**Q & A CHAT BOT ROUND 2**

**Nithish Reddy Konakati**

**R11904503**

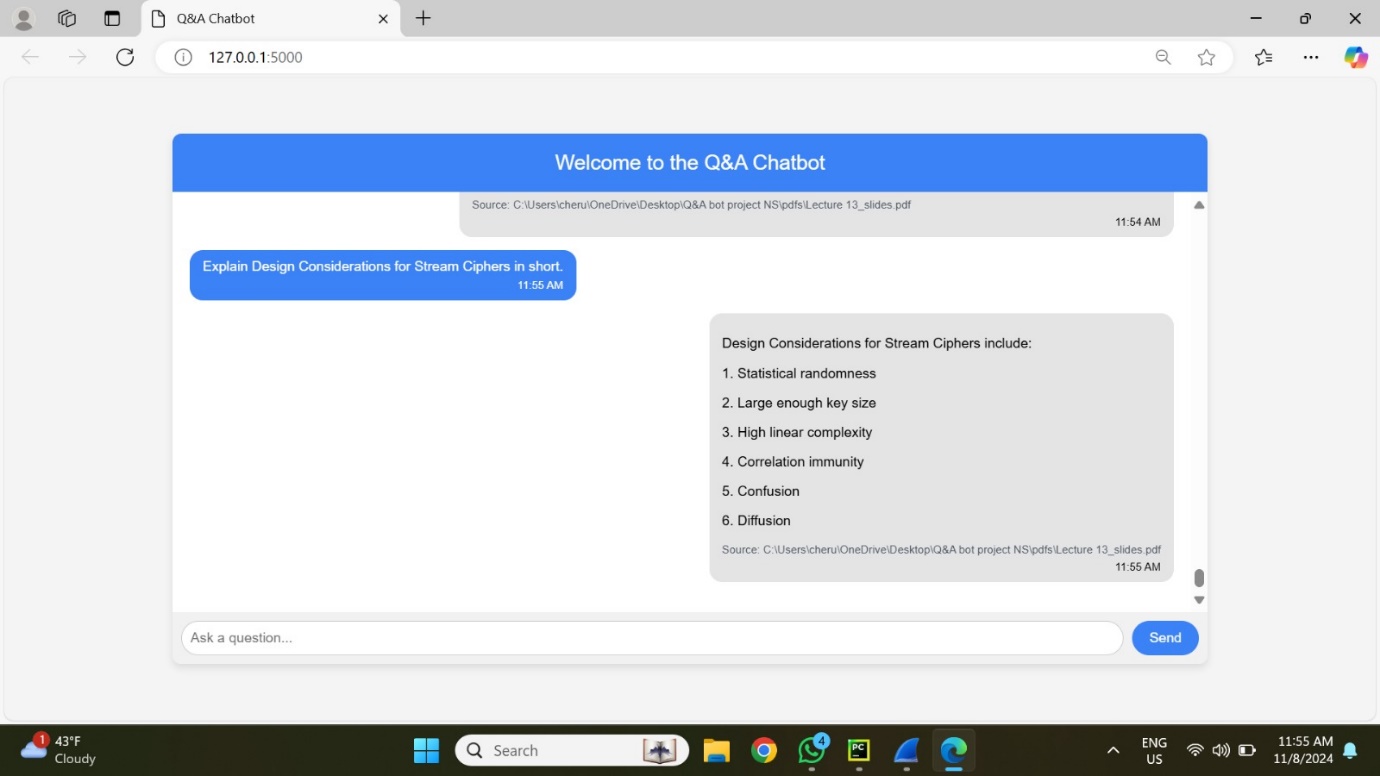
The Q&A Bot answers user questions about network security and is made to run locally. It uses information from a Chroma database and a local model to process these queries. Data is transferred between the user interface and the locally installed backend server using HTTP over TCP (127.0.0.1). The network packets recorded in Wireshark during a standard question-response exchange are analysed in this report, which also describes the packet characteristics and looks at the data flow between components.

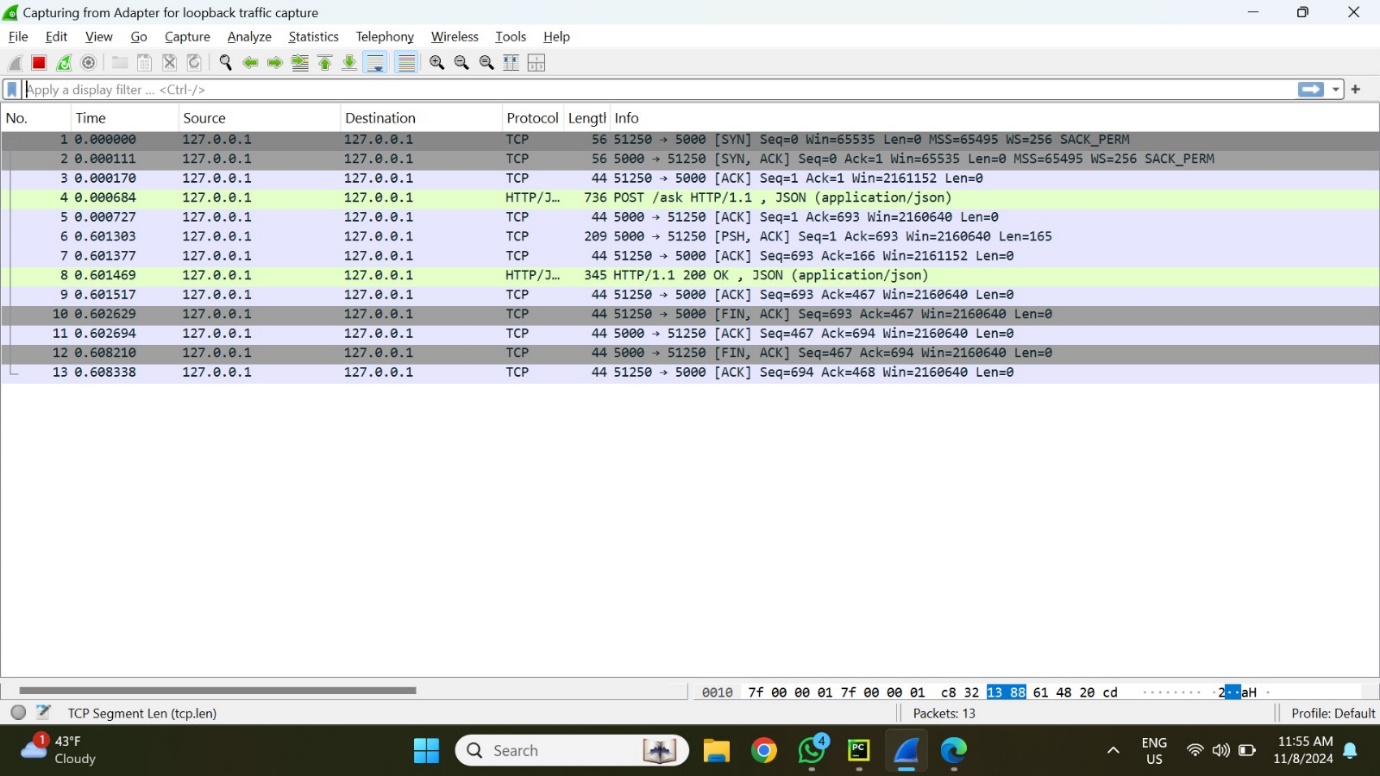
**Prompt-1:**

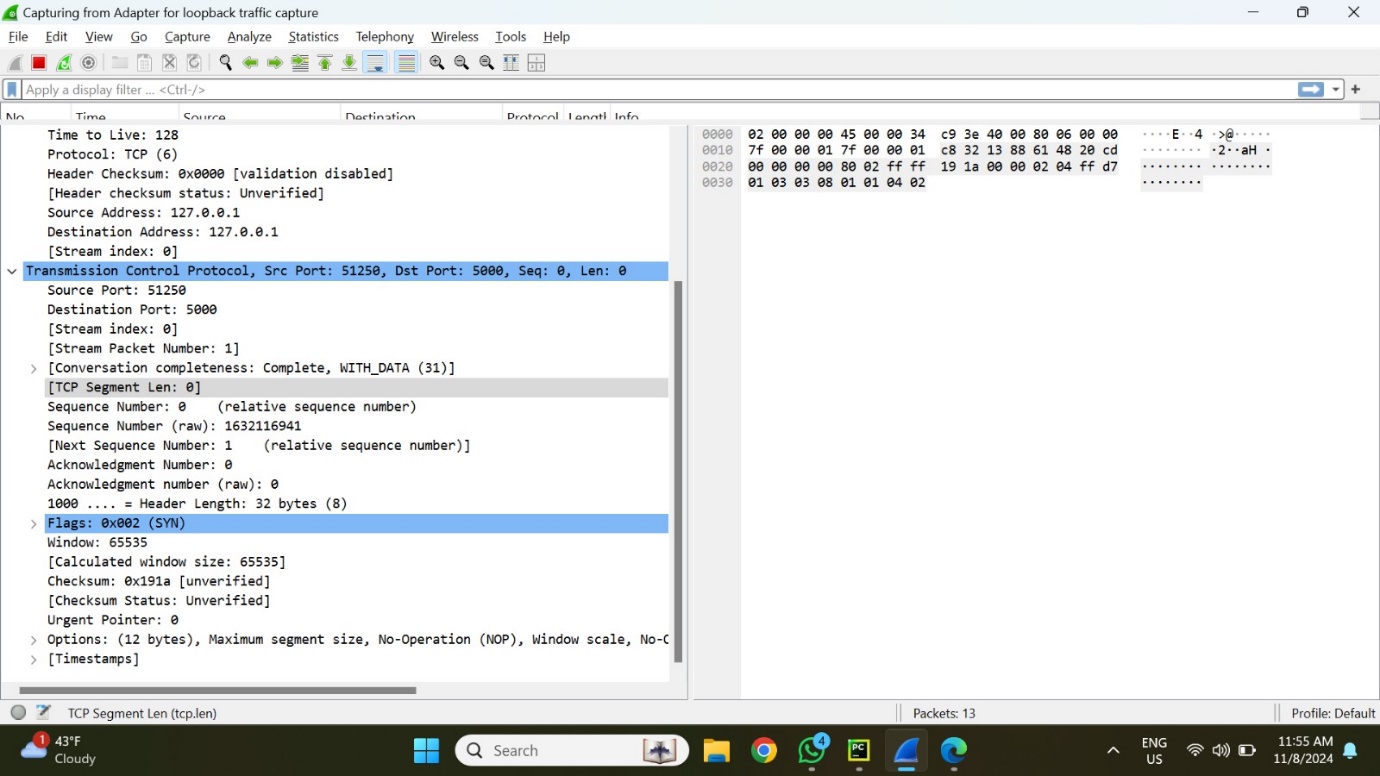
**Packet Analysis for User Query and Bot Response**

**1. User Query Transmission (Packets 1-8)**

Procedure: The data packet is transmitted from the client (user interface) to the server on the local computer when the user inputs a query. Below is the snap of the query and response.







**Source and Destination**:

* Source IP: 127.0.0.1
* Destination IP: 127.0.0.1 (local communication)

**Port Information**:

* Source Port: 51250
* Destination Port: 5000

**Protocol**: TCP (Transmission Control Protocol)

SYN Packet (Initial connection): A TCP SYN packet sent from the client to the server initiates the connection and creates the session.

ACK packets: To verify packet receipt, acknowledgments are sent back and forth.

HTTP Request: The query is submitted in JSON format as a POST request (seen in green in the Wireshark trace). The user's query data is contained in this HTTP packet.

**Key Parameters:**

**Sequence Number:** Begins with Seq=0 and increases with each packet.

**Acknowledgment:** The receiving side acknowledges each packet.

**Data Length:** The HTTP POST request packet (Packet 4) is longer due to the JSON payload (user query).

2. **Server Processing and Response (Packets 8-13)**

**Process:** The backend server executes the query by getting appropriate embeddings from Chroma and producing a response using the local language model. The response, including the source reference, is then returned to the user interface.

**HTTP Response:** The client receives a 200 OK status code, signifying successful processing.  
The response provides the query's answer as well as a reference that specifies the particular source (for example, "Lecture 13 slides").

**Packet Structure:** HTTP Response Packet (Packet 8) This packet comprises the server's JSON response, which includes both the generated answer and the citation.

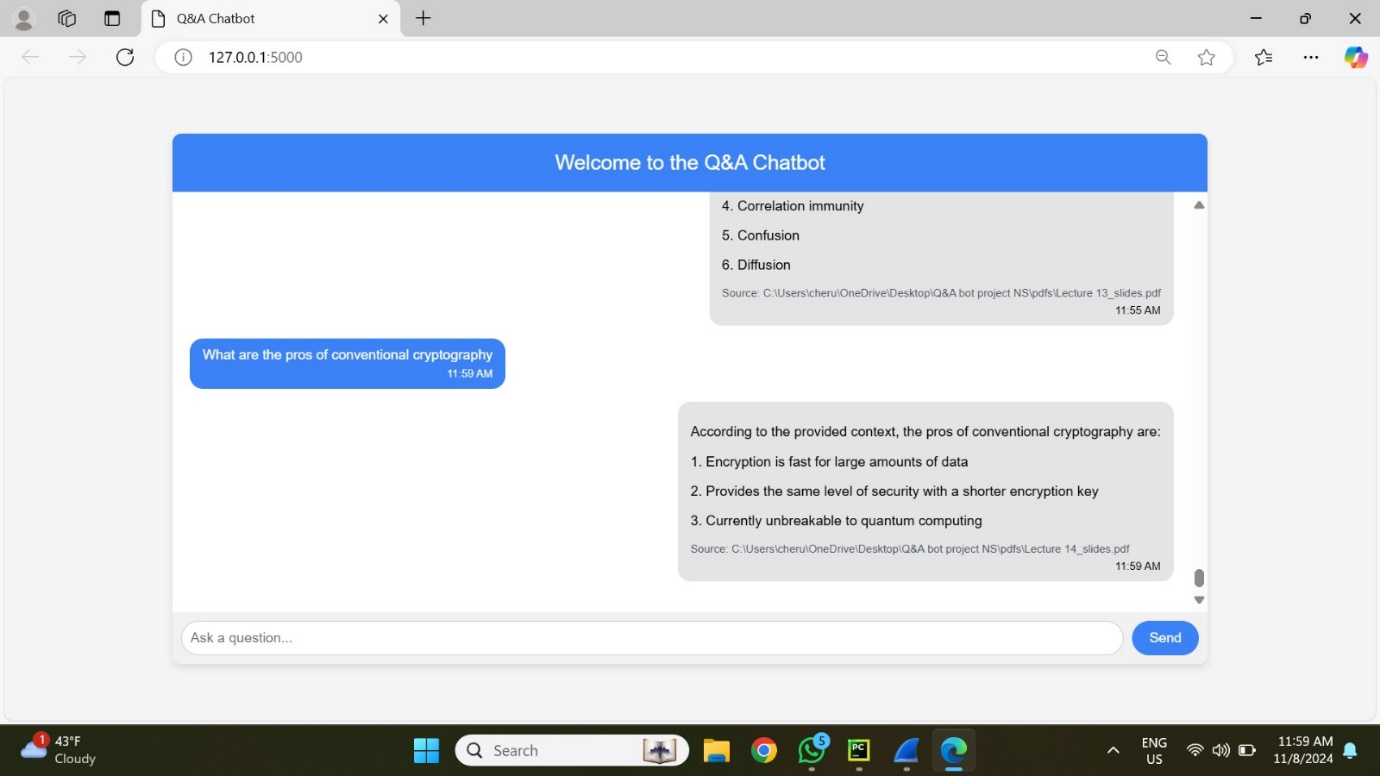
**Key Parameters:**

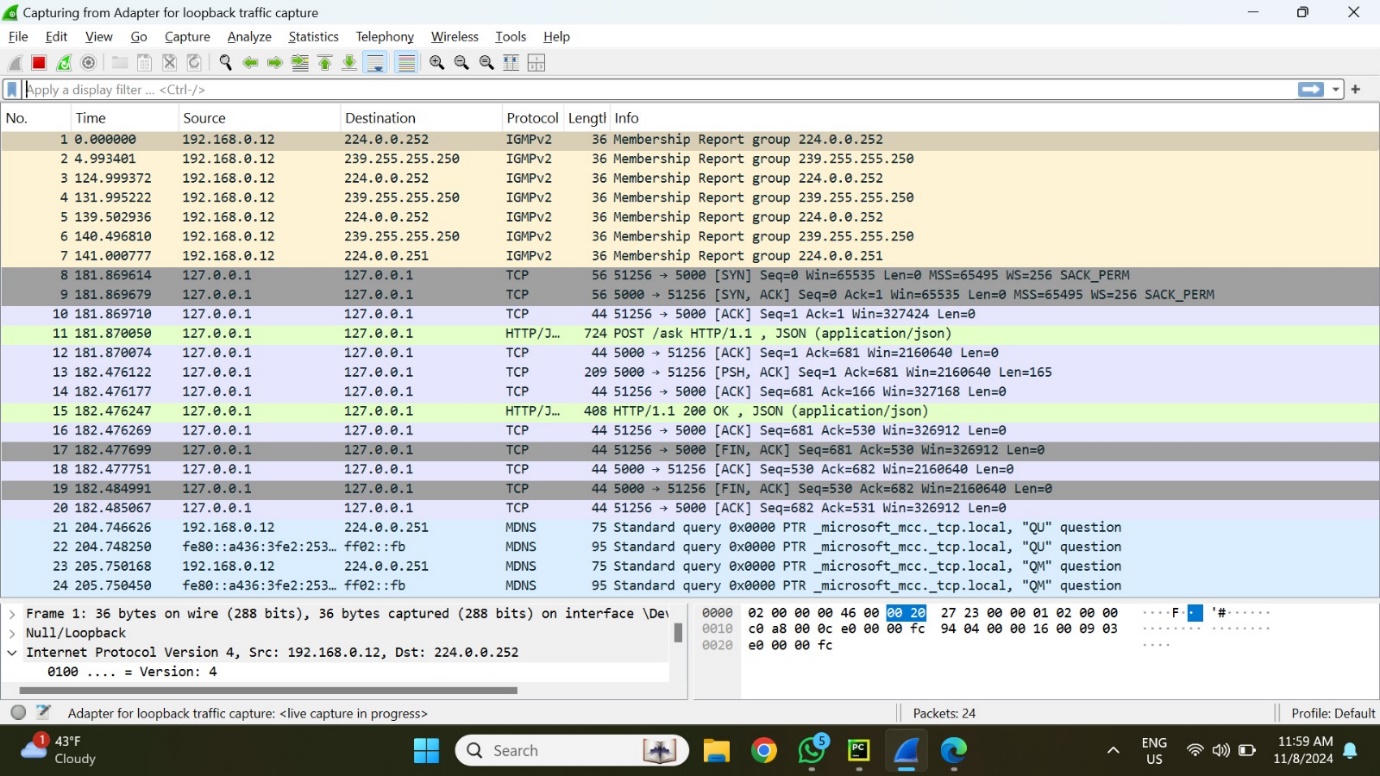
**Frame Length:** Response packets are often shorter than request packets because they simply include the text response.

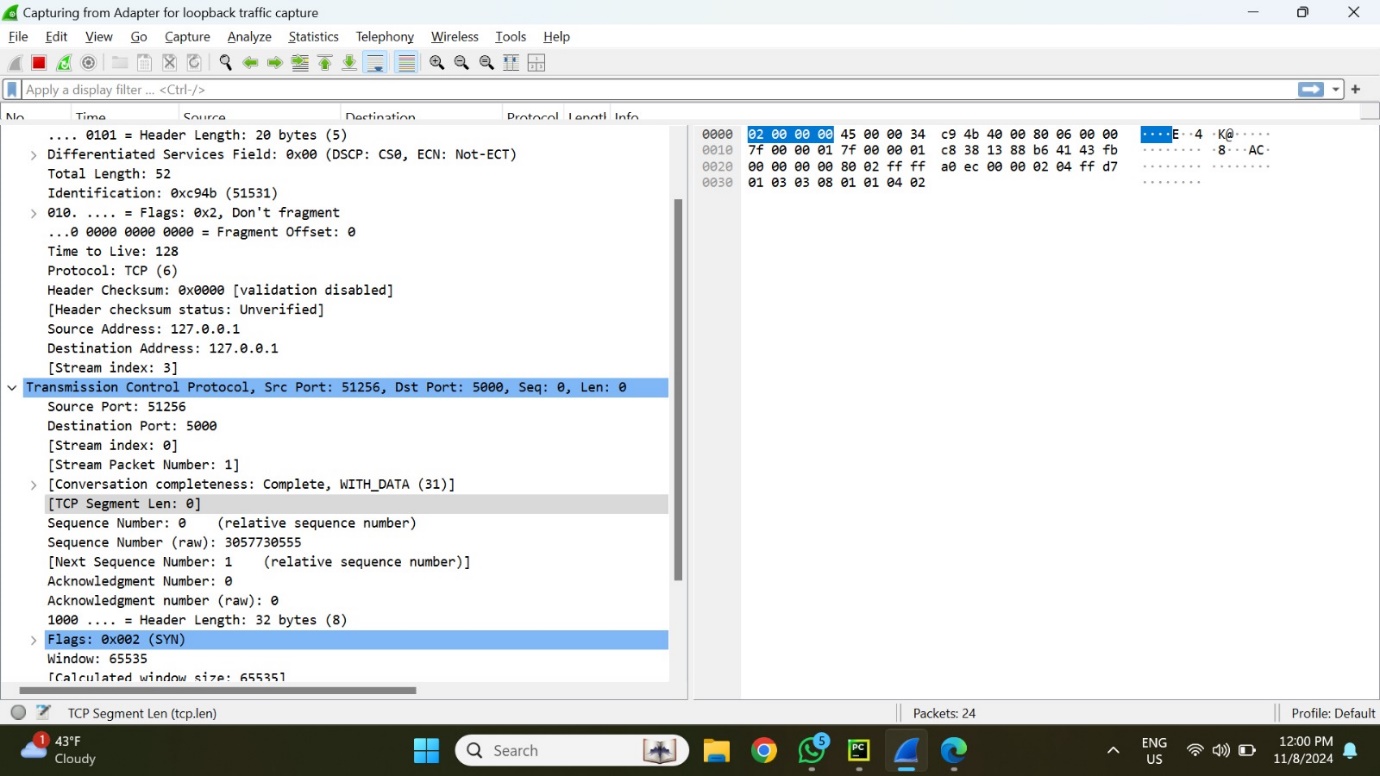
**Data Representation:** Wireshark displays the hexadecimal representation of the response data, including the exact bytes transferred.

**Prompt-2:  
1. User Query Transmission (Packets 1-8)**

Procedure: The data packet is transmitted from the client (user interface) to the server on the local computer when the user inputs a query. Below is the snap of the query and response.







**Source and Destination**:

* Source IP: 127.0.0.1
* Destination IP: 127.0.0.1 (local communication)

**Port Information**:

* Source Port: 51256
* Destination Port: 5000

**Protocol**: TCP (Transmission Control Protocol)

SYN Packet (Initial connection): A TCP SYN packet sent from the client to the server initiates the connection and creates the session.

ACK packets: To verify packet receipt, acknowledgments are sent back and forth.

HTTP Request: The query is submitted in JSON format as a POST request (seen in green in the Wireshark trace). The user's query data is contained in this HTTP packet.

**Key Parameters:**

**Sequence Number:** Begins with Seq=0 and increases with each packet.

**Acknowledgment:** The receiving side acknowledges each packet.

**Data Length:** The HTTP POST request packet (Packet 4) is longer due to the JSON payload (user query).

2. **Server Processing and Response (Packets 8-13)**

**Process:** The backend server executes the query by getting appropriate embeddings from Chroma and producing a response using the local language model. The response, including the source reference, is then returned to the user interface.

**HTTP Response:** The client receives a 200 OK status code, signifying successful processing.  
The response provides the query's answer as well as a reference that specifies the particular source (for example, "Lecture 14 slides").

**Packet Structure:** HTTP Response Packet (Packet 8) This packet comprises the server's JSON response, which includes both the generated answer and the citation.

**Key Parameters:**

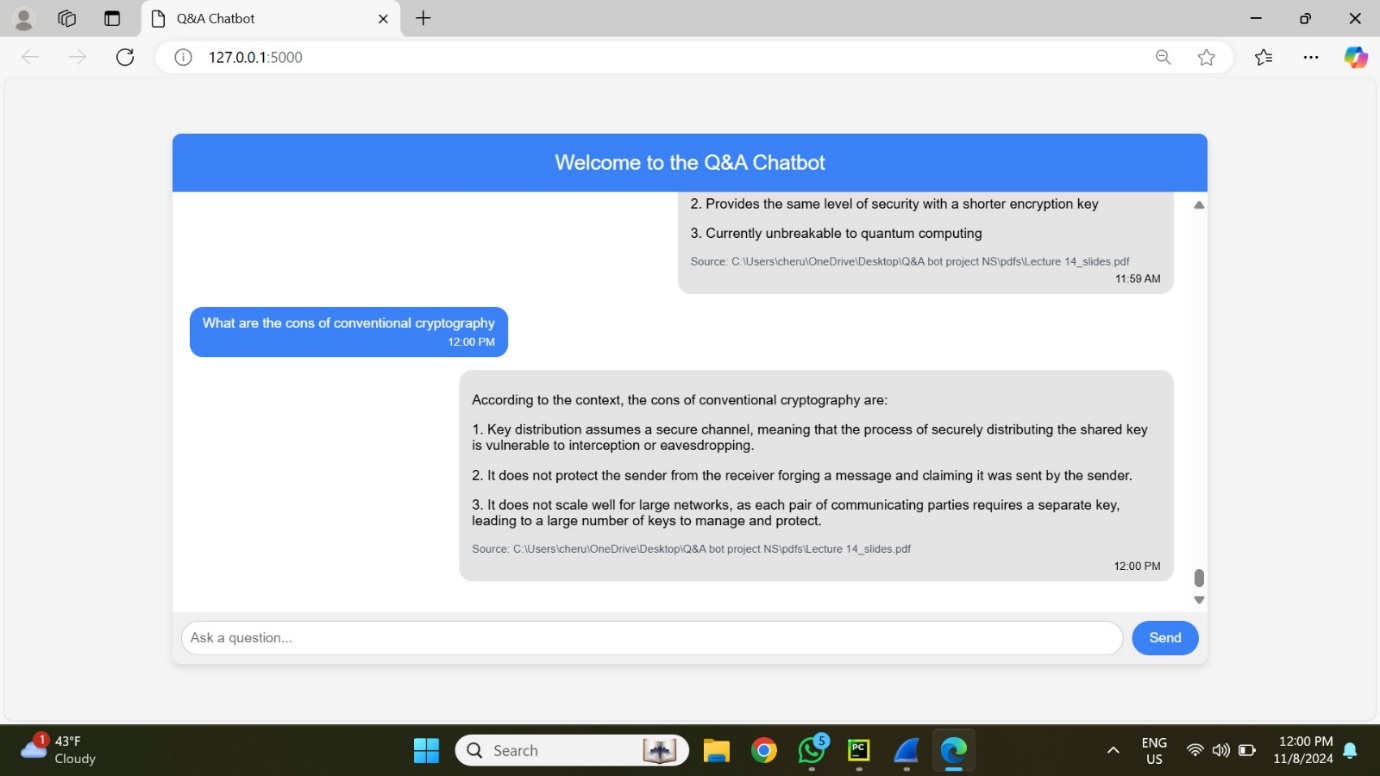
**Frame Length:** Response packets are often shorter than request packets because they simply include the text response.

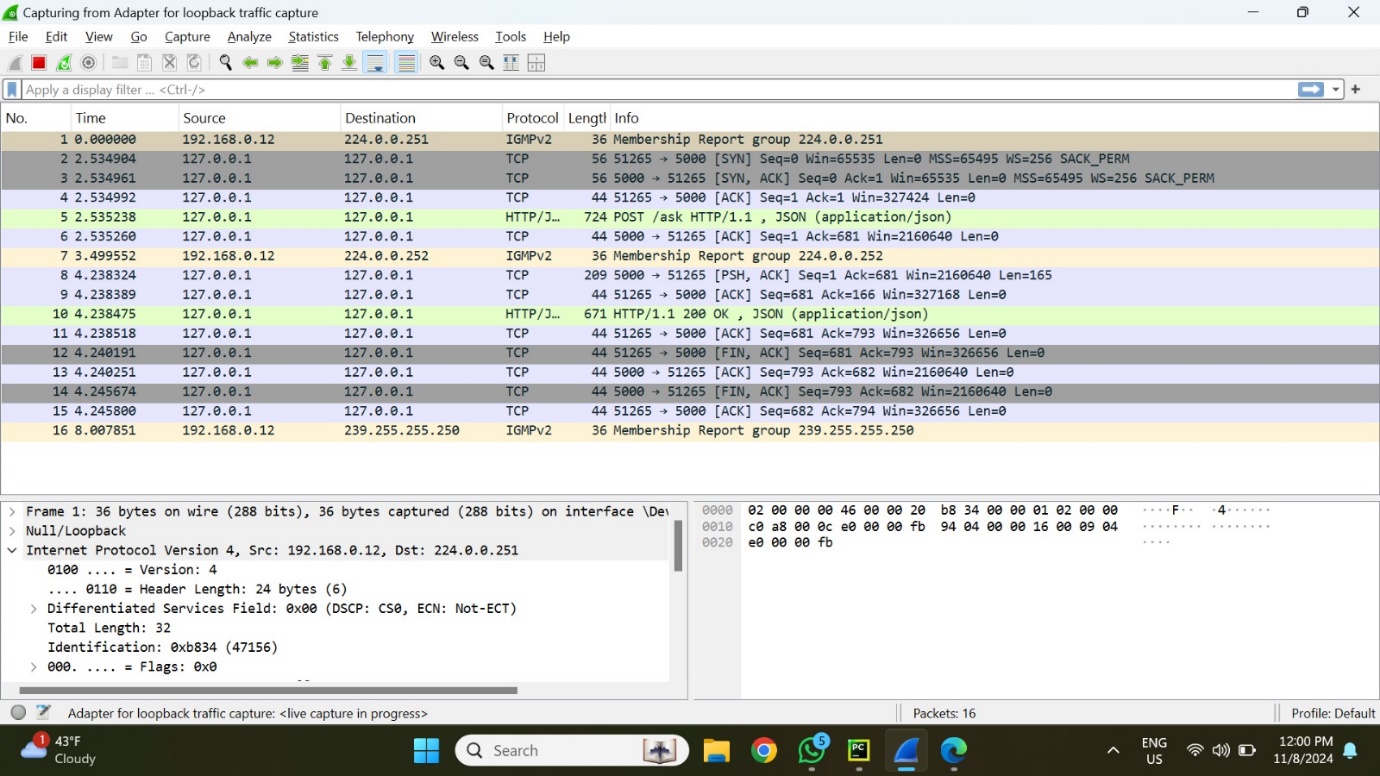
**Data Representation:** Wireshark displays the hexadecimal representation of the response data, including the exact bytes transferred.

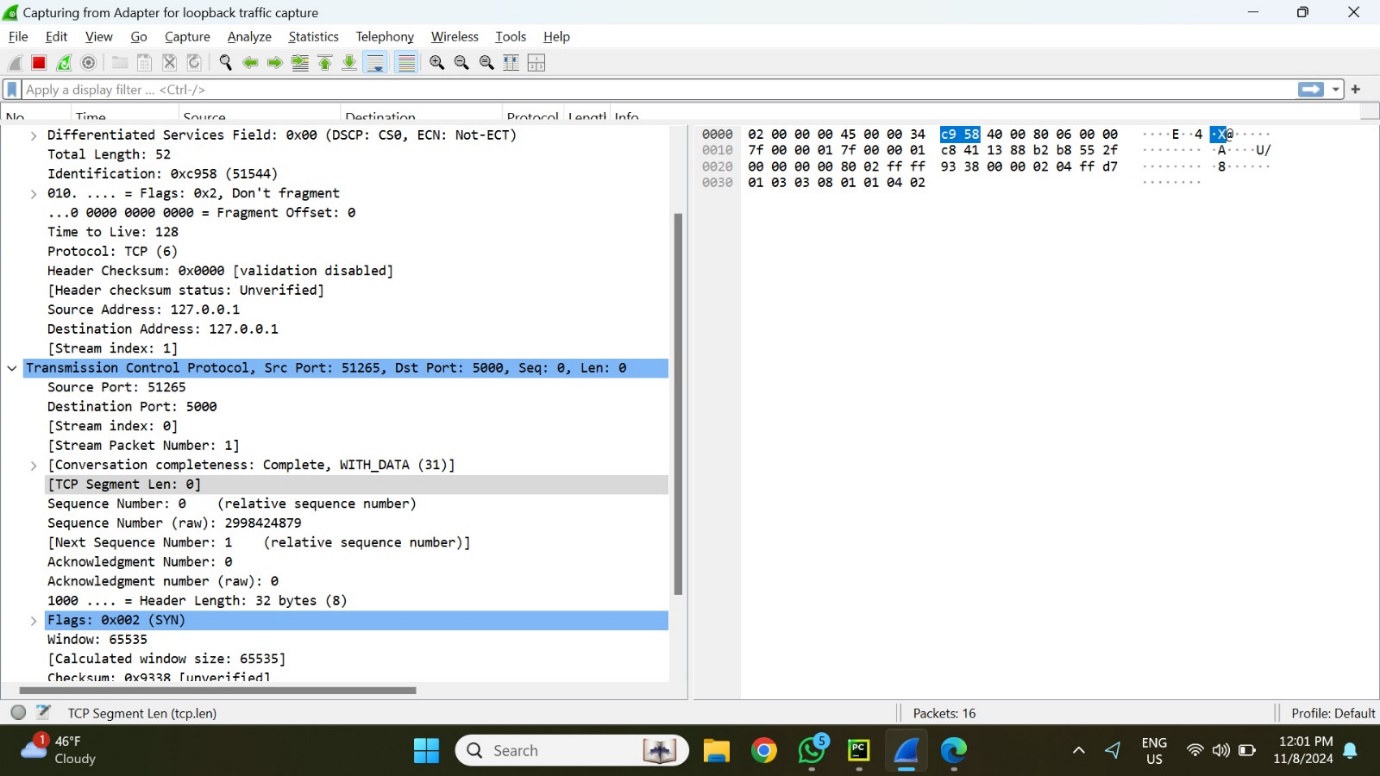
**Prompt-3:**

**1. User Query Transmission (Packets 1-8)**

Procedure: The data packet is transmitted from the client (user interface) to the server on the local computer when the user inputs a query. Below is the snap of the query and response.







**Source and Destination**:

* Source IP: 127.0.0.1
* Destination IP: 127.0.0.1 (local communication)

**Port Information**:

* Source Port: 51265
* Destination Port: 5000

**Protocol**: TCP (Transmission Control Protocol)

SYN Packet (Initial connection): A TCP SYN packet sent from the client to the server initiates the connection and creates the session.

ACK packets: To verify packet receipt, acknowledgments are sent back and forth.

HTTP Request: The query is submitted in JSON format as a POST request (seen in green in the Wireshark trace). The user's query data is contained in this HTTP packet.

**Key Parameters:**

**Sequence Number:** Begins with Seq=0 and increases with each packet.

**Acknowledgment:** The receiving side acknowledges each packet.

**Data Length:** The HTTP POST request packet (Packet 4) is longer due to the JSON payload (user query).

2. **Server Processing and Response (Packets 8-13)**

**Process:** The backend server executes the query by getting appropriate embeddings from Chroma and producing a response using the local language model. The response, including the source reference, is then returned to the user interface.

**HTTP Response:** The client receives a 200 OK status code, signifying successful processing.  
The response provides the query's answer as well as a reference that specifies the particular source (for example, "Lecture 14 slides").

**Packet Structure:** HTTP Response Packet (Packet 8) This packet comprises the server's JSON response, which includes both the generated answer and the citation.

**Key Parameters:**

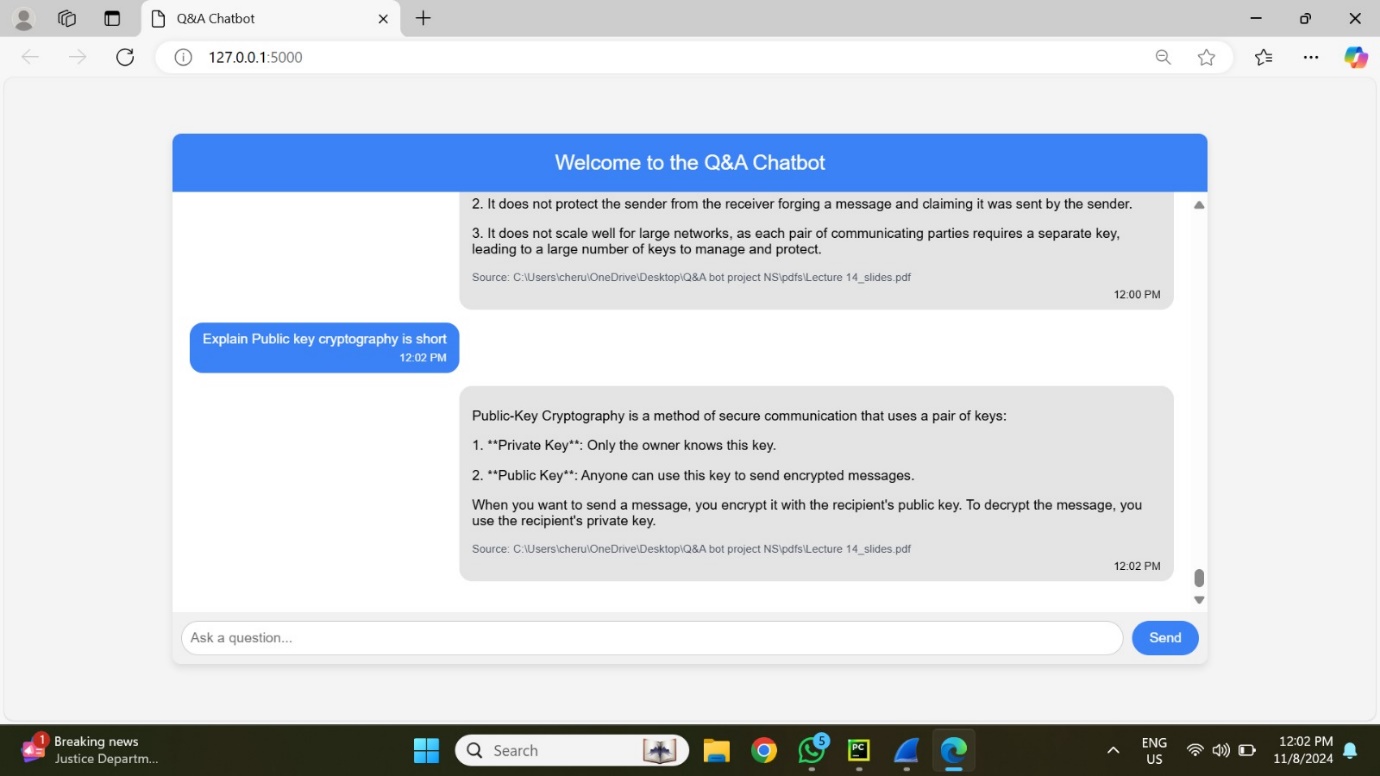
**Frame Length:** Response packets are often shorter than request packets because they simply include the text response.

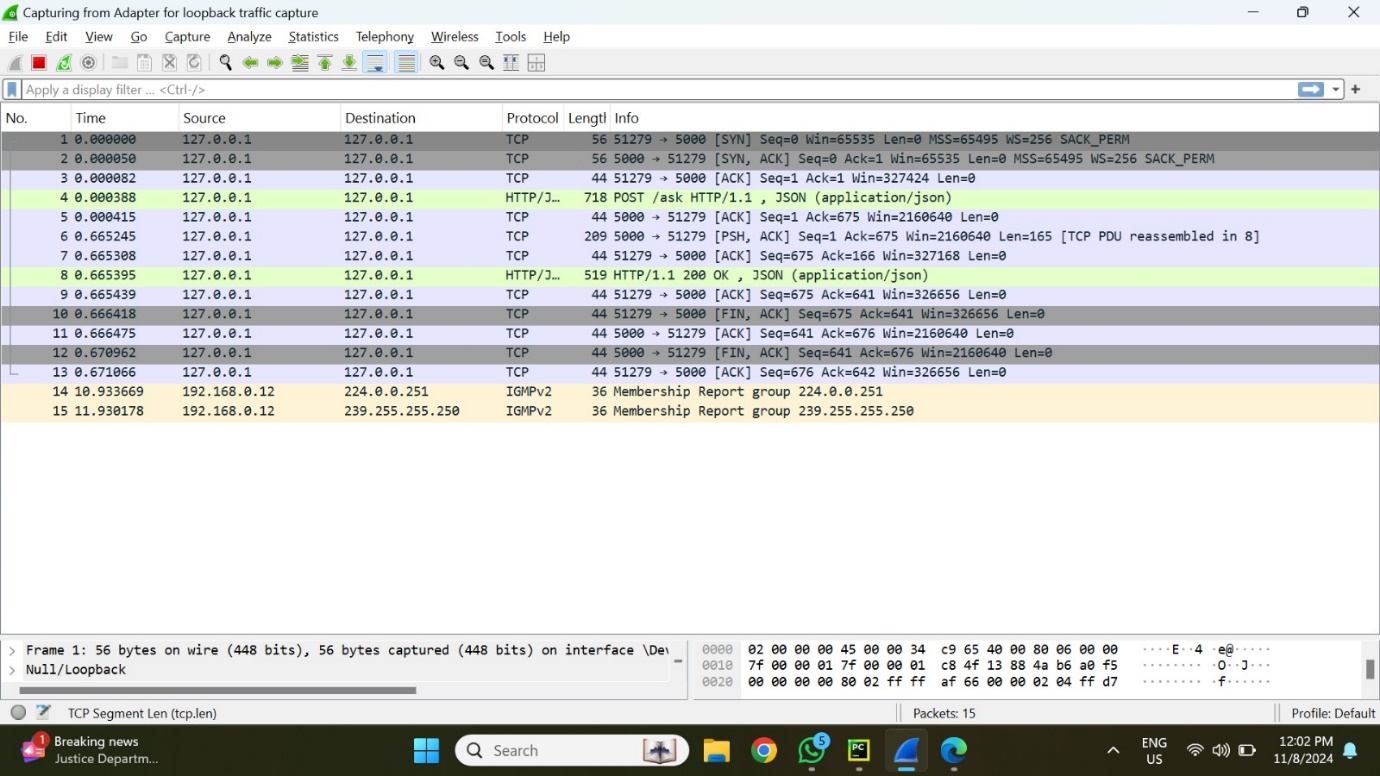
**Data Representation:** Wireshark displays the hexadecimal representation of the response data, including the exact bytes transferred.

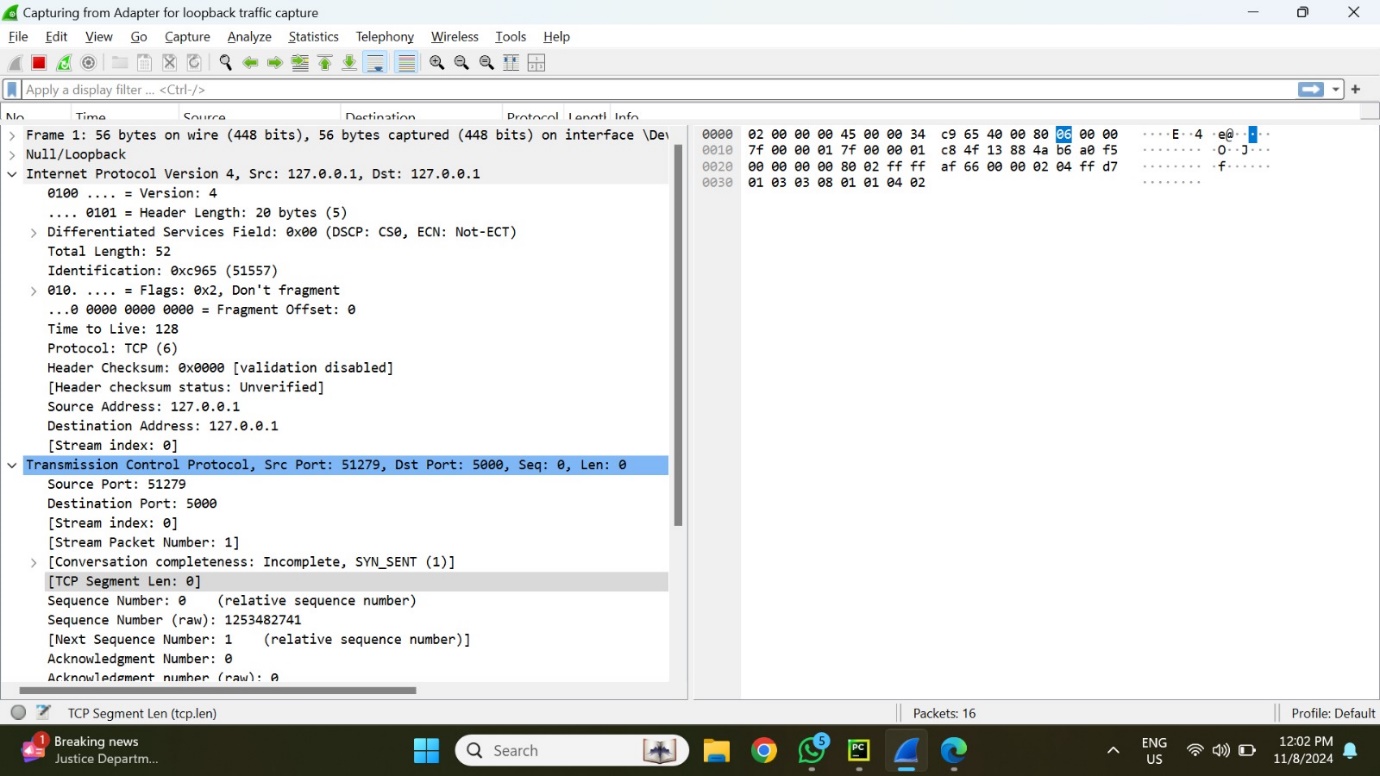
**Prompt-4:**

**1. User Query Transmission (Packets 1-8)**

Procedure: The data packet is transmitted from the client (user interface) to the server on the local computer when the user inputs a query. Below is the snap of the query and response.







**Source and Destination**:

* Source IP: 127.0.0.1
* Destination IP: 127.0.0.1 (local communication)

**Port Information**:

* Source Port: 51279
* Destination Port: 5000

**Protocol**: TCP (Transmission Control Protocol)

SYN Packet (Initial connection): A TCP SYN packet sent from the client to the server initiates the connection and creates the session.

ACK packets: To verify packet receipt, acknowledgments are sent back and forth.

HTTP Request: The query is submitted in JSON format as a POST request (seen in green in the Wireshark trace). The user's query data is contained in this HTTP packet.

**Key Parameters:**

**Sequence Number:** Begins with Seq=0 and increases with each packet.

**Acknowledgment:** The receiving side acknowledges each packet.

**Data Length:** The HTTP POST request packet (Packet 4) is longer due to the JSON payload (user query).

2. **Server Processing and Response (Packets 8-13)**

**Process:** The backend server executes the query by getting appropriate embeddings from Chroma and producing a response using the local language model. The response, including the source reference, is then returned to the user interface.

**HTTP Response:** The client receives a 200 OK status code, signifying successful processing.  
The response provides the query's answer as well as a reference that specifies the particular source (for example, "Lecture 14 slides").

**Packet Structure:** HTTP Response Packet (Packet 8) This packet comprises the server's JSON response, which includes both the generated answer and the citation.

**Key Parameters:**

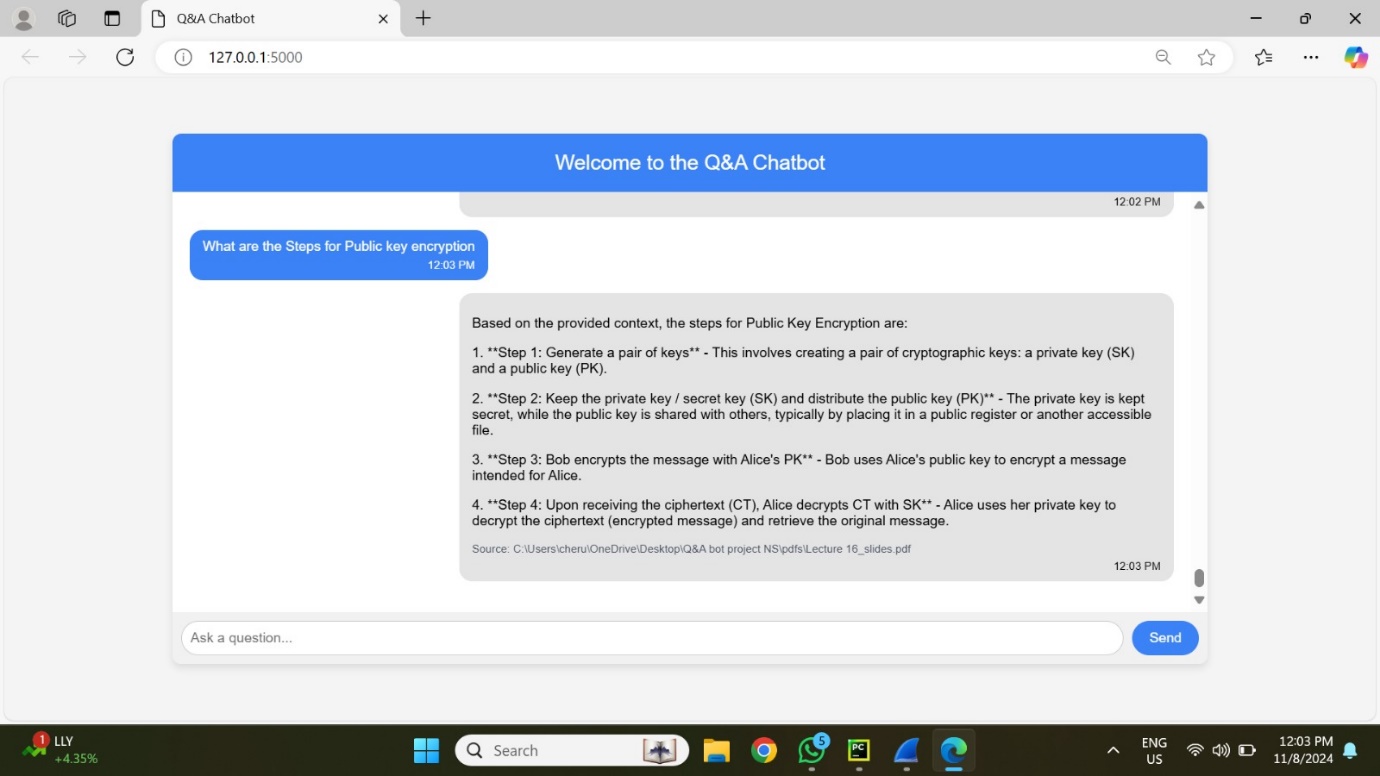
**Frame Length:** Response packets are often shorter than request packets because they simply include the text response.

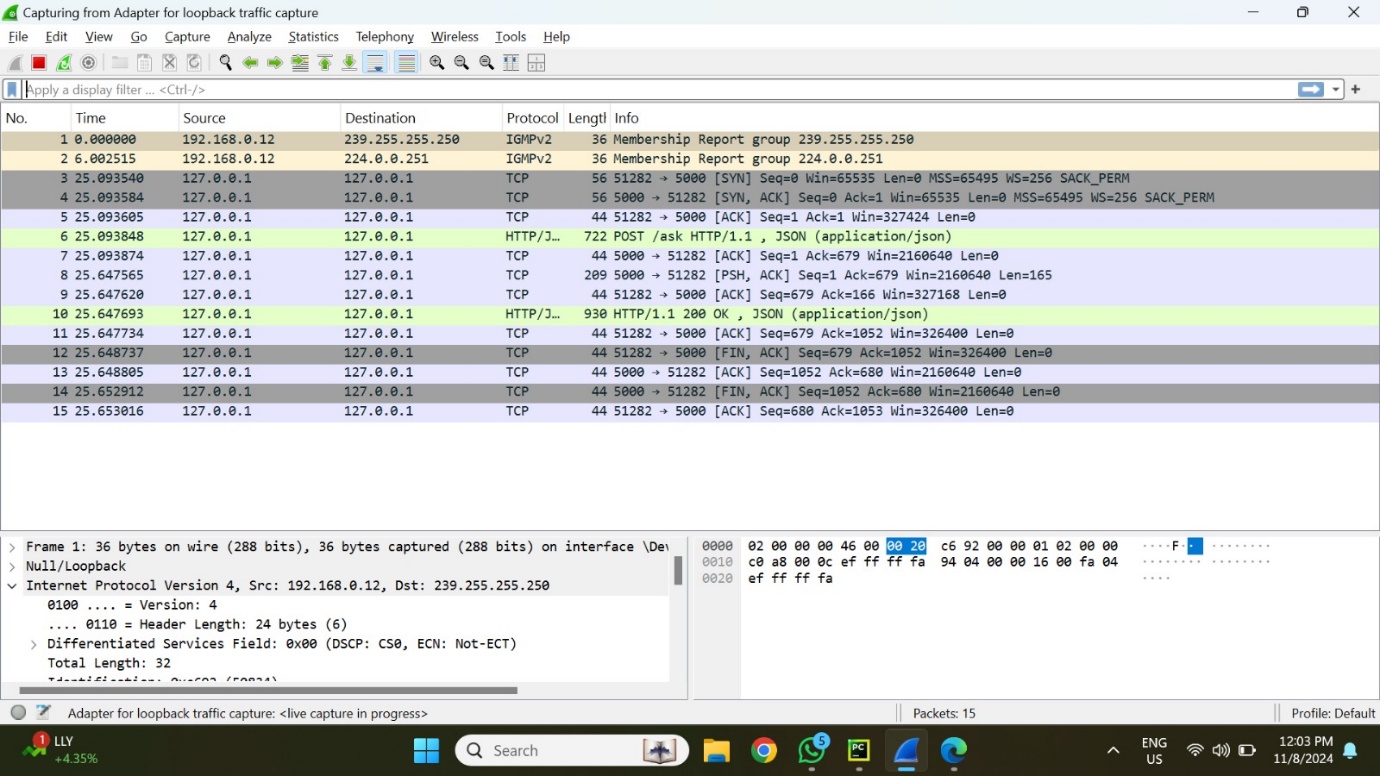
**Data Representation:** Wireshark displays the hexadecimal representation of the response data, including the exact bytes transferred.

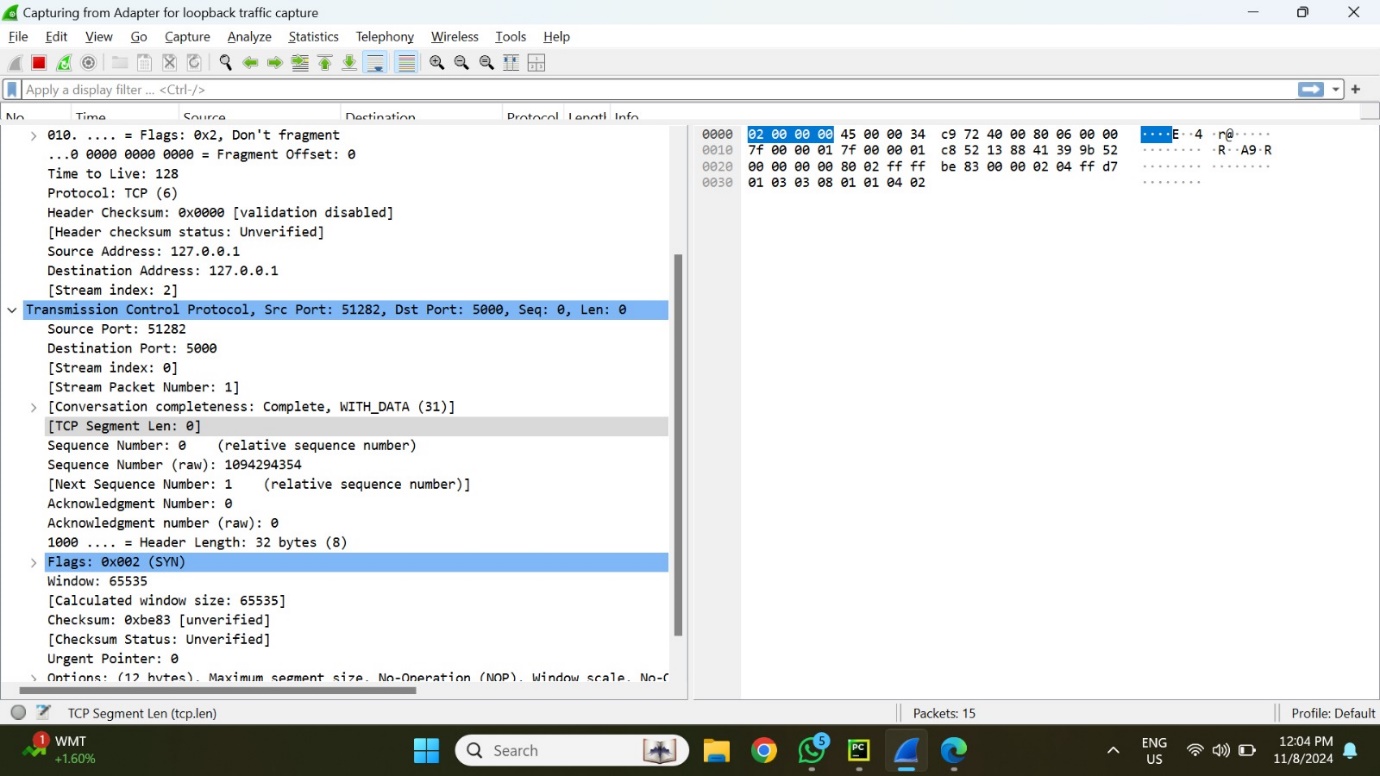
**Prompt-5:**

**1. User Query Transmission (Packets 1-8)**

Procedure: The data packet is transmitted from the client (user interface) to the server on the local computer when the user inputs a query. Below is the snap of the query and response.







**Source and Destination**:

* Source IP: 127.0.0.1
* Destination IP: 127.0.0.1 (local communication)

**Port Information**:

* Source Port: 51282
* Destination Port: 5000

**Protocol**: TCP (Transmission Control Protocol)

SYN Packet (Initial connection): A TCP SYN packet sent from the client to the server initiates the connection and creates the session.

ACK packets: To verify packet receipt, acknowledgments are sent back and forth.

HTTP Request: The query is submitted in JSON format as a POST request (seen in green in the Wireshark trace). The user's query data is contained in this HTTP packet.

**Key Parameters:**

**Sequence Number:** Begins with Seq=0 and increases with each packet.

**Acknowledgment:** The receiving side acknowledges each packet.

**Data Length:** The HTTP POST request packet (Packet 4) is longer due to the JSON payload (user query).

2. **Server Processing and Response (Packets 8-13)**

**Process:** The backend server executes the query by getting appropriate embeddings from Chroma and producing a response using the local language model. The response, including the source reference, is then returned to the user interface.

**HTTP Response:** The client receives a 200 OK status code, signifying successful processing.  
The response provides the query's answer as well as a reference that specifies the particular source (for example, "Lecture 16 slides").

**Packet Structure:** HTTP Response Packet (Packet 8) This packet comprises the server's JSON response, which includes both the generated answer and the citation.

**Key Parameters:**

**Frame Length:** Response packets are often shorter than request packets because they simply include the text response.

**Data Representation:** Wireshark displays the hexadecimal representation of the response data, including the exact bytes transferred.