

Create a AI Chatbot (Agent) in python, flask and mysql using tinyllama to fetch shipping details based on orderid

Raman Deep Singh

What is a ChatBot

Chatbot is a artificial intelligence agent. An AI Agent is a programming loop. An AI Agent takes a goal, thinks on that goal, perform tasks to achieve that goal, if goal is achieved agent or chatbot ends else it again takes the goal, thinks on the goal and performs tasks to achieve that goal.\

Following is the code for creating ai chatbot in python, flask and mysql using ai model Tinyllama

For example if you will enter

Track Order ORD1001

Or

Give details about order ORD1001

Or

ORD1001

It will give you the shipping details of the order ORD1001

Following is the code for mysql

```
CREATE DATABASE IF NOT EXISTS order_chatbot;
```

```
USE order_chatbot;
```

```
CREATE TABLE IF NOT EXISTS orders (  
  orderid VARCHAR(50) PRIMARY KEY,  
  details TEXT  
);
```

```
INSERT INTO orders (orderid, details) VALUES  
( 'ORD1001', 'Shipped via BlueDart. Expected delivery: 22 August 2026. Current location: Delhi Hub.'),  
( 'ORD1002', 'Out for delivery. Expected delivery: 19 August 2026. Delivery partner: Delhivery.'),  
( 'ORD1003', 'Order packed and ready for shipment. Courier: FedEx. Expected delivery: 24 August 2026.')  
ON DUPLICATE KEY UPDATE details = VALUES(details);
```

Following are the source code files

- Create a templates folder and place a file index.html
- In the main folder create app.py

In the main folder create requirements.txt

Code for index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Order Tracking Chatbot</title>
  <style>
    body{
      font-family: Arial, sans-serif;
      background: #f4f6f8;
      margin: 0;
      padding: 0;
    }
    .container {
      width: 600px;
      max-width: 90%;
      margin: 50px auto;
      background: white;
      border-radius: 12px;
      box-shadow: 0 4px 20px rgba(0,0,0,0.1);
      overflow: hidden;
    }
    .header {
      background: #222;
      color: white;
      padding: 20px;
      text-align: center;
    }
    .chat {
      height: 400px;
      overflow-y: auto;
      padding: 20px;
    }
    .message {
      margin: 10px 0;
      padding: 12px 15px;
      border-radius: 10px;
      max-width: 80%;
    }
    .user {
      background: #dbeafe;
      margin-left: auto;
    }
    .bot {
      background: #eeeeee;
      margin-right: auto;
```

```

    }
    .input-area {
        display: flex;
        padding: 15px;
        border-top: 1px solid #ddd;
    }
    input {
        flex: 1;
        padding: 12px;
        border: 1px solid #ccc;
        border-radius: 6px;
        font-size: 16px;
    }
    button {
        margin-left: 10px;
        padding: 12px 20px;
        background: #222;
        color: white;
        border: none;
        border-radius: 6px;
        cursor: pointer;
    }
    button:hover {
        background: #444;
    }
</style>
</head>
<body>
<div class="container">
    <div class="header">
        <h2> 📦 Order Tracking Chatbot</h2>
        <p>Powered by TinyLlama + Flask + MySQL</p>
    </div>

    <div id="chat" class="chat">
        <div class="message bot">
            Hello! 🙋 Please provide your order ID.
            For example: <b>ORD1001</b>
        </div>
    </div>

    <div class="input-area">
        <input type="text" id="message"
            placeholder="Enter your order ID..."
            onkeypress="handleKey(event)">
        <button onclick="sendMessage()">Send</button>
    </div>

```

</div>

<script>

```
async function sendMessage() {
  const input = document.getElementById("message");
  const message = input.value.trim();

  if (!message) return;

  addMessage(message, "user");
  input.value = "";
  addMessage("Checking your order...", "bot");

  try {
    const response = await fetch("/chat", {
      method: "POST",
      headers: {"Content-Type": "application/json"},
      body: JSON.stringify({message: message})
    });

    const data = await response.json();

    const chat = document.getElementById("chat");
    const messages = chat.getElementsByClassName("message");

    if (messages.length > 0) {
      messages[messages.length - 1].remove();
    }

    addMessage(data.response, "bot");
  } catch (error) {
    addMessage(
      "Sorry, something went wrong while checking your order.",
      "bot"
    );
    console.error(error);
  }
}

function addMessage(text, sender) {
  const chat = document.getElementById("chat");
  const div = document.createElement("div");
  div.className = "message " + sender;
  div.innerHTML = text;
  chat.appendChild(div);
  chat.scrollTop = chat.scrollHeight;
}
```

```
function handleKey(event) {  
  if (event.key === "Enter") sendMessage();  
}  
</script>  
</body>  
</html>
```

Code for app.py

```
from flask import Flask, render_template, request, jsonify  
import mysql.connector  
from transformers import AutoTokenizer, AutoModelForCausalLM  
import torch  
import re
```

```
app = Flask(__name__)
```

```
MODEL_NAME = "TinyLlama/TinyLlama-1.1B-Chat-v1.0"
```

```
print("Loading TinyLlama...")  
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)  
model = AutoModelForCausalLM.from_pretrained(  
    MODEL_NAME,  
    torch_dtype=torch.float32  
)  
print("TinyLlama loaded.")
```

```
def get_db_connection():  
    return mysql.connector.connect(  
        host="localhost",  
        user="root",  
        password="",  
        database="order_chatbot"  
    )
```

```
def extract_order_id(text):  
    pattern = r"\bORD[- ]?\d+\b"  
    match = re.search(pattern, text.upper())  
    if match:  
        return match.group(0).replace(" ", "").replace("-", "")  
    return None
```

```
def get_order_details(order_id):  
    connection = get_db_connection()  
    cursor = connection.cursor(dictionary=True)
```

```

query = """
    SELECT orderid, details
    FROM orders
    WHERE orderid = %s
    """

cursor.execute(query, (order_id,))
result = cursor.fetchone()

cursor.close()
connection.close()
return result

```

```

def generate_response(user_message, order_id, details):
    prompt = f"""<|system|>
You are a helpful order tracking chatbot.
Answer the customer's question using only the shipping information provided.
Do not invent shipping information.
Keep the answer short and friendly.
</s>
<|user|>
Customer message: {user_message}

Order ID: {order_id}

Shipping details:
{details}

Give the customer a helpful response.
</s>
<|assistant|>
    """

```

```

inputs = tokenizer(prompt, return_tensors="pt")

with torch.no_grad():
    outputs = model.generate(
        **inputs,
        max_new_tokens=500,
        temperature=0.3,
        do_sample=True,
        pad_token_id=tokenizer.eos_token_id
    )

response = tokenizer.decode(outputs[0], skip_special_tokens=True)

if "<|assistant|>" in response:

```

```

        response = response.split("<|assistant|>")[-1]

    return response.strip()

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/chat", methods=["POST"])
def chat():
    data = request.get_json()
    user_message = data.get("message", "").strip()

    if not user_message:
        return jsonify({"response": "Please enter your order ID."})

    order_id = extract_order_id(user_message)

    if not order_id:
        return jsonify({
            "response": "Please provide a valid order ID, for example ORD1001."
        })

    try:
        order = get_order_details(order_id)
    except mysql.connector.Error as e:
        return jsonify({
            "response": "Database connection error. Please check your MySQL settings."
        }, 500)

    if not order:
        return jsonify({
            "response": f"I couldn't find order {order_id}. Please check the order ID and try again."
        })

    response = generate_response(
        user_message,
        order["orderid"],
        order["details"]
    )

    return jsonify({
        "response": response,
        "orderid": order["orderid"],
        "details": order["details"]
    })

```

```
if __name__ == "__main__":  
    app.run(host="0.0.0.0", port=7860, debug=True)
```

Code for requirements.txt

```
Flask  
mysql-connector-python  
transformers  
torch  
accelerate  
sentencepiece
```

In the terminal of pycharm

Write

```
ollama pull tinyllama
```

and create database and tables in mysql with the mysql code given above

Run the project and it will show you the output

