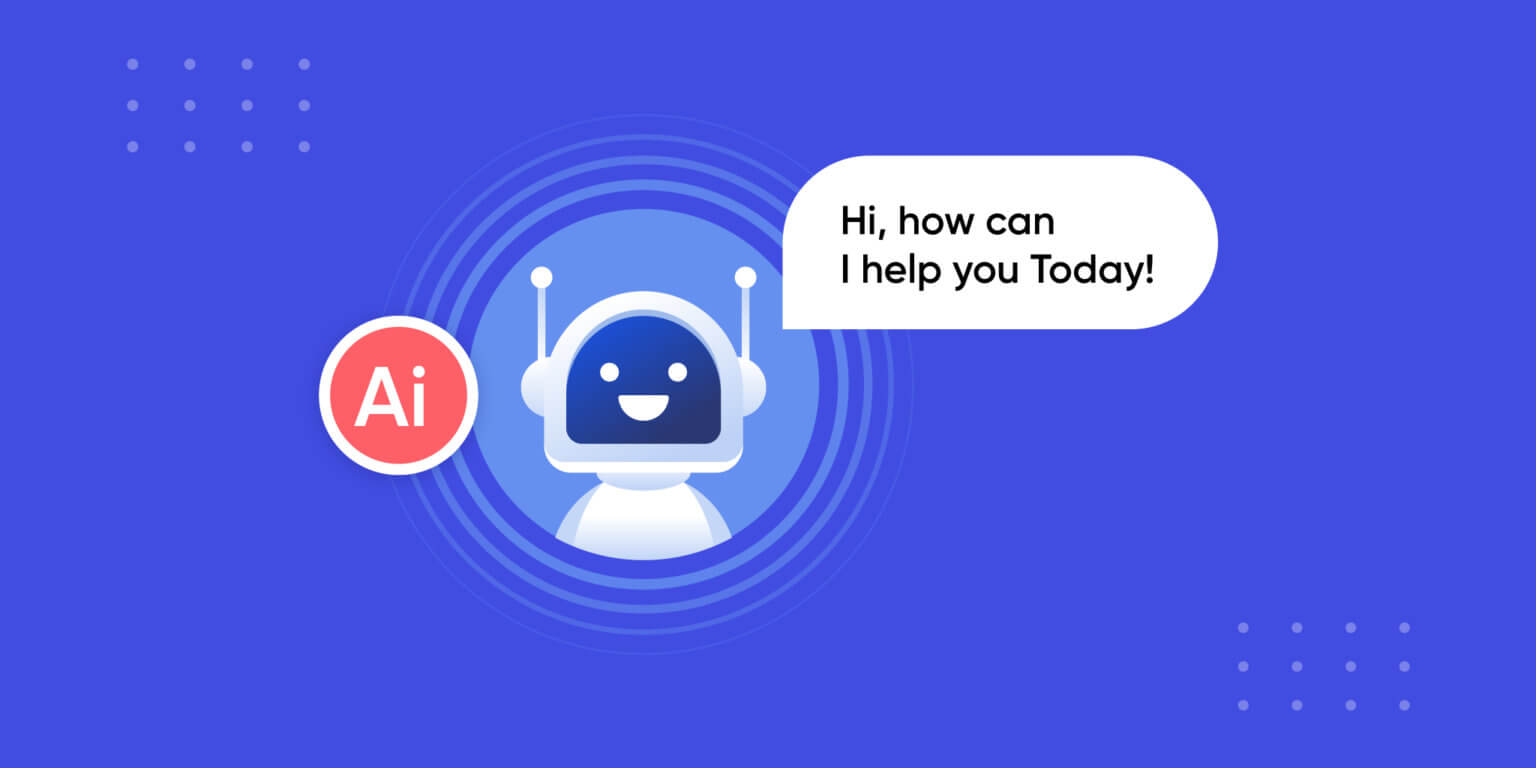
**DOMAIN : ARTIFICIAL INTELLIGENCE**

**PROJECT TITLE : CHATBOT USING PYTHON**

**PHASE :03 SUBMISSION**



Our Institutional Chatbot, developed using Python's advanced AI capabilities, facilitates seamless communication within our institution. It swiftly provides accurate assistance for diverse queries, ensuring a user-friendly experience for students, faculty, and staff. With stringent data security measures, it prioritizes privacy and confidentiality.

Adaptable and efficient, the chatbot streamlines communication and problem-solving, enhancing the overall user experience. Its sophisticated natural language processing abilities enable it to continually learn and evolve, catering to the dynamic needs of our institution while maintaining a high standard of reliability and engagement.

**CODE :**

**BACKEND:**

**Controller.py2:**

import spacy

import csv

from translate import Translator

nlp = spacy.load('en\_core\_web\_sm')

questions = []

answers = []

with open('dataset.csv', 'r') as file:

    reader = csv.reader(file)

    next(reader)  # Skip the header row

    for row in reader:

        questions.append(row[0])

        answers.append(row[1])

def get\_answer(user\_input):

    user\_input = user\_input.lower()

    doc = nlp(user\_input)

    for i, question in enumerate(questions):

        question\_doc = nlp(question.lower())

        similarity = doc.similarity(question\_doc)

        if similarity > 0.6:

            return answers[i]

    return "Sorry, I couldn't find an answer to your question."

def translate(sentence,code):

    translater = Translator(from\_lang="en",to\_lang=code)

    translated = translater.translate(sentence)

    return translated

**main.py:**

from flask import Flask

from controller import get\_answer, translate

from schema import question, translating

app = Flask(\_\_name\_\_)

@app.get("/")

def working():

    return "API Working"

@app.post('/ask')

def askQuestion(ques:question):

    return get\_answer(ques.question)

@app.post('/translate')

def translater(translating: translating):

    return translate(translating.sentence,translating.code)

if '\_\_main\_\_'==\_\_name\_\_:

    # uvicorn.run(

    #     'main:app',

    #     reload=True,

    #     port=8000,

    #     host='localhost'

    # )

    app.run(debug=True)

**schema.py:**

from pydantic import BaseModel

class question(BaseModel):

    question: str

class translating(BaseModel):

    sentence: str

    code: str

**dataset.csv:**

**Dataset Link:**[**https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot**](https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot)