****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **2/23/2024** | **Karol Guerra** |  |

## Client



## Instructions

Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* Respond to the steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Karol Guerra

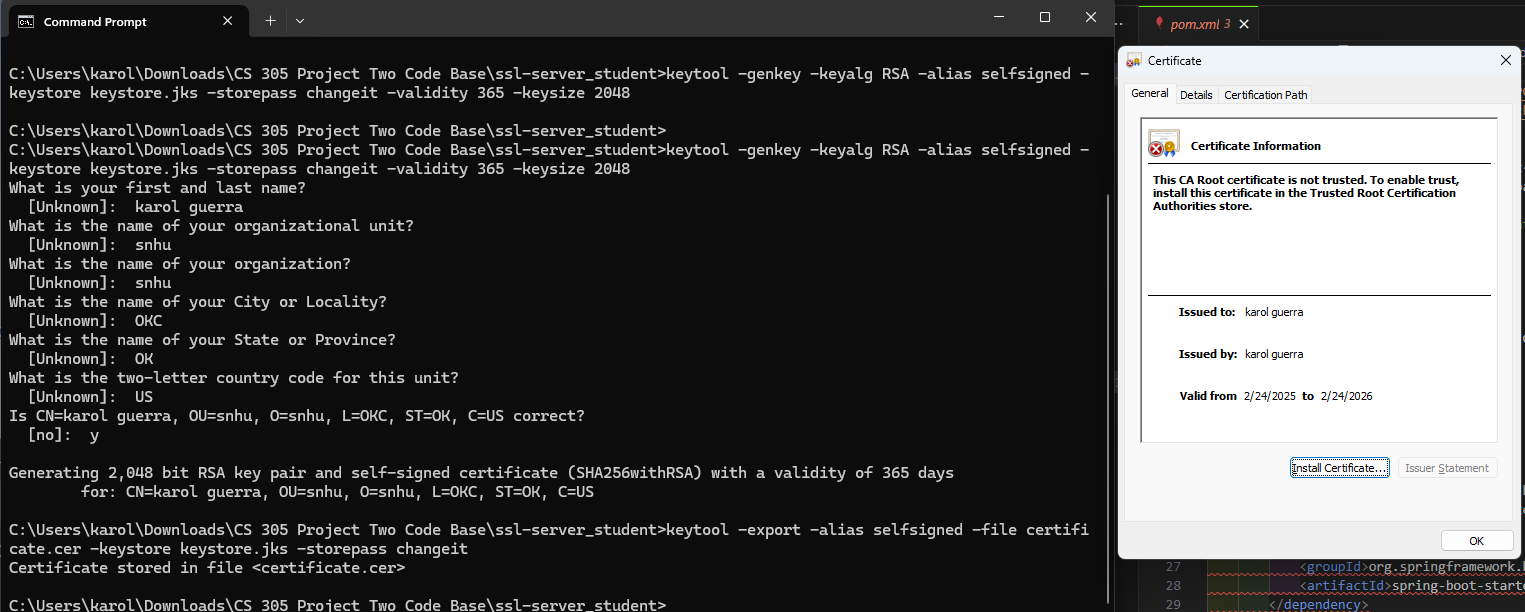
## Algorithm Cipher

Hash functions play a crucial role in maintaining data integrity. A hash function converts input data into a fixed-length output (hash value), which is unique for each input.

For strong hashing, I recommend using SHA-256 (Secure Hash Algorithm 256-bit). SHA-256 is part of the SHA-2 family, developed by NIST, and is widely used in applications requiring high security, such as digital signatures, password hashing, and data integrity verification. The 256-bit output ensures resistance against collision attacks, where two different inputs generate the same hash.

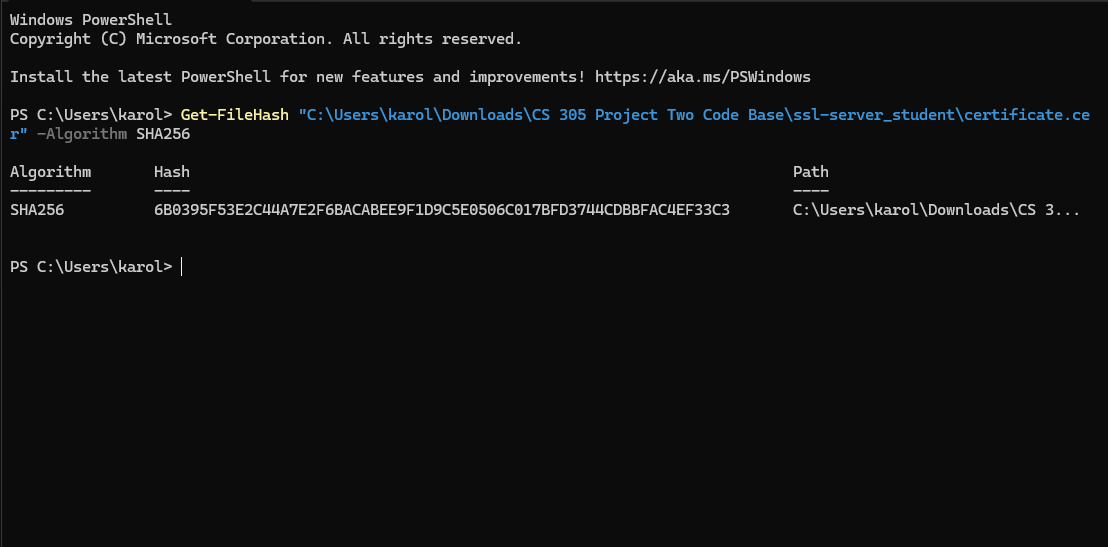
## Certificate Generation

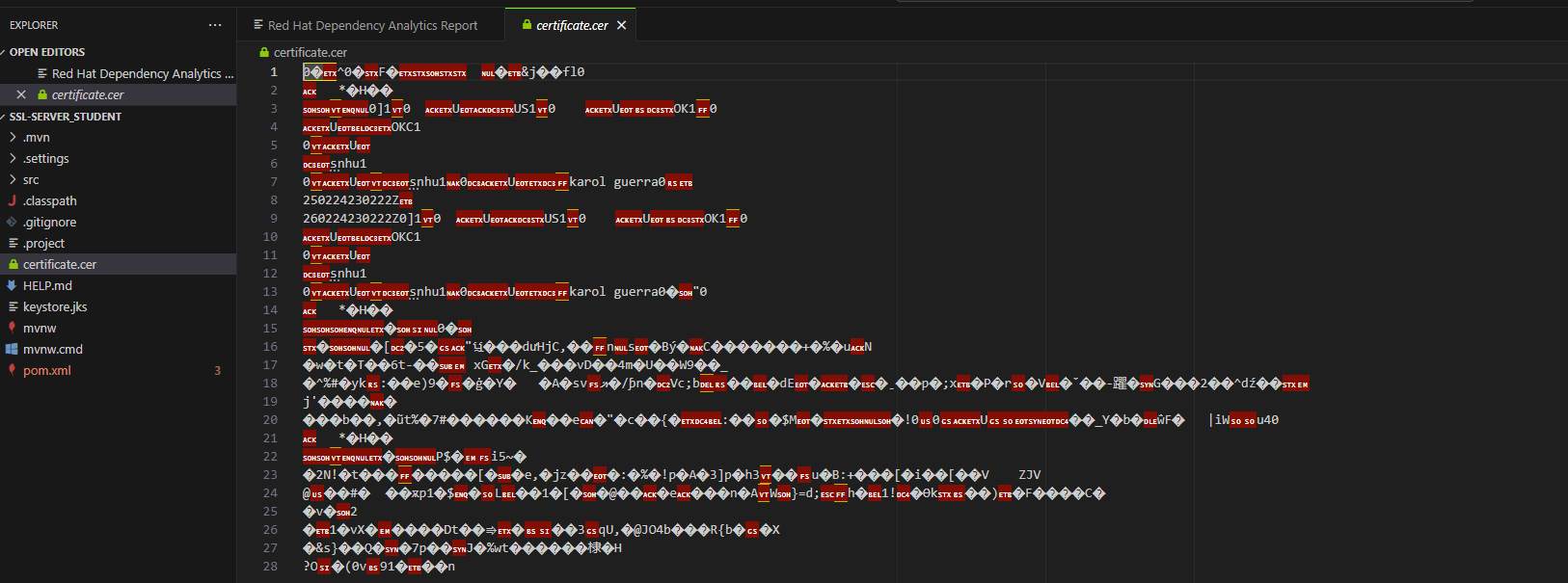
Insert a screenshot below of the CER file.

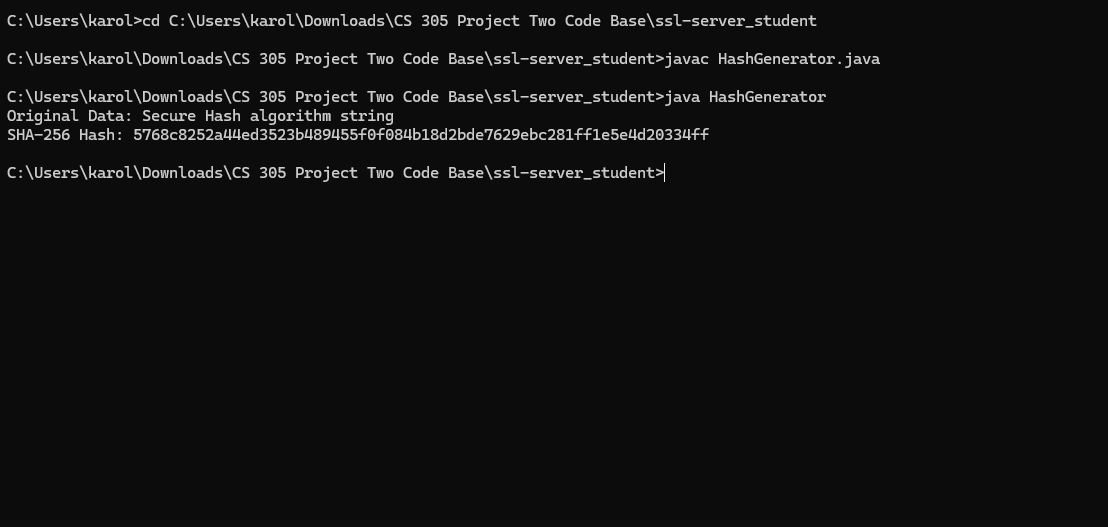
[Insert screenshots here.]  


## Deploy Cipher

Insert a screenshot below of the checksum verification.

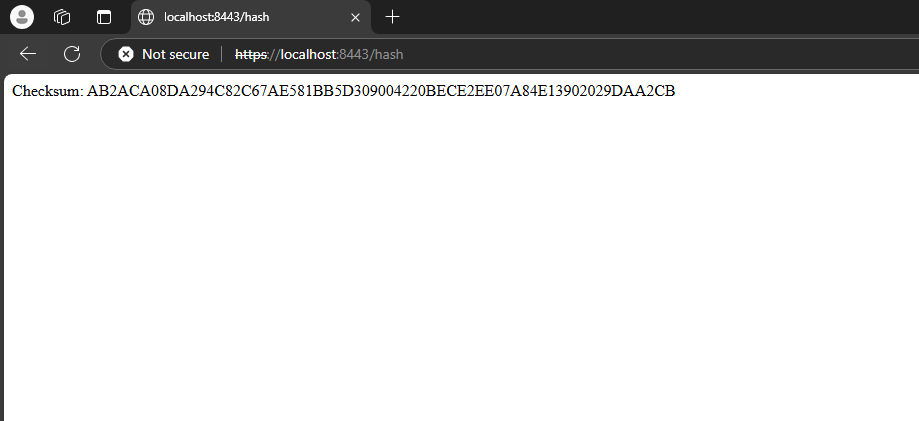
[Insert screenshots here.]  






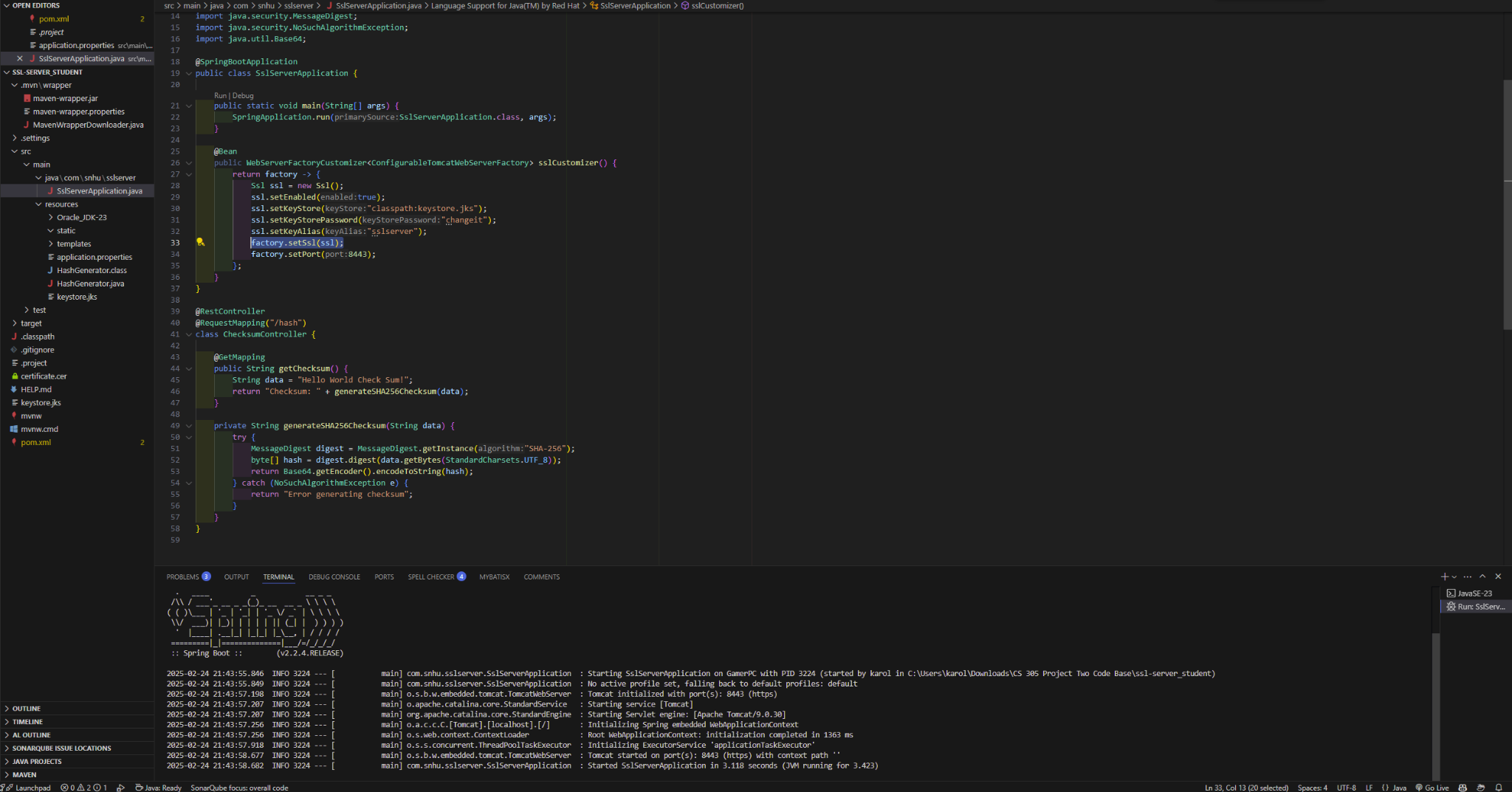
## Secure Communications

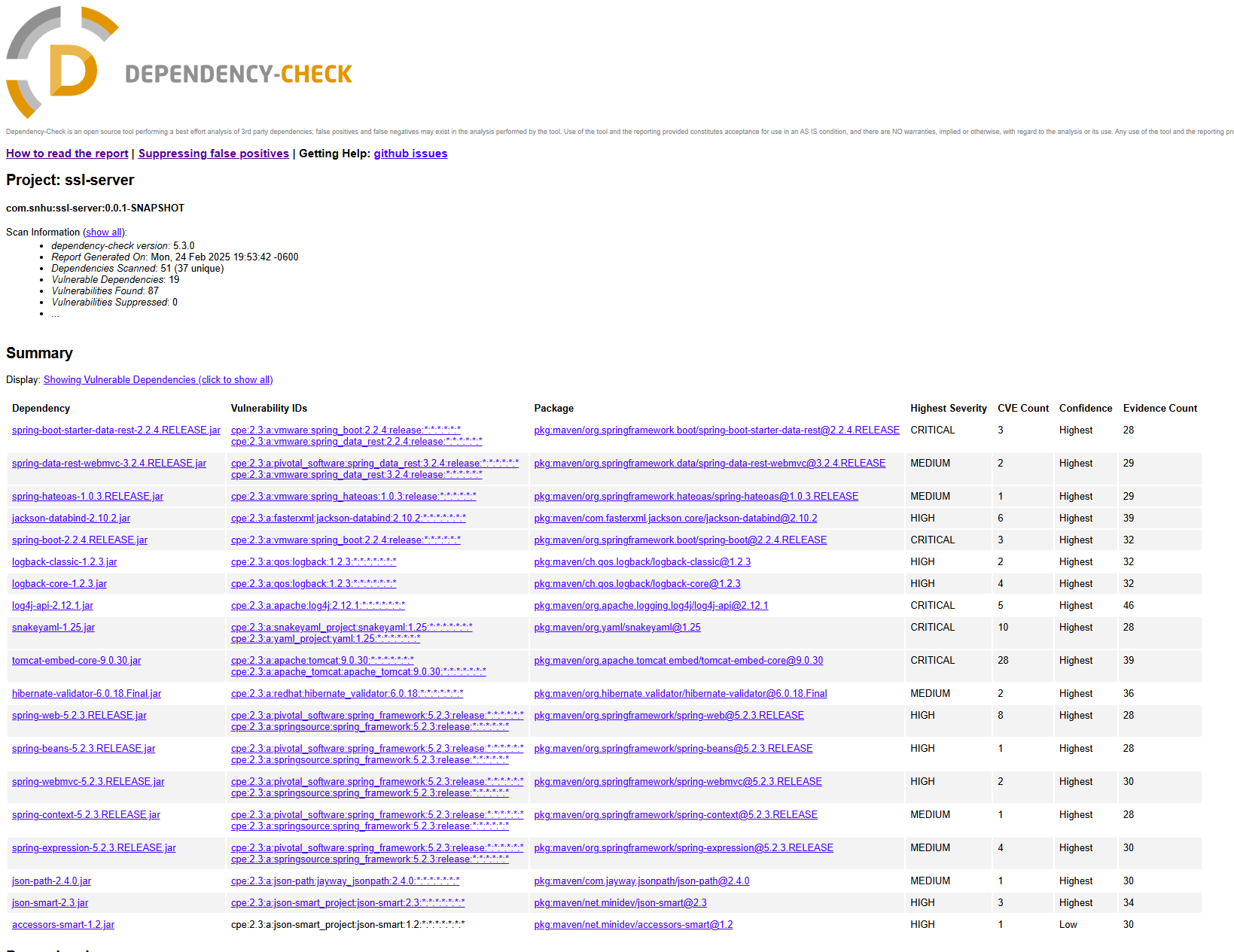
Insert a screenshot below of the web browser that shows a secure webpage.

[Insert screenshots here.]  


## Secondary Testing

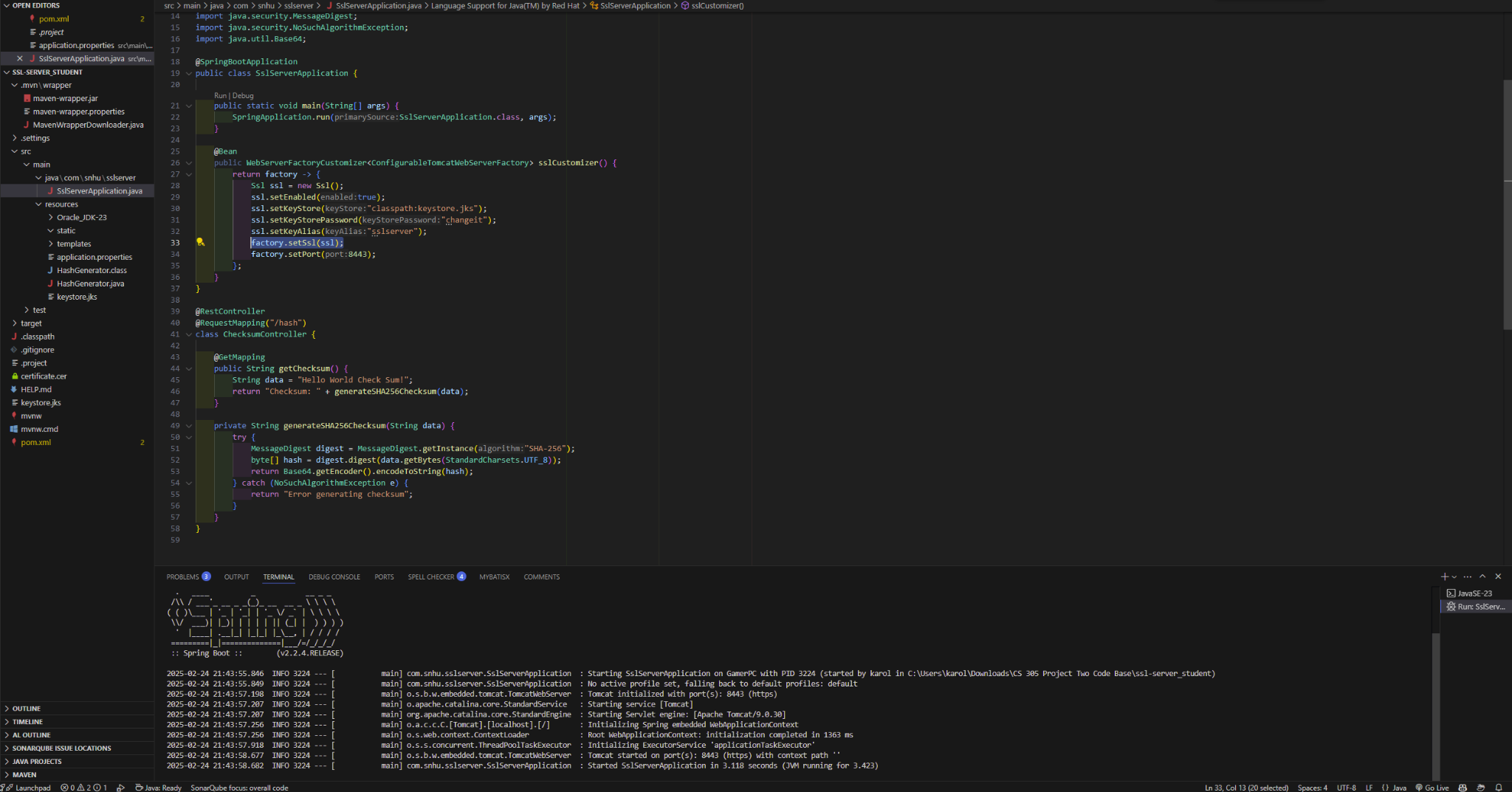
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

Insert a screenshot below of the refactored code executed without errors.



## Summary

A comprehensive vulnerability assessment was conducted to strategically identify and address critical, high risk, and low risk security concerns. This process shows where enhancements can be made across key areas, including encryption protocols, authentication mechanisms, secure communication channels, and overall security compliance posture.

This project focused on enhancing the security of the codebase by identifying and mitigating potential vulnerabilities through secure coding practices and refactoring. Additionally cryptographic practices were implemented using SSL, certificate generation, and utilizing port 8443 HTTPS. This project also implemented AES-256 encryption for data security and SHA-256 hashing for data integrity verification.

## Industry Standard Best Practices

To strengthen the security of any software application, it is important to check and scan for NIST and OWASP standards to ensure that the application is suitable for meeting safe guidelines. This project and code base emphasized areas of security such as encryption, securing data transmission, and reinforcing the overall integrity of the system. We implemented AES-256 encryption alongside SHA-256 hashing to protect sensitive information and enforced HTTPS with SSL to safeguard data in transit.