**Q&A CHAT BOT ROUND 2**

**YAMINI KUSA**

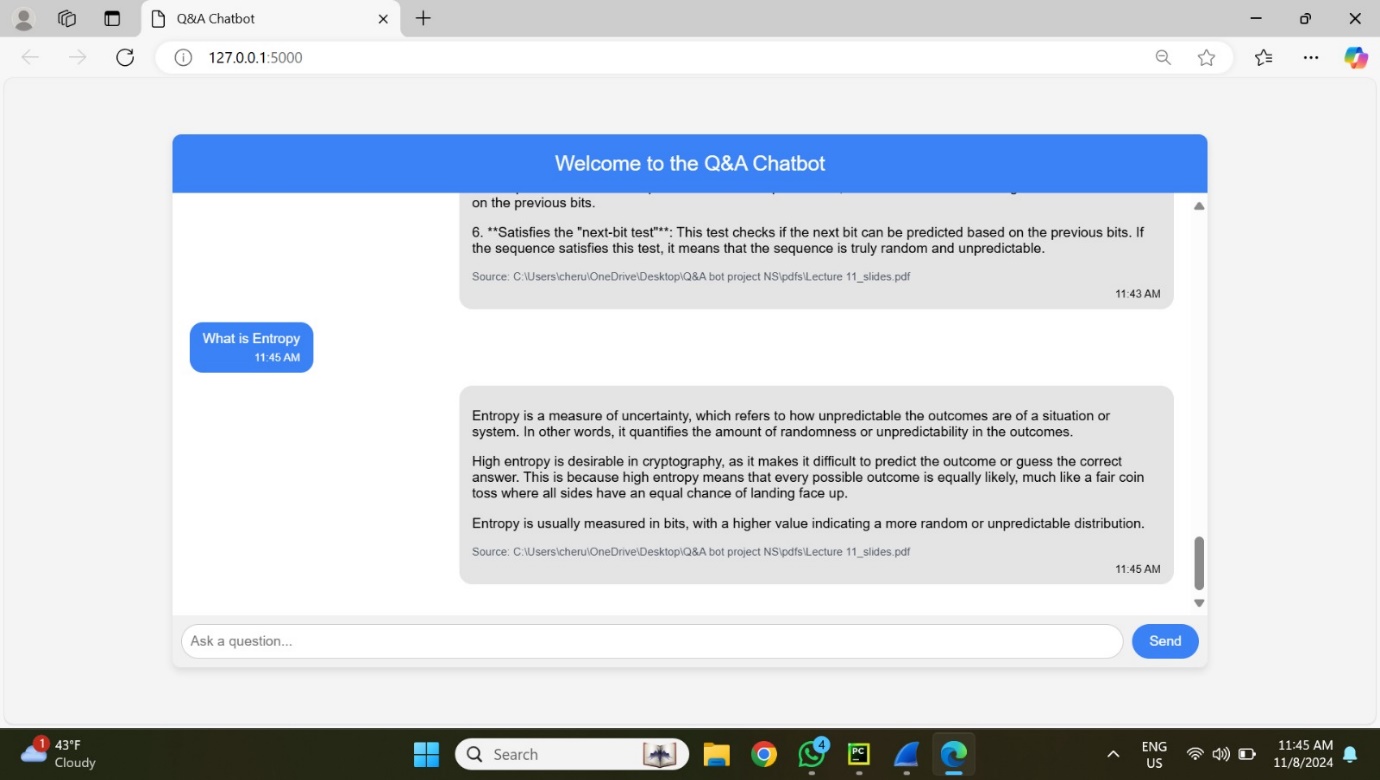
**R11898938**

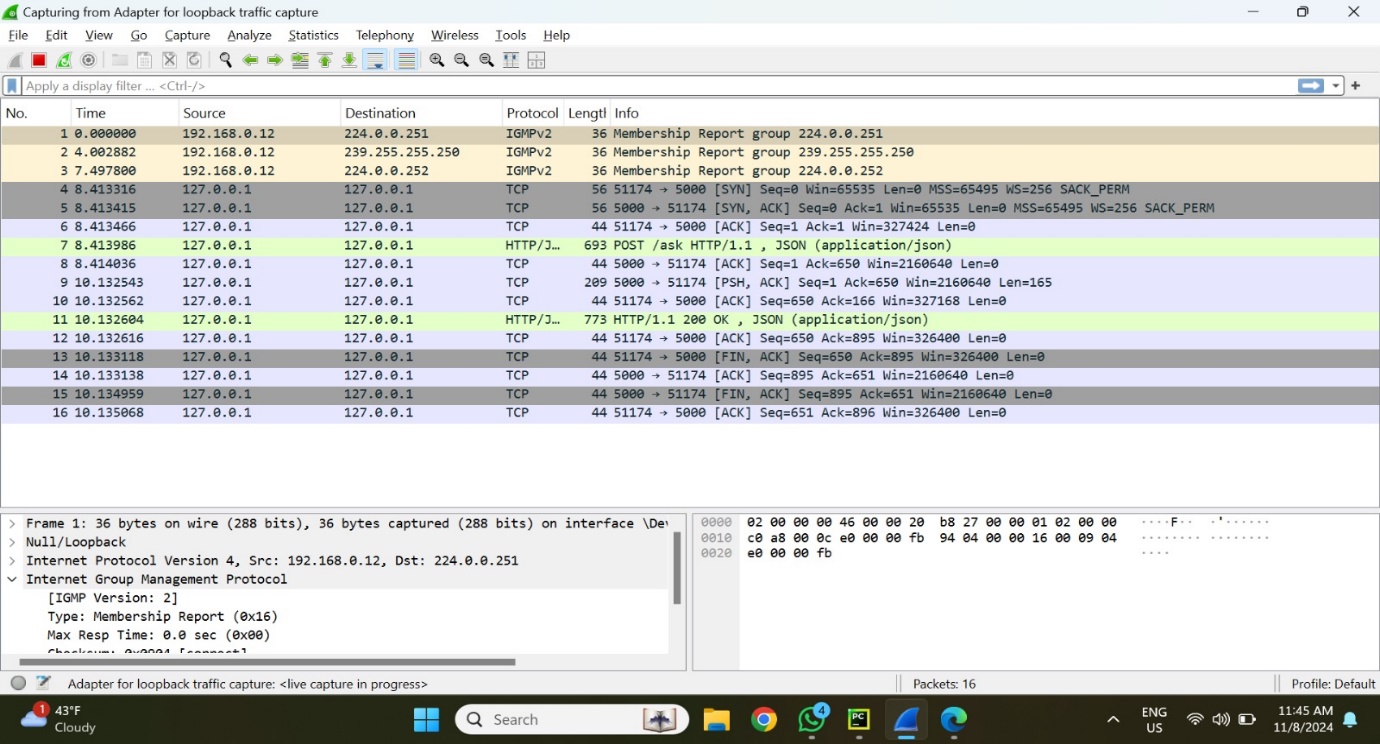
The Q&A Bot is designed to operate locally and responds to user inquiries on network security. To process these queries, it makes use of data from a local model and a Chroma database. HTTP over TCP (127.0.0.1) is used to transport data between the user interface and the locally installed backend server. This report examines the data flow between components, defines the packet characteristics, and analyzes the network packets captured in Wireshark during a typical question-response interaction.  
The Q&A Bot is designed to run locally and responds to user inquiries on network security. To process these queries, it makes use of data from a local model and a Chroma database. HTTP over TCP is used to transport data between the user interface and the locally installed backend server.

Prompt-1:

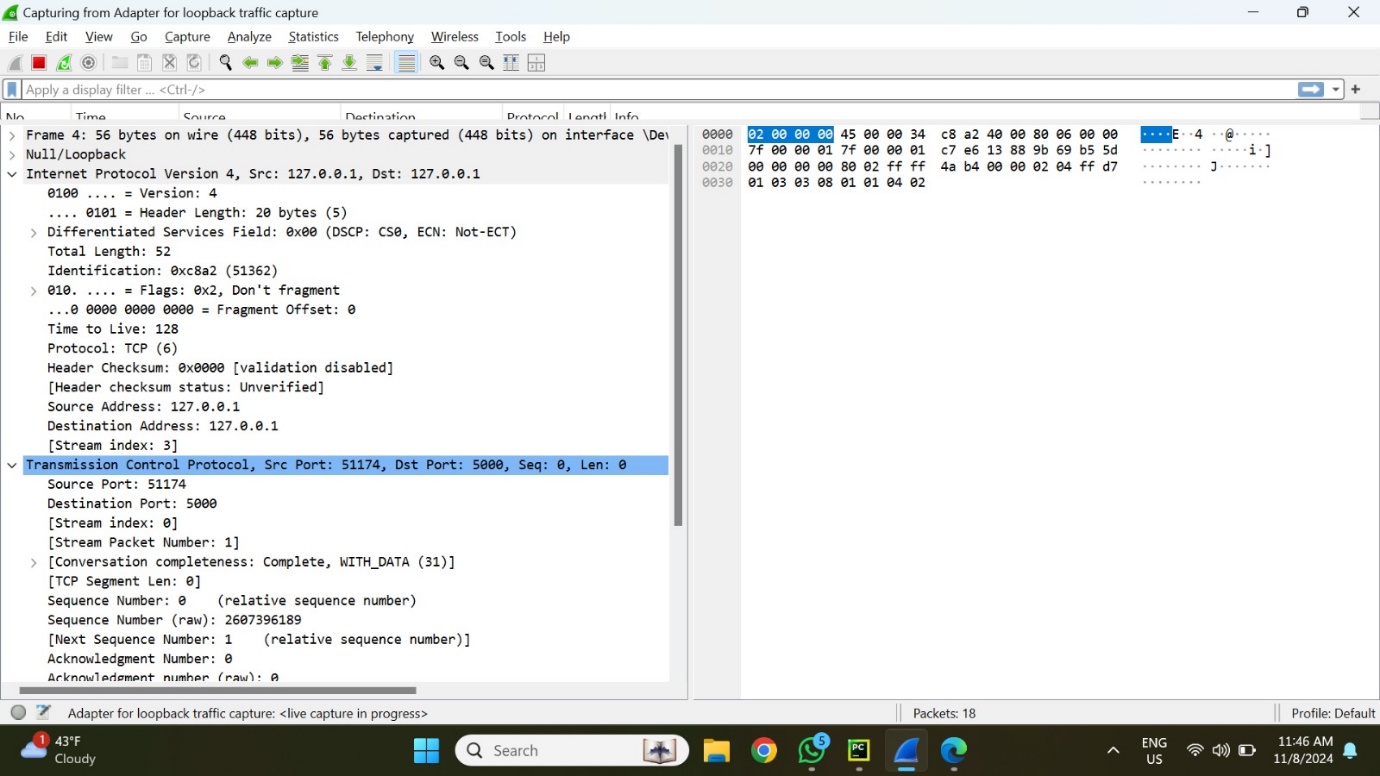
The chatbot is hosted locally and is accessible through 127.0.0.1:5000.

I have asked the chatbot, "What is Entropy," and the chatbot responded with a definition of entropy, explaining it as a measure of uncertainty and its relevance in cryptography.

This interaction shows that the chatbot is functioning as expected, responding to user queries with relevant information from provided sources. 

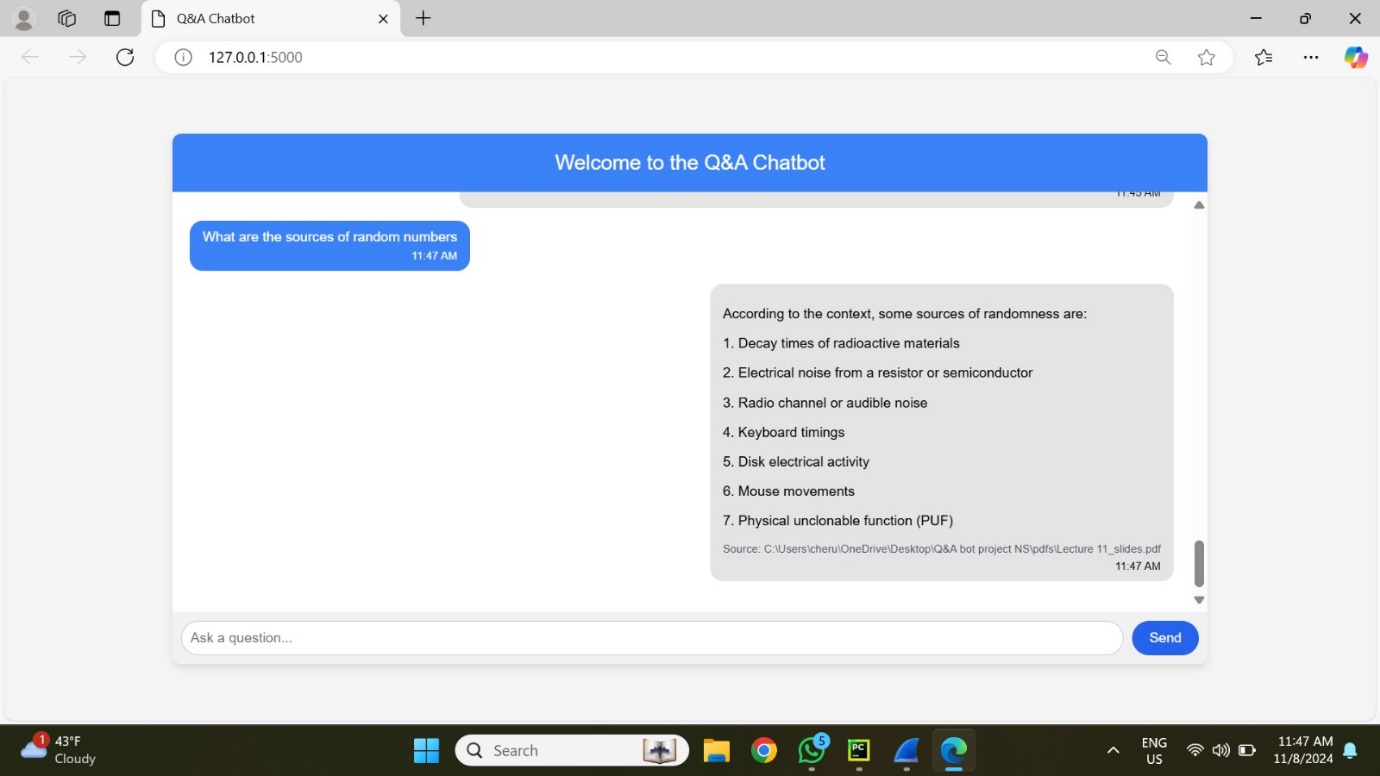


The first TCP handshake for a connection on port 5000 is displayed in packets 4-6. This creates a TCP connection between the client and server and consists of SYN, SYN-ACK, and ACK packets.   
Data transport and session termination are involved in packets 8–16:  
• HTTP/1.1 POST Request: Packet 7 displays an HTTP POST request that most likely contains JSON data and indicates that a Q&A chatbot is being asked a question.   
  
• HTTP/1.1 200 OK: The server's JSON-formatted response to the POST request, which may include the chatbot's response, is contained in packet 11.   
  
• TCP session termination is indicated by packets 12–16, where the connection is closed by FIN and ACK packets.



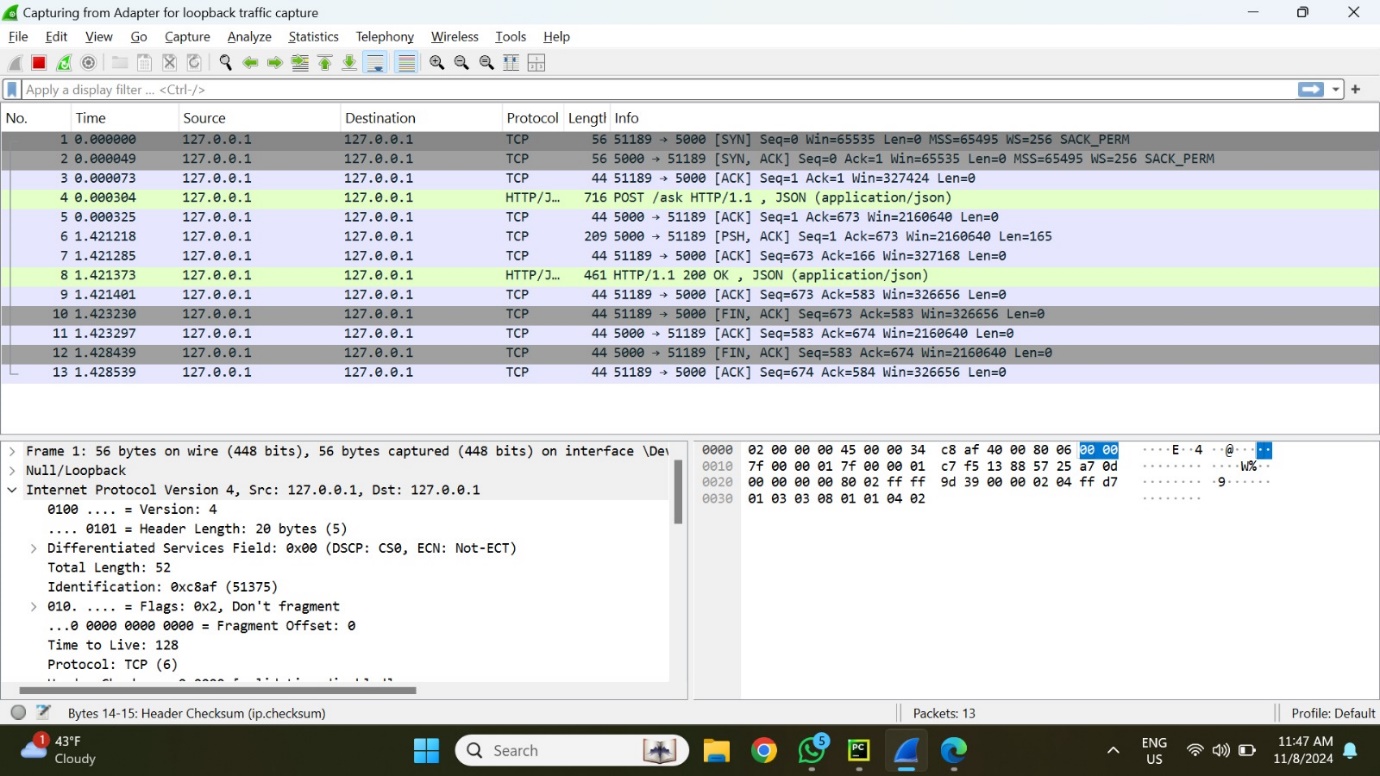
This picture displays a close-up of a Wireshark packet that was recorded on a loopback interface (127.0.0.1), which is commonly used for communication between local applications. Details of the packet include:   
IPv4, or Internet Protocol Version 4:  
The fact that the source and destination IP are both 127.0.0.1 suggests that the communication is local (loopback).   
This is a TCP packet since the Protocol field is set to TCP (6).   
The Time to Live (TTL), which is a standard default for local packets, is set to 128.   
TCP, or Transmission Control Protocol:   
Port of Source: 51174   
Port of destination: 5000   
Sequence Number: Sequence Number 0 is where the packet begins.   
Stream Index: Used to put talks back together, this indicates that it is a part of a stream.

Prompt-2:

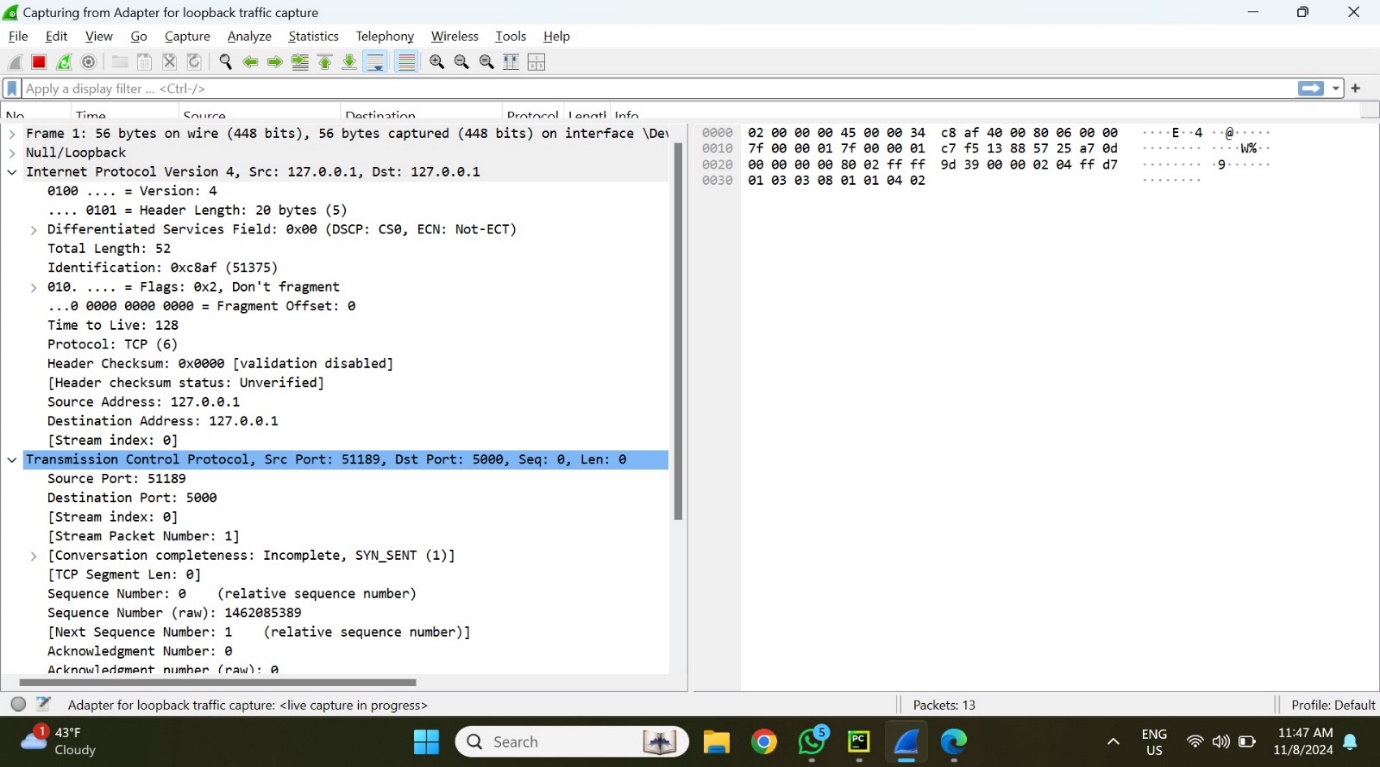


The prompt given by user and response given is shown in the above snapshot.

The source of the information tells about the resource document which actually contains the prompt information in this case is the Lecture 11\_sides.

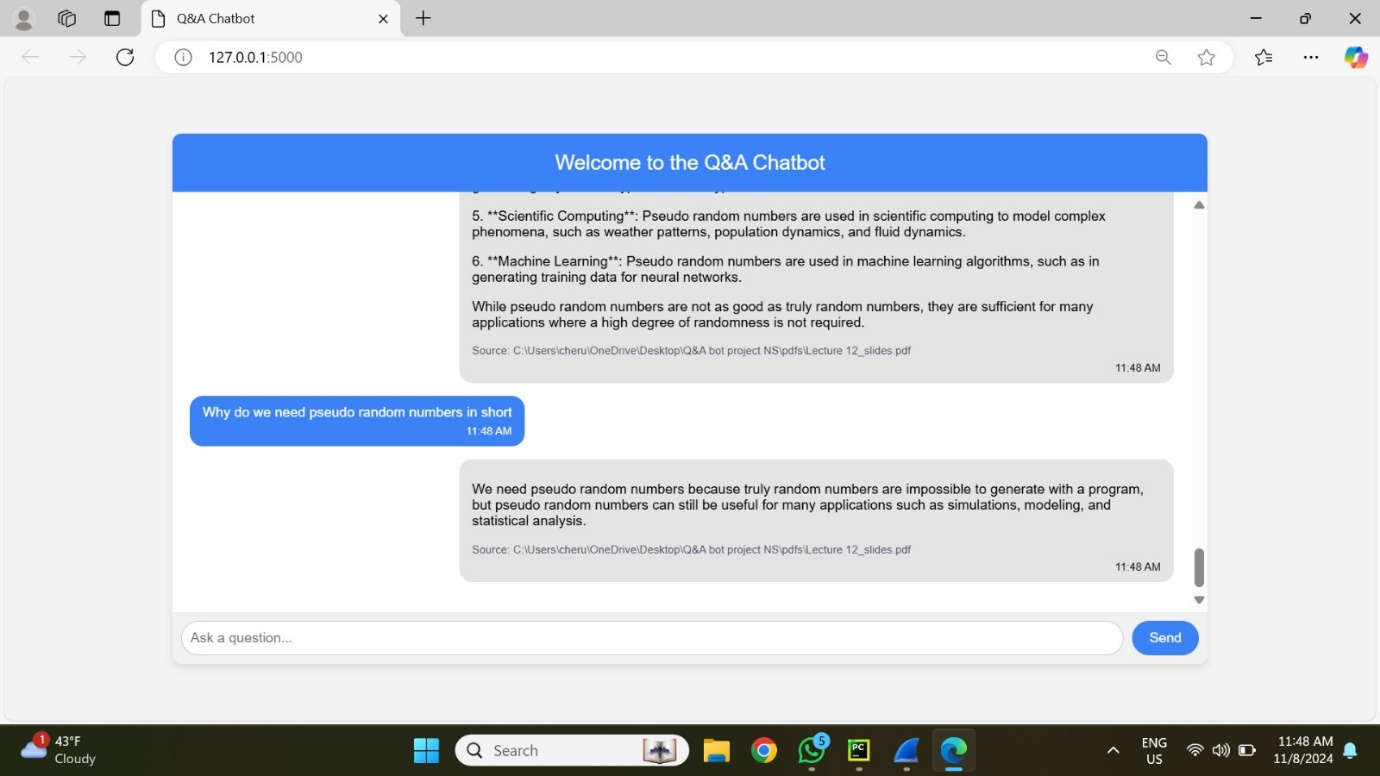


The packet transmitted during a TCP handshake and subsequent HTTP communication is shown in this image using Wireshark:   
• The typical three-way handshake to create a TCP connection is represented by the first SYN, SYN-ACK, and ACK packets.  
• Packet 6 indicates a client request to a chatbot or API by displaying an HTTP POST request (/ask HTTP/1.1) to the local server with JSON as the content type.   
• Packet 8 confirms that the request was successfully processed by displaying the server's HTTP response with status 200 OK.   
This demonstrates the complete request-response cycle lifetime in a localhost web application.

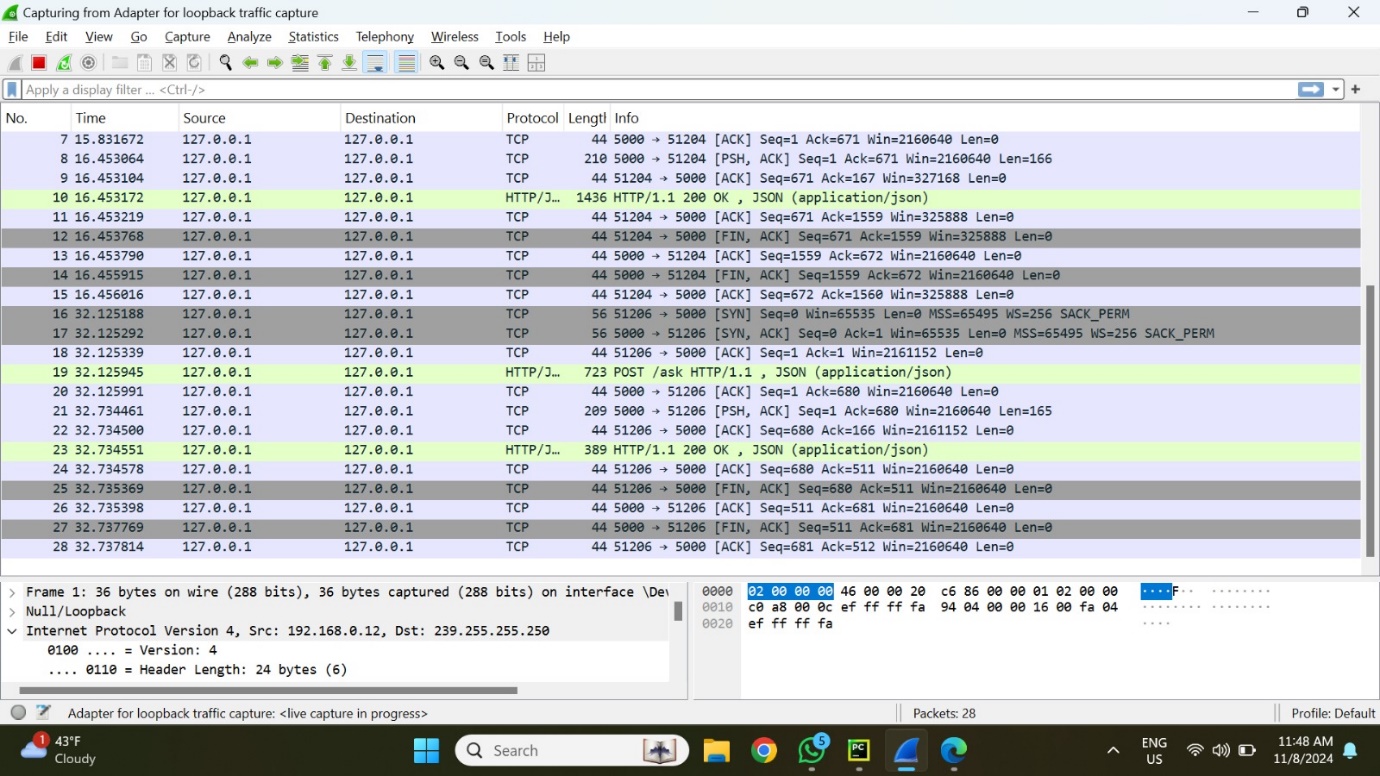


A TCP packet transmission on a loopback interface (127.0.0.1), signifying local communication, is captured by Wireshark in this picture. Important information includes:   
• IP Version: IPv4, signifying local connection with source and destination addresses of 127.0.0.1.  
• Protocol: TCP, with port 51189 as the source and port 5000 as the destination.   
• Flags: SYN, which signals the beginning of a TCP handshake, along with other information about the TCP segment, such as sequence numbers and checksum status.   
This packet most likely signifies the start of a TCP connection between a client and a local server.

Prompt-3:

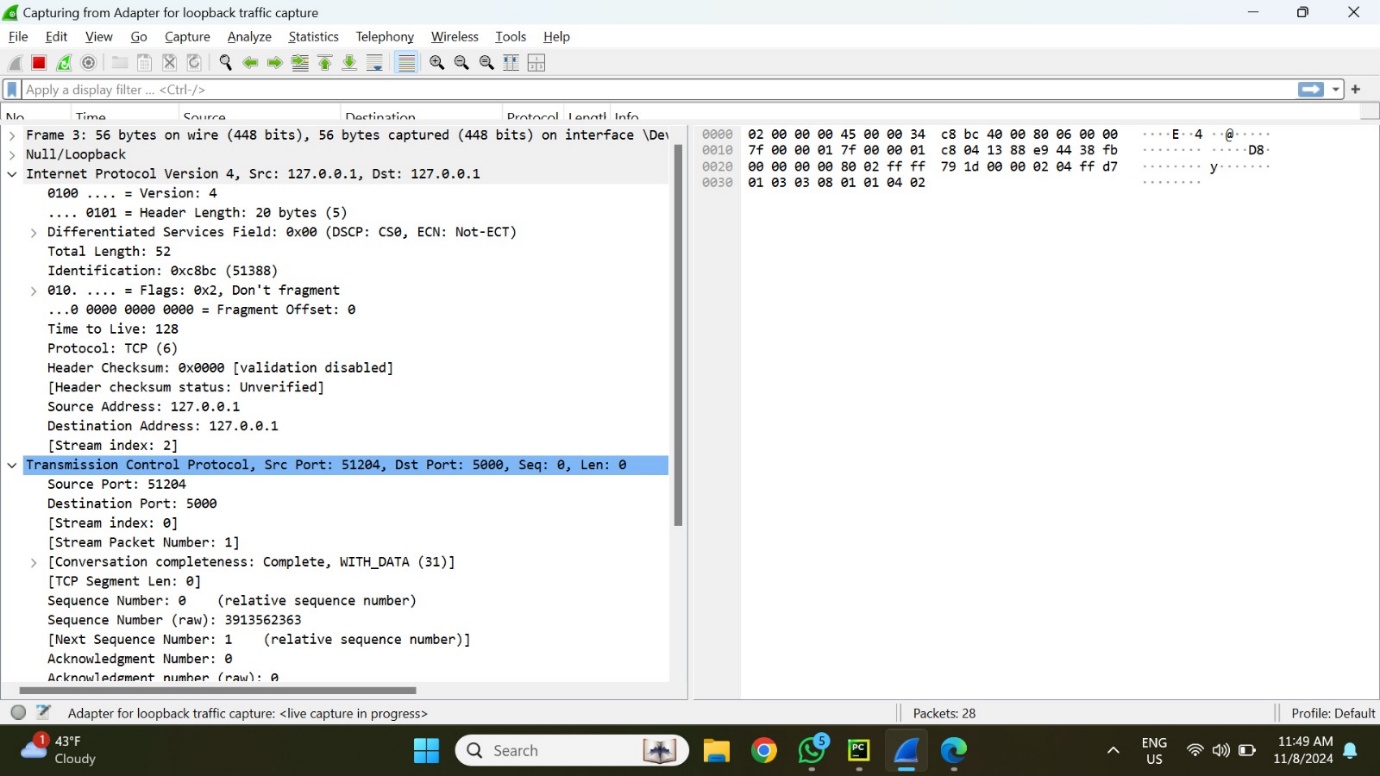
****

The prompt given by user and response given is shown in the above snapshot.The source of the information tells about the resource document which actually contains the prompt information in this case is the Lecture 12\_sides.

****

This image shows a broader list of packets in the capture, detailing various TCP and HTTP traffic between 127.0.0.1 (loopback address).

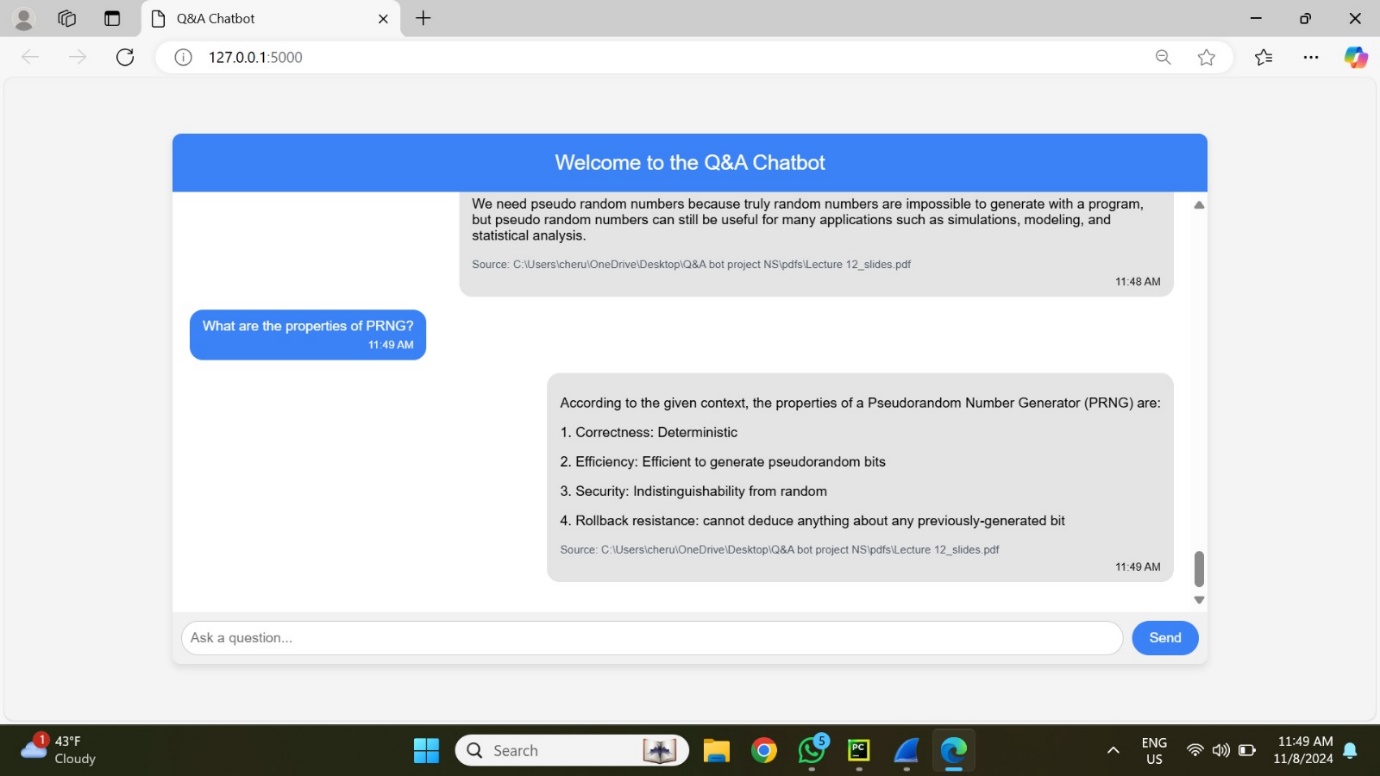
• Packet Information: Shows several TCP packets with a combination of data transfer, ACK, and SYN.   
• HTTP Traffic: POST requests containing JSON data and HTTP responses (200 OK) show that a local web server is being communicated with.  
• Source and Destination Ports: 51204 (source) and 5000 (destination) are noteworthy ports.   
• Frame Length: Frame 1 in this capture consists of 36 bytes collected and 36 bytes on the wire (288 bits).



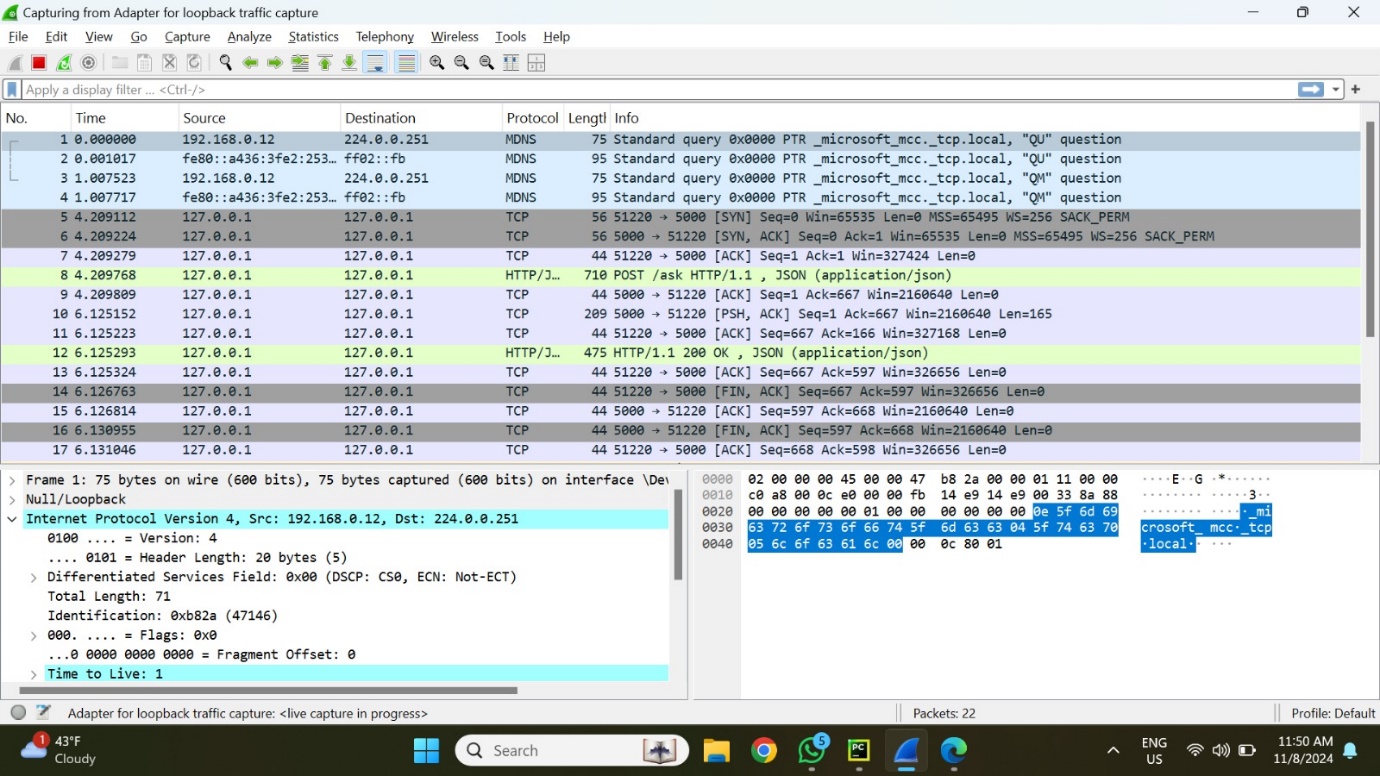
This picture displays a Wireshark packet capture. Important information includes:   
Frame Details: The frame consists of 56 bytes collected and 56 bytes on the wire (448 bits).  
Version 4 of the Internet Protocol (IP): Since both the source and destination IP addresses are 127.0.0.1, loopback traffic—data delivered to itself—is shown.   
TCP (Transmission Control Protocol) is the protocol. 52 bytes is the whole length.   
Time to Live: 128 (TTL).   
51204 is the source port and 5000 is the destination port for Transmission Control Protocol (TCP).   
 Sequence and Acknowledgement Numbers: Usually during the first connection handshake, the sequence number and acknowledgement number are both 0.

Data Length: 0 bytes, indicating that this is probably a payload-free acknowledgement or connection setup.

Prompt-4:

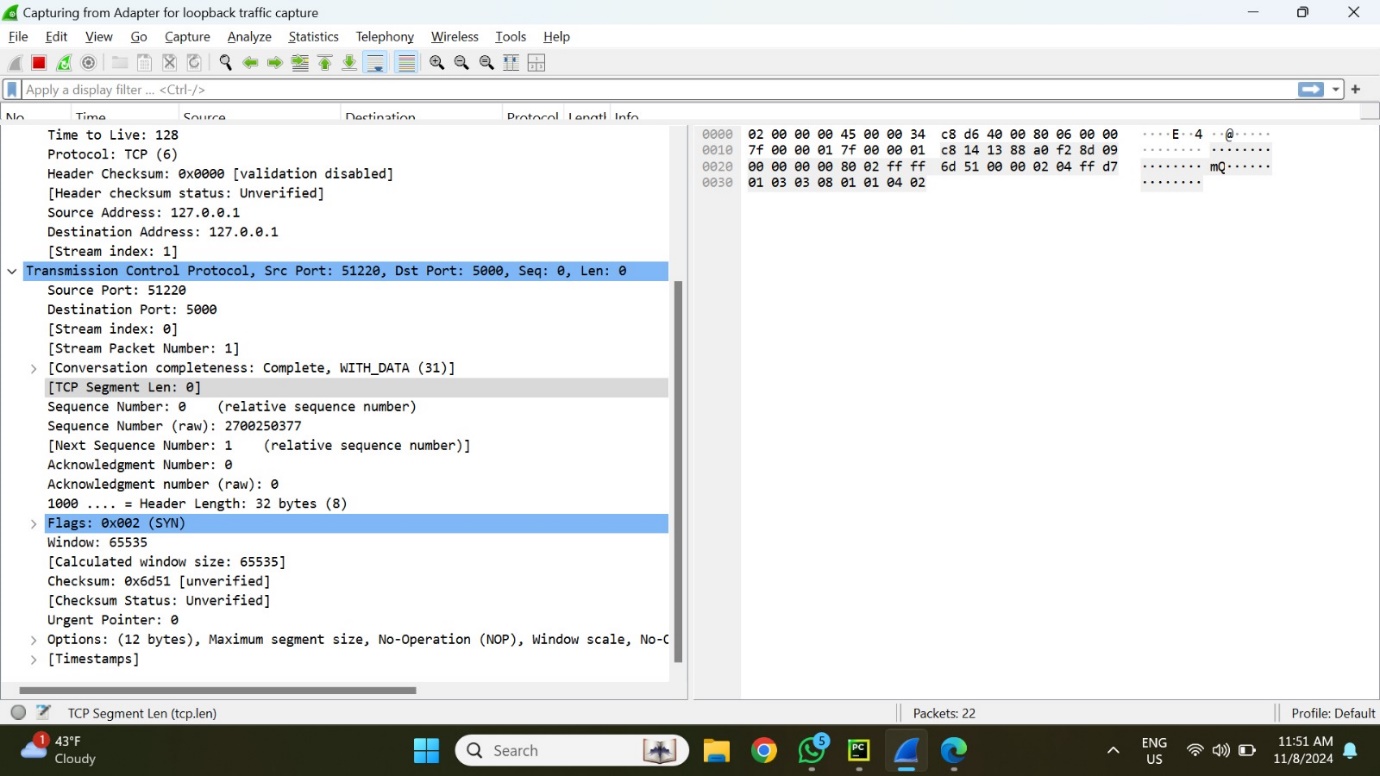


The prompt given by user and response given is shown in the above snapshot.The source of the information tells about the resource document which actually contains the prompt information in this case is the Lecture 12\_slides.pdf.



It provides an overview of packets captured on the localhost (127.0.0.1) interface, showing internal traffic between applications on the same computer. The selected packet includes details like **Source Port 51220** and **Destination Port 5000**, indicating communication with a local service, likely a web server or API.

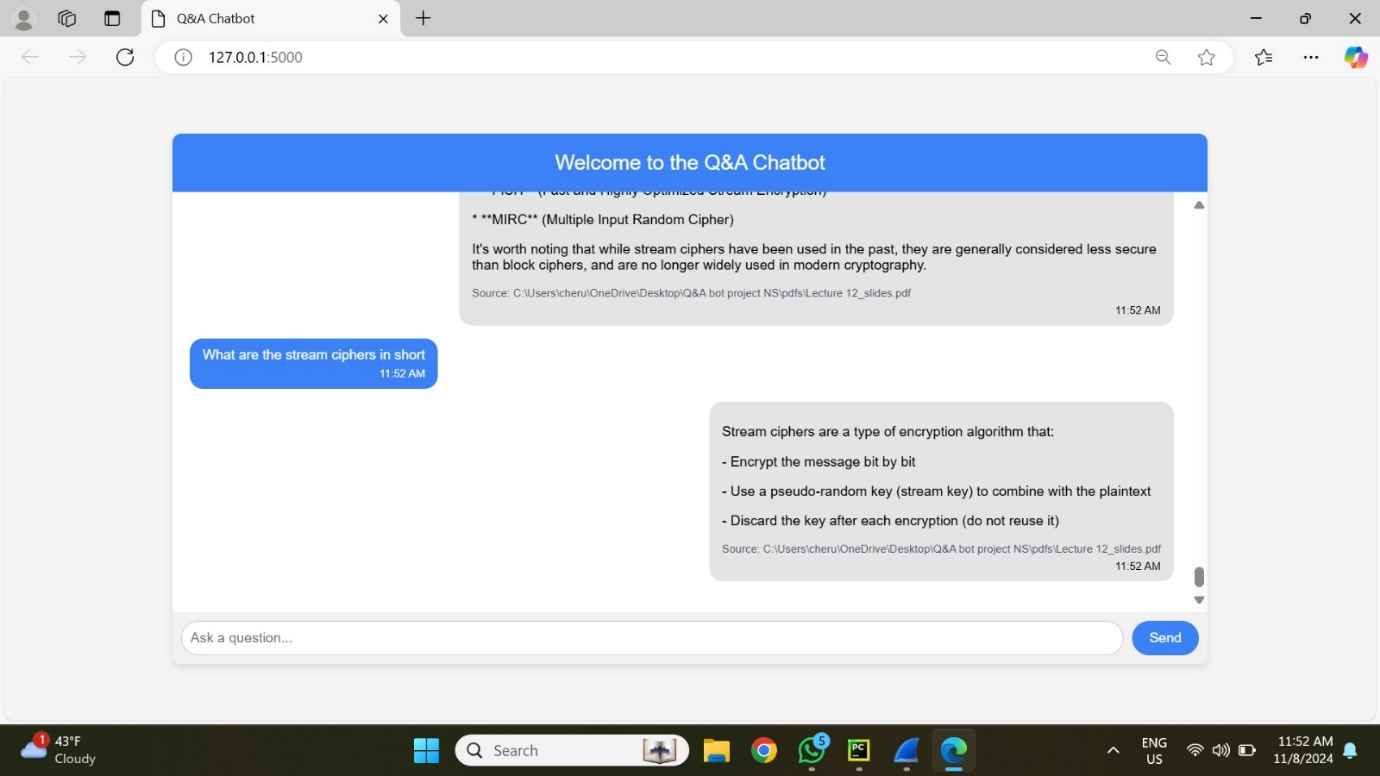
The **Frame Length** for this packet is **75 bytes**, representing the entire packet size on the network, including headers and payload. Various protocols are observed, such as TCP and HTTP, illustrating the typical setup of a local development environment.



This picture offers a close-up look at a particular TCP packet that was recorded using Wireshark. The packet is a part of a loopback traffic capture, which is from 127.0.0.1 to 127.0.0.1 and involves communication within the same device. Important information includes:   
• Ports of Origin and Destination: The packet starts at port 51220 and is routed to port 5000.  
• TCP Flags: A TCP handshake is started when the flag 0x002 (SYN) is present, indicating that this packet is SYN.   
• Sequence and Acknowledgement Numbers: Since this is the initial packet in the sequence, the sequence number is zero. Additionally, as is common for SYN packets, the acknowledgement number is zero as well.   
• Window Size: The 65535-byte window size indicates how much buffer space the receiver has for incoming data.   
• settings: There are TCP settings, like "maximum segment size,"

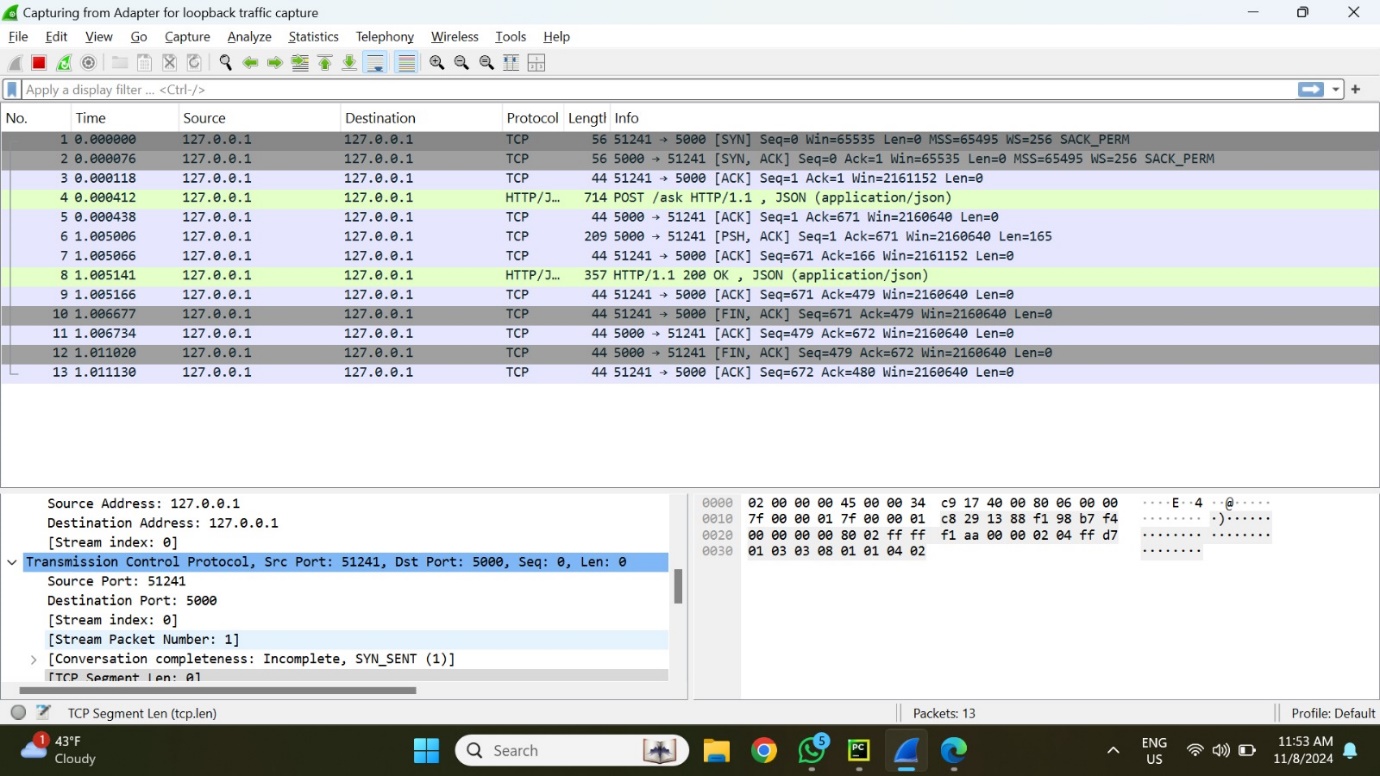
This SYN packet initiates the connection, marking the start of the TCP three-way handshake that will set up a reliable session between the client and server on the same machine.

Prompt-5:

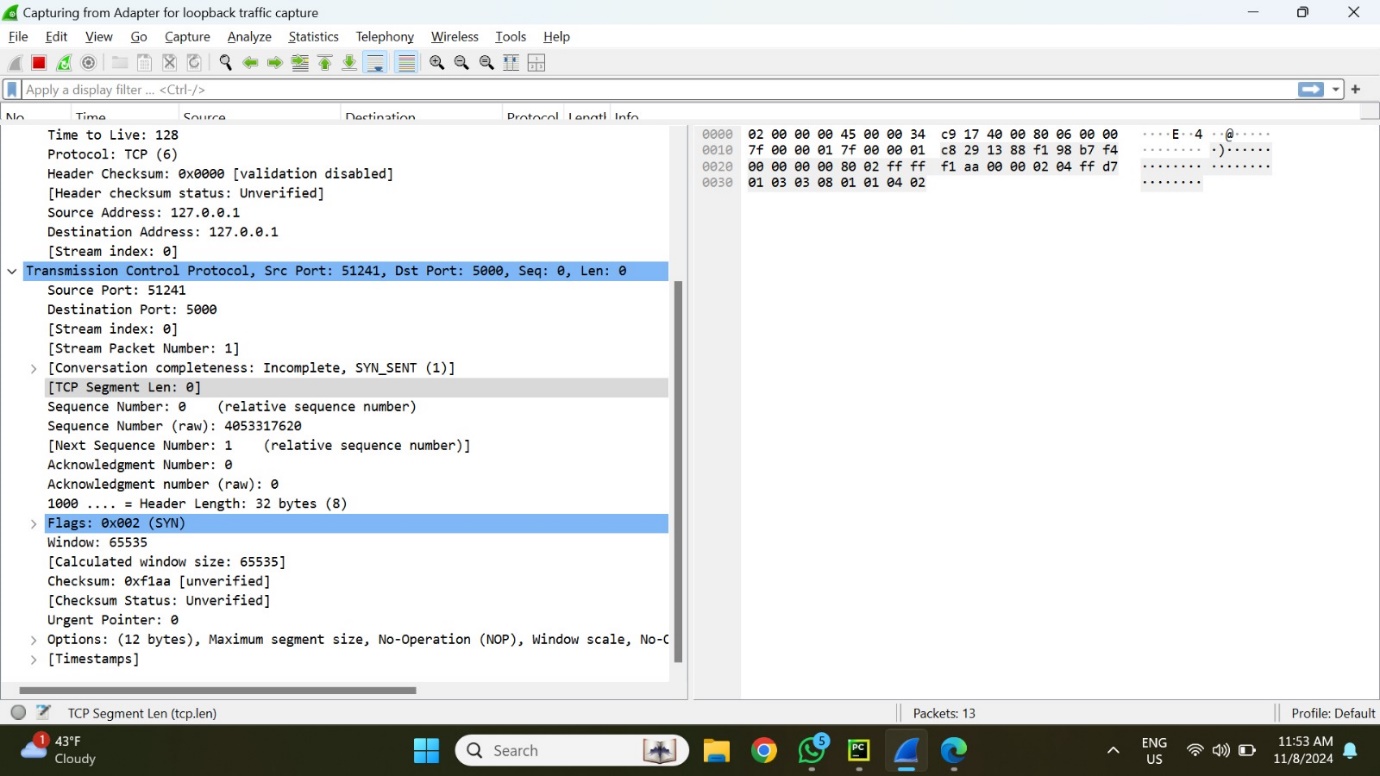


This shows a web-based Q&A chatbot running on 127.0.0.1:5000.

I have entered a question, "What are the stream ciphers in short," and the chatbot has responded with a brief explanation.The response also includes a source link pointing to a Lecture 12\_slides PDF document , suggesting the chatbot is referencing local files for answers.



This image shows a Wireshark capture session with loopback traffic, specifically on the localhost address 127.0.0.1.The packet list shows various TCP and HTTP transactions between source port 51241 and destination port 5000, indicating communication over TCP and HTTP protocols. Notably, there is a POST request with JSON content and a 200 OK response. This suggests that the chatbot's response may have been sent in response to a user request.The other TCP packets show handshakes and acknowledgments, with typical TCP flags like SYN, ACK, and FIN, indicating the establishment and closing of connections.



This image shows a close-up of a single packet captured in Wireshark, specifically the TCP SYN packet that initiates a connection between a client and server on the local loopback interface (127.0.0.1).

**Packet Overview**

* **Source Port**: 51241
* **Destination Port**: 5000
* **Protocol**: TCP (Transmission Control Protocol)
* **Time to Live (TTL)**: 128
* **Packet Length**: 56 bytes

The packet contains essential details for initiating a TCP connection, marking the first step in the **three-way handshake** process.

**Key Fields and Details**

1. Source and Destination Information

Source Port (51241): A dynamically assigned port on the client side initiating the connection.

Destination Port (5000): The server port being accessed, likely running a service (such as the Q&A chatbot) listening for requests on this port.

1. Sequence Number

Sequence Number: 0 (relative sequence number)

Raw Sequence Number: 4053317620

The order of bytes sent in the TCP connection is tracked using this first sequence number. Throughout the session, it enables both the sender and the recipient to verify packet sequence and receipt.

1. Flags

Flags Field (0x002): Contains the SYN (Synchronize) flag.

Window Size: 65535

This field specifies the amount of data (in bytes) that the sender is willing to receive in one TCP segment, indicating the client’s buffer size.

1. Options Field

Maximum Segment Size (MSS): 65495 bytes, which is the largest amount of data the sender can handle in a single segment.

Window Scale (WS): 256, allowing for a larger TCP window size to optimize data flow.

SACK\_PERM (Selective Acknowledgment Permitted): Indicates that selective acknowledgments are supported, allowing the receiver to inform the sender about non-contiguous data received, enhancing performance.

No-Operation (NOP): Used as padding to align fields.

1. Checksum

Checksum (0xf1aa): A value calculated by the sender to verify the integrity of the header. Here, the status is marked as "unverified," as Wireshark has not confirmed it.

1. TCP Segment Length

The segment length is 0 bytes in this SYN packet, as it does not carry any data. The SYN packet only establishes a connection, with actual data transferred in subsequent packets.