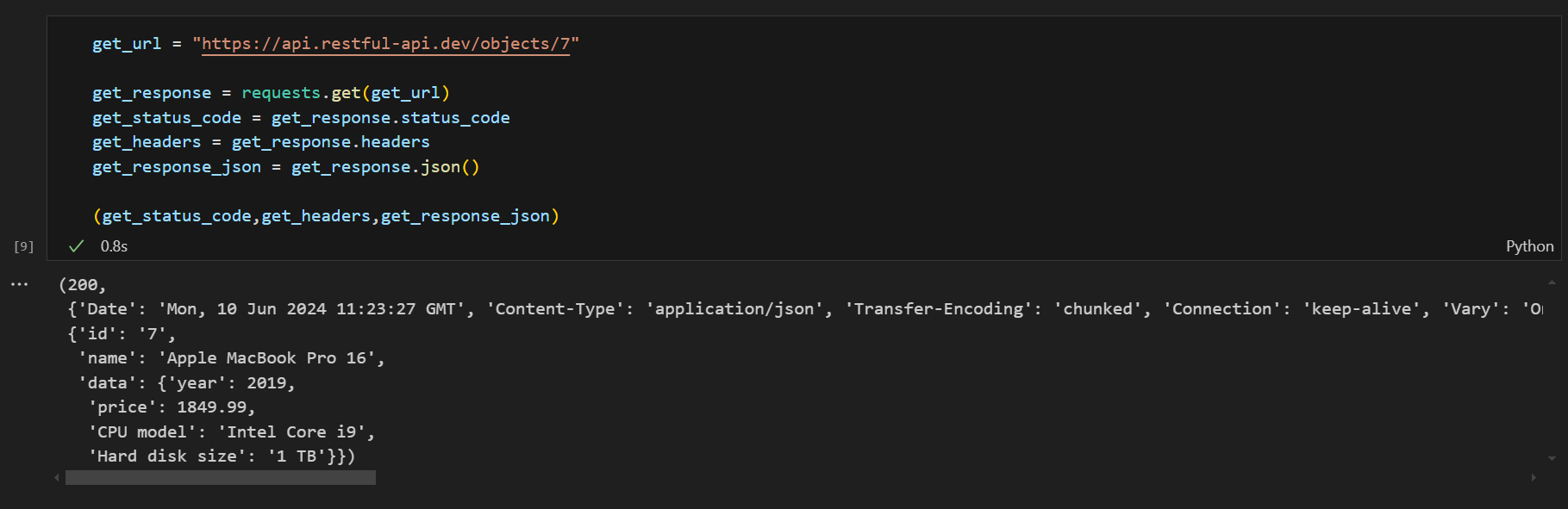
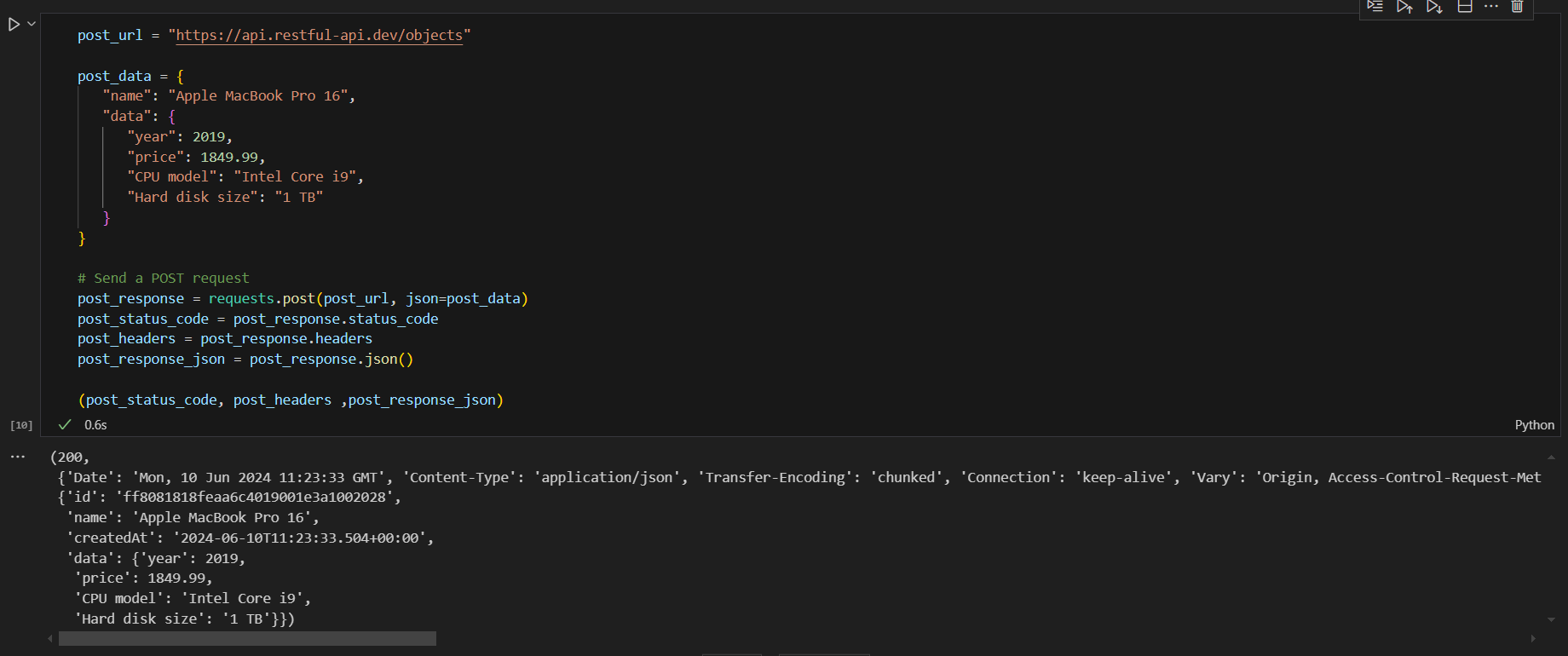
# **Network Protocols - HTTP & HTTPS**

## **HTTP Transaction Exploration**

1. **HTTP GET Request**:
   * URL: <https://api.restful-api.dev/objects/7>
   * Response Status Code: 200 (OK)
   * Response Body: JSON data of the object with ID 7.

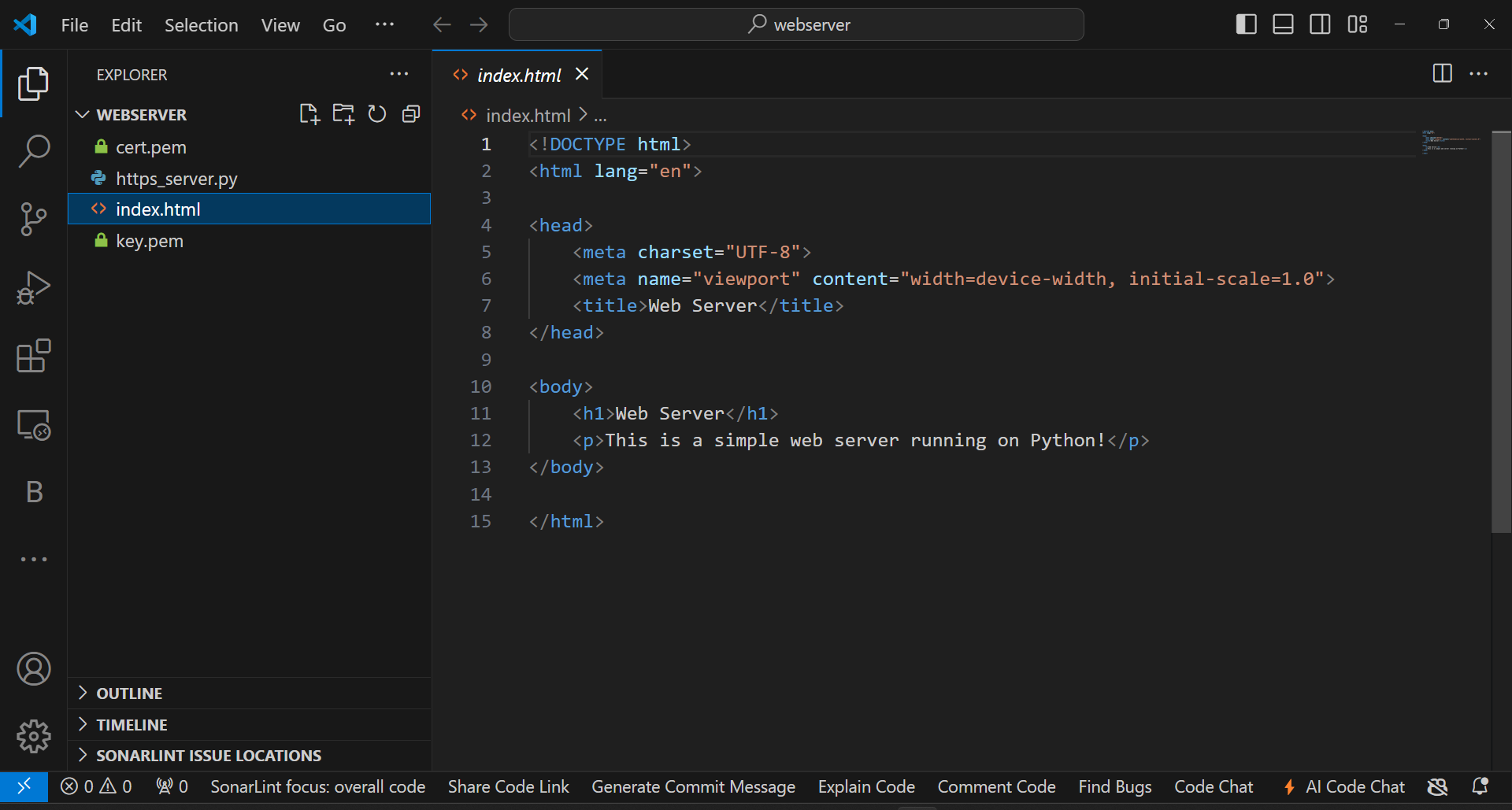


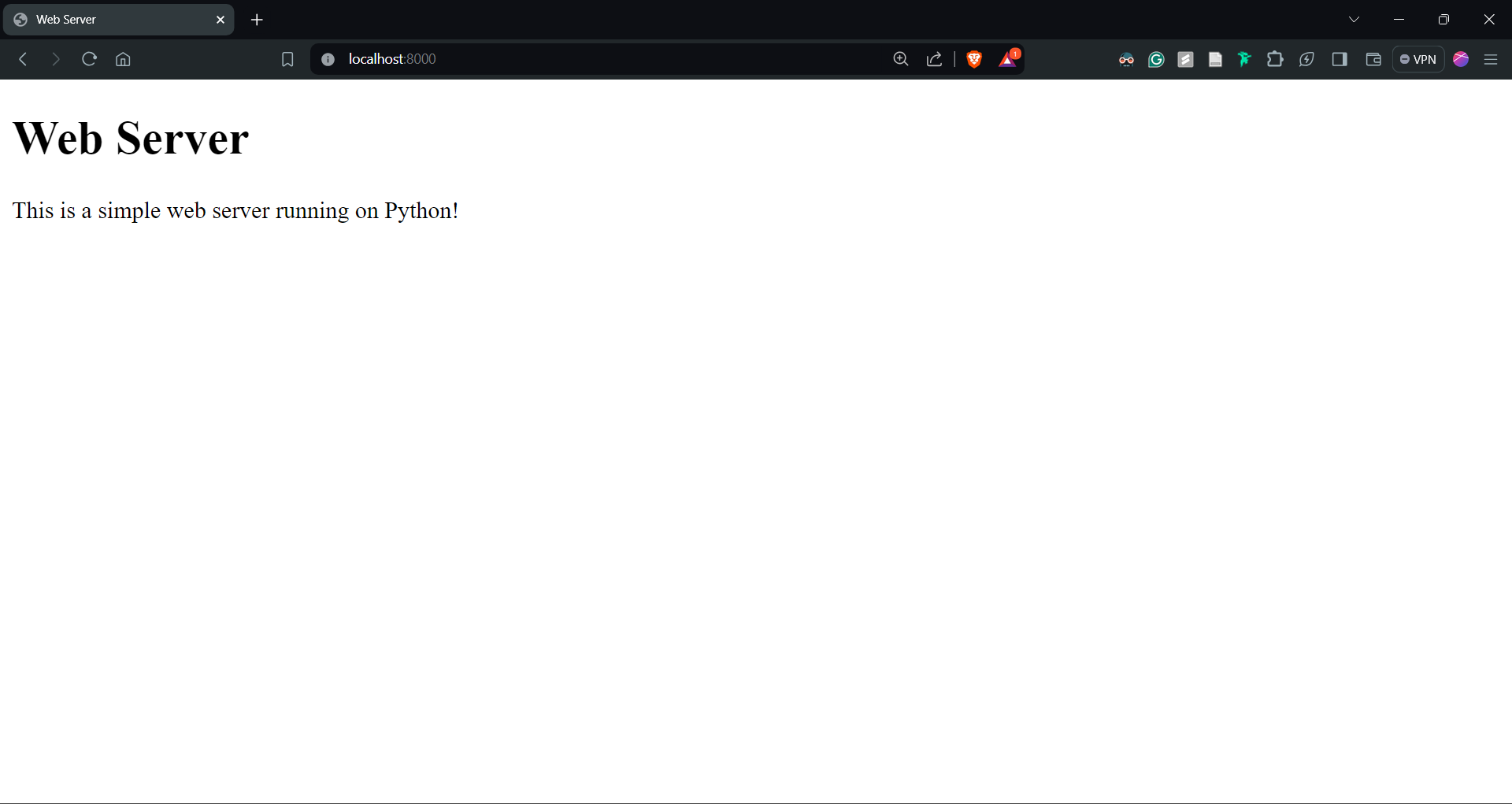
1. **HTTP POST Request**:
   * URL: <https://api.restful-api.dev/objects>
   * Request Data: {"name": "Apple MacBook Pro 16", "data": {"year": 2019,"price": 1849.99,"CPU model": "Intel Core i9","Hard disk size": "1 TB"}}
   * Response Status Code: 200
   * Response Body: JSON data of the created object.



## **Web Server Configuration**

I used Python's built-in HTTP server for simplicity. I then placed the HTML files and other static content inside the webserver directory. Then started the Python HTTP server with the command ` python -m http.server 8000`. The content was served at `http://localhost:8000`.





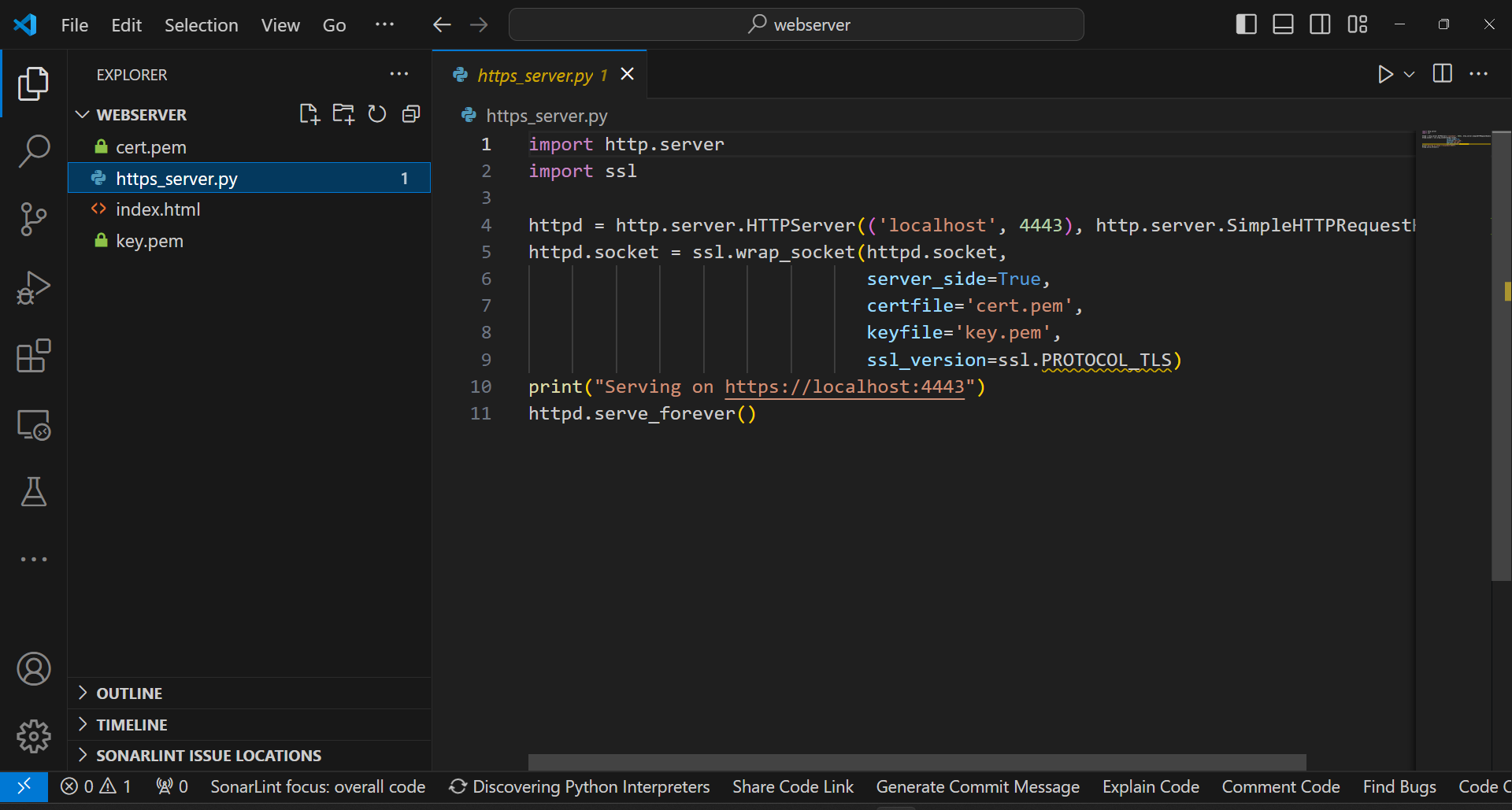
## **HTTPS Implementation**

For serving content over HTTPS, I generated SSL certificates and configured the server to use them. I used a simple Python HTTPS server for this purpose.

1. **Generating a self-signed SSL certificate**:
   1. Ran the following commands to generate a key and certificate:

`openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365 –nodes`

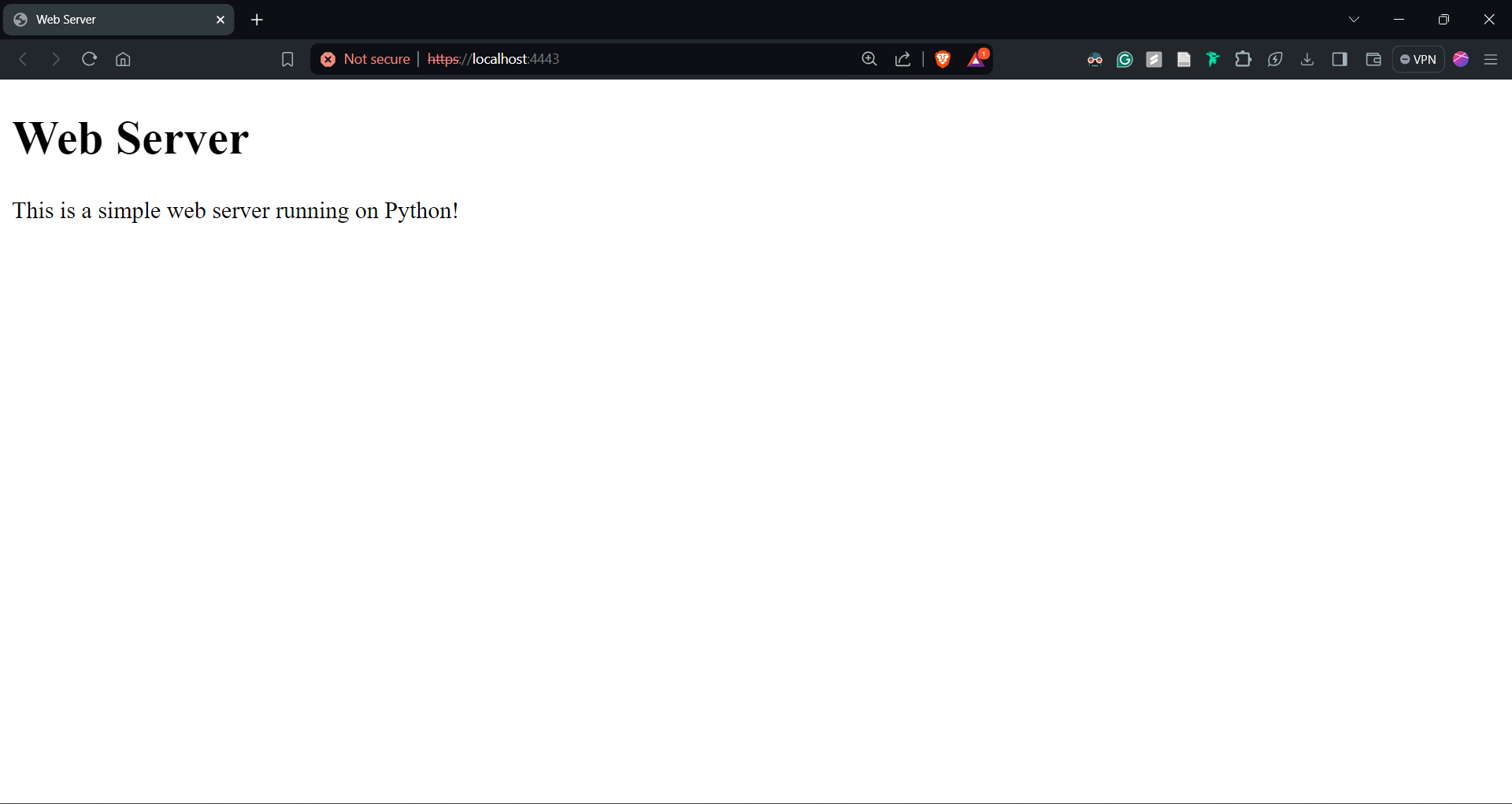
1. **Creating a Python HTTPS server script**:
   * Created a new file named https\_server.py in web content directory
   * `cert.pem` and `key.pem` are in the same directory as https\_server.py.



1. **Start the HTTPS server**:
   * Run the following command:

`python https\_server.py`

* + The content will now be served at <https://localhost:4443>.



## **Security Differences Between HTTP and HTTPS**

|  |  |  |
| --- | --- | --- |
|  | **HTTP** | **HTTPS** |
| **Encryption** | Data transmitted over HTTP is in plain text, which can be easily intercepted and read by malicious actors. | Data transmitted over HTTPS is encrypted using SSL/TLS, preventing eavesdropping and ensuring data privacy. |
| **Data Integrity** | Data integrity is not guaranteed. Data can be altered in transit without detection. | Ensures data integrity by using cryptographic hash functions, making it detectable if data is tampered with during transmission. |
| **Authentication** | Does not provide authentication of the server. Users cannot be sure they are communicating with the intended server. | Uses SSL/TLS certificates to authenticate the server, ensuring users are connected to the correct server. |