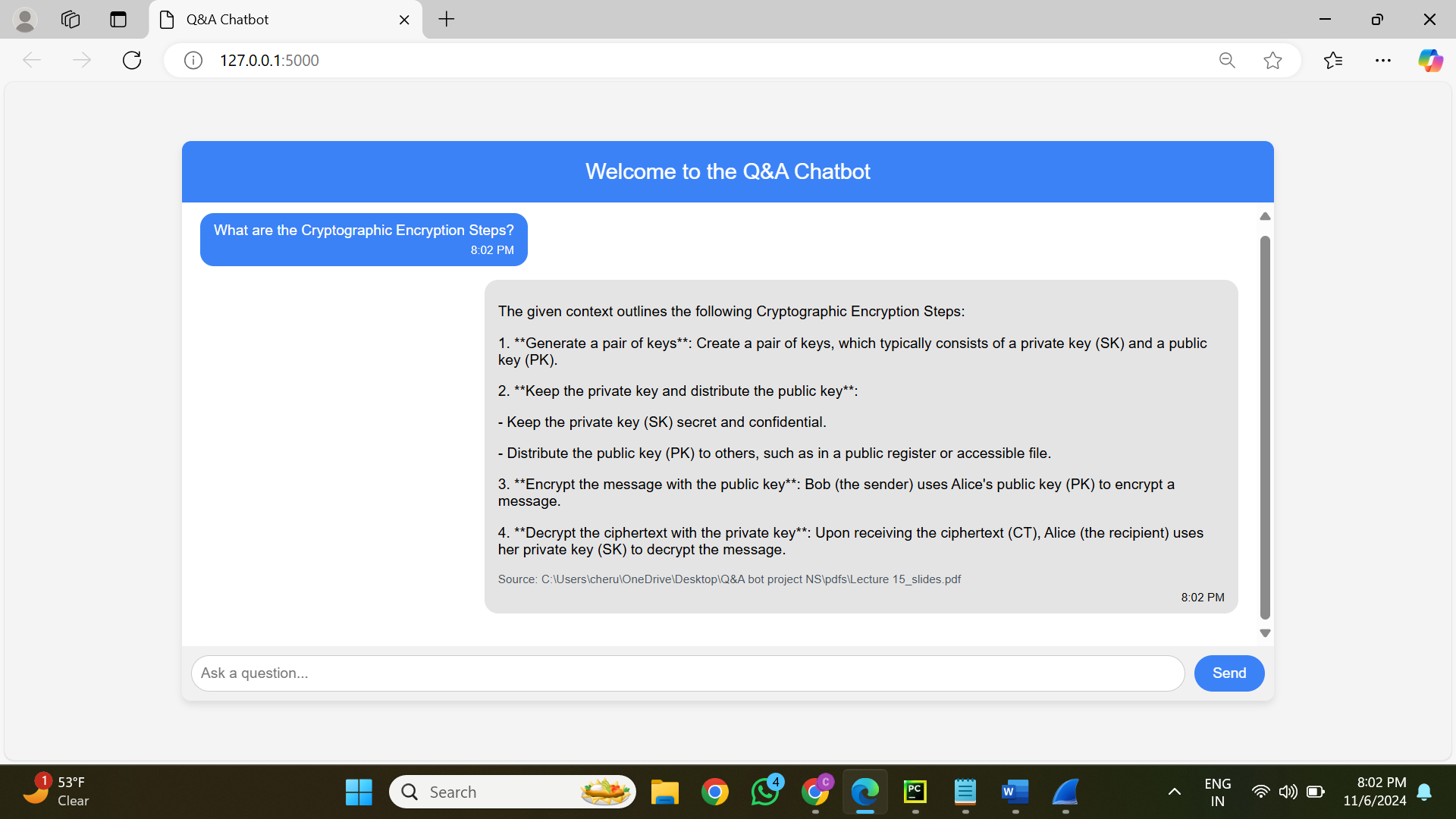
**OnDemand Professor Q&A Bot**

[**[GitHub Link]**](https://github.com/cherupallybhanuteja/Q-A_Bot_NetworkSecurity)

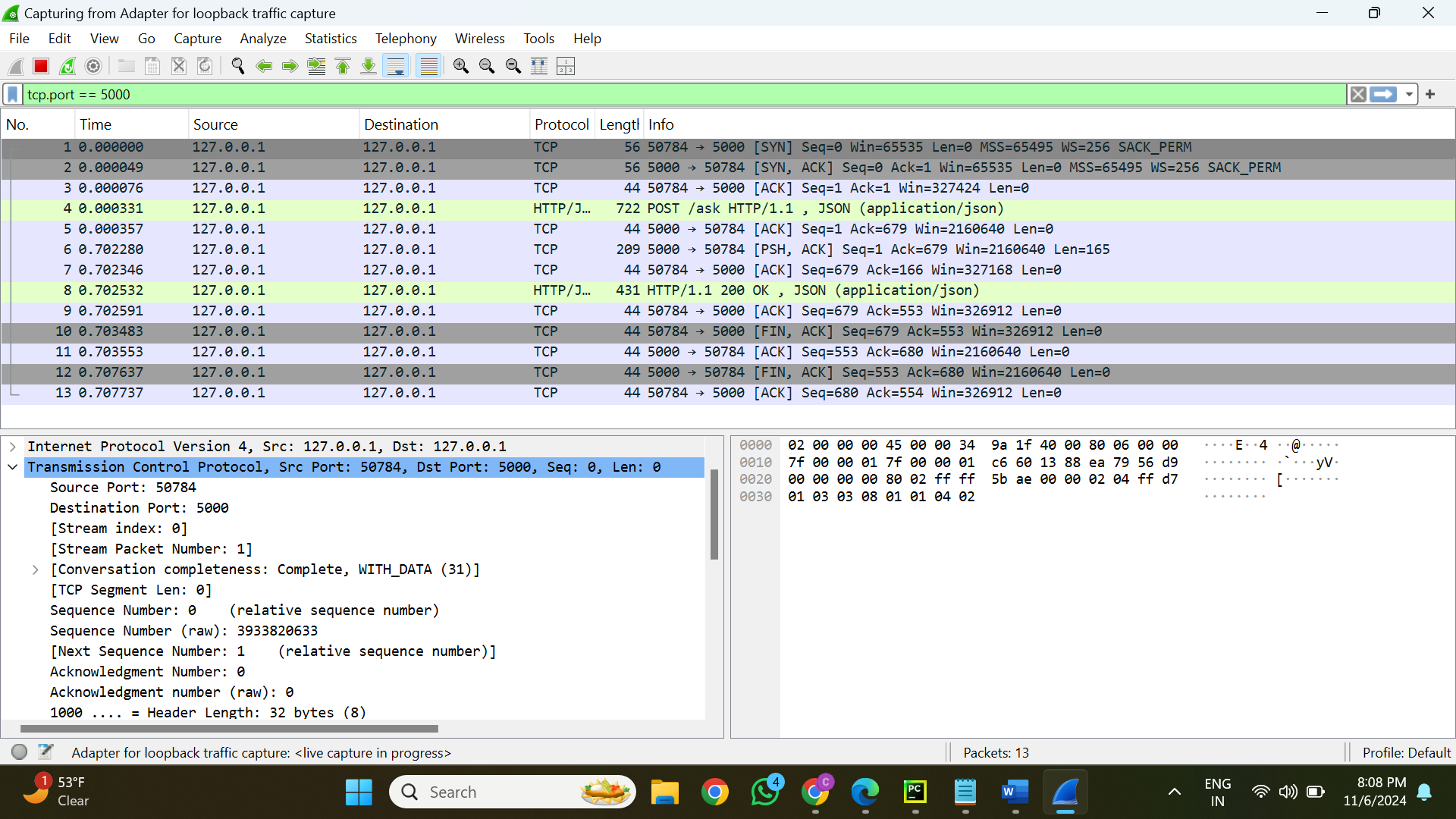
**ROUND – 2: Project Report**

**BHANU TEJA CHERUPALLY - R11892937**

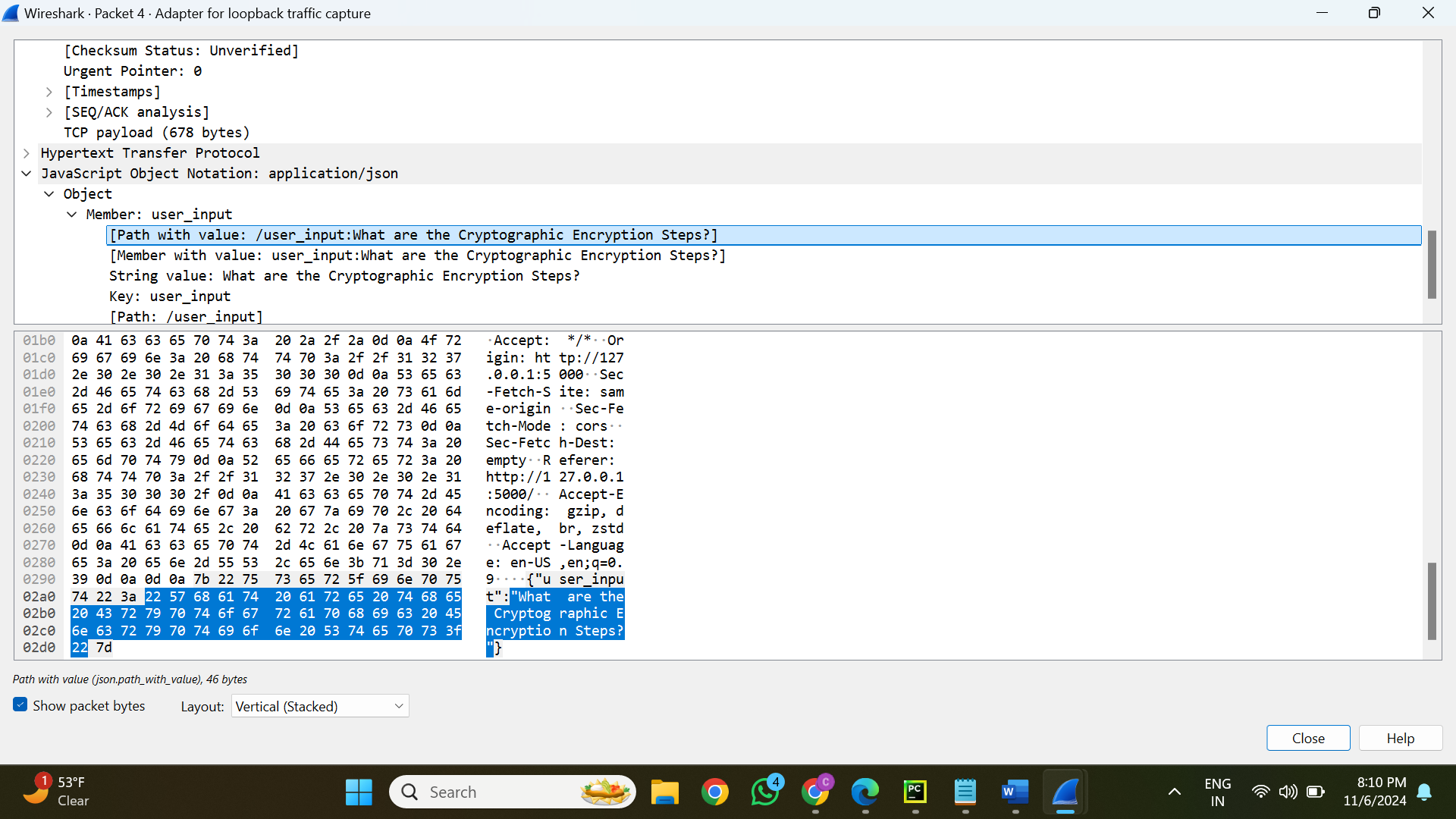
**PROMPT - 1**



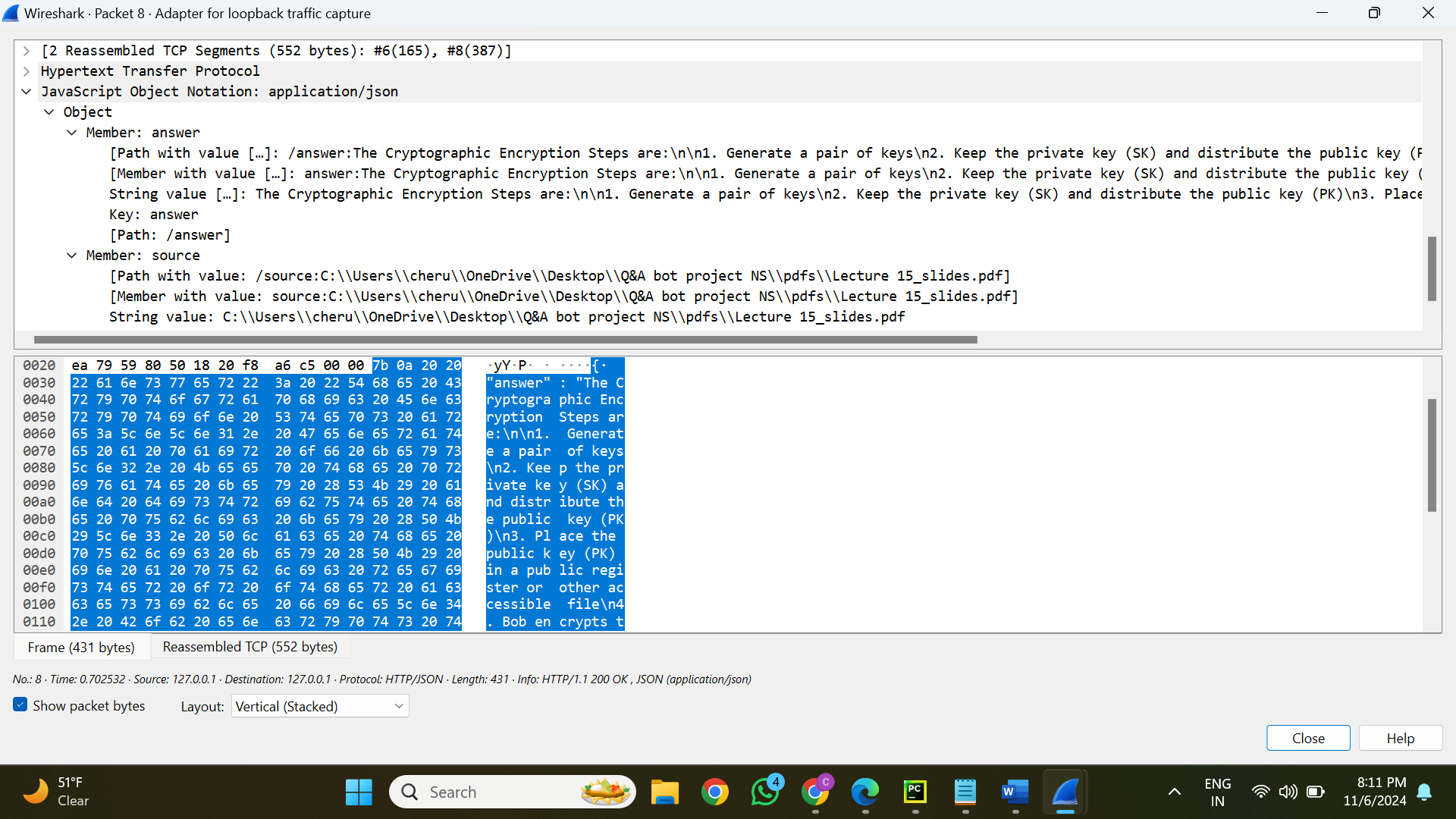
**Prompt Submission and Network Traffic Capture:**



**User Prompt Details in HTTP Request:**



**Server Response with Answer to User Prompt:**



**Step 1: Prompt:**

* **User Prompt:** “What are the Cryptographic Encryption Steps?”
* **Purpose:** To obtain the cryptographic encryption procedures.

**Step 4: Trace Data (Captured via Wireshark):**

1. **Request Packet Details:**
   * **Source Port:** 50784
   * **Destination Port:** 5000
   * **Protocol:** HTTP (POST request)
   * **Payload:** Contains the JSON object:

{ "user\_input": "What are the Cryptographic Encryption Steps?" }

1. **Response Packet Details:**
   * **Source Port:** 5000
   * **Destination Port:** 50784
   * **Protocol:** HTTP (200 OK)
   * **Payload:** The response includes:

The Cryptographic Encryption Steps are:

1. Generate a pair of keys.

2. Keep the private key (SK) and distribute the public key (PK).

3. Encrypt the message with the public key.

4. Decrypt the ciphertext with the private key.

* + **Citation:** Reference to the source document: Lecture 15\_slides.pdf.

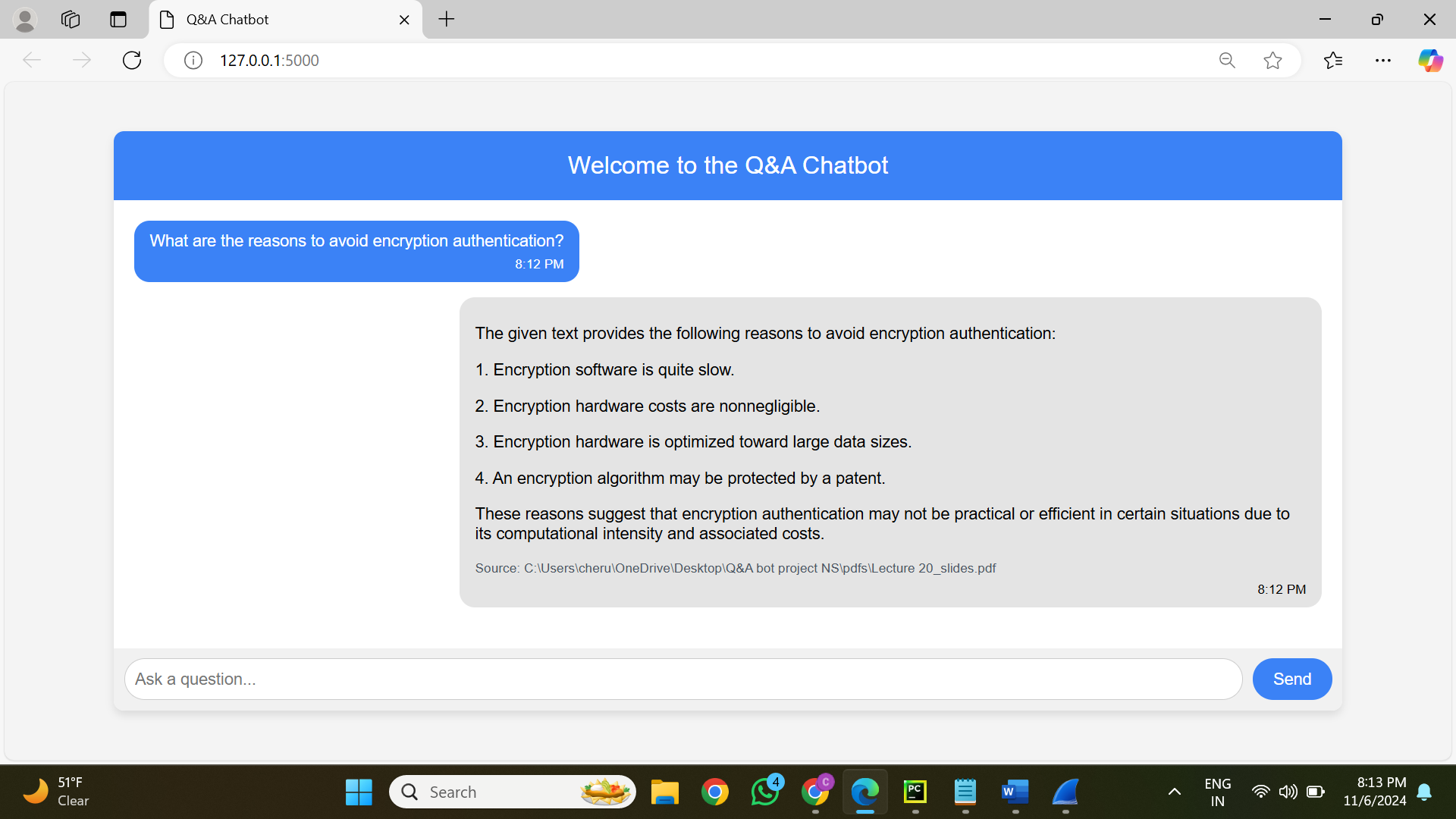
**Mapping of Step 1 to Step 4:**

* **Input (Prompt):** Included in the POST request as JSON.
* **Output (Response):** Provided to the client in JSON format after being retrieved from the LLM, along with citation information.

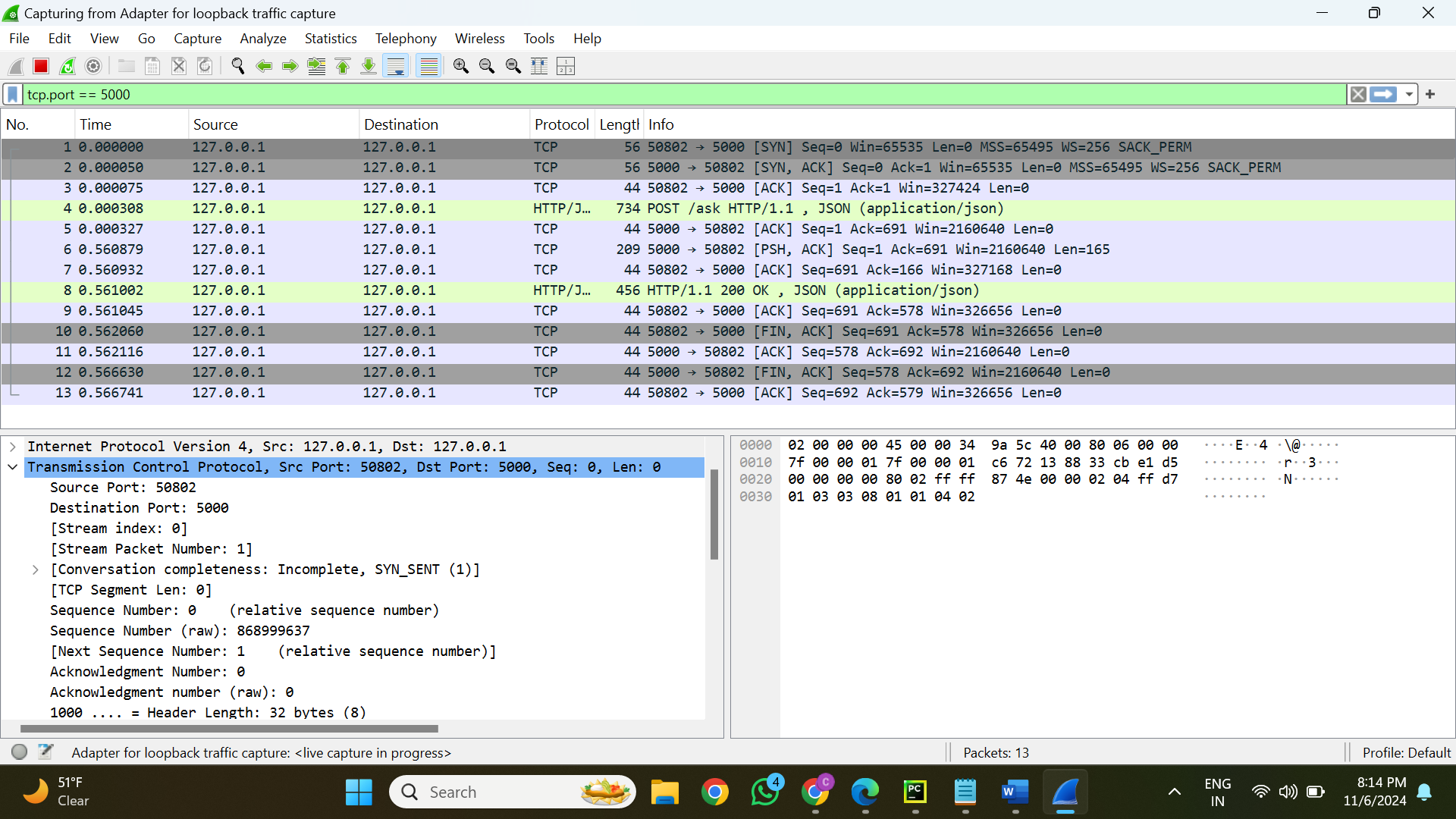
**Analysis of Captured Data:**

* **Request:** The server receives the user's prompt and processes it.
* **Processing:** The backend makes use of:
  + **Embedding Model:** Generates a numerical vector from the prompt.
  + **Vector Database:** Uses embeddings to retrieve pertinent document material.
  + **LLM:** Uses the context that was retrieved to generate the answer.
* **Response:** The user interface receives a JSON-encapsulated version of the processed response.

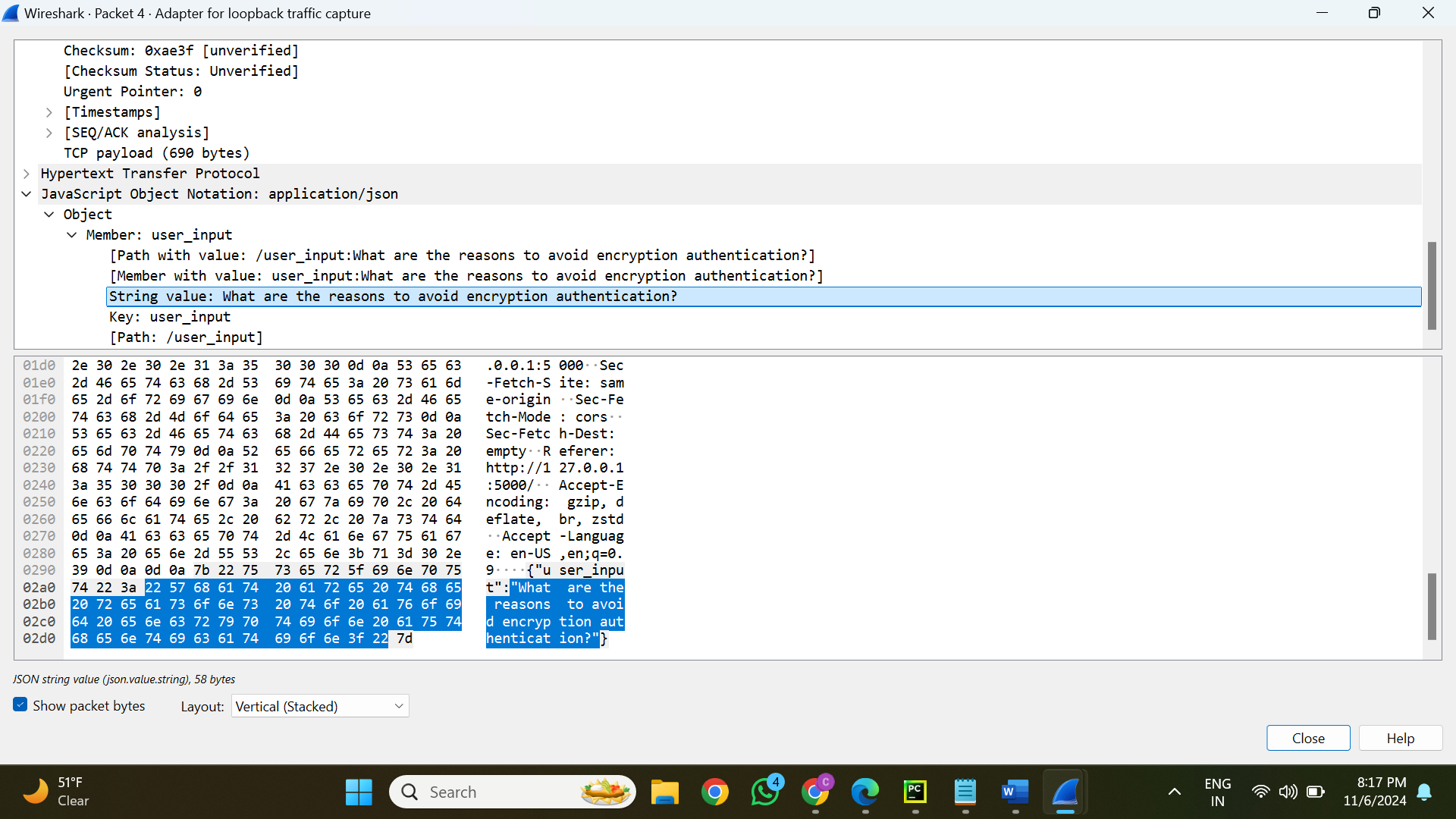
**PROMPT - 2**



**Prompt Submission and Network Traffic Capture:**



**User Prompt Details in HTTP Request:**



**Server Response with Answer to User Prompt:**



**Step 1: Prompt:**

* **User Prompt:** “What are the reasons to avoid encryption authentication?”
* **Purpose:** To recover the justifications for not using encryption authentication

**Step 4: Trace Data (Captured via Wireshark):**

**1. Request Packet Details:**

* + **Source Port:** 50802
  + **Destination Port:** 5000
  + **Protocol:** HTTP (POST request)
  + **Payload:** Contains the JSON object:

{ "user\_input": " What are the reasons to avoid encryption authentication?” }

**2. Response Packet Details:**

* + **Source Port:** 5000
  + **Destination Port:** 50802
  + **Protocol:** HTTP (200 OK)
  + **Payload:** The response includes:

According to the given context, the reasons to avoid encryption authentication are:

1. Encryption software is quite slow.

2. Encryption hardware costs are non-negligible.

3. Encryption hardware is optimized toward large data sizes.

4. An encryption algorithm may be protected by a patent.

* + **Citation:** Reference to the source document: Lecture 20\_slides.pdf

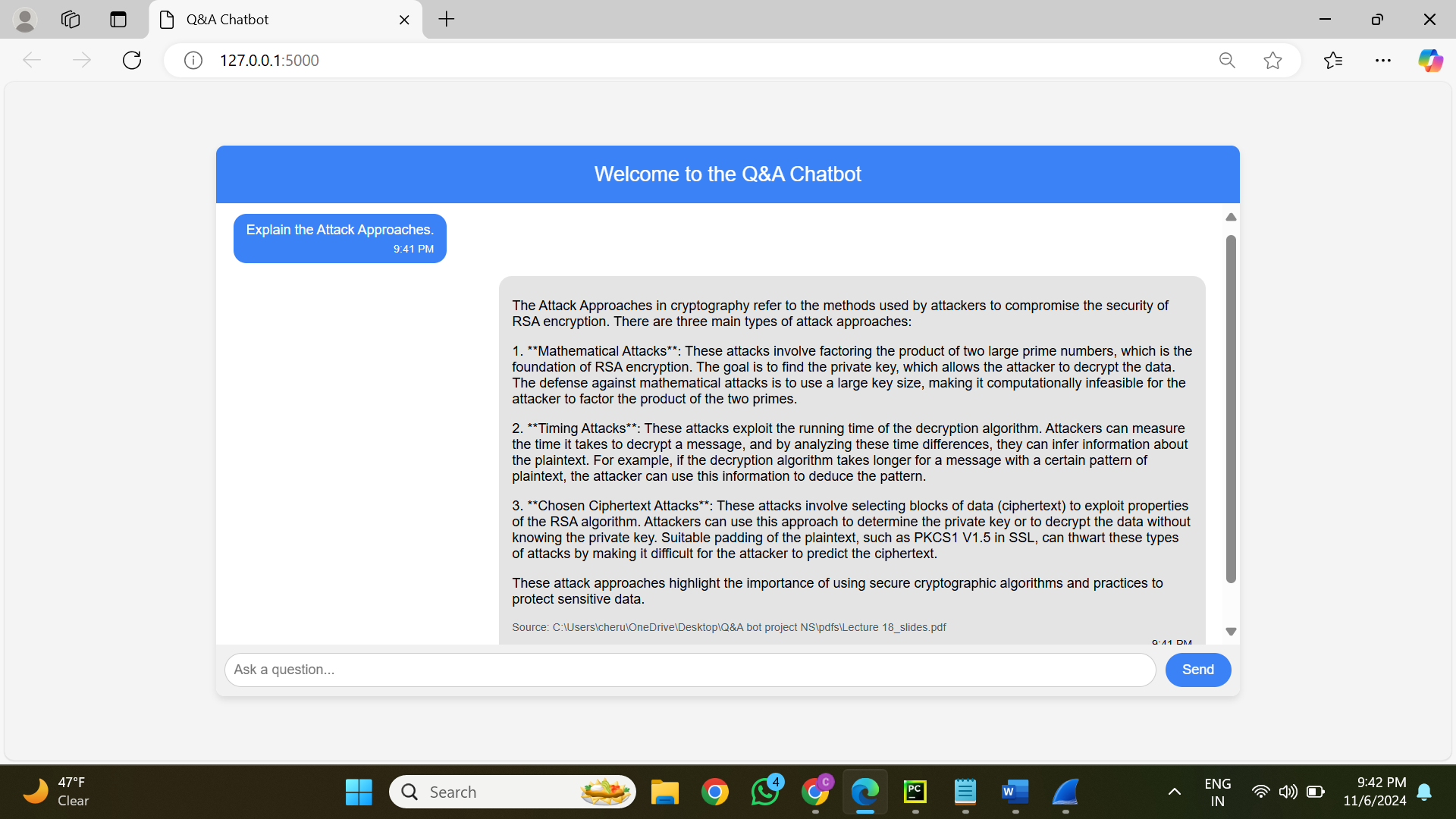
**Mapping of Step 1 to Step 4:**

* **Input (Prompt):** Included in the POST request as JSON.
* **Output (Response):** Provided to the client in JSON format after being retrieved from the LLM, along with citation information.

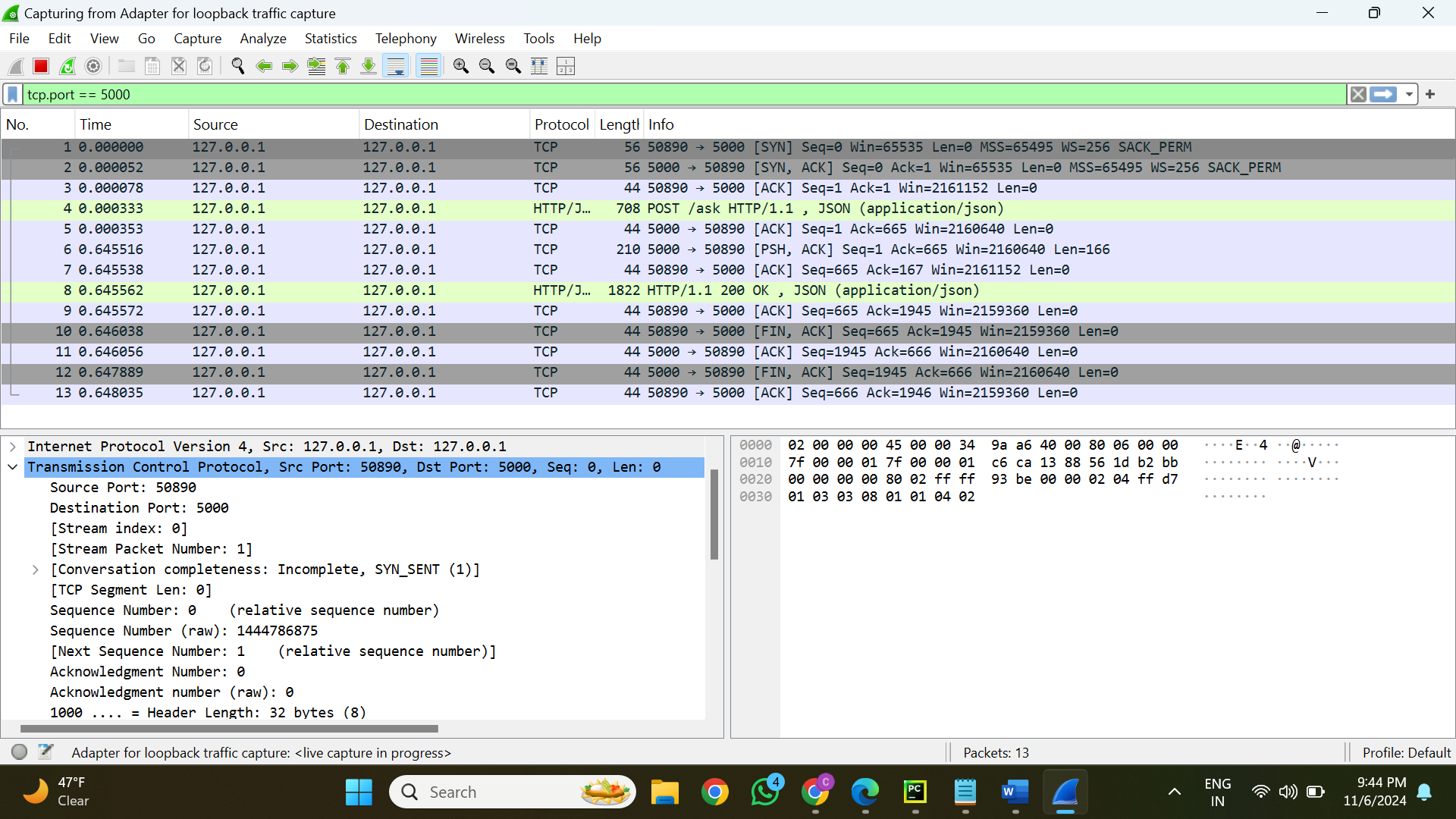
**Analysis of Captured Data:**

* **Request:** The server receives the user's prompt and processes it.
* **Processing:** The backend makes use of:
  + **Embedding Model:** Generates a numerical vector from the prompt.
  + **Vector Database:** Uses embeddings to retrieve pertinent document material.
  + **LLM:** Uses the context that was retrieved to generate the answer.
* **Response:** The user interface receives a JSON-encapsulated version of the processed response.

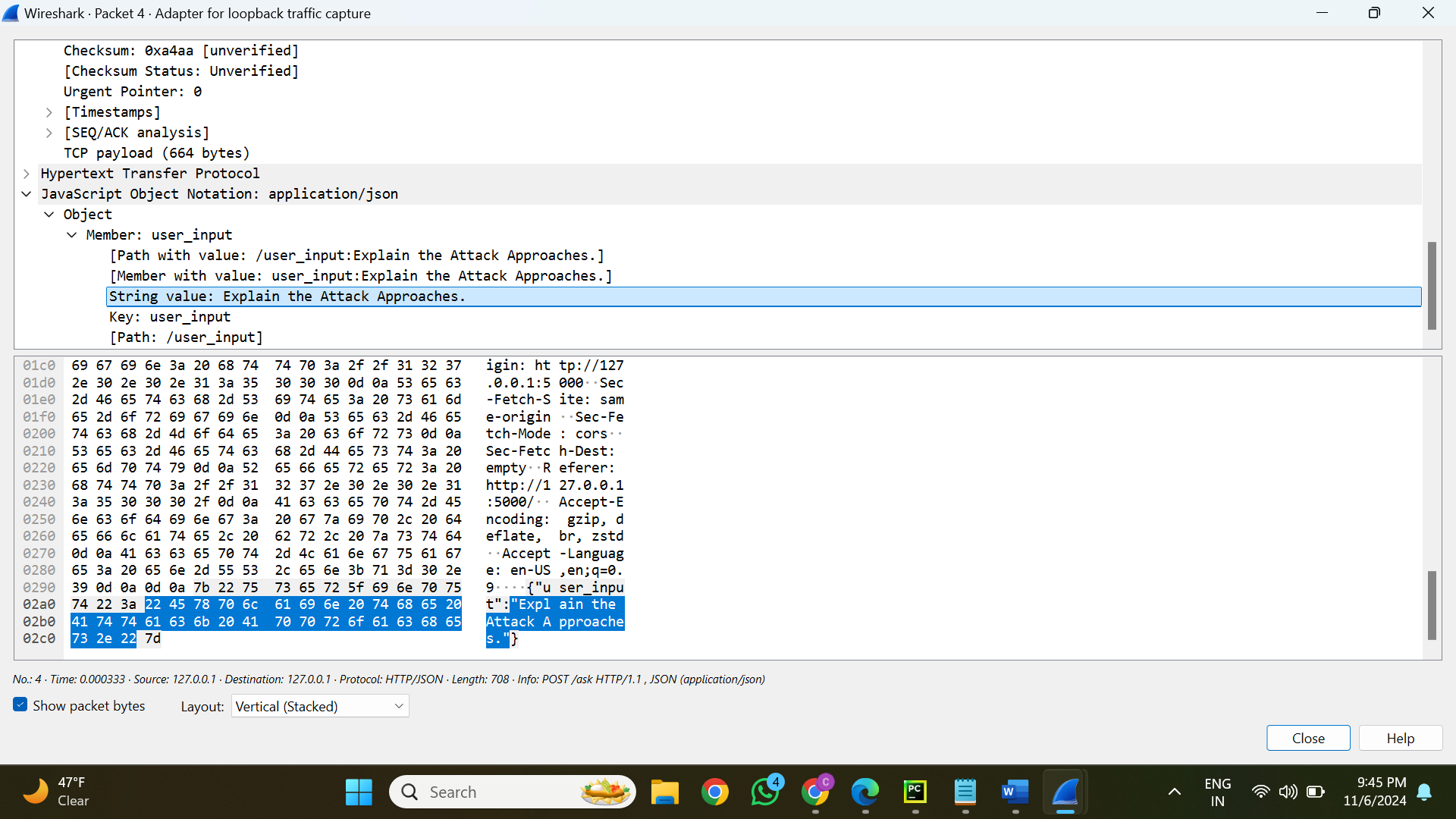
**PROMPT - 3**



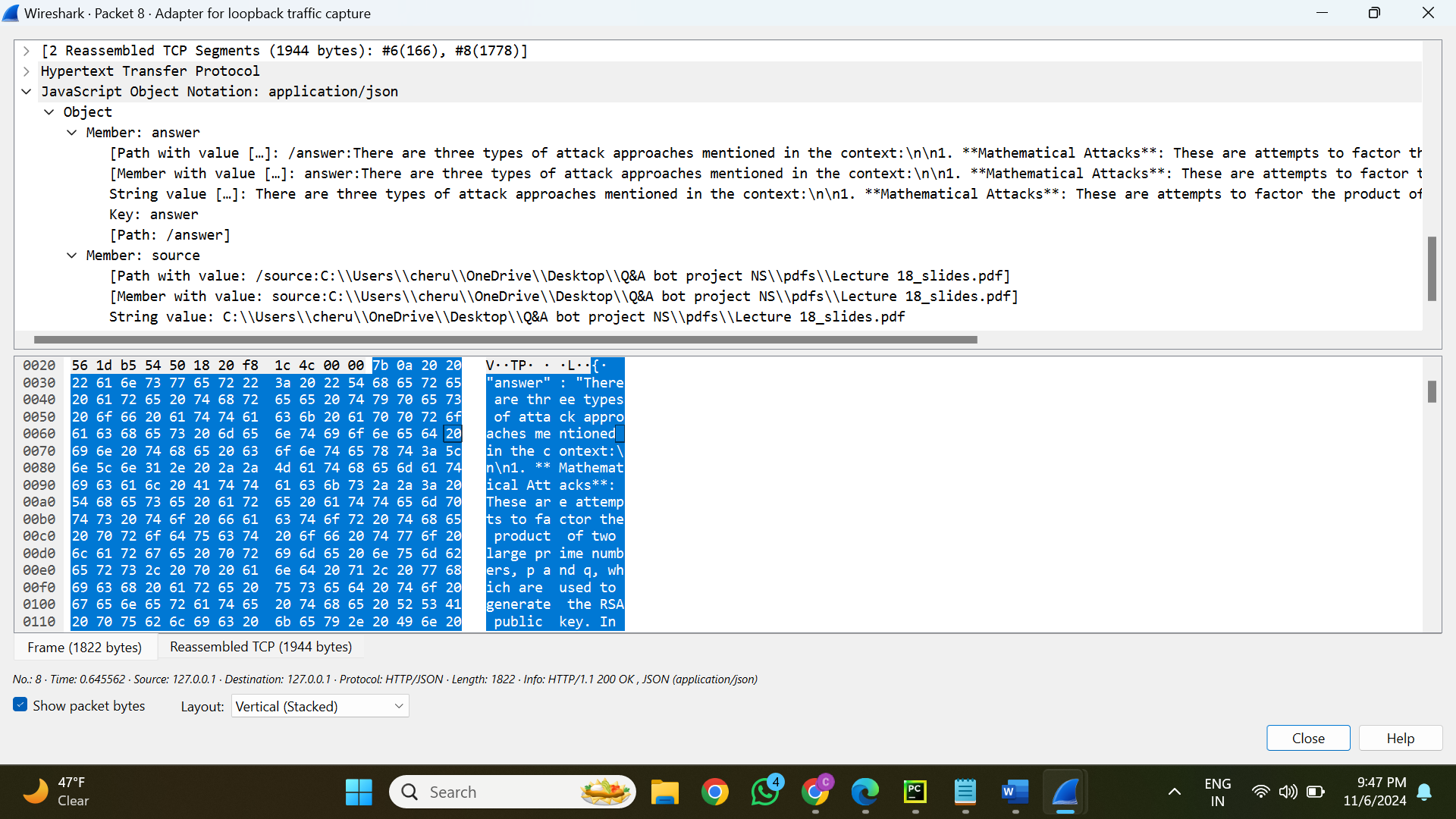
**Prompt Submission and Network Traffic Capture:**



**User Prompt Details in HTTP Request:**



**Server Response with Answer to User Prompt:**



**Step 1: Prompt:**

* **User Prompt:** “Explain the Attack Approaches.”
* **Purpose:** To obtain a description of different cryptography attack methods.

**Step 4: Trace Data (Captured via Wireshark):**

1. **Request Packet Details:**
   * **Source Port:** 50890
   * **Destination Port:** 5000
   * **Protocol:** HTTP (POST request)
   * **Payload:** Contains the JSON object:

{ "user\_input": “Explain the Attack Approaches.”}

1. **Response Packet Details:**
   * **Source Port:** 5000
   * **Destination Port:** 50890
   * **Protocol:** HTTP (200 OK)
   * **Payload:** The response includes:

The Attack Approaches in cryptography refer to methods used by attackers to compromise the security of RSA encryption. There are three main types of attack approaches:

1. Mathematical Attacks: These involve factoring the product of two large prime numbers.

2. Timing Attacks: These exploit the running time of decryption algorithms.

3. Chosen Ciphertext Attacks: These analyze blocks of ciphertext to exploit RSA's properties.

* + **Citation:** Reference to the source document: Lecture 18\_slides.pdf

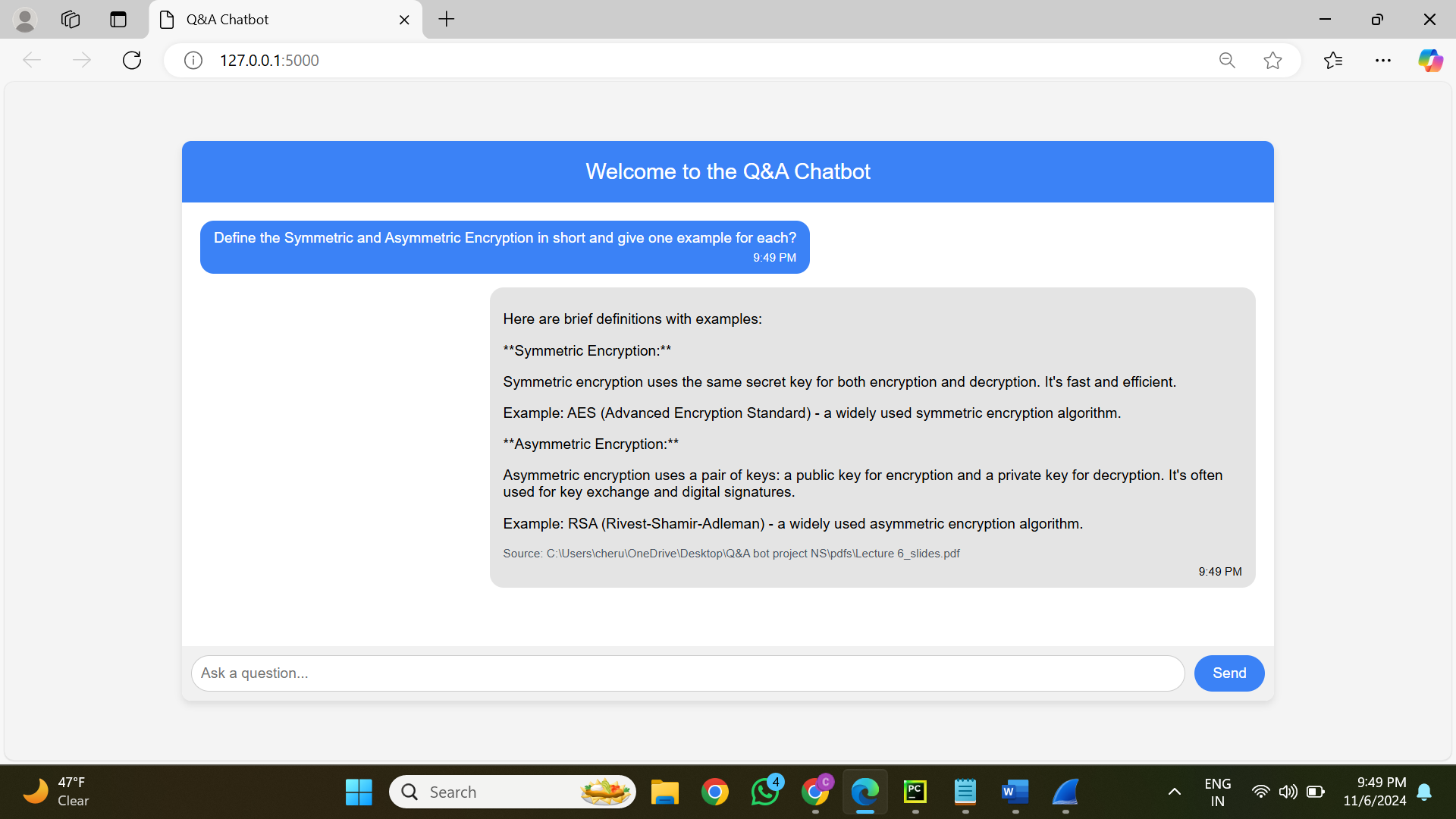
**Mapping of Step 1 to Step 4:**

* **Input (Prompt):** Included in the POST request as JSON.
* **Output (Response):** Provided to the client in JSON format after being retrieved from the LLM, along with citation information.

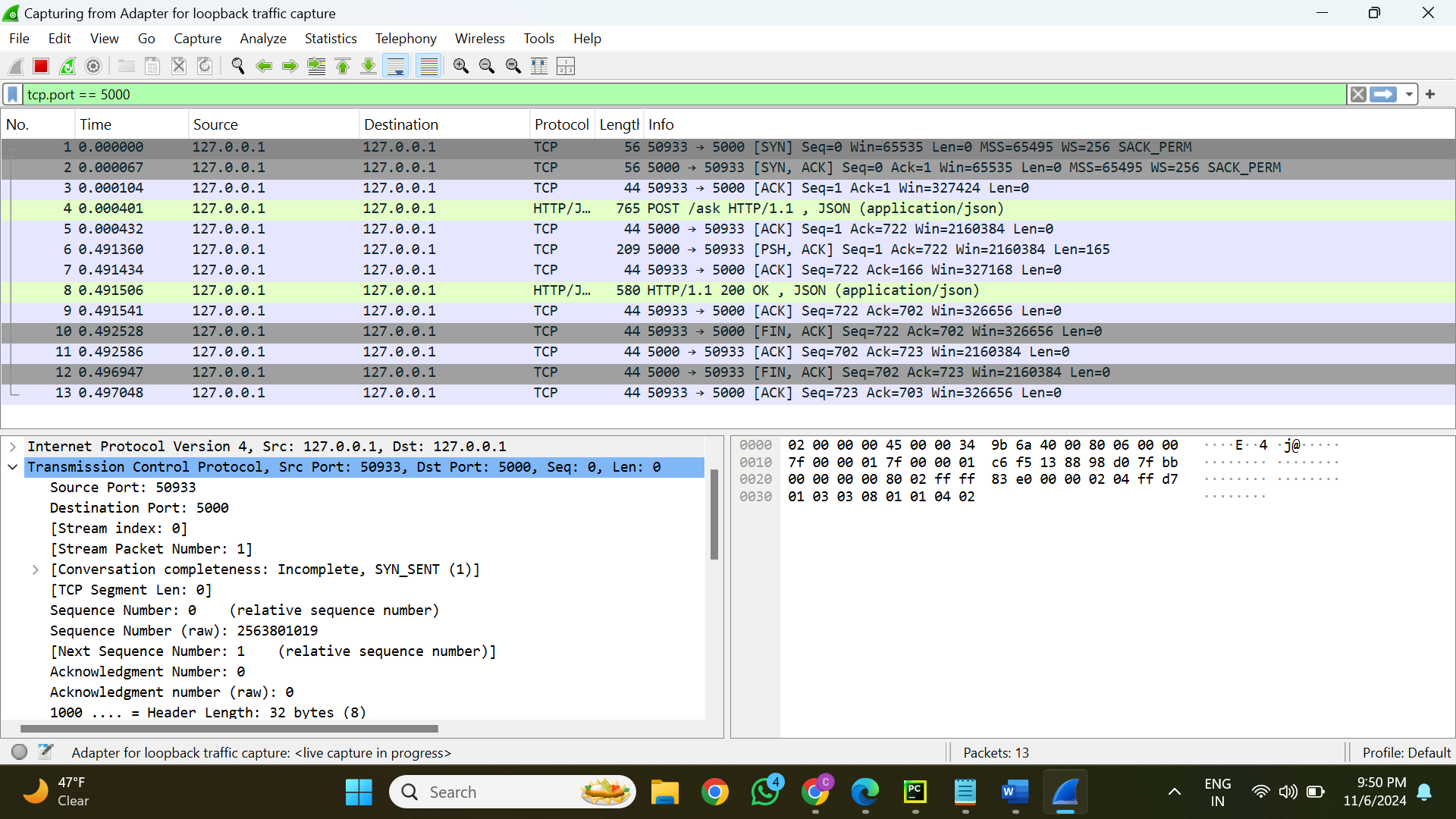
**Analysis of Captured Data:**

* **Request:** The server receives the user's prompt and processes it.
* **Processing:** The backend makes use of:
  + **Embedding Model:** Generates a numerical vector from the prompt.
  + **Vector Database:** Uses embeddings to retrieve pertinent document material.
  + **LLM:** Uses the context that was retrieved to generate the answer.
* **Response:** The user interface receives a JSON-encapsulated version of the processed response.

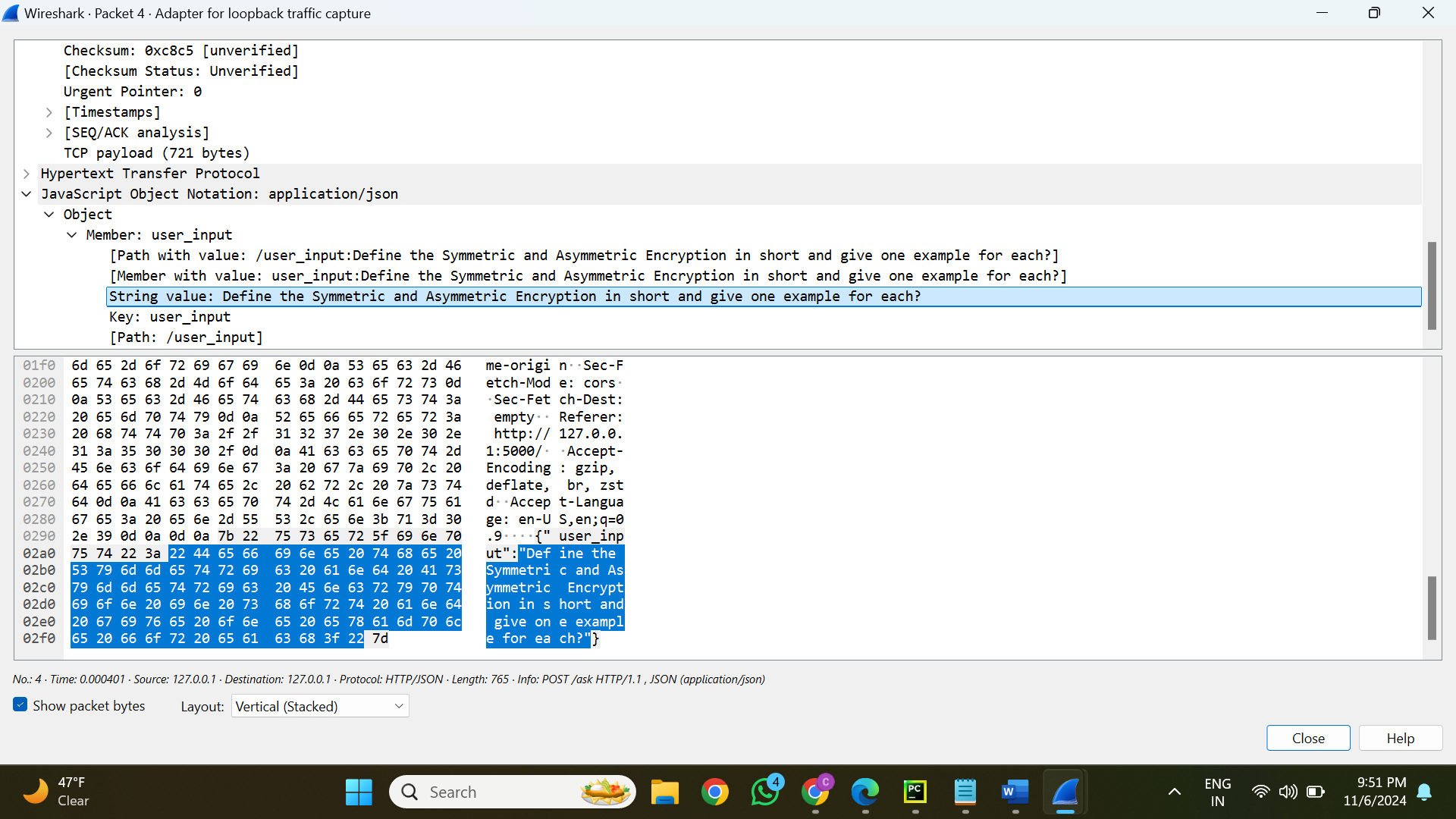
**PROMPT - 4**



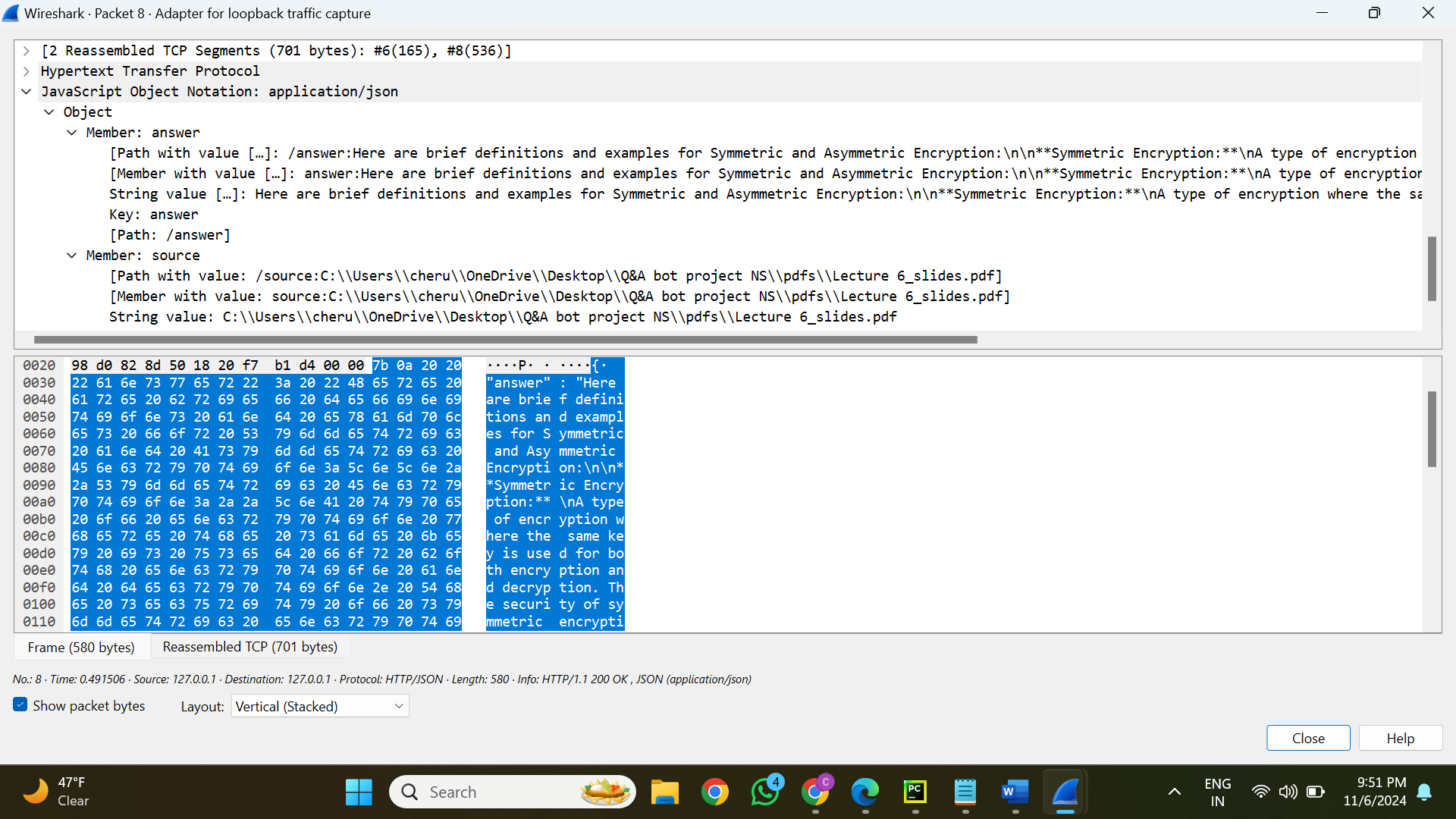
**Prompt Submission and Network Traffic Capture:**



**User Prompt Details in HTTP Request:**



**Server Response with Answer to User Prompt:**



**Step 1: Prompt:**

* **User Prompt:** “Define the Symmetric and Asymmetric Encryption in short and give one example for each.”
* **Purpose:** To obtain concise explanations on symmetric and asymmetric encryption as well as examples.

**Step 4: Trace Data (Captured via Wireshark):**

1. **Request Packet Details:**
   * **Source Port:** 50933
   * **Destination Port:** 5000
   * **Protocol:** HTTP (POST request)
   * **Payload:** Contains the JSON object:

{ "user\_input": “Define the Symmetric and Asymmetric Encryption in short and give one example for each?”}

1. **Response Packet Details:**
   * **Source Port:** 5000
   * **Destination Port:** 50933
   * **Protocol:** HTTP (200 OK)
   * **Payload:** The response includes:

Symmetric Encryption:

A type of encryption where the same secret key is used for both encryption and decryption.

Example: AES (Advanced Encryption Standard) - a widely used symmetric encryption algorithm.

Asymmetric Encryption:

A type of encryption that uses a pair of keys: a public key for encryption and a private key for decryption. Often used for key exchange and digital signatures.

Example: RSA (Rivest–Shamir–Adleman) - a widely used asymmetric encryption algorithm.

* + **Citation:** Reference to the source document: Lecture 6\_slides.pdf

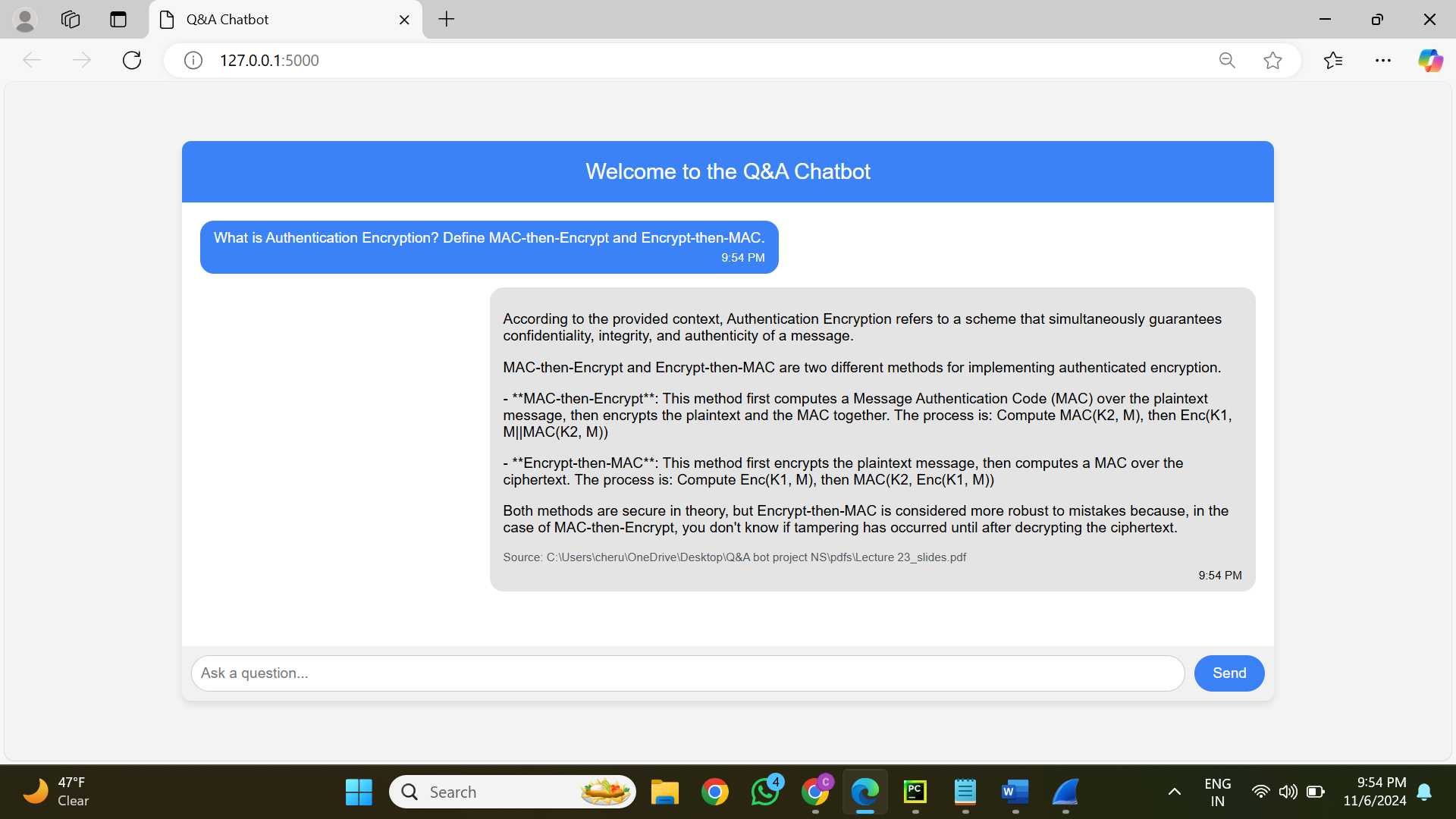
**Mapping of Step 1 to Step 4:**

* **Input (Prompt):** Included in the POST request as JSON.
* **Output (Response):** Provided to the client in JSON format after being retrieved from the LLM, along with citation information.

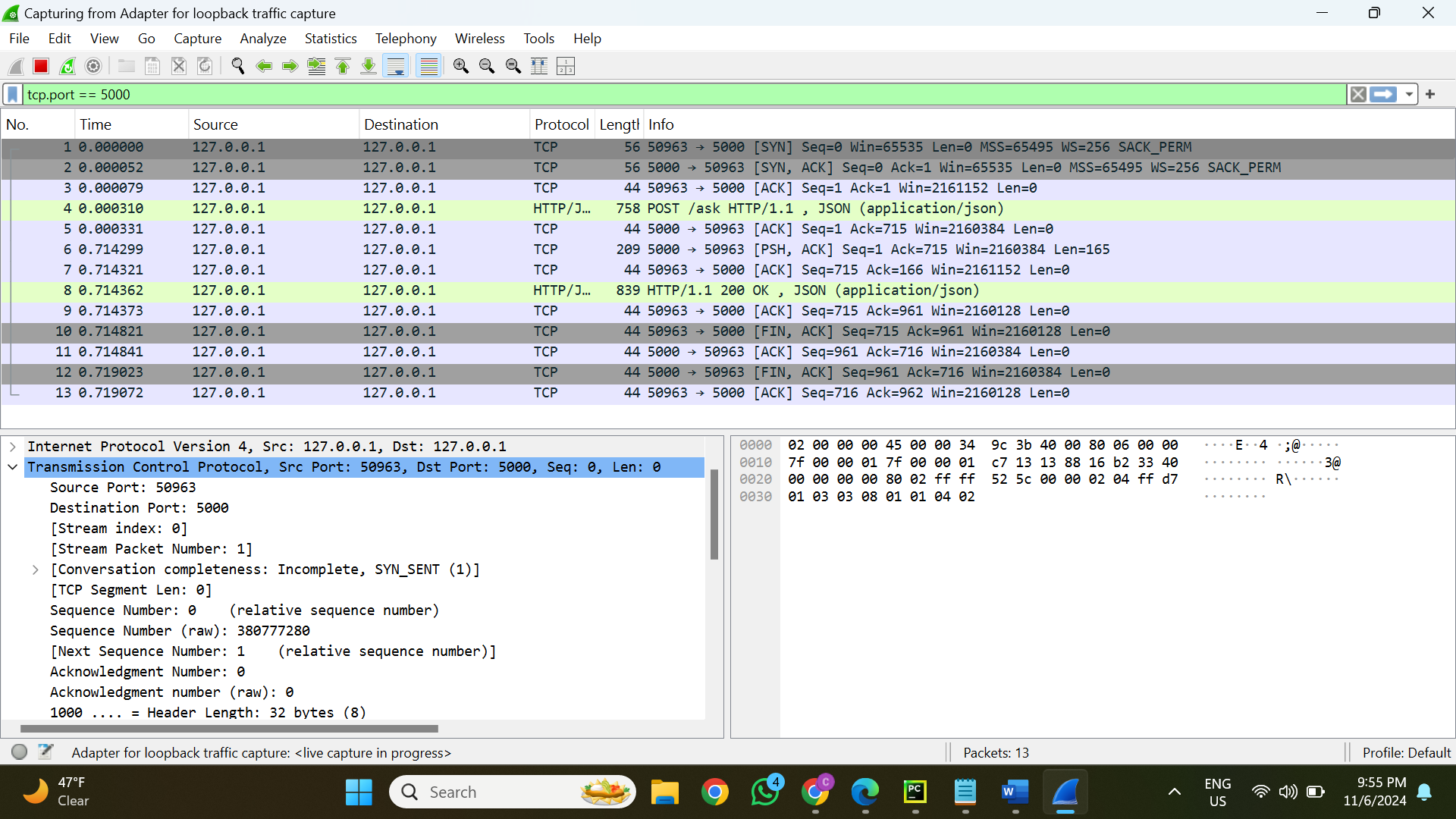
**Analysis of Captured Data:**

* **Request:** The server receives the user's prompt and processes it.
* **Processing:** The backend makes use of:
  + **Embedding Model:** Generates a numerical vector from the prompt.
  + **Vector Database:** Uses embeddings to retrieve pertinent document material.
  + **LLM:** Uses the context that was retrieved to generate the answer.
* **Response:** The user interface receives a JSON-encapsulated version of the processed response.

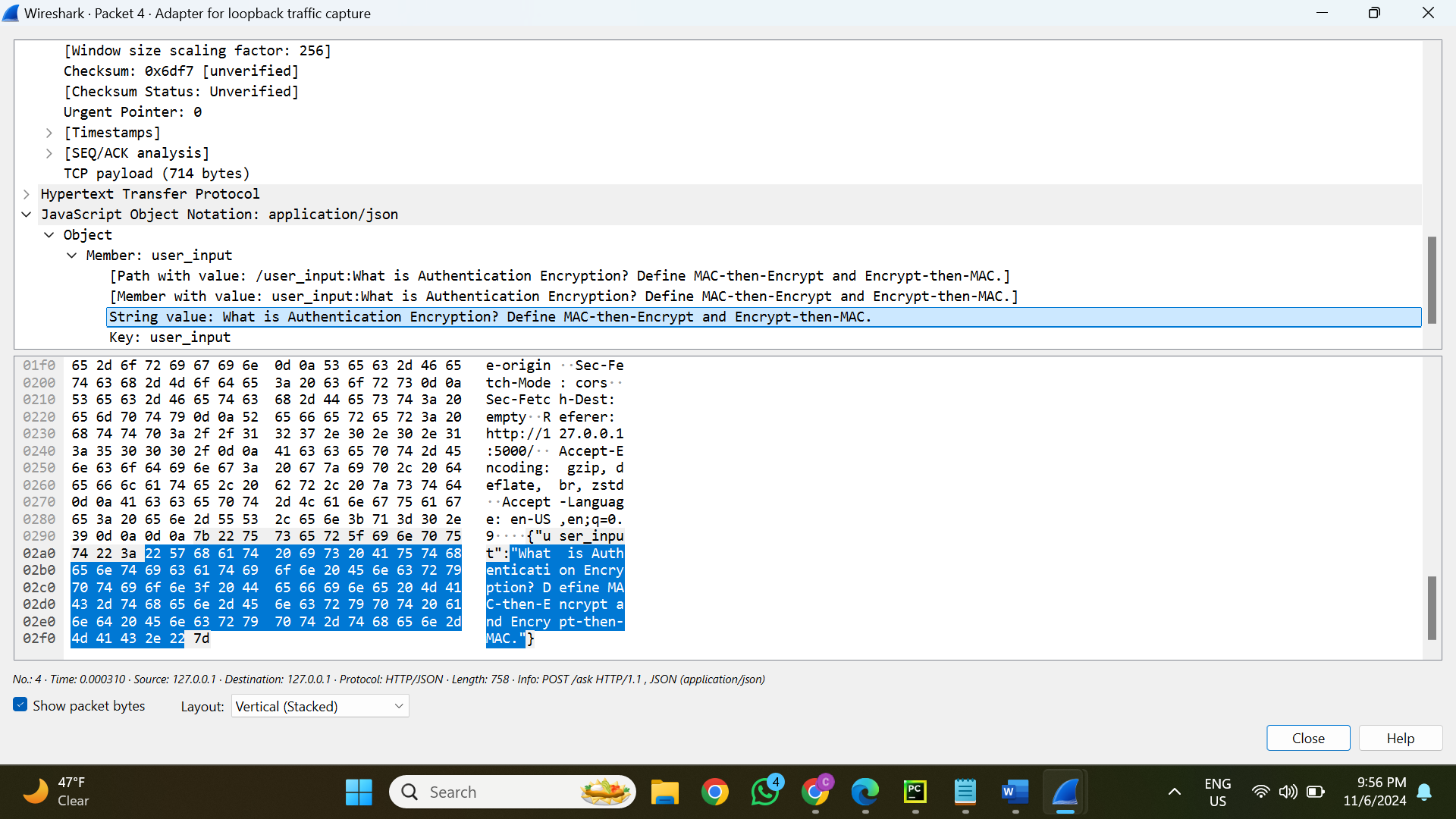
**PROMPT - 5**



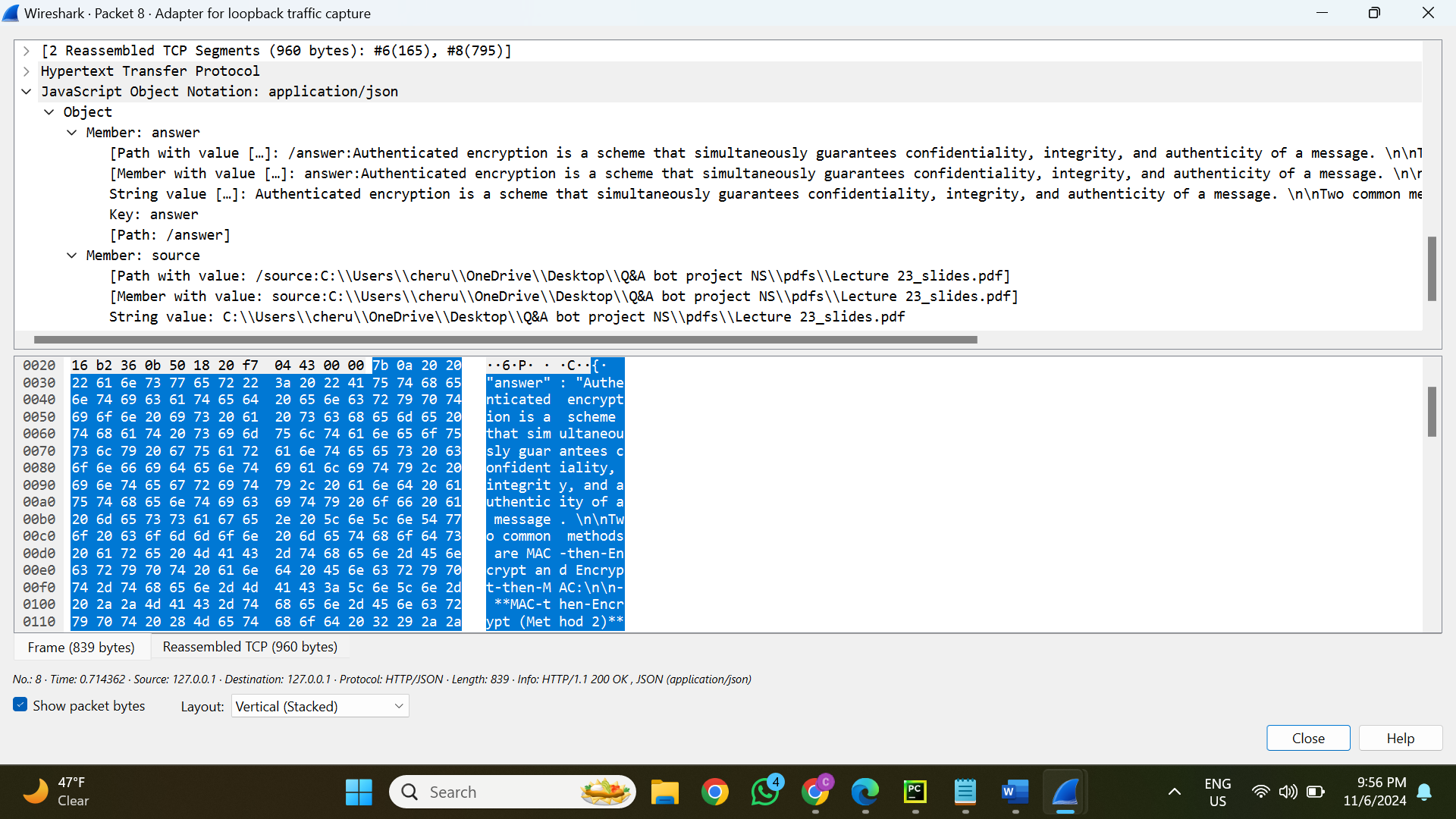
**Prompt Submission and Network Traffic Capture:**



**User Prompt Details in HTTP Request:**



**Server Response with Answer to User Prompt:**



**Step 1: Prompt:**

* **User Prompt:** “What is Authentication Encryption? Define MAC-then-Encrypt and Encrypt-then-MAC.”
* **Purpose:** To comprehend authentication encryption and differentiate between encryption-then-MAC and MAC-then-Encrypt.

**Step 4: Trace Data (Captured via Wireshark):**

1. **Request Packet Details:**
   * **Source Port:** 50963
   * **Destination Port:** 5000
   * **Protocol:** HTTP (POST request)
   * **Payload:** Contains the JSON object:

{ "user\_input": “What is Authentication Encryption? Define MAC-then-Encrypt and Encrypt-then-MAC.”}

1. **Response Packet Details:**
   * **Source Port:** 5000
   * **Destination Port:** 50963
   * **Protocol:** HTTP (200 OK)
   * **Payload:** The response includes:

Authentication Encryption refers to a scheme that simultaneously guarantees confidentiality, integrity, and authenticity of a message.

MAC-then-Encrypt: This method first computes a Message Authentication Code (MAC) over the plaintext message, then encrypts the plaintext and the MAC together. The process is: Compute MAC(K2, M), then Enc(K1, [M || MAC(K2, M)]).

Encrypt-then-MAC: This method first encrypts the plaintext message, then computes a MAC over the ciphertext. The process is: Compute Enc(K1, M), then MAC(K2, Enc(K1, M)).

Both methods are secure, but Encrypt-then-MAC is considered more robust as it avoids certain tampering risks.

* + **Citation:** Reference to the source document: Lecture 23\_slides.pdf

**Mapping of Step 1 to Step 4:**

* **Input (Prompt):** Included in the POST request as JSON.
* **Output (Response):** Provided to the client in JSON format after being retrieved from the LLM, along with citation information.

**Analysis of Captured Data:**

* **Request:** The server receives the user's prompt and processes it.
* **Processing:** The backend makes use of:
  + **Embedding Model:** Generates a numerical vector from the prompt.
  + **Vector Database:** Uses embeddings to retrieve pertinent document material.
  + **LLM:** Uses the context that was retrieved to generate the answer.
* **Response:** The user interface receives a JSON-encapsulated version of the processed response.