****

# 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** | **12/09/2022** | **Jonathan Campos** |  |

## 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

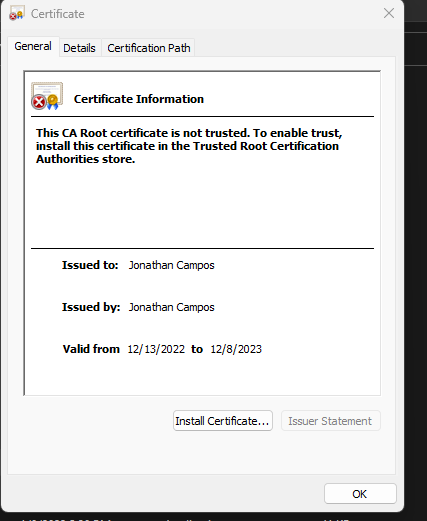
Jonathan Campos

## Algorithm Cipher

After reviewing a few resources, I have determined that the most suitable cipher algorithm that Artemis Financial should implement is AES (Advanced Encryption Standard). It is a symmetric block that the U.S. and Google clouds have deemed safe enough to encrypt classified data. This development is said to have a $250 billion-dollar economic impact on both the private and public sectors. The algorithm uses cryptographic keys to encrypt and decrypt data in blocks of 128 bits. The algorithm is said to be unbreakable, and the only risks are side-channel attacks and poor system configuration. The Federal Financial Institutions Examination Council states that you should encrypt sensitive information such as social security numbers, addresses, and phone numbers. AES encryption would be used to ensure no attack is able to recover any of the user information. The only time I can think of when you should not use the cipher algorithm is when you are encrypting passwords. Passwords should never be stored in databases. I would recommend you use a one way function to hash the data.

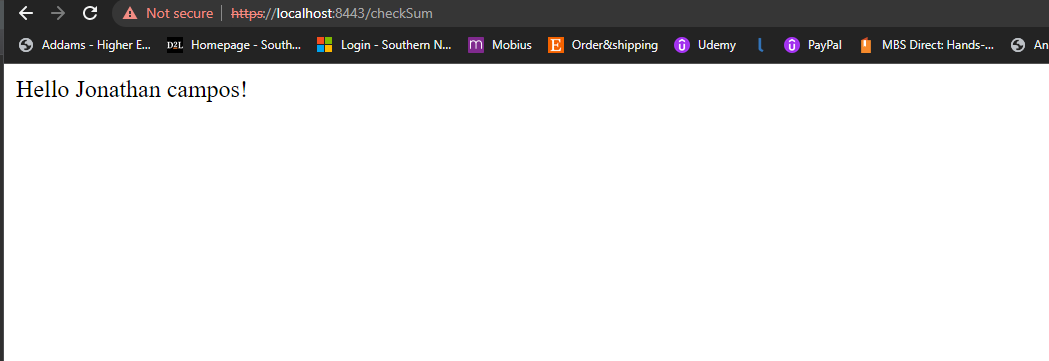
## Certificate Generation

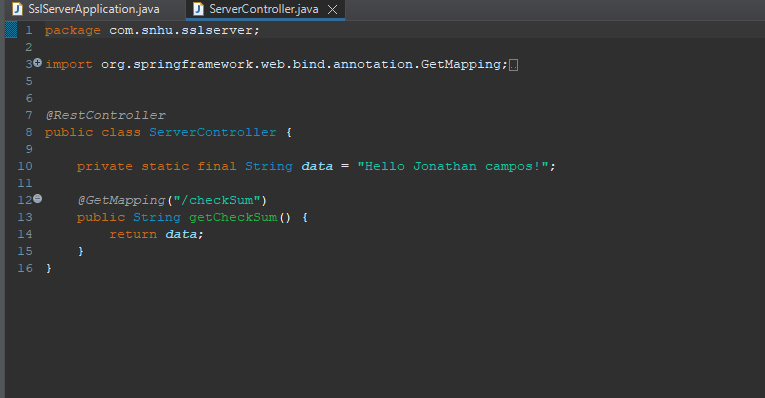
Insert a screenshot below of the CER file.



## Deploy Cipher

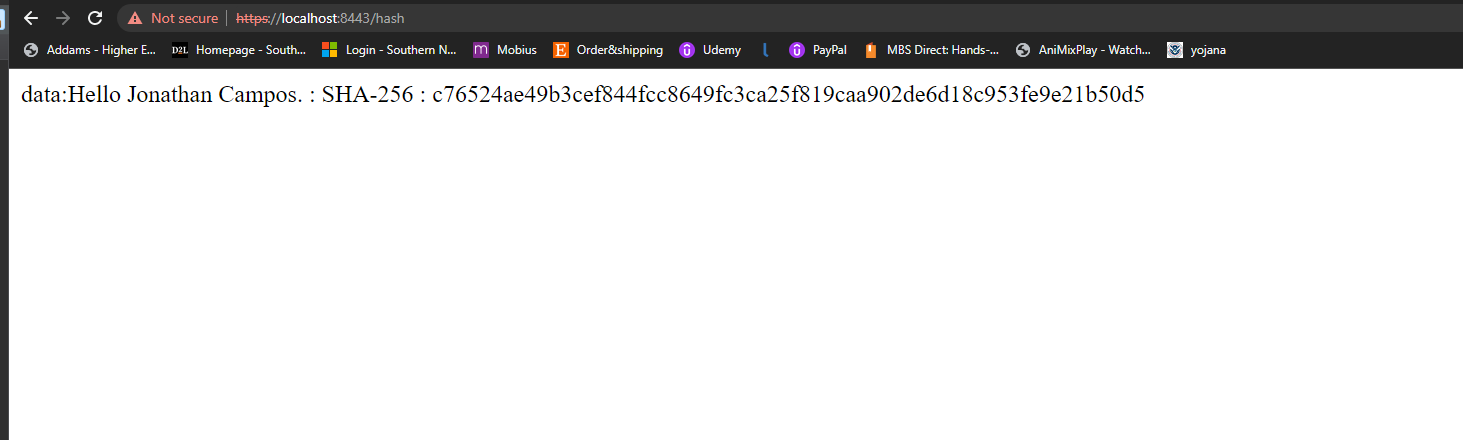
Insert a screenshot below of the checksum verification.





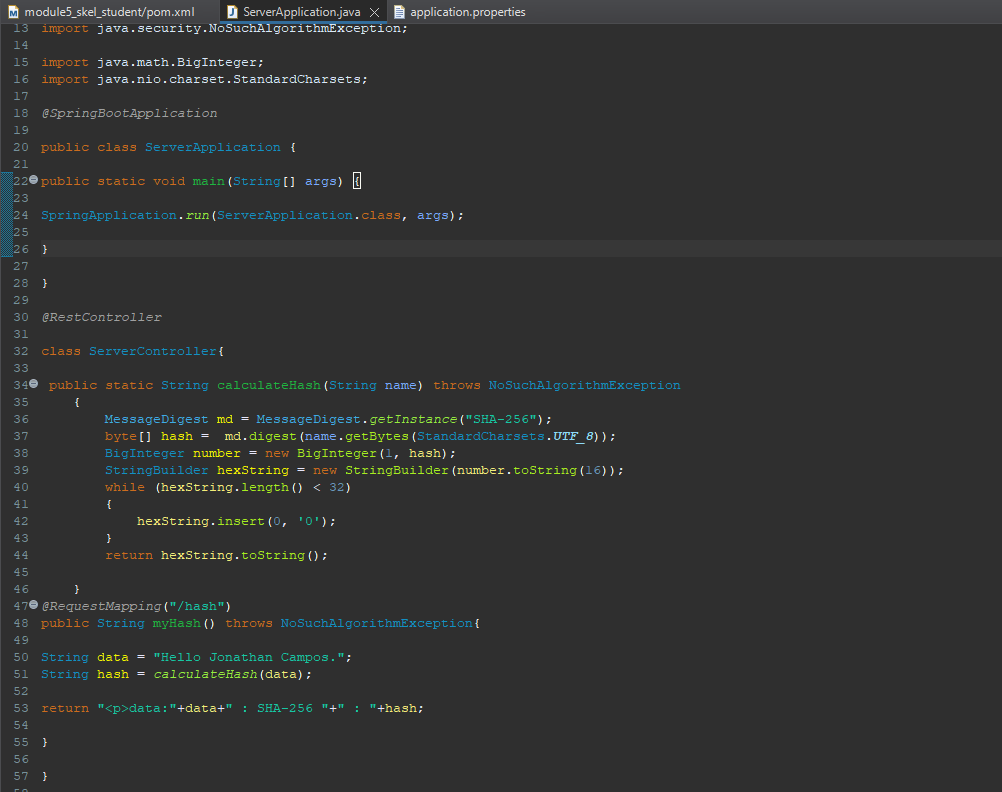
## Secure Communications

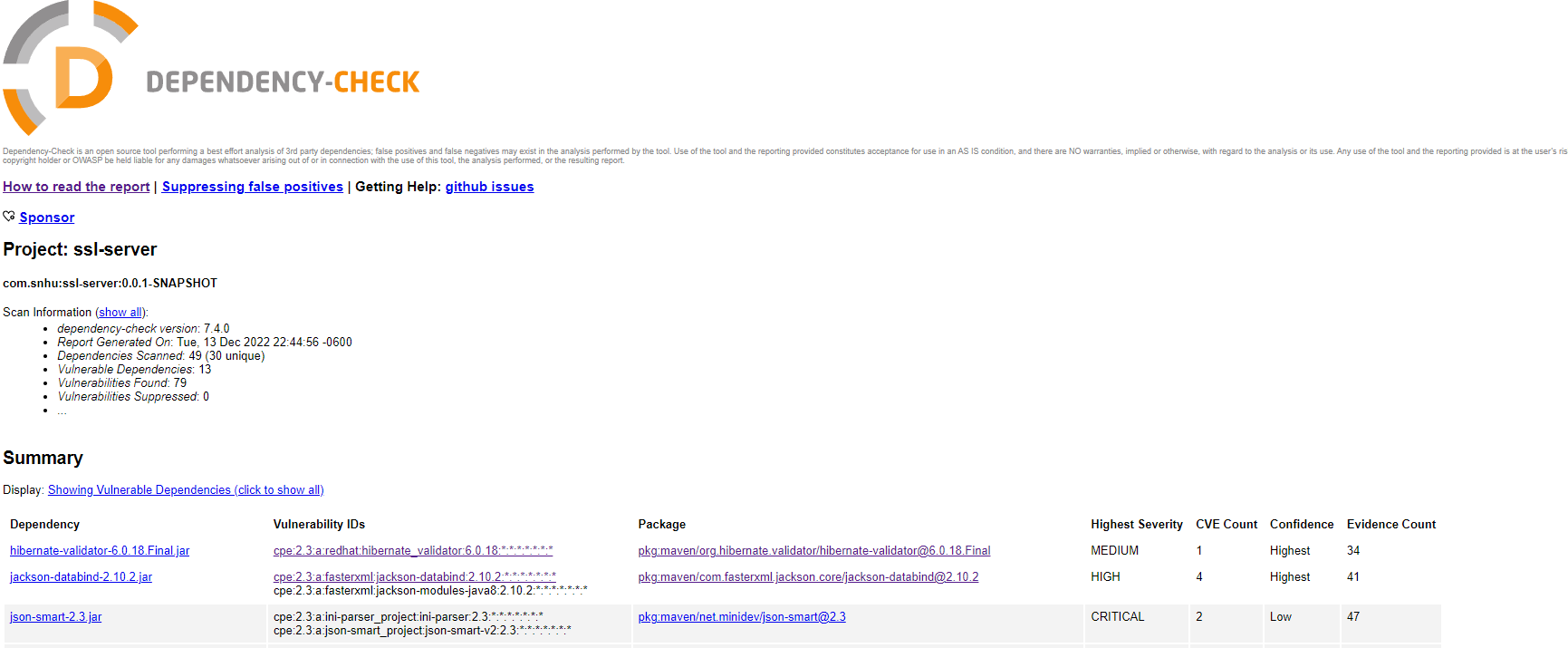
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

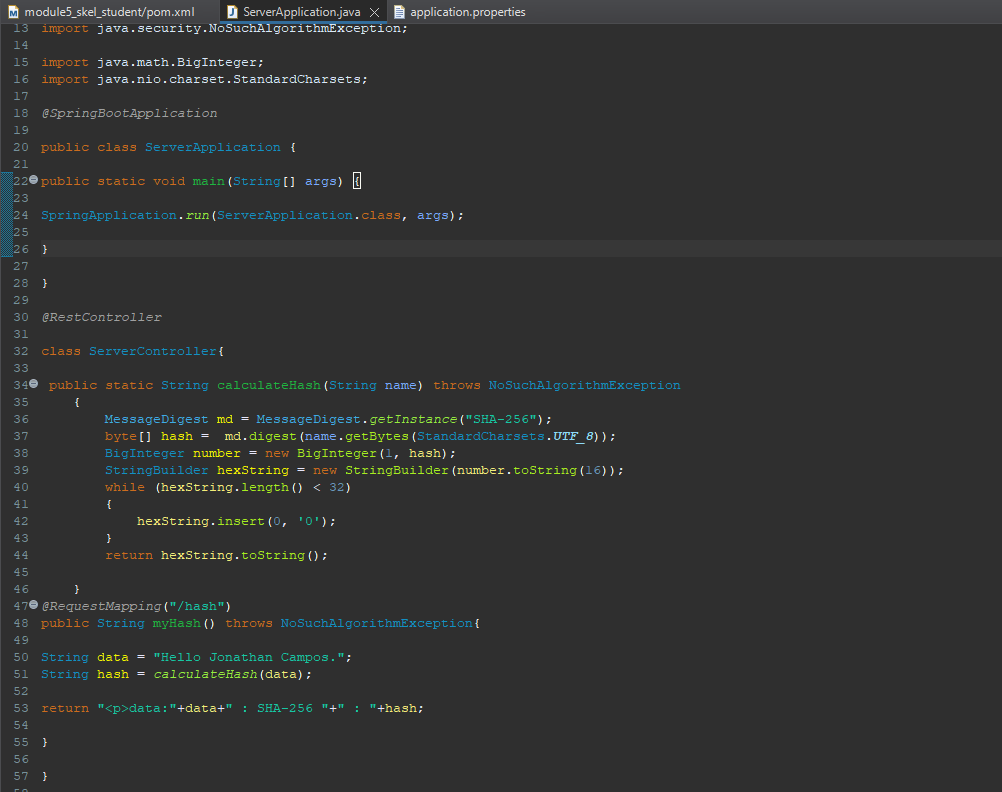
Insert screenshots below of the refactored code executed without errors and the dependency-check report.

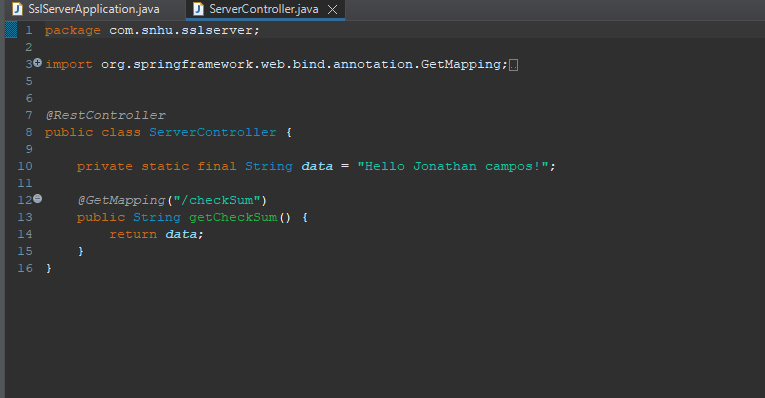




## Functional Testing

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





## Summary

When I was refactoring the code, I focused on improving the quality and the maintainability of the code. One way to add layers of security to the application is to follow the Vulnerability assessment process flow diagram. This allows you the developer to identify and analyze potential vulnerabilities. One way to identify any possible vulnerability is to perform code reviews. This usually requires a senior developer to review your code to ensure it is in compliance with the companies’ guidelines. If a issue is found the vulnerability should be discussed to estimate the potential impact it may have on the application. The next step is to develop a plan to address and release a patch that will resolve any issues that may causes a attacker to exploit the vulnerabilities. It is also vital to monitor the application after a patch to ensure you did not break any other component of the application.

## Industry Standard Best Practices

A best practice that was implemented in this project was the use of secured frameworks, libraires and the use of OWASP dependency checker. Regular testing and monitoring is also an important best practice for maintaining the security of a software application. This can include conducting regular security assessments to identify potential vulnerabilities, and implementing a system for monitoring the application for signs of unusual or potentially malicious activity. The value of applying these industry standard best practices for secure coding to a company’s overall wellbeing is significant. By implementing these best practices, a company can help to protect its sensitive data and systems from potential security threats.

References

*Owasp dependency-check*. OWASP Dependency-Check | OWASP Foundation. (n.d.). Retrieved December 13, 2022, from https://owasp.org/www-project-dependency-check/#:~:text=Dependency%2DCheck%20is%20a%20Software,identifier%20for%20a%20given%20dependency.

*NIST's encryption standard has minimum $250 billion economic benefit, according to New Study*. NIST. (2018, December 11). Retrieved December 13, 2022, from https://www.nist.gov/news-events/news/2018/09/nists-encryption-standard-has-minimum-250-billion-economic-benefit