Category Three Enhancement Narrative

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

The artifact selected for the category of algorithms and data structures was the course planner project from CS-300 course (Data Structures and Algorithms). The projects focused on aiding the "ABC University" client on developing a program to aid academic advisors in assisting students. The program loads a file of course information, in this case a list of computer science courses, and allows advisors to print schedules, or search for a specific course. This artifact was created roughly halfway through the degree program.

Justify the inclusion of the artifact in your ePortfolio.

I selected this artifact for use as it was an applicable scenario that would benefit from the use of a database for this category, as well as a greater demonstration of growth. Overall necessary underlying fixes were required for this artifact to promote a security mindset. Utilizing database concepts, an SQLite database will be implemented and configured to further enhance the usability of the application. Other improvements include adding more advanced concepts to the methods related to the database. Lastly, the code was maintained structurally and documented through the addition of the aforementioned enhancements.

Did you meet the course objectives you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

The main objective of using innovative techniques for the purpose of implementing a solution that will deliver value is met through the inclusion and improvements made through the use of database concepts, while the objective of developing a security mindset is benefited through the security related enhancements.

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?

Enhancements were made to introduce a database to the course planner program. SQLite was chosen as the database library. After reviewing available options, this choice was selected for its ease of use, embeddability, and serverless nature. For the purposes of this project, embeddability was beneficial, as a database was to be added into an existing C++ application. It was somewhat challenging to setup and configure the necessary components throughout the file structure and properties of the project.

Otherwise, my previous experience working with databases in this program was with full stack development where MongoDB was used. In that scenario the database was not serverless and relied upon running prior. This factor also contributed to the choice of using SQLite.

I deliberated on if the purpose of the database should be but ultimately decided that it should be used for student planning. The database allows a student user to create a schedule of courses that persists over time, while the original application remains as a means to parse data files from various sources and store data temporarily.

I created multiple functions for the purpose of accessing and manipulating data stored in the database: display student menu, connect to database, create student schedule, access student schedule, add courses to student schedule, and remove courses from student schedule. Display student menu was created to display a new menu of options for the database. Connect to database attempts to connect to the database, while handling errors when unable to connect. Create student schedule creates a table in the database if one does not already exist, with error handling. Access student schedule displays the courses currently contained in the database table. The remaining functions, for adding and deleting courses from the database, are similar. A course number is

taken as user input, and an attempt is made to either insert or delete the course (if found from the course number) from the student schedule. Error handling is implemented in both.

Additionally, handling of the new menu choices is accomplished utilizing a nested switch case within the main menu's switch case. This posed a challenge initially, as I had to contend with some scope related errors.

Ultimately, the code enhancements for this category became the most modular and maintainable components of the application. I learned how to incorporate a created data structure into queries and integrate a database into an existing application. The choice to enhance this artifact was due to the project guidelines indicating that an artifact could be enhanced for multiple categories. The database utilized in the web application was guided by step-by-step instruction and there was less of a ceiling to reach. Where implementing a database into this artifact would mean researching and starting from scratch while embedding and configuring. I felt it was more challenging and rewarding.

Resources:

Dezio, G. (2018, August 22). *SQL using C/C++ and SQLite*. GeeksforGeeks. https://www.geeksforgeeks.org/sql-using-c-c-and-sqlite/

Ferrell, R. (2023, August 3). *SQLite IF EXISTS: A Clear Guide to Conditional Statements - SQL DOCS*. SQL Documentation. https://sqldocs.org/sqlite/sqlite-if-exists/#:~:text=To%20use%20the%20IF%20EXISTS%20clause%20when%20altering%20a%20table,it%20returns%20without%20any%20errors

Piry, S. (2023, January 17). *Introduction to SQLite*. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-sqlite/