****

# 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** | **06/14/2023** | **David Waid** |  |

## 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

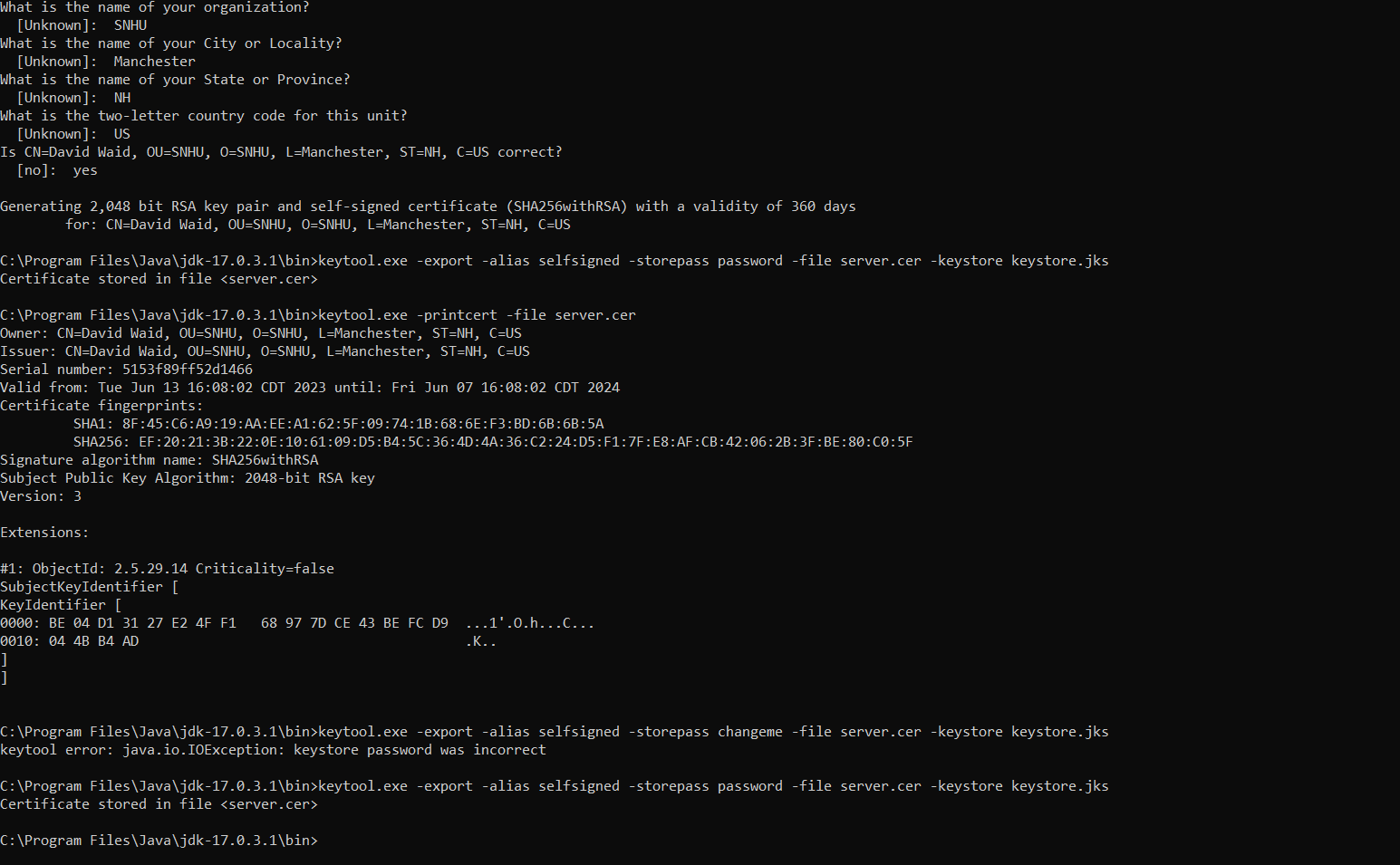
David Waid

## Algorithm Cipher

AES is my recommendation Artemis Financials' web application due to its strong security, widespread adoption, and extensive analysis. It is the standard encryption algorithm designated by the NIST and has proven resistance against known cryptographic attacks. AES operates on fixed-size data blocks, typically 128 bits, and employs mathematical transformations for encryption. It supports key sizes of 128, 192, or 256 bits, allowing flexibility based on data sensitivity. AES uses symmetric key encryption, where the same key is used for encryption and decryption. It has undergone thorough scrutiny by the cryptographic community and is supported by modern systems and libraries. AES ensures secure data protection for Artemis Financials' client information.

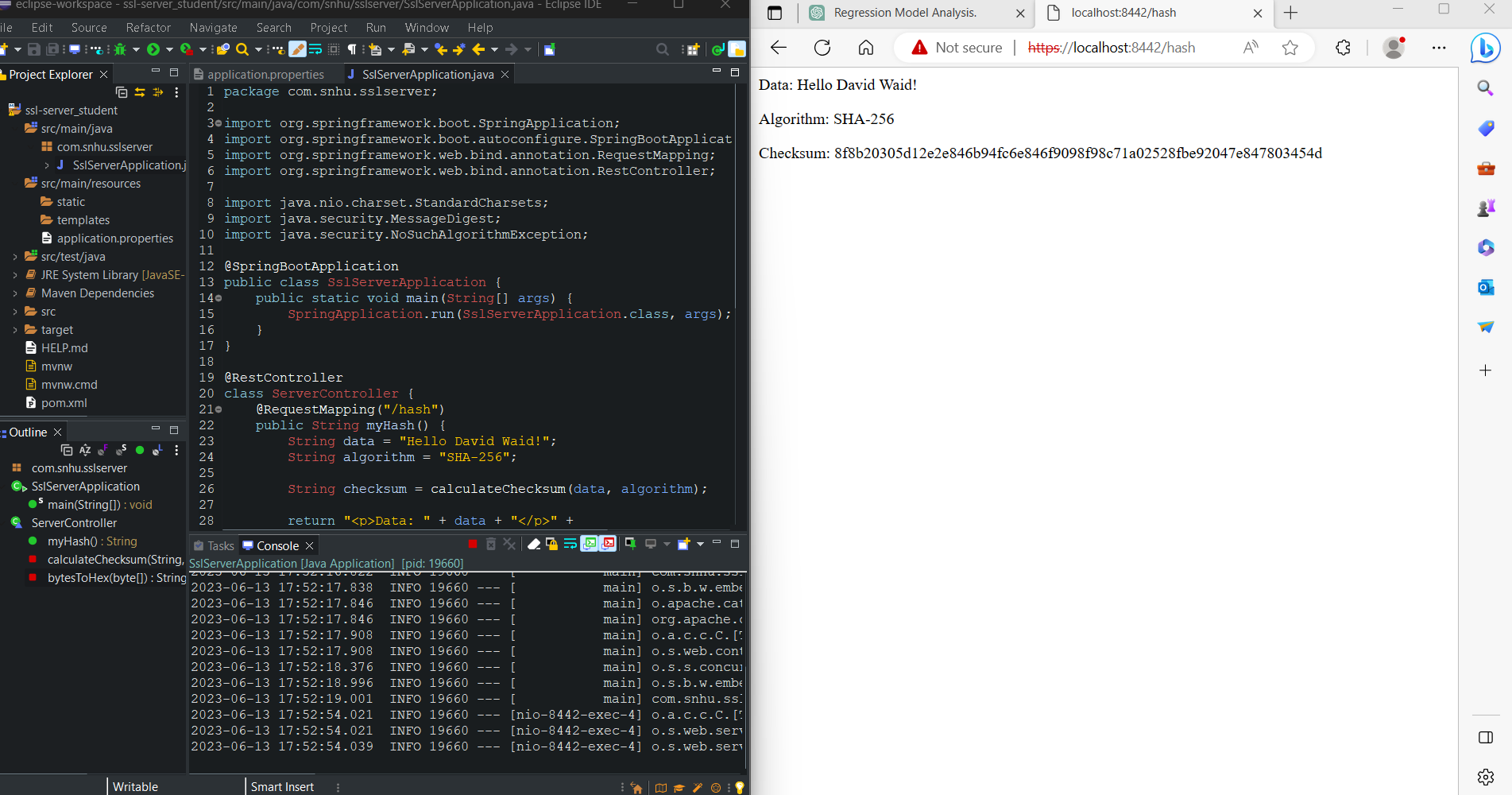
## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

Insert a screenshot below of the checksum verification.

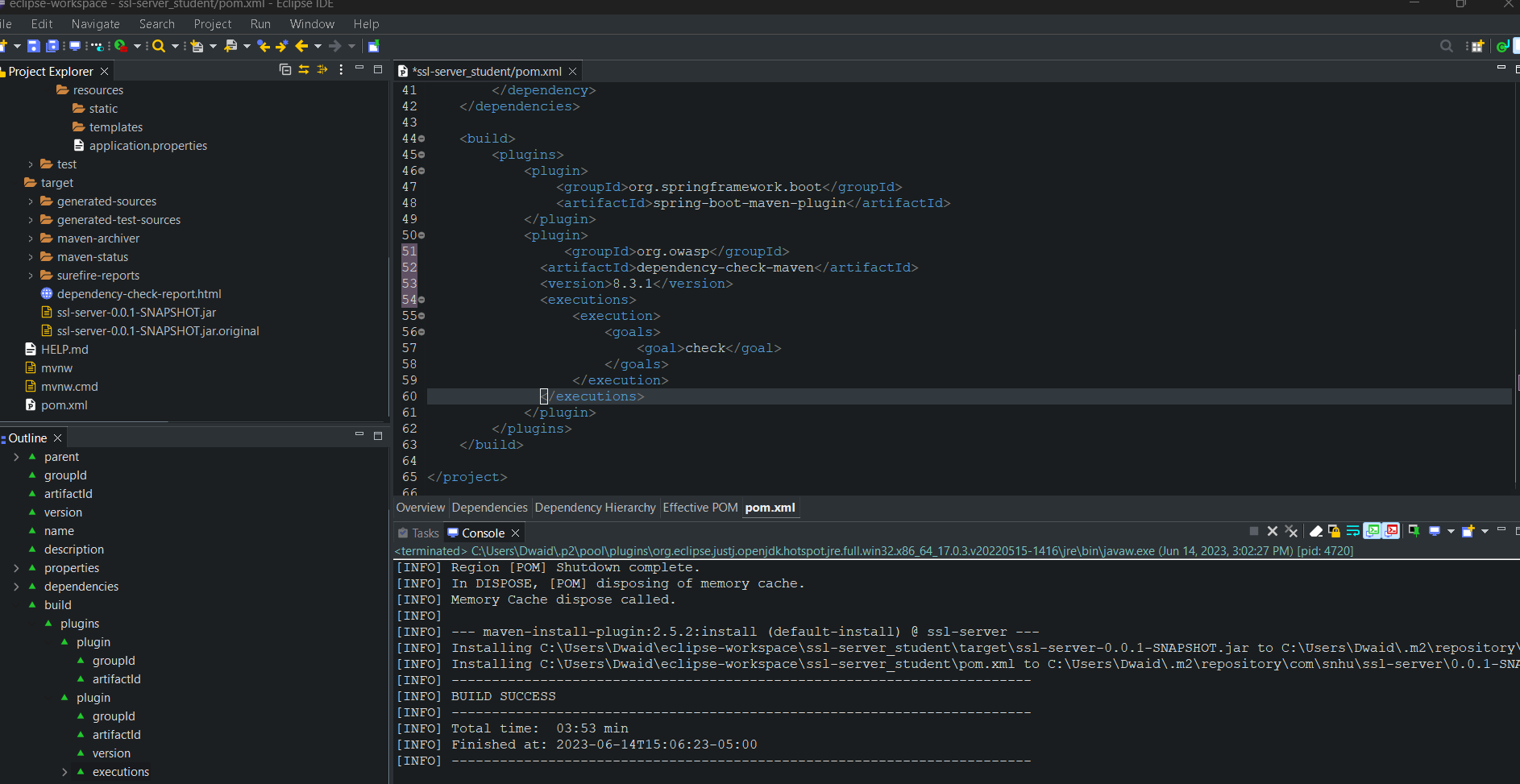


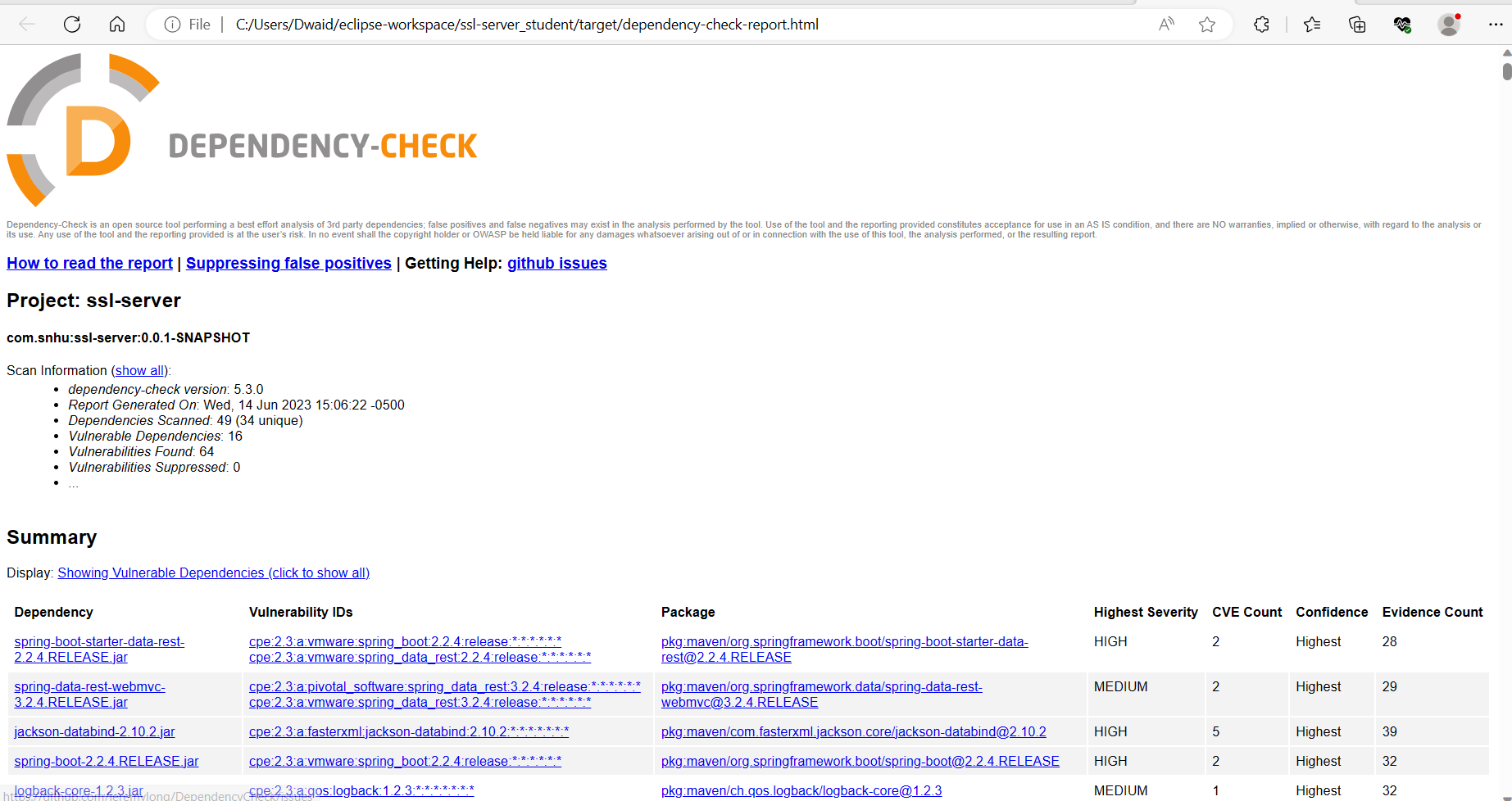
## Secure Communications

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

## Secondary Testing

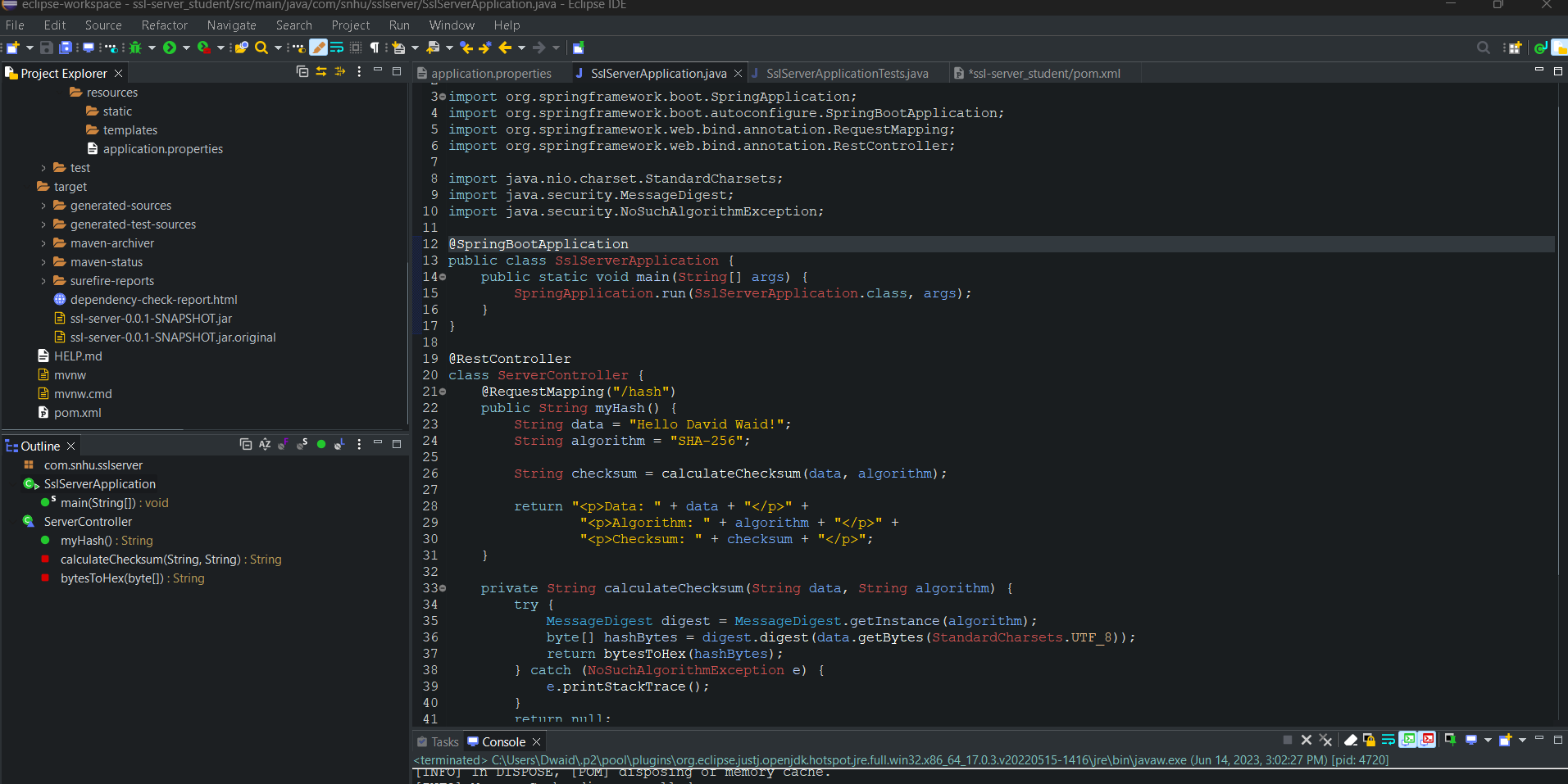
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

As the developer of the code, I have taken measures to address the identified vulnerabilities in the areas of cryptography, code error handling, and code quality. To ensure secure data transmission, I have implemented the SHA-256 algorithm for hashing data, providing integrity and protection against tampering. Error handling has been incorporated to catch the NoSuchAlgorithmException and print the stack trace, enabling effective troubleshooting and debugging. Adhering to coding conventions and employing appropriate Java classes and libraries ensures code readability and maintainability. By addressing these vulnerabilities, the code establishes a foundation for a more secure software application. However, it is important to continuously review and enhance security measures based on the specific application context and requirements to ensure ongoing protection and resilience.

## Industry Standard Best Practices

## In developing the code, I have applied industry standard best practices for secure coding to mitigate known security vulnerabilities. By utilizing secure cryptographic algorithms like SHA-256, I have ensured the current security of the software application, protecting sensitive data during transmission and reducing the risk of unauthorized access or tampering. Additionally, error handling mechanisms have been implemented to catch potential exceptions, providing valuable information for debugging and troubleshooting. Following coding conventions and using appropriate Java classes and libraries has ensured code readability and maintainability, reducing the likelihood of introducing vulnerabilities through coding errors.

## The value of applying industry standard best practices for secure coding is significant for the company's overall wellbeing. By adhering to these practices, the software application becomes more resilient against security breaches, data leaks, and unauthorized access. This enhances the company's reputation, builds customer trust, and ensures compliance with regulatory requirements. Moreover, the stability and reliability of the software are improved, reducing the potential for downtime, financial losses, and legal consequences.

## In addition to the secure coding practices, the use of Maven and its dependency management system has further enhanced the software application's security. Maven ensures that dependencies, including external libraries, are managed effectively, reducing the risk of incorporating outdated or vulnerable dependencies. By incorporating the dependency-check-maven plugin, vulnerability scanning of dependencies is performed, alerting to any known security vulnerabilities and enabling timely updates or alternative dependency choices.

## By implementing these industry standard best practices and leveraging Maven's capabilities, the software application's security is maintained, known vulnerabilities are mitigated, and the overall wellbeing of the company is safeguarded. These practices protect sensitive data, foster a secure environment, and contribute to the company's reputation, trust, compliance, and financial stability.