****

# 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** | **10/23/22** | **Kayvon Ghadiri** | **Original updates and changes to all fields** |

## 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

Kayvon Ghadiri

## Algorithm Cipher

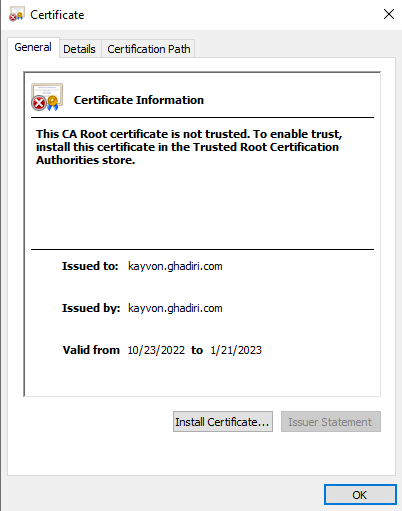
Based on what we are given; I think the best option we have is to apply the Advanced Encryption Standard or the AES for short. AES has been the standard in encryption for the US government for over 20 years, replacing Data Encryption Standard or DES. AES’s mostly used within public and private sectors and has also become known to be the golden standard.

The help ensure security remains fortified, we will also apply Secure Hashing Algorithms or SHA’s for short. SHA is a common hash function and when used simultaneously with AES, perpetrators and potential threats are vastly reduced. When it comes to AES, many different bits can be used, with the most common and secure one being AES-256.

AES uses the same key for both encryption and decryption making its keys symmetric. Since AES is symmetric, it will be easier for us to apply than an asymmetric encryption. AES does well at data block encryption as well as using expanded key lengths to remain secure. It is also important to remember that if the creator finds the key easy to remember, then it will also be easy for someone to think of and crack.

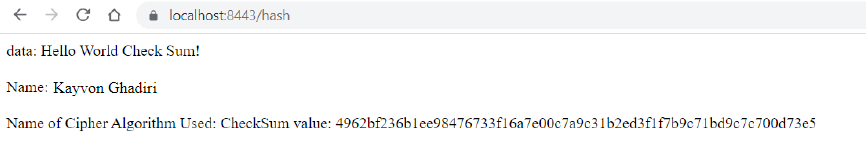
## Certificate Generation

Insert a screenshot below of the CER file.



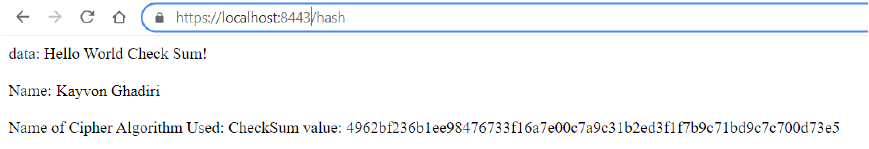
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

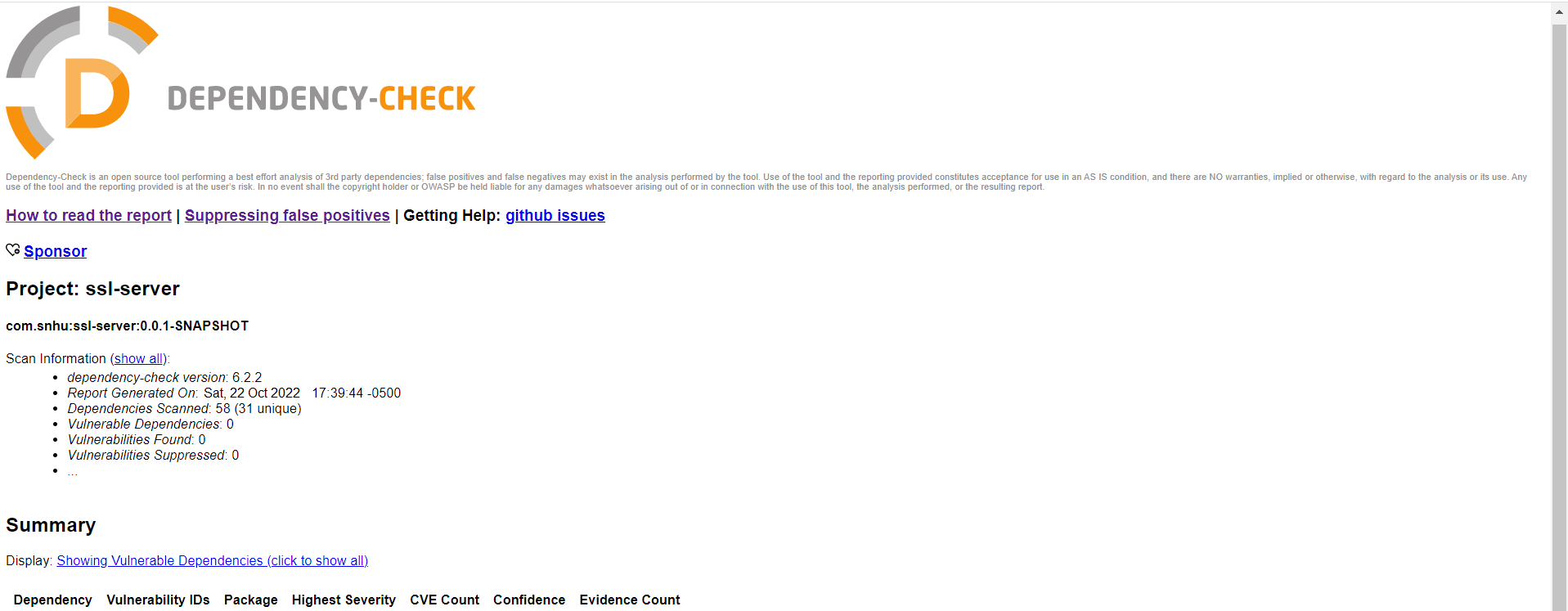
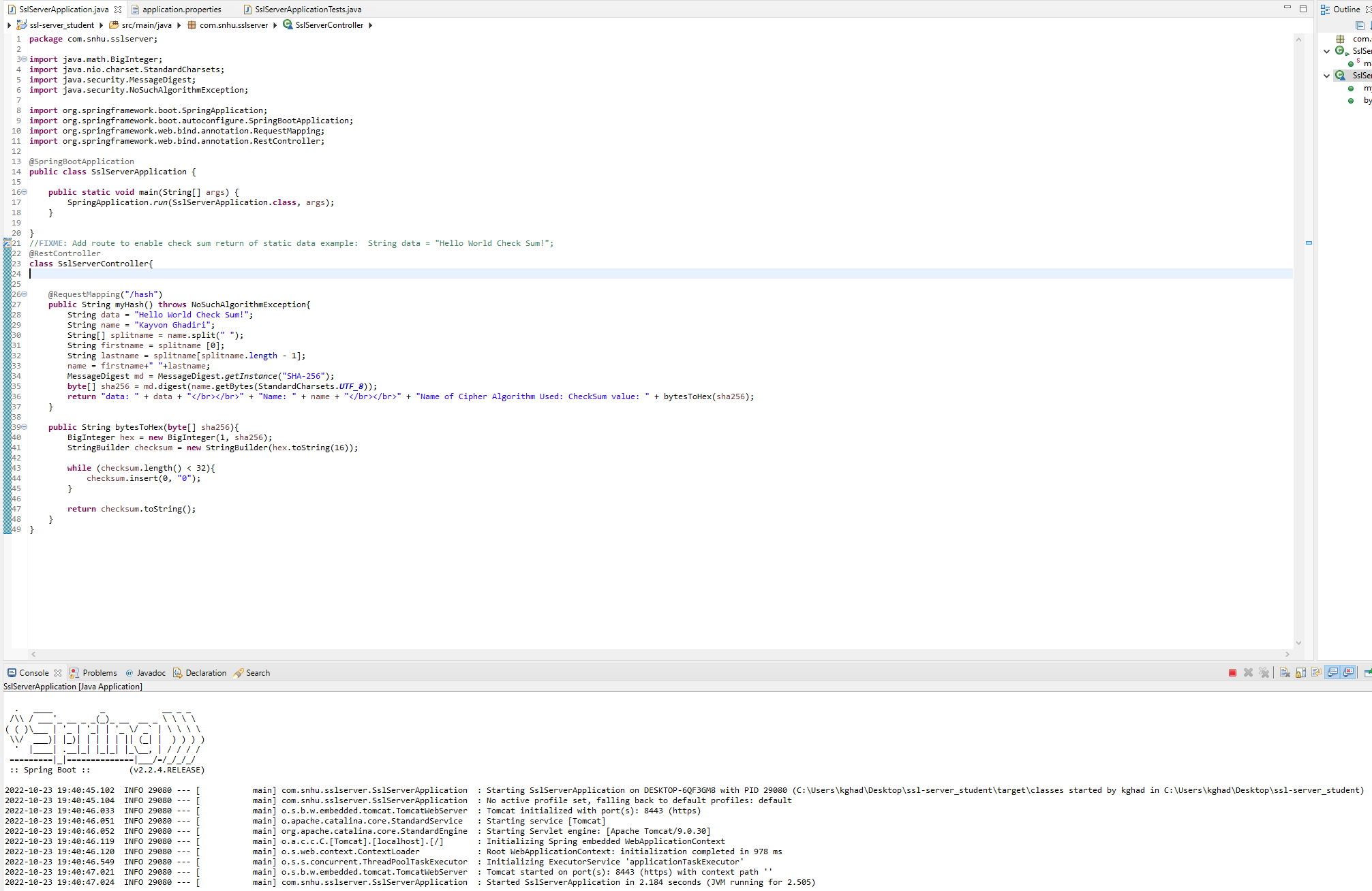
Insert a screenshot below of the web browser that shows a secure webpage.



Changed to https. Used <https://localhost:8443/hash> within a new window of the browser to show the secure connection.

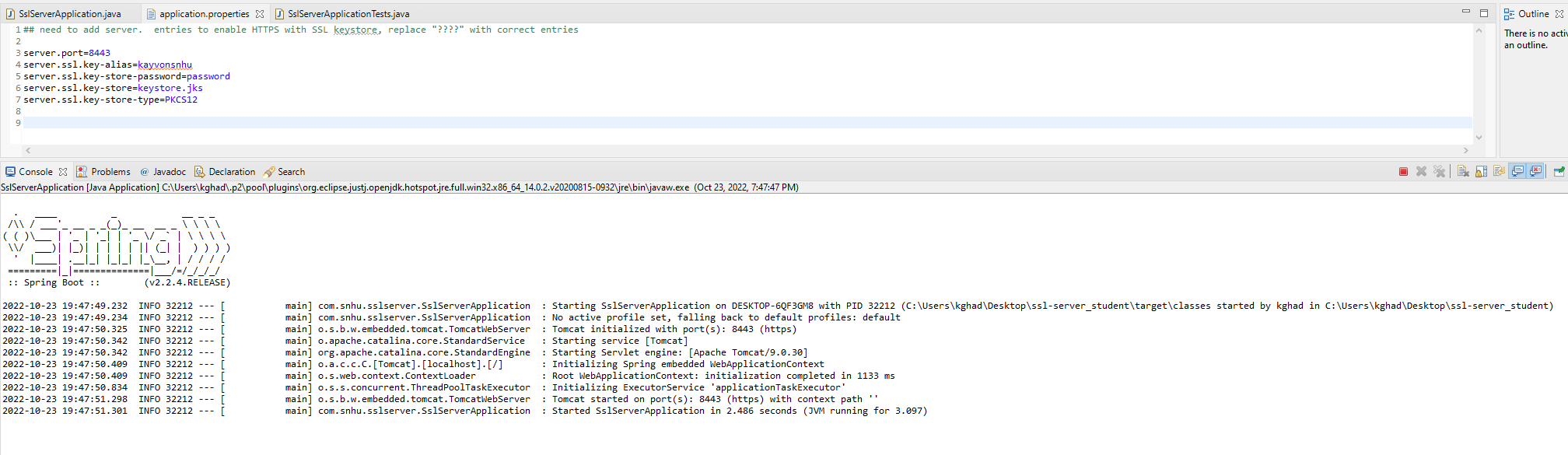
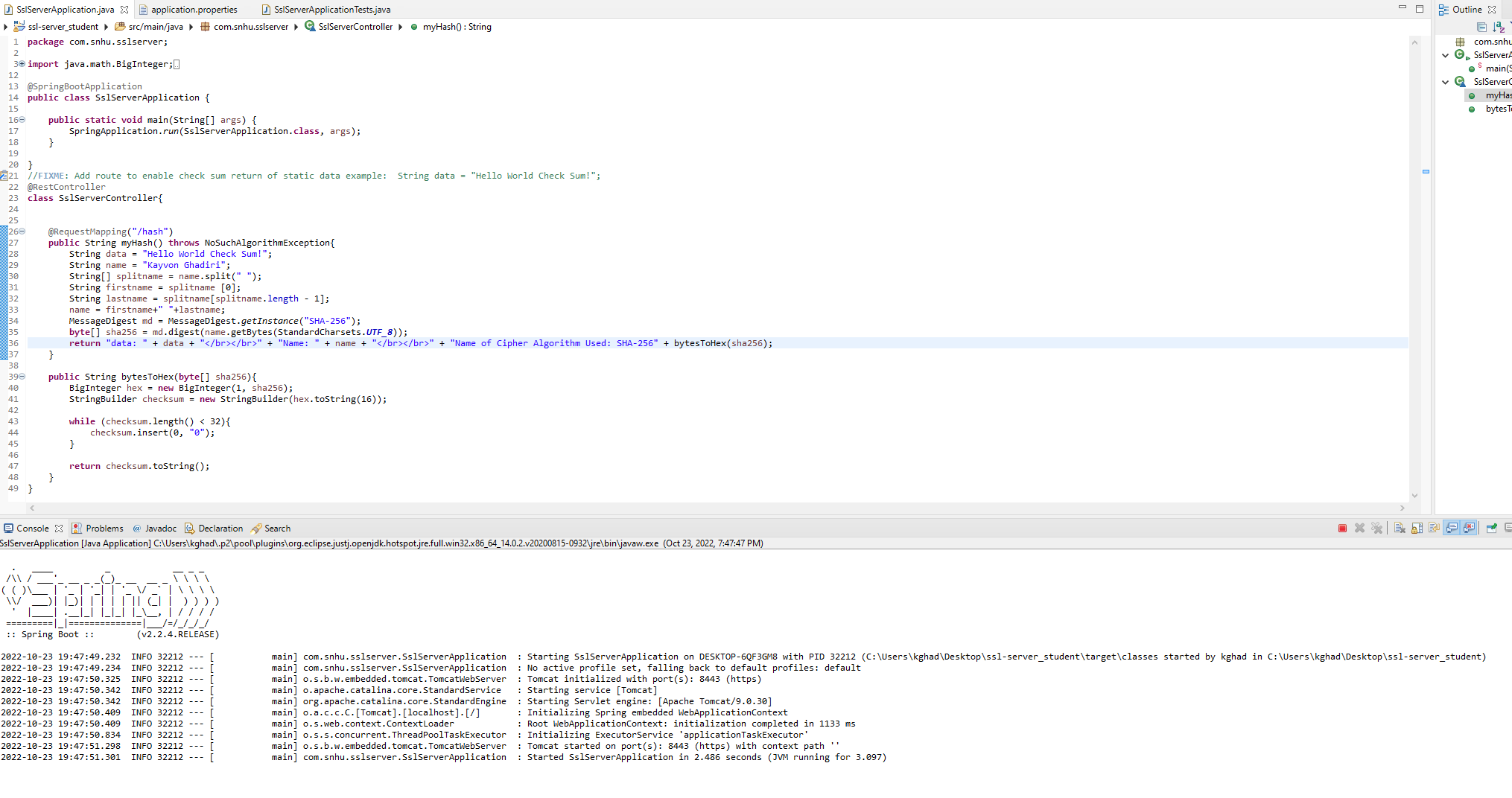
## Secondary Testing

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

One of the main changes we made was adopting self-signed certificates, which then leads to HTTPS becoming available. We then ensured all dependencies were highlighted and taken care of. All certificates were correctly created so HTTPS will be available when the application is complete. By securing the ability to use HTTPS, we can ensure that our website is secure and thus our users will feel more secure. Hashing was confirmed as functional within the checksum as well. Hashing our users data will also further our protection for our customers and our application. We also made sure all vulnerabilities were patched up and taken care of.

## Industry Standard Best Practices

Maintaining a product has to be one of the best practices you can do for your customers as with time bugs can appear, things can slow down, and vulnerabilities could arise. A standard practice is to field any of these concerns and issue patches to ensure the product remains current and at its best. Out of date systems also open the door to attackers, which further bolsters our need to keep updating our software. Another best industry practice is to only allow those who need higher access obtain that ability. At the end of the day, it is not a good idea to give everyone working for the company elevated access unless they absolutely need it. This will help keep things both organized as secure.