****

# Practices for Secure Software Report

Table of Contents

[Document Revision History 3](#_Toc102040754)

[Client 3](#_Toc102040755)

[Instructions 3](#_Toc102040756)

[Developer 4](#_Toc102040757)

[1. Algorithm Cipher 4](#_Toc102040758)

[2. Certificate Generation 4](#_Toc102040759)

[3. Deploy Cipher 4](#_Toc102040760)

[4. Secure Communications 4](#_Toc102040761)

[5. Secondary Testing 4](#_Toc102040762)

[6. Functional Testing 4](#_Toc102040763)

[7. Summary 4](#_Toc102040764)

[8. Industry Standard Best Practices 4](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **2/16/2023** | **Nicholas Glover** | **Completed the Secure Software Report** |

## 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

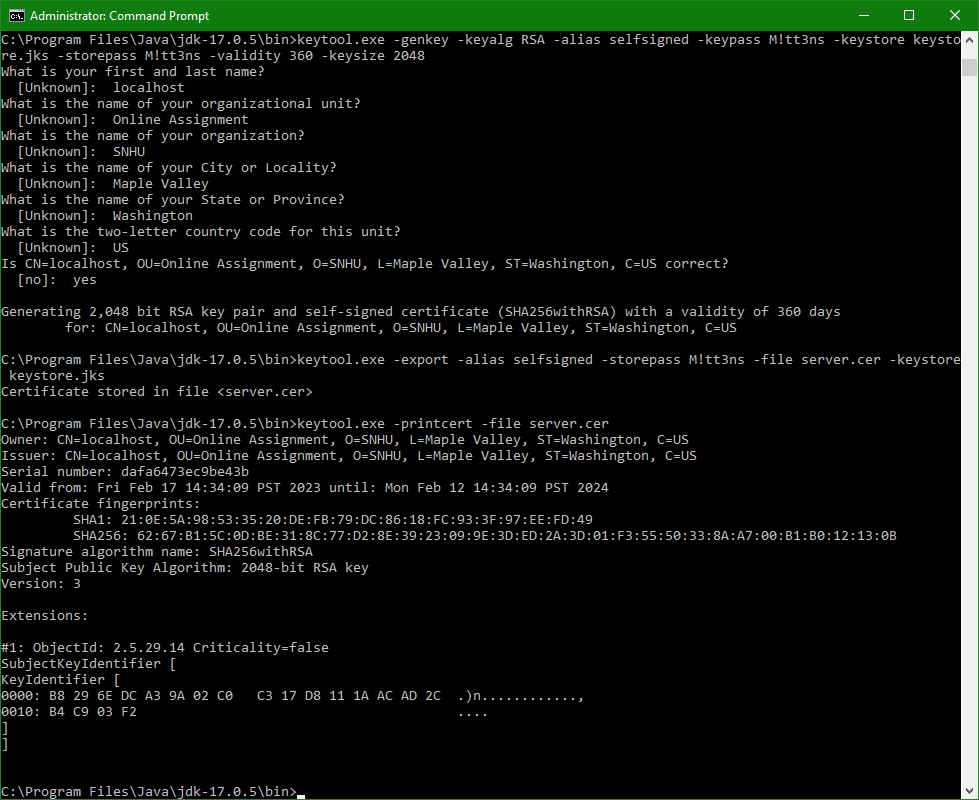
Nicholas Glover

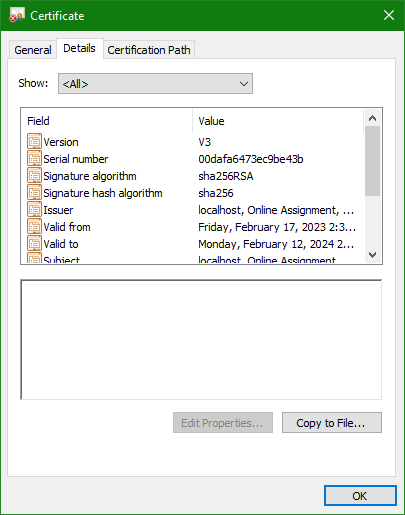
## Algorithm Cipher

For Artemis Financial I would recommend using the AES-256 encryption algorithm cipher. It is the most robust and modern encryption standard that currently exists and is also used by the government. It would allow the company to fulfill all of its secure communication needs with the data that they use from their clients. It is a symmetric block cipher that can both encrypt and decrypt data. It uses cryptographic keys that are 256 bits long, ensuring a very secure key. Since the cipher is symmetric, it uses the same key to encrypt and decrypt data. For the hash function, I would recommend using SHA3-512. It is the most secure hash function currently in use because it uses a completely different method of hashing than the more popular SHA2 hashing function. The bit levels of both AES-256 and SHA3-512 refer to the number of bits that the systems use. The algorithm cipher uses a 256-bit secret key and the hashing function generates a 512-bit “fingerprint” of the data it’s applied to. The use of random numbers is imperative to the security of the data. They ensure that the systems can’t be broken easily because there is no underlying pattern to them. The AES-256 algorithm is a symmetric block cipher as opposed to an asymmetric cipher which means that it uses the same key to encrypt and decrypt data. An asymmetric cipher would use different keys to complete the encryption and decryption operations. Encryption algorithms have been in use since computing became more prevalent in the 70’s. Originally a standard called DES was used until AES came along and was adopted by the government in 2001. AES is a much more robust system and DES was actually cracked not too long after AES was officially adopted.

## Certificate Generation

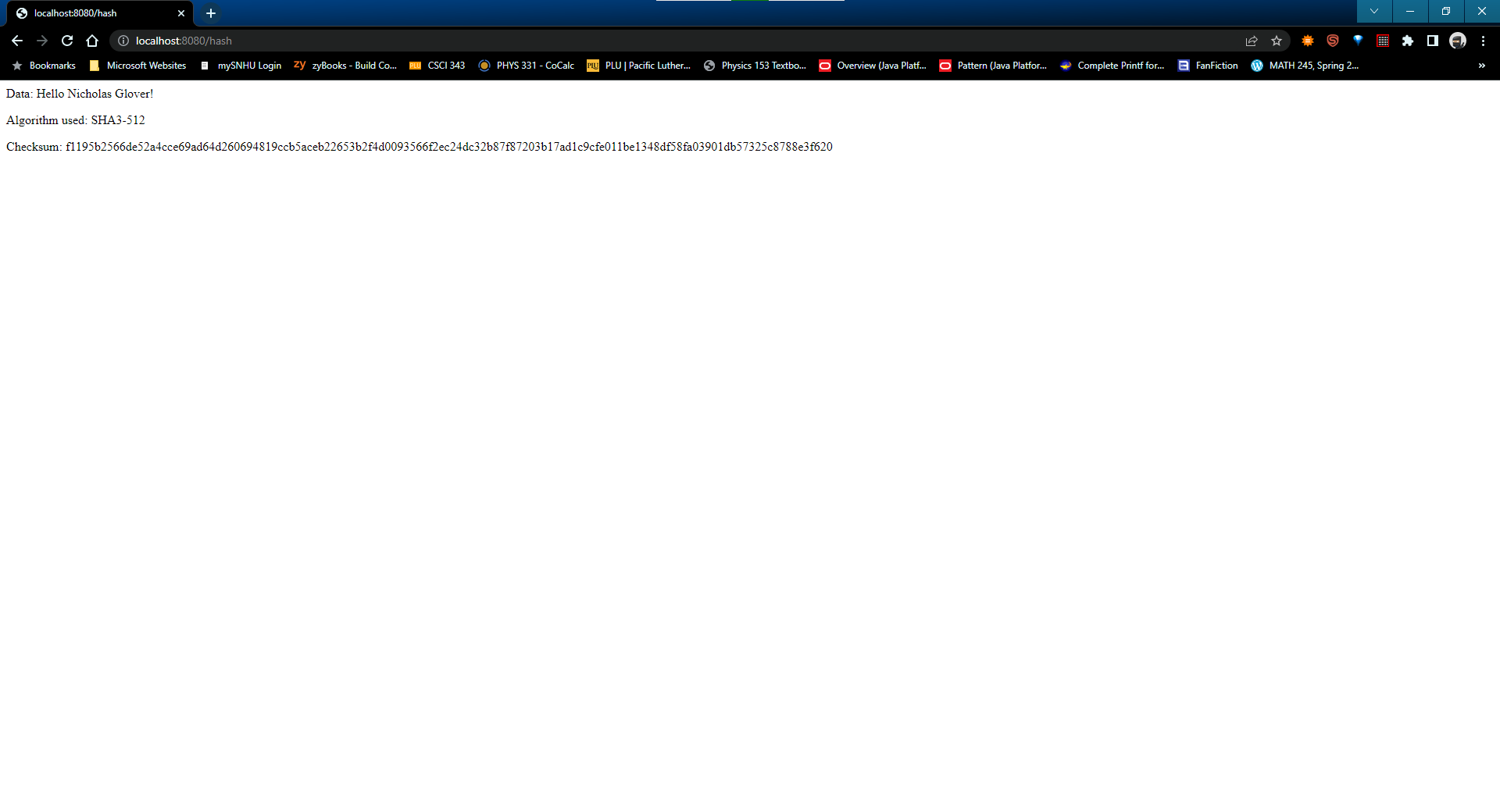
Insert a screenshot below of the CER file.





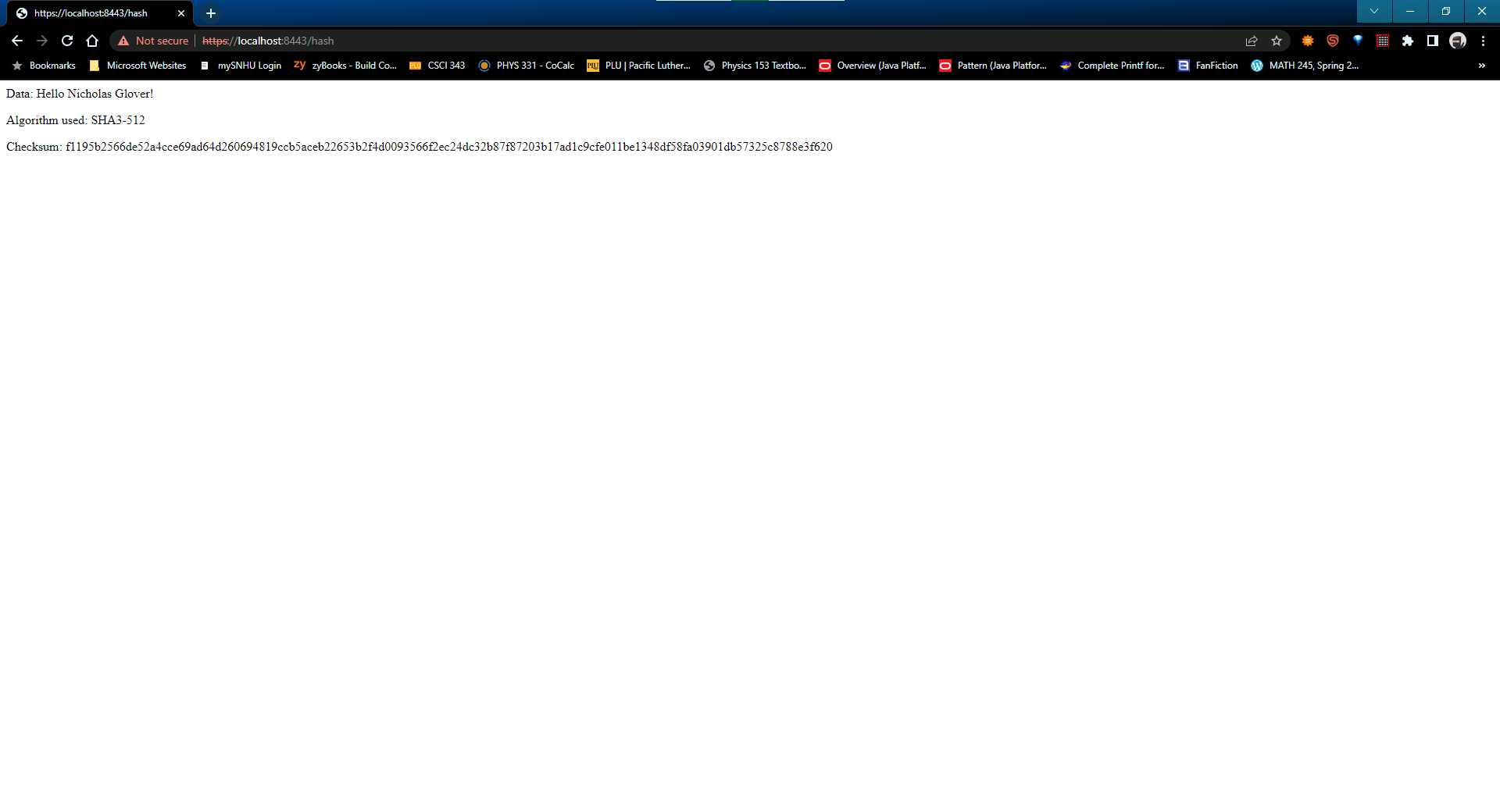
## Deploy Cipher

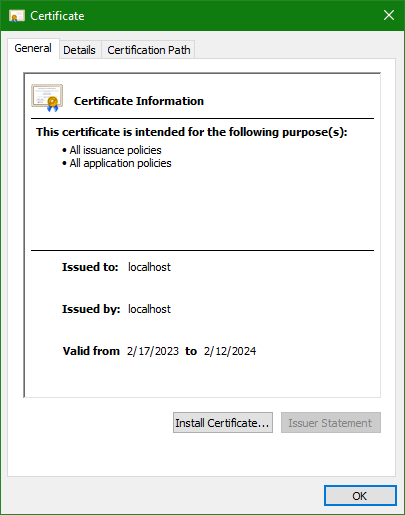
Insert a screenshot below of the checksum verification.



## Secure Communications

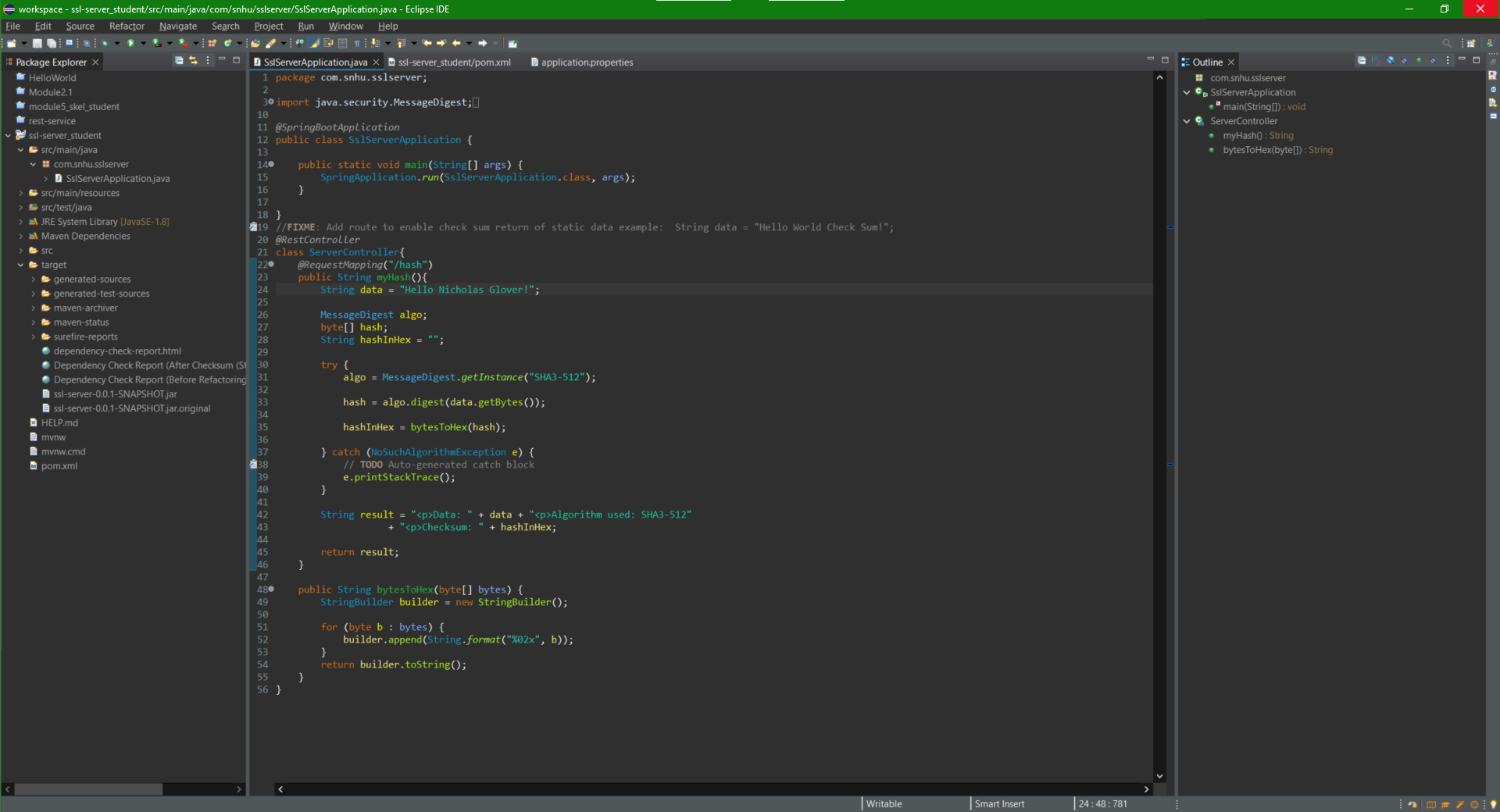
Insert a screenshot below of the web browser that shows a secure webpage.

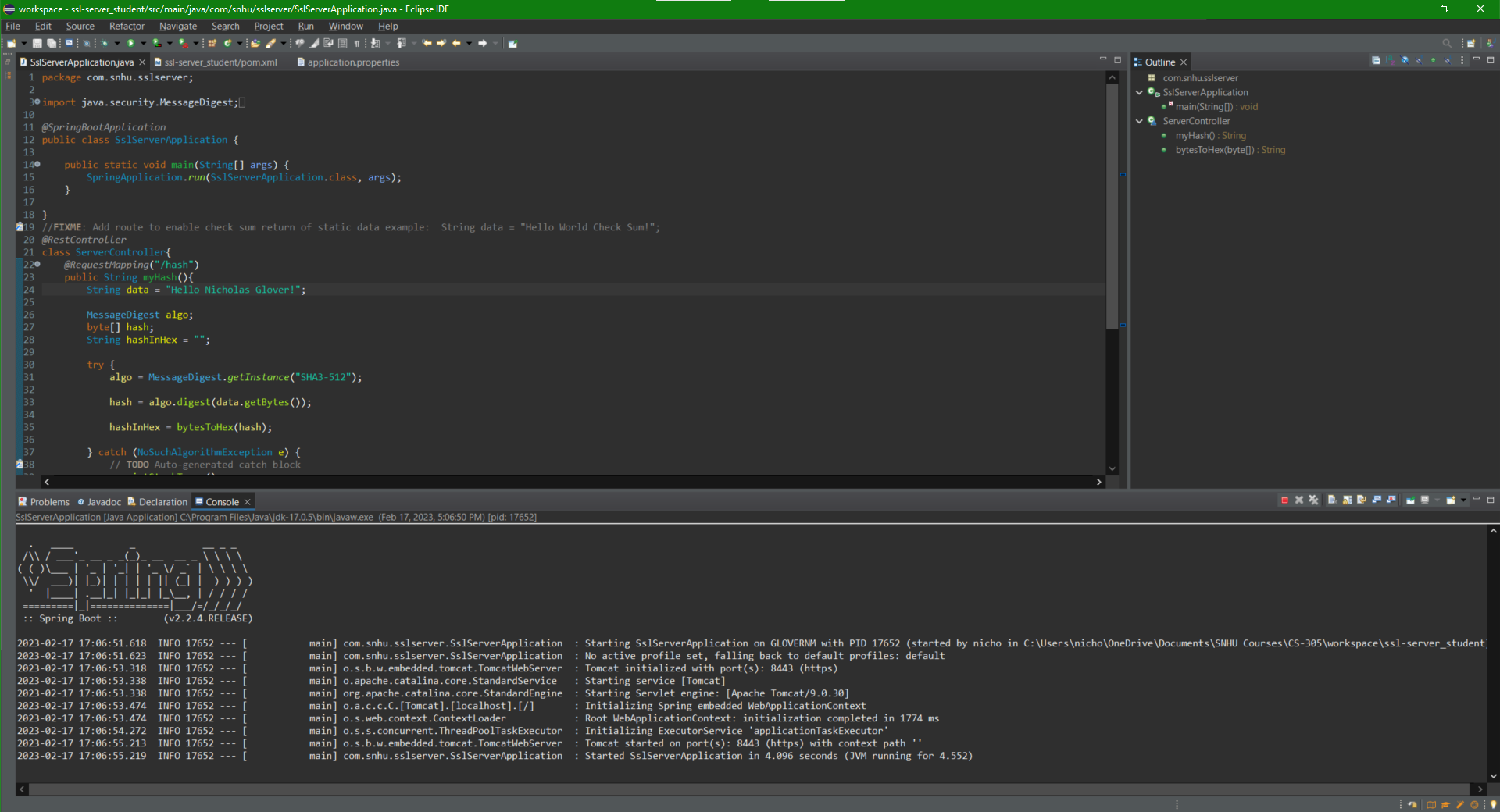




## Secondary Testing

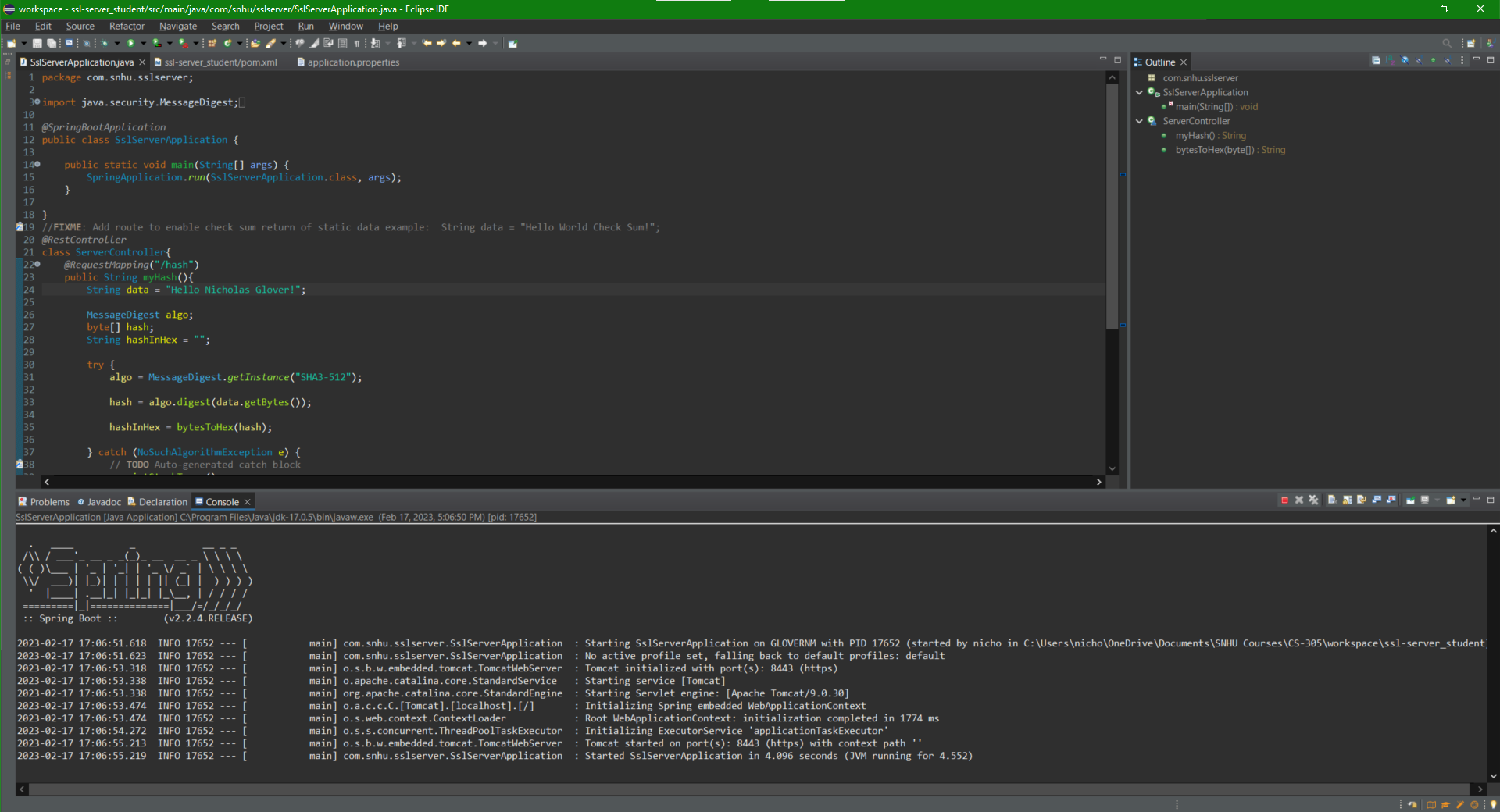
Insert screenshots below of the refactored code executed without errors and the dependency-check report.

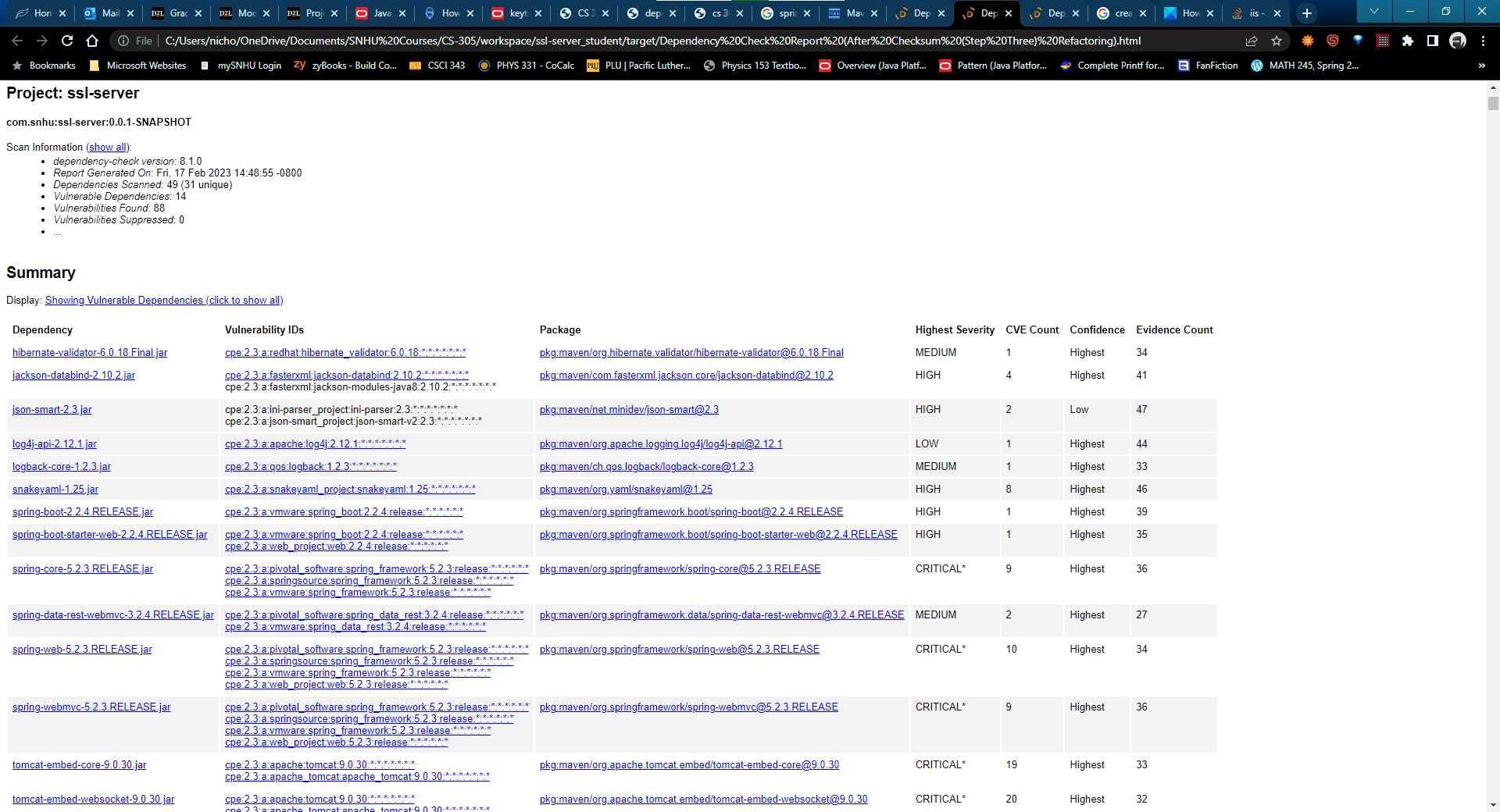




## Functional Testing

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





## Summary

By refactoring the code base, I’ve addressed a couple of the areas described in the Vulnerability Assessment Flow Diagram. The main areas that I’ve addressed are: input validation, cryptography, client/server, code error, and code quality. With input validation, I’ve implemented the use of checksums to ensure that all untrusted data can be verified before it gets used in the system. For cryptography, I’ve implemented the use of SHA3-512 and I’ve recommended the use of the AES-256 encryption algorithm cipher. I’ve included error checking in the code I’ve refactored and I made sure to follow the best programming practices so that the areas of code error and code quality are taken care of. Lastly, I’ve refactored the way the system communicates between client and server so that it uses https as opposed to http for added security.

## Industry Standard Best Practices

I used industry best practices when refactoring the code base by ensuring that all communication is secure. I did that through a couple of different methods. I implemented the use of the SHA3-512 hash function to make sure all data gets a digital fingerprint to check against, and I’ve recommended the use of the AES-256 system to make sure that data that gets communicated also remains encrypted. The industry best practices are paramount to Artemis Financial’s success. Given that it's a financial company, there are several government regulations that they need to fulfill. The main regulations all cover the fact that the client information used by the company needs to remain secure. Other than government regulations, clients need to know that their information is secure in order to trust the company. If the clients don’t trust the company to keep their data safe, the company won’t get very many more clients.