CS 499 Milestone Four Narrative: Database Enhancement

Name: Caeli Firth

Date: July 29, 2025

Course: CS 499 Computer Science Capstone

# Artifact Description

The selected artifact for the database enhancement is a user preferences module originally built for a web-based travel planning application. This module was first developed in CS 340: Database Design and Management and later expanded during CS 465. It was designed to store user search preferences, such as destination filters, budget constraints, and accommodation types. The artifact was initially based on a JSON file structure for data storage and has now been migrated to a SQLite relational database to support greater scalability, query efficiency, and security.

# Justification and Skills Showcased

This artifact was chosen because it clearly showcases my skills in database schema design, data validation, and integration of backend logic for efficient data access. In the enhancement process, I designed normalized relational tables, implemented foreign key relationships, and wrote parameterized queries using SQL to prevent injection vulnerabilities. I also created CRUD operations in the backend to allow users to manage their preferences securely and efficiently. Furthermore, I included exception handling, ensured referential integrity, and added logging for data access operations.

# Course Outcomes Met

This enhancement supports the following CS 499 outcomes:  
• Outcome 3: Designing solutions using computer science practices and evaluating trade-offs in data design.  
• Outcome 4: Demonstrating innovative techniques and tools for implementing database solutions.  
• Outcome 5: Developing a security mindset by addressing SQL injection risks, input validation, and access control.  
These outcomes were part of my enhancement goal from Module One and have been fully achieved through this iteration.

# Reflection on the Enhancement Process

Enhancing this artifact taught me how critical it is to move beyond basic file storage toward structured, query-optimized relational models. I gained deeper insight into relational database theory and SQL best practices, including indexing and join optimization. One of the biggest challenges I faced was handling invalid input from users and preventing malformed SQL queries, which I solved using parameterized query statements and exception handling. Another key learning moment came when I realized that proper indexing could dramatically improve query response times. This process has made me more confident in designing scalable, secure, and maintainable database systems.