**My artifact**

For my Databases artifact, I chose to use a Java based console application that adds and tracks service animals, specifically dogs and monkeys. I created this originally in IT 145, and it enabled users to add in, store, and find animal information like their breed/species, training status, and service country. The original code used in-memory to store the data in array lists, which led to the data being lost and recreated every time it was opened. This led to an application that was not scalable as a solution or user friendly.

**Justification**

I chose this artifact because it could be improved across all three categories, which made it a good indication of my skills and ability to grow. This enhancement shows my ability to create and integrate a SQLite database into a Java application so the code would be easier to scale and easier to maintain. The enhancement added an animal\_rescue.db which replaced my array list of rescue animals that can be updated and saved even when closing and reopening the application. The tables also incorporate fields that are only specific to a single animal rather than both, which it was not doing previously. The ability to insert, select, and delete records makes it user friendly. The code is also commented and organized to be easy to maintain.

**Course Outcomes**

* Demonstrate 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.
* Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

I feel that I met the course outcomes by displaying my ability to use SQLite and connect to Java, which uses industry-standard use of databases for efficiency. This integration adds practical value to the customer and gives it more functionality. As the shelter increases in size and depth, they will be able to easily maintain their records. My focus wasn’t on security primarily for this enhancement, I did take that into consideration when writing my code. I used prepared statements instead of strings to reduce the risk of SQL attacks.

**Reflection**

I learned a lot through my integration of SQLite into my java application. I hadn’t done that before, and it was much more difficult than I thought it would be. Originally, I was using in-memory storage with array lists to hold my rescue animal names and attributes so I had to do a lot to use a relational database. I had to create a schema, a table, connect SQLite to Java, and check that the user input matched the schema. One of my biggest struggles was with the set up of the project. Since I hadn’t done this type of integration before I accidentally placed my animal\_rescue database inside my src files which caused all sorts of problems since my path was looking for the database in the project folder. Eventually, I realized that my path was pointing incorrectly and adjusted the folder hierarchy to make it work. I had a similar issue with where I placed the JDBC jar since that was also in src, leading to it not being found when trying to run the program, but I was able to resolve that at the same time.

Another challenging part of the integration was creating a schema that could use attributes from both dog and monkey but would need to input null if the attribute did not apply. Initially I was just inputting values using a SQL input statement, but when I went through and used the intake method on my menu it didn’t work. Eventually after some trial and error, I finally settled on using default inputs and then using an else-if statement that checks if the attribute is needed and inputs ‘null’ if it's not. I also had to consider that the data had to validate so it could be consistent, which I had handled in a previous enhancement, but I went through and double-checked it to be sure.

