Karina Jacuinde

CS 499 Computer Science Capstone

Instructor Joseph Martinez

November 28th, 2025

Milestone 4 Narrative

This artifact is my Course Catalog program, originally created a few months ago for my CS300 class. The original version read course data form a CSV file and stored it in a Binary Search Tree. Since then, I've been enhancing it for my ePortfolio by translating it to Python and reorganizing the code and restructuring the use of data structures to make it cleaner and easier to understand.

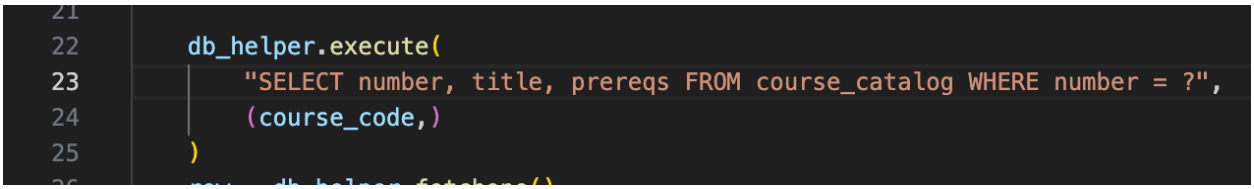
I selected this artifact because it represents all three areas of the ePortfolio requirement: software design, data structures and algorithms, and now databases. For this week's enhancement, I added a SQLite database so the program can store course information permanently instead of only relying on the CSV file. On the first run, the program reads the CSV and saves everything into the database. After that, it loads directly from the database, which makes the program faster and more realistic.

```python
data.py > ...
def load_data(file_name):

    # Try to read from DB
    db_helper.execute("SELECT number, title, prereqs FROM course_catalog")
    rows = db_helper.fetchall()

    # If table has data
    if rows:
        for number, title, prereq_text in rows:
            if prereq_text:
                prereqs = [p.strip().upper() for p in prereq_text.split(",")]
            else:
                prereqs = []

            course = Course(number, title, prereqs)
            courses_by_number[number] = course

        return courses_by_number

    # If DB empty, use CSV and fill DB
    with open(file_name, newline='') as csvfile:
        reader = csv.reader(csvfile)

        for row in reader:
            if not row:
                continue

            number = row[0].strip().upper()
            title = row[1].strip()
            prereqs = [p.strip().upper() for p in row[2:]]

            course = Course(number, title, prereqs)
```

I also updated the option that displays course information, so it now pulls the course details form the database. The recursive prerequisite feature still uses the dictionary as that is where I implemented it originally, this way the code stays simple and easy to follow. Overall, this enhancement shows that I can work with Python's database tools and built-in methods, create and manage tables, and integrate SQL with my existing program.

This enhancement helped me meet the course outcomes I planned for this milestone. By adding SQLite, I showed that I can use professional tools and techniques to store and manage data in a reliable and real-world way. This directly ties to the outcome about

designing computing solutions with the right standards and practices and implementing solutions that deliver value. I also met the security mindset outcome, because this enhancement uses parameterized SQL queries, to prevent SQL injections, and it builds on the input validation and error handling I added in earlier milestones.



Now that I've completed all three enhancements, I feel confident that after reviewing and polishing my work, I will have met all the course outcomes outlined for this capstone.