****

# 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** | **4/20/2024** | **Ryan Blegen** |  |

## 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

Ryan Blegen

## Algorithm Cipher

For this project I recommend using the SHA-256 algorithm. SHA stands for Secure Hashing Algorithm and was developed by the National Security Agency. This algorithm is one of the most widely used today and is often employed by financial institutions and crypto currencies like Bitcoin.

SHA-256 employs Java’s random number generator which creates non-reversible checksums. This ensures the validity of the file. SHA is a hashing algorithm and thus cannot be decrypted like an encryption cipher. SHA takes data and creates a fixed-size string of characters which is the value and digest. SHA-256 generates a 256-bit cypher. This means it can generate 1x10^77 possible combinations making it highly resistant to brute force attacks.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a certificate

Description automatically generated

## Deploy Cipher

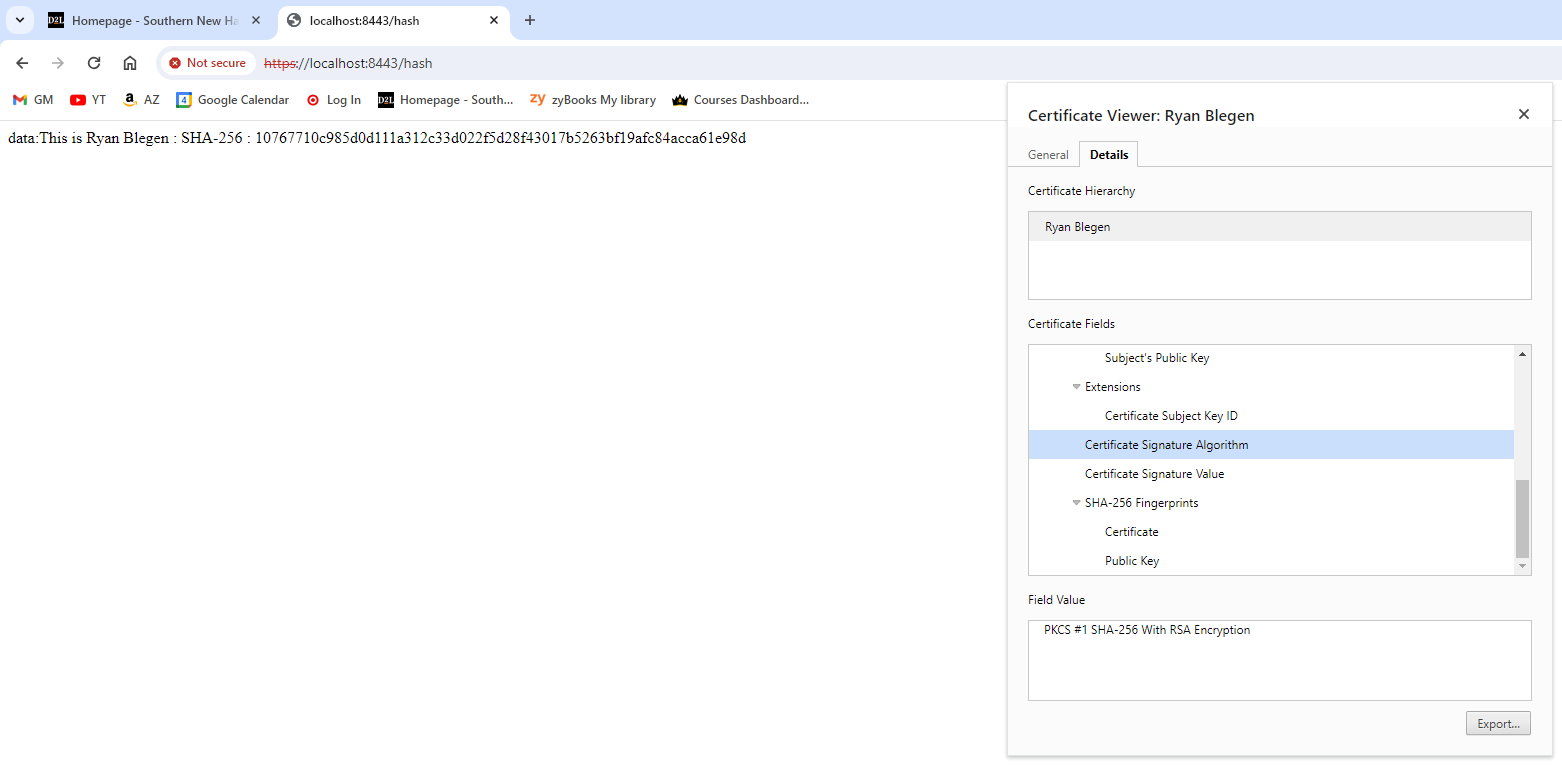
Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

## Secure Communications

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



As you can see the connection is encrypted via SHA-256 however chrome still shows the sight as not secure because the certificate is self-signed.

## Secondary Testing

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

A screenshot of a computer

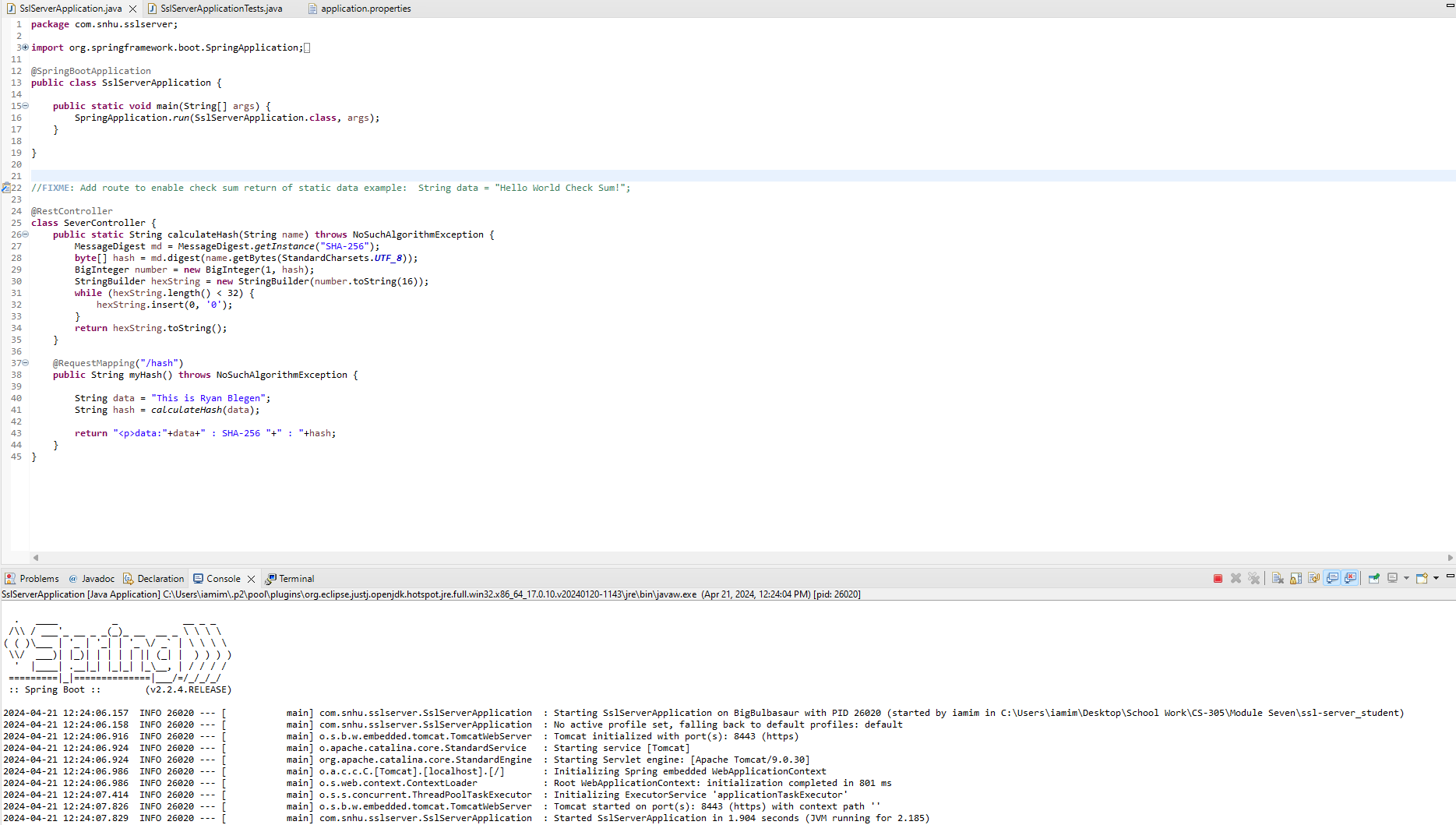
Description automatically generated

A screenshot of a computer program

Description automatically generated

## Functional Testing

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



A screenshot of a computer

Description automatically generated

## Summary

For this project I have addressed the following areas - cryptography, server/client communication, and code quality. To ensure the security of the program I utilized a 256-bit hashing algorithm to protect the sensitive data Artemis Financial deals with. Additionally, I have integrated a security certificate and enforced an SSL/TLS connection. Utilizing HTTPS data is encrypted in transit both ways offering more protection for the users. Having an SSL certificate gives legitimacy to the website.

There are multiple layers of protection for this project. The first layer of security is the SSL/TLS protocol. To implement the SSL/TLS protocol I incorporated a self-signed certificate and configured the application.properties. You can also obtain a certificate from a Certificate Authority. An HTTPS certificate ensures the authenticity of the website, giving them legitimacy to the user. With HTTPS data is encrypted in transit both ways. This ensures the personal financial data Artemis Financial deals with remains secure.

The second layer of security I used in this project is a 256-bit hashing algorithm with SHA-256. SHA-256 is a hashing function that takes an input and produces a fixed size output. It takes data and outputs a string of 64 hexadecimal characters. SHA-256 is not an encryption algorithm and thus can not be decrypted to its original data.

## Industry Standard Best Practices

Applying industry standard best practices is essential for a product’s overall wellbeing. Secure coding practices help identify and prevent vulnerabilities. By addressing security flaws early in the process, you reduce risk and potentially increased cost. Secure applications build better trust with customers. Following industry standard best practices can result in cost savings. Secure coding practices reduce the need for costly emergency patches or service downtime. It can add to the long-term sustainability of the product. Companies with robust security practices can better adapt to new and changing threats. Secure coding practices reduce risk, legal exposure, increase consumer confidence, increase savings and add to long-term success.