**Artifact Overview:**The Secure Database Access Module demonstrates the implementation of multiple secure coding principles within a C++ application. It simulates CRUD operations with role-based access control, encryption for sensitive data, safe memory management via RAII, and secure logging practices. This module serves as a reusable template for building secure database-backed applications.

**Learning and Skill Development:**  
Creating this artifact reinforced practical understanding of secure C++ coding standards. Input validation, encryption, and RBAC were emphasized to prevent common vulnerabilities like SQL injection and privilege escalation. Integrating RAII and assertions improved memory safety and code reliability. Logging without sensitive data demonstrated careful auditing practices.

**Alignment with Course Outcomes:**  
The artifact addresses program outcomes related to secure software design, implementation, and risk mitigation. It provides evidence of the ability to apply defense-in-depth principles by layering multiple security measures and demonstrates adherence to Green Pace’s C/C++ standards.

**Challenges and Resolutions:**

* Designing RBAC logic that was flexible yet secure required careful consideration of permissions.
* Encrypting data in a way that was secure but also compatible with database storage demanded understanding of cryptographic functions.
* Ensuring all destructors were noexcept and resources were managed automatically helped reinforce proper RAII practices.

**Conclusion:**  
This artifact exemplifies secure software development in C++ while adhering to industry standards and organizational policies. It provides both technical functionality and documentation suitable for submission in an ePortfolio as evidence of secure coding competence.