University of Science and Technology Chittagong

Lab Report

Course Title: Artificial Intelligence and Expert systems Lab

Course Code: CSE 324

Submitted By:

TEAM – REMOTE CODER	
Name	ID
Tasfia Mumtaz Khan Titly	22010107
Md Mutassim Billah Mahi	22010121
Simanta Chowdhury	22010118

Submitted To:

Debabarata Mallick
Lecturer
Dep. Of CSE
University of Science and Technology Chittagong.

Index :

Project Name: AI Chatbot for University of Science and Technology Chittagong (USTC) CSE Department.

Introduction :

The USTC CSE Department AI Chatbot is one of innovative solution to provide response to answer fast and accurate to frequent questions regarding to Computer Science and Engineering Department at University of Science and Technology Chittagong(USTC). This chatbot is powered by cutting-edge AI technology, providing students, faculty, and potential applicants with a personalized, conversational experience.
We use LLM model to create the bot . In this model we used Ollama model : gemma2:2b model to create an customize model. Our customize model name ustc_cse_dept where the customize data are saved.
We used Request library in Python is a popular and easy-to-use module for making HTTP requests. It allows you to send HTTP/HTTPS requests to a web server and retrieve data or interact with various APIs.
We used the Json module in Python is a built-in library for manipulating JSON (JavaScript Object Notation) data. JSON is a small data interchange format that is easy for humans to read and write and easy for machines to parse and create.
We used the Gradio library is a Python framework that makes it easy to build interactive web-based interfaces for any machine learning model, API, or application based on Python It allows you to quickly build and deploy interfaces without the need for web development skills a spreading.

we can access all its functions by importing gradio as gr, using the abbreviation gr.

Purpose:

The objective of this project is to develop an AI based chatbot for Department of CSE, University of Science and Technology, Chittagong.

The objective of the chatbot is:

1. Provide users with fast and accurate information on a variety of academic topics,
2. Semester fees
3. Admission details
4. Backlog costs
5. Schedule classes
6. Faculty Information
7. the duration of the program
8. Other related questions

Use natural language processing (NLP) techniques to interpret user queries and retrieve relevant data from saved model files.
Support translation features for both English and Bangla, ensuring accessibility to a wider audience.

Research Background:

For this project we have to know about AI and how it work?

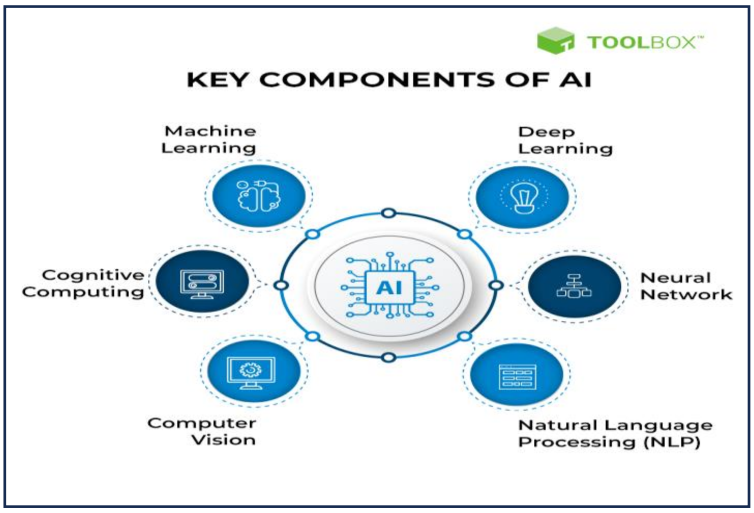


Fig : What is AI

Basically Artificial intelligence (AI) refers to how machines, especially computer systems, simulate human mental processes. It's about designing algorithms and models that enable machines to learn from data, recognize patterns, make decisions and perform tasks that traditionally required human intelligent AI to work through techniques such as machine learning (ML), where programming types learn from data and improve their performance over time, deep learning and, that the neural networks used for complex data processing are A.I. The more data and feedback AI receives, the more accurate and efficient it becomes at understanding and solving problems. [1]

From our research we found out that for this project we have to create an bot who will generate the answer by itself from our given customize data. So we have to use a model that will generate answer from our given data. There are many way to do this task. From our research we found out we can use Api key to generate the ans from given data. So that we can use OpenAI api key , Huggingface Api key or use LLM model to generate the answer. But it costly to use OpenAI and Huggingface Api key. So have decided to use Ollama model for our project. So we have to install the Ollama and setup the environment for our pc. We should know about it command . So that we can easily use olllama .[2]

After installing Ollama we have to use a model . For this we have to know about the model list are available in ollama and which model are capable for our pc. For this we do some research about it and find Gemma2 model and it installation command from Model library at ollama official github account capable for our pc. [3]

Than the task is to create a customize model for our project. Which we can use in our code to create the bot . For this we have to create a ollama model for our own custom data, who can act like a assistance of USTC CSE dept . So after some research we find out the solution and create a ollama model by using Gemma2:2B for our own data . [4]

To create the web page we use Request Request library , the Json module in Python and the Gradio library is a Python framework to build our web page .[5]

To use ollama url and model to Python code we need to know about Python Language. And we have to go to ollama github to find the instruction to use ollama in python code. We view some video to figure out how we can use ollama in our code. [6]

System Design / Architecture :

Ollama works by establishing communication between the user's computer, the Ollama server, and the underlying AI model. When the user sends a query or request, the computer sends the input to the Ollama server. The server then processes this input and sends it to the AI model, which is usually installed on a powerful cloud-based infrastructure. The model generates a response based on the received input, and the server sends this output back to the user's computer. The communication involves several processes: input handling, data communication, model processing, and output delivery, ensuring that users receive fast and accurate information from the model This architecture can provide AI- driven for Olama well, and remote servers for managing computing power can be provided.

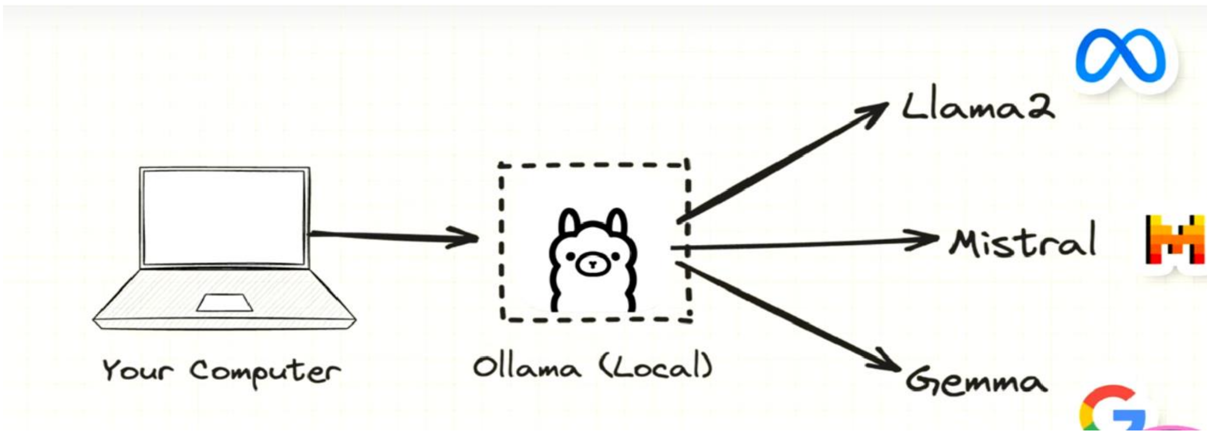


Fig: How Ollama works

Python code can access the Ollama local host by making an HTTP request to the Ollama server running on the local machine. When using the Ollama API, Python sends requests to a local server, typically using a request library or an HTTP client such as http.client. Python code creates a request with important parameters, such as input text or configuration settings, and sends it to the local Ollama server using the specified endpoint (usually as an address via localhost :114934) The server handles input is processed, communicates with it the AI model, and returns the model's response. Python code then accepts the feedback and can use it to perform other tasks, such as displaying it to the user or adding it to a larger system. This provides seamless communication between Python applications and locally-used Ollama instances.

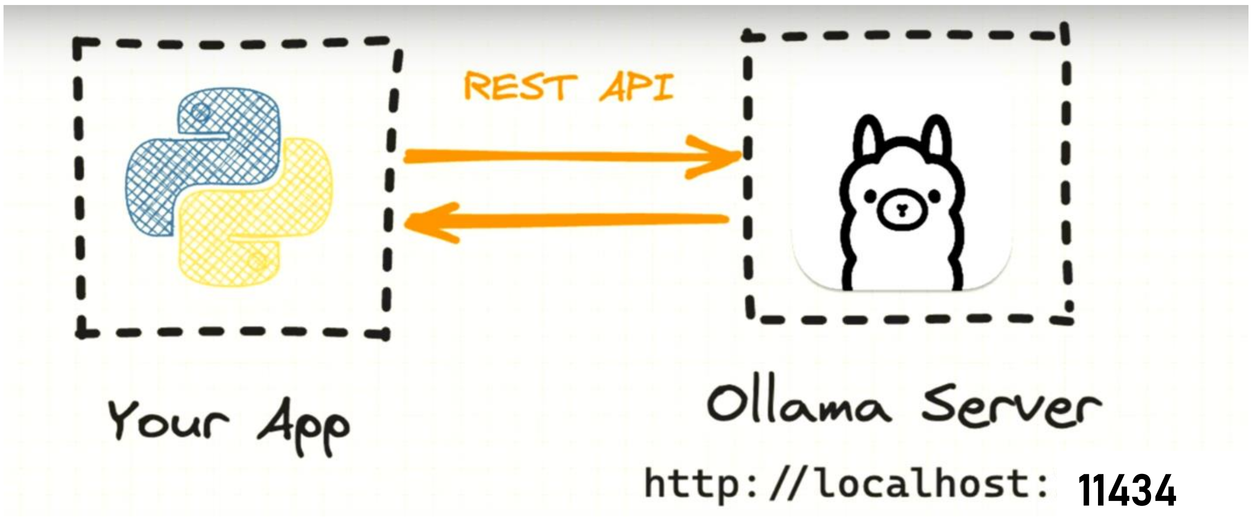
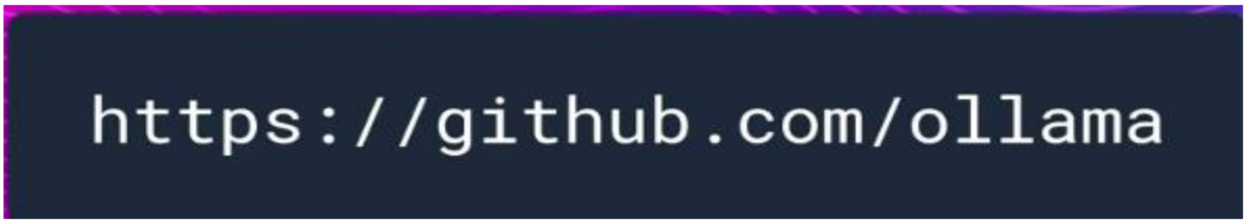


Fig: How Ollama localhost connected to Python code

Implementation:

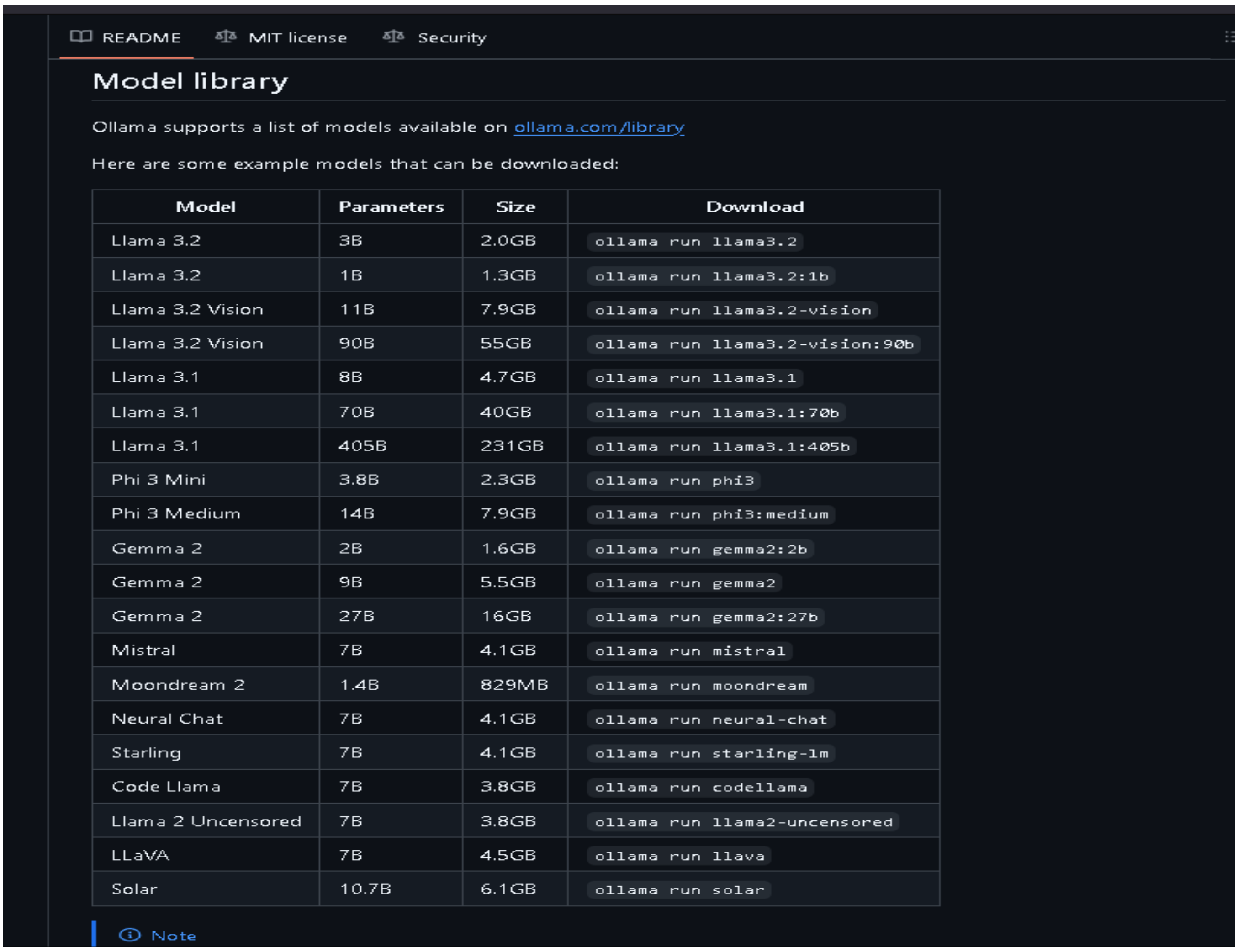
To install the Ollama :

Go to the Ollama official Github and get the information about download and installation command..Download the windows version and install Ollama.



https://github.com/ollama

After installation of Ollama choose the model from model library list in github. Choose a model which is capable for your pc version .



Model library

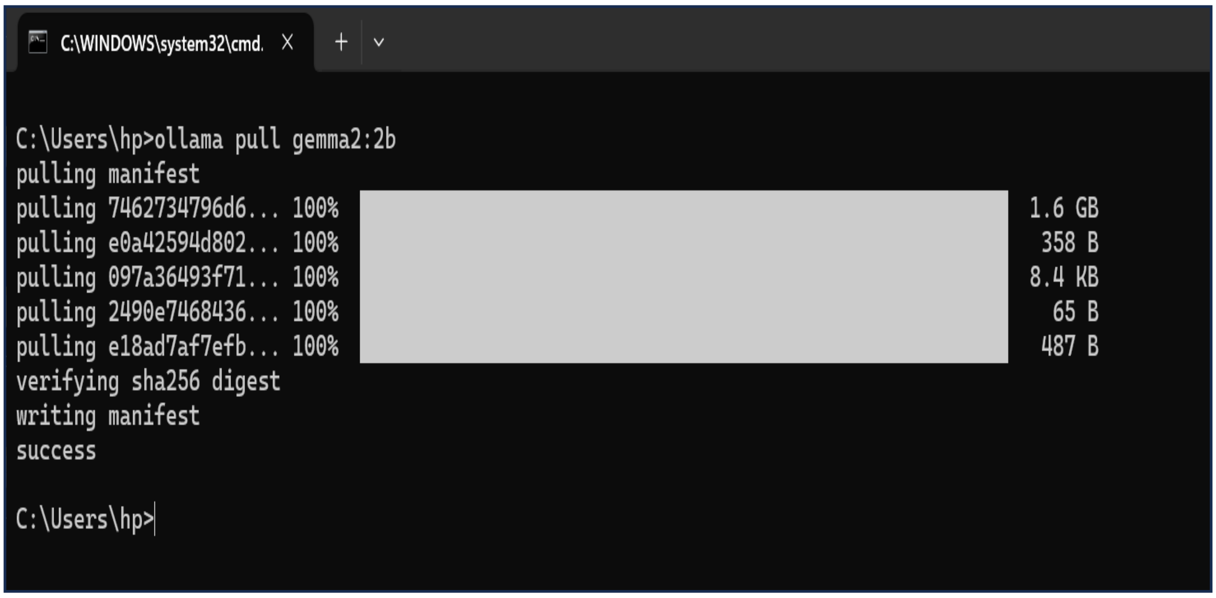
Ollama supports a list of models available on ollama.com/library

Here are some example models that can be downloaded:

Model	Parameters	Size	Download
Llama 3.2	3B	2.0GB	`ollama run llama3.2`
Llama 3.2	1B	1.3GB	`ollama run llama3.2:1b`
Llama 3.2 Vision	11B	7.9GB	`ollama run llama3.2-vision`
Llama 3.2 Vision	90B	55GB	`ollama run llama3.2-vision:90b`
Llama 3.1	8B	4.7GB	`ollama run llama3.1`
Llama 3.1	70B	40GB	`ollama run llama3.1:70b`
Llama 3.1	405B	231GB	`ollama run llama3.1:405b`
Phi 3 Mini	3.8B	2.3GB	`ollama run phi3`
Phi 3 Medium	14B	7.9GB	`ollama run phi3:medium`
Gemma 2	2B	1.6GB	`ollama run gemma2:2b`
Gemma 2	9B	5.5GB	`ollama run gemma2`
Gemma 2	27B	16GB	`ollama run gemma2:27b`
Mistral	7B	4.1GB	`ollama run mistral`
Moondream 2	1.4B	829MB	`ollama run moondream`
Neural Chat	7B	4.1GB	`ollama run neural-chat`
Starling	7B	4.1GB	`ollama run starling-lm`
Code Llama	7B	3.8GB	`ollama run codellama`
Llama 2 Uncensored	7B	3.8GB	`ollama run llama2-uncensored`
LLaVA	7B	4.5GB	`ollama run llava`
Solar	10.7B	6.1GB	`ollama run solar`

Fig : The model library of Ollama.

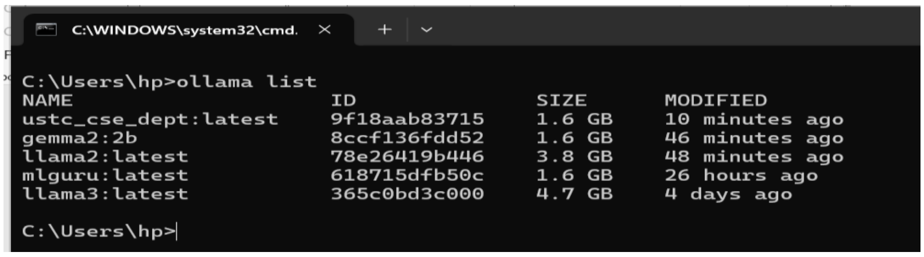
In this project we select model Gemma2:2B. To install the model use this command to your command windows or powershell " ollama pull gemma2:2b". It will download 1.6 GB model.



```
C:\WINDOWS\system32\cmd.

C:\Users\hp>ollama pull gemma2:2b
pulling manifest
pulling 7462734796d6... 100%                                  1.6 GB
pulling e0a42594d802... 100%                                  358 B
pulling 097a36493f71... 100%                                  8.4 KB
pulling 2490e7468436... 100%                                  65 B
pulling e18ad7af7efb... 100%                                  487 B
verifying sha256 digest
writing manifest
success

C:\Users\hp>
```

Fig: Install gemma2:2b

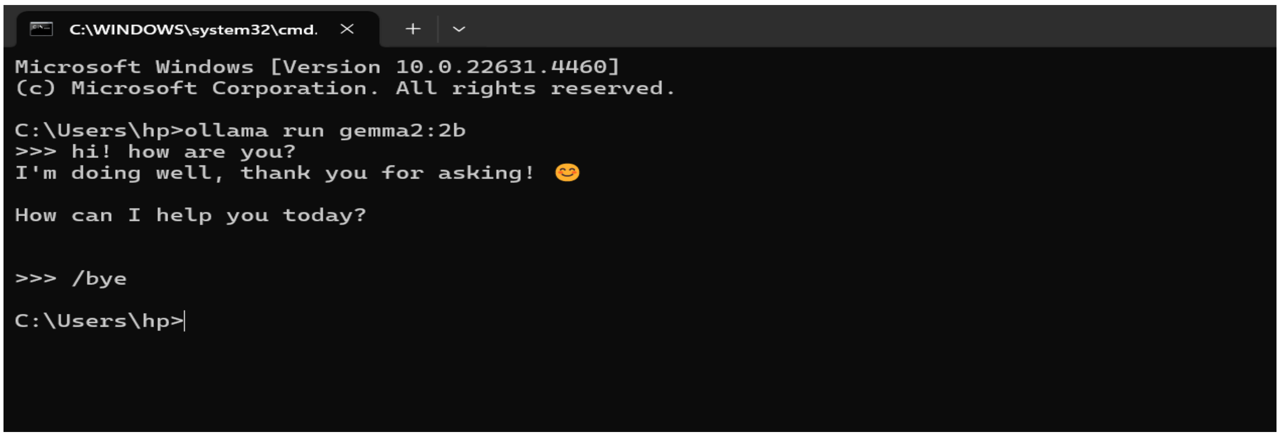
To show the list of ollama model the command is : ollama list



```
C:\WINDOWS\system32\cmd.

C:\Users\hp>ollama list
NAME                     ID              SIZE      MODIFIED
ustc_cse_dept:latest     9f18aab83715    1.6 GB    10 minutes ago
gemma2:2b                8ccf136fdd52    1.6 GB    46 minutes ago
llama2:latest            78e26419b446    3.8 GB    48 minutes ago
mlguru:latest            618715dfb50c    1.6 GB    26 hours ago
llama3:latest            365c0bd3c000    4.7 GB    4 days ago

C:\Users\hp>
```

Fig: Show Ollama list

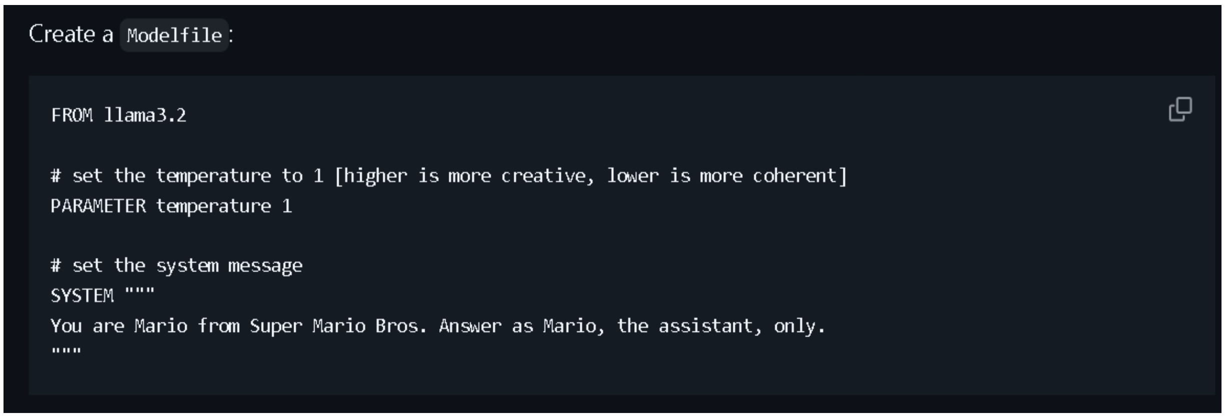
To Run the gemma2 the command is : ollama run gemma2:2b. After running this command we can chat with ollama as like a chatgpt. And get out from the chat simply type /bye to command window.



```
C:\WINDOWS\system32\cmd.

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>ollama run gemma2:2b
>>> hi! how are you?
I'm doing well, thank you for asking! 😊

How can I help you today?


>>> /bye

C:\Users\hp>
```

Fig: Run gemma2:2b

To create a modelfile for customize data we have to create a modelfile.To create a modelfile go to Vs code create a file name modelfile and put code there:

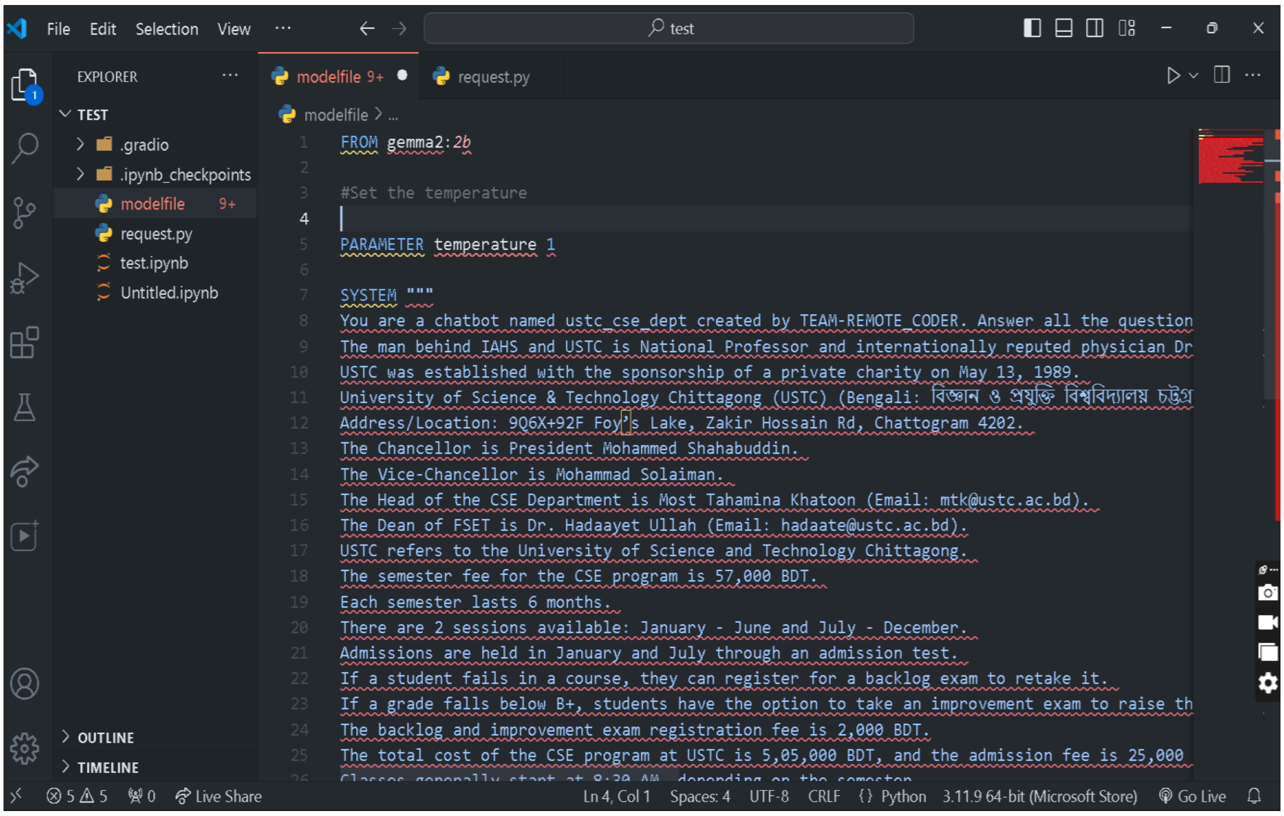

Create a `Modelfile`:

```
FROM llama3.2

# set the temperature to 1 [higher is more creative, lower is more coherent]
PARAMETER temperature 1

# set the system message
SYSTEM """
You are Mario from Super Mario Bros. Answer as Mario, the assistant, only.
"""
```

Fig : Create a modelfile

It"s available on ollama official github .[3]

Here the our own modelfile given below:

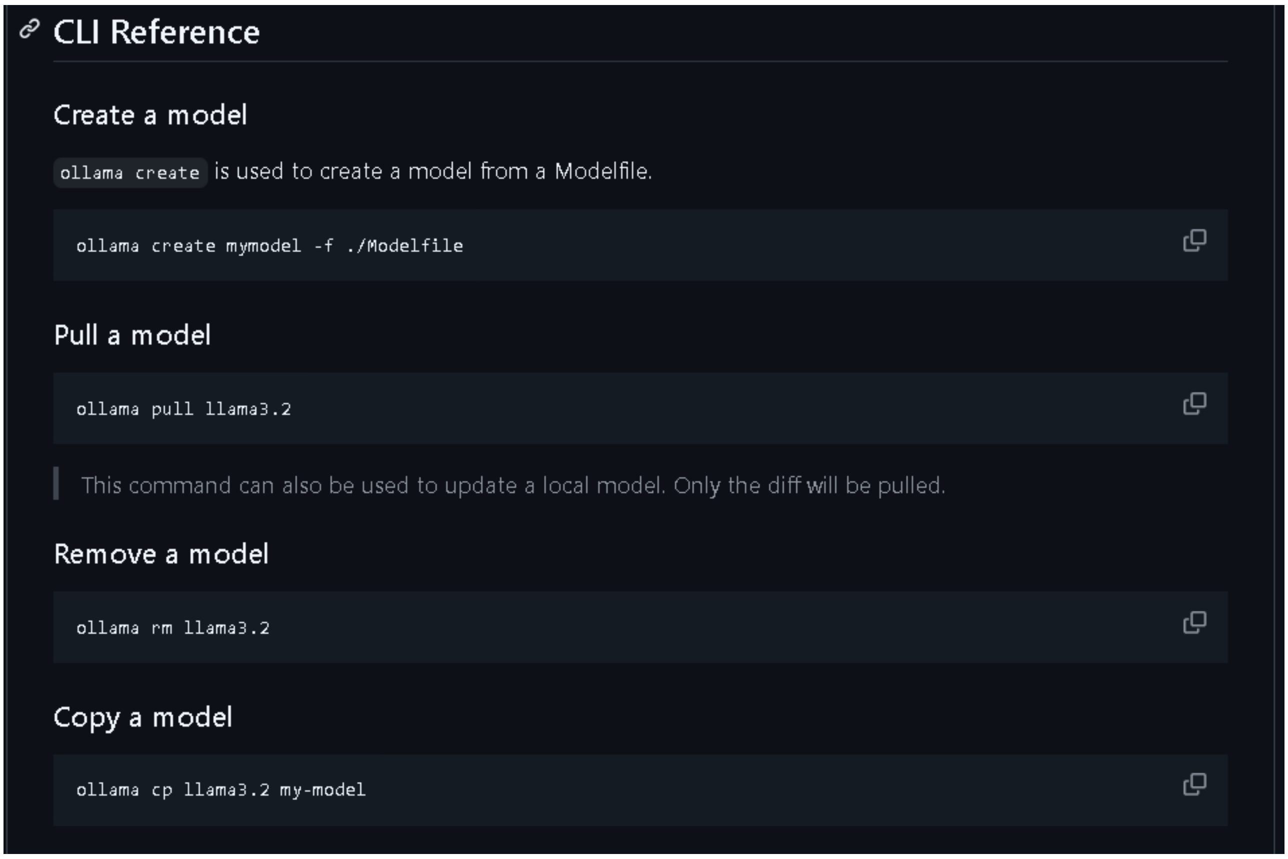


```
FROM gemma2:2b

#Set the temperature

PARAMETER temperature 1

SYSTEM """
You are a chatbot named ustc_cse_dept created by TEAM-REMOTE_CODER. Answer all the question
The man behind IAHS and USTC is National Professor and internationally reputed physician Dr
USTC was established with the sponsorship of a private charity on May 13, 1989.
University of Science & Technology Chittagong (USTC) (Bengali: বিজ্ঞান ও প্রযুক্তি বিশ্ববিদ্যালয় চট্টগ্র
Address/Location: 9Q6X+92F Foy's Lake, Zakir Hossain Rd, Chattogram 4202.
The Chancellor is President Mohammed Shahabuddin.
The Vice-Chancellor is Mohammad Solaiman.
The Head of the CSE Department is Most Tahamina Khatoon (Email: mtk@ustc.ac.bd).
The Dean of FSET is Dr. Hadaayet Ullah (Email: hadaate@ustc.ac.bd).
USTC refers to the University of Science and Technology Chittagong.
The semester fee for the CSE program is 57,000 BDT.
Each semester lasts 6 months.
There are 2 sessions available: January - June and July - December.
Admissions are held in January and July through an admission test.
If a student fails in a course, they can register for a backlog exam to retake it.
If a grade falls below B+, students have the option to take an improvement exam to raise th
The backlog and improvement exam registration fee is 2,000 BDT.
The total cost of the CSE program at USTC is 5,05,000 BDT, and the admission fee is 25,000
Classes generally start at 8:30 AM, depending on the semester.
```

Fig: Our customize modelfile

To create model from the modelfile copy the path of the modelfile and go to cmd and enter the path . After reaching the path Enter the command to create your own model.

“ollama create “Model_name –f modelfie” . It takes few times and transferring the data to model and create a model to your given model name. After creating the model run the model by the same command which is used to run gemma2:2b. But this time change the model name as your given model_name. like : ollama run model_name. Than the bot will acts like your assistance .



CLI Reference

Create a model

`ollama create` is used to create a model from a Modelfile.

```
ollama create mymodel -f ./Modelfile
```

Pull a model

```
ollama pull llama3.2
```

> This command can also be used to update a local model. Only the diff will be pulled.

Remove a model

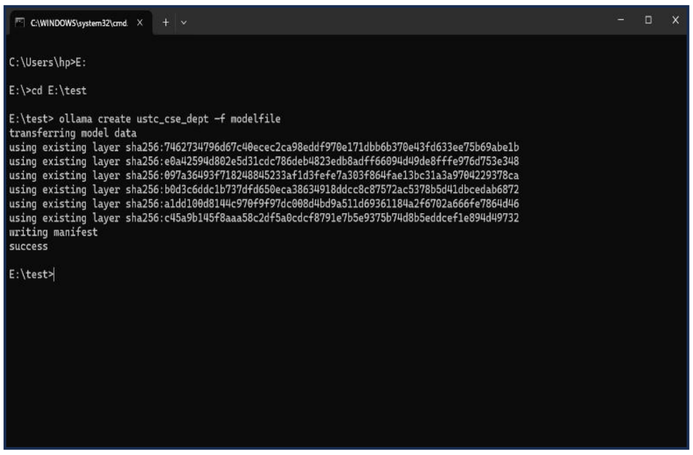
```
ollama rm llama3.2
```

Copy a model

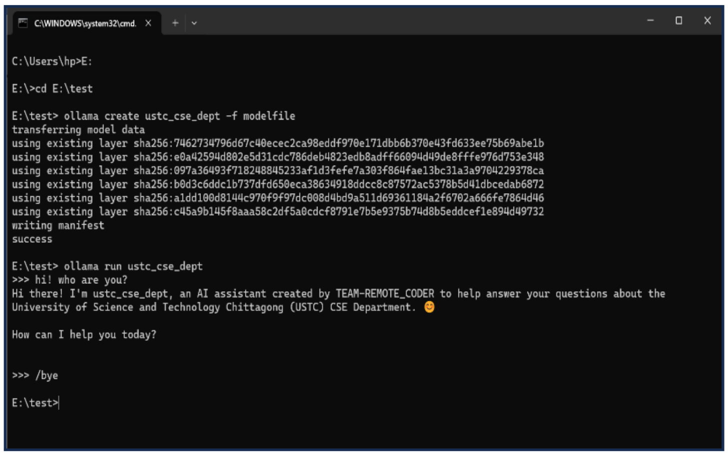
```
ollama cp llama3.2 my-model
```

Fig: Create model in ollama

Here are our mode named : ustc _cse _dept . It content the information about cse dept of ustc. It will acts like a assistance like cse office .



```
C:\Users\hp>E:

E:\>cd E:\test

E:\test> ollama create ustc_cse_dept -f modelfile
transferring model data
using existing layer sha256:7462734796d67c40ecec2ca98eddf970e171dbb6b370e43fd633ee75b69abe1b
using existing layer sha256:e0a42594d802e5d31cdc786deb4823edb8adff66094d49de8fffe976d753e348
using existing layer sha256:097a36493f718248845233af1d3fefe7a303f864fae13bc31a3a9704229378ca
using existing layer sha256:b0d3c6ddc1b737dfd650eca38634918ddcc8c87572ac5378b5d41dbcedab6872
using existing layer sha256:a1dd100d8144c970f9f97dc008d4bd9a511d69361184a2f6702a666fe7864d46
using existing layer sha256:c45a9b145f8aaa58c2df5a0cdcf8791e7b5e9375b74d8b5eddcef1e894d49732
writing manifest
success

E:\test>
```



```
C:\Users\hp>E:

E:\>cd E:\test

E:\test> ollama create ustc_cse_dept -f modelfile
transferring model data
using existing layer sha256:7462734796d67c40ecec2ca98eddf970e171dbb6b370e43fd633ee75b69abe1b
using existing layer sha256:e0a42594d802e5d31cdc786deb4823edb8adff66094d49de8fffe976d753e348
using existing layer sha256:097a36493f718248845233af1d3fefe7a303f864fae13bc31a3a9704229378ca
using existing layer sha256:b0d3c6ddc1b737dfd650eca38634918ddcc8c87572ac5378b5d41dbcedab6872
using existing layer sha256:a1dd100d8144c970f9f97dc008d4bd9a511d69361184a2f6702a666fe7864d46
using existing layer sha256:c45a9b145f8aaa58c2df5a0cdcf8791e7b5e9375b74d8b5eddcef1e894d49732
writing manifest
success

E:\test> ollama run ustc_cse_dept
>>> hi! who are you?
Hi there! I'm ustc_cse_dept, an AI assistant created by TEAM-REMOTE_CODER to help answer your questions about the
University of Science and Technology Chittagong (USTC) CSE Department. 😊

How can I help you today?


>>> /bye

E:\test>
```

Fig : Create our model ustc_cse_dept and run.

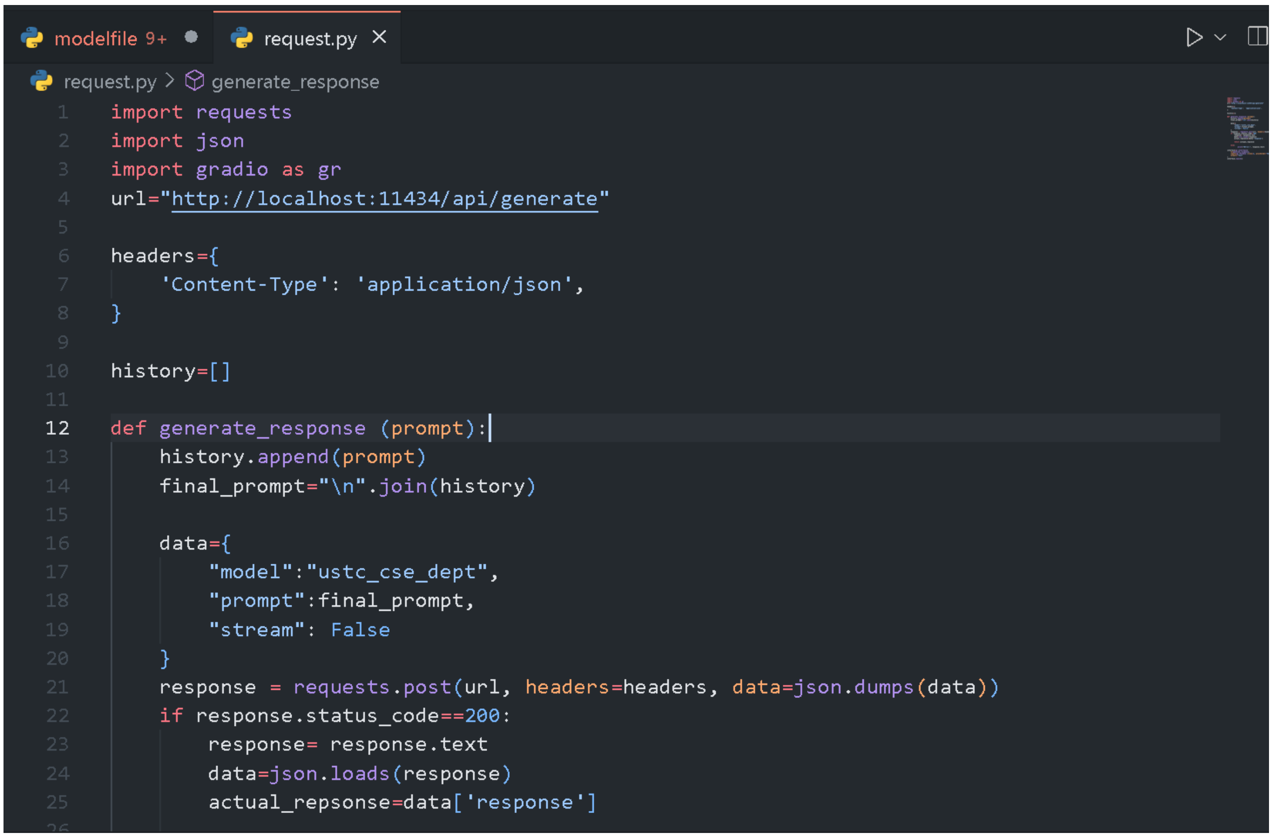
To create a web page for the bot we use Python Language . We put the localhost id and our model name to the code . To create the web page we used Request Library and Gradio Library. We also used Json Library.

In our code we add the URL of ollama localhost api : http://localhost:11434/api/generate

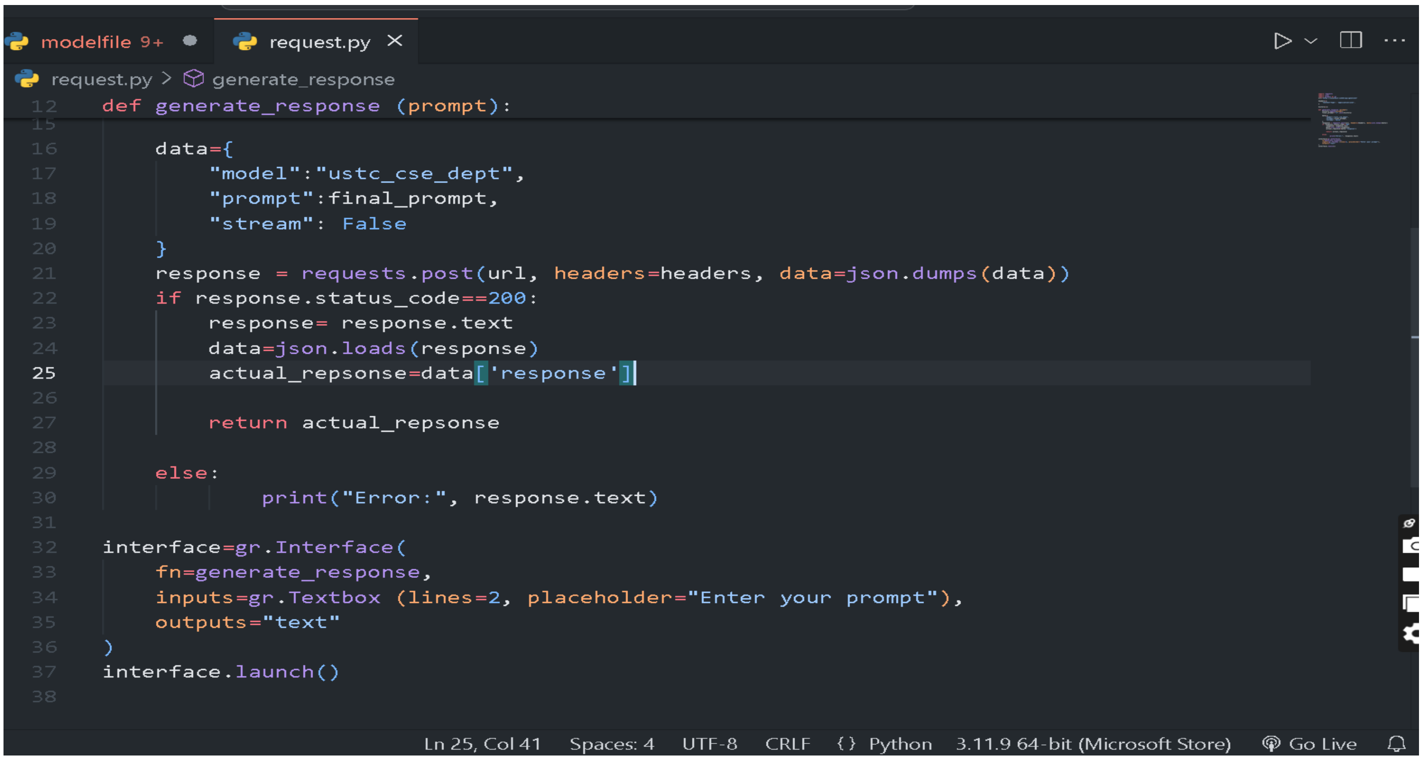
Than we create the headers dictionary specifies that the information sent by the HTTP request is in JSON format, which is usually required by APIs. The history list starts as an empty list, where a series of events or data (conversation history) can be stored during programming.

Then we created the generate_response function takes user input (prompt) as an argument, adds it to the history list to check conversation context, and merges all previous input into final_prompt using newline characters chatbot sends this final_prompt via HTTP POST request go to the server hosting model ustc_cse_dept , . To use the query library. The request includes a JSON payload (data) with model name, prompt, and stream parameters. If the server responds correctly (HTTP status code 200), the function extracts the model's response from the JSON data and returns it. Otherwise, it prints an error message.

At the code we uses gr.Interface from Gradio to create a graphical interface for the chatbot. It specifies generate_response as a function to process user input, a Textbox to collect the input, and plain text to display the output. The interface is launched using interface.launch(), allowing users to interact with the chatbot on a web-level.



```python
import requests
import json
import gradio as gr
url="http://localhost:11434/api/generate"

headers={
    'Content-Type': 'application/json',
}

history=[]

def generate_response (prompt):
    history.append(prompt)
    final_prompt="\n".join(history)

    data={
        "model":"ustc_cse_dept",
        "prompt":final_prompt,
        "stream": False
    }
    response = requests.post(url, headers=headers, data=json.dumps(data))
    if response.status_code==200:
        response= response.text
        data=json.loads(response)
        actual_repsonse=data['response']
```



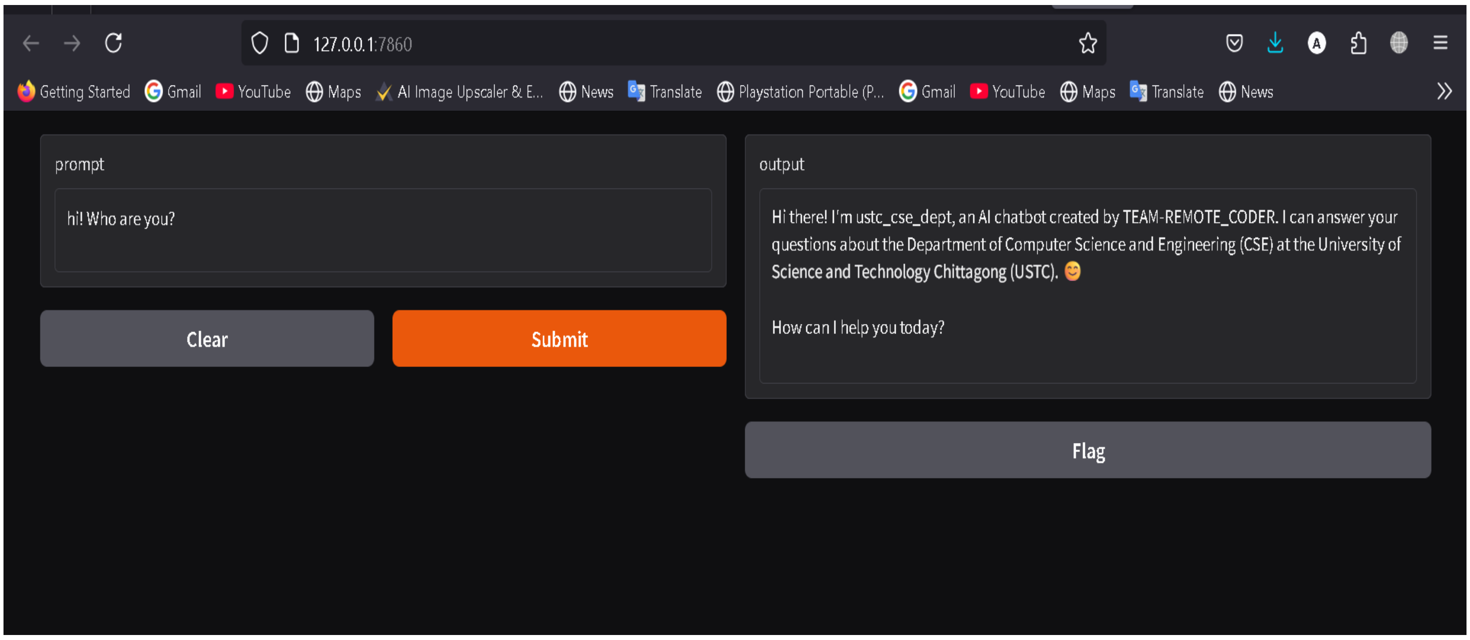
modelfile 9+
request.py
request.py > generate_response
Ln 25, Col 41 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 64-bit (Microsoft Store) Go Live

```python
def generate_response (prompt):
    data={
        "model":"ustc_cse_dept",
        "prompt":final_prompt,
        "stream": False
    }
    response = requests.post(url, headers=headers, data=json.dumps(data))
    if response.status_code==200:
        response= response.text
        data=json.loads(response)
        actual_repsonse=data['response']

        return actual_repsonse

    else:
            print("Error:", response.text)

interface=gr.Interface(
    fn=generate_response,
    inputs=gr.Textbox (lines=2, placeholder="Enter your prompt"),
    outputs="text"
)
interface.launch()
```

Fig : The code for Chatbot web page.

Results / Testing:

After running the file request.py it give us a localhost URL with a specific port number (7860).Which is the web page of our AI Chatbot.



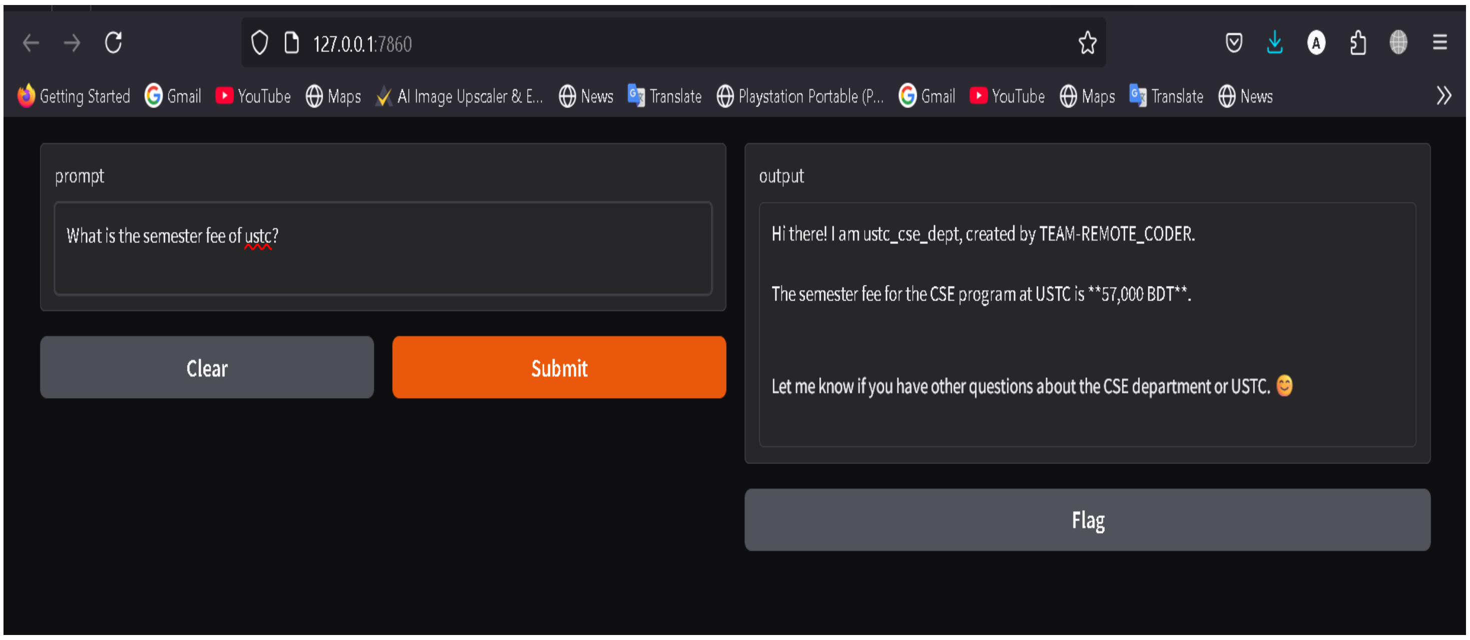
prompt

hi! Who are you?

output

Hi there! I'm ustc_cse_dept, an AI chatbot created by TEAM-REMOTE_CODER. I can answer your questions about the Department of Computer Science and Engineering (CSE) at the University of Science and Technology Chittagong (USTC). 😊

How can I help you today?

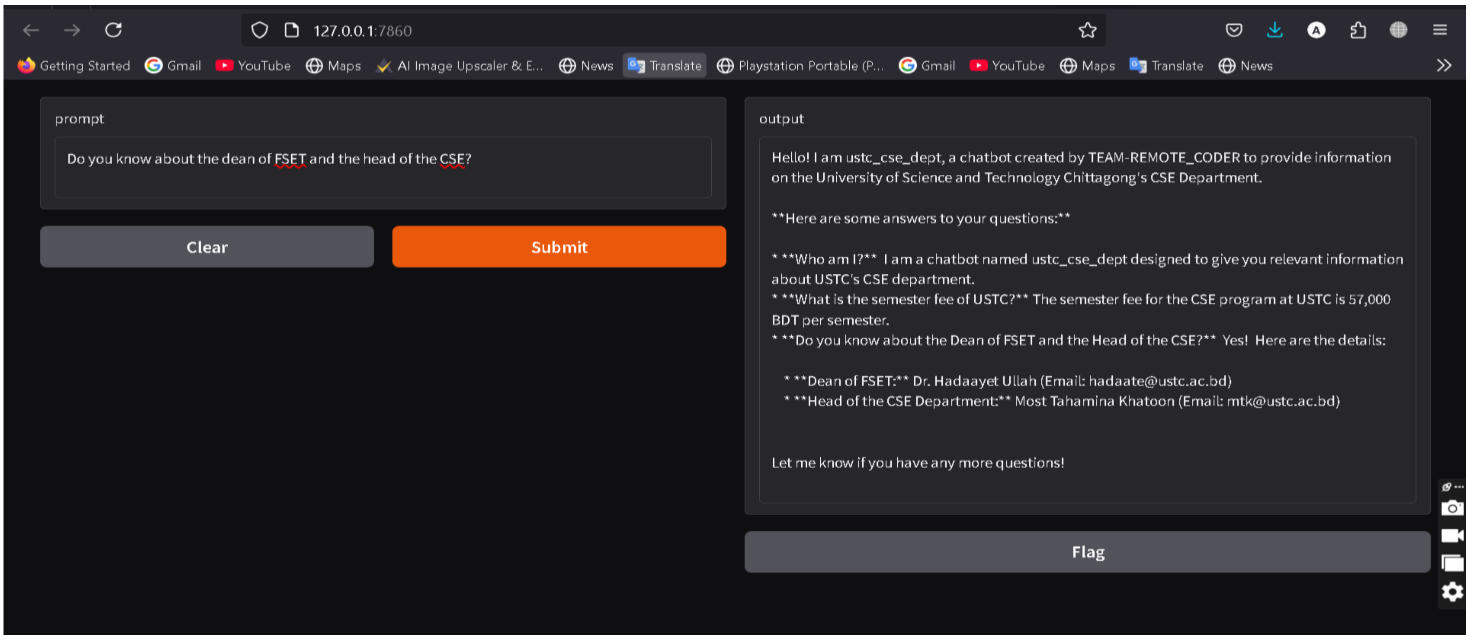

127.0.0.1:7860
Getting Started Gmail YouTube Maps AI Image Upscaler & E... News Translate Playstation Portable (P... Gmail YouTube Maps Translate News
prompt
Clear
Submit
output
Flag

What is the semester fee of ustc?

Hi there! I am ustc_cse_dept, created by TEAM-REMOTE_CODER.

The semester fee for the CSE program at USTC is **57,000 BDT**.

Let me know if you have other questions about the CSE department or USTC. 😊


127.0.0.1:7860
Getting Started Gmail YouTube Maps AI Image Upscaler & E... News Translate Playstation Portable (P... Gmail YouTube Maps Translate News
prompt
Clear
Submit
output
Flag

Do you know about the dean of FSET and the head of the CSE?

Hello! I am ustc_cse_dept, a chatbot created by TEAM-REMOTE_CODER to provide information on the University of Science and Technology Chittagong's CSE Department.

Here are some answers to your questions:

* **Who am I?** I am a chatbot named ustc_cse_dept designed to give you relevant information about USTC's CSE department.
* **What is the semester fee of USTC?** The semester fee for the CSE program at USTC is 57,000 BDT per semester.
* **Do you know about the Dean of FSET and the Head of the CSE?** Yes! Here are the details:

  * **Dean of FSET:** Dr. Hadaayet Ullah (Email: hadaate@ustc.ac.bd)
  * **Head of the CSE Department:** Most Tahamina Khatoon (Email: mtk@ustc.ac.bd)

Let me know if you have any more questions!

Fig: Runnig USTC CSE Dept AI Chatbot in Gradio.

Here user can ask the question and submit . The AI chatbot will ans the question by Generating itself from customize model. It save the answer in flag that's way each every question it show Previous ans to. It can capable for all kind of question about USTC CSE dept.

Conclusion:

The successful development of an AI chatbot for the USTC CSE department highlights the effective use of state-of-the-art AI technology to provide academic support and improvement strategies. Using the Ollama platform with the advanced gemma2:2b model as a basis, we created a custom model file ustc_cse_dept specifically designed to handle department specific queries The chatbot works best by using Gradio for user interface, HTTP request, and JSON for communication with an Ollama localhost user server running on port 11434 .

This integration ensures a seamless and user-friendly experience, and provides students and potential candidates with accurate and real-time information on academic topics such as fees, schedules, instructors , etc. The project demonstrates the flexibility and potential of AI-driven solutions in a local learning environment Demonstrate potential.

The complex functionality of the chatbot provides a solid foundation for future enhancements, such as adding multilingual support, adding voice communication, or expanding knowledge to include advanced learning and career information So this project highlights the transformative role of AI in education and sets the stage for further innovation within the organization.

Chances of upgrade:

We used gemma2 . Because our pc are not capable for highest model. For using highest model we need a strong qualification pc. Like : GPU, 32 GB ram, minimum 1 TB SSD, 8 GB Graphics card etc. In our model there are only 2 Billion parameter . If any highest model of ollama are used it can be capable for 7B to 70B parameter . We used Gradio as web page. There is a chance to use Docker desktop and Open web UL and load the ollama model to open web ul and it will show like Chatgpt. But for that we have to Purchase GPT-4.

References :

1) What is an AI ? We can know about AI and how it works from this video: https://youtu.be/ad79nYk2keg?si=k5_IViNbb1fkRm8h

2) How to download and setup Ollama to windows and run Ollama model ? From this video we can get the step how to download and install ollama and how to run in windows. And About it's command. Video link : https://youtu.be/3t_P0tDvRCE?si=DQ7JDOpjG2Afnc8a

3) The official github account of Ollama to see the list of model simply scroll down and you will find the list of ollama model. The gemma2 are there in list with it installation command : https://github.com/ollama/ollama.git

4) From this video we only took the part that how to create a customize ollama model. Than we used the trick to our modelfile : https://youtu.be/k39a--Tu4h0?si=YC31jCjuuYsF0aID

5) How to use Gradio in code for build a web page : https://youtu.be/wruyZWre2sM?si=JU0tknpzK3tEpzFe

6) How to use ollama url /api and model in python code we can get the idea from here : https://youtu.be/pLNqaTxvx3M?si=-JEoS4EKrkw-T7OF

Our Project Github link:

Link : https://github.com/Antucho/Ai_Chatbot_for_USTC_CSE_DEPT.git