****

# 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** | **[Date]** | **[Your Name]** |  |

## 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

Dante Nardulli

## Algorithm Cipher

For an algorithm cipher I would recommend Artemis Financial uses the Advanced Encryption Standard or AES. What is very efficient about AES is that it has symmetric encryption meaning it uses the same key for encryption and decryption. The secure hash algorithm or SHA-256is what I would recommend to use with AES to generate a hash that can be verified to detect any corrupt data or changes to the data. AES offers a few different bit sizes which all have a place and affect the security and performance of the encryption. We would be using 256 bit due to the size of Artemis Financial’s organization. AES also relies on randomly generated keys which ensure a unique encryption which makes the security much stronger. AES is known for being highly resistant to attacks, fast enough for real time transactions, and is regarded as the standard for encryption industries.

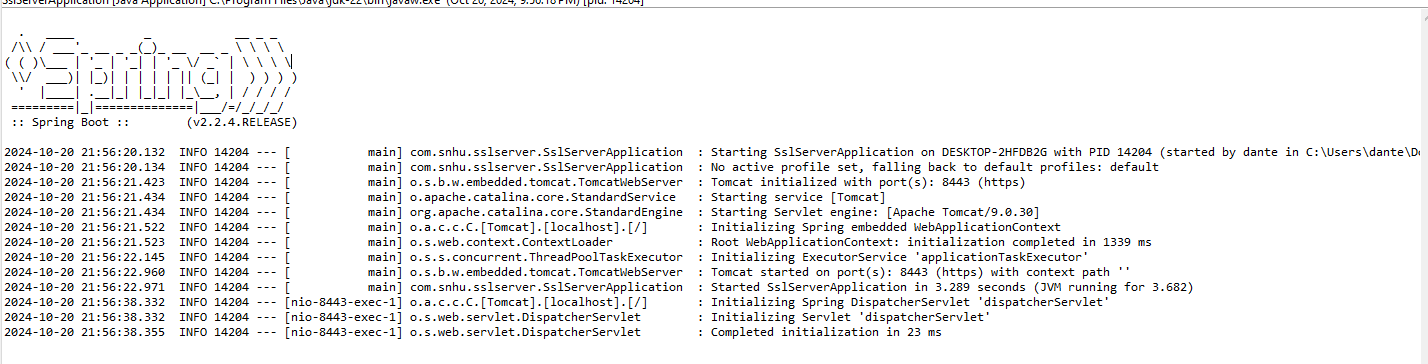
## Certificate Generation

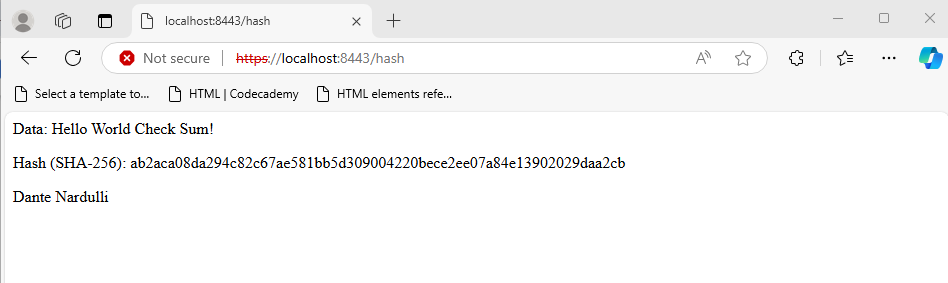
Insert a screenshot below of the CER file.



## Deploy Cipher

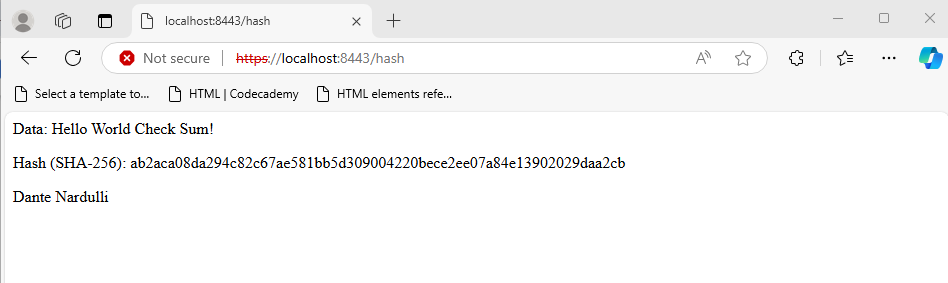
Insert a screenshot below of the checksum verification.





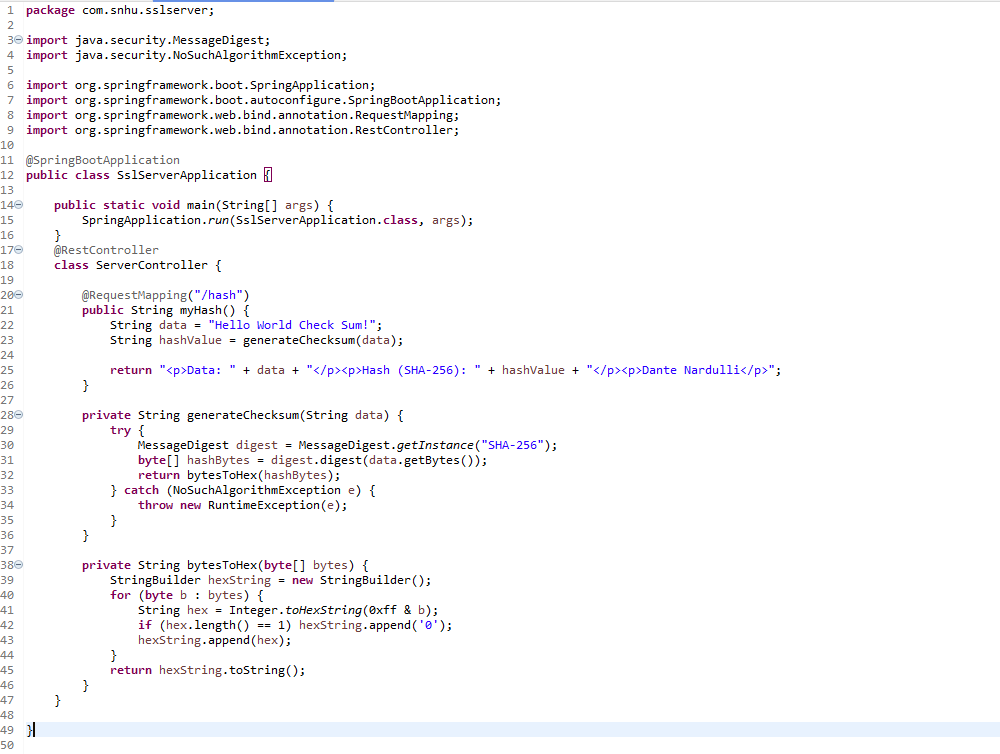
## Secure Communications

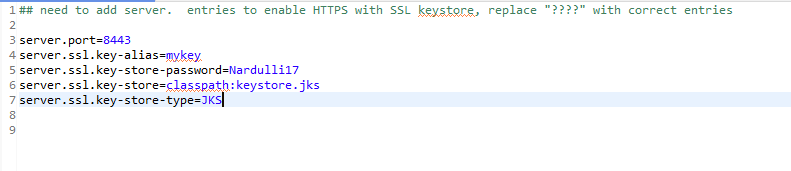
Insert a screenshot below of the web browser that shows a secure webpage.

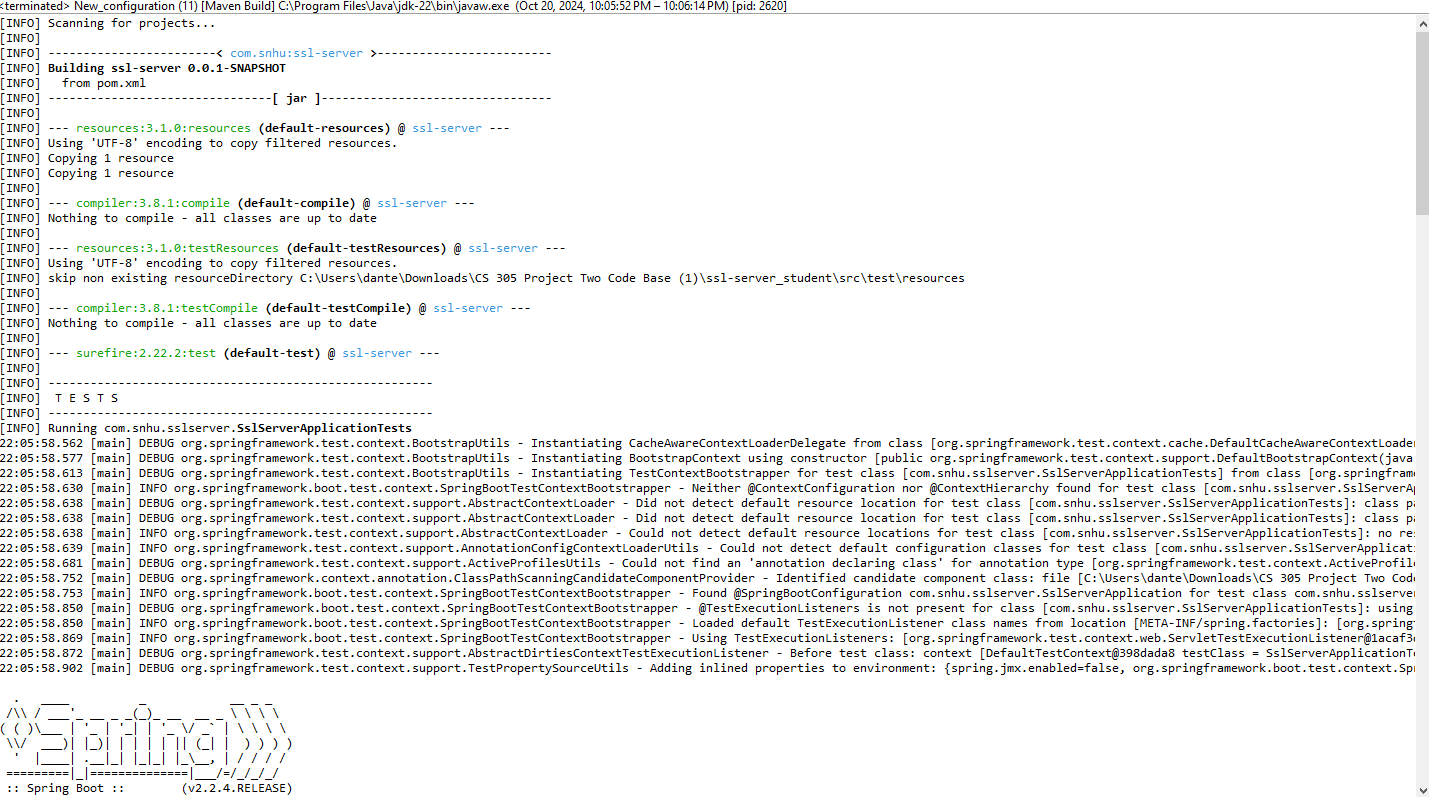


## Secondary Testing

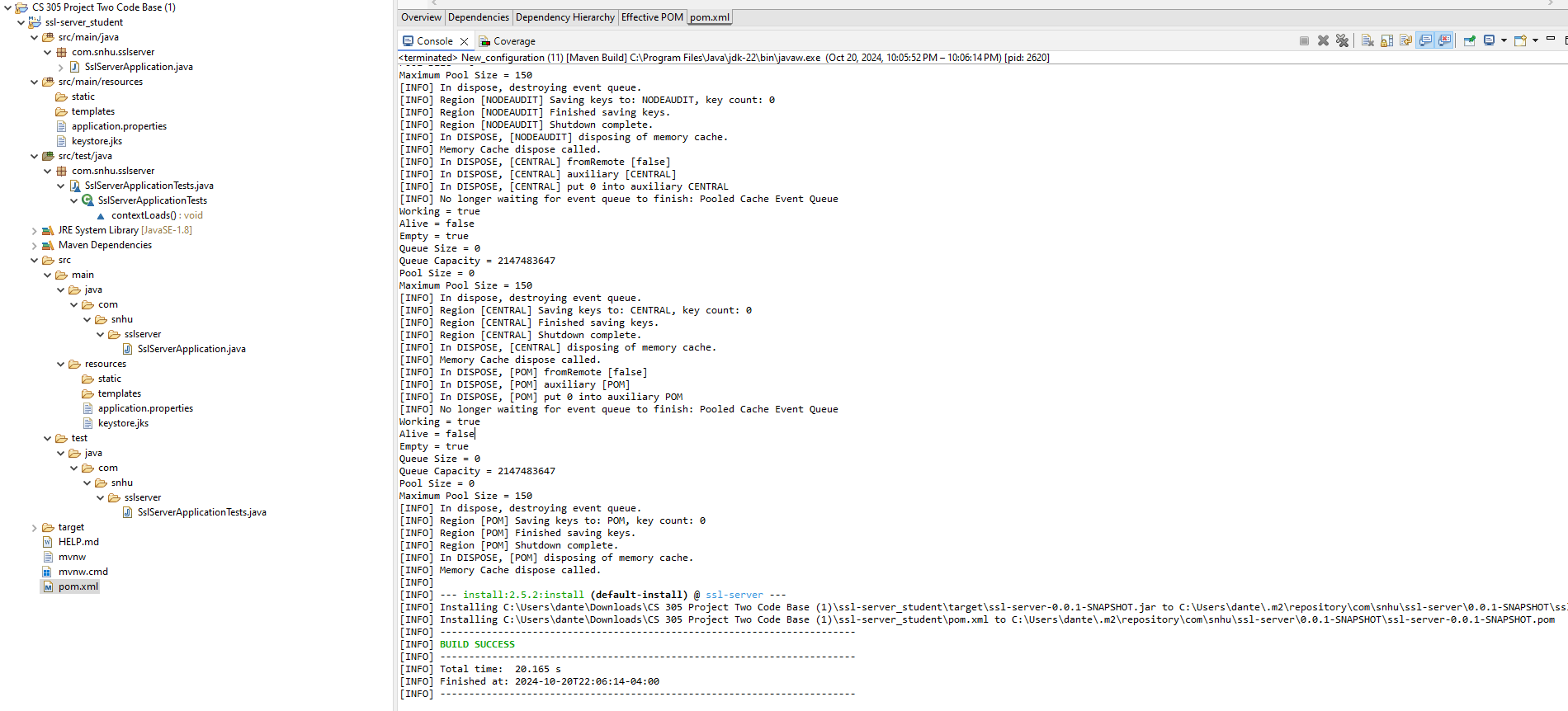
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





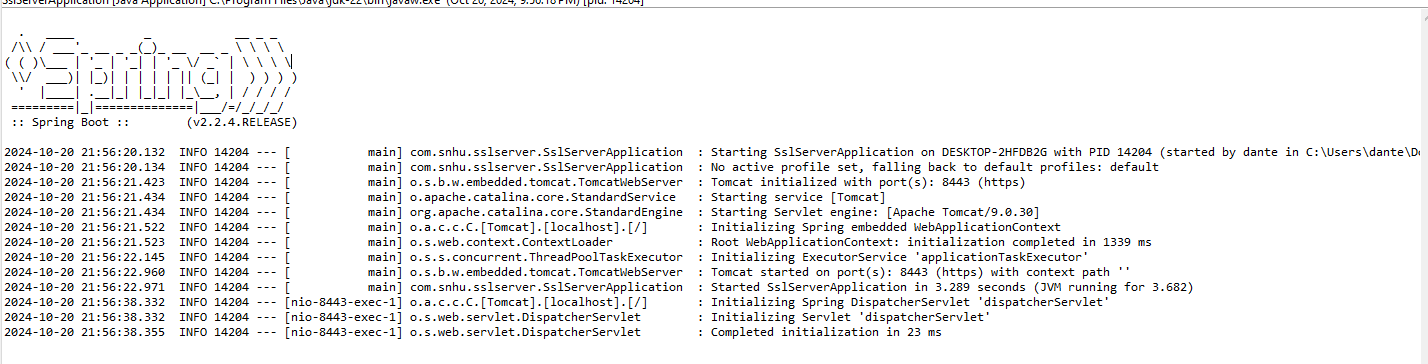


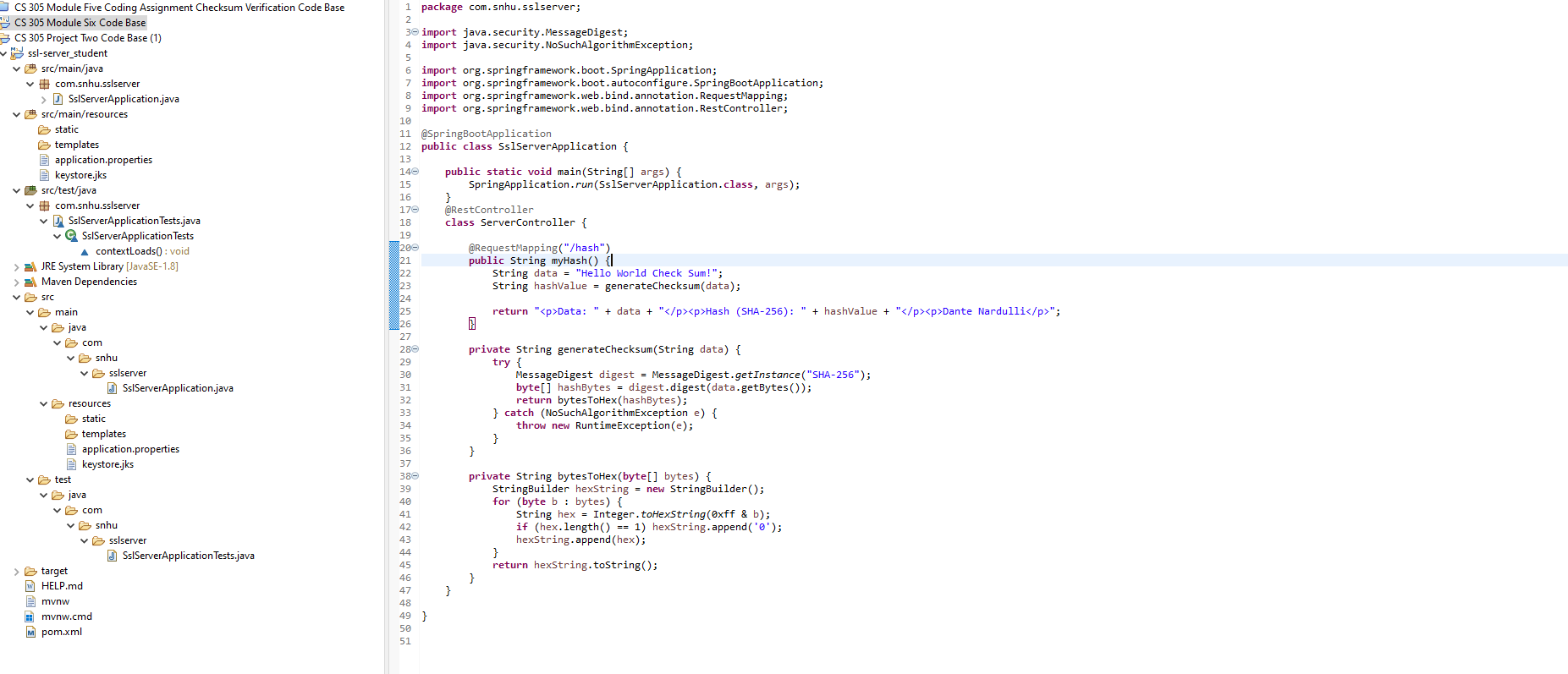




## Functional Testing

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





## Summary

My refactored code improves modularity, error handling and testing processes. I integrated AES encryption, key management, along with checksum implementation. I also configured it to utilize HTTPS with secure communication and a self-signed certificate. This code also mitigates vulnerability assessments for Artemis Financial.

## Industry Standard Best Practices

When it comes to the industry standard and best practices, AES is widely accepted as the standard for security and performance. I have implemented strong key generation using cryptographically secure methods. I also enforced HTTPS ensuring all data in transit is encrypted. Static and functional testing were used as well as maintaining an updated list of dependencies.