**A detailed report on NETWORK SECURITY CS 5342**

**PROJECT ROUND 2**

# OnDemand Professor Q&A Bot

**Ying Liu**

**(Associate Professor)**

Submitted by

**R11904751**

**Aparna Vempati**

**GitHub Url : https://github.com/cherupallybhanuteja/Q-A\_Bot\_NetworkSecurity**

# Project Objectives:

Our objective is to create a Q&A Bot that focuses on offering information and addressing queries regarding network security courses. To protect data privacy we prioritize using open source alternatives to ChatGPT that can function directly on a users device. This way all data, including course related details and the Q&A model remain securely stored and processed locally. By adopting this approach we eliminate the necessity of transmitting information, over the internet thereby reducing privacy risks.

# Tools & Technologies:

**Programming Languages**:

**Python**: One of the most prevalent backend language used today due to its ease of compatibility with AI libraries and data processing.

**JavaScript, HTML, and CSS**: Owed to the need of an attractive and responsive front end, this is also used for the development of web applications.

# User Interface:

**Flask**: Takes care of the webserver configurations, petitions and business takes care of the frontend routing, allowing an easy question – answer conversation with the bot.

# Vector Database:

**Chroma Database**: Well indexed documents stored as embeddings in a database to facilitate fast document search based on similarity and to ensure relevant course material is retrieved.

# Natural Language Processing (NLP):

**Hugging Face Embeddings**: Used to generate document embeddings that are quite a form of ranking-efficient class of deep convolution dynamics as well as attention based word similarities which can be utilized to assist the bot in retrieving topically related information.

**Groq API (Llama Model)**: Used in creating language model, that is, in creating given questions and providing corresponding answers. There are certain chat settings such as temperature and max\_length aimed at enhancing the output, that is, the output accuracy.

# Document Handling:

**PyPDFLoader**: Comprises some functions and kernel classes for Reader. Additionally, it includes functions like ConvertToPDF methods that enable simple reading and indexing of a document.

**PyPDF2**: enhancement for manipulation and parsing of clear text in PDF files. It decodes and retrieves text from a provided source document and prepares to save it in AudioManager.

# Embedding and Retrieval Framework:

**Sentence-transformers/msmarco-distilbert-base-v4**: A service implemented by Hugging. Provides a service where it is enabled to convert the textual content of a document into vectors thereby making it easy for the retrieving of similar documents.

# Error and Performance Management:

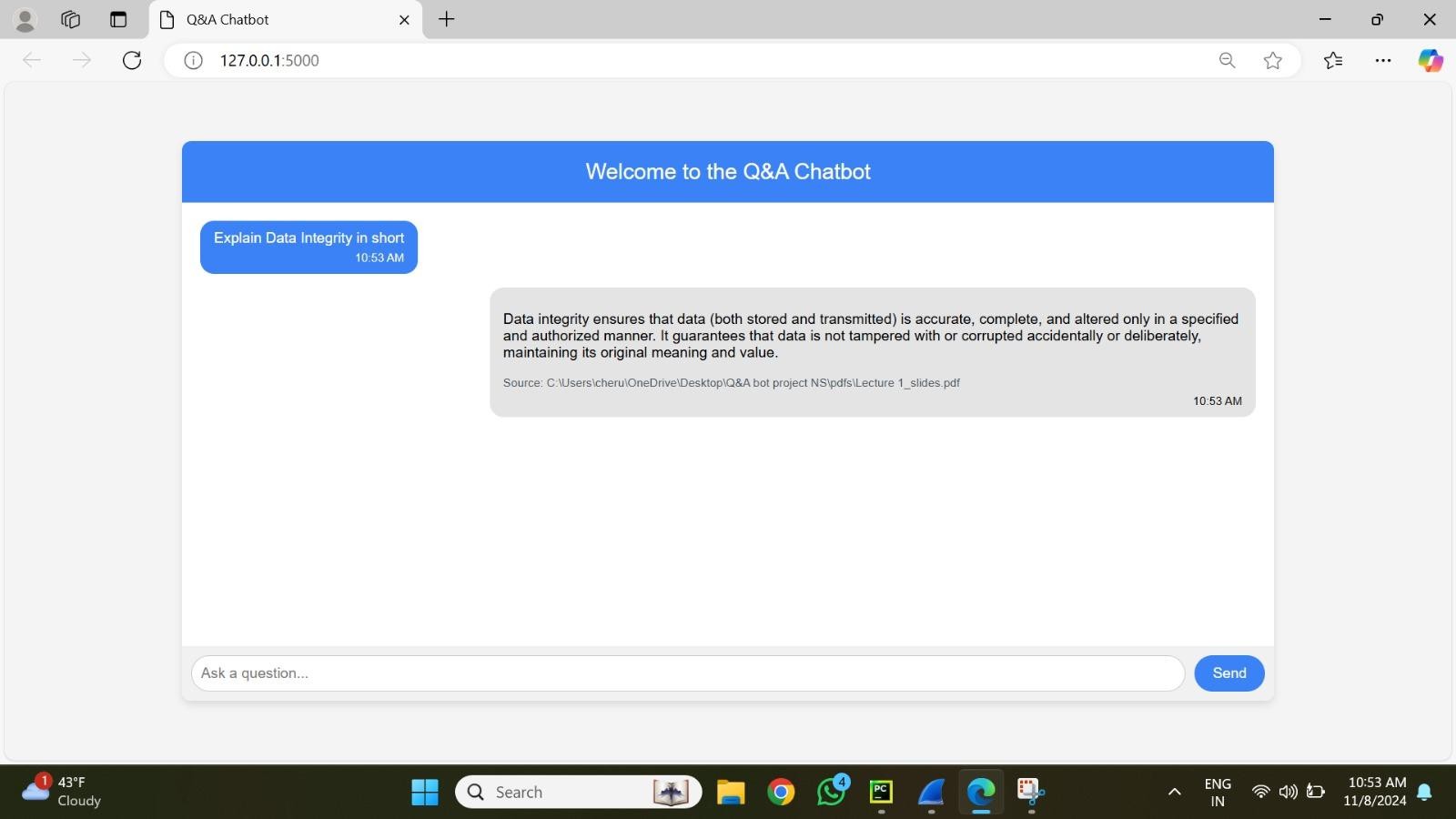
**Logging and Console Monitoring**. Monitor errors with the help of backend splurging and JavaScript console. It becomes more critical when dealing with JavaScript, HTML tags and requests on the net.

**Optimization Techniques**: For instance, in managing the surface of the Chroma database and working with large document sets. It results in effective utilization of data processing practices.

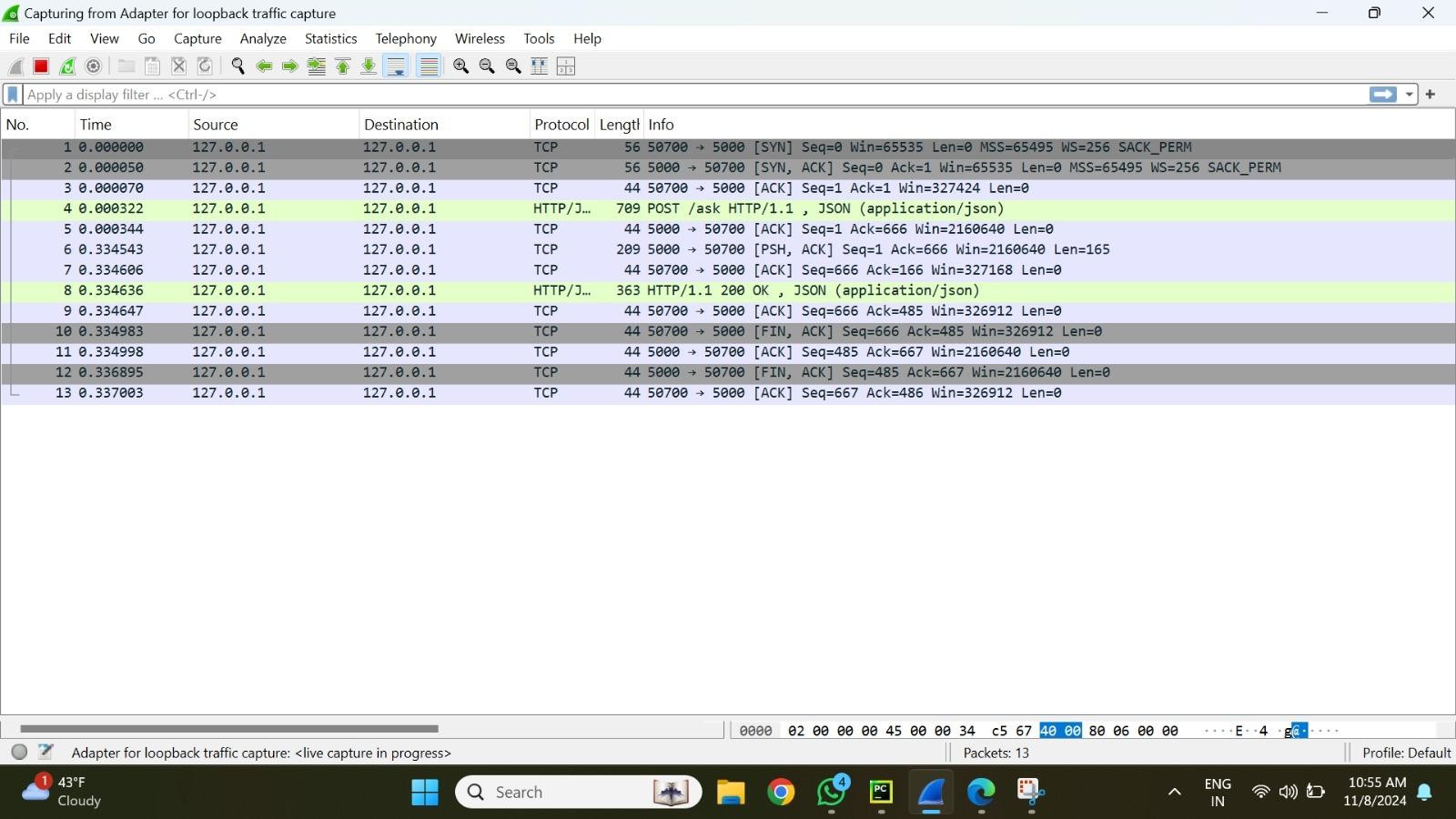
# Documentation:

The interactive search bot is a product that is trained on documents about network security, equipped to get precise and quick responses every time creators ask questions as for the matter.

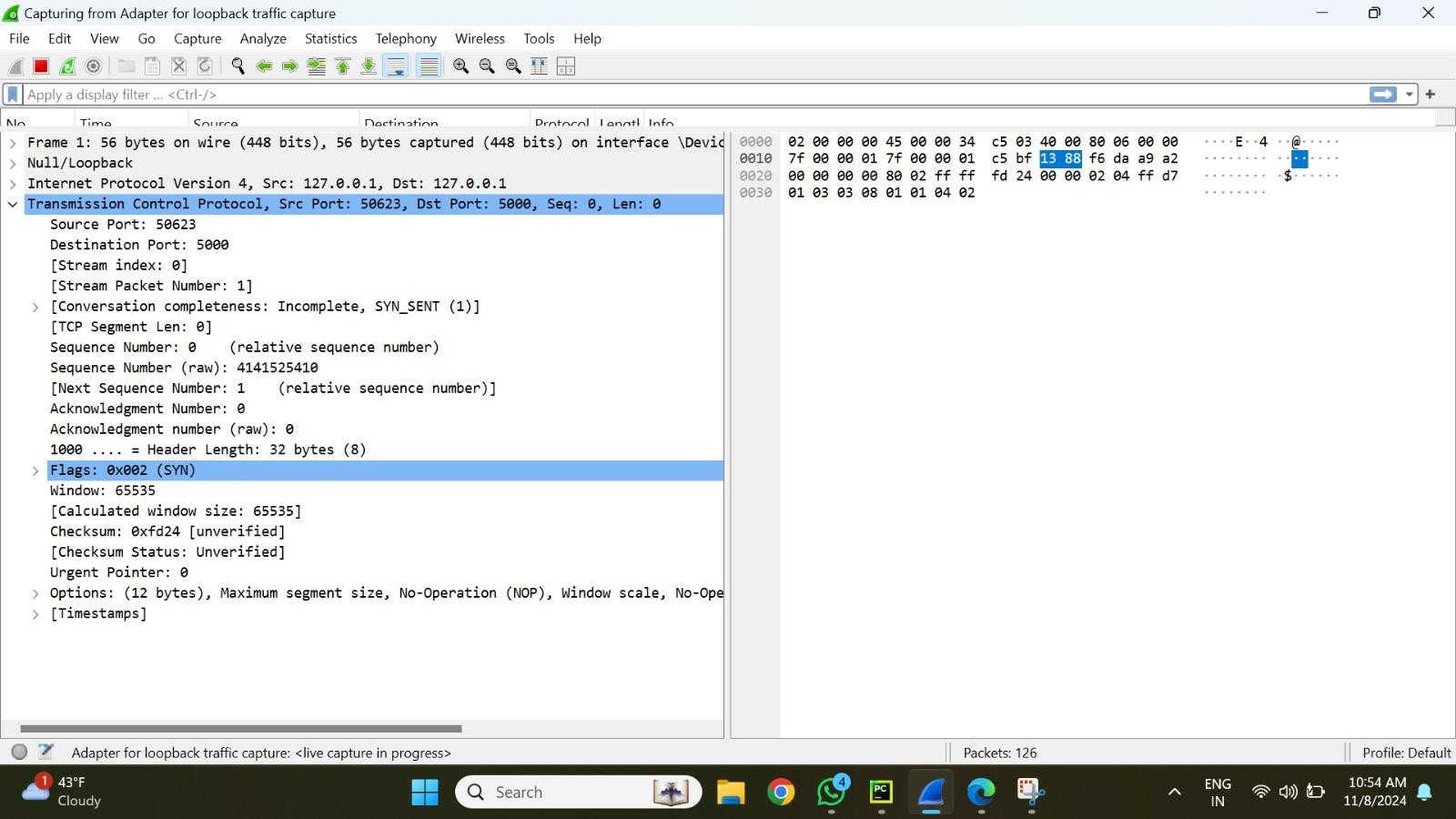
Five such queries were provided along with screen captures showing the bot’s responses. Moreover, network packets with the incoming and outgoing network packets are also shown here using snapshot from Wireshark packet tracer.

**PROMPT 1:Explain data integrity in short?**

After the specified command starts the process in the web-based client, the chatbot—a piece of software designed for chatting—is responsible for replicating the operation locally. The query that will be used to get patterns from the database's pattern base, Chroma DB, is encoded during the first processing stage. In the real retrieval stage, the matched vectors are used to enable the growing language apparatus ChatGPT.   
  
During its execution, Wireshark can be used to view this interaction. Source addresses and content are displayed along with the incoming requests and the associated TCP packets sent in Wireshark captures. The chatbot runs on localhost (127.0.0.1), and the client application runs on port 8000. When the command is sent through the local port 8000, the packet shown by Wireshark is sent to port 61920 as seen on the snapshot.



# Image 2



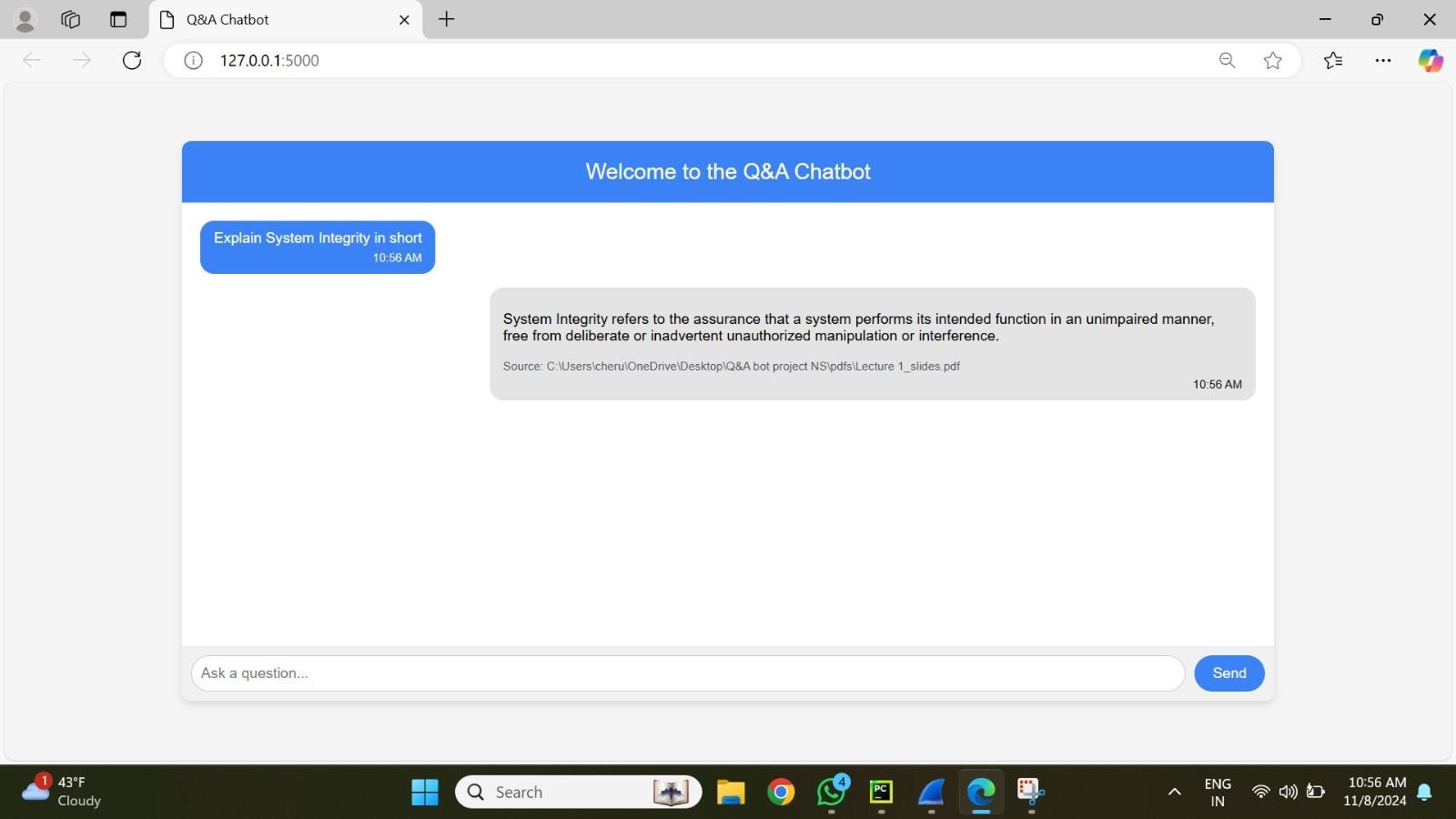
**Image 3**

We can see all the information related to the prompt request including the source and destination port and header information and time at which the packet is sent in the Wireshark.

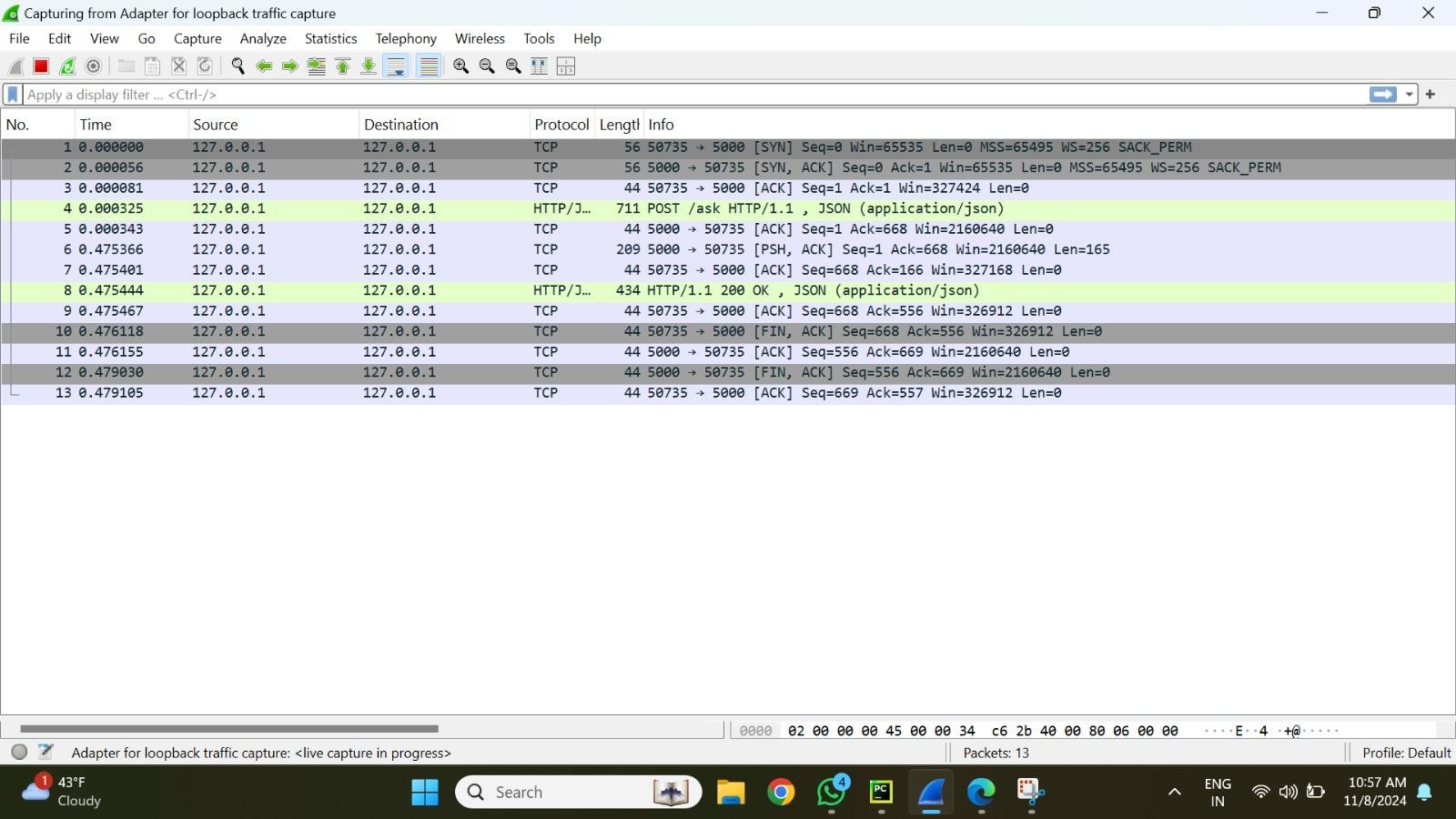
It uses TCP protocol and sends an ACK packet to the source

The second image shows a Wireshark capture of packets involved in the TCP traffic flow for establishing a connection.

* **TCP Handshake Initiation:** The SYN flag indicates the start of the TCP handshake, which initiates communication between source port 50623 and destination port 5000.
* **Data Integrity Assurance:** By facilitating precise packet delivery to the designated address, the handshake guarantees data integrity.
* **Header Elements:** In order to detect any missing or modified packets in real time, sequence numbers and other header information assist guarantee that packets arrive at their intended location.
* **TCP Packet Headers and Flags:** SYN, ACK, and FIN are among the flags found in multiple TCP headers that help ensure consistent communication.
* **HTTP 200 OK Confirmation:** An HTTP/1.1 200 OK response verifies that the data was successfully retrieved from the server and shows that the data was sent unchanged.
* **Ordered Delivery:**  By guaranteeing packet delivery in the proper sequence, the configuration maintains data integrity and verifies the legitimacy of the information.
* **Integrity Verification:**  The packet capture enhances integrity checking by shielding data from unintentional changes or rearranging while it is being transmitted.   
    
  Next, using the mapping vectors, the chatbot retrieves the data and feeds it into the large language model.   
    
  After receiving the embedding vectors and creating a response using the chatgpt model, the LLM sends the response back to the user interface via the localhost 8000 port.

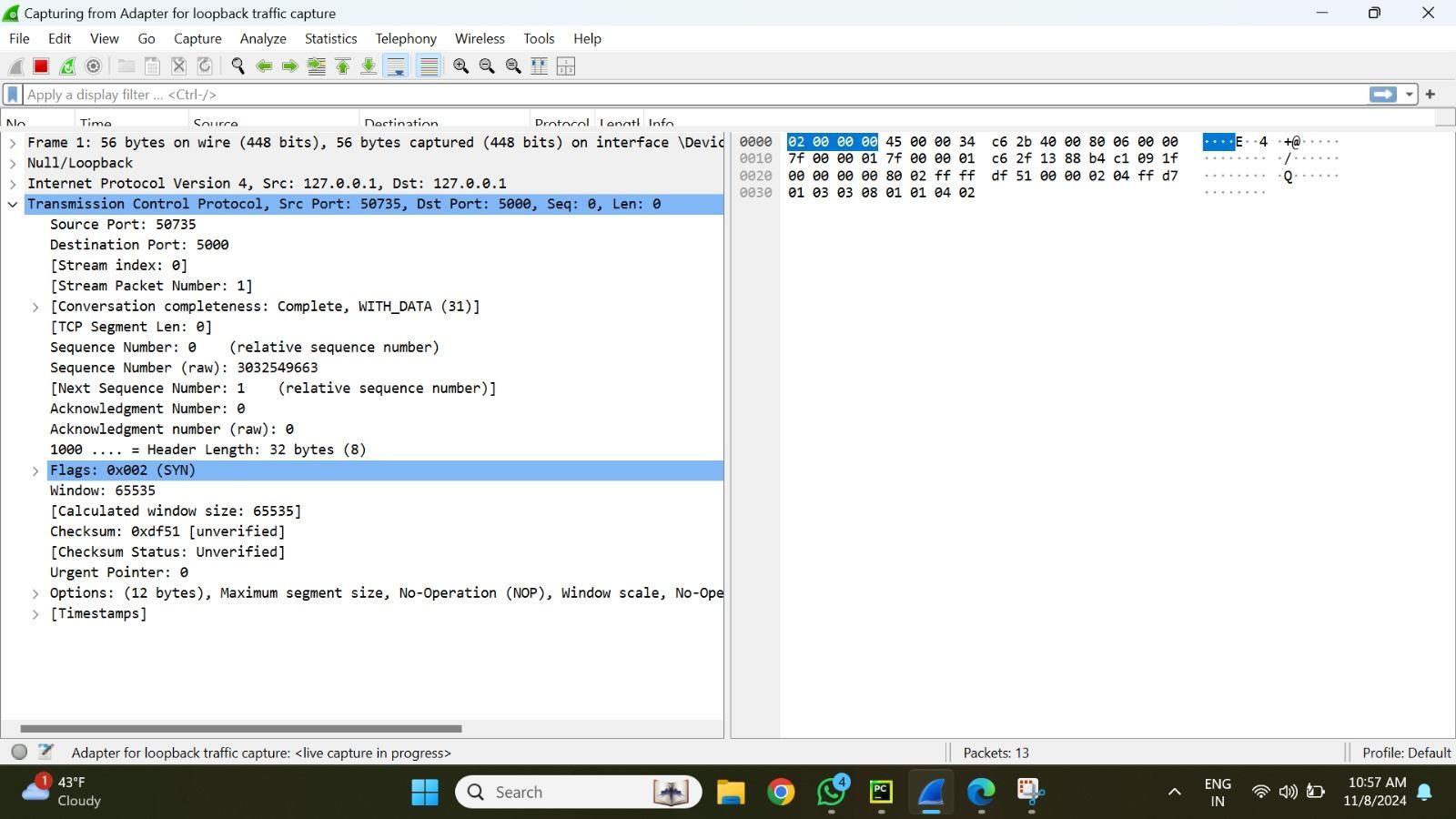
**PROMPT 2:Explain System Integrity in short**

* The user and chatbot are shown interacting in this picture. "Explain System Integrity in short," the user asks, and the chatbot provides a succinct explanation. The statement clarifies that system integrity guarantees that a system operates as intended, free from intervention or illegal modification. The user can clearly grasp the significance that system integrity plays in preserving a system's dependability and security thanks to this response. A basic description of the idea is provided by the chatbot's response, which was taken from the Stallings Textbook.



The packets related to processing this prompt are captured by Wireshark in the second image.

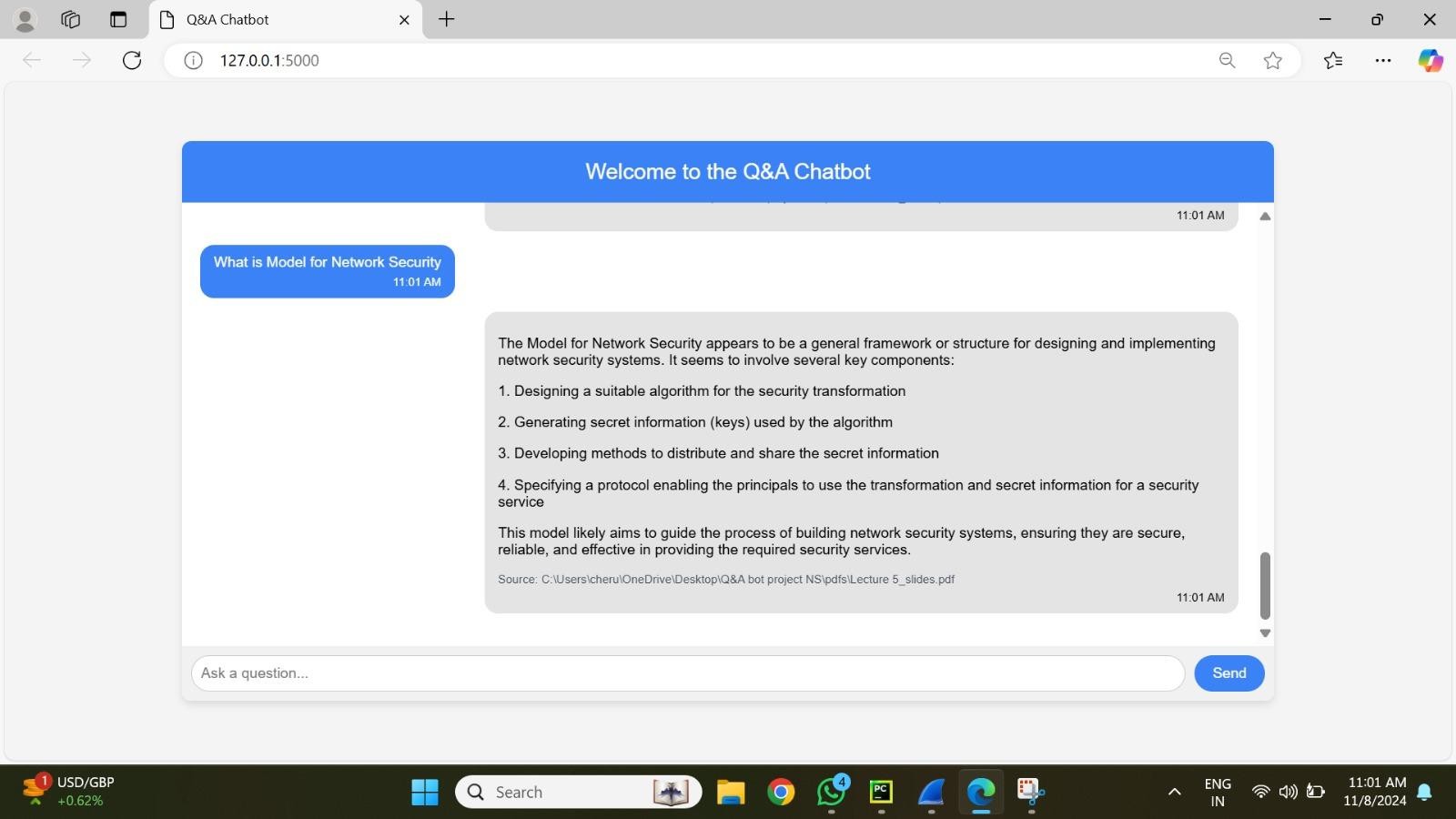
* Data is routed only within the local environment, as confirmed by the packet trace, which shows the data flow from the local server (127.0.0.1) at source port 5000 to destination port 50735 and vice versa.
* In order to establish the TCP connection, the packets show the following sequence of actions: SYN, SYN-ACK, and ACK. Data packets that represent the prompt and the chatbot's response follow.
* TCP is the protocol that was captured, and the data flow contains header information like sequence and acknowledgment numbers, guaranteeing dependable prompt and answer delivery inside the local configuration.



A close-up of one of the TCP packets may be seen in the third image. Ports 50735, 5000, and 50335 are among them.

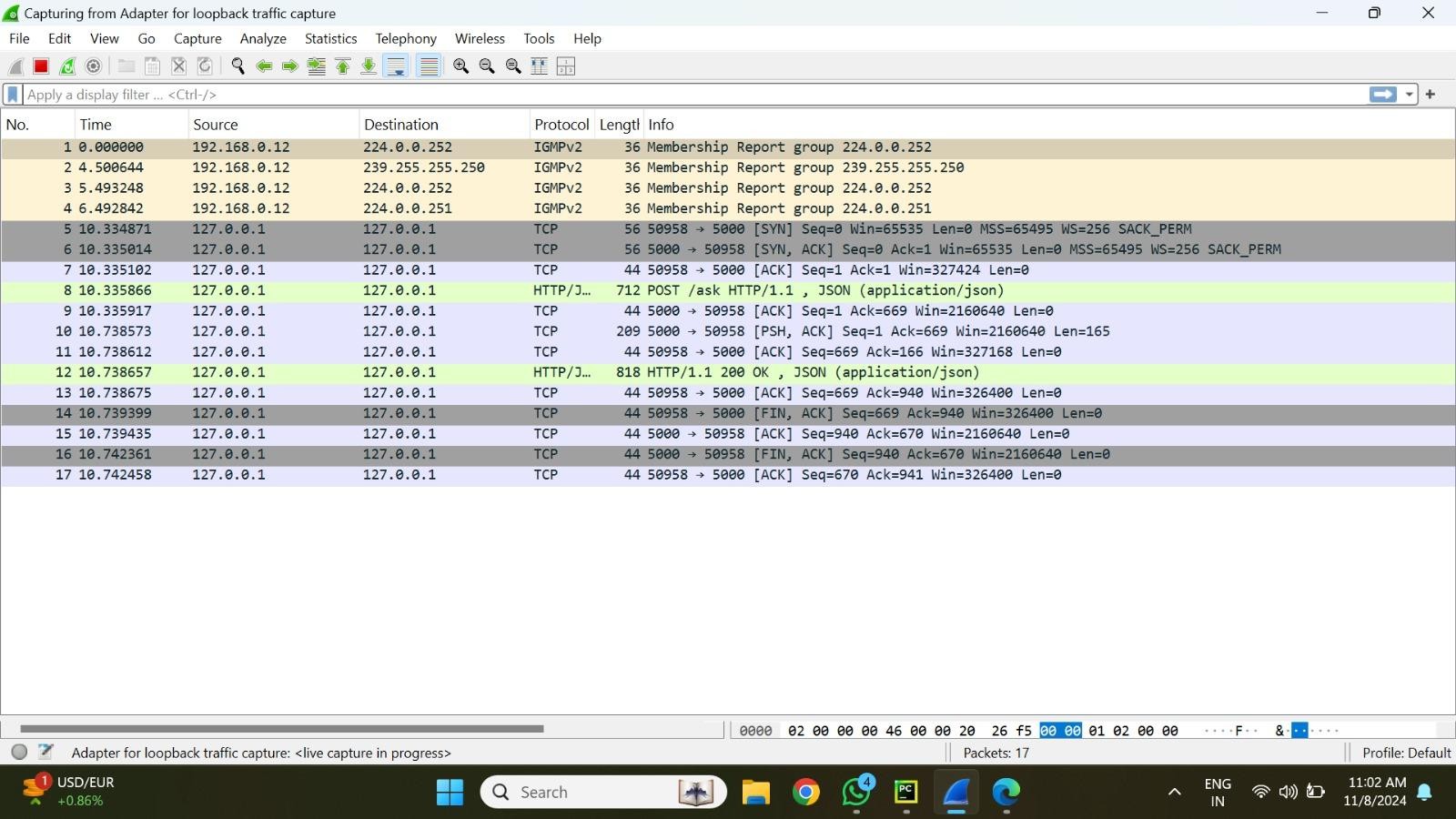
* SYN is a flag that signals the beginning of a new connection.   
  In order to keep track of and verify that packets are received in the correct order, sequence and acknowledgment numbers are useful.
* The buffer capacity is indicated by the window size of 65535.
* The checksum serves as an indicator of the packet content's data integrity.
* The address of the source is 127.0.0.1, while the address of the destination is 127.0.0.1.
* The initial SYN packet's sequence and acknowledgment numbers are 0 and 0, respectively.
* Time to Live, or TTL,: 64

After generating the embedding vector using resource documents and the prompt, the LLM employing the chatgpt model creates a response that is once more delivered to the localhost 8000 port back to the user interface

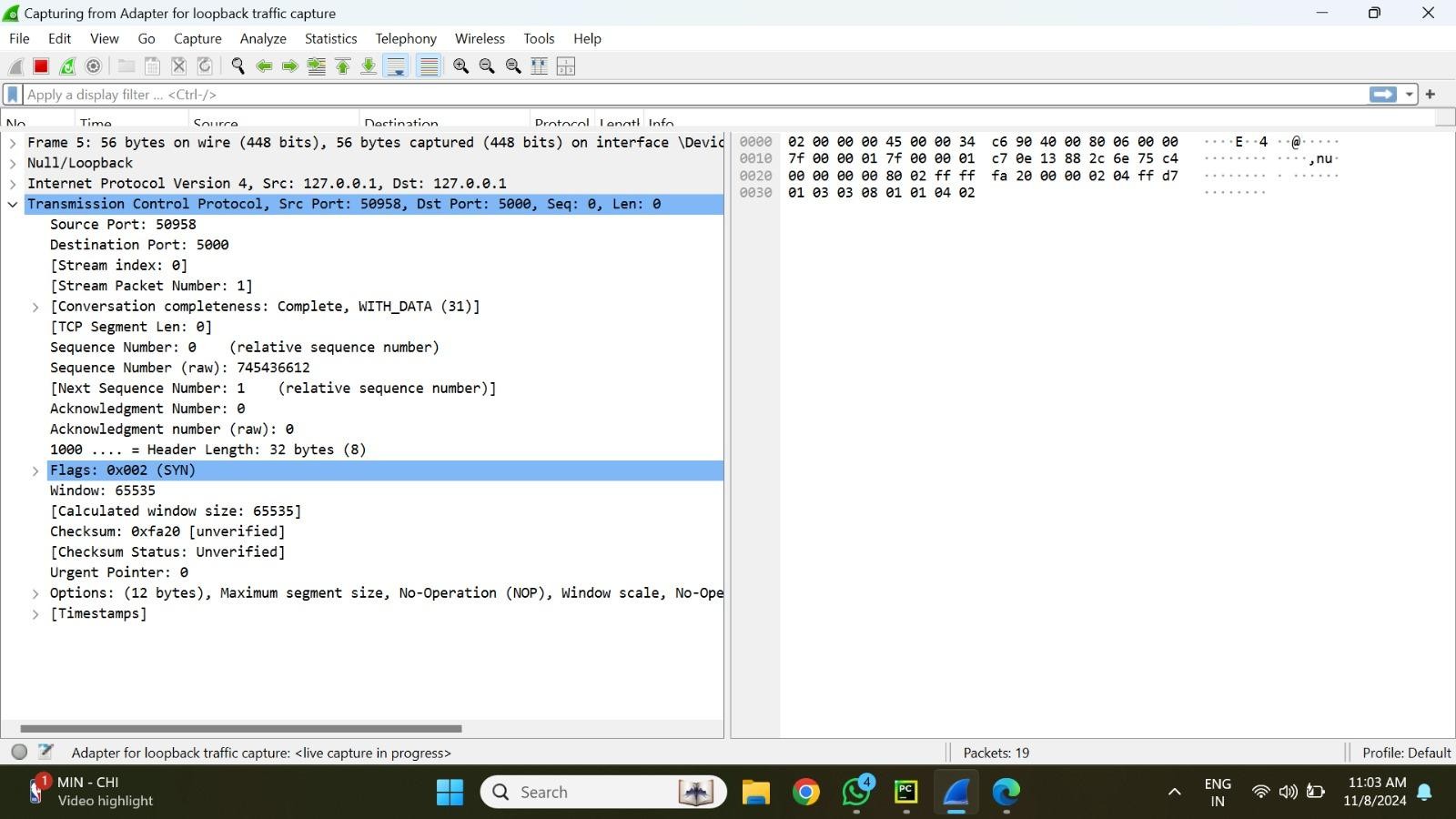
**Prompt 3:What is Model for Network Security?**

The image shows an interaction between a user and a chatbot, where the user asks, "What is Model for Network Security?" The chatbot provides an answer that outlines the Model for Network Security as a general framework used in designing and implementing network security systems. The model includes the following key components:

* **Designing a Suitable Algorithm** for security transformation.
* **Generating Secret Information (Keys)** used by the algorithm.
* **Developing Methods** for distributing and sharing the secret information securely.
* **Specifying a Protocol** that allows authorized users to use the security transformation and secret information for secure communication.
* The model is intended to guide the development of network security systems that are secure, reliable, and effective in providing necessary security services.



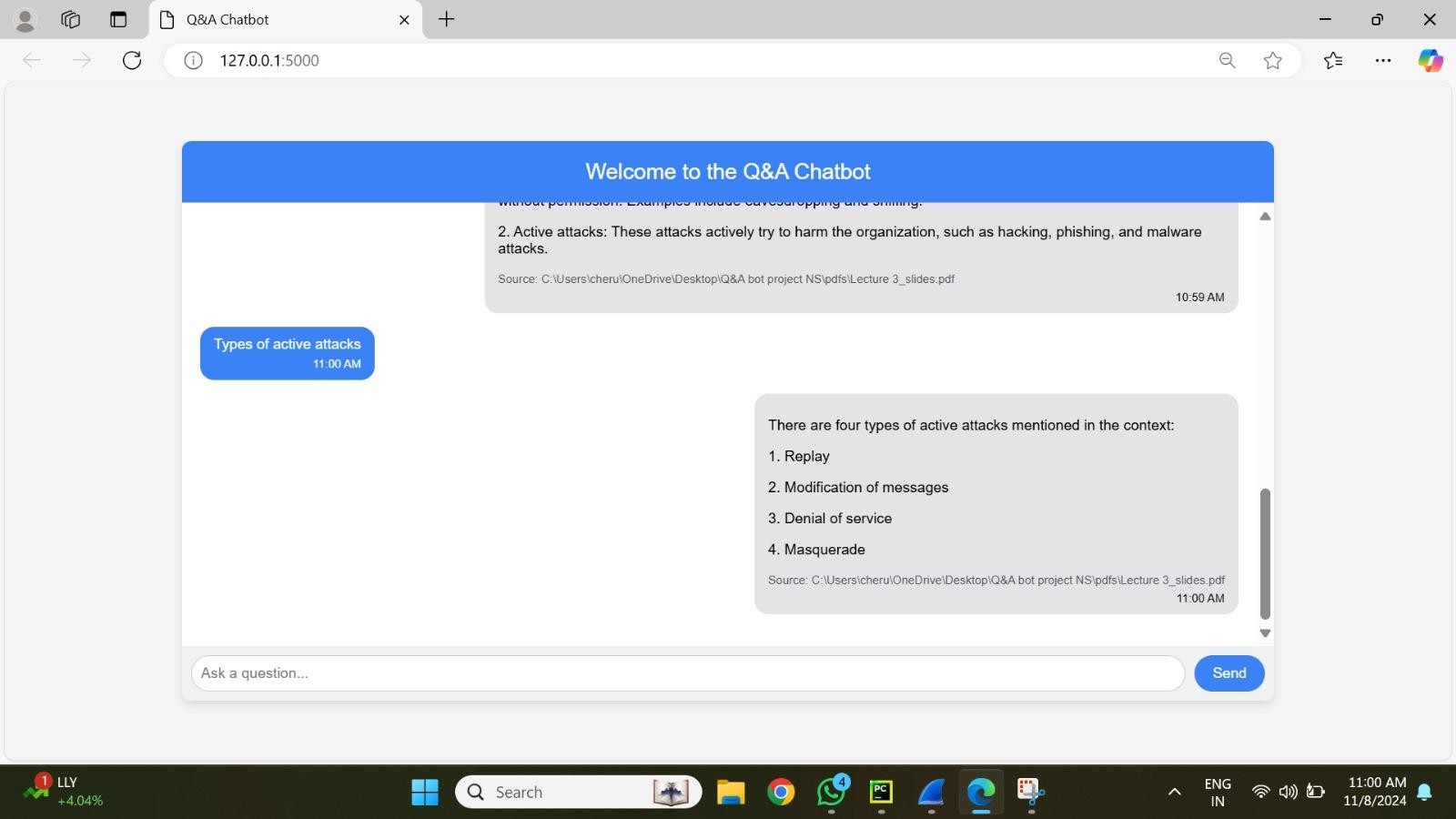
A Wireshark capture of the network packets related to the prompt processing is displayed in this image.

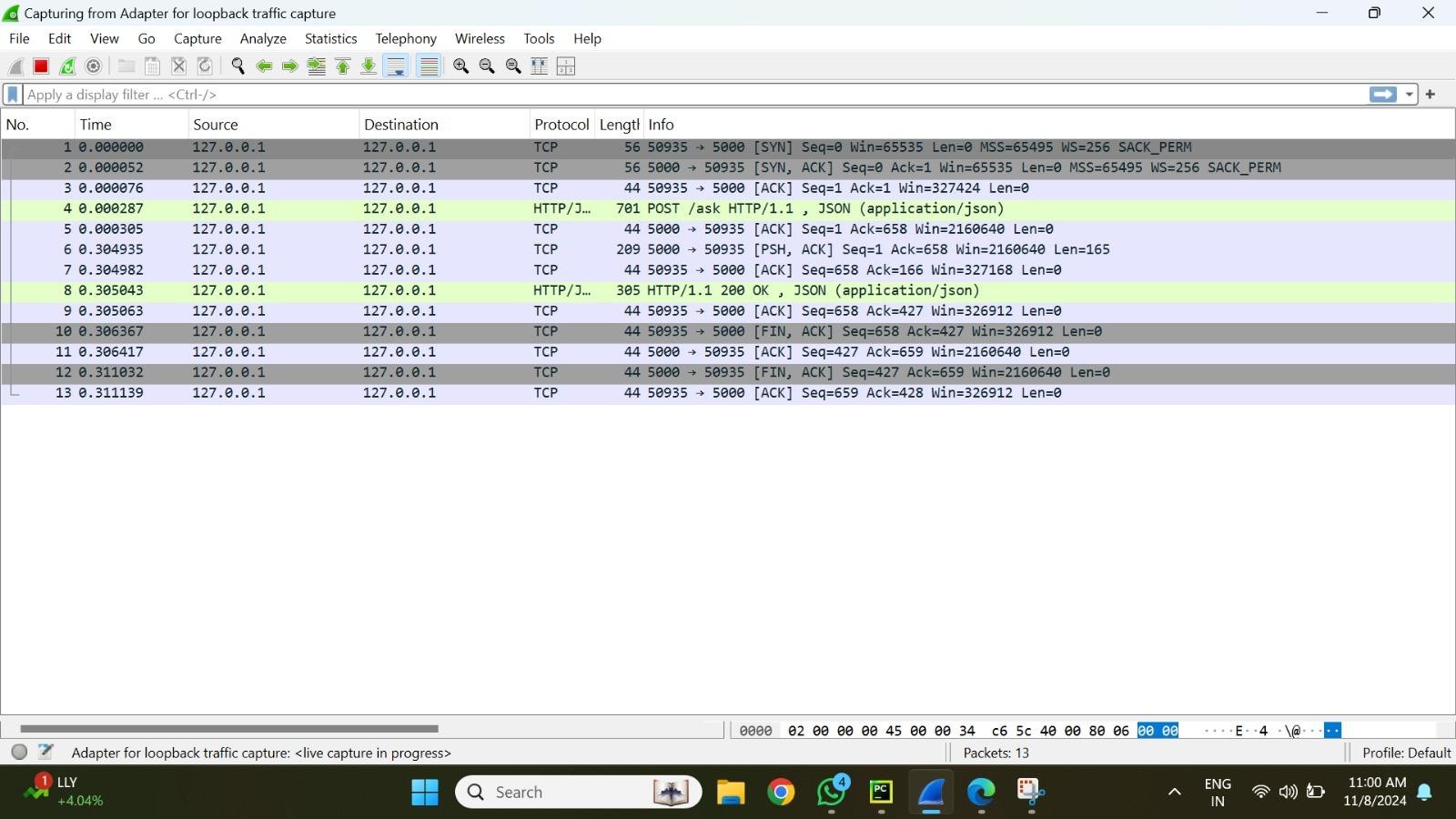
* The packets that were recorded show how data moves through a local environment (127.0.0.1).
* Traffic moves from source port 5000 to destination port 50735 and vice versa.
* The TCP connection is established through a sequence of actions:
* SYN (synchronize) packet
* SYN-ACK (synchronize-acknowledge) packet
* ACK (acknowledge) packet
* After the connection setup, data packets are exchanged, representing:
* The prompt input
* The chatbot's response
* The TCP protocol ensures ordered and reliable data transfer, using sequence and acknowledgment numbers to track each packet and prevent loss or duplication.
* HTTP POST requests and HTTP 200 OK responses indicate successful data exchange within the local testing setup.
* This setup confirms that both the prompt and the response were correctly routed between the client and server within the system.

An in-depth look at one of the TCP packets is displayed in the third figure. This comprises:

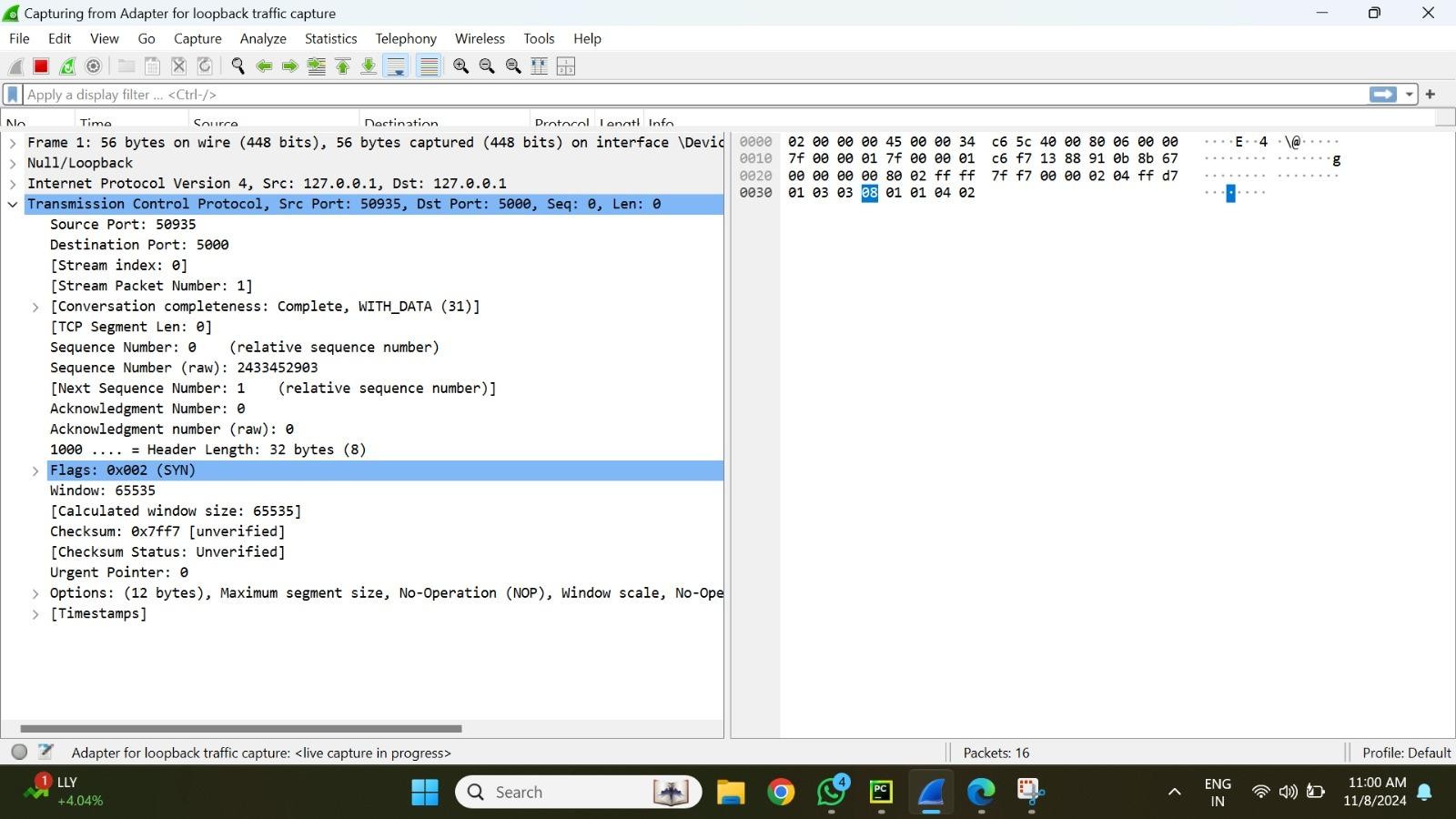
Port of origin: 50735, port of destination: 5000.   
SYN flags signify the beginning of a new connection   
The Order and Recognition Numbers: These aid in tracking and verifying that packets arrive in the correct order.   
This window size, which displays the buffer's capacity, is 65535.   
Checksum: Shows how reliable the packet's data is.   
ChromaDB receives the data packet, embeds the "Types of active attacks" prompt, and compares it with document vectors that have been saved. After receiving this matching data, the ChatGroq (Llama Model) produces a response that includes a summary of the four categories of ongoing attacks along with citations to their sources. Based on the course contents, this procedure guarantees a precise and pertinent response.

# PROMPT4:Types of active attacks



* An interaction between a user and a chatbot is captured in this photograph. In response to the user's question regarding "Types of active attacks," the chatbot provides a broad answer. It defines active attacks—such as malware, phishing, and hacking—as those that make a concerted effort to damage an organization. The user learns from the chatbot's response that active assaults entail direct interference or manipulation of data, whereas passive attacks concentrate more on illicit data access without immediately changing or destroying systems. This first reaction lays the groundwork for comprehending particular kinds of aggressive attacks.

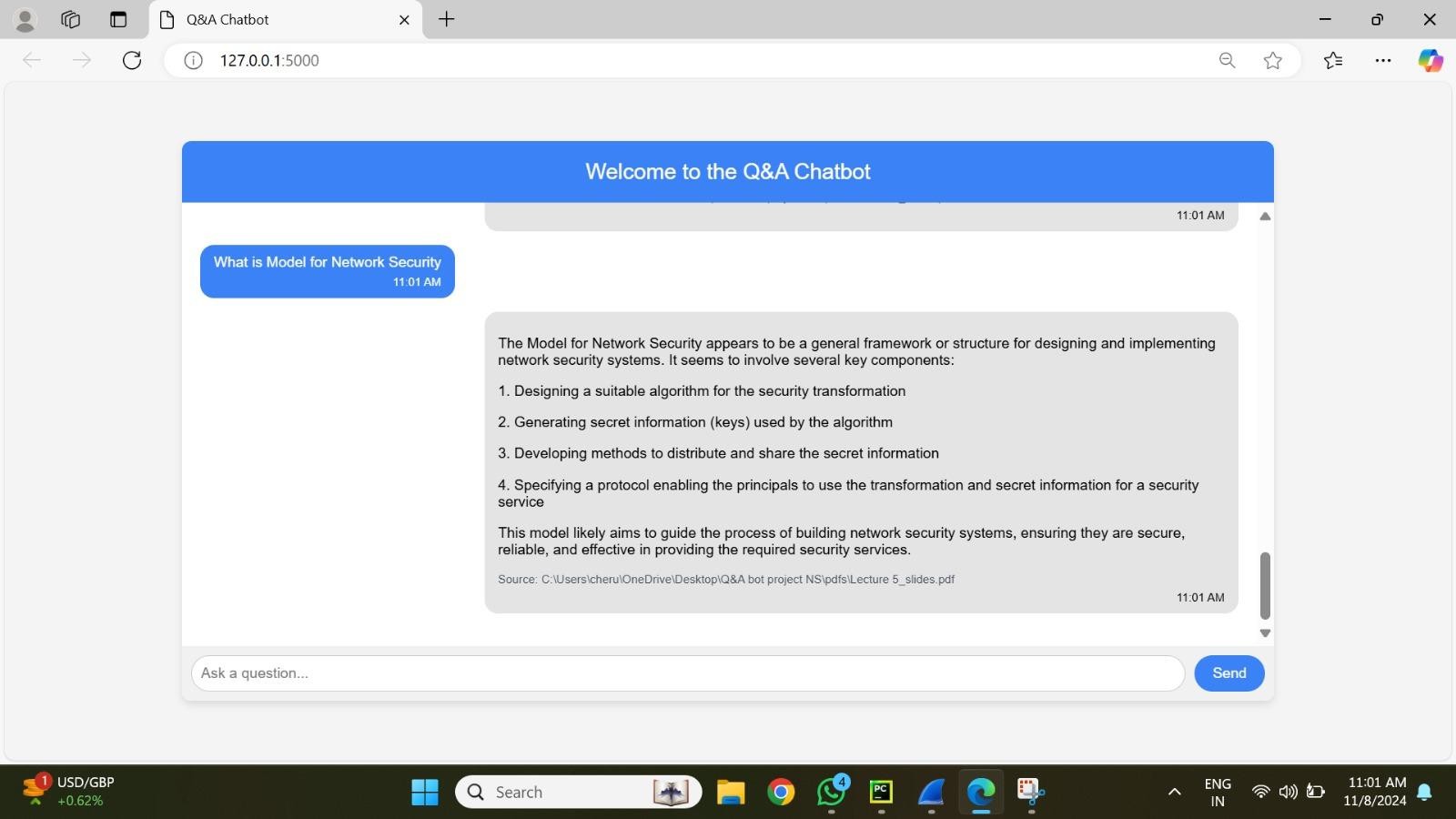
In the second image, a Wireshark capture of network packets pertaining to the request and response between the user interface and the chatbot is displayed.   
  
The IP addresses of the source and destination are both set to 127.0.0.1, indicating that data is flowing locally.   
In order to demonstrate interaction within the local machine, traffic flows from port 5000 to port 50735 and vice versa.   
● Protocol: Data transfer is accomplished by TCP.   
In order to establish a connection, the packets use the three-way TCP handshake, which consists of SYN, SYN-ACK, and ACK.   
In order to verify data interchange, HTTP POST requests and HTTP 200 OK answers are included.   
Verification of Successful Exchange: The sequence demonstrates that the response was processed appropriately and that the data exchange between the chatbot and the user interface is dependable.



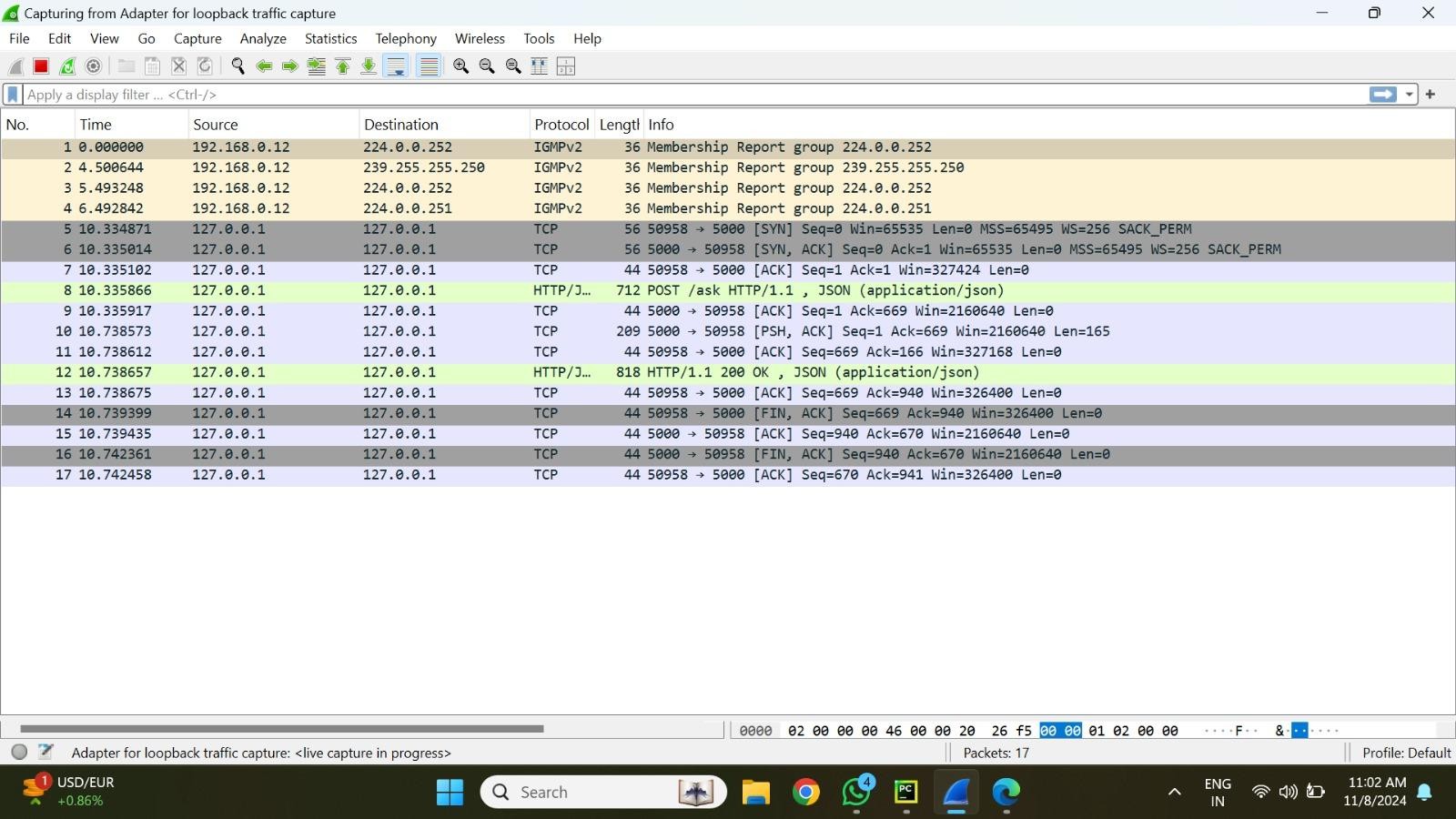
The third image provides a close-up view of a single TCP packet within the traffic capture.

* Source Port: 50735
* Destination Port: 5000
* SYN Flag: Indicates the start of a new connection.
* Sequence Number and Acknowledgment Number: Both set to 0, showing it is the initial packet in the handshake.
* Window Size: 65535, indicating the buffer capacity for receiving data.
* Checksum: Used to verify data integrity within the packet.
* Hexadecimal Representation: Displays the packet content in a low-level format, showing the data being transmitted.
* This packet detail highlights the structure and reliability of data transmission within the local environment, ensuring secure communication between the chatbot system and the user interface.

## PROMPT5:What is Model for Network Security?

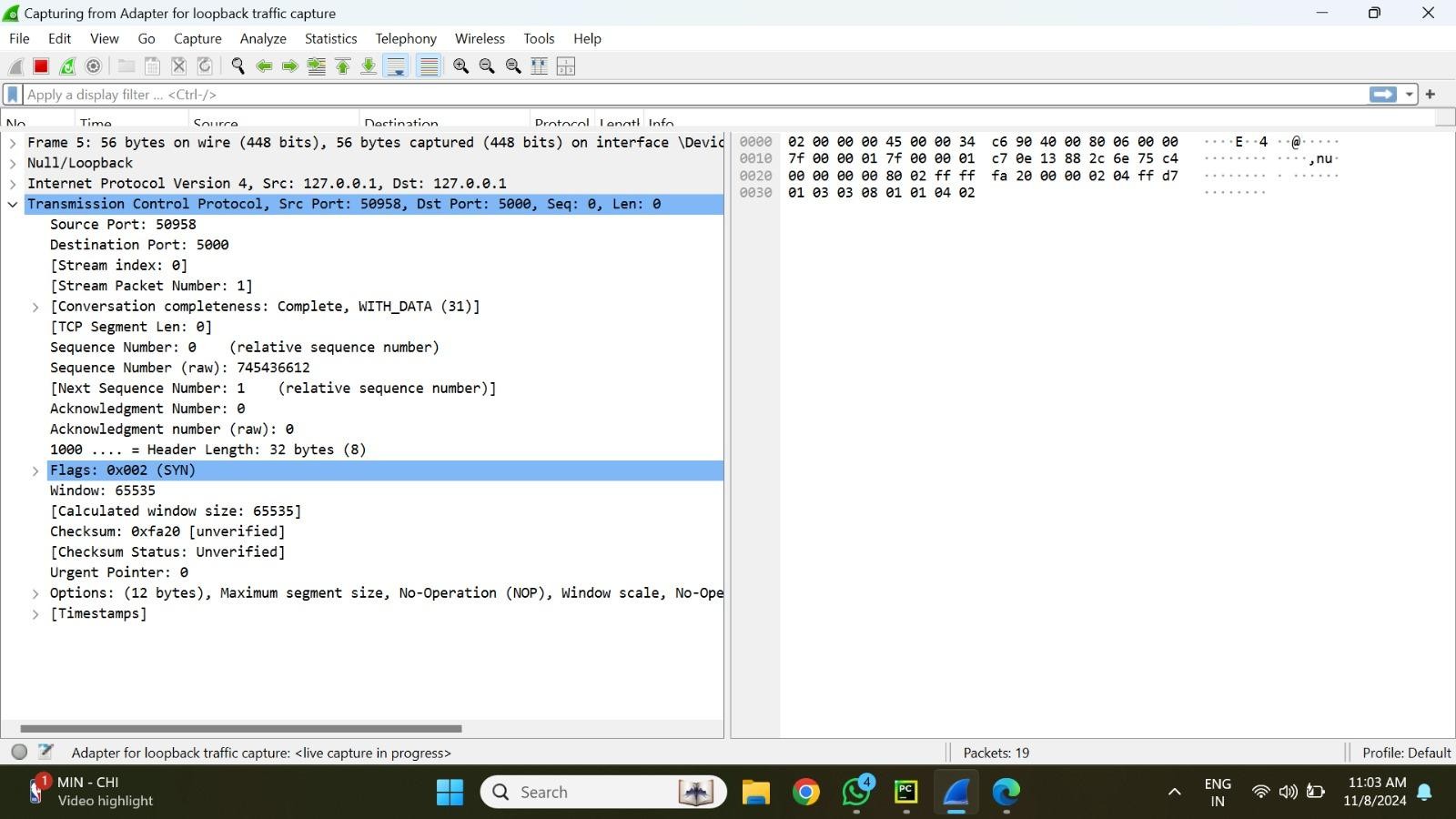


In this picture, a user asks a chatbot, "What is Model for Network Security?" In a thorough response, the chatbot explains that the Model for Network Security is a general framework for creating and putting into practice security solutions. It lists four primary elements:   
1. Creating an appropriate algorithm for security transformation.   
2. Producing the algorithm's secret data (keys).   
3. Creating strategies to disseminate and exchange confidential data.   
4. Outlining a methodology for securing communication with the transformation and secret data.



When the chatbot explains the network security paradigm, the second image displays a Wireshark capture of the network packets related to the request and response.

* + **Initial Traffic:** Multicast traffic (IGMPv2) for group membership reports initiates the collection.
  + **Local Traffic:**  TCP packets with source and destination IP addresses set to 127.0.0.1 follow IGMPv2 packets, signifying local communication.
  + **Ports:** Information moves from port 5000 to port 50958, indicating a local machine contact.
  + **Protocol:** TCP is the protocol used for reliable data transmission.
  + **TCP Handshake:**  To establish a connection, the packets use the standard three-way handshake (SYN, SYN-ACK, ACK).
  + **HTTP Requests and Responses:** Successful data transmission is shown by the visibility of HTTP POST requests and HTTP 200 OK responses.
  + **Confirmation of Successful Exchange:** Within the local setup, the sequence verifies that the user's request and the chatbot's response were correctly processed and communicated.



A close-up of a single TCP packet within the grab is shown in the third picture. The packet is a component of the chatbot's response flow because it has source port 50958 and destination port 5000. Important information is as follows:

● Flags: SYN, which indicates the start of a new connection.

* Since it is the first packet in the handshake,
* the sequence number is 0.
* The window size, which displays the buffer capacity, is 65535.

● Checksum: Verifies the packet's data integrity.

An overview of the sent data is provided by the hexadecimal representation of the packet content, which is shown on the right.

## Conclusion

* All things considered, this paper offered a thorough analysis of the network security model and the fundamental processes necessary to create safe communication channels. The significance of protocols like the TCP handshake in guaranteeing data integrity, dependability, and orderly delivery in local contexts was investigated by examining TCP traffic samples. The network security framework was also examined, with particular attention paid to protocol formulation, secure information dissemination, key generation, and algorithm design. All of these components work together to support the deployment of strong security measures that safeguard data while it is being transmitted.