# CS 305 Module Five Coding Assignment Checksum Verification

## Instructions

Using the instructions from theModule Five Coding Assignment Checksum Verification Guidelines and Rubric, replace the bracketed text with the relevant information in your own words.

## Algorithm Cipher

To ensure secure communication and data verification using a checksum, I would recommend using a combination of encryption and hash functions using the Advanced Encryption Standard Algorithm (AES), with 256-bits and the Secure Hash Algorithm 256-bit (SHA-256). To summarize them:

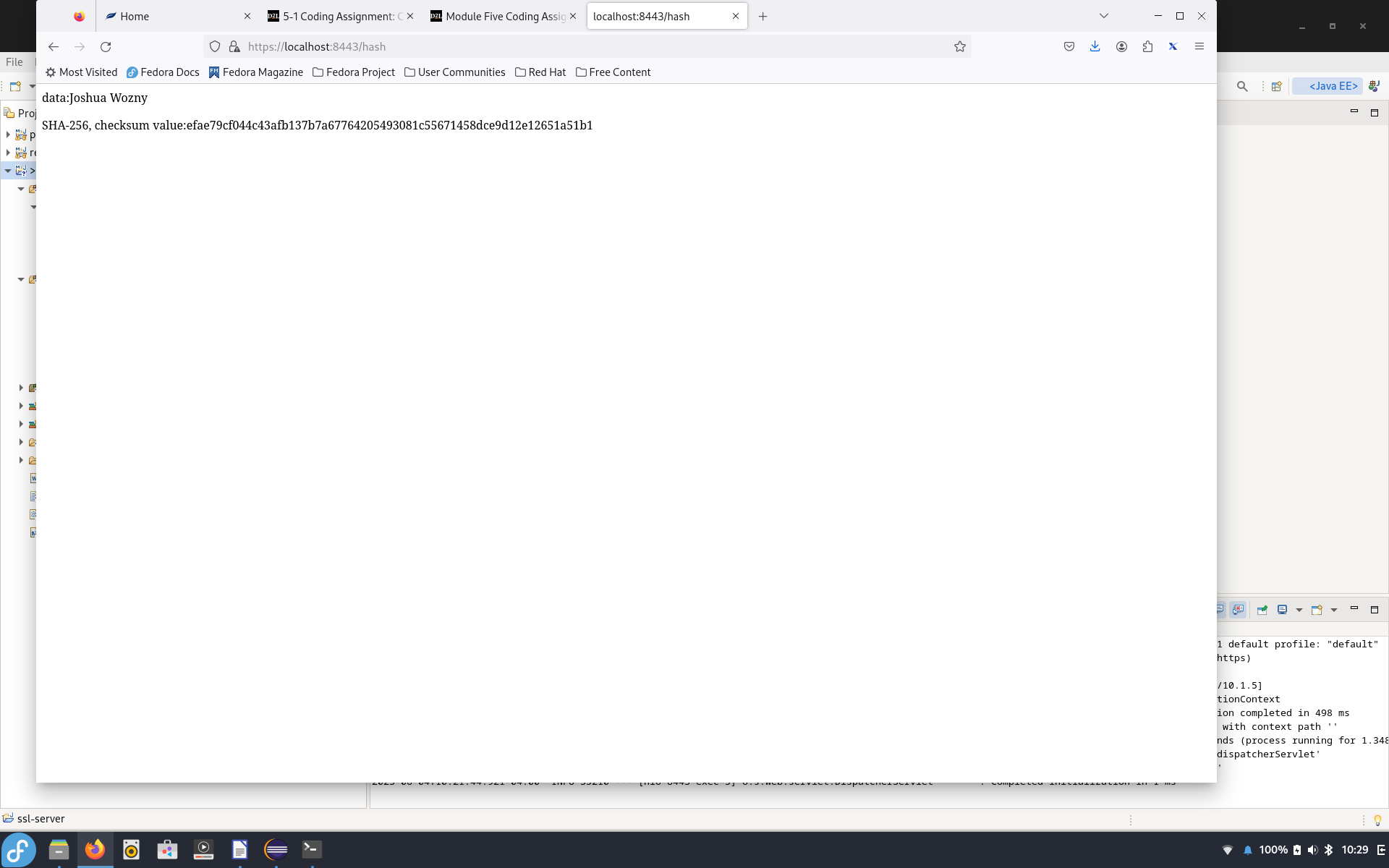
## Justification

* AES is a widely used symmetric encryption algorithm that provides a high level of security. It supports key sizes of 128, 192, and 256 bits and is considered secure for most applications. You can choose a key size based on your security requirements (Java Security Standard Algorithm Names, 2017) . Block ciphers built as permutations, like AES, ensure that they are collision free. Collissions occur when a has value occurs for two or more different inputs, which would limit accurate decryption and reduce security overall.
* SHA-256 is a cryptographic hash function that generates a fixed-size output of 256 bits. It is widely used for data integrity checks and can provide a high level of assurance that the data has not been tampered with (Ramesh Fadatare, 2020).

## Generate Checksum

Submitted 6/4/23 with this document.

## Verification



REFERENCES

Java Security Standard Algorithm Names. (2017). Oracle.com.

[https://docs.oracle.com/javase/9/docs/specs/security/standard-names.html#cipher-algorithm-names](https://docs.oracle.com/javase/9/docs/specs/security/standard-names.html" \l "cipher-algorithm-names)

Ramesh Fadatare. (2020, February 25). Java SHA-256 Hash With Salt Example. Javaguides.net; Blogger. <https://www.javaguides.net/2020/02/java-sha-256-hash-with-salt-example.html>