**Software Design and Engineering (SQL Injection Enhancements)**

This artifact demonstrates my ability to use well-founded and innovative techniques, skills and tools in computing practices and identify security vulnerabilities in applications. Specifically, this artifact reflects my ability to identify vulnerabilities to SQL Injection and put code in place to prevent such a vulnerability from being exploited. This artifact stretched my thinking about how code should be constructed to ensure the most efficient application possible, while also considering the most secure way to write that application. This required an evaluation of the program from bad actor’s perspective and determining the ways that an attempted attack on the system might be carried out.

Initially, this application handled potential SQL injections for strings, alpha, and numeric injections. However, upon further review, additional vulnerabilities were identified in the area of alphanumeric injections. These injections were not previously handled in the application and left the system vulnerable to an additional SQL injection method. This particular artifact is a strong example of my understanding of both C++ coding language and security principles. To identify the scenarios that might present an opportunity for a security flaw, understand how to remedy those, and then write the necessary code to do so is a strong demonstration of my understanding of these principles.

In this artifact, I demonstrated an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals by completing the enhancement to the SQL injection code, adding the check for alphanumeric injection to the existing string, alpha, and numeric checks. Also, in this artifact I developed a security mindset that anticipated adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigated those design flaws, and ensured privacy and enhanced security of data and resources by completing the enhancement related to alphanumeric SQL injection.

**Algorithms and Data Structures (Activity Trees)**

This artifact highlights my ability to design and evaluate computing solutions that solve given problems using algorithmic principles. Specifically, this artifact develops a mechanism for an activity tree. This tree evaluates user input against a current position and adds nodes to the left or right of the current position based on the comparison between user input and current position. This artifact stretched my evaluation of how systems would handle user inputs and the way that the solution would need to evaluate that input against a current position.

The existing functionality of this solution included scenarios for the algorithm to progress down either the right or left side of the activity tree depending on if the user input was greater than or less than the original position. Upon looking closer at this solution, I became aware that there was no error handling in place for a NULL user input. If a user mistakenly submitted the form without inputting a digit for evaluation the application had no logic to handle the situation and would have crashed. The enhancement I made in this scenario was to add error handling for the situation where a user made no input. The system would evaluate the NULL input and return to the user for an input without evaluation. This prevents the system crash and does not allow the application advance to the evaluation mode of the activity tree.

In this artifact, I designed and evaluated a computing solution that solved a given problem using algorithmic principles and computer science practices and standards appropriate to the solution, while managing trade-offs involved in design choices by enhancing the activity tree’s error handling capability in the case of a NULL input from a user. This particular artifact highlights the skill necessary in evaluating a solution for negative scenarios and then implementing solutions that handle those scenarios without negatively impacting the overall performance of the system.

**Databases (Employees Database)**

This artifact highlights my ability to evaluate a database solution and apply well-founded and innovative techniques, skills, and tools to implement a solution that delivers value. The artifact manages and “Employee” database that contains tables for “Employee” and “Department” information. These tables provide details on employee specific information and demographics as well as data on the different departments that exist within the organization.

Initially, this artifact demonstrated a number of SQL commands and queries to modify tables, insert data, and query data. In one instance of SQL joins, the query was written long hand with table\_name.column\_name schema for the query. This made the query both cumbersome to read but also to manage. The enhancement I made to this solution was to implement the industry-standard best practice of creating an alias for the tables in the join so the query was more simple to read and manage. By creating an alias of “A” for the Employee table and “B” for the Department table, the query is optimized for performance and usability.

In this artifact, I demonstrated an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals by completing the enhancement to improve query performance through table alias names. Each of the queries and commands required specific knowledge of SQL and database structure and concepts. However, the implementation of industry-standards of aliases for table names demonstrates a greater understanding of the concepts required to successfully navigate a database solution through structured query language.

**Project Goals**

The combination of enhancements made to these three artifacts also demonstrate capability in two other key areas. By fully commenting all coding solutions and providing background for each of the solutions in the headers, these solutions employ strategies for building collaborative environments that enable diverse audience to support organizational decision-making in the field of computer science. Critically, these enhancements to comments make it possible for development resources to open these solutions and, at a glance, have a basic understanding of form and function for the solutions. This eases the transition between resources that may need to complete work on these modules.

Lastly, these three artifacts in combination demonstrate the ability to design, develop, and deliver professional quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts by providing a solid written narrative of the initial function of each, the evaluation of the solution for enhancement, and then the description of the enhancement and explanation for why it was made. Each enhancement was technically sound with solid explanation of why the changes were necessary and how they made the code better.