## **1. Briefly describe the artifact. What is it? When was it created?**

This is a SQL injection demo program from CS 405. It creates a database with user info and shows how SQL injection attacks work. The original code tried to block injections by checking for OR statements but that wasn't enough.

## **2. Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

I picked this because database security is important and lots of websites get hacked through SQL injection. The original only checked for OR which is too basic. I improved it by adding better detection that looks for more attack patterns like 1=1 and comment markers. I also added a prepared statement function that uses parameterized queries which is the right way to stop injection. Plus I added input validation to check usernames only have safe characters.

## **3. Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

Yes this meets the database outcome. It shows I understand SQL injection and how to prevent it with prepared statements. No changes to my plans.

## **4. Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

I learned that blocking SQL injection is harder than just checking for one keyword. Attackers can use lots of different tricks. The hardest part was figuring out how to use prepared statements with SQLite in C++ because most tutorials are for PHP or Java. I had to look up sqlite3\_prepare\_v2 and sqlite3\_bind\_text functions and test them a bunch of times to get it working. Also learned that input validation is important as another layer of protection. It was cool to see the injection attempts get blocked when I ran the program.