# CS 305 Module Two Written Assignment Template

## Instructions

Replace the bracketed text with the relevant information in your own words. If you choose to include images or supporting materials, make certain to insert them in all the relevant locations in the document.

## Areas of Security

* Dependency Management
* Authentication and Authorization
* Data Encryption
* Patch Management

## Areas of Security Justification

* Dependency Management:
  + ensure the dependency tools can accurately identify outdated or vulnerable libraries within the project. All third-party dependencies need to be updated to the newest version while remaining compatible with one another.
* Authentication and Authorization:
  + The project needs to remain secure and only available to authorized users, these users need authentication to gain access to the code base as well. Insecure libraries in the dependencies that are outdated can cause data breaches.
* Data encryption:
  + If any libraries have vulnerabilities that directly impact encryption features it puts data at risk.
* Patch Management:
  + Keep dependencies up to date and compatible with libraries and other code usage modules.

## Code Review Summary

There are several vulnerabilities in key libraries that need to be addressed using the areas of security. Log4j, Spring Framework, and Jackson Databind need patched and upgraded to newer versions. MongoDB driver and SnakeYAML have moderate risks that can be solved by proper configuration as well. By updating these dependencies, applying security patches, and filtering out false positives, we can ensure the application is reliable and safe.

## Mitigation Plan

* Dependency Management:
  + Regularly update dependencies to their most recent versions.
    - Upgrade Spring Framework
    - Replace log4j-api since it is a depreciated library and use an alternative.
    - Automate dependency scanning in CI or CD pipelines so that vulnerabilities are caught early on.
    - Remove dependencies that are not being actively used.
    - Apply version constraints to dependencies so that only approved versions are used in the project.
* Authentication and Authorization:
  + Restrict access to dependency files so that only authorized users can see them, and they need to go through an authentication process to do so.
  + Harden configuration for libraries Spring Security to ensure authentication is mechanized with the most recent versions.
  + Use token-based APIs to implement OAuth 2.0 or JWT tokens for APIs that restrict unauthorized access.
  + Use role-based access control that enforces access policies where only specific roles can modify or execute configurations.
* Data Encryption:
  + Secure communication channels using HTTP or TLS for every method on vulnerable dependencies Spring Web and Jackson Databind.
  + Encrypt configuration files.
  + Ensure backward compatibility for libraries with these encryption features so that they are compatible with upgraded versions.
* Patch Management:
  + Address dependencies flagged with a Critical headline immediately (Spring Boot and Spring Web).
  + Patch automation for vulnerabilities before anything is pushed out.
  + Monitor the CVEs to ensure no new vulnerabilities pop up.
  + Ensure vulnerabilities are responded to immediately where frameworks such as Spring have high severity outlooks.