**CS 499 Milestone Four: Database Enhancement Narrative**

The selected artifact for database enhancement is the "Module5\_EncryptionCoding" project from CS-405: Secure Coding. This project initially focused on encryption and decryption of data using XOR-based transformations and stored encrypted data in text files. While functional, this method lacked secure storage, user authentication, and database-backed management.

This artifact is included in the ePortfolio to showcase my ability to integrate secure database solutions within software applications. The enhancement demonstrates my proficiency in implementing secure storage mechanisms for sensitive data, utilizing SQL databases (SQLite) to manage encrypted information, and ensuring data integrity and preventing unauthorized access.

To align with best practices in secure database management, several enhancements were made. A SQLite database (encryption.db) was introduced to replace plaintext file storage, and a new table encrypted\_data was created to securely store encrypted strings. Encrypted data is now stored securely within the database using parameterized queries to prevent SQL injection vulnerabilities. Retrieval functions were implemented to fetch and display encrypted data securely. Additionally, structured query functions were added to ensure safe data handling, and an SQL schema was designed to allow future expansion, such as user authentication.

This enhancement aligns with multiple CS program outcomes, including secure coding practices by ensuring encrypted data is stored securely and retrieved safely using SQL, database management through the application of SQL techniques for structured and persistent data storage, and software engineering by improving system architecture through the integration of a database backend for scalability and security.

During the enhancement process, I learned key principles of secure database management, including the importance of replacing plaintext storage with database-backed solutions, implementing SQLite efficiently in C++ projects, and protecting data against injection attacks through parameterized queries. One challenge was adapting the existing encryption logic to integrate seamlessly with database operations, requiring the restructuring of file-based I/O methods into secure database transactions.

Further improvements could include implementing user authentication with hashed passwords for access control, expanding database functionality with stored procedures for automated data processing, and developing a web interface for encrypted data management.

This milestone provided valuable experience in secure database integration, aligning with industry best practices for software development. The improvements showcase my ability to design, implement, and secure database-driven applications, reinforcing my competencies in secure software engineering.