The artifact is a SQL Injection program. The program is intended to catch SQL Injections in progress, prevent them by blocking them, and then display a message that there was a suspected SQL Injection. This artifact was in a secure coding course I took in 21EW1. The reason why I selected this artifact was that it focused on security and creating secure code. It's crucial in today's society to develop and have secure code to help protect sensitive information and prevent vulnerabilities. This artifact's component that showcases my skills and abilities is my correct use of the escape all user input method. I was able to implement the technique correctly and prevented an SQL Injection. This shows my security mindset skills in anticipating future vulnerabilities and then creating prevention methods. A hacker can use malicious code to exploit the software; therefore, we must implement secure code when designing the software. Therefore, we must always design with security in mind from the beginning.

My skills show that instead of waiting for a vulnerability to happen and then fixing it, I incorporated secure code throughout my program. I accomplished this by using a SQL prevention method called "Escape All User Supplied Input." The escape all user supplied input works by escaping the users' input before it is put into the query. This helps block and prevent SQL Injections from happening, preventing attackers from stealing confidential data and stopping data breaches. My skills and abilities with the artifact showcase my software design, security mindset with enhanced security, and my problem-solving skills using data structures and algorithms to ensure the software does not get exploited.

The artifact was improved by adding comments for lines of code that explained their purpose, the added description of the use for the program, and explained what the escape all user input method is. Finally, I added more escape all user method code to the program to cover other data areas, like ID and PASSWORD. Before, I only had created code for the NAME portion of the data. I was able to meet most of my planned enhancements. However, I did have trouble implementing other SQL Injection prevention methods, like the use of prepared statements and use of stored procedures.

The process of enhancing this artifact showed me the importance of secure coding and that this is an area I need to continually work in to build my skills and knowledge in creating secure code. It's essential to know all possible SQL Injection prevention methods and to be able to implement them all successfully. The main challenge I faced enhancing this artifact was implementing a different SQL Injection prevention method.

\*The program before coding a prevention method. The below picture shows a successful SQL Injection.\*

Text

Description automatically generated